Concepts & Examples ScreenOS Reference Guide

Release 6.3.0, Rev. 01

Juniper Networks, Inc. CA 94089

1194 North Mathilda

Avenue Sunnyvale , USA 408-745-2000

Revision 01
Published: 2009-08-21

Revision History
August 2009—Revision 01

END USER LICENSE AGREEMENT

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at http://www.gnu.org/licenses/gpl.html, and a copy of the LGPL at http://www.gnu.org/licenses/lgpl.html.

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous

agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

Table of Contents

Part 5 Virtual Private Networks

Chapter 35 **Routing Information Protocol** **1307**

Chapter 40 **Protocol Independent Multicast** **1425**

Part 12 WAN, DSL, Dial, and Wireless

Chapter 58 Wide Area Networks 1869

List of Figures

Part 11 High Availability

Part 12 WAN, DSL, Dial, and Wireless

List of Tables

Part 1
Overview

Chapter 1

About the Concepts & Examples ScreenOS Reference Guide

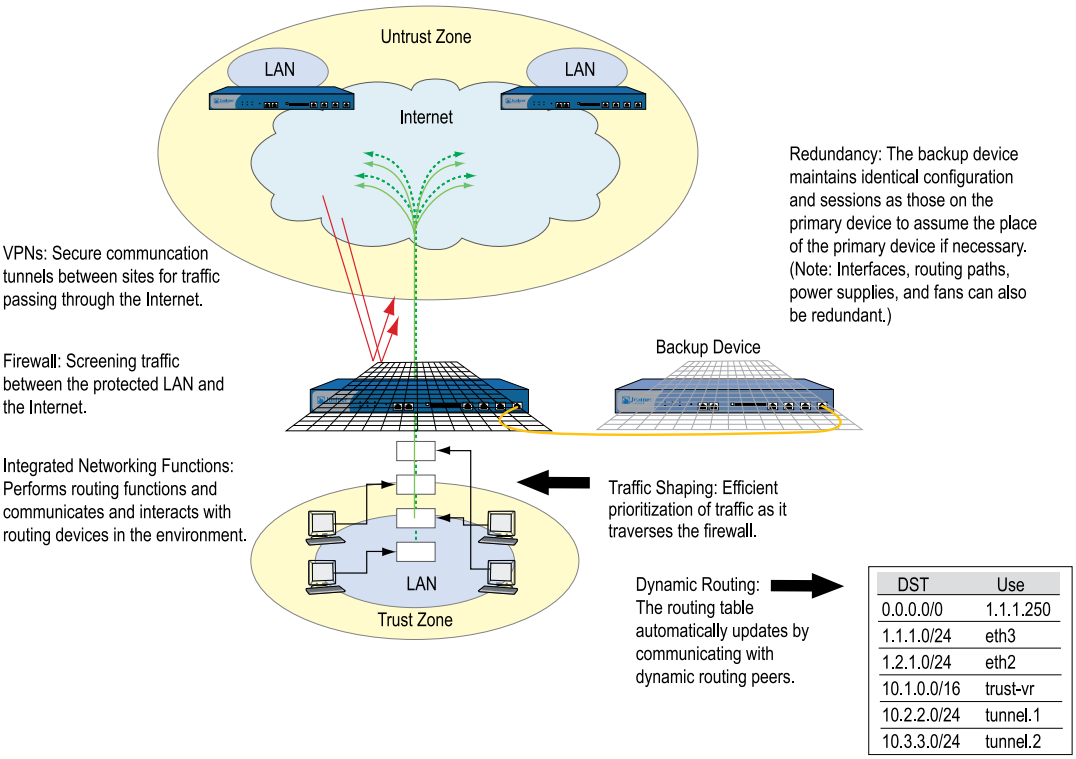
Juniper Networks security devices integrate the following firewall, virtual private network (VPN), and traffic-shaping features to provide flexible protection for security zones when connecting to the Internet:

- **Firewall:** A firewall screens traffic crossing the boundary between a private LAN and the public network, such as the Internet.
- **Layered Security:** The layered security solution is deployed at different locations to repel attacks. If one layer fails, the next one catches the attack. Some functions help protect remote locations with site-to-site VPNs. Devices deployed at the perimeter repel network-based attacks. Another layer, using Intrusion Detection Prevention (IDP) and Deep Inspection, automatically detects and prevents attacks from inflicting damages.

  Network segmentation, the final security layer (also known as virtualization), divides the network up into secure domains to protect critical resources from unauthorized roaming users and network attacks.
- **Content Security:** Protects users from malicious URLs and provides embedded antivirus scanning and Web filtering. In addition, works with third-party products to provide external antivirus scanning, antispam, and Web filtering.
- **VPN:** A VPN provides a secure communications channel between two or more remote network appliances.
- **Integrated Networking Functions:** Dynamic routing protocols learn reachability and advertise dynamically changing network topologies. In addition, traffic-shaping functionality allows administrative monitoring and control of traffic passing across the Juniper Networks firewall to maintain a network's quality-of-service (QoS) level.
- **Centralized Management:** The Network and Security Manager (NSM) tool simplifies configuration, deployment, and management of security devices.
- **Redundancy:** High availability of interfaces, routing paths, security devices, and—on high-end Juniper Networks devices—power supplies and fans, to avoid a single point of failure in any of these areas.

NOTE: For information about Juniper Networks compliance with Federal Information Processing Standards (FIPS) and for instructions on setting a FIPS-compliant security device in FIPS mode, see the platform-specific Cryptographic Module Security Policy document on the documentation CD.

Figure 1: Key Features in ScreenOS



The ScreenOS system provides all the features needed to set up and manage any security appliance or system. This document is a reference guide for configuring and managing a Juniper Networks security device through ScreenOS.

Part Organization

The *Concepts & Examples ScreenOS Reference Guide* is a multi-part manual. The following information outlines and summarizes the material in each part:

Part 1: Overview

- Provides a high level description of the contents for Concepts and Examples Combined Reference Guide .

Part 2: Fundamentals

- "ScreenOS Architecture" on page 17 presents the fundamental elements of the architecture in ScreenOS and concludes with a four-part example illustrating an enterprise-based configuration incorporating most of those elements. In this and all subsequent chapters, each concept is accompanied by illustrative examples.
- "Zones" on page 43 explains security zones, tunnel zones, and function zones.
- "Interfaces" on page 51 describes the various physical, logical, and virtual interfaces on security devices.
- "Interface Modes" on page 99 explains the concepts behind transparent, Network Address Translation (NAT), and route interface operational modes.
- "Building Blocks for Policies" on page 129 discusses the elements used for creating policies and virtual private networks (VPNs): addresses (including VIP addresses), services, and DIP pools. It also presents several example configurations that support the H.323 protocol.
- "Policies" on page 197 explores the components and functions of policies and offers guidance on their creation and application.
- "Traffic Shaping" on page 233 explains how you can prioritize services and manage bandwidth at the interface and policy levels.
- "System Parameters" on page 263 presents the concepts behind Domain Name System (DNS) addressing, using Dynamic Host Configuration Protocol (DHCP) to assign or relay TCP/IP settings, downloading and uploading system configurations and software, and setting the system clock.

Part 3: Administration

- "Administration" on page 309 explains the different means available for managing a security device both locally and remotely. This chapter also explains the privileges pertaining to each of the four levels of network administrators that can be defined.
- "Monitoring Security Devices" on page 371 explains various monitoring methods and provides guidance in interpreting monitoring output.

Part 4: Attack Detection and Defense Mechanisms

- "Protecting a Network" on page 433 outlines the basic stages of an attack and the firewall options available to combat the attacker at each stage.
- "Reconnaissance Deterrence" on page 439 describes the options available for blocking IP address sweeps, port scans, and attempts to discover the type of operating system (OS) of a targeted system.
- "Denial of Service Attack Defenses" on page 463 explains firewall, network, and OS-specific DoS attacks and how ScreenOS mitigates such attacks.
- "Content Monitoring and Filtering" on page 495 describes how to protect users from malicious uniform resource locators (URLs) and how to configure the security device to work with third party products to provide antivirus scanning, antispam, and Web filtering.
- "Deep Inspection" on page 206 describes how to configure the Juniper Networks security device to obtain Deep Inspection (DI) attack object updates, how to create user-defined attack objects and attack object groups, and how to apply DI at the policy level.
- "Intrusion Detection and Prevention" on page 615 describes Juniper Networks Intrusion Detection and Prevention (IDP) technology, which can both detect and stop attacks when deployed inline to your network. The chapter describes how to apply IDP at the policy level to drop malicious packets or connections before the attacks can enter your network.
- "Suspicious Packet Attributes" on page 697 presents several SCREEN options that protect network resources from potential attacks indicated by unusual IP and ICMP packet attributes.
- "Contexts for User-Defined Signatures" on page 2263, provides descriptions of contexts that you can specify when defining a stateful signature attack object.

Part 5: Virtual Private Networks

- "Internet Protocol Security" on page 707 provides background information about IPsec, presents a flow sequence for Phase 1 in IKE negotiations in aggressive and main modes, and concludes with information about IKE and IPsec packet encapsulation.
- "Public Key Cryptography" on page 741 provides an introduction to public key cryptography, certificate use, and certificate revocation list (CRL) use within the context of Public Key Infrastructure (PKI).
- "Virtual Private Network Guidelines" on page 769 offers some useful information to help in the selection of the available VPN options. It also presents a packet flow chart to demystify VPN packet processing.
- "Site-to-Site Virtual Private Networks" on page 801 provides extensive examples of VPN configurations connecting two private networks.
- "Dialup Virtual Private Networks" on page 887 provides extensive examples of client-to-LAN communication using AutoKey IKE. It also details group IKE ID and shared IKE ID configurations.
- "Layer 2 Tunneling Protocol" on page 933 explains Layer 2 Tunneling Protocol (L2TP) and provides configuration examples for L2TP and L2TP-over-IPsec.

- "Advanced Virtual Private Network Features" on page 961 contains information and examples for the more advanced VPN configurations, such as NAT-Traversal, VPN monitoring, binding multiple tunnels to a single tunnel interface, and hub-and-spoke and back-to-back tunnel designs.
- "AutoConnect-Virtual Private Networks" on page 1059 describes how ScreenOS uses Next Hop Resolution Protocol (NHRP) messages to enable security devices to set up AutoConnect VPNs as needed. The chapter provides an example of a typical scenario in which AC-VPN might be used.

Part 6: Voice-over-Internet Protocol

- "H.323 Application Layer Gateway" on page 1091 describes the H.323 protocol and provides examples of typical scenarios.
- "Session Initiation Protocol Application Layer Gateway" on page 1105 describes the Session Initiation Protocol (SIP) and shows how the SIP ALG processes calls in route and Network Address Translation (NAT) modes. Examples of typical scenarios follow a summary of the SIP architecture.
- "Media Gateway Control Protocol Application Layer Gateway" on page 1157 presents an overview of the Media Gateway Control Protocol (MGCP) ALG and lists the firewall security features of the implementation. Examples of typical scenarios follow a summary of the MGCP architecture.
- "Skinny Client Control Protocol Application Layer Gateway" on page 1171 presents an overview of the Skinny Client Control Protocol (SCCP) ALG and lists the firewall security features of the implementation. Examples of typical scenarios follow a summary of the SCCP architecture.
- "Apple iChat Application Layer Gateway" on page 1203 presents an overview of the AppleiChat ALG and lists the firewall security features of the implementation. Examples of typical scenarios follow a summary of the AppleiChat architecture.

Part 7: Routing

- "Static Routing" on page 1221 describes the ScreenOS routing table, the basic routing process on the security device, and how to configure static routes on security devices.
- "Routing" on page 1235 explains how to configure virtual routers on security devices and how to redistribute routing table entries between protocols or between virtual routers.
- "Open Shortest Path First" on page 1269 describes how to configure the OSPF
- "Routing Information Protocol" on page 1307 describes how to configure the RIP dynamic routing protocol on security devices.
- "Border Gateway Protocol" on page 1337 describes how to configure the BGP
- "Policy-Based Routing" on page 1373 describes policy based routing (PBR). PBR provides a flexible routing mechanism for data forwarding over networks that rely on Application Layer support such as for antivirus (AV), deep inspection (DI), or Web filtering.
- "Multicast Routing" on page 1391 introduces basic multicast routing concepts.

- "Internet Group Management Protocol" on page 1399 describes how to configure the Internet Group Management Protocol (IGMP) on security devices.
- "Protocol Independent Multicast" on page 1425 explains how to configure Protocol Independent Multicast - Sparse Mode (PIM-SM) and Protocol Independent Multicast - Source Specific Multicast (PIM-SSM) on Juniper Networks security devices.
- "ICMP Router Discovery Protocol" on page 1461 explains how to set up an Internet Control Messages Protocol (ICMP) message exchange between a host and a router.

Part 8: Address Translation

- "Address Translation" on page 1469 gives an overview of the various translation options, which are covered in detail in subsequent chapters.
- "Source Network Address Translation" on page 1481 describes NAT-src, the translation of the source IP address in a packet header, with and without Port Address Translation (PAT).
- "Destination Network Address Translation" on page 1499 describes NAT-dst, the translation of the destination IP address in a packet header, with and without destination port address mapping. This section also includes information about the packet flow when doing NAT-src, routing considerations, and address shifting.
- "Mapped and Virtual Addresses" on page 1535 describes the mapping of one destination IP address to another based on IP address alone (mapped IP) or based on destination IP address and destination port number (virtual IP).

Part 9: User Authentication

- "Authentication" on page 1565 details the various authentication methods and uses that ScreenOS supports.
- "Authentication Servers" on page 1577 presents the options of using one of four possible types of external authentication server—RADIUS, SecurID, TACACS+, or LDAP—or the internal database and shows how to configure the security device to work with each type.
- "Infranet Authentication" on page 1607 details how the security device is deployed in a unified access control (UAC) solution. Juniper Networks *unified access control* solution (UAC) secures and assures the delivery of applications and services across an enterprise infranet.
- "Authentication Users" on page 1615 explains how to define profiles for authentication users and how to add them to user groups stored either locally or on an external RADIUS authentication server.
- "IKE, XAuth, and L2TP Users" on page 1637 explains how to define IKE, XAuth, and L2TP users. Although the XAuth section focuses primarily on using the security device as an XAuth server, it also includes a subsection on configuring select security devices to act as an XAuth client.
- "Extensible Authentication for Wireless and Ethernet Interfaces" on page 1661 explains the options available for and examples of how to use the Extensible Authentication Protocol to provide authentication for Ethernet and wireless interfaces.

Part 10: Virtual Systems

- "Virtual Systems" on page 1679 discusses virtual systems and profiles, objects, and administrative tasks.
- "Traffic Sorting" on page 1713 explains how ScreenOS sorts traffic.
- "VLAN-Based Traffic Classification" on page 1723 describes VLAN-based traffic classification for virtual systems, and VLAN retagging.
- "IP-Based Traffic Classification" on page 1757 explains IP-based traffic classification for virtual systems.

Part 11: High Availability

- "NetScreen Redundancy Protocol" on page 1765 explains how to cable, configure, and manage Juniper Networks security devices in a redundant group to provide high availability (HA) using NetScreen Redundancy Protocol (NSRP).
- "Interface Redundancy and Failover" on page 1817 describes the various ways in which Juniper Networks security devices provide interface redundancy.

Part 12: WAN, DSL, Dial, and Wireless

- "Wide Area Networks" on page 1869 describes how to configure a wide area network (WAN).
- "Digital Subscriber Line" on page 1949 describes the asymmetric digital subscriber line (ADSL) and G.symmetrical digital subscriber line (G.SHDSL) interfaces.
- "ISP Failover and Dial Recovery" on page 1995 describes how to set priority and define conditions for ISP failover and how to configure a dialup recovery solution.
- "Wireless Local Area Network" on page 2001 describes the wireless interfaces on Juniper Networks wireless devices and provides example configurations.
- "Wireless Information" on page 2267 lists available channels, frequencies, and regulatory domains and lists the channels that are available on wireless devices for each country.

Part 13: General Packet Radio Service

- "GPRS" on page 2049 describes the GPRS Tunneling Protocol (GTP) features in ScreenOS and demonstrates how to configure GTP functionality on a Juniper Networks security device.

Part 14: Dual-Stack Architecture with IPv6

- "Internet Protocol Version 6 Introduction" on page 2089 explains IPv6 headers, concepts, and tunneling guidelines.
- "IPv6 Configuration" on page 2097 explains how to configure an interface for operation as an IPv6 router or host.
- "Connection and Network Services" on page 2123 explains how to configure Dynamic Host Configuration protocol version 6 (DHCPv6), Domain Name Services (DNS), Point-to-Point Protocol over Ethernet (PPPoE), and fragmentation.

- “Static and Dynamic Routing” on page 2141 explains how to set up static and dynamic routing. This chapter explains ScreenOS support for Routing Information Protocol-Next Generation (RIPng).
- “Address Translation” on page 2173 explains how to use Network Address Translation (NAT) with dynamic IP (DIP) and mapped-IP (MIP) addresses to traverse IPv4/IPv6 boundaries.
- “IPv6 in an IPv4 Environment” on page 2189 explains manual and dynamic tunneling.
- “IPsec Tunneling” on page 2203 explains how to configure IPsec tunneling to connect dissimilar hosts.
- “IPv6 XAuth User Authentication” on page 2223 explains how to configure Remote Authentication Dial In User Service (RADIUS) and IPsec Access Session (IAS) management.
- “Switching” on page 2275 lists options for using the security device as a switch to pass IPv6 traffic.

Document Conventions

This document uses the conventions described in the following sections:

Web User Interface Conventions

The Web user interface (WebUI) contains a navigational path and configuration settings. To enter configuration settings, begin by clicking a menu item in the navigation tree on the left side of the screen. As you proceed, your navigation path appears at the top of the screen, with each page separated by angle brackets.

The following example shows the WebUI path and parameters for defining an address:

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

```
Address Name: addr_1
IP Address/Domain Name:
   IP/Netmask: (select), 10.2.2.5/32
Zone: Untrust
```

To open Online Help for configuration settings, click the question mark (?) in the upper right of the screen.

The navigation tree also provides a Help > Config Guide configuration page to help you configure security policies and Internet Protocol Security (IPsec). Select an option

from the list, and follow the instructions on the page. Click the **?** character in the upper right for Online Help on the Config Guide.

Command Line Interface Conventions

The following conventions are used to present the syntax of command line interface (CLI) commands in text and examples.

In text, commands are in **boldface** type and variables are in *italic* type.

In examples:

- Variables are in *italic* type.
- Anything inside square brackets [ ] is optional.
- Anything inside braces **{ }** is required.
- If there is more than one choice, each choice is separated by a pipe (|). For example, the following command means "set the management options for the ethernet1, the ethernet2, *or* the ethernet3 interface" :

NOTE: When entering a keyword, you only have to type enough letters to identify the word uniquely. Typing **set adm u whee j12fmt54** will enter the command **set admin user wheezer j12fmt54**. However, all the commands documented in this guide are presented in their entirety.

Naming Conventions and Character Types

ScreenOS employs the following conventions regarding the names of objects—such as addresses, admin users, auth servers, IKE gateways, virtual systems, VPN tunnels, and zones—defined in ScreenOS configurations:

- If a name string includes one or more spaces, the entire string must be enclosed within double quotes; for example:

  set address trust "local LAN" 10.1.1.0/24

- Any leading spaces or trailing text within a set of double quotes are trimmed; for example, **" local LAN "** becomes **"local LAN"** .
- Multiple consecutive spaces are treated as a single space.
- Name strings are case-sensitive, although many CLI keywords are case-insensitive. For example, **"local LAN"** is different from **"local lan"** .

ScreenOS supports the following character types:

- Single-byte character sets (SBCS) and multiple-byte character sets (MBCS). Examples of SBCS are ASCII, European, and Hebrew. Examples of MBCS—also referred to as double-byte character sets (DBCS)—are Chinese, Korean, and Japanese.

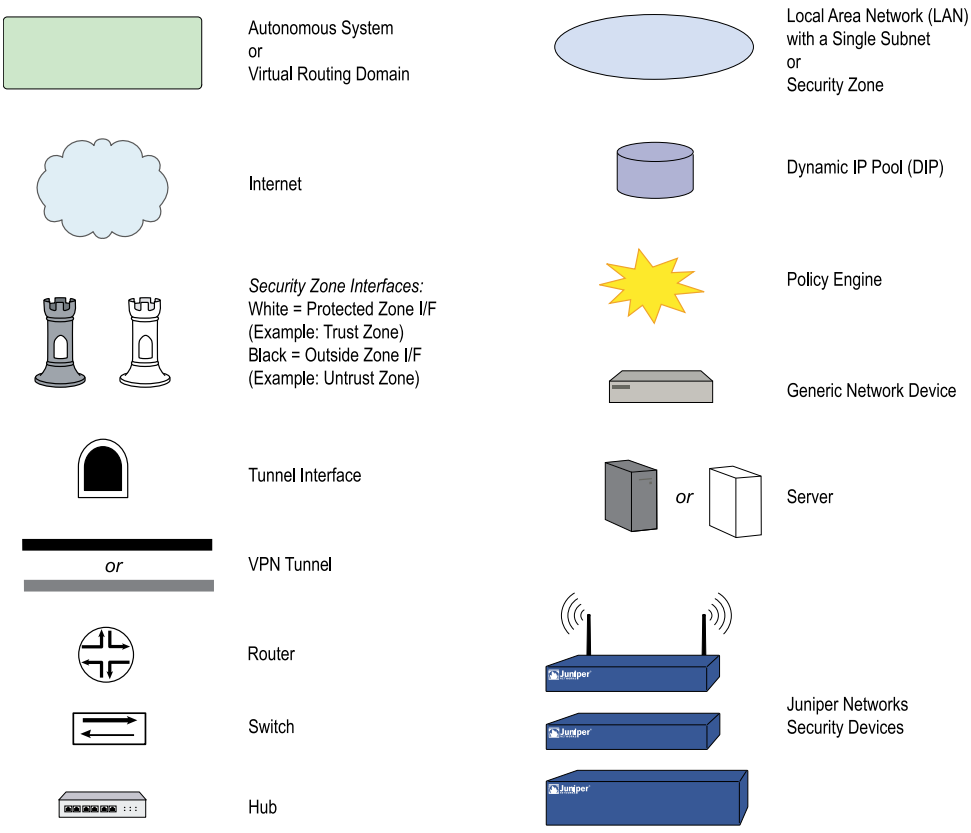
- ASCII characters from 32 (0x20 in hexadecimals) to 255 (0xff), except double quotes (“), which have special significance as an indicator of the beginning or end of a name string that includes spaces.

NOTE: A console connection only supports SBCS. The WebUI supports both SBCS and MBCS, depending on the character sets that your browser supports.

Illustration Conventions

Figure 2 on page 12 shows the basic set of images used in illustrations throughout this guide.

Figure 2: Images in Illustrations



Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need postsales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/customers/support/downloads/710059.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings—http://www.juniper.net/customers/support/
- Search for known bugs—Find product documentation—http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base—http://kb.juniper.net/
- Download the latest versions of software and review your release notes—http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications—http://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum—http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Manager—http://www.juniper.net/customers/cm/
- To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool—https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Manager tool in the CSC at http://www.juniper.net/customers/cm/.
- Call 1-888-314-JTAC (1-888-314-5822—toll free in USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, visit us at http://www.juniper.net/customers/support/requesting-support/.

Document Feedback

If you find any errors or omissions in this document, contact Juniper Networks at techpubs-comments@juniper.net.

Part 2
Fundamentals

This guide describes the ScreenOS architecture and its elements, including examples for configuring various elements. This guide contains the following chapters:

- "ScreenOS Architecture" on page 17 presents the fundamental elements of the architecture in ScreenOS and concludes with a four-part example illustrating an enterprise-based configuration incorporating most of those elements. In this and all subsequent chapters, each concept is accompanied by illustrative examples.
- "Zones" on page 43 explains security, tunnel, and function zones.
- "Interfaces" on page 51 describes the various physical, logical, and virtual interfaces on security devices.
- "Interface Modes" on page 99 explains the concepts behind transparent, Network Address Translation (NAT), and route interface operational modes.
- "Building Blocks for Policies" on page 129 discusses the elements used for creating policies and virtual private networks (VPNs): addresses (including VIP addresses), services, and DIP pools. It also presents several example configurations that support the H.323 protocol.
- "Policies" on page 197 explores the components and functions of policies and offers guidance on their creation and application.
- "Traffic Shaping" on page 233 explains how you can prioritize services and manage bandwidth at the interface and policy levels.
- "System Parameters" on page 263 presents the concepts behind Domain Name System (DNS) addressing, using Dynamic Host Configuration Protocol (DHCP) to assign or relay TCP/IP settings, downloading and uploading system configurations and software, and setting the system clock.

Chapter 2

ScreenOS Architecture

Juniper Networks ScreenOS architecture offers you flexibility in designing the layout of your network security. On Juniper Networks security devices with more than two interfaces, you can create numerous security zones and configure policies to regulate traffic between and within zones. You can bind one or more interfaces to each zone and enable different management and firewall options for each zone. ScreenOS allows you to create the number of zones required by your network environment, assign the number of interfaces required by each zone, and design each interface according to your needs.

This chapter presents an overview of ScreenOS. It contains the following sections:

The chapter concludes with a four-part example that illustrates a basic configuration for a security device using ScreenOS:

Security Zones

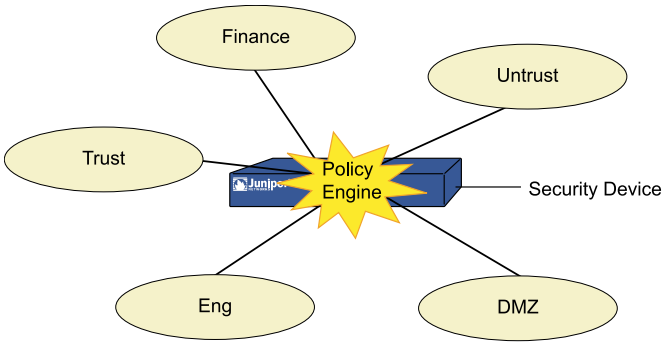
A security zone is a collection of one or more network segments requiring the regulation of inbound and outbound traffic via policies (see “Policies” on page 20). Security zones are logical entities to which one or more interfaces are bound. With many types of Juniper Networks security devices, you can define multiple security zones, the exact number of which you determine based on your network needs. In addition to user-defined zones, you can also use the predefined zones: Trust, Untrust, and DMZ (for Layer 3 operation), or V1-Trust, V1-Untrust, and V1-DMZ (for Layer 2

operation). If you want, you can continue using just the predefined zones. You can also ignore the predefined zones and use user-defined zones exclusively. Optionally, you can use both kinds of zones—predefined and user-defined—side by side. This flexibility for zone configuration allows you to create a network design that best suits your specific needs. See Figure 3 on page 18.

NOTE: The one security zone that requires no network segment is the global zone. (For more information, see “Global Zone” on page 45.) Additionally, any zone without an interface bound to it nor any address book entries can also be said not to contain any network segments.
If you upgrade from an earlier version of ScreenOS, all your configurations for these zones remain intact.
You cannot delete a predefined security zone. You can, however, delete a user-defined zone. When you delete a security zone, you also automatically delete all addresses configured for that zone.

Figure 3 on page 18 shows a network configured with five security zones—three default zones (Trust, Untrust, DMZ) and two user-defined zones (Finance, Eng). Traffic passes from one security zone to another only if a policy permits it.

Figure 3: Predefined Security Zones



Security Zone Interfaces

An interface for a security zone can be thought of as a doorway through which TCP/IP traffic can pass between that zone and any other zone.

Through the policies you define, you can permit traffic between zones to flow in one direction or in both. With the routes that you define, you specify the interfaces that traffic from one zone to another must use. Because you can bind multiple interfaces to a zone, the routes you chart are important for directing traffic to the interfaces of your choice.

NOTE: For traffic to flow between interfaces bound to the same zone, no policy is required because both interfaces have security equivalency. ScreenOS requires policies for traffic between zones, not within a zone.

To permit traffic to flow from zone to zone, you bind an interface to the zone and—for an interface in route or NAT mode (see “Interface Modes” on page 99)—assign an IP address to the interface. Two common interface types are physical interfaces and—for those devices with virtual-system support—subinterfaces (that is, a Layer 2 substantiation of a physical interface). For more information, see “Interfaces” on page 51.

Physical Interfaces

A physical interface relates to components that are physically present on the security device. The interface-naming convention differs from device to device.

NOTE: To see the naming conventions for a specific security device, see the hardware guide for that device.

Subinterfaces

On devices that support virtual LANs (VLANs), you can logically divide a physical interface into several virtual subinterfaces, each of which borrows the bandwidth it needs from the physical interface from which it stems. A subinterface is an abstraction that functions identically to a physical interface and is distinguished by 802.1Q VLAN tagging. The security device directs traffic to and from a zone with a subinterface via its IP address and VLAN tag. For convenience, administrators usually use the same number for a VLAN tag as the subinterface number. For example, the interface ethernet1/2 using VLAN tag 3 is named ethernet1/2.3. This refers to the interface module in the first bay, the second port on that module, and subinterface number **3** (ethernet1/2.3).

NOTE: 802.1Q is an IEEE standard that defines the mechanisms for the implementation of virtual bridged LANs and the ethernet frame formats used to indicate VLAN membership via VLAN tagging.

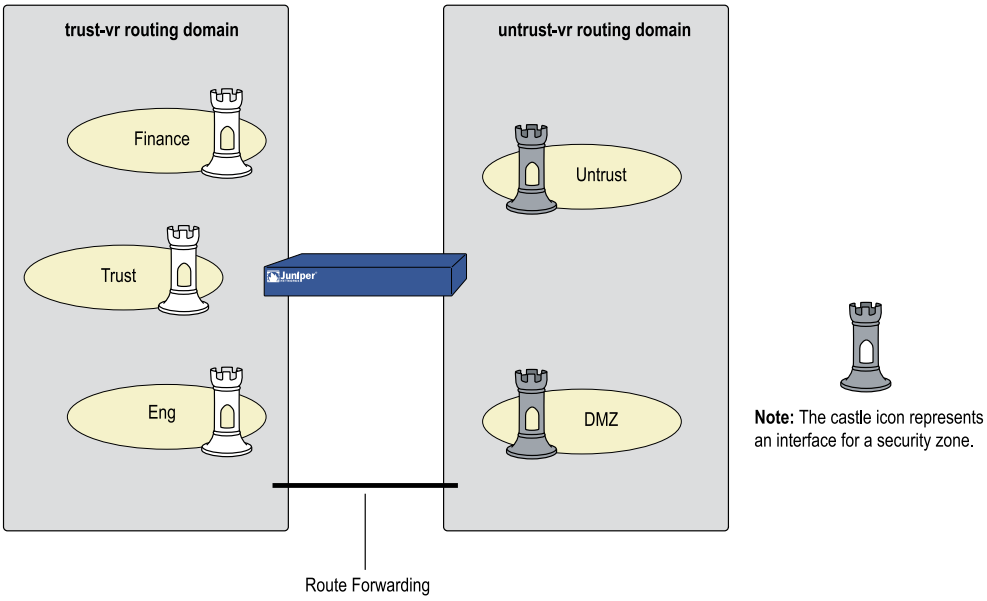
Note that although a subinterface shares part of its identity with a physical interface, the zone to which you bind it is not dependent on the zone to which you bind the physical interface. You can bind the subinterface ethernet1/2.3 to a different zone than that to which you bind the physical interface ethernet1/2, or to which you bind ethernet1/2.2. Similarly, there are no restrictions in terms of IP-address assignments. The term subinterface does not imply that its address be in a subnet of the address space of the physical interface.

Virtual Routers

A virtual router (VR) functions as a router. It has its own interfaces and its own unicast and multicast routing tables. In ScreenOS, a security device supports two predefined virtual routers. This allows the security device to maintain two separate unicast and multicast routing tables and to conceal the routing information in one virtual router from the other. For example, the untrust-vr is typically used for communication with untrusted parties and does not contain any routing information for the protected

zones. Routing information for the protected zones is maintained by the trust-vr. Thus, no internal network information can be gathered by the covert extraction of routes from the untrust-vr, see Figure 4 on page 20.

Figure 4: Virtual Router Security Zones



When there are two virtual routers on a security device, traffic is not automatically forwarded between zones that reside in different VRs, even if there are policies that permit the traffic. If you want traffic to pass between virtual routers, you need to either export routes between the VRs or configure a static route in one VR that defines the other VR as the next hop. For more information about using two virtual routers, see "Routing" on page 1219.

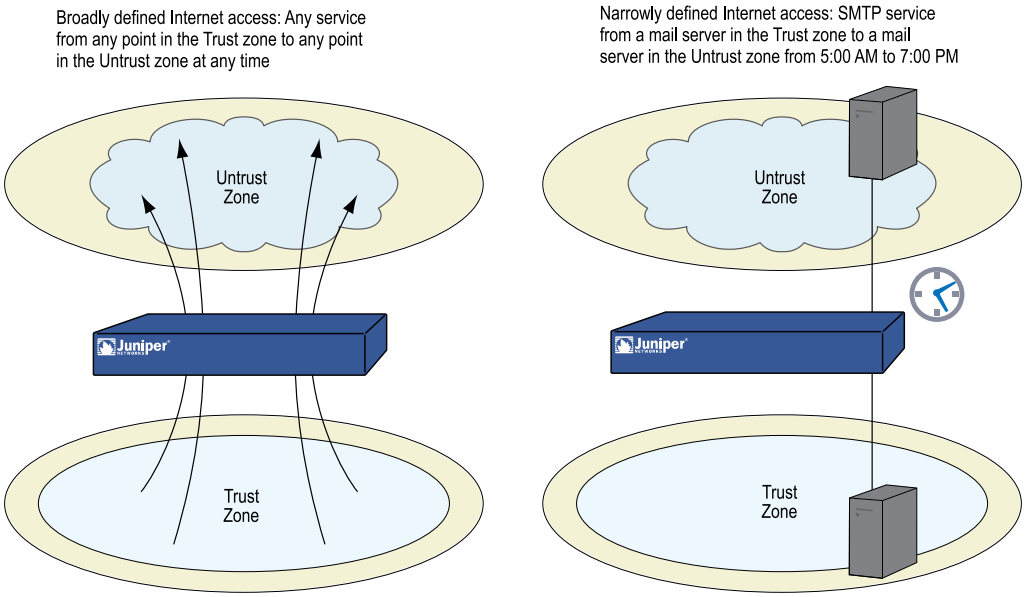
Policies

Juniper Networks security devices secure a network by inspecting, and then allowing or denying, all connection attempts that require passage from one security zone to another.

By default, a security device denies all traffic in all directions. Through the creation of policies, you can control the traffic flow from zone to zone by defining the kinds of traffic permitted to pass from specified sources to specified destinations at scheduled times. At the broadest level, you can allow all kinds of traffic from any source in one zone to any destination in all other zones without any scheduling restrictions. At the narrowest level, you can create a policy that allows only one kind of traffic between a specified host in one zone and another specified host in another zone during a scheduled period, see Figure 5 on page 21.

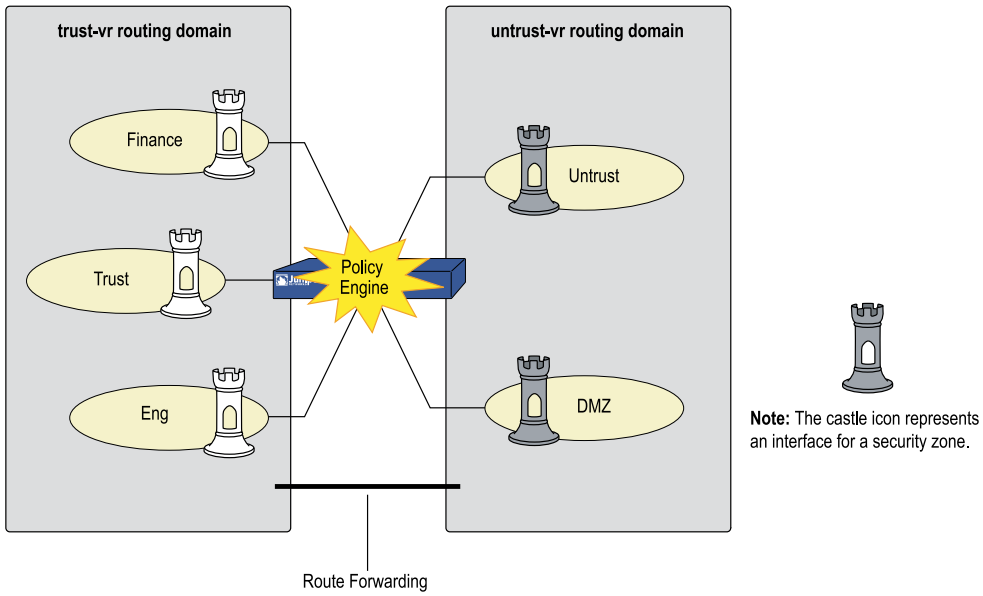
NOTE: Some security devices ship with a default policy that allows all outbound traffic from the Trust to the Untrust zone but denies all inbound traffic from the Untrust zone to the Trust zone.

Figure 5: Default Policy



Every time a packet attempts to pass from one zone to another or between two interfaces bound to the same zone, the security device checks its policy set lists for a policy that permits such traffic (see “Policy Set Lists” on page 200). To allow traffic to pass from one security zone to another—for example, from zone A to zone B—you must configure a policy that permits zone A to send traffic to zone B. To allow traffic to flow the other way, you must configure another policy permitting traffic from zone B to zone A. For any traffic to pass from one zone to another, there must be a policy that permits it. Also, if intrazone blocking is enabled, there must be a policy to permit traffic to pass from one interface to another within that zone. See Figure 6 on page 22.

Figure 6: Policy Architecture



NOTE: For more information, see "Policies" on page 197.

If you configure multicast routing on a security device, you might have to configure multicast policies. By default, a security device does not permit multicast control traffic between zones. Multicast control traffic refers to the messages transmitted by multicast protocols, such as Protocol Independent Multicast (PIM). Multicast policies control the flow of multicast control traffic only. To allow data traffic (both unicast and multicast) to pass between zones, you must configure firewall policies. (For more information, see "Multicast Policies" on page 1396 .)

Virtual Private Networks

ScreenOS supports several virtual private network (VPN) configuration options. The two main types are as follows:

- **Route-based VPN**—A route lookup determines which traffic the security device encapsulates. Policies either permit or deny traffic to the destination specified in the route. If the policy permits the traffic and the route references a tunnel interface bound to a VPN tunnel, then the security device also encapsulates it. This configuration separates the application of policies from the application of VPN tunnels. Once configured, such tunnels exist as available resources for securing traffic en route between one security zone and another.
- **Policy-based VPN**—A policy lookup determines which traffic the security device encapsulates when the policy references a particular VPN tunnel and specifies "tunnel" as the action.

A route-based VPN is good choice for site-to-site VPN configurations because you can be apply multiple policies to traffic passing through a single VPN tunnel. A

policy-based VPN is a good choice for dialup VPN configurations because the dialup client might not have an internal IP address to which you can set a route. See Figure 7 on page 24.

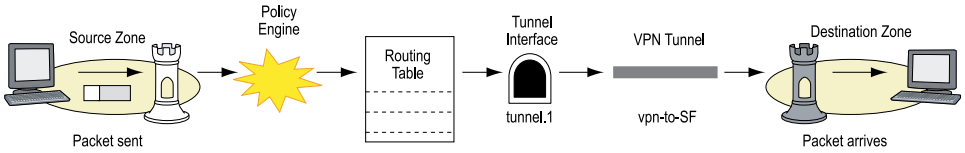
The following steps provide a sense of the main elements involved in a route-based VPN configuration:

1. While configuring the VPN tunnel (for example, vpn-to-SF, where SF is the destination or end entity), specify a physical interface or subinterface on the local device as the outgoing interface. (The IP address for this interface is what the remote peer must use when configuring its remote gateway.)
2. Create a tunnel interface (for example, tunnel.1), and bind it to a security zone.

NOTE: You do not have to bind the tunnel interface to the same zone for which VPN traffic is destined. Traffic to any zone can access a tunnel interface if a route points to that interface.

3. Bind the tunnel interface tunnel.1 to the VPN tunnel vpn-to-SF.
4. To direct traffic through this tunnel, set up a route stating that traffic to SF must use *tunnel.1*.

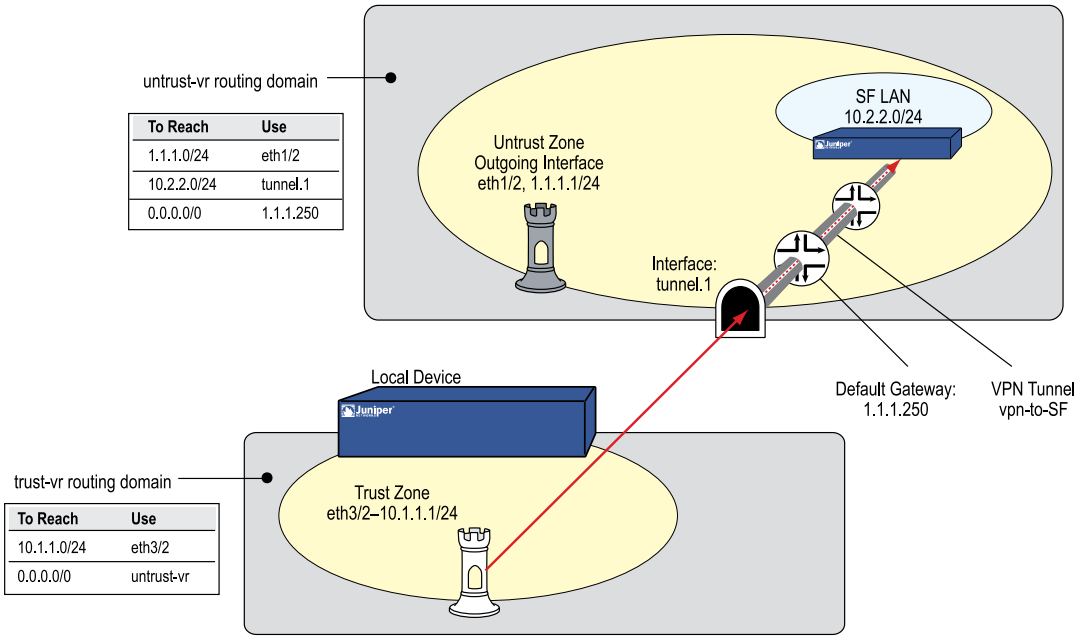
Figure 7: VPN Traffic


Source Zone
Policy Engine
Routing Table
Tunnel Interface
VPN Tunnel
Destination Zone
tunnel.1
vpn-to-SF
Packet sent
Packet arrives

At this point, the tunnel is ready for traffic bound for SF. You can now create address-book entries, such as “Trust LAN” (10.1.1.0/24) and “ SF LAN” (10.2.2.0/24) and set up policies to permit or block different types of traffic from a specified source, such as Trust LAN, to a specified destination, such as SF LAN. See Figure 8 on page 26.

Figure 8: VPN Traffic from Untrust Security Zone

The local security device routes traffic from the Trust zone to SF LAN in the Untrust zone through the tunnel.1 interface. Because tunnel.1 is bound to the VPN tunnel vpn-to-SF, the device encrypts the traffic and sends it through that tunnel to the remote peer.

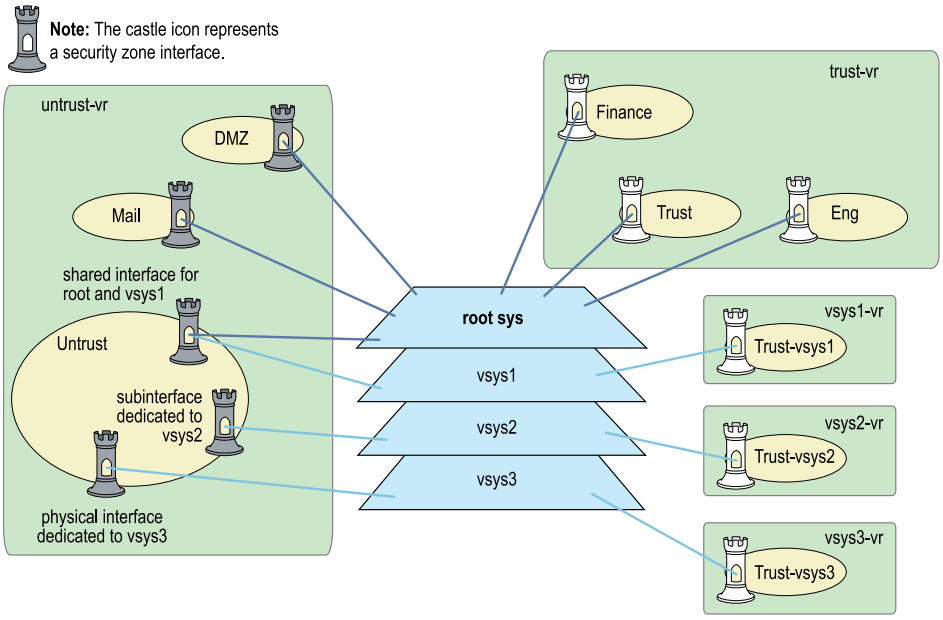


NOTE: For detailed information about VPNs, see “Virtual Private Networks” on page 705 .

Virtual Systems

Some Juniper Networks security devices support virtual systems (vsys). A virtual system is a subdivision of the main system that appears to the user to be a standalone entity. Virtual systems reside separately from each other and from the root system within the same security device. The application of ScreenOS to virtual systems involves the coordination of three main components: zones, interfaces, and virtual routers. Figure 9 on page 27 presents a conceptual overview of how ScreenOS integrates these components at both the root and vsys levels.

Figure 9: Vsys Architecture

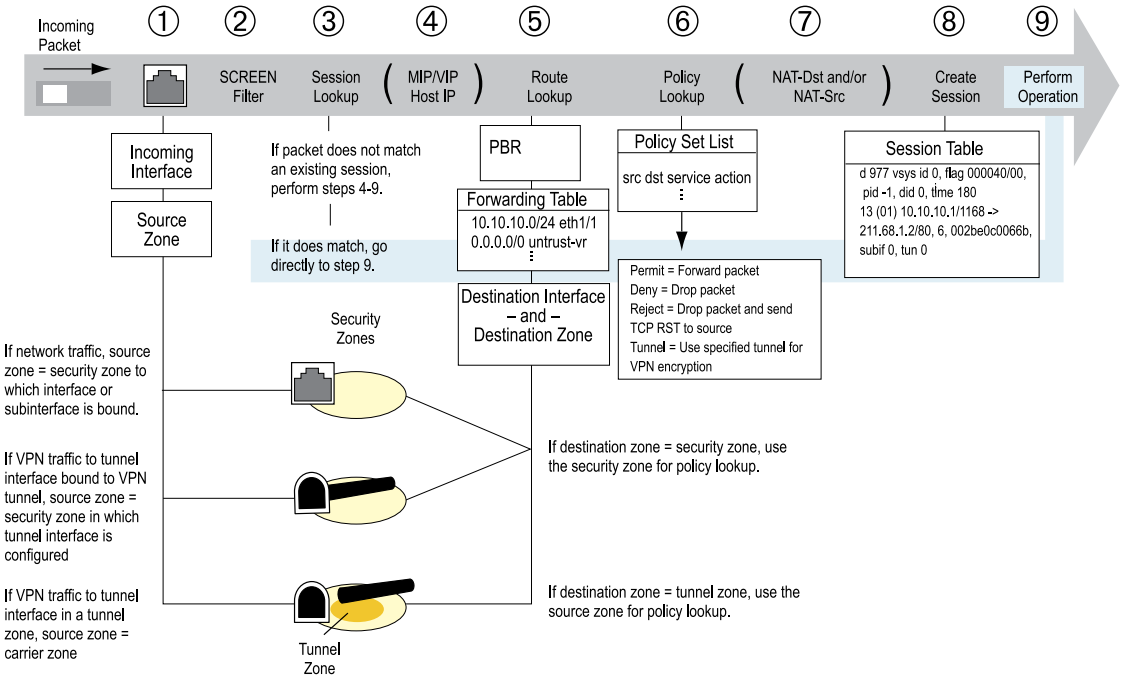


NOTE: For further information about virtual systems and the application of zones, interfaces, and virtual routers within the context of virtual systems, see “Virtual Systems” on page 1677 .

Packet-Flow Sequence

In ScreenOS, the flow sequence of an incoming packet progresses as presented in Figure 10 on page 28.

Figure 10: Packet Flow Sequence Through Security Zones



1. The interface module identifies the incoming interface and, consequently, the source zone to which the interface is bound.

   The interface module uses the following criteria to determine the source zone:

   - If the packet is not encapsulated, the source zone is the security zone to which the incoming interface or subinterface is bound.
   - If the packet is encapsulated and the tunnel interface is bound to a VPN tunnel, the source zone is the security zone in which the tunnel interface is configured.
   - If the packet is encapsulated and the tunnel interface is in a tunnel zone, the source zone is the corresponding carrier zone (a security zone that carries a tunnel zone) for that tunnel zone.

2. If you have enabled SCREEN options for the source zone, the security device activates the SCREEN module at this point. SCREEN checking can produce one of the following three results:

   - If a SCREEN mechanism detects anomalous behavior for which it is configured to block the packet, the security device drops the packet and makes an entry in the event log.
   - If a SCREEN mechanism detects anomalous behavior for which it is configured to record the event but not block the packet, the security device records the event in the SCREEN counters list for the ingress interface and proceeds to the next step.
   - If the SCREEN mechanisms detect no anomalous behavior, the security device proceeds to the next step.

3. The session module performs a session lookup, attempting to match the packet with an existing session.
   - If the packet does not match an existing session, the security device performs First Packet Processing, a procedure involving steps 4 through 9.
   - If the packet matches an existing session, the security device performs Fast Processing, using the information available from the existing session entry to process the packet. Fast Processing bypasses steps 4 through 8 because the information generated by those steps has already been obtained during the processing of the first packet in the session.
4. If a Mapped IP (MIP) or Virtual IP (VIP) address is used, the address-mapping module resolves the MIP or VIP so that the routing table can search for the actual host address.
5. Prior to route lookup, ScreenOS checks the packet for policy based routing (PBR). If PBR is enabled on that in-interface, the following actions apply to the packet:
   - The PBR policy bound to that in-interface is applied to the packet.
   - If no PBR policy exists at the interface level, the PBR policy bound to the zone associated with the in-interface is applied to the packet.
   - If no PBR policy exists at the zone level, the PBR policy bound to the VR associated with the in-interface is applied to the packet.

NOTE: For more information about policy based routing, see “Routing” on page 1219.

   If PBR is not enabled, the route table lookup finds the interface that leads to the destination address. In so doing, the interface module identifies the destination zone to which that interface is bound.

   The interface module uses the following criteria to determine the destination zone:

   If the destination zone is a security zone, that zone is used for the policy lookup.
   - If the destination zone is a tunnel zone, the corresponding carrier zone is used for the policy lookup.
   - If the destination zone is the same as the source zone and intrazone blocking is disabled for that zone, the security device bypasses steps 6 and 7 and creates a session (step 8). If intrazone blocking is enabled, then the security device drops the packet.
6. The policy engine searches the policy set lists for a policy between the addresses in the identified source and destination zones.

   The action configured in the policy determines how the security device handles the packet:

- If the action is **permit**, the security device will forward the packet to its destination.
- If the action is **deny**, the security device will drop the packet.
- If the action is **reject**, the security device will drop the packet and—if the protocol is TCP—send a reset (RST) to the source IP address.
- If the action is **tunnel**, the security device will forward the packet to the VPN module, which encapsulates the packet and transmits it using the specified VPN tunnel settings.

7. If destination address translation (NAT-dst) is specified in the policy, the NAT module translates the original destination address in the IP packet header to a different address.

   If source address translation is specified (either interface-based NAT or policy-based NAT-src), the NAT module translates the source address in the IP packet header before forwarding it either to its destination or to the VPN module.

   (If both NAT-dst and NAT-src are specified in the same policy, the security device first performs NAT-dst and then NAT-src.)

8. The session module creates a new entry in the session table containing the results of steps 1 through 7.

   The security device then uses the information maintained in the session entry when processing subsequent packets of the same session.

9. The security device performs the operation specified in the session.

   Some typical operations are source address translation, VPN tunnel selection and encryption, decryption, and packet forwarding.

Jumbo Frames

On some devices you can increase throughput by increasing the maximum packet size, or message transmission unit (MTU), the device can process. Refer to the installation and configuration guide for your device to find out if it supports jumbo frames.

Frame size ranges from 1514 through 9830 bytes. To put the device in jumbo frame mode, set the maximum frame size to a value from1515 through 9830 inclusive, for example: **set envar max-frame-size = 9830**. Use the **unset envar max-frame-size** command to return the device to normal maximum frame size, which is 1514 bytes (alternatively, you can use the command: **set envar max-frame-size = 1514**). The maximum frame size does not include the 4-byte frame check sequence at the end of the frame. You must restart the system for changes to environmental variables to take effect.

In jumbo frame mode, the following apply:

- Deep inspection (DI) is not supported.
- Packets sent through aggregate interfaces might be out of order.

- NSRP forwarding is not supported.
- Maximum firewall or VPN throughput requires at least four sessions (for firewall) or tunnels (for VPN).

ScreenOS Architecture Example

The following sections comprise a four-part example that illustrates some of the concepts covered in the previous sections:

- "Example: (Part 1) Enterprise with Six Zones" on page 31 on page 14
- Example: (Part 2) Interfaces for Six Zones on page 33
- Example: (Part 3) Two Routing Domains on page 35
- Example: (Part 4) Policies on page 37

Example: (Part 1) Enterprise with Six Zones

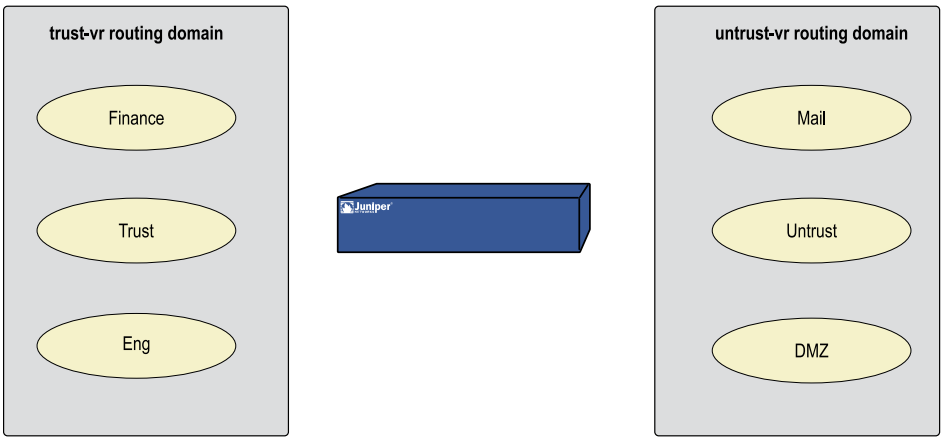
This is the first of a four-part example, the purpose of which is to illustrate some of the concepts covered in the previous sections. For the second part, in which the interfaces for each zone are set, see "Example: (Part 2) Interfaces for Six Zones" on page 33. Here you configure the following six zones for an enterprise:

- Finance
- Trust
- Eng
- Mail
- Untrust
- DMZ

The Trust, Untrust, and DMZ zones are pre-configured. You must define the Finance, Eng, and Mail zones. By default, a user-defined zone is placed in the trust-vr routing domain. Thus, you do not have to specify a virtual router for the Finance and Eng zones. However, in addition to configuring the Mail zone, you must also specify that it be in the untrust-vr routing domain. You must also shift virtual router bindings for the Untrust and DMZ zones from the trust-vr to the untrust-vr, see Figure 11 on page 32.

NOTE: For more information about virtual routers and their routing domains, see "Routing" on page 1219.

Figure 11: Zone-to-Virtual Router Bindings



WebUI

Network > Zones > New: Enter the following, then click **OK**:

Zone Name: Finance
Virtual Router Name: trust-vr
Zone Type: Layer 3: (select)

Network > Zones > New: Enter the following, then click **OK**:

Zone Name: Eng
Virtual Router Name: trust-vr
Zone Type: Layer 3: (select)

Network > Zones > New: Enter the following, then click **OK**:

Zone Name: Mail
Virtual Router Name: untrust-vr
Zone Type: Layer 3: (select)

Network > Zones > Edit (for Untrust): Select **untrust-vr** in the Virtual Router Name drop-down list, then click **OK**.

Network > Zones > Edit (for DMZ): Select **untrust-vr** in the Virtual Router Name drop-down list, then click **OK**.

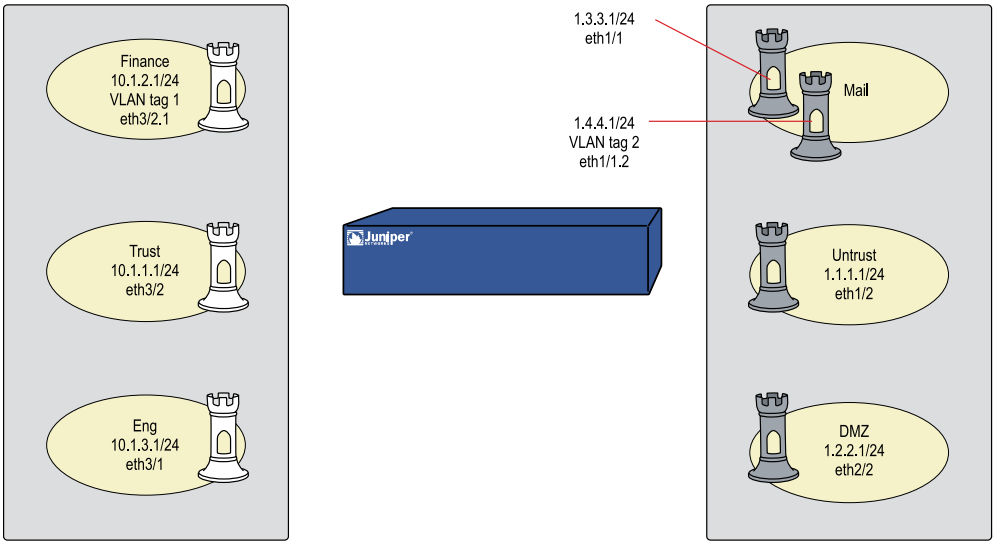
CLI

```
set zone name finance
set zone name eng
set zone name mail
set zone mail vrouter untrust-vr
set zone untrust vrouter untrust-vr
set zone dmz vrouter untrust-vr
save
```

Example: (Part 2) Interfaces for Six Zones

This is the second part of an ongoing example. For the first part, in which zones are configured, see "Example: (Part 1) Enterprise with Six Zones" on page 31. For the next part, in which virtual routers are configured, see "Example: (Part 3) Two Routing Domains" on page 35. This part of the example demonstrates how to bind interfaces to zones and configure them with an IP address and various management options, see Figure 12 on page 33.

Figure 12: Interface-to-Zone Bindings



WebUI

1. **Interface ethernet3/2**

   Network > Interfaces > Edit (for ethernet3/2): Enter the following, then click **OK**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Manageable: (select)
   Management Services: WebUI, Telnet, SNMP, SSH (select)
   Other Services: Ping (select)

2. **Interface ethernet3/2.1**

   Network > Interfaces > Sub-IF New: Enter the following, then click **OK**:

   Interface Name: ethernet3/2.1
   Zone Name: Finance
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.2.1/24

VLAN Tag: 1
Other Services: Ping (select)

3. **Interface ethernet3/1**

   Network > Interfaces > Edit (for ethernet3/1): Enter the following, then click **OK**:

   Zone Name: Eng
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.3.1/24
   Other Services: Ping (select)

4. **Interface ethernet1/1**

   Network > Interfaces > Edit (for ethernet1/1): Enter the following, then click **OK**:

   Zone Name: Mail
   Static IP: (select this option when present)
   IP Address/Netmask: 1.3.3.1/24

5. **Interface ethernet1/1.2**

   Network > Interfaces > Sub-IF New: Enter the following, then click **OK**:

   Interface Name: ethernet1/1.2
   Zone Name: Mail
   Static IP: (select this option when present)
   IP Address/Netmask: 1.4.4.1/24
   VLAN Tag: 2

6. **Interface ethernet1/2**

   Network > Interfaces > Edit (for ethernet1/2): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24
   Manageable: (select)
   Management Services: SNMP (select)

7. **Interface ethernet2/2**

   Network > Interfaces > Edit (for ethernet2/2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select)
   IP Address/Netmask: 1.2.2.1/24

CLI

1. **Interface ethernet3/2**

```
set interface ethernet3/2 zone trust
set interface ethernet3/2 ip 10.1.1.1/24
set interface ethernet3/2 manage ping
set interface ethernet3/2 manage webui
set interface ethernet3/2 manage telnet
set interface ethernet3/2 manage snmp
set interface ethernet3/2 manage ssh
```

2. **Interface ethernet3/2.1**

```
set interface ethernet3/2.1 tag 1 zone finance
set interface ethernet3/2.1 ip 10.1.2.1/24
set interface ethernet3/2.1 manage ping
```

3. **Interface ethernet3/1**

```
set interface ethernet3/1 zone eng
set interface ethernet3/1 ip 10.1.3.1/24
set interface ethernet3/1 manage ping
```

4. **Interface ethernet1/1**

```
set interface ethernet1/1 zone mail
set interface ethernet1/1 ip 1.3.3.1/24
```

5. **Interface ethernet1/1.2**

```
set interface ethernet1/1.2 tag 2 zone mail
set interface ethernet1/1.2 ip 1.4.4.1/24
```

6. **Interface ethernet1/2**

```
set interface ethernet1/2 zone untrust
set interface ethernet1/2 ip 1.1.1.1/24
set interface ethernet1/2 manage snmp
```

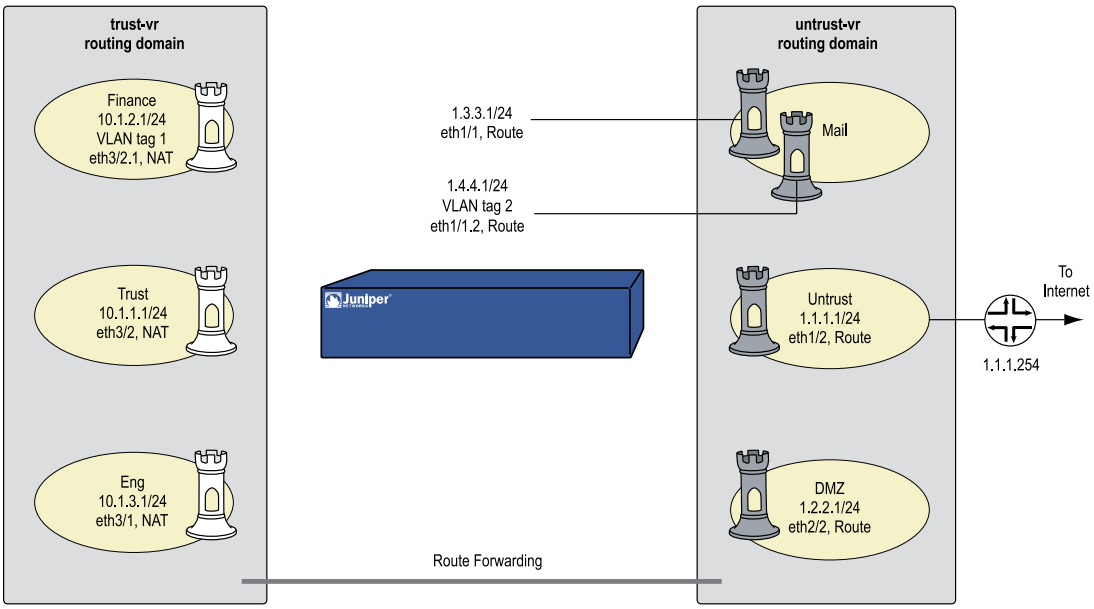
7. **Interface ethernet2/2**

```
set interface ethernet2/2 zone dmz
set interface ethernet2/2 ip 1.2.2.1/24
save
```

Example: (Part 3) Two Routing Domains

This is the third part of an ongoing example. For the previous part, in which interfaces for the various security zones are defined, see “Example: (Part 2) Interfaces for Six Zones” on page 33. For the next part, in which the policies are set, see “Example: (Part 4) Policies” on page 37. In this example, you only have to configure a route for the default gateway to the Internet. The other routes are automatically created by the security device when you create the interface IP addresses, see Figure 13 on page 36.

Figure 13: Routing Domains



WebUI

Network > Routing > Destination > (select trust-vr) New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Next Hop Virtual Router Name: (select); untrust-vr

Network > Routing > Destination > (select untrust-vr) New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet1/2
Gateway IP Address: 1.1.1.254

CLI

```
set vrouter trust-vr route 0.0.0.0/0 vrouter untrust-vr
set vrouter untrust-vr route 0.0.0.0/0 interface eth1/2 gateway 1.1.1.254
save
```

The security device automatically creates the routes shown in Table 1 on page 36 and Table 2 on page 37 (except as indicated).

Table 1: Route Table for trust-vr

To Reach:	Use Interface:	Use Gateway/Vrouter:	Created by:
0.0.0.0/0	n/a	untrust-vr	User-configured

Table 1: Route Table for trust-vr *(continued)*

To Reach:	Use Interface:	Use Gateway/Vrouter:	Created by:
10.1.3.0/24	eth3/1	0.0.0.0	Security device
10.1.1.0/24	eth3/2	0.0.0.0	Security device
10.1.2.0/24	eth3/2.1	0.0.0.0	Security device

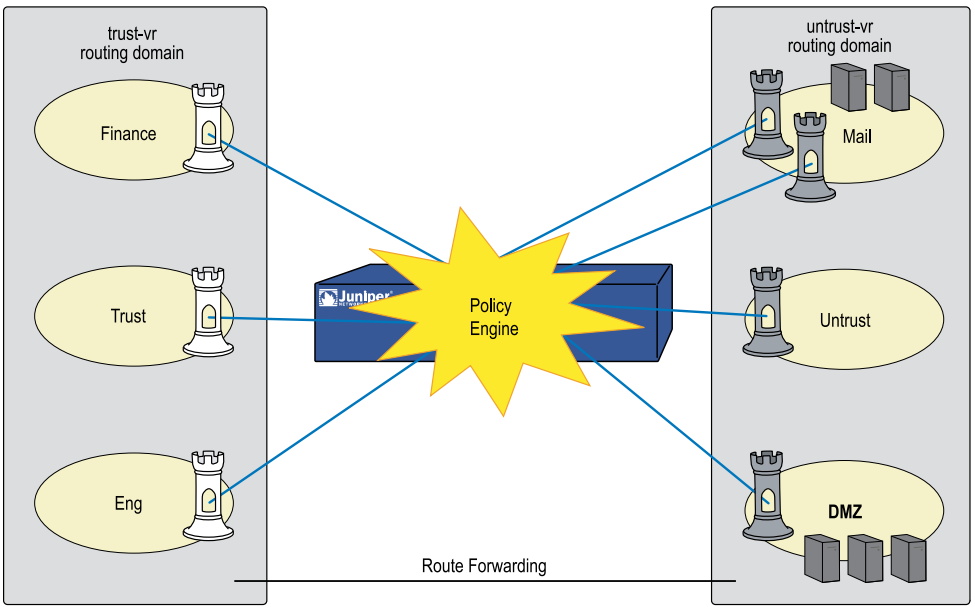
Table 2: Route Table for untrust-vr

To Reach:	Use Interface:	Use Gateway/Vrouter:	Created by:
1.2.2.0/24	eth2/2	0.0.0.0	Security device
1.1.1.0/24	eth1/2	0.0.0.0	Security device
1.4.4.0/24	eth1/1.2	0.0.0.0	Security device
1.3.3.0/24	eth1/1	0.0.0.0	Security device
0.0.0.0/0	eth1/2	1.1.1.254	User-configured

Example: (Part 4) Policies

This is the last part of an ongoing example. The previous part is "Example: (Part 3) Two Routing Domains" on page 35. This part of the example demonstrates how to configure new policies. See Figure 14 on page 37.

Figure 14: Policies



For the purpose of this example, before you begin configuring new policies, you need to create new service groups.

NOTE: When you create a zone, the security device automatically creates the address **Any** for all hosts within that zone. This example makes use of the address **Any** for the hosts.

WebUI

1. **Service Groups**

   Policy > Policy Elements > Services > Groups > New: Enter the following, then click **OK**:

   Group Name: Mail-Pop3

   Select **Mail**, then use the < < button to move that service from the Available Members column to the Group Members column.

   Select **Pop3**, then use the < < button to move that service from the Available Members column to the Group Members column.

   Policy > Policy Elements > Services > Groups > New: Enter the following, then click **OK**:

   Group Name: HTTP-FTPGet

   Select **HTTP**, then use the < < button to move that service from the Available Members column to the Group Members column.

   Select **FTP-Get**, then use the < < button to move that service from the Available Members column to the Group Members column.

2. **Policies**

   Policy > Policies > (From: Finance, To: Mail) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: Mail-Pop3
   Action: Permit

   Policy > Policies > (From: Trust, To: Mail) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: Mail-Pop3

Action: Permit

Policy > Policies > (From: Eng, To: Mail) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: Mail-Pop3
Action: Permit

Policy > Policies > (From: Untrust, To: Mail) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: Mail
Action: Permit

Policy > Policies > (From: Finance, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: HTTP-FTPGet
Action: Permit

Policy > Policies > (From: Finance, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: HTTP-FTPGet
Action: Permit

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: HTTP-FTPGet
Action: Permit

Policy > Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:

Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: HTTP-FTPGet
Action: Permit

Policy > Policies > (From: Eng, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: HTTP-FTPGet
Action: Permit

Policy > Policies > (From: Eng, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: FTP-Put
Action: Permit

Policy > Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: HTTP-FTPGet
Action: Permit

CLI

1. **Service Groups**

```
set group service mail-pop3 add mail
set group service mail-pop3 add pop3
set group service http-ftpget add http
set group service http-ftpget add ftp-get
```

2. **Policies**

```
set policy from finance to mail any any mail-pop3 permit
set policy from trust to mail any any mail-pop3 permit
set policy from eng to mail any any mail-pop3 permit
set policy from untrust to mail any any mail permit
set policy from finance to untrust any any http-ftpget permit
set policy from finance to dmz any any http-ftpget permit
set policy from trust to untrust any any http-ftpget permit
set policy from trust to dmz any any http-ftpget permit
```

```
set policy from eng to untrust any any http-ftpget permit
set policy from eng to dmz any any http-ftpget permit
set policy from eng to dmz any any ftp-put permit
set policy from untrust to dmz any any http-ftpget permit
save
```

Chapter 3
Zones

A zone can be a segment of network space to which security measures are applied (a security zone), a logical segment to which a VPN tunnel interface is bound (a tunnel zone), or either a physical or a logical entity that performs a specific function (a function zone).

This chapter examines each type of zone, with particular emphasis given to the security zone. It contains the following sections:

Viewing Preconfigured Zones

When you first boot up a security device, you can see a number of preconfigured zones. To view these zones using the WebUI, click **Network > Zones** in the menu column on the left. See Figure 15 on page 44.

To view these zones using the CLI, use the **get zone** command. See Figure 16 on page 44.

Figure 15: Network > Zones Page in the WebUI

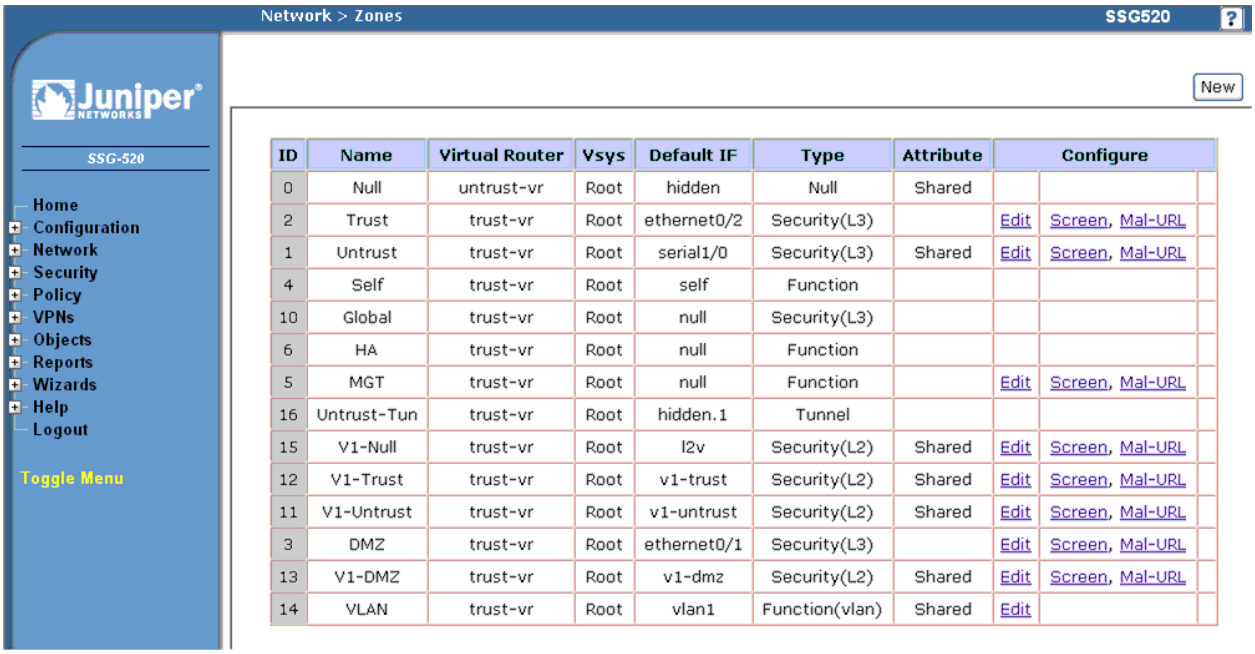
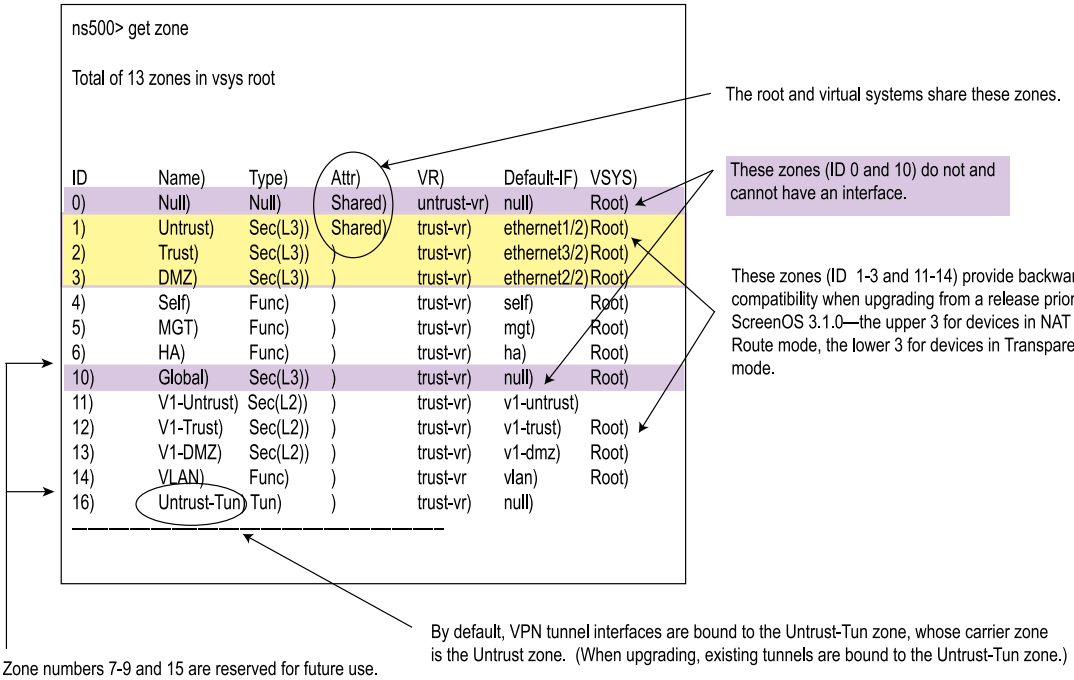


Figure 16 on page 44 shows the output of the **get zone** CLI command.

Figure 16: Get Zone Output



The preconfigured zones shown in Figure 15 on page 44 and Figure 16 on page 44 can be grouped into four different zone types:

- Security: Untrust, Trust, DMZ, Global, V1-Untrust, V1-Trust, V1-DMZ, V1-Null
- Tunnel: Untrust-Tun
- Function: Self, MGT, HA, VLAN
- Null: Null

Security Zones

On a single security device, you can configure multiple security zones, dividing the network into segments to which you can apply various security options to satisfy the needs of each segment. At a minimum, you must define two security zones, basically to protect one area of the network from the other. On some security platforms, you can define many security zones, bringing finer granularity to your network security design—and without deploying multiple security appliances to do so.

Global Zone

You can identify a security zone because it has an address book and can be referenced in policies. The Global zone satisfies these criteria. However, it does not have one element that all other security zones have—an interface. The Global zone serves as a storage area for Mapped IP (MIP) and Virtual IP (VIP) addresses. The predefined Global zone address "Any" applies to all MIPs, VIPs, and other user-defined addresses set in the Global zone. Because traffic going to these addresses is mapped to other addresses, the Global zone does not require an interface for traffic to flow through it.

The Global zone also contains addresses for use in global policies. For information about global policies, see "Global Policies" on page 200.

NOTE: Any policy that uses the Global zone as its destination cannot support NAT or traffic shaping.

SCREEN Options

A Juniper Networks firewall secures a network by inspecting, and then allowing or denying, all connection attempts that require passage from one security zone to another. For every security zone, and the MGT zone, you can enable a set of predefined SCREEN options that detect and block various kinds of traffic that the security device determines as potentially harmful.

For more information about available SCREEN options, see "Attack Detection and Defense Mechanisms" on page 431 .

Binding a Tunnel Interface to a Tunnel Zone

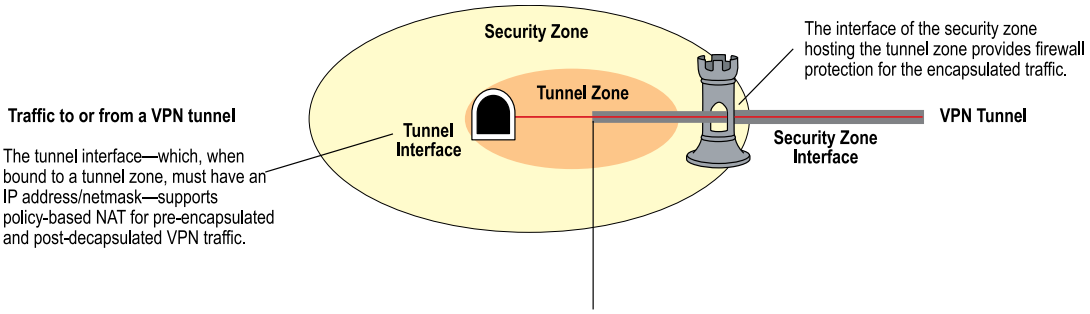
A tunnel zone is a logical segment that hosts one or more tunnel interfaces. A tunnel zone is conceptually affiliated with a security zone in a “child-parent” relationship. The security zone acting as the “ parent,” which you can also conceive of as a carrier zone, provides the firewall protection to the encapsulated traffic. The tunnel zone provides packet encapsulation/decapsulation, and—by supporting tunnel interfaces with IP addresses and netmasks that can host Mapped IP (MIP) addresses and Dynamic IP (DIP) pools—can also provide policy-based NAT services.

The security device uses the routing information for the carrier zone to direct traffic to the tunnel endpoint. The default tunnel zone is Untrust-Tun, and it is associated with the Untrust zone. You can create other tunnel zones and bind them to other security zones, with a maximum of one tunnel zone per carrier zone per virtual system.

NOTE: The root system and all virtual systems can share the Untrust zone. However, each system has its own separate Untrust-Tun zone.

By default, a tunnel zone is in the trust-vr routing domain, but you can also move a tunnel zone into another routing domain. See Figure 17 on page 46.

Figure 17: Tunnel Zone Routing Domain



When upgrading from a version of ScreenOS earlier than 3.1.0, existing tunnel interfaces are bound by default to the preconfigured Untrust-Tun tunnel zone, which is a “child” of the preconfigured Untrust security zone. You can bind multiple tunnel zones to the same security zone; however, you cannot bind a tunnel zone to another tunnel zone.

In this example, you create a tunnel interface and name it tunnel.3. You bind it to the Untrust-Tun zone, and assign it IP address 3.3.3.3/24. You then define a Mapped IP (MIP) address on tunnel.3, translating 3.3.3.5 to 10.1.1.5, which is the address of a server in the Trust zone. Both the Untrust zone, which is the carrier zone for the Untrust-Tun zone, and the Trust zone are in the trust-vr routing domain.

WebUI

1. **Tunnel Interface**

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.3
   Zone (VR): Untrust-Tun (trust-vr)
   Fixed IP: (select)
   IP Address / Netmask: 3.3.3.3/24

2. **MIP**

   Network > Interfaces > Edit (for tunnel.3) > MIP > New: Enter the following, then click **OK**:

   Mapped IP: 3.3.3.5
   Host IP Address: 10.1.1.5
   Netmask: 255.255.255.255
   Host Virtual Router Name: trust-vr

CLI

1. **Tunnel Interface**

   ```
   set interface tunnel.3 zone Untrust-Tun
   set interface tunnel.3 ip 3.3.3.3/24
   ```

2. **MIP**

   ```
   set interface tunnel.3 mip 3.3.3.5 host 10.1.1.5
   save
   ```

Configuring Security Zones and Tunnel Zones

For best performance, always save your changes after creating a zone. The processes for creating, modifying, and deleting Layer 3 or Layer 2 security zones and tunnel zones are quite similar.

NOTE: You cannot delete predefined security zones or the predefined tunnel zone, although you can edit them.

Creating a Zone

To create a Layer 3 or Layer 2 security zone, or to create a tunnel zone, use either the WebUI or CLI.

WebUI

Network > Zones > New: Enter the following, then click **OK**:

Zone Name: Type a name for the zone.

Virtual Router Name: Select the virtual router in whose routing domain you want to place the zone.

Zone Type:

Select **Layer 3** to create a zone to which you can bind interfaces in NAT or route mode.

Select **Layer 2** to create a zone to which you can bind interfaces in transparent mode.

Select **Tunnel Out Zone** when creating a tunnel zone and binding it to a carrier zone, then select a specific carrier zone from the drop-down list.

Block Intra-Zone Traffic: Select this option to block traffic between hosts within the same security zone. By default, intra-zone blocking is disabled.

NOTE: The name of a Layer 2 security zone must begin with "L2-" ; for example, " L2-Corp" or " L2-XNet."

CLI

```
set zone name zone [ l2 vlan_id_num | tunnel sec_zone ]
set zone zone block
set zone zone vrouter name_str
save
```

NOTE: When creating a Layer 2 security zone, the VLAN ID number must be 1 (for VLAN1).

Modifying a Zone

To modify the name of a security zone or tunnel zone, or to change the carrier zone for a tunnel zone, you must first delete the zone and then create it again with the changes. You can change the intra-zone blocking option and the virtual router on an existing zone.

NOTE: Before you can remove a zone, you must first unbind all interfaces bound to it.
Before you can change the virtual router for a zone, you must first remove any interfaces bound to it.

WebUI

1. **Modifying the Zone Name**

   Network > Zones: Click **Remove** (for the security zone or tunnel zone whose name you want to change or for the tunnel zone whose carrier zone you want to change).

   When the prompt appears, asking for confirmation of the removal, click **Yes**.

   Network > Zones > New: Enter the zone settings with your changes, then click **OK**.

2. **Changing the Intra-Zone Blocking Option or Virtual Router**

   Network > Zones > Edit (for the zone that you want to modify): Enter the following, then click **OK**:

   Virtual Router Name: From the drop-down list, select the virtual router into whose routing domain you want to move the zone.

   Block Intra-Zone Traffic: To enable, select the check box. To disable, clear it.

CLI

1. **Modifying the Zone Name**

```
unset zone zone
set zone name zone [ l2 vlan_id_num | tunnel sec_zone ]
```

2. **Changing the Intra-Zone Blocking Option or Virtual Router**

```
{ set | unset } zone zone block
set zone zone vrouter name_str
save
```

Deleting a Zone

For best performance, always save your changes and reboot after deleting a zone. To delete a security zone or tunnel zone, do either of the following:

WebUI

Network > Zones: Click **Remove** (for the zone you want to delete).

When the prompt appears, asking for confirmation of the removal, click **Yes**.

CLI

```
unset zone zone
```

```
save
```

NOTE: Before you can remove a zone, you must first unbind all interfaces bound to it. To unbind an interface from a zone, see "Binding an Interface to a Security Zone" on page 60.

Function Zones

The five function zones are Null, MGT, HA, Self, and VLAN. Each zone exists for a single purpose, as explained in Table 3 on page 50.

Table 3: Function Zones

Zone	Description
Null Zone	This zone serves as temporary storage for any interfaces that are not bound to any other zone.
MGT Zone	This zone hosts the out-of-band management interface, MGT. You can set firewall options on this zone to protect the management interface from different types of attacks. For more information about firewall options, see "Attack Detection and Defense Mechanisms" on page 431.
HA Zone	This zone hosts the high availability interfaces, HA1 and HA2. Although you can set interfaces for the HA zone, the zone itself cannot be configured.
Self Zone	This zone hosts the interface for remote management connections. When you connect to the security device via HTTP, SCS, or Telnet, you connect to the Self zone.
VLAN Zone	This zone hosts the VLAN1 interface, which you use to manage the device and terminate VPN traffic when the device is in transparent mode. You can also set firewall options on this zone to protect the VLAN1 interface from various attacks.

Chapter 4
Interfaces

Physical interfaces and subinterfaces allow traffic to enter and exit a security zone. To allow network traffic to flow in and out of a security zone, you must bind an interface to that zone and, if it is a Layer 3 zone, assign it an IP address. Then, you must configure policies to allow traffic to pass from interface to interface between zones. You can assign multiple interfaces to a zone, but you cannot assign a single interface to multiple zones.

This chapter contains the following sections:

Interface Types

This section describes logical interfaces, function zone interfaces, and tunnel interfaces. For information about viewing a table of all these interfaces, see “Viewing Interfaces” on page 59.

Logical Interfaces

The purpose of logical interfaces is to provide an opening through which network traffic can pass between zones. ScreenOS supports the following types of logical Interfaces:

Physical Interfaces

The name of a physical interface is composed of the media type, slot number (for some devices), and index number, for example, ethernet3/2, ethernet0/2, wireless2, wireless0/2, bgroup2, serial0/0, serial0/2, bri0/0, or adsl0/2. You can bind a physical interface to any security zone where it acts as a doorway through which traffic enters and exits the zone. Without an interface, no traffic can enter or leave the zone.

On security devices that support changes to interface-to-zone bindings, three of the physical ethernet interfaces are prebound to specific Layer 3 security zones—Trust, Untrust, and DMZ. Which interface is bound to which zone is specific to each platform. (For more information about security zones, see“Configuring Security Zone Interfaces” on page 60 “Configuring Security Zone Interfaces” on page 60.)

Wireless Interfaces

A wireless interface, like a physical interface, acts as a doorway through which traffic enters and exits a security zone. Each wireless security device allows up to four wireless interfaces (wireless0/0 — wireless0/3) to be active simultaneously.

A wireless interface cannot be bound to the Untrust security zone. (For more information, see “Wireless Local Area Network” on page 2001.)

Bridge Group Interfaces

Some security devices support bridge groups (bgroups). Bgroups let you group multiple Ethernet and wireless interfaces together. Each bgroup constitutes its own broadcast domain and provides high-speed Ethernet switching between interfaces within the group. You can assign a single IP address to each bgroup interface. You can bind a bgroup interface to any zone.

For devices that are preconfigured with bridge groups, there are two different bridge group numbering systems. On some devices, the preconfigured bgroup interfaces are identified as bgroup0 through bgroup3. On other devices the preconfigured bgroup interfaces are identified as bgroup0/0 through bgroup0/2.

On some devices that support field-installable modules such as PIMs or mini-PIMs, you can create bgroups containing some or all of the Ethernet ports on the module. On these devices, you create bgroups identified as bgroupx/y, where x is the slot number for the module containing the grouped ports and y is a number you assign when creating the bridge group.

You can bind a bridge group interface to any zone. (For more information, see “Binding an Interface to a Security Zone” on page 60.)

Subinterfaces

A subinterface, like a physical interface, acts as a doorway through which traffic enters and exits a security zone. You can logically divide a physical interface into several virtual subinterfaces. Each virtual subinterface borrows the bandwidth it needs from the physical interface from which it stems, thus its name is an extension of the physical interface name, for example, ethernet3/2.1 or ethernet0/2.1

You can bind a subinterface to any Layer 3 zone. You can bind a subinterface to the same zone as its physical interface, or you can bind it to a different zone. (For more information, see "Binding an Interface to a Security Zone" on page 60.)

Aggregate Interfaces

Some security devices support aggregate interfaces. An aggregate interface is the accumulation of two or more physical interfaces that share the traffic load directed to the IP address of the aggregate interface equally among themselves. By using an aggregate interface, you can increase the amount of bandwidth available to a single IP address. Also, if one member of an aggregate interface fails, the other member or members can continue processing traffic—although with less bandwidth than previously available.

NOTE: For more information about aggregate interfaces, see "Interface Redundancy and Failover" on page 1817.

Redundant Interfaces

You can bind two physical interfaces together to create one redundant interface, which you can then bind to a security zone. One of the two physical interfaces acts as the primary interface and handles all the traffic directed to the redundant interface. The other physical interface acts as the secondary interface and stands by in case the active interface experiences a failure. If that occurs, traffic to the redundant interface fails over to the secondary interface, which becomes the new primary interface. The use of redundant interfaces provides a first line of redundancy before escalating a failover to the device level.

NOTE: For more information about redundant interfaces, see "Interface Redundancy and Failover" on page 1817.

Virtual Security Interfaces

Virtual security interfaces (VSIs) are the virtual interfaces that two security devices forming a virtual security device (VSD) share when operating in HA mode. Network and VPN traffic use the IP address and virtual MAC address of a VSI. The VSD then maps the traffic to the physical interface, subinterface, or redundant interface to which you have previously bound the VSI. When two security devices are operating in HA mode, you must bind security zone interfaces that you want to provide

uninterrupted service in the event of a device failover to one or more virtual security devices (VSDs). When you bind an interface to a VSD, the result is a virtual security interface (VSI).

NOTE: For more information about VSIs and how they function with VSDs in an HA cluster, see "High Availability" on page 1763.

Function Zone Interfaces

Function zone interfaces, such as Management and HA, serve a special purpose.

Management Interfaces

On some security devices, you can manage the device through a separate physical interface—the Management (MGT) interface—moving administrative traffic outside the regular network user traffic. Separating administrative traffic from network user traffic greatly increases security and ensures constant management bandwidth.

NOTE: For information about configuring the device for administration, see "Administration" on page 309 .

High Availability Interfaces

The high availability (HA) interface is a physical port used exclusively for HA functions. In a redundant group, one unit serves as the primary device—performing network firewall, VPN, and traffic-shaping functions—while the other unit serves as the backup device, waiting to take over the firewall functions when the primary unit fails. This is an Active/Passive configuration. You can also set up both members of the cluster to be primary and backup for each other. This is an Active/Active configuration. Both configurations are explained fully in "High Availability" on page 1763.

Virtual HA Interfaces

On security devices without a dedicated HA interface, a virtual HA interface provides the same functionality. Because there is no separate physical port used exclusively for HA traffic, you must bind the virtual HA interface to one of the physical ethernet ports. You use the same procedure for binding a network interface to the HA zone as you do for binding a network interface to a security zone.

Tunnel Interfaces

A tunnel interface acts as a doorway to a VPN tunnel. Traffic enters and exits a VPN tunnel via a tunnel interface.

When you bind a tunnel interface to a VPN tunnel, you can reference that tunnel interface in a route to a specific destination and then reference that destination in one or more policies. With this approach, you can finely control the flow of traffic

through the tunnel. It also provides dynamic routing support for VPN traffic. When there is no tunnel interface bound to a VPN tunnel, you must specify the tunnel in the policy itself and choose **tunnel** as the action. Because the action **tunnel** implies permission, you cannot specifically deny traffic from a VPN tunnel.

You can perform policy-based NAT on outgoing or incoming traffic using a pool of Dynamic IP (DIP) addresses in the same subnet as the tunnel interface. A typical reason for using policy-based NAT on a tunnel interface is to avoid IP address conflicts between the two sites on either end of the VPN tunnel.

You must bind a route-based VPN tunnel to a tunnel interface so that the security device can route traffic to and from it. You can bind a route-based VPN tunnel to a tunnel interface that is either numbered (with IP address/netmask) or unnumbered (without IP address/netmask). If the tunnel interface is unnumbered, you must specify an interface from which the tunnel interface borrows an IP address. The security device only uses the borrowed IP address as a source address when the security device itself initiates traffic—such as OSPF messages—through the tunnel. The tunnel interface can borrow the IP address from an interface in the same security zone or from an interface in a different one as long as both zones are in the same routing domain.

You can achieve very secure control of VPN traffic routing by binding all the unnumbered tunnel interfaces to one zone, which is in its own virtual routing domain, and borrowing the IP address from a loopback interface bound to the same zone. For example, you can bind all the unnumbered tunnel interfaces to a user-defined zone named “VPN” and configure them to borrow an IP address from the loopback.1 interface, also bound to the VPN zone. The VPN zone is in a user-defined routing domain named “ vpn-vr.” You put all destination addresses to which the tunnels lead in the VPN zone. Your routes to these addresses point to the tunnel interfaces, and your policies control VPN traffic between other zones and the VPN zone, see Figure 18 on page 55.

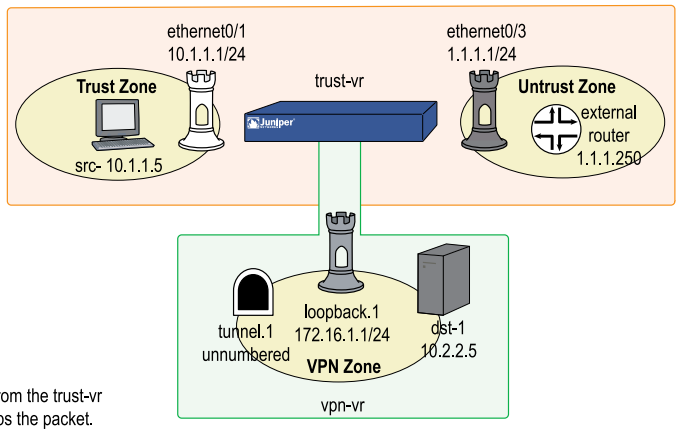
Figure 18: Unnumbered Tunnel Interface Bindings

```
set vrouter name vpn-vr
set zone name vpn vrouter vpn-vr
set interface loopback.1 zone vpn
set interface loopback.1 ip 172.16.1.1/24
set interface tunnel.1 zone vpn
set interface tunnel.1 ip unnumbered loopback.1

Configure addresses for src-1 and dst-1.
Configure a VPN tunnel and bind it to tunnel.1.

set vrouter trust-vr route 10.2.2.5/32 vrouter vpn-vr
set vrouter trust-vr route 0.0.0.0/0 interface ethernet0/3
      gateway 1.1.1.250
set vrouter vpn-vr route 10.2.2.5 interface tunnel.1

set policy from trust to vpn scr-1 dst-1 any permit
```



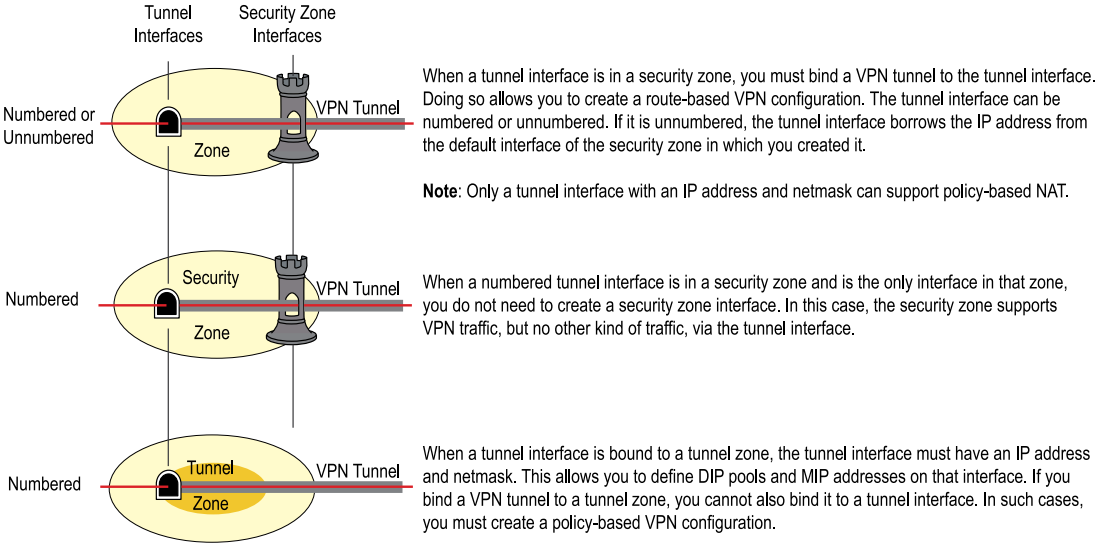
The security device sends traffic destined for 10.2.2.5/32 from the trust-vr to the vpn-vr. If tunnel.1 becomes disabled, the device drops the packet. Because the default route (to 0.0.0.0/0) is only in the trust-vr, the device does not attempt to send the packet in plain text out ethernet0/3.

Putting all the tunnel interfaces in such a zone is very secure because there is no chance for the failure of a VPN, which causes the route to the associated tunnel interface to become inactive, to redirect traffic intended for tunneling to use a non-tunneled route—such as the default route. (For several suggestions about how to avoid such a problem, see *"Route-Based Virtual Private Network Security Considerations" on page 793*.)

You can also bind a tunnel interface to a tunnel zone. When you do, it must have an IP address. The purpose of binding a tunnel interface to a tunnel zone is to make NAT services available for policy-based VPN tunnels. See Figure 19 on page 56.

NOTE: Network Address Translation (NAT) services include Dynamic IP (DIP) pools and Mapped IP (MIP) addresses defined in the same subnet as an interface.

Figure 19: Tunnel Interface to Zone Binding



Conceptually, you can view VPN tunnels as pipes that you have laid. They extend from the local device to remote gateways, and the tunnel interfaces are the openings to these pipes. The pipes are always there, available for use whenever the routing engine directs traffic to one of their interfaces.

Generally, assign an IP address to a tunnel interface if you want the interface to support one or more Dynamic IP (DIP) pools for source address translation (NAT-src) and Mapped IP (MIP) addresses for destination address translation (NAT-dst). For more information about VPNs and address translation, see *"VPN Sites with Overlapping Addresses" on page 863*. You can create a tunnel interface with an IP address and netmask in either a security or tunnel zone.

If the tunnel interface does not need to support address translation, and your configuration does not require the tunnel interface to be bound to a tunnel zone, you can specify the interface as unnumbered. You must bind an unnumbered tunnel interface to a security zone; you cannot bind it to a tunnel zone. You must also specify

an interface with an IP address that is in the same virtual routing domain as the security zone to which the unnumbered interface is bound. The unnumbered tunnel interface borrows the IP address from that interface.

NOTE: For examples showing how to bind a tunnel interface to a tunnel, see the route-based VPN examples in *"Site-to-Site VPN Configurations" on page 801* and *"Dialup Virtual Private Networks" on page 887*.

If you are transmitting multicast packets through a VPN tunnel, you can enable Generic Routing Encapsulation (GRE) on the tunnel interfaces to encapsulate multicast packets in unicast packets. Juniper Networks security devices support GREv1 for encapsulating IP packets in IPv4 unicast packets. For additional information about GRE, see *"Configuring Generic Routing Encapsulation on Tunnel Interfaces" on page 1394*.

The tunnel interface becomes the VSI (for a backup VSD group) when NSRP is enabled.

The tunnel interface inherits the VSD group ID from the carrier interface (a VSI) and sets the VSI flag. The VSI flag and VSD group ID of the tunnel interface both change when the carrier interface changes from VSI to local and from local to VSI.

NOTE: If the carrier interface is not a VSI, the VSD group ID is set to 0, the default, and the VSI flag is reset. For more information about VSIs and how they function with VSDs in an HA cluster, see *"High Availability" on page 1763*.

The tunnel interface, being a logical interface, has no MAC address or physical interface link state. The different logical link states for a tunnel interface are **up, down, ready**, and **inactive**.

If a VPN configuration forces changes on the carrier interface, the tunnel interface also changes its VSD group ID and VSI flag. When the carrier interface is in the inactive state, the tunnel interface link state is set to **inactive**.

The following table compares the link states for a carrier interface and a tunnel interface on a backup VSD group:

VSD Group State	Carrier Interface Link State	Tunnel Interface Link State
Backup	Inactive	Inactive
Backup	Down	Down

If a VSD group changes from primary to backup, the tunnel interface with the same VSD group ID changes its link state from **up/down/ready** to **inactive**. Similarly, if a VSD group changes from backup to primary, the tunnel interface with the same VSD group ID changes its link state from inactive to **up/down/ready** according to VPN logic.

Deleting Tunnel Interfaces

You cannot immediately delete a tunnel interface that hosts Mapped IP addresses (MIPs) or Dynamic IP (DIP) address pools. Before you delete a tunnel interface hosting any of these features, you must first delete any policies that reference them. Then you must delete the MIPs and DIP pools on the tunnel interface. Also, if a route-based VPN configuration references a tunnel interface, you must first delete the VPN configuration before you can delete the tunnel interface.

In this example, tunnel interface tunnel.2 is linked to DIP pool 8. DIP pool 8 is referenced in a policy (ID 10) for VPN traffic from the Trust zone to the Untrust zone through a VPN tunnel named vpn1. To remove the tunnel interface, you must first delete the policy (or remove the reference to DIP pool 8 from the policy), and then the DIP pool. Then, you must unbind tunnel.2 from vpn1. After removing all the configurations that depend on the tunnel interface, you can then delete it.

WebUI

1. **Deleting Policy 10, Which References DIP Pool 8**

   Policy > Policies (From: Trust, To: Untrust): Click **Remove** for Policy ID 10.

2. **Deleting DIP Pool 8, Which Is Linked to tunnel.2**

   Network > Interfaces > Edit (for tunnel.2) > DIP: Click **Remove** for DIP ID 8.

3. **Unbinding tunnel.2 from vpn1**

   VPNs > AutoKey IKE > Edit (for vpn1) > Advanced: Select **None** in the Bind to: Tunnel Interface drop-down list, click **Return**, and then click **OK**.

4. **Deleting tunnel.2**

   Network > Interfaces: Click **Remove** for tunnel.2.

CLI

1. **Deleting Policy 10, Which References DIP Pool 8**

   ```
   unset policy 10
   ```

2. **Deleting DIP Pool 8, Which Is Linked to tunnel.2**

   ```
   unset interface tunnel.2 dip 8
   ```

3. **Unbinding tunnel.2 from vpn1**

   ```
   unset vpn vpn1 bind interface
   ```

4. **Deleting tunnel.2**

   ```
   unset interface tunnel.2
   save
   ```

Viewing Interfaces

You can view a table that lists all interfaces on your security device. Because they are predefined, physical interfaces are listed regardless of whether or not you configure them. Subinterfaces and tunnel interfaces are only listed once you create and configure them.

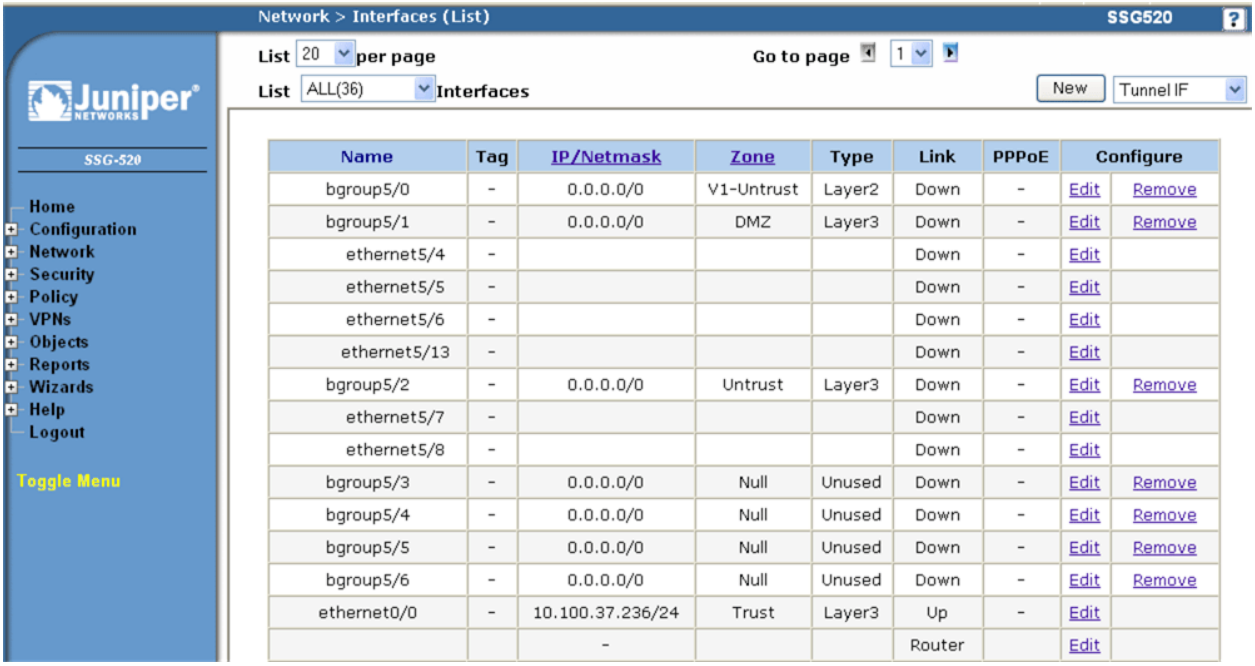
To view the interface table in the WebUI, click **Network > Interfaces**. You can specify the types of interfaces to display from the List Interfaces drop-down list.

To view the interface table in the CLI, use the **get interface** command.

The interface table displays the following information about each interface:

- **Name:** This field identifies the name of the interface.
- **IP/Netmask:** This field identifies the IP address and netmask address of the interface.
- **Zone:** This field identifies the zone to which the interface is bound.
- **Type:** This field indicates the interface type: Layer 2, Layer 3, tunnel, redundant, aggregate, VSI.
- **Link:** This field identifies whether the interface is active (up) or inactive (down).
- **Configure:** This field allows you modify or remove interfaces.

Figure 20: WebUI Interface Table



Name	Tag	IP/Netmask	Zone	Type	Link	PPPoE	Configure	
bgroup5/0	-	0.0.0.0/0	V1-Untrust	Layer2	Down	-	Edit	Remove
bgroup5/1	-	0.0.0.0/0	DMZ	Layer3	Down	-	Edit	Remove
ethernet5/4	-				Down	-	Edit	
ethernet5/5	-				Down	-	Edit	
ethernet5/6	-				Down	-	Edit	
ethernet5/13	-				Down	-	Edit	
bgroup5/2	-	0.0.0.0/0	Untrust	Layer3	Down	-	Edit	Remove
ethernet5/7	-				Down	-	Edit	
ethernet5/8	-				Down	-	Edit	
bgroup5/3	-	0.0.0.0/0	Null	Unused	Down	-	Edit	Remove
bgroup5/4	-	0.0.0.0/0	Null	Unused	Down	-	Edit	Remove
bgroup5/5	-	0.0.0.0/0	Null	Unused	Down	-	Edit	Remove
bgroup5/6	-	0.0.0.0/0	Null	Unused	Down	-	Edit	Remove
ethernet0/0	-	10.100.37.236/24	Trust	Layer3	Up	-	Edit	
		-			Router		Edit	

Figure 21: CLI Interface Table

```
ns500-> get interface

A - Active, I - Inactive, U - Up, D - Down, R - Ready

Interfaces in vsys Root:
Name           IP Address          Zone         MAC             VLAN State VSD Vsys
eth1/1         0.0.0.0/0           Null         0010.db0d.4ddc    -   D    -   Root
eth1/2         10.100.37.155/24    Untrust      0010.db0d.4dde    -   U    -   Root
eth2/1         0.0.0.0/0           Null         0010.db0d.4ddb    -   D    -   Root
eth2/2         1.1.2.5/24          Untrust      0010.db0d.4ddd    -   D    -   Root
eth3/1         2.2.2.0/24          Untrust      0010.db0d.4dd8    -   D    -   Root
eth3/2         10.1.2.155/24       Trust        0010.db0d.4dda    -   U    -   Root
eth4/1         3.3.3.0/24          Untrust      0010.db0d.4dd7    -   D    -   Root
eth4/2         0.0.0.0/0           Null         0010.db0d.4dd9    -   D    -   Root
mgt            0.0.0.0/0           MGT          0010.db0d.4dd0    -   D    -   Root
ha1            0.0.0.0/0           HA           0010.db0d.4dd5    -   D    -   Root
ha2            0.0.0.0/0           HA           0010.db0d.4dd6    -   D    -   Root
vlan1          0.0.0.0/0           VLAN         0010.db0d.4ddf    1   D    -   Root
null           0.0.0.0/0           Null         0010.dbff.0100    -   U    0   Root
ns500->
```

Configuring Security Zone Interfaces

This section describes how to configure the following aspects of security zone interfaces:

- Binding and unbinding an interface to a security zone
- Assigning an address to a Layer 3 (L3) security zone interface
- Modifying physical interfaces and subinterfaces
- Creating subinterfaces
- Deleting subinterfaces

NOTE: For information about setting traffic bandwidth for an interface, see “Traffic Shaping” on page 233. For more information about the management and other services options available per interface, see “Controlling Administrative Traffic” on page 339.

Binding an Interface to a Security Zone

You can bind some physical interfaces to either an L2 or L3 security zone. WAN interfaces, except for ADSL, cannot be bound to L2 security zones. You can bind a subinterface only to an L3 security zone because a subinterface requires an IP address. You can only assign an IP address to an interface after you have bound it to an L3 security zone. Wireless interfaces cannot be bound to the Untrust security zone.

Some security devices allow you to group multiple interfaces. Before adding an interface to a group, the interface must be set to the Null security zone. After interfaces are added to a group, the group interface must be assigned to a security zone for connection to be established.

You can configure Layer 3 interfaces that are bound to security zones to accept or reject gratuitous ARP (G-ARP) requests and replies. Such interfaces include physical Ethernet interfaces, subinterfaces, redundant interfaces, aggregate interfaces, bgroup interfaces, and management (MGT) interfaces. This setting is not supported on loopback interface, tunnel interface, all serial interfaces (including serial interface,dialer interface, multilink interface on wan interface), dsl interface, wan interface (including t1 e1 isdn interface) and wlan interface.

You can, similarly, configure Layer 2 zones such as V1-Trust, V1-Untrust, V1-DMZ or some user-defined L2-zone and VLANx interface to accept or reject G-ARP requests and replies.

Gratuitous ARP is used to update the ARP cache of interfaces in a network. In addition, G-ARP helps detect IP conflicts. When an interface receives an ARP request containing a source IP that matches its own, it indicates an IP conflict. Also, frequent G-ARP transmissions could indicate bad Ethernet cabling or hardware.

You can configure an interface to reject G-ARP requests and replies based on your security concerns. Accepting gratuitous ARP requests and replies might make the network vulnerable to ARP spoofing attacks. However, you should enable G-ARP in HA environments where the next upstream or downstream device is also in an HA configuration and does not use virtual MAC addresses.

If the connected HA device fails, the Virtual IP (VIP) shifts to the backup device. Because the backup device uses its own actual MAC instead of a shared virtual MAC, to ensure proper forwarding of traffic you should accept G-ARP request from the connected HA device to understand the change in the network.

In this example, you bind ethernet0/5 to the Trust zone.

WebUI

Network > Interfaces > Edit (for ethernet0/5): Select **Trust** from the Zone Name drop-down list, then click **Apply**.

CLI

```
set interface ethernet0/5 zone trust
save
```

In this example, you set ethernet0/3 and ethernet0/4 to be in the Null security zone, group the interfaces in bgroup1, then bind the group to the DMZ security zone:

WebUI

Network > Interfaces > Edit (for ethernet0/3): Select **Null** from the Zone Name drop-down list, then click **Apply**.

Network > Interfaces > Edit (for ethernet0/4): Select **Null** from the Zone Name drop-down list, then click **Apply**.

Network > Interfaces > Edit (for bgroup1): Select **DMZ** from the Zone Name drop-down list, then click **Apply**.

Network > Interfaces > Edit (for bgroup1): Check **ethernet0/3** and **ethernet0/4** in the Bind to Current Bgroup column, then click **Apply**.

CLI

```
set interface ethernet0/3 zone null
set interface ehternet0/4 zone null
set interface bgroup1 port ethernet0/3
set interface bgroup1 port ethernet0/4
set interface bgroup1 zone DMZ
save
```

Unbinding an Interface from a Security Zone

If an interface is unnumbered, you can unbind it from one security zone and bind it to another. If an interface is numbered, you first must set its IP address and netmask to 0.0.0.0. Then, you can unbind it from one security zone and bind it to another one, and (optionally) reassign it an IP address/netmask.

In this example, ethernet0/3 has the IP address 210.1.1.1/24 and is bound to the Untrust zone. You set its IP address and netmask to 0.0.0.0/0 and bind it to the Null zone.

WebUI

Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

```
Zone Name: Null
IP Address/Netmask: 0.0.0.0/0
```

CLI

```
set interface ethernet0/3 ip 0.0.0.0/0
set interface ethernet0/3 zone null
save
```

To unbind an interface from a group and reassign it to a different security zone, the interface must be released from the bgroup. Releasing the interface from the bgroup puts the interface in the Null security zone. Once in the Null security zone, the interface can be bound to any security zone then configured with an IP address.

WebUI

Network > Interfaces > Edit (for bgroup1) > Bind Port: Deselect **ethernet0/3** in the Bind to Current Bgroup column, then click **Apply**.

Network > Interfaces > Edit (for ethernet0/3): Select **Trust** from the Zone Name drop-down list, then click **Apply**.

CLI

```
unset interface bgroup1 port ethernet0/3
```

```
set interface ethernet0/3 zone trust
save
```

Addressing an L3 Security Zone Interface

When defining a Layer 3 (L3) security zone interface or subinterface, you must assign it an IP address and a netmask. If you bind the interface to a zone in the trust-vr, you can also specify the interface mode as NAT or route. (If the zone to which you bind the interface is in the untrust-vr, the interface is always in route mode.)

NOTE: For examples of NAT and route mode configurations, see "Interface Modes" on page 99.

The two basic types of IP addresses to be considered when making interface address assignments are as follows:

- Public addresses, which Internet service providers (ISPs) supply for use on a public network like the Internet and which must be unique
- Private addresses, which a local network administrator assigns for use on a private network and which other administrators can assign for use on other private networks as well

NOTE: When you add an IP address to an interface, the security device checks via an ARP request to make sure that the IP address does not already exist on the local network. (The physical link must be up at the time.) If the IP address already exists, a warning is displayed.

Public IP Addresses

If an interface connects to a public network, it must have a public IP address. Also, if an L3 security zone in the untrust-vr connects to a public network and the interfaces of zones in the trust-vr are in route mode, then all the addresses in the zones in the trust-vr—for interfaces and for hosts—must also be public addresses. Public IP addresses fall into three classes, A, B, and C, as shown in Table 4 on page 63.

Table 4: Public Address Ranges

Address Class	Address Range	Excluded Address Range
A	0.0.0.0 – 127.255.255.255	10.0.0.0 – 10.255.255.255, 127.0.0.0 – 127.255.255.255
B	128.0.0.0 – 191.255.255.255	172.16.0.0 – 172.31.255.255
C	192.0.0.0 – 223.255.255.255	192.168.0.0 – 192.168.255.255

NOTE: There are also D and E class addresses, which are reserved for special purposes.

An IP address is composed of four octets, each octet being 8 bits long. In a class A address, the first 8 bits indicate the network ID, and the final 24 bits indicate the host ID (nnn.hhh.hhh.hhh). In a class B address, the first 16 bits indicate the network ID, and the final 16 bits indicate the host ID (nnn.nnn.hhh.hhh). In a class C address, the first 24 bits indicate the network ID, and the final 8 bits indicate the host ID (nnn.nnn.nnn.hhh).

Through the application of subnet masks (or netmasks), you can further divide networks. A netmask essentially masks part of the host ID so that the masked part becomes a subnet of the network ID. For example, the 24-bit mask in the address 10.2.3.4/24 indicates that the first 8 bits (that is, the first octet—010) identify the network portion of this private class A address, the next 16 bits (that is, the second and third octets—002.003) identify the subnetwork portion of the address, and the last 8 bits (the last octet—004) identify the host portion of the address. Using subnets to narrow large network address spaces into smaller subdivisions greatly increases the efficient delivery of IP datagrams.

NOTE: The dotted-decimal equivalent of a 24-bit mask is 255.255.255.0

Private IP Addresses

If an interface connects to a private network, a local network administrator can assign it any address, although it is conventional to use an address from the range of addresses reserved for private use—10.0.0.0/8, 172.16.0.0 – 172.31.255.255, 192.168.0.0/16— as defined in RFC 1918, *Address Allocation for Private Internets*.

If an L3 security zone in the untrust-vr connects to a public network and the interfaces bound to zones in the trust-vr are in NAT mode, then all the addresses in the zones in the trust-vr—for interfaces and for hosts—can be private addresses.

Addressing an Interface

In this example, you assign ethernet0/5 the IP address 210.1.1.1/24 and give it the Manage IP address 210.1.1.5. (Note that the Manage IP address must be in the same subnet as the security zone interface IP address.) Finally, you set the interface in NAT mode, which translates all internal IP addresses to the default interfaces bound to the other security zones.

NOTE: The default interface in a security zone is the first interface bound to the zone. To learn which interface is the default interface for a zone, see the Default IF column on Network > Zones in the WebUI, or the Default-If column in the output from the **get zone** command in the CLI.

WebUI

Network > Interfaces > Edit (for ethernet0/5): Enter the following, then click **OK**:

```
IP Address/Netmask: 210.1.1.1/24
Manage IP: 210.1.1.5
```

CLI

```
set interface ethernet0/5 ip 210.1.1.1/24
set interface ethernet0/5 manage-ip 210.1.1.5
save
```

Modifying Interface Settings

After you have configured a physical interface, a subinterface, a redundant interface, an aggregate interface, or a Virtual Security Interface (VSI), you can later change any of the following settings should the need arise:

- IP address and netmask.
- Manage IP address.
- (L3 zone interfaces) Management and network services.
- (Subinterface) Subinterface ID number and VLAN tag number.
- (Interfaces bound to L3 security zones in the trust-vr) interface mode—NAT or route.
- (Physical interface) Traffic bandwidth settings (see "Traffic Shaping" on page 233).
- (Physical, redundant, and aggregate interfaces) Maximum Transmission Unit (MTU) size.
- (L3 interfaces) Block traffic from coming in and going out the same interface, including traffic between a primary and secondary subnet or between secondary subnets (this is done using the CLI **set interface** command with the **route-deny** option).

For physical interfaces on some security devices, you can force the physical state of the link to be down or up. By forcing the physical state of the link to be down, you can simulate a disconnect of the cable from the interface port. (This is done with the CLI **set interface** command with the **phy link-down** option.)

In this example, you make some modifications to ethernet0/1, an interface bound to the Trust zone. You change the Manage IP address from 10.1.1.2 to 10.1.1.12. To enforce tighter security of administrative traffic, you also change the management services options, enabling SCS and SSL and disabling Telnet and WebUI.

WebUI

Network > Interfaces > Edit (for ethernet0/1): Make the following modifications, then click **OK**:

Manage IP: 10.1.1.12
Management Services: (select) SSH, SSL; (clear) Telnet, WebUI

CLI

```
set interface ethernet0/1 manage-ip 10.1.1.12
set interface ethernet0/1 manage ssh
set interface ethernet0/1 manage ssl
unset interface ethernet0/1 manage telnet
unset interface ethernet0/1 manage web
save
```

Creating a Subinterface in the Root System

You can create a subinterface on any physical interface in the root system or virtual system. A subinterface makes use of VLAN tagging to distinguish traffic bound for it from traffic bound for other interfaces. Note that although a subinterface stems from a physical interface, from which it borrows the bandwidth it needs, you can bind a subinterface to any zone, not necessarily that to which its "parent" interface is bound. Additionally, the IP address of a subinterface must be in a different subnet from the IP addresses of all other physical interfaces and subinterfaces.

NOTE: You can also configure subinterfaces on redundant interfaces and VSIs. For an example that includes the configuration of a subinterface on a redundant interface, see "Virtual System Failover" on page 1832.

In this example, you create a subinterface for the Trust zone in the root system. You configure the subinterface on ethernet0/1, which is bound to the Trust zone. You bind the subinterface to a user-defined zone named "accounting," which is in the trust-vr. You assign it subinterface ID 3, IP address 10.2.1.1/24, and VLAN tag ID 3. The interface mode is NAT.

WebUI

Network > Interfaces > New Sub-IF: Enter the following, then click **OK**:

Interface Name: ethernet0/1.3
Zone Name: accounting
IP Address/Netmask: 10.2.1.1/24
VLAN Tag: 3

CLI

```
set interface ethernet0/1.3 zone accounting
set interface ethernet0/1.3 ip 10.2.1.1/24 tag 3
save
```

Deleting a Subinterface

You cannot immediately delete a subinterface that hosts Mapped IP addresses (MIPs), Virtual IP addresses (VIPs), or Dynamic IP (DIP) address pools. Before you delete a subinterface hosting any of these features, you must first delete any policies or IKE gateways that reference them. Then you must delete the MIPs, VIPs, and DIP pools on the subinterface.

In this example, you delete the subinterface ethernet0/1.1.

WebUI

Network > Interfaces: Click **Remove** for ethernet0/1.1.

A system message prompts you to confirm the removal.

Click **Yes** to delete the subinterface.

CLI

```
unset interface ethernet0/1.1
save
```

Creating a Secondary IP Address

Each ScreenOS interface has a single, unique primary IP address. However, some situations demand that an interface have multiple IP addresses. For example, an organization might have additional IP address assignments and might not wish to add a router to accommodate them. In addition, an organization might have more network devices than its subnet can handle, as when there are more than 254 hosts connected to a LAN. To solve such problems, you can add secondary IP addresses to an interface in the Trust, DMZ, or user-defined zone.

NOTE: You cannot set multiple secondary IP addresses for interfaces in the Untrust zone.

Secondary addresses have certain properties that affect how you can implement such addresses. These properties are as follows:

- There can be no subnet address overlap between any two secondary IP addresses. In addition, there can be no subnet address overlap between a secondary IP and any existing subnet on the security device.
- When you manage a security device through a secondary IP address, the address always has the same management properties as the primary IP address. Consequently, you cannot specify a separate management configuration for the secondary IP address.

- You cannot configure a gateway for a secondary IP address.
- Whenever you create a new secondary IP address, the security device automatically creates a corresponding routing table entry. When you delete a secondary IP address, the device automatically deletes its routing table entry.

Enabling or disabling routing between two secondary IP addresses causes no change in the routing table. For example, if you disable routing between two such addresses, the security device drops any packets directed from one interface to the other, but no change occurs in the routing table.

In this example, you set up a secondary IP address—192.168.2.1/24—for ethernet0/1, an interface that has IP address 10.1.1.1/24 and is bound to the Trust zone.

WebUI

Network > Interfaces > Edit (for ethernet0/1) > Secondary IP: Enter the following, then click **Add**:

IP Address/Netmask: 192.168.2.1/24

CLI

```
set interface ethernet0/1 ip 192.168.2.1/24 secondary
save
```

Backup System Interfaces

The interface backup feature allows you to configure a backup interface that can take over traffic from a configured primary interface. You can back up any type of interface with any other type of interface supported on the platform. The only requirement is that both interfaces must be in the untrust zone.

You set a backup interface so that the security device can switch traffic over to it in the event that the primary interface goes down (is unplugged, or fails), destinations on the primary interface become unreachable, or the tunnel bound to the primary interface becomes inactive. When the connection through the primary interface is restored, ScreenOS automatically switches traffic from the backup interface to the primary. The interface backup feature also provides a way for you manually to force the primary interface to switch over to the backup, and to force the backup to switch over to the primary. Each primary interface can have only one backup interface, and each backup interface can have only one primary.

You can configure the security device to switch over to the backup interface when any of the following conditions are met on the primary interface:

- Certain IP addresses become unreachable through the interface.
- Certain VPN tunnels on the interface become unreachable.
- A preconfigured route becomes unreachable through the interface.

For the security device to switch traffic over to a backup interface for any of these reasons, you must first configure the primary interface for that purpose, then configure the backup interface accordingly. You must also configure two default routes, one for the primary interface and one for the backup interface. You can configure the backup interface feature through the WebUI or at the CLI.

Configuring a Backup Interface

ScreenOS determines when to switch over to a backup interface by tracking or monitoring activity on the primary interface. You can configure the following types of backup interfaces:

- IP Tracking
- Tunnel-if tracking
- Route monitoring

Configuring an IP Tracking Backup Interface

In this example, you configure ScreenOS to track IP address 10.1.1.1 on the primary interface (ethernet0/0) and to switch over to the backup interface (dialer1) in the event that this address becomes unreachable. For a discussion of how IP tracking works, see “Interface Failover with IP Tracking” on page 1825.

WebUI

1. **Configure Interfaces**

   Network > Interfaces > (for ethernet0/0) Edit > Monitor > Add: Enter the following, then click **Apply**:

   Track IP: 10.1.1.1
   Weight: 200
   Interval: 2
   Threshold: 5
   Time Out: 1

   Network > Interfaces > Backup: Enter the following, then click **Apply**:

   Primary interface (select): ethernet0/0
   Backup Interface (select): dialer1

2. **Configure Routes**

   Network > Routing > Destination > trust-tr New: Enter the following, then click **Apply**:

   IP Address/Netmask: 0.0.0.0/0
   Interface (select): ethernet0/0

   Network > Routing > Destination > trust-tr New: Enter the following, then click **Apply**:

```
IP Address/Netmask: 0.0.0.0/0
Interface (select): dialer1
```

CLI

1. **Configure Interfaces**

```
set interface ethernet0/0 monitor track-ip
set interface ethernet0/0 monitor track-ip threshold 100
set interface ethernet0/0 monitor trackip ip 10.1.1.1 interval 2
set interface ethernet0/0 monitor trackip ip 10.1.1.1 threshold 5
set interface ethernet0/0 monitor trackip ip 10.1.1.1 weight 200
set interface ethernet0/0 backup interface dialer1 type track-ip
```

2. **Configure Routes**

```
set route 0.0.0.0/0 interface ether0/0
set route 0.0.0.0/0 interface dialer1
save
```

Configuring a Tunnel-if Backup Interface

In this example, you configure a pair of unidirectional VPN tunnels on ethernet0/0—one on Router-1 and one on Router-2—and you configure dialer1 as the backup interface on Route-1. The tunnels connect hosts in the Trust zone at a branch site to an SMTP server in the Trust zone at the corporate site. The zones at each site are in the trust-vr routing domain.

You configure both tunnels with the primary Untrust zone interface (ethernet0/0) as the outgoing interface, and the backup VPN tunnel with the backup Untrust zone interface (dialer1) as the outgoing interface. The security device monitors the primary VPN tunnels to determine when to switch over to the backup. It does this by comparing the backup weight with the VPN monitor threshold. You set the threshold to 100 (**set vpnmonitor threshold 100**) and the backup weight to 200 (**set vpn vpn backup-weight 200**). When the primary interface becomes inactive for any reason, ScreenOS compares the VPN monitor threshold with the backup weight, and if the backup weight is greater than the threshold, it switches the device to the backup.

You also enable the VPN monitor rekey feature. In the event of a failover, this feature enables the security device to revert traffic from the backup interface to the primary if the accumulated weight of the VPN tunnels on the primary interface becomes greater than the VPN monitor threshold.

The security device in the branch site receives its Untrust zone interfaces address, default gateway, and DNS server addresses dynamically from two different ISPs. Each ISP uses a different protocol. ISP-1 uses DHCP to assign an address to ethernet0/0, and ISP-2 uses PPP to assign an address to dialer1. The security device at the corporate site has a static IP address (2.2.2.2). The IP address of its default gateway is 2.2.2.250.

The destination address for VPN monitoring is not the default—the remote gateway IP address (2.2.2.2)—but the addresses of the server (10.2.2.10). If you use the remote

gateway IP address and it becomes unreachable, the primary tunnel always switches over to the backup.

NOTE: Because this example is extensive, only the CLI configuration is included in its entirety. The WebUI section lists the navigational paths to the pages where you can set the various elements of the configuration. You can see what you need to set by referring to the CLI commands.

WebUI (for Router-1)

1. **Configure Interfaces**

   Network > Interfaces > Edit (for ethernet0/0)

   Network > Interfaces > Edit (for bri1/0)

   Network > Interfaces > New Tunnel IF

2. **Configure VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey Advanced > Gateway > New

3. **Configure Asymmetric VPN**

   Network > Zones > Edit (for Trust)

4. **Configure Backup Interface**

   Network > Interfaces > Backup

5. **Configure Routes**

   Network > Routing > Destination > trust-vr

CLI (for Router-1)

1. **Configure Interfaces**

```
set interface ethernet0/0 zone untrust
set interface ethernet0/0 dhcp client
set interface bri2/0 isdn switch-type etsi
set interface dialer1 zone untrust
set dialer pool name pool-1
set interface bri2/0 dialer-pool-member pool-1
set interface dialer1 dialer-pool pool-1
set ppp profile isdn-ppp

set ppp profile isdn-ppp auth type chap
set ppp profile isdn-ppp auth local-name juniper
set ppp profile isdn-ppp auth secret juniper
```

```
set ppp profile isdn-ppp passive
set ppp dialer1 ppp profile isdn-ppp

set interface tunnel.1 untrust
set interface tunnel.1 ip unnumbered interface bgroup0
set interface tunnel.2 untrust
set interface tunnel.2 ip unnumbered interface bgroup0
```

2. **Configure VPN**

```
set ike gateway corp1 address 2.2.2.2 aggressive local-id ssg5ssg20-e0
outgoing-interface ethernet0/0 preshare juniper1 sec-level basic
set ike gateway corp1 address 2.2.2.2 aggressive local-id ssg5ssg20-dialer
outgoing-interface dialer1 preshare juniper2 sec-level basic

set vpn vpn1 gateway corp1 sec-level basic
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 smtp
set vpn vpn1 monitor source-interface bgroup0 destination-ip 10.2.2.10 rekey
set vpn vpn1 backup-weight 200

set vpn vpn2 gateway corp2 sec-level basic
set vpn vpn2 bind interface tunnel.2
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 smtp
set vpn vpn2 monitor source-interface bgroup0 destination-ip 10.2.2.10 rekey
```

3. **Configure Asymmetric VPN**

```
set zone trust asymmetric-vpn
```

4. **Configure Backup Interface**

```
set interface ethernet0/0 backup interface dialer1 type tunnel-if
set vpnmonitor threshold 100
```

5. **Configure Routes**

```
set route 10.2.2.10/32 interface tunnel.1
set route 10.2.2.10/32 interface tunnel.2
set route 0.0.0.0/0 interface ethernet0/0
save
```

WebUI (for Router-2)

1. **Configure Interfaces**

   Network > Interfaces > Edit (for bgroup1)

   Network > Interfaces > Edit (for ethernet0/0)

   Network > Interfaces > New Tunnel IF

2. **Configure Addresses**

   Policy > Policy Elements > Addresses > List > New

3. **Configure VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey Advanced > Gateway > New

4. **Configure Asymmetric VPN**

   Network > Zones > Edit (for Trust)

   Network > Interfaces > Edit (for ethernet0/0)

5. **Configure Route**

   Network > Routing > Destination > trust-vr

6. **Configure Policy**

   Policy > Policies (From Untrust to trust) > New

CLI (for Router-2)

1. **Configure Interfaces**

```
set interface bgroup zone trust
set interface bgroup ip 10.2.2.1/24
set interface bgroup nat
set interface ethernet0/0 zone untrust
set interface ethernet0/0 ip 2.2.2.2/24
set interface tunnel.1 zone trust
set interface tunnel.1 ip unnumbered interface bgroup0
set interface tunnel.2 zone untrust
set interface tunnel.2 ip unnumbered interface bgroup0
```

2. **Configure Addresses**

```
set address untrust branch 10.1.1.0/24
set address trust smtp-1 10.2.2.10/24
set address trust http-1 10.2.2.15/32
set group address trust servers add smtp-1
set group service vpn-srv add smtp
```

3. **Configure VPN**

```
set ike gateway branch1 dynamic ssg5ssg20-e0 aggressive outgoing-interface
ethernet0/0 preshare juniper1 sec-level basic
set ike gateway branch2 dynamic ssg5ssg20-dialer aggressive outgoing-interface
 ethernet0/0 preshare juniper2 sec-level basic
set vpn vpn1 gateway branch1 sec-level basic
set vpn vpn1 gind interface tunnel.1
set vpn pvn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 smtp
set vpn vpn2 gateway branch2 sec-level basic
set vpn vpn2 bind interface tunnel.2
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 smtp
```

4. **Configure Asymmetric VPN**

```
set zone trust asymmetric-vpn
```

5. **Configure Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet 0/0 gateway 2.2.2.250
```

6. **Configure Policy**

```
set policy from untrust to trust branch servers vpn-srv permit
  save
```

Configuring a Route Monitoring Backup Interface

In this example, on the primary interface (ethernet0/0) you configure a route to the network segment 5.5.5.0/24 using gateway 10.10.10.1; and you configure dialer1 as the backup interface, with a route to the same network segment. You then configure a default route for the primary interface to the same gateway (10.10.10.1), and a default route for the dialer1 interface.

WebUI

1. **Configure Interfaces**

   Network > Routing > Destination > trust-vr > New: Enter the following, then click **Apply**:

   IP Address/Netmask: 5.5.5.0/24
   Gateway: (select)
   Interface: (select) ethernet0/0
   Gateway IP Address: 10.10.10.1

   Network > Interfaces > Backup: Enter the following, then click **Apply**:

   Primary Interface: (select) ethernet0/0
   Backup Interface: (select) dialer1
   Type: (select) route vrouter: (select) trust-vr
   IP Address/Netmask: 5.5.5.0/24

2. **Configure Route**

   Network > Routing > Source Interface (for ethernet0/0) > New: Enter the following, then click **Okay**:

   Gateway: 10.10.10.1

   Network > Routing > Destination (for trust-vr) > New: Enter the following, then click **OK**:

   Interface (select): Null

CLI

```
set vrouter trust-vr route 5.5.5.0/24 interface ethernet0/0 gateway 10.10.10.1
```

```
set interface ethernet0/0 backup interface dialer1 type route vrouter trust-vr
5.5.5.0/24
set route 0.0.0.0/0 interface ethernet0/0 gateway 10.10.10.1
set route 0.0.0.0/0 interface dialer1
save
```

Loopback Interfaces

A loopback interface is a logical interface that emulates a physical interface on the security device. However, unlike a physical interface, a loopback interface is always in the up state as long as the device on which it resides is up. Loopback interfaces are named loopback.*id_num*, where *id_num* is a number greater than or equal to 1 and denotes a unique loopback interface on the device. Like a physical interface, you must assign an IP address to a loopback interface and bind it to a security zone.

NOTE: The maximum id_num value you can specify is platform-specific.

After defining a loopback interface, you can then define other interfaces as members of its group. Traffic can reach a loopback interface if it arrives through one of the interfaces in its group. Any interface type can be a member of a loopback interface group—physical interface, subinterface, tunnel interface, redundant interface, or VSI.

After creating a loopback interface, you can use it in many of the same ways as a physical interface:

- Setting the Loopback Interface for Management on page 76
- Setting BGP on a Loopback Interface on page 76
- Setting VSIs on a Loopback Interface on page 77
- Setting the Loopback Interface as a Source Interface on page 77

NOTE: You cannot bind a loopback interface to an HA zone, nor can you configure a loopback interface for Layer 2 operation or as a redundant/aggregate interface. You cannot configure the following features on loopback interfaces: NTP, DNS, VIP, secondary IP, track IP, or WebAuth.

When a loopback interface is used as the outgoing Interface in a VPN configuration, then the loopback interface must be in the same zone as the outgoing physical interface.

You can define a MIP on a loopback interface. This allows the MIP to be accessed by a group of interfaces; this capability is unique to loopback interfaces. For information about using the loopback interface with MIPs, see “MIP and the Loopback Interface” on page 1545.

You can manage the security device using either the IP address of a loopback interface or the Manage IP address that you assign to a loopback interface.

Creating a Loopback Interface

In the following example, you create the loopback interface loopback.1, bind it to the Untrust zone, and assign the IP address 1.1.1.27/24 to it.

WebUI

Network > Interfaces > New Loopback IF: Enter the following, then click **OK**:

Interface Name: loopback.1
Zone: Untrust (select)
IP Address/Netmask: 1.1.1.27/24

CLI

```
set interface loopback.1 zone untrust
set interface loopback.1 ip 1.1.1.27
save
```

NOTE: The loopback interface is not directly accessible from networks or hosts that reside in other zones. You must define a policy to permit traffic to and from the interface.

Setting the Loopback Interface for Management

In the following example, you configure the previously defined loopback.1 interface as a management interface for the device.

WebUI

Network > Interfaces > loopback.1 > Edit: Select all the management options, then click **OK**:

CLI

```
set interface loopback.1 manage
save
```

Setting BGP on a Loopback Interface

The loopback interface can support the BGP dynamic routing protocol on the security device. In the following example, you enable BGP on the loopback.1 interface.

NOTE: To enable BGP on the loopback interface, you must first create a BGP instance for the virtual router in which you plan to bind the interface. For information about configuring BGP on Juniper Networks security devices, See "Routing" on page 1235.

WebUI

Network > Interfaces > loopback.1 > Edit: Select **Protocol BGP**, then click **OK**:

CLI

```
set interface loopback.1 protocol bgp
save
```

Setting VSIs on a Loopback Interface

You can configure Virtual Security Interfaces (VSIs) for NSRP on a loopback interface. The physical state of the VSI on the loopback interface is always up. The interface can be active or not, depending upon the state of the VSD group to which the interface belongs.

WebUI

Network > Interfaces > New VSI IF: Enter the following, then click **OK**:

```
Interface Name: VSI Base: loopback.1
VSD Group: 1
IP Address/Netmask: 1.1.1.1/24
```

CLI

```
set interface loopback.1:1 ip 1.1.1.1/24
save
```

Setting the Loopback Interface as a Source Interface

You can use a loopback interface as a source interface for certain traffic that originates from the security device. (When you define a source interface for an application, the specified source interface address is used instead of the outbound interface address to communicate with an external device.) In the following example, you specify that the security device uses the previously defined loopback.1 interface for sending syslog packets.

WebUI

Configuration > Report Settings > Syslog: Enter the following, then click **Apply**:

```
Enable Syslog Messages: (select)
Source interface: loopback.1 (select)
Syslog Servers:
```

No.: 1 (select)
IP/Hostname: 10.1.1.1
Traffic Log: (select)
Event Log: (select)

CLI

```
set syslog config 10.1.1.1 log all
set syslog src-interface loopback.1
set syslog enable
save
```

Interface State Changes

An interface can be in one of the states described in Table 5 on page 78.

Table 5: Interface States

State	Description
Physically Up	For physical Ethernet interfaces operating at either Layer 2 (transparent mode) or Layer 3 (route mode) in the OSI Model. An interface is physically up when it is cabled to another network device and can establish a link to that device.
Logically Up	For both physical interfaces and logical interfaces (subinterfaces, redundant interfaces, and aggregate interfaces). An interface is logically up when traffic passing through that interface is able to reach specified devices (at tracked IP addresses) on a network.
Physically Down	An interface is physically down when it is not cabled to another network device or when it is cabled but cannot establish a link. You can also force an interface to be physically down with the following CLI command: **set interface interface phy link-down**.
Logically Down	An interface is logically down when traffic passing through that interface cannot reach specified devices (at tracked IP addresses) on a network.

The physical state of an interface takes precedence over its logical state. An interface can be physically up and—at the same time—be either logically up or logically down. If an interface is physically down, its logical state becomes irrelevant.

When the state of an interface is up, all routes that make use of that interface remain active and usable. When the state of an interface is down, the security device deactivates all routes using that interface—although, depending on whether the interface is physically or logically down, traffic might still flow through an interface whose state is down (see “Down Interfaces and Traffic Flow” on page 92). To compensate for the loss of routes caused by the loss of an interface, you can configure alternate routes using an alternate interface.

Depending on how you set up the action that an observed interface state change can cause, a state change from up to down in a monitored interface can cause the

monitoring interface to change its state from down to up. To configure this behavior, you can use the following CLI command:

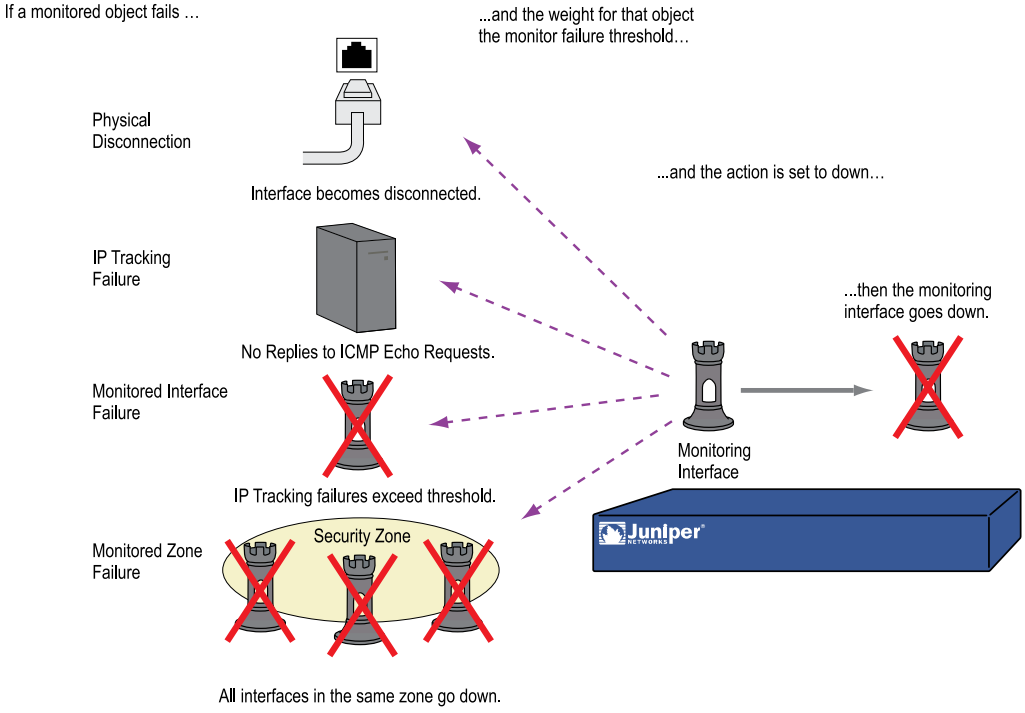
```
set interface interface monitor threshold number action up { logically | physically }
```

When you enter the above command, the security device automatically forces the monitoring interface into a down state. If the monitored object (tracked IP address, interface, zone) fails, then the state of the monitoring interface becomes up—either logically or physically, per your configuration.

An interface can monitor objects for one or more of the following events. See Figure 22 on page 79. Each of these events by itself or in combination can cause the state of the monitoring interface to change from up to down and from down to up:

- Physical disconnection/reconnection
- IP tracking failure/success
- Failure/success of a monitored interface
- Failure/success of a monitored security zone

Figure 22: Interface State Monitoring



If, after failing, a monitored object succeeds (the interface is reconnected or IP tracking again succeeds), then the monitoring interface comes back up. There is approximately a one-second delay between the monitored object succeeding and the monitoring interface reactivating.

Each of the above events is presented in the following sections:

Physical Connection Monitoring

All physical interfaces on a security device monitor the state of their physical connection to other network devices. When an interface is connected to and has established a link with another network device, its state is physically up, and all routes that use that interface are active.

You can see the state of an interface in the State column in the output of the **get interface** command and in the Link column on Network > Interfaces in the WebUI. The state can be up or down.

You can see the state of a route in the Status field of the **get route id** number *command* and on Network > Routing > Destination in the WebUI. If there is an asterisk, the route is active. If there is no asterisk, it is inactive.

An interface changes its state whenever you restart the security device. In previous ScreenOS releases, such tracked information about state changes would be lost when the information was overwritten in the event log. In order to retain information about an interface's state changes, the current ScreenOS release supports the use of a counter that records the number of times an interface changes its state.

The counter value increments whenever the admin manually changes the interface hardware status or when the security device restarts. The admin can reset the interface counter to zero whenever it exceeds its allotted size.

To reset the interface counter:

```
clear interface interface status-change
save
```

The output of the **get interface** command displays the status change count and the last time the interface state changed.

NOTE: Only physical, WAN, aggregate, and bgroup interfaces support the use of the counter for tracking interface state changes.

To view the type of gigabit interface (SX or LX), use the get interface *interface-name* command:

```
Interface ethernet3/1:
  description ethernet3/1
  number 7, if_info 458808, if_index 0
  link down, phy-link down/auto
  type LX,
  status change:0
  vsys Root, zone Null, vr untrust-vr
  *ip 0.0.0.0/0   mac 0010.dbc8.0d07
  pmtu-v4 disabled
  ping disabled, telnet disabled, SSH disabled, SNMP disabled
  web disabled, ident-reset disabled, SSL disabled
```

```
NHRP disabled
bandwidth: physical 0Mbps, configured 0Mbps
```

NOTE: The 10 gigabit interface type will not be displayed.

Tracking IP Addresses

The security device can track specified IP addresses through an interface so that when one or more of them become unreachable, the security device can deactivate all routes associated with that interface even if the physical link is still active. A deactivated route becomes active again after the security device regains contact with those IP addresses.

NOTE: For some ScreenOS devices, this action also causes a failover to the backup interface that is bound to the same zone as the interface on which IP tracking is configured (see "Interface Failover with IP Tracking" on page 1825).

ScreenOS uses Layer 3 path monitoring, or IP tracking, similar to that used for NSRP, to monitor the reachability of specified IP addresses through an interface. For example, if an interface connects directly to a router, you can track the next-hop address on the interface to determine if the router is still reachable. When you configure IP tracking on an interface, the security device sends ping requests on the interface to up to four target IP addresses at user-defined intervals. The security device monitors these targets to determine if it receives a response. If there is no response from a target for a specified number of times, that IP address is deemed to be unreachable. Failure to elicit a response from one or more targets can cause the security device to deactivate routes associated with that interface. If another route to the same destination is available, the security device then redirects traffic to use the new route.

You can define IP tracking on the following interfaces for which you have configured a Manage IP address:

- Physical interface bound to a security zone (not the HA or MGT function zones)

NOTE: The interface can operate at Layer 2 (transparent mode) or Layer 3 (route mode).

- Subinterface
- Redundant interface
- Aggregate interface

NOTE: Although the interface can be redundant or an aggregate, it cannot be a member of a redundant or aggregate interface.

On devices that support virtual systems (vsys), the interface on which you set IP tracking can belong to the root system or to a vsys. However, to set IP tracking on a shared interface, you can only set it at the root level.

NOTE: From a vsys, you can set interface monitoring to monitor a shared interface from an interface that belongs to the vsys. However, from within a vsys, you cannot set interface monitoring from a shared interface. For more information, see "Interface Monitoring" on page 85.

For each interface, you can configure up to four IP addresses for the security device to track. On a single device, you can configure up to 64 track IP addresses. That total includes all track IP addresses whether they are for interface-based IP tracking, for NSRP-based IP tracking, at the root level, or at the vsys level.

The tracked IP addresses do not have to be in the same subnetwork as the interface. For each IP address to be tracked, you can specify the following:

- Interval, in seconds, at which the pings are sent to the specified IP address.
- Number of consecutive unsuccessful ping attempts before the connection to the IP address is considered failed.
- Weight of the failed IP connection (once the sum of the weights of all failed IP connections crosses a specified threshold, routes that are associated with the interface are deactivated).
- Time-out, in seconds, which allow the users to specify the expiration time of the ping or ARP requests sent to the specified IP address. The ping or ARP request will be counted as a failure if the response time of the request exceeds the specified time-out value.

You can also configure the security device to track the default gateway for an interface that is a PPPoE or DHCP client. To do that, use the dynamic option: (CLI) **set interface** interface **monitor dynamic** or (WebUI) Network > Interfaces > Edit (for the DHCP or PPPoE client interface) > Monitor > Track IP > Add: Select **Dynamic**.

NOTE: When you configure an IP address for the security device to track, the security device does not add a host route for that IP address to the routing table.

There are two types of thresholds in configuring tracking IP addresses:

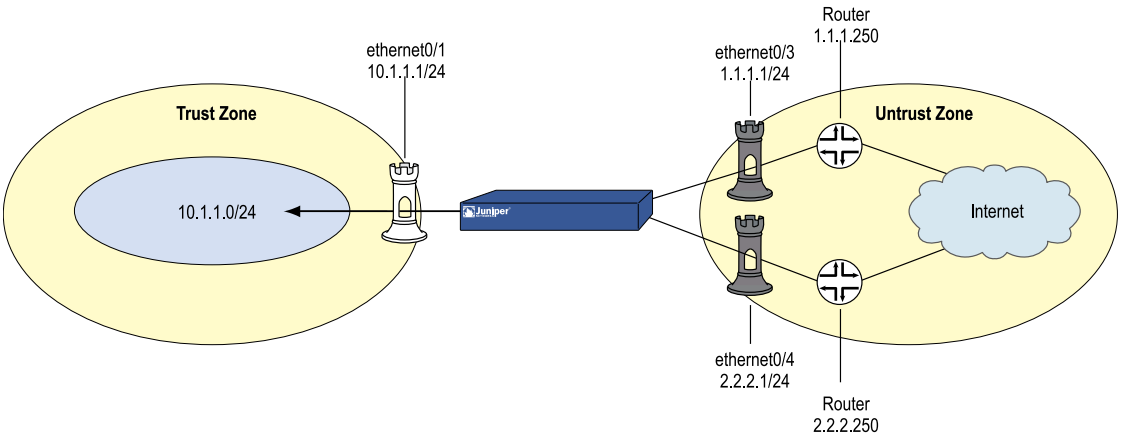
- Failure threshold for a specific tracked IP address—The number of consecutive failures to elicit a ping response from a specific IP address that constitutes a failure in reaching the IP address. Not exceeding the threshold indicates an acceptable level of connectivity with the address; exceeding the threshold indicates an unacceptable level. You set this threshold for each IP address at any value between 1 and 200. The default is 3.
- Failure threshold for IP tracking on the interface—The total weight of the cumulative failed attempts to reach IP addresses on the interface that causes routes associated with the interface to be deactivated. You can set this threshold at any value between 1 and 255. The default is 1, which means a failure to reach

any configured tracked IP address causes routes associated with the interface to be deactivated.

By applying a weight, or a value, to a tracked IP address, you can adjust the importance of connectivity to that address in relation to reaching other tracked addresses. You can assign comparatively greater weights to relatively more important addresses and less weight to relatively less important addresses. Note that the assigned weights only come into play when the failure threshold for a specific tracked IP address is reached. For example, if the failure threshold for IP tracking on an interface is 3, failure of a single tracked IP address with a weight of 3 meets the failure threshold for IP tracking on the interface, which causes routes associated with the interface to be deactivated. The failure of a single tracked IP address with a weight of 1 would not meet the failure threshold for IP tracking on the interface, and routes associated with the interface would remain active.

In the following example, the interface ethernet0/1 is bound to the Trust zone and assigned the network address 10.1.1.1/24. The interfaces ethernet0/3 and ethernet0/4 are bound to the Untrust zone. The ethernet0/3 interface is assigned the network address 1.1.1.1/24 and is connected to the router at 1.1.1.250. The ethernet0/4 interface is assigned the network address 2.2.2.1/24 and is connected to the router at 2.2.2.250. See Figure 23 on page 83.

Figure 23: Interface IP Tracking



There are two default routes configured: one uses ethernet0/3 as the outbound interface with the router address 1.1.1.250 as the gateway; the other uses ethernet0/4 as the outbound interface with the router address 2.2.2.250 as the gateway and is configured with a metric value of 10. The default route that uses ethernet0/3 is the preferred route since it has a lower metric (the default metric value for static routes is 1). The following output from the **get route** command shows four active routes for the trust-vr (active routes are denoted with an asterisk). The default route through ethernet0/3 is active, while the default route through ethernet0/4 is not active because it is less preferred.

Figure 24: Get Route Output

```
device-> get route
untrust-vr (0 entries)
------------------------------------------------------------------------
C - Connected, S - Static, A - Auto-Exported, I - Imported, R - RIP
iB - IBGP, eB - EBGP, O - OSPF, E1 - OSPF external type 1
E2 - OSPF external type 2
trust-vr (4 entries)
------------------------------------------------------------------------
   ID   IP-Prefix    InterfaceGateway     P   Pref  Mtr  Vsys
------------------------------------------------------------------------
*  4    0.0.0.0/0    eth0/3    1.1.1.250  S   20    1    Root
*  2    1.1.1.0/24   eth0/3    0.0.0.0    C   0     0    Root
   3    0.0.0.0/0    eth0/4    2.2.2.250  S   20    10   Root
*  6    2.2.2.0/24   eth0/4    0.0.0.0    C   0     1    Root
*  5    10.1.1.0/24  eth0/1    0.0.0.0    C   20    1    Root
```

If the route through ethernet0/3 becomes unavailable, the default route through ethernet0/4 becomes active. You enable and configure IP tracking on the ethernet0/3 interface to monitor the router address 1.1.1.250. If IP tracking fails to reach 1.1.1.250, all routes associated with the ethernet0/3 interface become inactive on the security device. As a result, the default route through ethernet0/4 becomes active. When IP tracking is again able to reach 1.1.1.250, the default route through ethernet0/3 becomes active and, at the same time, the default route through ethernet0/4 becomes inactive, because it is less preferred than the default route through ethernet0/3.

The following example enables IP tracking with an interface failure threshold of 5 and configures IP tracking on the ethernet0/3 interface to monitor the router IP address 1.1.1.250, which is assigned a weight of 10. The interval is set to 20 (the default is 1), and the time-out option is set to 10 for the tracked IP address.

WebUI

Network > Interfaces > Edit (for ethernet0/3) > Monitor: Enter the following, then click **Apply**:

Enable Track IP: (select)
Threshold: 5
> Monitor Track IP ADD: Enter the following, then click **Add**:
Static: (select)
  Track IP: 1.1.1.250
  Weight: 10
   Interval: 20
  Time Out: 10

CLI

```
set interface ethernet0/3 monitor track-ip ip 1.1.1.250 weight 10
set interface ethernet0/3 monitor track-ip threshold 5
set interface ethernet0/3 monitor track-ip
set interface ethernet0/3 monitor track-ip ip 1.1.1.250 interval 20
set interface ethernet0/3 monitor track-ip ip 1.1.1.250  time-out 10
save
```

In the example, the failure threshold for the target address is set to the default value of 3. That is, if the target does not return a response to three consecutive pings, a weight of 10 is applied toward the failure threshold for IP tracking on the interface. Because the failure threshold for IP tracking on the interface is 5, a weight of 10 causes routes associated with the interface to be deactivated on the security device.The ping request will be counted as failure if the response time of the request exceeds the **time-out** value of 10 seconds.

You can verify the status of the IP tracking on the interface by issuing the CLI command **get interface ethernet0/3 monitor track-ip**, as shown in Figure 25 on page 85.

Figure 25: Get Interface Output

```
device-> get interface eth0/3 monitor tr
ip address                          intval threshold wei tmout gateway        fail-count success
1.1.1.250                               1         5  10     1 0.0.0.0                6 84%
failure weight: 10, threshold: 5, failed: 1 ip(s) failed, weighted sum = 10
```

The **get route** command shows that the default route through ethernet0/4 is now active, while all routes through ethernet0/3 are no longer active.

Figure 26: Get Route Output With Activated Interfaces

```
device-> get route
untrust-vr (0 entries)
-----------------------------------------------------------------------
C - Connected, S - Static, A - Auto-Exported, I - Imported, R - RIP
iB - IBGP, eB - EBGP, O - OSPF, E1 - OSPF external type 1
E2 - OSPF external type 2
trust-vr (4 entries)
-----------------------------------------------------+-----------------
   ID   IP-Prefix    Interface Gateway    P   Pref  Mtr   Vsys
-----------------------------------------------------------------------
    4   0.0.0.0/0    eth0/3    1.1.1.250  S   20    1     Root
    2   1.1.1.0/24   eth0/3    0.0.0.0    C   0     0     Root
*   3   0.0.0.0/0    eth0/4    2.2.2.250  S   20    10    Root
*   6   2.2.2.0/24   eth0/4    0.0.0.0    C   0     1     Root
*   5   10.1.1.0/24  eth0/1    0.0.0.0    C   20    1     Root
```

Note that even though the routes through ethernet0/3 are no longer active, IP tracking uses the routes associated with ethernet0/3 to continue sending ping requests to the target IP address. When IP tracking is again able to reach 1.1.1.250, the default route through ethernet0/3 again becomes active on the security device. At the same time, the default route through ethernet0/4 becomes inactive, since it is less preferred than the default route through ethernet0/3.

Interface Monitoring

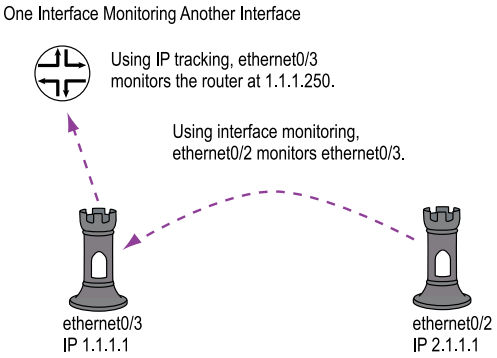
A security device can monitor the physical and logical state of interfaces and then take action based on observed changes, see Figure 27 on page 86. For example, if

the state of a monitored interface changes from up to down, the following can occur, as shown in Table 6 on page 86.

Table 6: Monitored Interface

If:	Then:
The physical state of an interface changes from up to down	The state change might trigger another interface that is monitoring the one that just went down to also go down. You can specify whether you want the second interface to be physically or logically down. The state change of either interface going physically down, or the combined weight of both going physically down together, might trigger an NSRP failover. An NSRP device or a VSD group failover can only occur as a result of a change to the physical state of an interface.
The logical state of an interface changes from up to down as the result of an IP tracking failure	The state change might trigger another interface that is monitoring the one that just went down to also go down. Although the first interface is down logically, you can specify whether you want the down state of the second interface to be logical or physical.

Figure 27: Ethernet0/3 and Ethernet0/2 Interface Monitoring



To set interface monitoring, do either of the following:

WebUI

Network > Interfaces > Edit (for the interface you want to do the monitoring) > Monitor > Edit Interface: Enter the following, then click **Apply**:

Interface Name: Select the interface that you want to be monitored.
Weight: Enter a weight between 1 and 255.

CLI

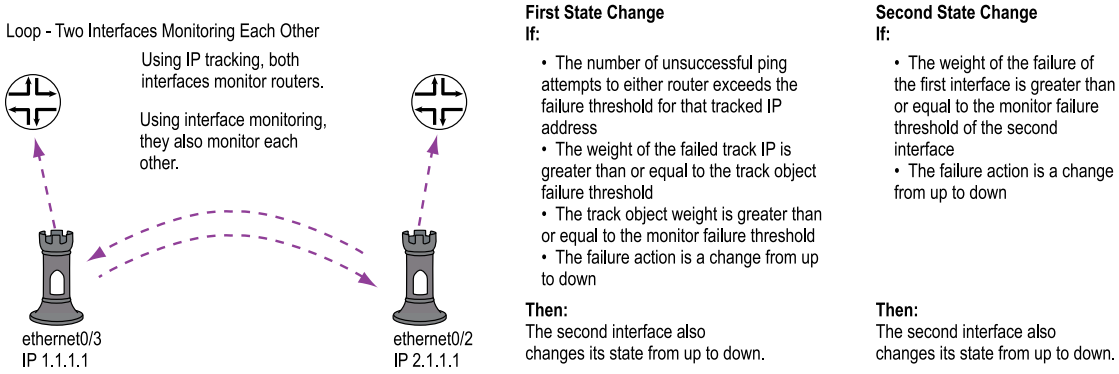
```
set interface interface1 monitor interface interface2 [ weight number ]
save
```

If you do not set a weight, the security device applies the default value, 255.

If two interfaces monitor each other, they form a loop. In that case, if either interface changes state, the other interface in the loop also changes state. See Figure 28 on page 87.

NOTE: An interface can only be in one loop at a time. We do not support configurations in which one interface belongs to multiple loops.

Figure 28: Loop Monitoring

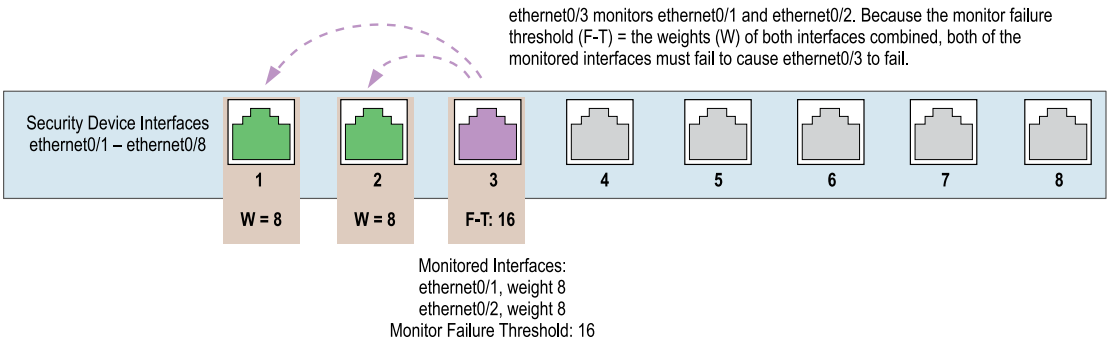


Monitoring Two Interfaces

In this example, you configure ethernet0/3 to monitor two interfaces—ethernet0/1 and ethernet0/2. Because the weight for each monitored interface (8 + 8) equals the monitor failure threshold (16), both ethernet0/1 and ethernet0/2 must fail (and change their state from up to down) concurrently to cause ethernet0/3 to fail (and change its state from up to down). See Figure 29 on page 88.

NOTE: This example omits the configuration of IP tracking on the ethernet0/1 and ethernet0/2 interfaces (see "Tracking IP Addresses" on page 81). Without IP tracking, the only way that ethernet0/1 and ethernet0/2 might fail is if they become physically disconnected from other network devices or if they cannot maintain links with those devices.
If you set the monitor failure threshold to 8—or leave it at 16 and set the weight of each monitored interface to 16—the failure of either ethernet0/1 or ethernet0/2 can cause ethernet0/3 to fail.

Figure 29: Two-Loop Interface Monitoring



WebUI

Network > Interfaces > Edit (for ethernet0/3) > Monitor > Edit Interface: Enter the following, then click **Apply**:

ethernet0/1: (select); Weight: 8
ethernet0/2: (select); Weight: 8

Network > Interfaces > Edit (for ethernet0/3) > Monitor: Enter **16** in the Monitor Threshold field, then click **Apply**.

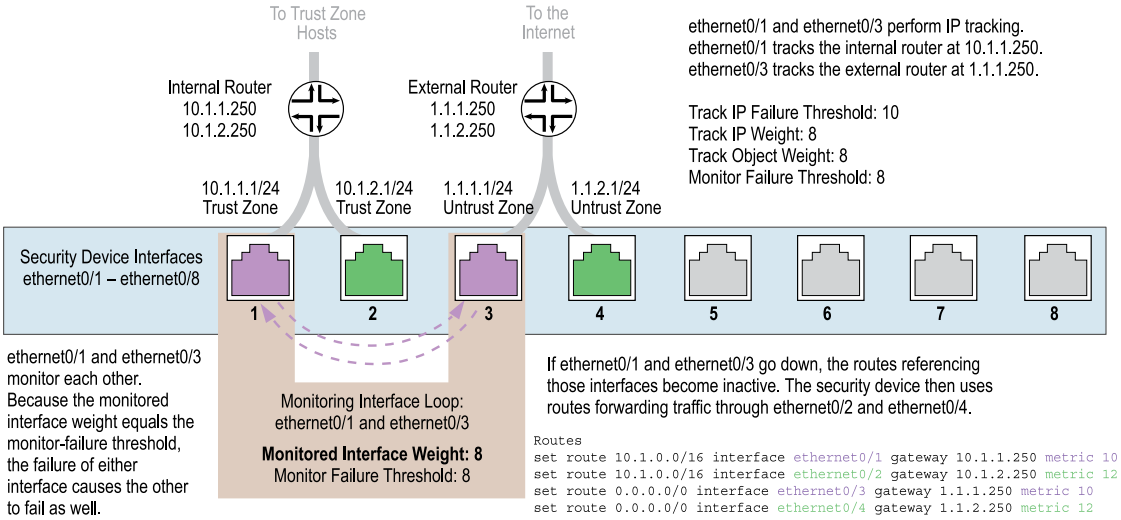
CLI

```
set interface ethernet0/3 monitor interface ethernet0/1 weight 8
set interface ethernet0/3 monitor interface ethernet0/2 weight 8
set interface ethernet0/3 monitor threshold 16
save
```

Monitoring an Interface Loop

In this example, you first configure IP tracking for two interfaces—ethernet0/1 and ethernet0/3. Then you configure these interfaces to monitor each other so that if one changes its state, the other does likewise. Finally, you define two sets of routes. The first set forwards traffic through ethernet0/1 and ethernet0/3. The second set has the same destination addresses, but these routes have lower ranked metrics and use different egress interfaces (ethernet0/2 and ethernet0/4) and different gateways from the first set. With this configuration, if the first set of interfaces fails, the security device can reroute all traffic through the second set. All zones are in the trust-vr routing domain.

Figure 30: Four-Interface Loop Monitoring



WebUI

1. **IP Tracking**

   Network > Interfaces > Edit (for ethernet0/1) > Monitor: Enter the following, then click **Apply**:

   Enable Track IP: (select)
   Monitor Threshold: 8
   Track IP Option: Threshold: 8
   Weight: 8

NOTE: To control whether the state of an interface becomes logically or physically down (or up), you must use the CLI command **set interface** interface **monitor threshold** number **action { down | up } { logically | physically }**. Only physical interfaces bound to any security zone other than the Null zone can be physically up or down.

> Monitor Track IP ADD: Enter the following, then click **Add**:

Static: (select)
Track IP: 10.1.1.250
Weight: 8
Interval: 3 Seconds
Threshold: 10

Network > Interfaces > Edit (for ethernet0/3) > Monitor: Enter the following, then click **Apply**:

Enable Track IP: (select)
Monitor Threshold: 8
Track IP Option: Threshold: 8

Weight: 8

> Monitor Track IP ADD: Enter the following, then click **Add**:

Static: (select)
Track IP: 1.1.1.250
Weight: 8
Interval: 3 Seconds
Threshold: 10

2. **Interface Monitoring**

Network > Interfaces > Edit (for ethernet0/1) > Monitor > Edit Interface: Enter the following, then click **Apply**:

ethernet0/3: (select); Weight: 8

Network > Interfaces > Edit (for ethernet0/3) > Monitor > Edit Interface: Enter the following, then click **Apply**:

ethernet0/1: (select); Weight: 8

3. **Routes**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.0.0/16
Gateway: (select)
    Interface: ethernet0/1
    Gateway IP Address: 10.1.1.250
    Metric: 10

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.0.0/16
Gateway: (select)
    Interface: ethernet0/2
    Gateway IP Address: 10.1.2.250
    Metric: 12

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
    Interface: ethernet0/3
    Gateway IP Address: 1.1.1.250
    Metric: 10

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)

Interface: ethernet0/4
Gateway IP Address: 1.1.2.250
Metric: 12

CLI

1. **IP Tracking**

```
set interface ethernet0/1 track-ip ip 10.1.1.250 weight 8
set interface ethernet0/1 track-ip threshold 8
set interface ethernet0/1 track-ip weight 8
set interface ethernet0/1 track-ip
set interface ethernet0/3 track-ip ip 1.1.1.250 weight 8
set interface ethernet0/3 track-ip threshold 8
set interface ethernet0/3 track-ip weight 8
set interface ethernet0/3 track-ip
```

2. **Interface Monitoring**

```
set interface ethernet0/1 monitor interface ethernet0/3 weight 8
set interface ethernet0/1 monitor threshold 8 action down physically
set interface ethernet0/3 monitor interface ethernet0/1 weight 8
set interface ethernet0/3 monitor threshold 8 action down physically
```

3. **Routes**

```
set vrouter trust-vr route 10.1.0.0/16 interface ethernet0/1 gateway 10.1.1.250
 metric 10
set vrouter trust-vr route 10.1.0.0/16 interface ethernet0/2 gateway 10.1.2.250
 metric 12
set vrouter trust-vr route 0.0.0.0/0 interface ethernet0/3 gateway 1.1.1.250
metric 10
set vrouter trust-vr route 0.0.0.0/0 interface ethernet0/4 gateway 1.1.2.250
metric 12
save
```

Security Zone Monitoring

In addition to monitoring individual interfaces, an interface can monitor all the interfaces in a security zone—any security zone other than its own. For an entire security zone to fail, every interface bound to that zone must fail. As long as one interface bound to a monitored zone is up, the security device considers the entire zone to be up.

To configure an interface to monitor a security zone, do either of the following:

WebUI

Network > Interfaces > Edit (for the interface you want to do the monitoring) > Monitor > Edit Zone: Enter the following, then click **Apply**:

Zone Name: Select the zone that you want to be monitored.

Weight: Enter a weight between 1 and 255.

CLI

```
set interface interface monitor zone zone [ weight number ]
```

If you do not set a weight, the security device applies the default value, 255.

Down Interfaces and Traffic Flow

Configuring IP tracking on an interface allows the security device to reroute outgoing traffic through a different interface if certain IP addresses become unreachable through the first interface. However, while the security device might deactivate routes associated with an interface because of an IP tracking failure, the interface can remain physically active and still send and receive traffic. For example, the security device continues to process incoming traffic for an existing session that might arrive on the original interface on which IP tracking failed. Also, the security device continues to use the interface to send ping requests to target IP addresses to determine if the targets again become reachable. In these situations, traffic still passes through an interface on which IP tracking has failed and for which the security device has deactivated routes.

How the security device handles session traffic on such an interface depends upon the following:

- If the interface on which you configure IP tracking functions as an egress interface for a session, session replies might continue to arrive at the interface and the security device still processes them.
- If the interface on which you configure IP tracking functions as an ingress interface for a session, applying the **set arp always-on-dest** command causes the security device to reroute session replies to another interface. If you do not set this command, the security device forwards session replies through the interface on which IP tracking failed even though the security device has deactivated routes using that interface. (By default, this command is unset.)

  By default, a security device caches a session initiator's MAC address when it receives the initial packet for a new session. If you enter the CLI command **set arp always-on-dest**, the security device does not cache a session initiator's MAC address. Instead, the security device performs an ARP lookup when processing the reply to that initial packet. If the initiator's MAC address is in the ARP table, the security device uses that. If the MAC address is not in the ARP table, the security device sends an ARP request for the destination MAC address and then adds the MAC address it receives to its ARP table. The security device performs another ARP lookup whenever a route change occurs.

"Failure on the Egress Interface" on page 93 describes separate scenarios in which IP tracking fails on the egress interface and on the ingress interface; and, in the case of the latter, what occurs when you use the command **set arp always-on-dest**.

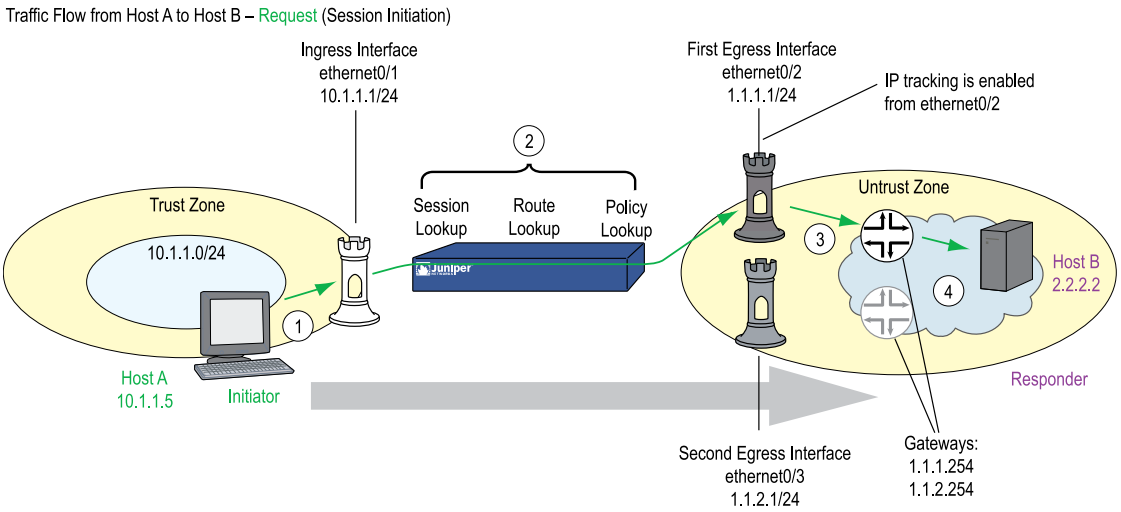
NOTE: "Failure on the Egress Interface" on page 93 describes how IP tracking triggers routing changes and how those changes can affect the packet flow through all Juniper Networks security devices.

Failure on the Egress Interface

In the following scenario, you configure IP tracking on ethernet0/2, which is the egress interface for sessions from Host A to Host B. Host A initiates the session by sending a packet to Host B, as shown in Figure 31 on page 93.

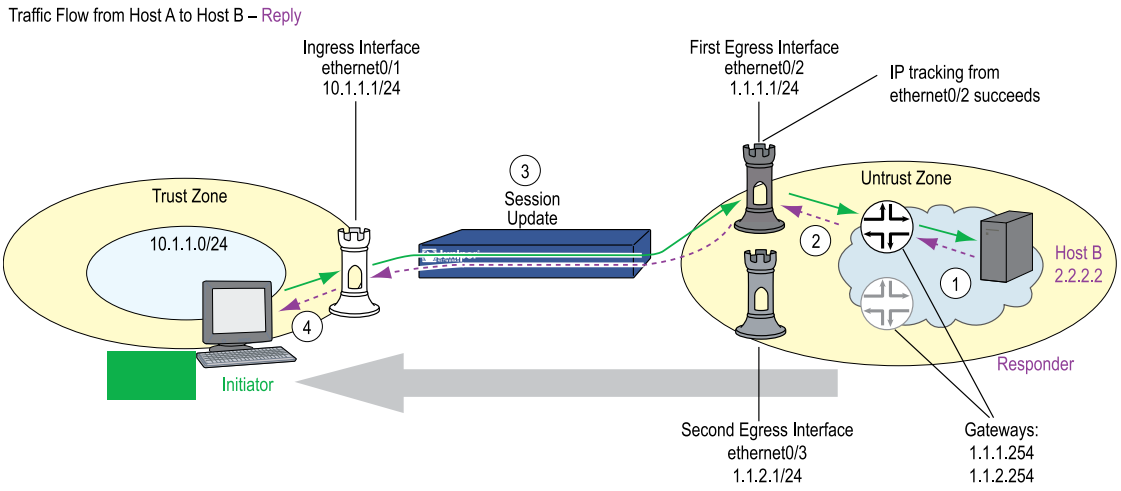
NOTE: You must first create two routes to Host B, and both the egress interfaces must be in the same zone so that the same policy applies to traffic before and after the rerouting occurs.

Figure 31: Host A and Host B IP Tracking



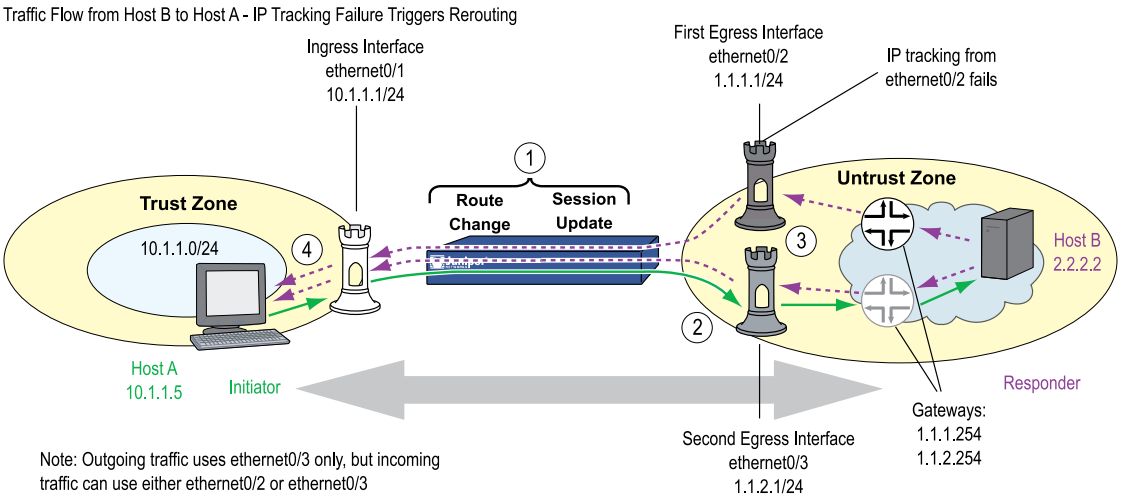
When Host B replies to Host A, the return traffic follows a similar path back through the security device, as shown in Figure 32 on page 94.

Figure 32: Host B to Host A Egress Traffic Flow



If IP tracking on ethernet0/2 fails, the security device deactivates routes that use ethernet0/2 and uses ethernet0/3 for outbound traffic to Host B. However, replies from Host B to Host A can arrive through either ethernet0/2 or ethernet0/3 and the security device forwards them through ethernet0/1 to Host A. See Figure 33 on page 94.

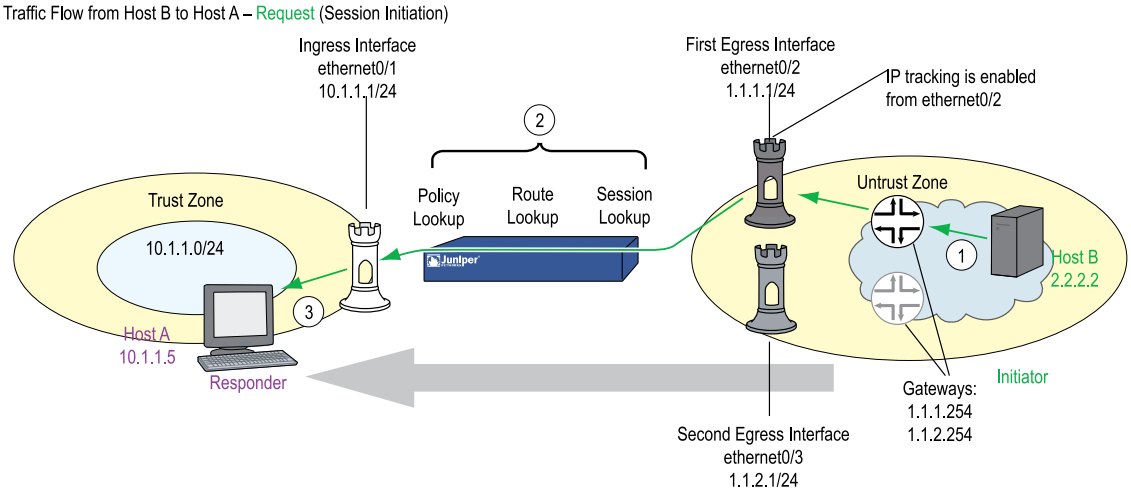
Figure 33: Egress IP Tracking Failure



Failure on the Ingress Interface

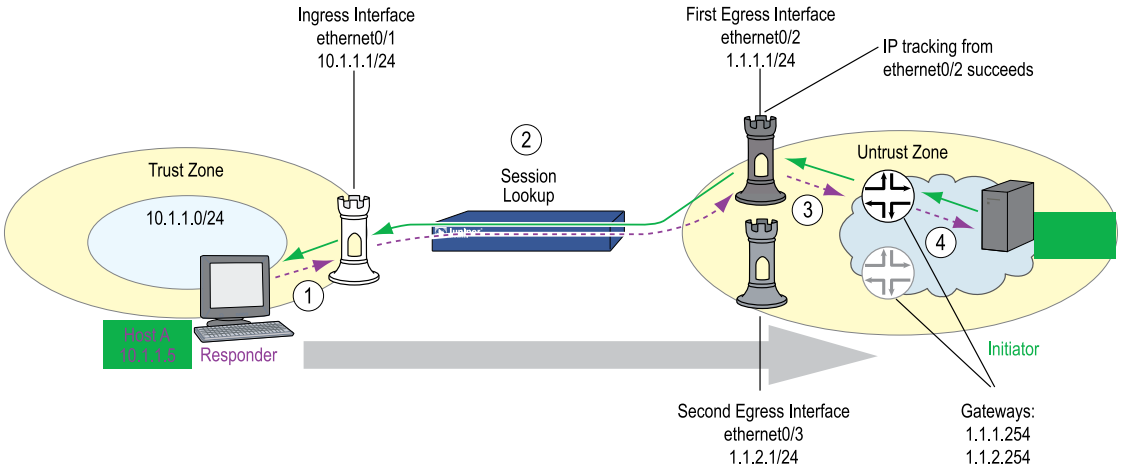
In the following scenario, you again configure IP tracking on ethernet0/2, but this time ethernet0/2 is the ingress interface on the security device for sessions from Host B to Host A. Host B initiates the session by sending a packet to Host A, as shown in Figure 34 on page 95.

Figure 34: Host B to Host A Ingress Traffic Flow



When Host A replies to Host B, the return traffic follows a similar path back through the security device, as shown in Figure 35 on page 95.

Figure 35: Ingress Host A to Host B Traffic Flow



1. Host A at 10.1.1.5 sends a reply packet destined for Host B (2.2.2.2) to ethernet0/1 at 10.1.1.1.
2. The security device performs a session lookup. Because this is a reply, the device matches it with an existing session and refreshes the session table entry.
3. By using the cached MAC address for the gateway at 1.1.1.254, or by doing an ARP lookup to discover its MAC address, the device forwards the packet through ethernet0/2 to the gateway.
4. When the gateway at 1.1.1.254 receives the reply, it forwards it to its next hop. Routing continues until Host B receives it.

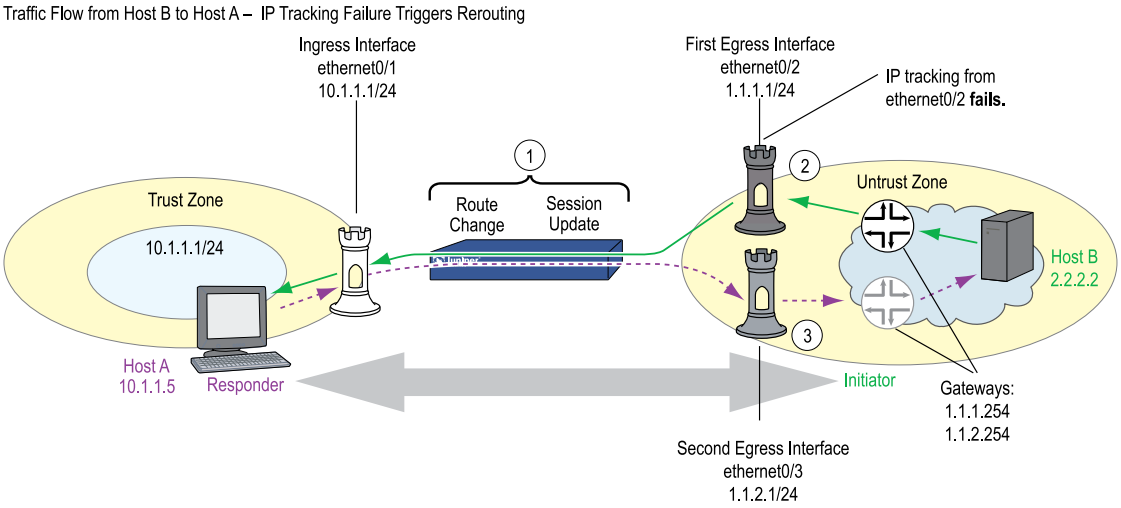
If IP tracking on ethernet0/2 fails, the security device deactivates routes that use ethernet0/2 and uses ethernet0/3 for outbound traffic to Host B. However, requests from Host B to Host A can still arrive through ethernet0/2 and the security device still forwards them to Host A through ethernet0/1. The data flow for requests from Host B to Host A looks the same after an IP tracking failure as it did before. However,

the replies from Host A can take one of two different paths, depending on the application of the **set arp always-on-dest** command.

If you set the command **set arp always-on-dest**, the security device sends an ARP request for the destination MAC address when processing the reply to the first packet in a session or when a route change occurs. (When this command is unset, the security device caches the session initiator's MAC address and uses that when processing replies. By default, this command is unset).

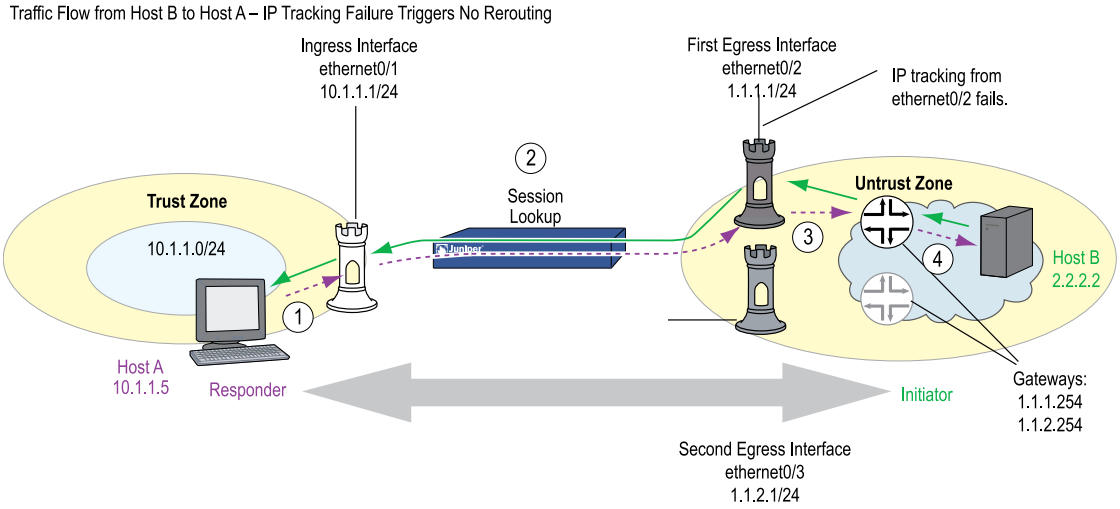
When IP tracking on ethernet0/2 fails, the security device first deactivates all routes using ethernet0/2 and then does a route lookup. It finds another route to reach Host B through ethernet0/3 and the gateway at 1.1.2.254. It then scans its session table and redirects all sessions to the new route. If you have the **set arp always-on-dest** command enabled, the security device does an ARP lookup when it receives the next packet from Host A because it is in a session affected by the route change. Despite the ingress interface on which packets from Host B arrive, the security device sends all further replies from Host A through ethernet0/3 to the gateway at 1.1.2.254. See Figure 36 on page 96.

Figure 36: Ingress IP Tracking Failure with Traffic Rerouting



If you have set the command **unset arp always-on-dest** (which is the default configuration), the security device uses the MAC address for the gateway at 1.1.1.1 that it cached when Host B sent the initial session packet. The security device continues to send session replies through ethernet0/2. In this case, the IP tracking failure caused no change in the flow of data through the security device.

Figure 37: Ingress IP Tracking Failure with No Rerouting



Chapter 5

Interface Modes

Interfaces can operate in three different modes: Network Address Translation (NAT), route, and transparent. If an interface bound to a Layer 3 zone has an IP address, you can define the operational mode for that interface as either NAT or route. An interface bound to a Layer 2 zone (such as the predefined v1-trust, v1-untrust, and v1-dmz zones, or a user-defined Layer 2 zone) must be in transparent mode. You select an operational mode when you configure an interface.

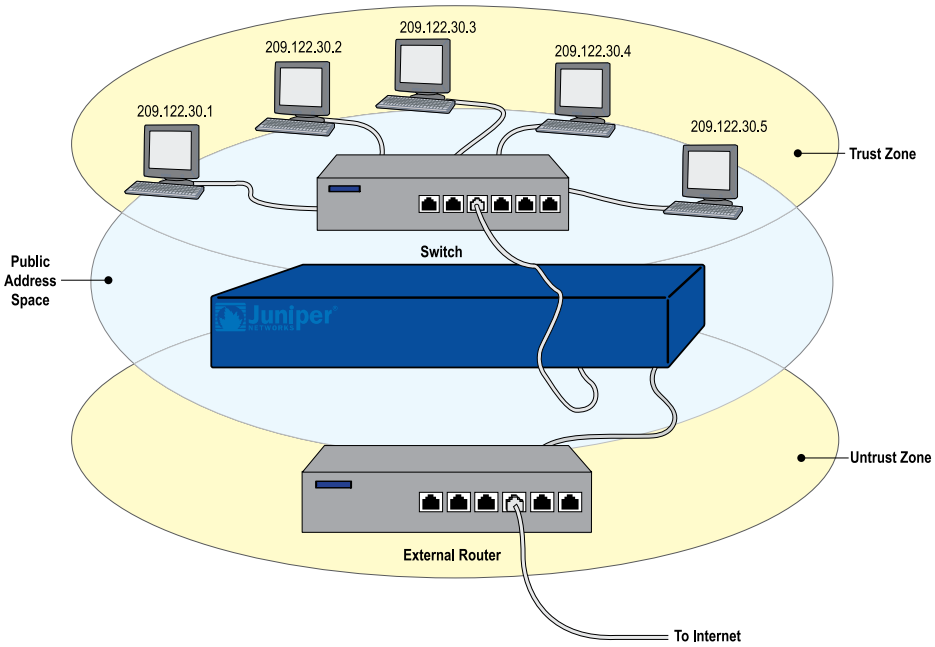
NOTE: Although you can define the operational mode for an interface bound to any Layer 3 zone as NAT, the security device only performs NAT on traffic passing through that interface en-route to the Untrust zone. ScreenOS does not perform NAT on traffic destined for any zone other than the Untrust zone. Also, note that ScreenOS allows you to set an Untrust zone interface in NAT mode, but doing so activates no NAT operations.

This chapter contains the following sections:

Transparent Mode

When an interface is in transparent mode, the security device filters packets traversing the firewall without modifying any of the source or destination information in the IP packet header. All interfaces behave as though they are part of the same network, with the security device acting much like a Layer 2 switch or bridge. In transparent mode, the IP addresses of interfaces are set at 0.0.0.0, making the presence of the security device transparent (invisible) to users. See Figure 38 on page 100.

Figure 38: Transparent Mode


209.122.30.3
209.122.30.2
209.122.30.4
209.122.30.1
209.122.30.5
Trust Zone
Switch
Public
Address
Space
Untrust Zone
External Router
To Internet

Transparent mode is a convenient means for protecting Web servers or any other kind of servers that mainly receive traffic from untrusted sources. Using transparent mode offers the following benefits:

- No need to reconfigure the IP settings of routers or protected servers
- No need to create mapped IP (MIP) or virtual IP (VIP) addresses for incoming traffic to reach protected servers

NOTE: Transparent mode is supported for both IPv4 and IPv6 traffic.

Zone Settings

By default, ScreenOS creates one function zone, the VLAN zone, and three L2 security zones: V1-Trust, V1-Untrust, and V1-DMZ.

VLAN Zone

The VLAN zone hosts the VLAN1 interface, which has the same configuration and management abilities as a physical interface. When the security device is in transparent mode, you use the VLAN1 interface for managing the device and terminating VPN traffic. You can configure the VLAN1 interface to permit hosts in the L2 security zones to manage the device. To do that, you must set the VLAN1 interface IP address in the same subnet as the hosts in the L2 security zones.

For management traffic, the VLAN1 Manage IP takes precedence over the VLAN1 interface IP. You can set the VLAN1 Manage IP for management traffic and dedicate the VLAN1 interface IP solely for VPN tunnel termination.

Predefined Layer 2 Zones

ScreenOS provides three L2 security zones by default: V1-Trust, V1-Untrust, and V1-DMZ. These three zones share the same L2 domain. When you configure an interface in one of the zones, it gets added to the L2 domain shared by all interfaces in all the L2 zones. All hosts in the L2 zones must be on the same subnet to communicate.

When the device is in transparent mode, you use the VLAN1 interface to manage the device. For management traffic to reach the VLAN1 interface, you must enable the management options on the VLAN1 interface and on the zone(s) through which the management traffic passes. By default, all management options are enabled in the V1-Trust zone. To enable hosts in other zones to manage the device, you must set those options on the zones to which they belong.

NOTE: To see which physical interfaces are prebound to the L2 zones for each Juniper Networks security device, see the hardware guide for that device.

Traffic Forwarding

A security device operating at Layer 2 does not permit any inter-zone traffic unless there is a policy configured on the device. For more information about setting policies, see "Policies" on page 197. After you configure a policy on the security device, it does the following:

- Allows or denies the traffic specified in the policy.
- Allows ARP and L2 non-IP multicast and broadcast traffic. The security device can then receive and pass L2 broadcast traffic for the Spanning Tree Protocol (STP).
- Continues to block all non-IP and non-ARP unicast and IPsec traffic.

You can change the forwarding behavior of the device as follows:

- To block all L2 non-IP and non-ARP traffic, including multicast and broadcast traffic, enter the **unset interface vlan1 bypass-non-ip-all** command.
- To allow all L2 non-IP traffic to pass through the device, enter the **set interface vlan1 bypass-non-ip** command.
- To revert to the default behavior of the device, which is to block all non-IP and non-ARP unicast traffic, enter the **unset interface vlan1 bypass-non-ip** command.

  Note that the **unset interface vlan1 bypass-non-ip-all** command always overwrites the **unset interface vlan1 bypass-non-ip** command when both commands are in the configuration file. Therefore, if you had previously entered the **unset interface vlan1 bypass-non-ip-all** command, and you now want the device to revert to its default behavior of blocking only the non-IP and non-ARP unicast traffic, you should first enter the **set interface vlan1 bypass-non-ip** command to allow all non-IP and non-ARP traffic, including multicast, unicast, and broadcast traffic to pass through the device. Then you must enter the **unset interface vlan1 bypass-non-ip** command to block only the non-IP, non-ARP unicast traffic.
- To allow a security device to pass IPsec traffic without attempting to terminate it, use the **set interface vlan1 bypass-others-ipsec** command. The security device then allows the IPsec traffic to pass through to other VPN termination points.

NOTE: A security device with interfaces in transparent mode requires routes for two purposes: to direct self-initiated traffic, such as SNMP traps, and to forward VPN traffic after encapsulating or decapsulating it.

Forwarding IPv6 traffic

Juniper Networks security devices support IPv6 traffic in transparent mode. You can configure which IPv6 protocol the device can pass through in transparent mode by using the WebUI or CLI.

- To broadcast IPv6 packets that does not have a definite out interface defined in mac-learning table to all possible out interfaces, use **set interface vlan1** broadcast-ipv6 flood
- To allow the security device to forward ICMPv6 NDP traffic, use the **set interface vlan1 bypass-icmpv6-ndp** command. Similarly, you can configure the device to forward other ICMPv6 traffic such as Multicast Listener Discovery (MLD), Multicast Router Discovery (MRD), Mobile support protocol (MSP) and Secure Neighbor Discovery (SND) by using the WebUI or the CLI.

  For example, to configure the device to forward ICMpv6 MLD traffic:

WebUI

Network > Interfaces > Edit (For VLAN1). Perform the following action, then click **OK**.

Bypass IPv6 Multicast Listener Discovery packets (Select)

CLI

```
set interface vlan1 bypass-icmpv6-mld
```

To allow a security device to pass IPsec traffic encapsulated in an IPv6 protocol without decrypting the packets, use the **set interface vlan1 bypass-ipv6-others-ipsec** command. The security device then allows the IPsec traffic to pass through to other VPN termination points.

Unknown Unicast Options

When a host or any kind of network device does not know the MAC address associated with the IP address of another device, it uses the Address Resolution Protocol (ARP) to obtain it. The requestor broadcasts an ARP query (arp-q) to all the other devices on the same subnet. The arp-q requests the device at the specified destination IP address to send back an ARP reply (arp-r), which provides the requestor with the MAC address of the replier. When all the other devices on the subnet receive the arp-q, they check the destination IP address and, because it is not their IP address, drop the packet. Only the device with the specified IP address returns an arp-r. After a device matches an IP address with a MAC address, it stores the information in its ARP cache.

As ARP traffic passes through a security device in transparent mode, the device notes the source MAC address in each packet and learns which interface leads to that MAC address. In fact, the security device learns which interface leads to which MAC address by noting the source MAC addresses in all packets it receives. It then stores this information in its forwarding table.

NOTE: A security device in transparent mode does not permit any traffic between zones unless there is a policy configured on the device. For more information about how the device forwards traffic when it is in transparent mode, see "Traffic Forwarding" on page 103.

The situation can arise when a device sends a unicast packet with a destination MAC address, which it has in its ARP cache, but which the security device does not have in its forwarding table. For example, the security device clears its forwarding table every time it reboots. (You can also clear the forwarding table with the CLI command **clear arp**.) When a security device in transparent mode receives a unicast packet for which it has no entry in its forwarding table, it can follow one of two courses:

- After doing a policy lookup to determine the zones to which traffic from the source address is permitted, flood the initial packet out the interfaces bound to those zones, and then continue using whichever interface receives a reply. This is the Flood option, which is enabled by default.
- Drop the initial packet, flood ARP queries (and, optionally, trace-route packets, which are ICMP echo requests with the time-to-live value set to 1) out all interfaces (except the interface at which the packet arrived), and then send subsequent packets through whichever interface receives an ARP (or trace-route) reply from the router or host whose MAC address matches the destination MAC address in the initial packet. The trace-route option allows the security device to discover the destination MAC address when the destination IP address is in a nonadjacent subnet.

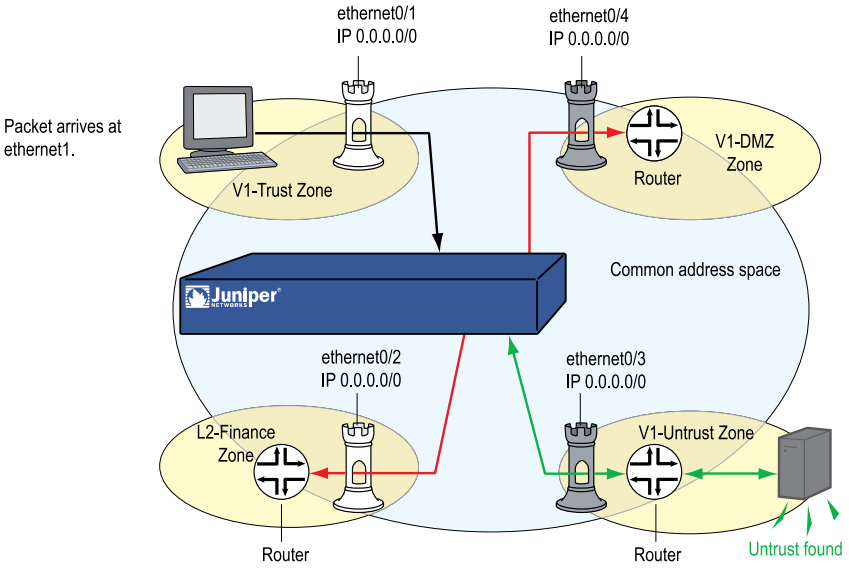
NOTE: Of the two methods—flood and ARP/trace-route—ARP/trace-route is more secure because the security device floods ARP queries and trace-route packets—not the initial packet—out all interfaces.

Flood Method

The flood method forwards packets in the same manner as most Layer 2 switches. A switch maintains a forwarding table that contains MAC addresses and associated ports for each Layer 2 domain. The table also contains the corresponding interface through which the switch can forward traffic to each device. Every time a packet arrives with a new source MAC address in its frame header, the switch adds the MAC address to its forwarding table. It also tracks the interface at which the packet arrived. If the destination MAC address is unknown to the switch, the switch duplicates the packet and floods it out all interfaces (other than the interface at which the packet arrived). It learns the previously unknown MAC address and its corresponding interface when a reply with that MAC address arrives at one of its interfaces.

When you enable the flood method and the security device receives an ethernet frame with a destination MAC address that is not listed in the security device MAC table, it floods the packet out all interfaces.

Figure 39: Flood Method



The security device floods the packet out ethernet0/2 but receives no reply.

To enable the flood method for handling unknown unicast packets, do either of the following:

WebUI

Network > Interfaces > Edit (for VLAN1): For the broadcast options, select **Flood**, then click **OK**:

CLI

```
set interface vlan1 broadcast flood
save
```

ARP/Trace-Route Method

When you enable the ARP method with the trace-route option and the security device receives an ethernet frame with a destination MAC address that is not listed in its MAC table, the security device performs the following series of actions:

NOTE: When you enable the ARP method, the trace-route option is enabled by default. You can also enable the ARP method without the trace-route option. However, this method only allows the security device to discover the destination MAC address for a unicast packet if the destination IP address is in the same subnet as the ingress IP address. (For more information about the ingress IP address, see the following Note.)

1. The security device notes the destination MAC address in the initial packet (and, if it is not already there, adds the source MAC address and its corresponding interface to its forwarding table).
2. The security device drops the initial packet.
3. The security device generates two packets—ARP query (arp-q) and a trace-route (an ICMP echo request, or PING) with a time-to-live (TTL) field of 1—and floods those packets out all interfaces except the interface at which the initial packet arrived. For the arp-q packets and ICMP echo requests, the security device uses the source and destination IP addresses from the initial packet. For arp-q packets, the security device replaces the source MAC address from the initial packet with the MAC address for VLAN1, and it replaces the destination MAC address from the initial packet with ffff.ffff.ffff. For the trace-route option, the security device uses the source and destination MAC addresses from the initial packet in the ICMP echo requests that it broadcasts.

   If the destination IP address belongs to a device in the same subnet as the ingress IP address, the host returns an ARP reply (arp-r) with its MAC address, thus indicating the interface through which the security device must forward traffic destined for that address. (See Figure 40 on page 108.)

NOTE: The ingress IP address refers to the IP address of the last device to send the packet to the security device. This device might be the source that sent the packet or a router forwarding the packet.

   If the destination IP address belongs to a device in a subnet beyond that of the ingress IP address, the trace-route returns the IP and MAC addresses of the router leading to the destination, and more significantly, indicates the interface through which the security device must forward traffic destined for that MAC address. (See Figure 41 on page 109.)

NOTE: Actually, the trace-route returns the IP and MAC addresses of all the routers in the subnet. The security device then matches the destination MAC address from the initial packet with the source MAC address on the arp-r packets to determine which router to target, and consequently, which interface to use to reach that target.

4. Combining the destination MAC address gleaned from the initial packet with the interface leading to that MAC address, the security device adds a new entry to its forwarding table.
5. The security device forwards all subsequent packets it receives out the correct interface to the destination.

To enable the ARP/trace-route method for handling unknown unicast packets, do either of the following:

WebUI

Network > Interfaces > Edit (for VLAN1): For the broadcast options, select **ARP**, then click **OK**:

CLI

```
set interface vlan1 broadcast arp
save
```

NOTE: The trace-route option is enabled by default. If you want to use ARP without the trace-route option, enter the following command: **unset interface vlan1 broadcast arp trace-route**. This command unsets the trace-route option but does not unset ARP as the method for handling unknown unicast packets.

Figure 40 on page 108 shows how the ARP method can locate the destination MAC when the destination IP address is in an adjacent subnet.

Figure 40: ARP Method

Note: Only the relevant elements of the packet header and the last four digits in the MAC addresses are shown below.

If the following packet:

Ethernet Frame			IP Datagram	
dst	src	type	src	dst
11bb	11aa	0800	210.1.1.5	210.1.1.75

arrives at ethernet0/1 and the forwarding table does not have an entry for MAC address 00bb.11bb.11bb, the security device floods the following arp-q packet:

Ethernet Frame			ARM Message	
dst	src	type	src	dst
ffff	39ce	0806	210.1.1.5	210.1.1.75

When the device receives the following arp-r at ethernet0/2:

Ethernet Frame			ARM Message	
dst	src	type	src	dst
39ce	11bb	0806	210.1.1.75	210.1.1.5

It can now associate the MAC address with the interface leading to it.

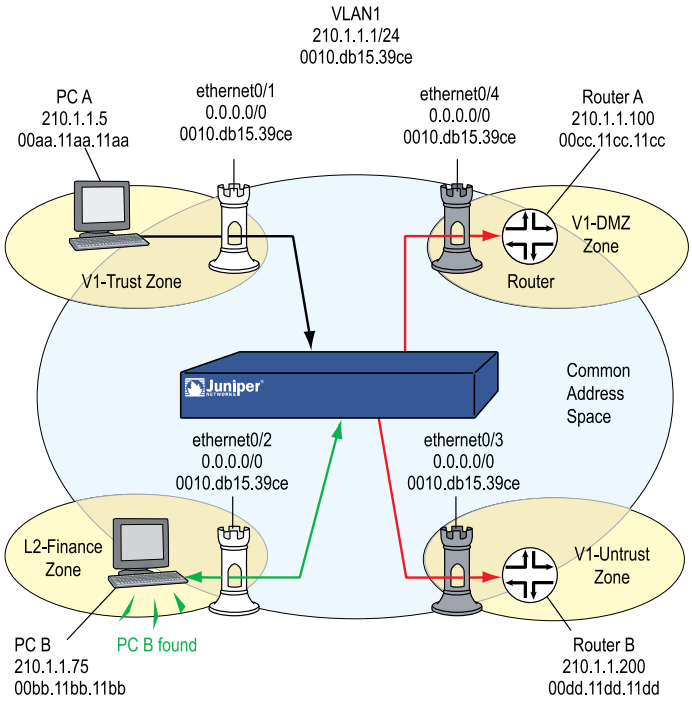


Figure 41 on page 109 shows how the trace-route option can locate the destination MAC when the destination IP address is in a nonadjacent subnet.

Figure 41: Trace-Route

Note: Only the relevant elements of the packet header and the last four digits in the MAC addresses are shown below.

If the following packet:

Ethernet Frame			IP Datagram	
dst	src	type	src	dst
11dd	11aa	0800	210.1.1.5	195.1.1.5

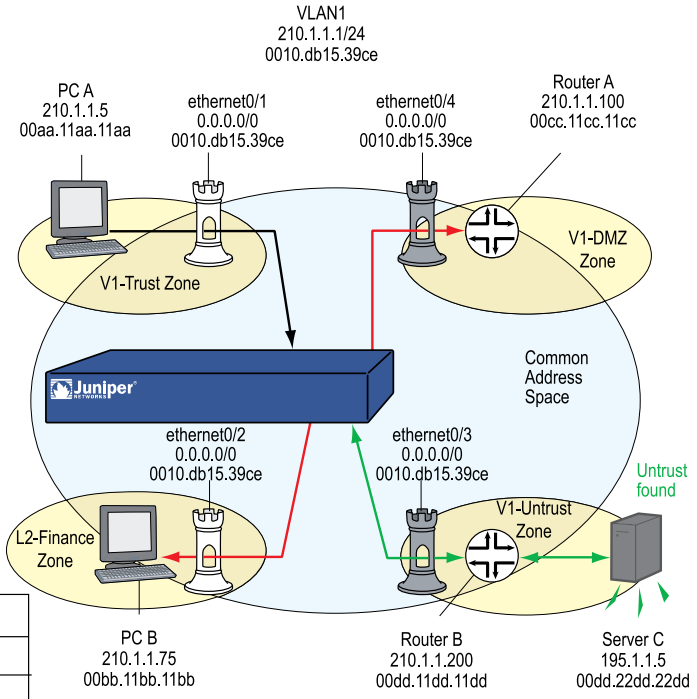
arrives at ethernet0/1 and the forwarding table does not have an entry for MAC address 00dd.11dd.11dd, the security device floods the following trace-route packet out ethernet0/2, ethernet0/, and ethernet0/4:

Ethernet Frame			ICMP Message		
dst	src	type	src	dst	TTL
11dd	11aa	0800	210.1.1.5	195.1.1.5	1

When the device receives the following response at ethernet0/3:

Ethernet Frame			ICMP Message		
dst	src	type	src	dst	msg
11aa	11dd	0800	210.1.1.200	210.1.1.5	Time Exceeded

It can now associate the MAC address with the interface leading to it.



Configuring VLAN1 Interface for Management

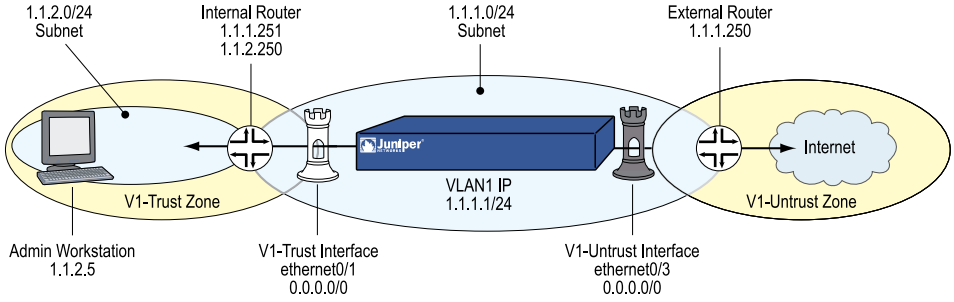
In this example, you configure the security device for management to its VLAN1 interface as follows:

- Assign the VLAN1 interface an IP address of 1.1.1.1/24.
- Enable Web, Telnet, SSH, and Ping on both the VLAN1 interface and the V1-Trust security zone.

NOTE: By default, ScreenOS enables the management options for the VLAN1 interface and V1-Trust security zone. Enabling these options is included in this example for illustrative purposes only. Unless you have previously disabled them, you really do not need to enable them manually.
To manage the device from a Layer 2 security zone, you must set the same management options for both the VLAN1 interface and the Layer 2 security zone.

- Add a route in the trust virtual router (all Layer 2 security zones are in the trust-vr routing domain) to enable management traffic to flow between the security device and an administrative workstation beyond the immediate subnet of the security device. All security zones are in the trust-vr routing domain.

Figure 42: Transparent VLAN


1.1.2.0/24
Subnet
Internal Router
1.1.1.251
1.1.2.250
1.1.1.0/24
Subnet
External Router
1.1.1.250
Internet
V1-Trust Zone
VLAN1 IP
1.1.1.1/24
V1-Untrust Zone
Admin Workstation
1.1.2.5
V1-Trust Interface
ethernet0/1
0.0.0.0/0
V1-Untrust Interface
ethernet0/3
0.0.0.0/0

WebUI

1. **VLAN1 Interface**

   Network > Interfaces > Edit (for VLAN1): Enter the following, then click **OK**:

   IP Address/Netmask: 1.1.1.1/24
   Management Services: WebUI, Telnet, SSH (select)
   Other Services: Ping (select)

2. **V1-Trust Zone**

   Network > Zones > Edit (for V1-Trust): Select the following, then click **OK**:

   Management Services: WebUI, Telnet, SSH
   Other Services: Ping

3. **Route**

   Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 1.1.2.0/24
   Gateway: (select)
   Interface: vlan1(trust-vr)
   Gateway IP Address: 1.1.1.251
   Metric: 1

CLI

1. **VLAN1 Interface**

```
set interface vlan1 ip 1.1.1.1/24
set interface vlan1 manage web
set interface vlan1 manage telnet
set interface vlan1 manage ssh
set interface vlan1 manage ping
```

2. **V1-Trust Zone**

```
set zone v1-trust manage web
set zone v1-trust manage telnet
set zone v1-trust manage ssh
set zone v1-trust manage ping
```

3. **Route**

```
set vrouter trust-vr route 1.1.2.0/24 interface vlan1 gateway 1.1.1.251 metric
 1
save
```

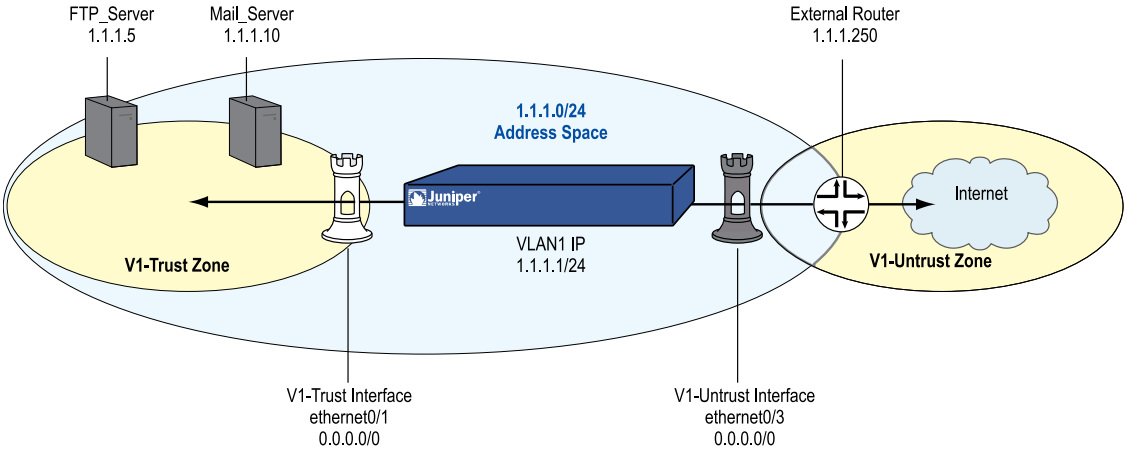
Configuring Transparent Mode

The following example illustrates a basic configuration for a single LAN protected by a security device in transparent mode. Policies permit outgoing traffic for all hosts in the V1-Trust zone, incoming SMTP services for the mail server, and incoming FTP-GET services for the FTP server.

To increase the security of management traffic, you change the HTTP port number for WebUI management from 80 to 5555, and the Telnet port number for CLI management from 23 to 4646. You use the VLAN1 IP address—1.1.1.1/24—to manage the security device from the V1-Trust security zone. You define addresses for the FTP and mail servers. You also configure a default route to the external router at 1.1.1.250, so that the security device can send outbound VPN traffic to it. (The default gateway on all hosts in the V1-Trust zone is also 1.1.1.250.)

NOTE: For an example of configuring a VPN tunnel for a security device with interfaces in transparent mode, see "Transparent Mode VPN" on page 875 .

Figure 43: Basic Transparent Mode



WebUI

1. **VLAN1 Interface**

   Network > Interfaces > Edit (for the VLAN1 interface): Enter the following, then click **OK**:

   IP Address/Netmask: 1.1.1.1/24
   Management Services: WebUI, Telnet (select)
   Other Services: Ping (select)

2. **HTTP Port**

Configuration > Admin > Management: In the HTTP Port field, type **5555** and then click **Apply**.

NOTE: The default port number is 80. Changing this to any number between 1024 and 32,767 is advised for discouraging unauthorized access to the configuration. When logging in to manage the device later, enter the following in the URL field of your browser: http://1.1.1.1:5555.

3. **Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

   Zone Name: V1-Trust
   IP Address/Netmask: 0.0.0.0/0

   Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

   Zone Name: V1-Untrust
   IP Address/Netmask: 0.0.0.0/0

4. **V1-Trust Zone**

   Network > Zones > Edit (for v1-trust): Select the following, then click **OK**:

   Management Services: WebUI, Telnet
   Other Services: Ping

5. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: FTP_Server
   IP Address/Domain Name:
   IP/Netmask: (select), 1.1.1.5/32
   Zone: V1-Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Mail_Server
   IP Address/Domain Name:
   IP/Netmask: (select), 1.1.1.10/32
   Zone: V1-Trust

6. **Route**

   Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 0.0.0.0/0
   Gateway: (select)

Interface: vlan1(trust-vr)
Gateway IP Address: 1.1.1.250
Metric: 1

7. **Policies**

Policy > Policies > (From: V1-Trust, To: V1-Untrust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: Any
Action: Permit

Policy > Policies > (From: V1-Untrust, To: V1-Trust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Mail_Server
Service: Mail
Action: Permit

Policy > Policies > (From: V1-Untrust, To: V1-Trust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), FTP_Server
Service: FTP-GET
Action: Permit

CLI

1. **VLAN1**

```
set interface vlan1 ip 1.1.1.1/24
set interface vlan1 manage web
set interface vlan1 manage telnet
set interface vlan1 manage ping
```

2. **Telnet**

```
set admin telnet port 4646
```

NOTE: The default port number for Telnet is 23. Changing this to any number between 1024 and 32,767 is advised for discouraging unauthorized access to the configuration. When logging in to manage the device later via Telnet, enter the following address: 1.1.1.1 4646.

3. **Interfaces**

```
set interface ethernet0/1 ip 0.0.0.0/0
set interface ethernet0/1 zone v1-trust
set interface ethernet0/3 ip 0.0.0.0/0
set interface ethernet0/3 zone v1-untrust
```

4. **V1-Trust Zone**

```
set zone v1-trust manage web
set zone v1-trust manage telnet
set zone v1-trust manage ping
```

5. **Addresses**

```
set address v1-trust FTP_Server 1.1.1.5/32
set address v1-trust Mail_Server 1.1.1.10/32
```

6. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface vlan1 gateway 1.1.1.250 metric 1
```

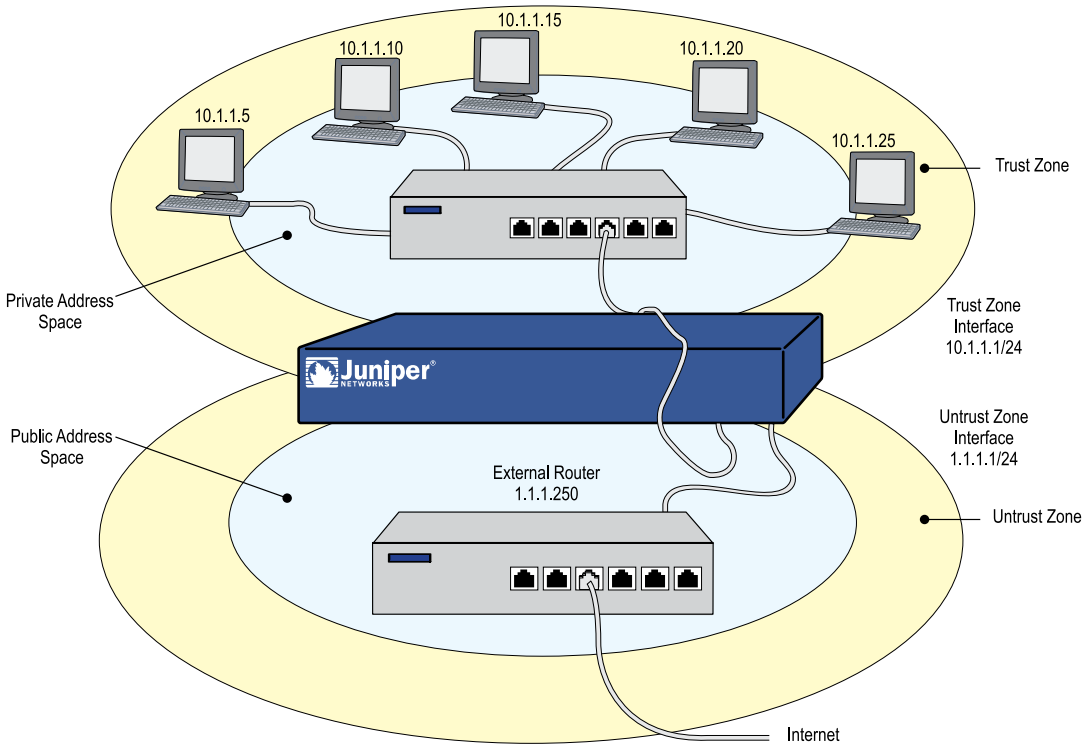
7. **Policies**

```
set policy from v1-trust to v1-untrust any any any permit
set policy from v1-untrust to v1-trust any Mail_Server mail permit
set policy from v1-untrust to v1-trust any FTP_Server ftp-get permit
save
```

NAT Mode

When an ingress interface is in Network Address Translation (NAT) mode, the security device, acting like a Layer 3 switch (or router), translates two components in the header of an outgoing IP packet destined for the Untrust zone: its source IP address and source port number. The security device replaces the source IP address of the originating host with the IP address of the Untrust zone interface. Also, it replaces the source port number with another random port number generated by the security device.

Figure 44: NAT Topology



When the reply packet arrives at the security device, the device translates two components in the IP header of the incoming packet: the destination address and port number, which are translated back to the original numbers. The security device then forwards the packet to its destination.

NAT adds a level of security not provided in transparent mode: The addresses of hosts sending traffic through an ingress interface in NAT mode (such as a Trust zone interface) are never exposed to hosts in the egress zone (such as the Untrust zone) unless the two zones are in the same virtual routing domain and the security device is advertising routes to peers through a dynamic routing protocol (DRP). Even then, the Trust zone addresses are only reachable if you have a policy permitting inbound traffic to them. (If you want to keep the Trust zone addresses hidden while using a DRP, then put the Untrust zone in the untrust-vr and the Trust zone in the trust-vr, and do not export routes for internal addresses in the trust-vr to the untrust-vr.)

If the security device uses static routing and just one virtual router, the internal addresses remain hidden when traffic is outbound, due to interface-based NAT. The policies you configure control inbound traffic. If you use only Mapped IP (MIP) and Virtual IP (VIP) addresses as the destinations in your inbound policies, the internal addresses still remain hidden.

Also, NAT preserves the use of public IP addresses. In many environments, resources are not available to provide public IP addresses for all devices on the network. NAT services allow many private IP addresses to have access to Internet resources through one or a few public IP addresses. The following IP address ranges are reserved for private IP networks and must not get routed on the Internet:

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

Inbound and Outbound NAT Traffic

A host in a zone sending traffic through an interface in NAT mode can initiate traffic to the Untrust zone—assuming that a policy permits it. In releases prior to ScreenOS 5.0.0, a host behind an interface in NAT mode was unable to receive traffic from the Untrust zone unless a Mapped IP (MIP), Virtual IP (VIP), or VPN tunnel was set up for it. However, in ScreenOS 5.0.0, traffic to a zone with a NAT-enabled interface from any zone—including the Untrust zone—does not need to use a MIP, VIP, or VPN. If you want to preserve the privacy of addresses or if you are using private addresses that do not occur on a public network such as the Internet, you can still define a MIP, VIP, or VPN for traffic to reach them. However, if issues of privacy and private IP addresses are not a concern, traffic from the Untrust zone can reach hosts behind an interface in NAT mode directly, without the use of a MIP, VIP, or VPN.

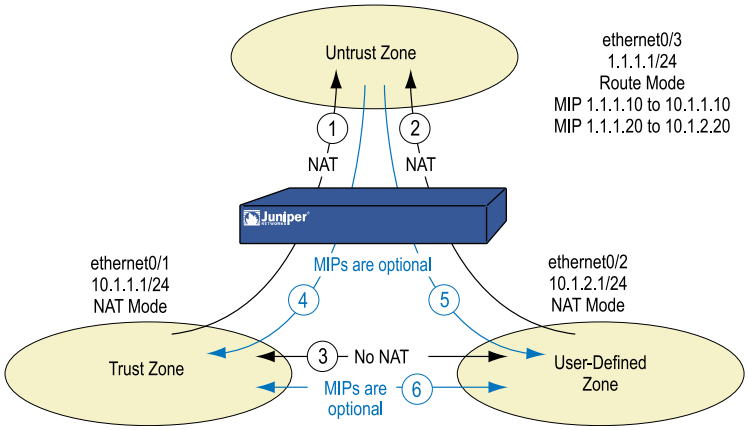
NOTE: You can define a Virtual IP (VIP) address only on an interface bound to the Untrust zone.

Figure 45: NAT Traffic Flow

1. Interface-based NAT on traffic from the Trust zone to the Untrust zone.
2. Interface-based NAT on traffic from the User-Defined zone to the Untrust zone.

   (Note: This is possible only if the User-Defined and Untrust zones are in different virtual routing domains.)
3. No interface-based NAT on traffic between the Trust and User-Defined zones.

4 and 5. You can use MIPs, VIPs, or VPNs for traffic from the Untrust zone to reach the Trust zone or the User-Defined zone, but they are not required.



NOTE: For more information about MIPs, see “Mapped IP Addresses” on page 1535. For more about VIPs, see “Virtual IP Addresses” on page 1552.

Interface Settings

For NAT mode, define the following interface shown in Table 7 on page 119, where ip_addr1 and ip_addr2 represent numbers in an IP address, mask represents the numbers in a netmask, vlan_id_num represents the number of a VLAN tag, zone represents the name of a zone, and number represents the bandwidth size in kbps.

Table 7: NAT Mode Interface Settings

Zone Interfaces	Settings	Zone Subinterfaces	Notes
Trust, DMZ, and user-defined zones using NAT	IP: ip_addr1 Netmask: mask Manage IP: ip_addr2 Traffic Bandwidth: number NAT: (select)	IP: ip_addr1 Netmask: mask VLAN Tag: vlan_id_num Zone Name: zone NAT: (select)	*Manage IP:* You can set the Manage IP address for each interface. Its primary purpose is to provide an IP address for administrative traffic separate from network traffic. You can also use the Manage IP address for accessing a specific device when it is in a high availability configuration. *Traffic Bandwidth:* An optional setting for traffic shaping. NAT: Select *NAT* or *Route* to define the interface mode.
Untrust	IP: ip_addr1 Netmask: mask Manage IP: ip_addr2 Traffic Bandwidth: number	IP: Netmask: mask VLAN Tag: vlan_id_num Zone Name: zone	*Untrust:* Although you are able to select NAT as the interface mode on an interface bound to the Untrust zone, the security device does not perform any NAT operations on that interface.

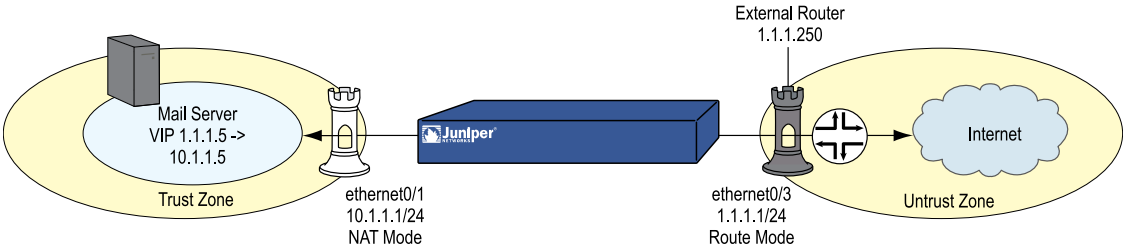
NOTE: In NAT mode, you can manage a security device from any interface—and from multiple interfaces—using the system IP address, interface IP addresses, Manage IP addresses, or the MGT IP address.

Configuring NAT Mode

The following example illustrates a simple configuration for a LAN with a single subnet in the Trust zone. The LAN is protected by a security device in NAT mode. Policies permit outgoing traffic for all hosts in the Trust zone and incoming mail for the mail server. The incoming mail is routed to the mail server through a Virtual IP address. Both the Trust and Untrust zones are in the trust-vr routing domain.

NOTE: Compare Figure 46 on page 120 with that for route mode in Figure 48 on page 126.

Figure 46: Device in NAT Mode



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

NOTE: By default, any interface that you bind to the Trust zone is in NAT mode. Consequently, this option is already enabled for interfaces bound to the Trust zone.

   Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24
   Interface Mode: Route

NOTE: If the IP address in the Untrust zone on the security device is dynamically assigned by an ISP, leave the IP address and netmask fields empty and select **Obtain IP using DHCP**. If the ISP uses Point-to-Point Protocol over Ethernet, select **Obtain IP using PPPoE**, click the **Create new PPPoE settings** link, and enter the name and password.

---

2. **VIP**

   Network > Interfaces > Edit (for ethernet0/3) > VIP: Enter the following, then click **Add**:

   Virtual IP Address: 1.1.1.5

   Network > Interfaces > Edit (for ethernet0/3) > VIP > New VIP Service: Enter the following, then click **OK**:

Virtual Port: 25
Map to Service: Mail
Map to IP: 10.1.1.5

NOTE: For information about Virtual IP (VIP) addresses, see "Virtual IP Addresses" on page 1552.

3. **Route**

   Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 0.0.0.0/0
   Gateway: (select)
   Interface: ethernet0/3
   Gateway IP Address: 1.1.1.250

4. **Policies**

   Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: ANY
   Action: Permit

   Policy > Policies > (From: Untrust, To: Global) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), VIP(1.1.1.5)
   Service: MAIL
   Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 10.1.1.1/24
set interface ethernet0/1 nat
set interface ethernet0/3 zone untrust
set interface ethernet0/3 ip 1.1.1.1/24
set interface ethernet0/3 route
```

NOTE: The **set interface ethernet**n **nat** command determines that the security device operates in NAT mode.
If the IP address in the Untrust zone on the security device is dynamically assigned by an ISP, use the following command: **set interface untrust dhcp**. If the ISP uses Point-to-Point Protocol over Ethernet, use the **set pppoe** and **exec pppoe** commands. For more information, see the *ScreenOS CLI Reference Guide: IPv4 Command Descriptions*.

2. **VIP**

```
set interface ethernet0/3 vip 1.1.1.5 25 mail 10.1.1.5
```

3. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet0/3 gateway 1.1.1.250
```

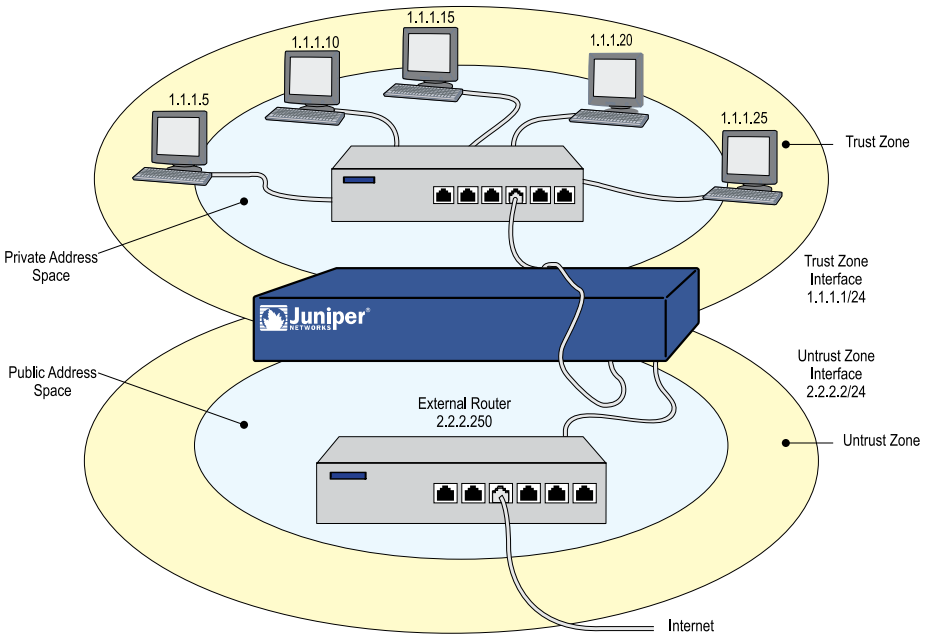
4. **Policies**

```
set policy from trust to untrust any any any permit
set policy from untrust to global any vip(1.1.1.5) mail permit
save
```

Route Mode

When an interface is in route mode, the security device routes traffic between different zones without performing source NAT (NAT-src); that is, the source address and port number in the IP packet header remain unchanged as it traverses the security device. Unlike NAT-src, you do not need to establish Mapped IP (MIP) and Virtual IP (VIP) addresses to allow inbound traffic to reach hosts when the destination zone interface is in route mode. Unlike transparent mode, the interfaces in each zone are on different subnets.

Figure 47: Route Mode Topology


1.1.1.15
1.1.1.10
1.1.1.20
1.1.1.5
1.1.1.25
Trust Zone
Private Address Space
Trust Zone Interface 1.1.1.1/24
Juniper
Public Address Space
Untrust Zone Interface 2.2.2.2/24
External Router 2.2.2.250
Untrust Zone
Internet

You do not have to apply Source Network Address Translation (NAT-src) at the interface level so that all source addresses initiating outgoing traffic get translated to the IP address of the destination zone interface. Instead, you can perform NAT-src selectively at the policy level. You can determine which traffic to route and on which traffic to perform NAT-src by creating policies that enable NAT-src for specified source addresses on either incoming or outgoing traffic. For network traffic, NAT can use the IP address or addresses of the destination zone interface from a Dynamic IP (DIP) pool, which is in the same subnet as the destination zone interface. For VPN traffic, NAT can use a tunnel interface IP address or an address from its associated DIP pool.

NOTE: For more information about configuring policy-based NAT-src, see "Source Network Address Translation" on page 1481.

Interface Settings

For route mode, define the following interface settings as shown in Table 8 on page 125, where ip_addr1 and ip_addr2 represent numbers in an IP address, mask represents the numbers in a netmask, vlan_id_num represents the number of a VLAN tag, zone represents the name of a zone, and number represents the bandwidth size in kbps.

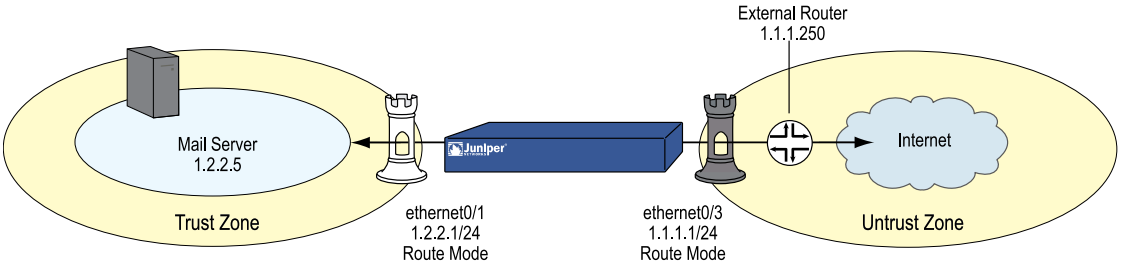
Table 8: Route Mode Interface Settings

Zone Interfaces	Settings	Zone Subinterfaces	Notes
Trust, Untrust, DMZ, and user-defined zones using NAT	IP: ip_addr1 Netmask: mask Manage IP: ip_addr2 Traffic Bandwidth: number Route: (select)	IP: ip_addr1 Netmask: mask VLAN Tag: vlan_id_num Zone Name: zone Route: (select)	**Manage IP:** You can set the Manage IP address for each interface. Its primary purpose is to provide an IP address for administrative traffic separate from network traffic. You can also use the Manage IP address for accessing a specific device when it is in a high availability configuration. **Traffic Bandwidth:** An optional setting for traffic shaping. **NAT:** Select **NAT** or **Route** to define the interface mode.

Configuring Route Mode

In "Configuring NAT Mode" on page 119, the hosts in the Trust zone LAN have private IP addresses and a Mapped IP for the mail server. In the following example of the same network protected by a security device operating in route mode, note that the hosts have public IP addresses and that a MIP is unnecessary for the mail server. Both security zones are in the trust-vr routing domain.

Figure 48: Device in Route Mode



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.2.2.1/24
   Enter the following, then click **OK**:
   Interface Mode: Route

   Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

NOTE: Selecting **Route** determines that the security device operates in route mode, without performing NAT on traffic entering or exiting the Trust zone.
If the IP address in the Untrust zone on the security device is dynamically assigned by an ISP, leave the IP address and netmask fields empty and select **Obtain IP using DHCP**. If the ISP uses Point-to-Point Protocol over Ethernet, select **Obtain IP using PPPoE**, click the **Create new PPPoE settings** link, and then enter the name and password.

---

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following and then click **OK**:

   Address Name: Mail Server
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.5/32
   Zone: Trust

3. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet0/3
Gateway IP Address: 1.1.1.250

4. **Policies**

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: ANY
Action: Permit

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Mail Server
Service: MAIL
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 1.2.2.1/24
set interface ethernet0/1 route
set interface ethernet0/3 zone untrust
set interface ethernet0/3 ip 1.1.1.1/24
set interface ethernet0/3 route
```

NOTE: The **set interface ethernet** number **route** command determines that the security device operates in route mode.

2. **Address**

```
set address trust mail_server 1.2.2.5/24
```

3. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet0/3 gateway 1.1.1.250
```

4. **Policies**

```
set policy from trust to untrust any any any permit
set policy from untrust to trust any mail_server mail permit
save
```

Chapter 6

Building Blocks for Policies

This chapter discusses the components, or building blocks, that you can reference in policies. It contains the following sections:

NOTE: For information about user authentication, see "User Authentication" on page 1563.

Addresses

ScreenOS classifies the addresses of all other devices by location and by netmask. Each zone possesses its own list of addresses and address groups.

Because an individual host has a single IP address defined, it must have a netmask setting of 255.255.255.255 (which masks out all but this host). Subnets have an IP address and a netmask (for example, 255.255.255.0 or 255.255.0.0).

A wildcard address contains a range of address, enabling you to reduce the number of policies that you create. A wildcard address has an IP address and a wildcard mask (for example, 0.0.255.0). For more information on wildcard addresses, see "Wildcard Addresses" on page 203.

Before you can configure policies to permit, deny, or tunnel traffic to and from individual hosts and subnets, you must make entries for them in ScreenOS address lists, which are organized by zones.

NOTE: You do not have to make address entries for "Any." This term automatically applies to all devices physically located within their respective zones.

Address Entries

Before you can set up many of the Juniper Networks firewall, VPN, and traffic-shaping features, you need to define addresses in one or more address lists. The address list for a security zone contains the IP addresses or domain names of hosts or subnets whose traffic is either allowed, blocked, encrypted, or user-authenticated.

NOTE: Before you can use domain names for address entries, you must configure the security device for Domain Name System (DNS) services. For information about DNS configuration, see "Domain Name System Support" on page 263.
For information regarding ScreenOS naming conventions—which apply to the names you create for addresses—see "Naming Conventions and Character Types" on page 11.

Adding an Address

In this example, you add the subnet "Sunnyvale_Eng" with the IP address 10.1.10.0/24 as an address in the Trust zone, and the address www.juniper.net as an address in the Untrust zone.

WebUI

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Sunnyvale_Eng
IP Address/Domain Name:
    IP/Netmask: (select), 10.1.10.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Juniper
IP Address/Domain Name:
    Domain Name: (select), www.juniper.net
Zone: Untrust

CLI

```
set address trust Sunnyvale_Eng 10.1.10.0/24
set address untrust Juniper www.juniper.net
save
```

Modifying an Address

In this example, you change the address entry for the address "Sunnyvale_Eng" to reflect that this department is specifically for software engineering and has a different IP address—10.1.40.0/24.

WebUI

Policy > Policy Elements > Addresses > List > Edit (for Sunnyvale_Eng): Change the name and IP address to the following, then click **OK**:

```
Address Name: Sunnyvale_SW_Eng
IP Address/Domain Name:
      IP/Netmask: (select), 10.1.40.0/24
Zone: Trust
```

CLI

```
unset address trust Sunnyvale_Eng
set address trust Sunnyvale_SW_Eng 10.1.40.0/24
save
```

NOTE: After you define an address—or an address group—and associate it with a policy, you cannot change the address location to another zone (such as from Trust to Untrust). To change its location, you must first disassociate it from the underlying policy.

Deleting an Address

In this example, you remove the address entry for the address "Sunnyvale_SW_Eng".

WebUI

Policy > Policy Elements > Addresses > List: Click **Remove** in the Configure column for Sunnyvale_SW_Eng.

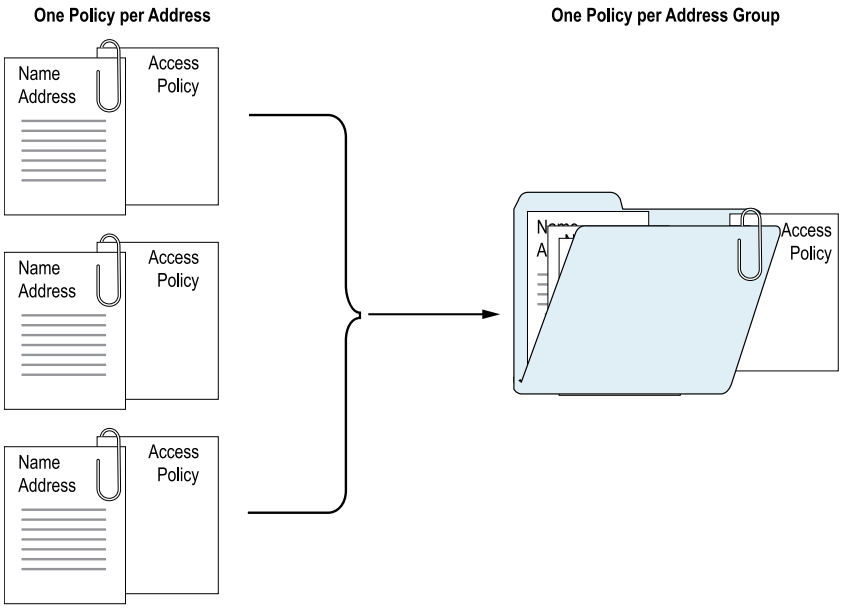
CLI

```
unset address trust "Sunnyvale_SW_Eng"
save
```

Address Groups

"Address Entries" on page 130 explained how you create, modify, and delete address book entries for individual hosts and subnets. As you add addresses to an address list, it becomes difficult to manage how policies affect each address entry. ScreenOS allows you to create groups of addresses. Rather than manage a large number of address entries, you can manage a small number of groups. Changes you make to the group are applied to each address entry in the group.

Figure 49: Address Groups



The address group option has the following features:

- You can create address groups in any zone.
- You can create address groups with existing users, or you can create empty address groups and later fill them with users.
- An address group can be a member of another address group.

NOTE: To ensure that a group does not accidentally contain itself as a member, the security device performs a sanity check when you add one group to another. For example, if you add group A as a member to group B, the security device automatically checks that A does not already contain B as its member.

- You can reference an address group entry in a policy like an individual address book entry.
- ScreenOS applies policies to each member of the group by internally creating individual policies for each group member. While you only have to create one policy for a group, ScreenOS actually creates an internal policy for each member in the group (as well as for each service configured for each user).

NOTE: The automatic nature by which the security device applies policies to each address group member saves you from having to create them individually for each address. Furthermore, ScreenOS writes these policies to the application-specific integrated circuit (ASIC), which speeds up lookups.

- When you delete an individual address book entry from the address book, the security device automatically removes it from all groups to which it belonged.

The following constraints apply to address groups:

- Address groups can only contain addresses that belong to the same zone.
- Address names cannot be the same as group names. If the name “Paris” is used for an individual address entry, it cannot be used for a group name.
- If an address group is referenced in a policy, the group cannot be removed. It can, however, be edited.
- When a single policy is assigned to an address group, it is applied to each group member individually, and the security device makes an entry for each member in the access control list (ACL). If you are not vigilant, it is possible to exceed the number of available policy resources, especially if both the source and destination addresses are address groups and the specified service is a service group.
- You cannot add the predefined addresses: “Any,” “ All Virtual IPs,” and “ Dial-Up VPN” to groups.

Creating an Address Group

In the following example, you create a group named “HQ 2nd Floor” that includes “ Santa Clara Eng” and “ Tech Pubs,” two addresses that you have already entered in the address book for the Trust zone.

WebUI

Policy > Policy Elements > Addresses > Groups > (for Zone: Trust) New: Enter the following group name, move the following addresses, then click **OK**:

Group Name: HQ 2nd Floor

Select **Santa Clara Eng** and use the < < button to move the address from the Available Members column to the Group Members column.

Select **Tech Pubs** and use the < < button to move the address from the Available Members column to the Group Members column.

CLI

```
set group address trust “HQ 2nd Floor”  add “ Santa Clara Eng”
set group address trust “ HQ 2nd Floor”  add “ Tech Pubs”
save
```

Editing an Address Group Entry

In this example, you add “Support” (an address that you have already entered in the address book) to the “ HQ 2nd Floor” address group.

WebUI

Policy > Policy Elements > Addresses > Groups > (for Zone: Trust) Edit (for HQ 2nd Floor): Move the following address, then click **OK**:

Select **Support** and use the < < button to move the address from the Available Members column to the Group Members column.

CLI

```
set group address trust “HQ 2nd Floor”  add Support
save
```

Removing a Member and a Group

In this example, you remove the member “Support” from the HQ 2nd Floor address group, and delete “ Sales,” an address group that you had previously created.

WebUI

Policy > Policy Elements > Addresses > Groups > (for Zone: Trust) Edit (HQ 2nd Floor): Move the following address, then click **OK**:

Select **Support** and use the > > button to move the address from the Group Members column to the Available Members column.

Policy > Policy Elements > Addresses > Groups > (Zone: Trust): Click **Remove** in the Configure column for Sales.

CLI

```
unset group address trust “HQ 2nd Floor”  remove Support
unset group address trust Sales
save
```

NOTE: The security device does not automatically delete a group from which you have removed all names.

Services

Services are types of traffic for which protocol standards exist. Each service has a transport protocol and destination port number(s) associated with it, such as TCP/port 21 for FTP and TCP/port 23 for Telnet. When you create a policy, you must specify a service for it. You can select one of the predefined services from the service book, or a custom service or service group that you created. You can see which service you can use in a policy by viewing the Service drop-down list on the Policy Configuration page (WebUI) or by using the **get service** command (CLI).

Predefined Services

You can view the list of predefined or custom services or service groups on the security device using the WebUI or the CLI.

- Using the WebUI:

- Policy > Policy Elements > Services > Predefined
- Policy > Policy Elements > Services > Custom
- Policy > Policy Elements > Services > Groups

- Using the CLI:

  get service [group | predefined | user]

NOTE: Each predefined service has a source port range of 1-65535, which includes the entire set of valid port numbers. This prevents potential attackers from gaining access by using a source port outside of the range. If you need to use a different source port range for any predefined service, create a custom service. For information, see "Custom Services" on page 149.

This section contains information about the following predefined services:

You can find more detailed information about some of these listed on the following pages:

Internet Control Messaging Protocol

Internet Control Messaging Protocol (ICMP) is a part of IP and provides a way to query a network (ICMP Query messages) and to receive feedback from the network for error patterns (ICMP Error messages). ICMP does not, however, guarantee error message delivery or report all lost datagrams; for these reasons it is not considered a reliable protocol. ICMP codes and types describe ICMP Query messages and ICMP Error messages.

You can choose to permit or deny any or specific types of ICMP messages to improve network security. Some types of ICMP messages can be exploited to gain information about your network that might compromise security. For example, ICMP, TCP, or UDP packets can be constructed to return ICMP error messages that contain information about a network, such as its topology, and access list filtering characteristics. Table 9 on page 136 lists ICMP message names, the corresponding code, type, and description.

Table 9: ICMP Information

ICMP Message Name	Type	Code	Description
ICMP-ANY	all	all	ICMP-ANY affects any protocol using ICMP. Denying ICMP-ANY impairs any attempt to ping or monitor a network using ICMP. Permitting ICMP-ANY allows all ICMP messages.
ICMP-ADDRESS-MASK ■ Request ■ Reply	17 18	0 0	ICMP Address Mask Query is used for systems that need the local subnet mask from a bootstrap server. Denying ICMP Address Mask request messages can adversely affect diskless systems. Permitting ICMP Address Mask request messages might allow others to fingerprint the operating system of a host in your network.
ICMP-DEST-UNREACH	3	0	ICMP Destination Unreachable Error message indicates that the destination host is configured to reject the packets. Codes 0, 1, 4, or 5 can be from a gateway. Codes 2 or 3 from a host (RFC 792). Denying ICMP Destination Unreachable Error messages can remove the assumption that a host is up and running behind a security device. Permitting ICMP Destination Unreachable Error messages can allow some assumptions, such as security filtering, to be made about the network.
ICMP Fragment Needed	3	4	ICMP Fragmentation Error message indicates that fragmentation is needed but the don't fragment flag is set. Denying these messages from the Internet (untrust) to the trusted network is recommended.

Table 9: ICMP Information *(continued)*

ICMP Message Name	Type	Code	Description
ICMP Fragment Reassembly	11	1	ICMP Fragment Reassembly Time Exceeded Error indicates that a host reassembling a fragmented message ran out of time and dropped the packet. This message is sometimes sent. Denying these messages from the Internet (untrust) to the trusted network is recommended.
ICMP-HOST-UNREACH	3	1	ICMP Host Unreachable Error messages indicate that routing table entries do not list or list as infinity a particular host. Sometimes this error is sent by gateways that cannot fragment when a packet requiring fragmentation is received. Denying these messages from the Internet (untrust) to the trusted network is recommended. Permitting these messages allows others to be able to determine your internal hosts IP addresses by a process of elimination or make assumptions about gateways and fragmentation.
ICMP-INFO ■ Request ■ Reply	15 16	0 0	ICMP-INFO Query messages allow diskless host systems to query the network and self-configure. Denying ICMP Address Mask request messages can adversely affect diskless systems. Permitting ICMP Address Mask request messages might allow others to broadcast information queries to a network segment to determine computer type.
ICMP-PARAMETER-PROBLEM	12	0	ICMP Parameter Problem Error messages notify you when incorrect header parameters are present and caused a packet to be discarded Denying these messages from the Internet (untrust) to a trusted network is recommended. Permitting ICMP Parameter Problem error messages allows others to make assumptions about your network.
ICMP-PORT-UNREACH	3	3	ICMP Port Unreachable Error messages indicate that gateways processing datagrams requesting certain ports are unavailable or unsupported in the network. Denying these messages from the Internet (untrust) to the trusted network is recommended. Permitting ICMP Port Unreachable Error messages can allow others to determine which ports you use for certain protocols.
ICMP-PROTOCOL-UNREACH	3	2	ICMP Protocol Unreachable Error messages indicate that gateways processing datagrams requesting certain protocols are unavailable or unsupported in the network. Denying these messages from the Internet (untrust) to the trusted network is recommended. Permitting ICMP Protocol Unreachable Error messages can allow others to determine what protocols your network is running.

Table 9: ICMP Information *(continued)*

ICMP Message Name	Type	Code	Description
ICMP-REDIRECT	5	0	ICMP Redirect Network Error messages are sent by routers. Denying these messages from the Internet (untrust) to the trusted network is recommended.
ICMP-REDIRECT-HOST	5	1	ICMP redirect messages indicate datagrams destined for the specified host to be sent along another path.
ICMP-REDIRECT-TOS-HOST	5	3	ICMP Redirect Type of Service (TOS) and Host Error is a type of message.
ICMP-REDIRECT-TOS-NET	5	2	ICMP Redirect TOS and Network Error is a type of message.
ICMP-SOURCE-QUENCH	4	0	ICMP Source Quench Error message indicates that a router does not have the buffer space available to accept, queue, and send the packets on to the next hop. Denying these messages will not help or impair internal network performance. Permitting these messages can allow others to know that a router is congested making it a viable attack target.
ICMP-SOURCE-ROUTE-FAIL	3	5	ICMP Source Route Failed Error message Denying these messages from the Internet (untrust) is recommended.
ICMP-TIME-EXCEEDED	11	0	ICMP Time to Live (TTL) Exceeded Error message indicates that a packet's TTL setting reached zero before the packet reached its destination. This ensures that older packets are discarded before resent ones are processed. Denying these messages from a trusted network out to the Internet is recommended.
ICMP-TIMESTAMP ■ Request ■ Reply	13 14	0 0	ICMP-TIMESTAMP Query messages provide the mechanism to synchronize time and coordinate time distribution in a large, diverse network.
Ping (ICMP ECHO)	8	0	Packet Internet Groper is a utility to determine whether a specific host is accessible by its IP address.0/0 echo reply. Denying ping functionality removes your ability to check to see if a host is active. Permitting ping can allow others to execute a denial of service (DoS) or Smurf attack.
ICMP-ECHO-FRAGMENT-ASSEMBLY-EXPIRE	11	1	ICMP Fragment Echo Reassembly Time Expired error message indicates that the reassembly time was exceeded. Denying these messages is recommended.
Traceroute ■ Forward ■ Discard	30 30	0 1	Traceroute is a utility to indicate the path to access a specific host. Denying this utility from the Internet (untrust) to your internal network (trust) is recommended.

Handling ICMP Unreachable Errors

For different levels of security, the default behavior for ICMP unreachable errors from downstream routers is handled as follows:

- Sessions do not close for ICMP type 3 code 4 messages.

  ICMP messages pass through without dropping sessions. Packets are however, dropped per session.

- Sessions do not close on receiving any kind of ICMP unreachable messages.
- Sessions store ICMP unreachable message, thereby restricting the number of messages flowing through to 1.

  One ICMP unreachable message is generated globally per device. The remaining ICMP unreachable errors are dropped.

Internet-Related Predefined Services

Table 10 on page 139 lists Internet-related predefined services. Depending on your network requirements, you can choose to permit or deny any or all of these services. Each entry lists the service name, default receiving port, and service description.

Table 10: Predefined Services

Service Name	Port(s)	Service Description
AOL	5190-5193	America Online Internet Service Provider (ISP) provides Internet, chat, and instant messaging services.
DHCP-Relay	67 (default)	Dynamic Host Configuration.
DHCP	68 (client) 67 (server)	Dynamic Host Configuration Protocol allocates network addresses and delivers configuration parameters from server to hosts.
DNS	53	Domain Name System translates domain names into IP addresses.
FTP ■ FTP-Get ■ FTP-Put	20 (data) 21 (control)	File Transfer Protocol (FTP) allows the sending and receiving of files between machines. You can choose to deny or permit ANY (GET or PUT) or to selectively permit or deny either GET or PUT. GET receives files from another machine and PUT sends files to another machine. We recommend denying FTP services from untrusted sources (Internet).
Gopher	70	Gopher organizes and displays Internet servers' contents as a hierarchically structured list of files. We recommend denying Gopher access to avoid exposing your network structure.

Table 10: Predefined Services *(continued)*

Service Name	Port(s)	Service Description
HTTP	8080	HyperText Transfer Protocol is the underlying protocol used by the World Wide Web (WWW). Denying HTTP service disables your users from viewing the Internet. Permitting HTTP service allows your trusted hosts to view the Internet.
HTTP-EXT	—	Hypertext Transfer Protocol with extended non-standard ports
HTTPS	443	Hypertext Transfer Protocol with Secure Socket Layer (SSL) is a protocol for transmitting private documents through the Internet. Denying HTTPS disables your users from shopping on the Internet and from accessing certain online resources that require secure password exchange. Permitting HTTPS allows your trusted hosts to participate in password exchange, shop online and visit various protected online resources that require user login.
Internet Locator Service	—	Internet Locator Service includes LDAP, User Locator Service, and LDAP over TSL/SSL.
IRC	6665-6669	Internet Relay Chat (IRC) allows people connected to the Internet to join live discussions.
LDAP	389	Lightweight Directory Access Protocol is a set of protocols used to access information directories.
PC-Anywhere	—	PC-Anywhere is a remote control and file transfer software.
TFTP	69	TFTP is a protocol for simple file transfer.
WAIS	—	Wide Area Information Server is a program that finds documents on the Internet.

Microsoft Remote Procedure Call Services

Table 11 on page 141 lists predefined Microsoft services, parameters associated with each service, and a brief description of each service. Parameters include Universal Unique Identifiers (UUIDs) and TCP/UDP source and destination ports. A UUID is a 128-bit unique number generated from a hardware address, a timestamp, and seed values.

Table 11: Microsoft Services

Service	Parameter/UUID	Description
MS-RPC-EPM	135 e1af8308-5d1f-11c9-91a4-08002b14a0fa	Microsoft Remote Procedure Call (RPC) Endpoint Mapper (EPM) Protocol
MS-RPC-ANY	—	Any Microsoft Remote Procedure Call (RPC) Services
MS-AD	4 members	Microsoft Active Directory Service Group includes: ■ MS-AD-BR ■ MS-AD-DRSUAPI ■ MS-AD-DSROLE ■ MS-AD-DSSETUP
MS-EXCHANGE	6 members	Microsoft Exchange Service Group includes: ■ MS-EXCHANGE-DATABASE ■ MS-EXCHANGE-DIRECTORY ■ MS-EXCHANGE-INFO-STORE ■ MS-EXCHANGE-MTA ■ MS-EXCHANGE-STORE ■ MS-EXCHANGE-SYSATD
MS-IIS	6 members	Microsoft IIS Server Service Group includes: ■ MS-IIS-COM ■ MS-IIS-IMAP4 ■ MS-IIS-INETINFO ■ MS-IIS-NNTP ■ MS-IIS-POP3 ■ MS-IIS-SMTP
MS-AD-BR	ecec0d70-a603-11d0-96b1-00a0c91ece30 16e0cf3a-a604-11d0-96b1-00a0c91ece30	Microsoft Active Directory Backup and Restore Services
MS-AD-DRSUAPI	e3514235-4b06-11d1-ab04-00c04fc2dcd2	Microsoft Active Directory Replication Service
MS-AD-DSROLE	1cbcad78-df0b-4934-b558-87839ea501c9	Microsoft Active Directory DSROLE Service
MS-AD-DSSETUP	3919286a-b10c-11d0-9ba8-00c04fd92ef5	Microsoft Active Directory Setup Service
MS-DTC	906b0ce0-c70b-1067-b317-00dd010662da	Microsoft Distributed Transaction Coordinator Service
MS-EXCHANGE-DATABASE	1a190310-bb9c-11cd-90f8-00aa00466520	Microsoft Exchange Database Service
MS-EXCHANGE-DIRECTORY	f5cc5a18-4264-101a-8c59-08002b2f8426 f5cc5a7c-4264-101a-8c59-08002b2f8426 f5cc59b4-4264-101a-8c59-08002b2f8426	Microsoft Exchange Directory Service

Table 11: Microsoft Services *(continued)*

Service	Parameter/UUID	Description
MS-EXCHANGE-INFO-STORE	0e4a0156-dd5d-11d2-8c2f-00c04fb6bcde 1453c42c-0fa6-11d2-a910-00c04f990f3b 10f24e8e-0fa6-11d2-a910-00c04f990f3b 1544f5e0-613c-11d1-93df-00c04fd7bd09	Microsoft Exchange Information Store Service
MS-EXCHANGE-MTA	9e8ee830-4459-11ce-979b-00aa005ffebe 38a94e72-a9bc-11d2-8faf-00c04fa378ff	Microsoft Exchange MTA Service
MS-EXCHANGE-STORE	99e66040-b032-11d0-97a4-00c04fd6551d 89742ace-a9ed-11cf-9c0c-08002be7ae86 a4f1db00-ca47-1067-b31e-00dd010662da a4f1db00-ca47-1067-b31f-00dd010662da	Microsoft Exchange Store Service
MS-EXCHANGE-SYSATD	67df7c70-0f04-11ce-b13f-00aa003bac6c f930c514-1215-11d3-99a5-00a0c9b61b04 83d72bf0-0d89-11ce-b13f-00aa003bac6c 469d6ec0-0d87-11ce-b13f-00aa003bac6c 06ed1d30-d3d3-11cd-b80e-00aa004b9c30	Microsoft Exchange System Attendant Service
MS-FRS	f5cc59b4-4264-101a-8c59-08002b2f8426 d049b186-814f-11d1-9a3c-00c04fc9b232 a00c021c-2be2-11d2-b678-0000f87a8f8e	Microsoft File Replication Service
MS-IIS-COM	70b51430-b6ca-11d0-b9b9-00a0c922e750 a9e69612-b80d-11d0-b9b9-00a0c922e70	Microsoft Internet Information Server COM GUID/UUID Service
MS-IIS-IMAP4	2465e9e0-a873-11d0-930b-00a0c90ab17c	Microsoft Internet Information Server IMAP4 Service
MS-IIS-INETINFO	82ad4280-036b-11cf-972c-00aa006887b0	Microsoft Internet Information Server Administration Service
MS-IIS-NNTP	4f82f460-0e21-11cf-909e-00805f48a135	Microsoft Internet Information Server NNTP Service
MS-IIS-POP3	1be617c0-31a5-11cf-a7d8-00805f48a135	Microsoft Internet Information Server POP3 Service
MS-IIS-SMTP	8cfb5d70-31a4-11cf-a7d8-00805f48a135	Microsoft Internet Information Server STMP Service

Table 11: Microsoft Services *(continued)*

Service	Parameter/UUID	Description
MS-ISMSERV	68dcd486-669e-11d1-ab0c-00c04fc2dcd2 130ceefb-e466-11d1-b78b-00c04fa32883	Microsoft Inter-site Messaging Service
MS-MESSENGER	17fdd703-1827-4e34-79d4-24a55c53bb37 5a7b91f8-ff00-11d0-a9b2-00c04fb6e6fc	Microsoft Messenger Service
MS-MQQM	fdb3a030-065f-11d1-bb9b-00a024ea5525 76d12b80-3467-11d3-91ff-0090272f9ea3 1088a980-eae5-11d0-8d9b-00a02453c33 5b5b3580-b0e0-11d1-b92d-0060081e87f0 41208ee0-e970-11d1-9b9e-00e02c064c39	Microsoft Windows Message Queue Management Service
MS-NETLOGON	12345678-1234-abcd-ef00-01234567cffb	Microsoft Netlogon Service
MS-SCHEDULER	1ff70682-0a51-30e8-076d-740be8cee98b 378e52b0-c0a9-11cf-822d-00aa0051e40f 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53	Microsoft Scheduler Service
MS-WIN-DNS	50abc2a4-574d-40b3-9d66-ee4fd5fba076	Microsoft Windows DNS Server
MS-WINS	45f52c28-7f9f-101a-b52b-08002b2efabe 811109bf-a4e1-11d1-ab54-00a0c91e9b45	Microsoft WINS Service

Dynamic Routing Protocols

Depending on your network requirements, you can choose to permit or deny messages generated from and packets of these dynamic routing protocols. Table 12 on page 143 lists each supported dynamic routing protocol by name, port, and description.

Table 12: Dynamic Routing Protocols

Dynamic Routing Protocol	Port	Description
RIP	520	RIP is a common distance-vector routing protocol.
OSPF	89	OSPF is a common link-state routing protocol.
BGP	179	BGP is an exterior/interdomain routing protocol.

Streaming Video

Table 13 on page 144 lists each supported streaming video service by name and includes the default port and description. Depending on your network requirements, you can choose to permit or deny any or all of these services.

Table 13: Streaming Video Services

Service	Port	Description
H.323	TCP source 1-65535; TCP destination 1720, 1503, 389, 522, 1731 UDP source 1-65535; UDP source 1719	H.323 is a standard approved by the International Telecommunication Union (ITU) that defines how audiovisual conference data is transmitted across networks.
NetMeeting	TCP source 1-65535; TCP destination 1720, 1503, 389, 522 UDP source 1719	Microsoft NetMeeting uses TCP to provide teleconferencing (video and audio) services over the Internet.
Real media	TCP source 1-65535; TCP destination 7070	Real Media is streaming video and audio technology.
RTSP	554	Real Time Streaming Protocol (RTSP) for streaming media applications
SIP	5056	Session Initiation Protocol (SIP) is an Application Layer control protocol for creating, modifying, and terminating sessions.
VDO Live	TCP source 1-65535; TCP destination 7000-7010	VDOLive is a scalable, video streaming technology.

Sun Remote Procedure Call Services

Table 14 on page 144 lists each Sun Remote Procedure Call Application Layer Gateway (RPC ALG) service name, program numbers, and full name.

Table 14: Remote Procedure Call Application Layer Gateway Services

Service	Program Numbers	Full Name
SUN-RPC-PORTMAPPER	111 100000	Sun RPC Portmapper Protocol
SUN-RPC-ANY	ANY	Any Sun RPC services
SUN-RPC-PROGRAM-MOUNTD	100005	Sun RPC Mount Daemon
SUN-RPC-PROGRAM-NFS	100003 100227	Sun RPC Network File System
SUN-RPC-PROGRAM-NLOCKMGR	100021	Sun RPC Network Lock Manager

Table 14: Remote Procedure Call Application Layer Gateway Services *(continued)*

Service	Program Numbers	Full Name
SUN-RPC-PROGRAM-RQUOTAD	100011	Sun RPC Remote Quota Daemon
SUN-RPC-PROGRAM-RSTATD	100001	Sun RPC Remote Status Daemon
SUN-RPC-PROGRAM-RUSERD	100002	Sun RPC Remote User Daemon
SUN-RPC-PROGRAM-SADMIND	100232	Sun RPC System Administration Daemon
SUN-RPC-PROGRAM-SPRAYD	100012	Sun RPC SPRAY Daemon
SUN-RPC-PROGRAM-STATUS	100024	Sun RPC STATUS
SUN-RPC-PROGRAM-WALLD	100008	Sun RPC WALL Daemon
SUN-RPC-PROGRAM-YPBIND	100007	SUN RPC Yellow Page Bind Service

Security and Tunnel Services

Table 15 on page 145 lists each supported service and gives the default port(s) and a description of each entry.

Table 15: Supported Protocol Services

Service	Port	Description
IKE	UDP source 1-65535; UDP destination 500 4500 (used for NAT traversal)	IKE is a protocol to obtain authenticated keying material for use with ISAKMP for. When configuring auto IKE, you can choose from three predefined Phase 1 or Phase 2 proposals: ■ Standard: AES and 3DES ■ Basic: DES and two different types of authentication algorithms ■ Compatible: four commonly used authentication and encryption algorithms
L2TP	1723	L2TP combines PPTP with Layer 2 Forwarding (L2F) for remote access.
PPTP	—	Point-to-Point Tunneling Protocol allows corporations to extend their own private network through private *tunnels* over the public Internet.

IP-Related Services

Table 16 on page 146 lists the predefined IP-related services. Each entry includes the default port and a description of the service.

Table 16: IP-Related Services

Service	Port	Description
Any	—	Any service
TCP-ANY	1-65535	Any protocol using the Transport Control Protocol TCPMUX port 1
UDP-ANY	137	Any protocol using the User Datagram Protocol

Instant Messaging Services

Table 17 on page 146 lists predefined Internet-messaging services. Each entry includes the name of the service, the default or assigned port, and a description of the service.

Table 17: Internet-Messaging Services

Service	Port	Description
Gnutella	6346 (default)	Gnutella File Sharing Protocol is a public domain file sharing protocol that operates over a distributed network. You can assign any port, but the default is 6346.
MSN	1863	Microsoft Network Messenger is a utility that allows you to send instant messages and talk online.
NNTP	119	Network News Transport Protocol is a protocol used to post, distribute, and retrieve USENET messages.
SMB	445	Server Message Block (SMB) Protocol over IP allows you to read and write files to a server on a network.
YMSG	5010	Yahoo! Messenger is a utility that allows you to check when others are online, send instant messages, and talk online.

Management Services

Table 18 on page 146 lists the predefined management services. Each entry includes the name of the service, the default or assigned port, and a description of the service.

Table 18: Management Services

Service	Port	Description
NBNAME	137	NetBIOS Name Service displays all NetBIOS name packets sent on UDP port 137.
NDBDS	138	NetBIOS Datagram Service, published by IBM, provides connectionless (datagram) services to PCs connected with a broadcast medium to locate resources, initiate sessions and terminate sessions. It is unreliable and the packets are not sequenced.

Table 18: Management Services *(continued)*

Service	Port	Description
NFS	—	Network File System uses UDP to allow network users to access shared files stored on computers of different types. SUN RPC is a building block of NFS.
NS Global	—	NS-Global is the central management protocol for Juniper Networks Firewall/VPN devices.
NS Global PRO	—	NS Global-PRO is the scalable monitoring system for the Juniper Networks Firewall/VPN device family.
NSM	—	Network and Security Manager
NTP	123	Network Time Protocol provides a way for computers to synchronize to a time referent.
RLOGIN	513	RLOGIN starts a terminal session on a remote host.
RSH	514	RSH executes a shell command on a remote host.
SNMP	161	Simple Network Management Protocol is a set of protocols for managing complex networks.
SQL*Net V1	66	SQL*Net Version 1 is a database language that allows for the creation, access, modification, and protection of data.
SQL*Net V2	66	SQL*Net Version 2 is a database language that allows for the creation, access, modification, and protection of data.
MSSQL	1433 (default instance)	Microsoft SQL is a proprietary database server tool that allows for the creation, access, modification, and protection of data.
SSH	22	Secure Shell is a program to log into another computer over a network through strong authentication and secure communications on an unsecure channel.
SYSLOG	514	Syslog is a UNIX program that sends messages to the system logger.
Talk	517-518	Talk is a visual communication program that copies lines from your terminal to that of another user.
Telnet	23	Telnet is a UNIX program that provides a standard method of interfacing terminal devices and terminal-oriented processes to each other.
WinFrame	—	WinFrame is a technology that allows users on non-Windows machines to run Windows applications.
X-Windows	—	X-Windows is the windowing and graphics system that Motif and OpenLook are based on.

Mail Services

Table 19 on page 148 lists the predefined mail services. Each includes the name of the service, the default or assigned port number, and a description of the service.

Table 19: Mail Services

Service	Port	Description
IMAP	143	Internet Message Access Protocol is a protocol used for retrieving messages.
Mail (SMTP)	25	Simple Mail Transfer Protocol is a protocol for sending messages between servers.
POP3	110	Post office protocol is a protocol used for retrieving email.

UNIX Services

Table 20 on page 148 lists the predefined UNIX services. Each entry includes the name of the service, the default or assigned port, and a description of the service.

Table 20: UNIX Services

Service	Port	Description
FINGER	79	Finger is a UNIX program that provides information about the users.
UUCP	117	Unix-to-Unix Copy Protocol (UUCP) is a UNIX utility that enables file transfers between two computers over a direct serial or modem connection.

Miscellaneous Services

Table 21 on page 148 lists predefined miscellaneous services. Each entry includes the service name, default or assigned port, and a description of the service.

Table 21: Miscellaneous Services

Service	Port	Description
CHARGEN	19	Character Generator Protocol is a UDP or TCP-based debugging and measurement tool.
DISCARD	9	Discard Protocol is an application layer protocol that describes a process for discarding TCP or UDP data sent to port 9.
IDENT	113	Identification Protocol is a TCP/IP application layer protocol used for TCP client authentication.
LPR	515 listen; 721-731 source range (inclusive)	Line Printer Daemon Protocol is a TCP-based protocol used for printing services.
RADIUS	1812	Remote Authentication Dial-In User Service is a server program used for authentication and accounting purposes.
SQLMON	1434 (SQL Monitor Port)	SQL monitor (Microsoft)

Table 21: Miscellaneous Services *(continued)*

Service	Port	Description
VNC	5800	Virtual Network Computing facilitates viewing and interacting with another computer or mobile device connected to the Internet.
WHOIS	43	Network Directory Service Protocol is a way to look up domain names.
IPSEC-NAT	—	IPSEC-NAT allows Network Address Translation for ISAKMP and ESP packets.
SCCP	2000	Cisco Station Call Control Protocol uses the signaling connection control port (SCCP) to provide high availability and flow control.
VOIP	—	Voice over IP Service Group provides voice services over the Internet and includes H.323 and Session Initiation Protocol (SIP).

Custom Services

Instead of using predefined services, you can easily create custom services. You can assign each custom service the following attributes:

- Name
- Transport protocol
- Source and destination port numbers for services using TCP or UDP
- Type and code values for services using ICMP
- Timeout value

If you create a custom service in a virtual system (vsys) that has the same name as a previously defined custom service in the root system, the service in the vsys takes the default timeout for the specified transport protocol (TCP, UDP, or ICMP). To define a custom timeout for a service in a vsys that is different from the default when a custom service with the same name in the root system has its own timeout, create the custom service in the vsys and root system in the following order:

1. First, create the custom service with a custom timeout in the vsys.
2. Then create another custom service with the same name but a different timeout in the root system.

The following examples describe how to add, modify, and remove a custom service.

NOTE: For information regarding ScreenOS naming conventions—which apply to the names you create for custom services—see “Naming Conventions and Character Types” on page 11.

Adding a Custom Service

To add a custom service to the service book, you need the following information:

- A name for the service: in this example “cust-telnet.”
- A range of source port numbers: 1 – 65535.
- A range of destination port numbers to receive the service request: for example: 23000 – 23000.
- Whether the service uses TCP or UDP protocol, or some other protocol as defined by the Internet specifications. In this example, the protocol is TCP.

WebUI

Policy > Policy Elements > Services > Custom > New: Enter the following, then click **OK**:

Service Name: cust-telnet
Service Timeout: Custom (select), 30 (type)
Transport Protocol: TCP (select)
Source Port Low: 1
Source Port High: 65535
Destination Port Low: 23000
Destination Port High: 23000

CLI

```
set service cust-telnet protocol tcp src-port 1-65535 dst-port 23000-23000
set service cust-telnet timeout 30
save
```

NOTE: The timeout value is in minutes. If you do not set it, the timeout value of a custom service is 180 minutes. If you do not want a service to time out, enter **never**.

Modifying a Custom Service

In this example, you modify the custom service “cust-telnet” by changing the destination port range to 23230-23230.

Use the **set service** service_name **clear** command to remove the definition of a custom service without removing the service from the service book:

WebUI

Policy > Policy Elements > Services > Custom > Edit (for cust-telnet): Enter the following, then click **OK**:

Destination Port Low: 23230
Destination Port High: 23230

CLI

```
set service cust-telnet clear
set service cust-telnet + tcp src-port 1-65535 dst-port 23230-23230
```

```
save
```

Removing a Custom Service

In this example, you remove the custom service “cust-telnet” .

WebUI

Policy > Policy Elements > Services > Custom: Click **Remove** in the Configure column for “cust-telnet” .

CLI

```
unset service cust-telnet
save
```

Setting a Service Timeout

The timeout value you set for a service determines the session timeout. You can set the timeout for a predefined or custom service; you can use the service default timeout, specify a custom timeout, or use no timeout at all. Service timeout behavior is the same in virtual systems (vsys) security domains as at the root level.

Service Timeout Configuration and Lookup

Service timeout values are stored in the service entry database and in the corresponding vsys TCP and UDP port-based timeout tables. When you set a service timeout value, the security device updates these tables with the new value. There are also default timeout values in the services entry database, which are taken from predefined services. You can set a timeout but you cannot alter the default values.

Services with multiple rule entries share the same timeout value. If multiple services share the same protocol and destination port range, all services share the last timeout value configured.

For single service entries, service timeout lookup proceeds as follows:

1. The specified timeout in the service entry database, if set.
2. The default timeout in the service entry database, if specified in the predefined service.
3. The protocol-based default timeout table.

Table 22: Protocol-Based Default Timeout Table

Protocol	Default Timeout (minutes)
TCP	30
UDP	1
ICMP	1

Table 22: Protocol-Based Default Timeout Table *(continued)*

Protocol	Default Timeout (minutes)
OSPF	1
Other	30

For service groups, including hidden groups created in multi-cell policy configurations, and for the predefined service "ANY" (if timeout is not set), service timeout lookup proceeds as follows:

1. The vsys TCP and UDP port-based timeout table, if a timeout is set.
2. The protocol-based default timeout table.

Contingencies

When setting timeouts, be aware of the following:

- If a service contains several service rule entries, all rule entries share the same timeout. The timeout table is updated for each rule entry that matches the protocol (for UDP and TCP—other protocols use the default). You need define the service timeout only once. For example, if you create a service with two rules, the following commands will set the timeout to 20 minutes for both rules:

  ```
  set service test protocol tcp dst-port 1035-1035 timeout 20
  set service test + udp src-port 1-65535 dst-port 1111-1111
  ```

- If multiple services are configured with the same protocol and overlapping destination ports, the latest service timeout configured overrides the others in the port-based table. For example:

  ```
  set service ftp-1 protocol tcp src 0-65535 dst 2121-2121 timeout 10
  set service telnet-1 protocol tcp src 0-65535 dst 2100-2148 timeout 20
  ```

  With this configuration, the security device applies the 20-minute timeout for destination port 2121 in a service group, because the destination port numbers for telnet-1 (2100-2148) overlap those for ftp-1 (2121), and you defined telnet-1 after you defined ftp-1.

  To modify a service timeout when multiple services use the same protocol and an overlapping destination port range, you must unset the service and reset it with the new timeout value. This is because, during reboot, services are loaded according to creation time, not modification time.

  To avoid the unintended application of the wrong timeout to a service, do not create services with overlapping destination port numbers.

- If you unset a service timeout, the default protocol-based timeout in the service entry database is used, and the timeout values in both the service entry and port-based timeout tables are updated with the default value.

If the modified service has overlapping destination ports with other services, the default protocol-based timeout might not be the desired value. In that case, reboot the security device, or set the service timeout again for the desired timeout to take effect.

- When you modify a predefined service and reboot, the modified service might not be the last one in the configuration. This is because predefined services are loaded before custom services, and any change made to a custom service, even if made earlier, will show as the later than the predefined service change when you reboot.

  For example, if you create the following service:

  ```
  set service my_service protocol tcp dst-port 179-179 timeout 60
  ```

  and later modify the timeout of the predefine service BGP as follows:

  ```
  set service bgp timeout 75
  ```

  the BGP service will use the 75-minute timeout value, because it is now written to the service entry database. But the timeout for port 179, the port BGP uses, is also changed to 75 in the TCP port-based timeout table. After you reboot, the BGP service will continue to use the 75-minute timeout which, as a single service, it gets from the service entry database. But the timeout in the TCP port-based table for port 179 will now be 60. You can verify this by entering the **get service bgp** command.

  This has no effect on single services. But if you add BGP or my_service to a service group, the 60-minute timeout value will be used for destination port 179. This is because service group timeout is taken from the port-based timeout table, if one is set.

  To ensure predictability when you modify a predefine service timeout, therefore, you can create a similar service, for example:

  ```
  set service my-bgp protocol tcp dst-port 179-179 timeout 75
  ```

Example

In the following example, you change the timeout threshold for the FTP predefined service to 75 minutes:

WebUI

Policy > Policy Elements > Services > Predefined > Edit (FTP): Enter the following, then click **OK**:

Service Timeout: Custom (select), 75 (type)

CLI

```
set service FTP timeout 75
save
```

Defining a Custom Internet Control Message Protocol Service

ScreenOS supports Internet Control Message Protocol (ICMP) as well as several ICMP messages, as predefined or custom services. When configuring a custom ICMP service, you must define a type and code. There are different message types within ICMP. For example:

type 0 = Echo Request message

type 3 = Destination Unreachable message

NOTE: For more information about ICMP types and codes, see RFC 792, *Internet Control Message Protocol*.

An ICMP message type can also have a message code. The code provides more specific information about the message, as shown in Table 23 on page 154.

Table 23: Message Descriptions

Message Type	Message Code
5 = Redirect	0 = Redirect Datagram for the Network (or subnet)
	1 = Redirect Datagram for the Host
	2 = Redirect Datagram for the Type of Service and Network
	3 = Redirect Datagram for the Type of Service and Host
11 = Time Exceeded Codes	0 = Time to Live exceeded in Transit
	1 = Fragment Reassembly Time Exceeded

ScreenOS supports any type or code within the 0–255 range.

In this example, you define a custom service named “host-unreachable” using ICMP as the transport protocol. The type is 3 (for Destination Unreachable) and the code is 1 (for Host Unreachable). You set the timeout value at 2 minutes.

WebUI

Policy > Policy Elements > Services > Custom: Enter the following, then click **OK**:

Service Name: host-unreachable
Service Timeout: Custom (select), 2 (type)
Transport Protocol: ICMP (select)
ICMP Type: 3

ICMP Code: 1

CLI

```
set service host-unreachable protocol icmp type 5 code 0
set service host-unreachable timeout 2
save
```

Remote Shell Application Layer Gateway

Remote Shell Application Layer Gateway (RSH ALG) allows authenticated users to run shell commands on remote hosts. Juniper Networks security devices support the RSH service in transparent (Layer 2), route (Layer 3), and NAT modes, but the devices do not support port translation of RSH traffic.

Sun Remote Procedure Call Application Layer Gateway

Sun remote procedure call (RPC)—also known as Open Network Computing (ONC) RPC—provides a way for a program running on one host to call procedures in a program running on another host. Because of the large number of RPC services the transport address of an RPC service is dynamically negotiated based on the service's program number and version number. Several binding protocols are defined for mapping the RPC program number and version number to a transport address.

Juniper Networks security devices support Sun RPC as a predefined service and allow traffic based on a policy you configure. The Sun RPC Application Layer Gateway (ALG) provides the functionality for security devices to handle the dynamic transport address negotiation mechanism of Sun RPC and to ensure program number-based firewall policy enforcement. You can define a firewall policy to permit all RPC requests or to permit according to a specific program number.

Although the ALG for IPv4 supports route and NAT modes for incoming and outgoing requests, the IPv6 ALG does not support NAT, NAT-Protocol Translation (NAT-PT), or transparent mode. In addition, a TCP segment in a Sun RPC stream might be fragmented, so it might not include an intact Sun RPC protocol data unit (PDU). Such fragmentation occurs in the RPC layer; so the security device does not support parsing a fragmented packet coming through a Sun RPC ALG over IPv4 and IPv6. In addition, the IPv6 ALG does not support multicast CALLIT packets.

NOTE: The Sun RPC ALG for IPv6 supports Netscreen Redundancy Protocol (NSRP).

Typical RPC Call Scenario

When a client calls a remote service, it needs to find the transport address of the service—in the case of TCP/UDP, this is a port number. A typical procedure for this case is as follows:

1. The client sends the GETPORT message to the RPCBIND service on the remote machine. The GETPORT message contains the program number, and version and procedure number of the remote service it wants to call.
2. The RPCBIND service replies with a port number.
3. The client calls the remote service using the port number returned.
4. The remote service replies to the client.

A client also can use the CALLIT message to call the remote service directly, without knowing the port number of the service. In this case, the procedure is as follows:

1. The client sends a CALLIT message to the RPCBIND service on the remote machine. The CALLIT message contains the program number, and the version and procedure number of the remote service it wants to call.
2. RPCBIND calls the service for the client.
3. RCPBIND replies to the client if the call has been successful. The reply contains the call result and the services's port number.

Customizing Sun RPC Services

Because Sun RPC services use dynamically negotiated ports, you cannot use regular service objects based on fixed TCP/UDP ports to permit them in a security policy. Instead, you must create Sun RPC service objects using program numbers. For example, the Sun RPC network file system (NFS) uses two program numbers: 100003 and 100227. The corresponding TCP/UDP ports are dynamic. In order to permit the program numbers, you create a sun-rpc-nfs service object that contains these two numbers. The ALG maps the program numbers into dynamically negotiated TCP/UDP ports and permits or denies the service based on a policy you configure.

In this example, you create a service object called my-sunrpc-nfs to use the Sun RPC NFS, which is identified by two program IDs: 100003 and 100227.

WebUI

Policy > Policy Elements > Services > Sun RPC Services > New: Enter the following, then click **Apply**:

```
Service Name: my-sunrpc-nfs
Service Timeout: (select)
Program ID Low: 100003
Program ID High: 100003
Program ID Low: 100227
Program ID High: 100227
```

CLI

```
set service my-sunrpc-nfs protocol sun-rpc program 100003-100003
set service my-sunrpc-nfs + sun-rpc program 100227-100227
save
```

Customizing Microsoft Remote Procedure Call Application Layer Gateway

Microsoft remote procedure call (MS RPC) is the Microsoft implementation of the Distributed Computing Environment (DCE) RPC. Like the Sun RPC (see "Sun Remote Procedure Call Application Layer Gateway" on page 155), MS RPC provides a way for a program running on one host to call procedures in a program running on another host. Because of the large number of RPC services, the transport address of an RPC service is dynamically negotiated based on the service program's Universal Unique IDentifier (UUID). The Endpoint Mapper (EPM) binding protocol is defined in ScreenOS to map the specific UUID to a transport address.

Juniper Networks security devices support MS RPC as a predefined service; they allow traffic based on a policy you configure. The ALG provides the functionality for security devices to handle the dynamic transport address negotiation mechanism of MS RPC and to ensure UUID-based firewall policy enforcement. You can define a firewall policy to permit all RPC requests or to permit by specific UUID number. While the ALG for IPv4 supports route and NAT modes for incoming and outgoing requests, the IPv6 ALG does not support NAT, NAT-PT, or transparent mode.

In addition, because a TCP segment in an MS RPC stream might be fragmented, it might not include an intact MS RPC PDU. Such fragmentation occurs in the RPC layer; so the security device does not support parsing a fragmented packet coming through an MS RPC ALG over IPv4 and IPv6.

The MS RPC ALG for IPv6 does not support the Endpoint Mapper protocol over UDP, but it does supports NSRP.

Because MS RPC services use dynamically negotiated ports, you cannot use regular service objects based on fixed TCP/UDP ports to permit them in a security policy. Instead, you must create MS RPC service objects using UUIDs. The MS Exchange Info Store service, for example, uses the following four UUIDs:

- 0e4a0156-dd5d-11d2-8c2f-00c04fb6bcde
- 1453c42c-0fa6-11d2-a910-00c04f990f3b
- 10f24e8e-0fa6-11d2-a910-00c04f990f3b
- 1544f5e0-613c-11d1-93df-00c04fd7bd09

The corresponding TCP/UDP ports are dynamic. To permit them, you create an **ms-exchange-info-store** service object that contains these four UUIDs. The ALG maps the program numbers into dynamically negotiated TCP/UDP ports based on these four UUIDs and permits or denies the service based on a policy you configure.

In this example, you create a service object called **my-ex-info-store** that includes the UUIDs for the MS Exchange Info Store service.

WebUI

Policy > Policy Elements > Services > MS RPC: Enter the following, then click **Apply**:

```
Service Name: my-ex-info-store
UUID: 0e4a0156-dd5d-11d2-8c2f-00c04fb6bcde
UUID: 1453c42c-0fa6-11d2-a910-00c04f990f3b
UUID: 10f24e8e-0fa6-11d2-a910-00c04f990f3b
UUID: 1544f5e0-613c-11d1-93df-00c04fd7bd09
```

CLI

```
set service my-ex-info-store protocol ms-rpc uuid
0e4a0156-dd5d-11d2-8c2f-00c04fb6bcde
set service my-ex-info-store + ms-rpc uuid 1453c42c-0fa6-11d2-a910-00c04f990f3b
set service my-ex-info-store + ms-rpc uuid 10f24e8e-0fa6-11d2-a910-00c04f990f3b
set service my-ex-info-store + ms-rpc uuid 1544f5e0-613c-11d1-93df-00c04fd7bd09
save
```

Real-Time Streaming Protocol Application Layer Gateway

Real-Time Streaming Protocol (RTSP) is an application layer protocol used to control delivery of one or more synchronized streams of multimedia, such as audio and video. Although RTSP is capable of delivering the data streams itself—interleaving continuous media streams with the control stream—it is more typically used as a kind of network remote control for multimedia servers. The protocol was designed as a means for selecting delivery channels, such as User Datagram Protocol (UDP), multicast UDP, and TCP, and for selecting a delivery mechanism based on Real-Time Protocol (RTP). RTSP may also use the Session Description Protocol (SDP) as a means of providing information to the client for aggregate control of a presentation consisting of streams from one or more servers, and non-aggregate control of a presentation consisting of multiple streams from a single server. The data sources can be live feeds or stored clips.

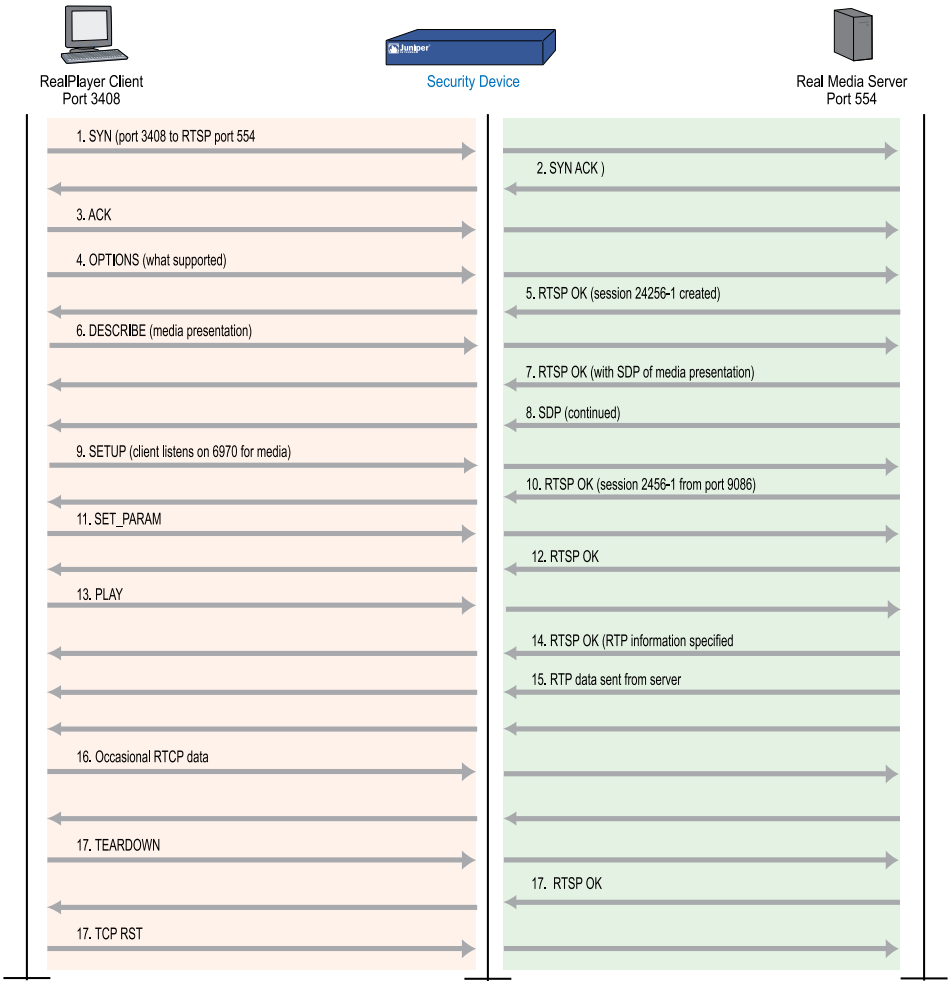
Juniper Networks security devices support RTSP as a service and allow or deny RTSP traffic based on the policy you configure. The RTSP Application Layer Gateway (ALG) is needed because RTSP uses dynamically assigned port numbers that are conveyed in the packet payload when endpoints establish a control connection. The ALG keeps track of the dynamically assigned port numbers and opens pinholes on the security device accordingly. In Network Address Translation (NAT) mode, the ALG translates IP addresses and ports as necessary. Security devices support RTSP in route and transparent modes and in both interface-based and policy-based NAT mode.

Figure 50 on page 160 illustrates a typical RTSP session. The client initiates the session (when the user clicks the Play button in a RealPlayer application, for example) and establishes a TCP connection to the RTSP server on port 554, then sends the OPTIONS message (messages are also called methods), to find out what audio and video features the server supports. The server responds to the OPTIONS message by specifying the name and version of the server, and a session identifier, for example, 24256-1. (For more information about methods, see “SIP Request Methods” on page 1106, and see RFC 2326, Section 11.)

The client then sends the DESCRIBE message with the URL of the actual media file the client wants. The server responds to the DESCRIBE message with a description of the media in SDP format. The client then sends the SETUP message, which specifies the transport mechanisms acceptable to the client for streamed media, for example RTP/RTCP or RDT, and the ports on which it receives the media. When using NAT,

the RTSP ALG keeps track of these ports and translates them as necessary. The server responds to the SETUP message and selects one of the transport protocols, and, in this way, both client and server agree on a mechanism for media transport. The client then sends the PLAY message, and the server begins streaming the media to the client.

Figure 50: Typical RTSP Session


RealPlayer Client
Port 3408
Security Device
Real Media Server
Port 554
1. SYN (port 3408 to RTSP port 554
2. SYN ACK)
3. ACK
4. OPTIONS (what supported)
5. RTSP OK (session 24256-1 created)
6. DESCRIBE (media presentation)
7. RTSP OK (with SDP of media presentation)
8. SDP (continued)
9. SETUP (client listens on 6970 for media)
10. RTSP OK (session 2456-1 from port 9086)
11. SET_PARAM
12. RTSP OK
13. PLAY
14. RTSP OK (RTP information specified
15. RTP data sent from server
16. Occasional RTCP data
17. TEARDOWN
17. RTSP OK
17. TCP RST

Dual-Stack Environment

You enable the RTSP ALG by using the **set alg rtsp enable** command. When you use this command to enable ALG in a dual-stack environment, the IPv4 and IPv6 RTSP ALGs are enabled at the same time. This feature has the following limitations:

- The ALG does not support NAT for IPv6.
- The ALG does not support transparent mode for IPv6.

NOTE: The ALG for IPv6 now supports Netscreen Redundancy Protocol (NSRP).

RTSP Request Methods

Table 24 on page 162 lists methods that can be performed on a resource (media object), the direction or directions in which information flows, and whether the method is required, recommended, or optional. Presentation refers to information such as network addresses, encoding, and content about a set of one or more streams presented to the client as a complete media feed. A Stream is a single media instance, for example audio or video, as well as all packets created by a source within the session.

Table 24: RTSP Request Methods

Method	Direction	Object	Requirement	Description
OPTIONS	Client to Server	Presentation, Stream	Client to Server required	Client queries the server about what audio or video features it supports, as well as such things as the name and version of the server, and session ID.
	Server to Client	Presentation, Stream	Server to Client optional	
DESCRIBE	Client to Server	Presentation, Stream	Recommended	For exchange of media initialization information, such as clock rates, color tables, and any transport-independent information the client needs for playback of the media stream. Typically the client sends the URL of the of file it is requesting, and the server responds with a description of the media in SDP format.
ANNOUNCE	Client to Server	Presentation, Stream	Optional	Client uses this method to post a description of the presentation or media object identified by the request URL. The server uses this method to update the session description in real-time.
	Server to Client	Presentation, Stream		
SETUP	Client to Server	Stream	Required	Client specifies acceptable transport mechanisms to be used, such as the ports on which it will receive the media stream, and the transport protocol.

Table 24: RTSP Request Methods *(continued)*

Method	Direction	Object	Requirement	Description
GET_PARAMETER	Client to Server	Presentation, Stream	Optional	Retrieves the value of a presentation or stream parameter specified in the URL. This method can be used with no entity body to test client or server aliveness. Ping can also be used to test for aliveness.
	Server to Client			
SET_PARAMETER	Client to Server	Presentation, Stream	Optional	Client uses this method to set the value of a parameter for a presentation or stream specified by the URI. Due to firewall considerations, this method cannot be used to set transport parameters.
	Server to Client			
PLAY	Client to Server	Presentation, Stream	Required	Instructs the server to begin sending data using the mechanism specified in SETUP. The Client does not issue PLAY requests until all SETUP requests are successful. The server queues PLAY requests in order, and delays executing any new PLAY request until an active PLAY request is completed. PLAY requests may or may not contain a specified range. The range may contain a time parameter—specified in Coordinated Universal Time (UTC)—for start of playback, which can also be used to synchronize streams from different sources.
PAUSE	Client to Server	Presentation, Stream	Recommended	Temporarily halts delivery of an active presentation. If the request URL specifies a particular stream, for example audio, this is equivalent to muting. Synchronization of tracks is maintained when playback or recording is resumed, although servers may close the session if PAUSE is for the duration specified in the timeout parameter in SETUP. A PAUSE request discards all queued PLAY requests.
RECORD	Client to Server	Presentation, Stream	Optional	Initiates recording a range of media defined in the presentation description. A UTC timestamp indicates start and end times, otherwise the server uses the start and end times in the presentation description.
REDIRECT	Server to Client	Presentation, Stream	Optional	Informs the client it must connect to a different server, and contains location information and possibly a range parameter for that new URL. To continue to receive media for this URL, the client must issue a TEARDOWN request for the current session and a SETUP for the new session.
TEARDOWN	Client to Server	Presentation, Stream	Required	Stops stream delivery for the given URI and frees the resources associated with it. Unless all transport parameters are defined by the session description, a SETUP request must be issued before the session can be played again.

RTSP Status Codes

RTSP uses status codes to provide information about client and server requests. Status codes include a machine-readable three digit result code, and a human-readable reason phrase. It is at the client's discretion whether to display the reason phrase. Status codes are described in Table 25 on page 164.

Table 25: RSTP Status Codes

Code	Number	Description
Informational	100 to 199	Request has been received and is being processed
Success	200 to 299	Action has been received successfully, understood, and accepted
Redirection	300 to 399	Further action is necessary to complete the request
Client Error	400 to 499	Request contains bad syntax and cannot be fulfilled
Server Error	500 to 599	Server failed to fulfill an apparently valid request

Table 26 on page 164 lists all status codes defined for RTSP 1.0, and recommended reason phrases. Reason phrases can be revised or redefined without affecting the operation of the protocol.

Table 26: RTSP 1.0 Status Codes

Status Code	Reason Phrase	Status Code	Reason Phrase
100	Continue	414	Request-URI Too Large
200	OK	415	Unsupported Media Type
201	Created	451	Unsupported Media Type
250	Low on Storage Space	452	Conference Not Found
300	Multiple Choices	453	Not Enough Bandwidth
301	Moved Permanently	454	Session Not Found
303	See Other	455	Method Not Valid in This State
304	Not Modified	456	Header Field Not Valid for Resource
305	Use Proxy	457	Invalid Range
400	Bad Request	458	Parameter is Read-Only
401	Unauthorized	459	Aggregate operation not allowed

Table 26: RTSP 1.0 Status Codes *(continued)*

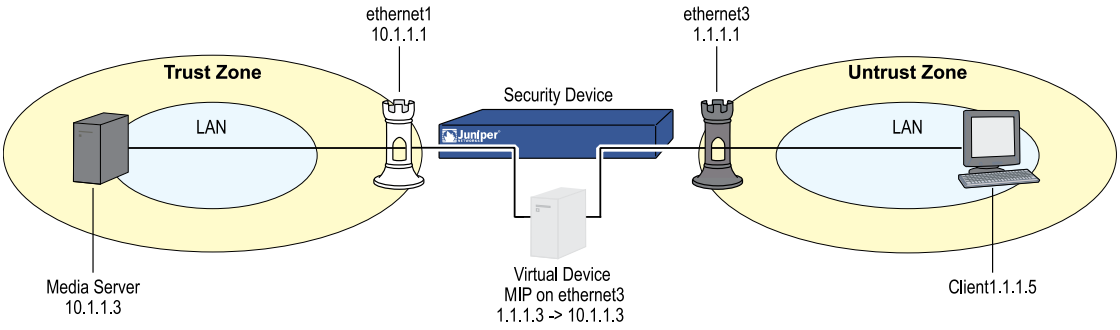
Status Code	Reason Phrase	Status Code	Reason Phrase
402	Payment Required	460	Only aggregate operation allowed
403	Forbidden	461	Unsupported transport
404	Not Found	462	Destination unreachable
405	Method Not Allowed	500	Internal Server Error
406	Not Acceptable	501	Not Implemented
407	Proxy Authentication Required	502	Bad Gateway
408	Request Time-out	503	Service Unavailable
410	Gone	504	Gateway Time-out
411	Length Required	505	RTSP Version not supported
412	Precondition Failed	551	Option not supported
413	Request Entity Too Large		

NOTE: For complete definitions of status codes, see RFC 2326, *Real Time Streaming Protocol (RTSP)*.

Configuring a Media Server in a Private Domain

In this example, the media server is in the Trust zone and the client is in the Untrust zone. You put a MIP on the ethernet0/3 interface to the media server in the Trust zone, then create a policy to allow RTSP traffic to flow from the client in the Untrust zone to the media server in the Trust zone.

Figure 51: RTSP Private Domain



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Manage IP: 10.1.1.2

   Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **Apply**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24
   Manage IP: 1.1.1.2

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: media_server
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.3/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: client
   IP Address/Domain Name:
   IP/Netmask: (select), 1.1.1.5/24
   Zone: Untrust

3. **MIP**

   Network > Interfaces > Edit (for ethernet0/3) > MIP > New: Enter the following, then click **OK**:

   Mapped IP: 1.1.1.3
   Host IP Address: 10.1.1.5

4. **Policy**

   Policy > Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), client
   Destination Address:

Address Book Entry: (select), MIP(1.1.1.3)
Service: RTSP
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet0/1 trust
set interface ethernet0/1 ip 10.1.1.1
set interface ethernet0/3 untrust
set interface ethernet0/3 ip 1.1.1.1
```

2. **Addresses**

```
set address trust media_server 10.1.1.3/24
set address untrust client 1.1.1.5
```

3. **MIP**

```
set interface ethernet0/3 mip (1.1.1.3) host 10.1.1.3
```

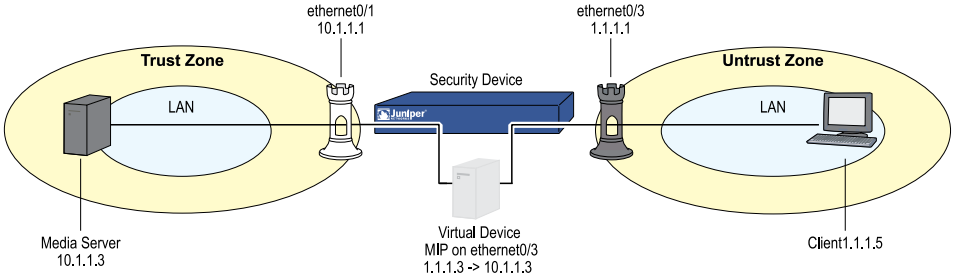
4. **Policy**

```
set policy from untrust to trust client mip(1.1.1.3) rtsp permit
save
```

Configuring a Media Server in a Public Domain

In this example, the media server is in the Untrust zone and the client is in the Trust zone. You put a DIP pool on the ethernet0/3 interface to do NAT when the media server to responds to the client from the Untrust zone, then create a policy to allow RTSP traffic to flow from the Trust to the Untrust zone.

Figure 52: RTSP Public Domain


ethernet0/1
10.1.1.1
ethernet0/3
1.1.1.1
Trust Zone
Security Device
Untrust Zone
LAN
LAN
Media Server
10.1.1.3
Virtual Device
MIP on ethernet0/3
1.1.1.3 -> 10.1.1.3
Client1.1.1.5

WebUI

1. **Interface**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Manage IP: 10.1.1.2

   Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **Apply**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24
   Manage IP: 1.1.1.2

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: client
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.3/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: media_server
   IP Address/Domain Name:
   IP/Netmask: (select), 1.1.1.3/24
   Zone: Untrust

3. **DIP Pool**

   Network > Interfaces > Edit (for ethernet0/3) > DIP > New: Enter the following, then click **OK**:

   ID: 5
   IP Address Range: (select) 1.1.1.5 ~ 1.1.1.50
   Port Translation: (select)

4. **Policy**

   Policy > Policies > (From: Trust, To: Untrust) > New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry (select): client

Destination Address:
Address Book Entry (select): media_server
Service: RTSP
Action: Permit

> Advanced: Enter the following, then click **OK**:

NAT:
Source Translation: (select)
(DIP on): 5 (1.1.1.5-1.1.1.50)/port-xlate

CLI

1. **Interface**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 10.1.1.1
set interface ethernet0/3 zone untrust
set interface ethernet0/3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address trust client ip 10.1.1.3/24
set address untrust media_server ip 1.1.1.3/24
```

3. **DIP Pool**

```
set interface ethernet0/3 dip 5 1.1.5 1.1.1.50
```

4. **Policy**

```
set policy from trust to untrust client media_server rtsp nat dip 5 permit
save
```

Stream Control Transmission Protocol Application Layer Gateway

Stream Control Transmission Protocol (SCTP) is a Transport Layer protocol that provides a reliable transport service that supports data transfer across the network, in sequence and without errors. In transporting signaling messages to and from Signaling System 7 (SS7) gateways, SCTP provides advantages over the following network-protocol configurations:

- Transport Control Protocol (TCP) for SS7 for 3G mobile networks
- Session Initiation Protocol (SIP) for Voice-over-Internet Protocol (VoIP) networks

SCTP is effective when used as the transport protocol for applications that require monitoring and session-loss detection. For such applications, the SCTP path/session failure-detection mechanisms, especially the heartbeat, actively monitor the connectivity of the session. SCTP differs from TCP in having multi-homing capabilities at either or both ends and several streams within a connection, typically referred to an association. A TCP stream represents a sequence of bytes; an SCTP stream represents a sequence of messages.

You can configure the security device to perform stateful inspection on all SCTP traffic without performing Deep Inspection (DI). If you enable stateful inspection of SCTP traffic, the SCTP ALG drops any anomalous SCTP packets.

SCTP Protocol Filtering

Beginning with ScreenOS 6.3.0, the existing SCTP stateful firewall supports protocol filtering.

NOTE: SCTP protocol filtering is effective only when the SCTP ALG is enabled.

Protocol filtering is enabled by the SCTP payload protocol settings. SCTP payload protocol enables the following:

- Identifying the data type being carried out by the SCTP data chunk
- Creating an SCTP profile
- Configuring the security device to permit or deny traffic
- Binding the new SCTP profile to a policy

NOTE: ScreenOS supports SCTP protocol filtering on NS5000 and ISG security devices only.

Point-to-Point Tunneling Protocol Application Layer Gateway

Juniper Networks security devices support port address translation (PAT) for Point-to-Point Tunneling Protocol (PPTP) traffic. PPTP provides IP security at the Network Layer. PPTP consists of a control connection and a data tunnel—the control connection runs over TCP and helps in establishing and disconnecting calls, and the data tunnel handles encapsulated Point-to-Point Protocol (PPP) packets carried over IP.

PPTP uses TCP port 1723 for its control connection and Generic Routing Encapsulation (GRE - IP protocol 47) for tunneling the encapsulated PPP data. The GRE traffic carries no port number, making it difficult for the security device to distinguish between two clients with the same public IP. PPTP uses the source IP address and the Call ID field in the GRE header to identify a tunnel. When multiple clients sharing the same public IP address establish tunnels with the same PPTP server, they may get the same Call ID. You can translate the Call ID value in both the control message and the data traffic, but only when the client is in a private network and the server is in a public network.

PPTP clients can either directly connect to the Internet or dial into a network access server (ISP) to reach the Internet. The security device that protects the PPTP clients can translate the clients' private IP addresses to a pool of public IP addresses using Network Address Translation-Port Translation (NAT-PT). Because the GRE traffic carries no port number for address translation, the PPTP ALG treats the Call ID field as a port number as a way of distinguishing multiple clients.

After the PPTP client establishes a TCP connection with the PPTP server, the client sends a Start Control Connection Request message to establish a control connection with the server. The server replies with a Start Control Connection Reply message. The client then sends a request to establish a call and sends an Outgoing Call Request message. The security device assigns a Call ID (bytes 12-13 of the control message) that is unique to the tunnel. The server replies with an Outgoing Call Reply message, which carries its own Call ID (bytes 12-13) and the client's Call ID (bytes 14-15). The PPTP ALG parses the control connection messages for the Call ID to identify the call to which a particular PPP packet belongs. The ALG identifies an Outgoing Call Request message using the Control Message Type field (bytes 8-9) with the value 7. When the ALG receives this message, it parses the control message for the Call ID field (bytes 12-13). The security device translates the Call ID so that it is unique across multiple calls from the same translated client IP. After receiving Outgoing Call Response message, the ALG holds this message and opens a pinhole in order to accept GRE traffic that the PPTP server sends. An Outgoing Call Request message contains the following elements:

- Protocol used for the outgoing call request message (GRE)
- Source IP address (PPTP server IP)
- Destination IP address (translated client IP)
- Destination port number (translated client Call ID)

The ALG identifies an Outgoing Call Reply message using the Control Message Type field (bytes 8-9) with the value 8. The ALG parses these control messages for the Call ID field (bytes 12-13) and the client's Call ID (bytes 14-15). The ALG then uses the client's Call ID value to find the mapping created for the other direction, and then opens a pinhole to accept the GRE traffic that the client sends. An Outgoing Call Reply message contains the following elements:

- Protocol used for the outgoing call reply message (GRE)
- Source IP address (PPTP client IP)
- Destination IP address (PPTP server IP)
- Destination port number (PPTP server Call ID)

Each pinhole that the ALG opens creates a session for data traffic arriving in that direction. The ALG opens two data sessions for each tunnel:

- Traffic from the PPTP client to the server, using the server's Call ID as the destination port
- Traffic from the PPTP server to the client, using the client's translated Call ID as the destination port

The default timeout value of the control connection is 30 minutes. The ALG closes the pinhole when the data session exceeds the timeout value or is idle for long time. When you close the control session through the ALG, the security device closes all control connections and data sessions.

NOTE: Because the PPTP ALG requires Port Address Translation (PAT), you cannot create a hardware session for the data traffic that uses GRE, because the hardware does not support encapsulation using GRE. On such Juniper Networks security devices, ScreenOS must handle both the control traffic and the data traffic in software path.

Call ID translation is not required when the client is on the public network and the server is on the private side. In this case, no PAT is involved and the Call ID value between any client–server pair is unique.

Configuring the PPTP ALG

You can enable the PPTP ALG using the WebUI or the CLI.

WebUI

Security > ALG > Basic: Select the PPTP check box, then click **OK**.

CLI

```
set alg pptp enable
```

Service Groups

A service group is a set of services that you have gathered together under one name. After you create a group containing several services, you can then apply services at the group level to policies, thus simplifying administration.

The ScreenOS service group option has the following features:

- Each service book entry can be referenced by one or more service groups.
- Each service group can contain predefined and user-defined service book entries.

Service groups are subject to the following limitations:

- Service groups cannot have the same names as services; therefore, if you have a service named “FTP,” you cannot have a service group named “ FTP.”
- If a service group is referenced in a policy, you can edit the group but you cannot remove it until you have first removed the reference to it in the policy.
- If a custom service book entry is deleted from the service book, the entry is also removed from all the groups in which it was referenced.
- One service group cannot contain another service group as a member.
- The all-inclusive service term “ANY” cannot be added to groups.
- A service can be part of only one group at a time.

Creating a Service Group

In this example, you create a service group named grp1 that includes IKE, FTP, and LDAP services.

WebUI

Policy > Policy Elements > Services > Groups > New: Enter the following group name, move the following services, then click **OK**:

Group Name: grp1

Select **IKE** and use the < < button to move the service from the Available Members column to the Group Members column.

Select **FTP** and use the < < button to move the service from the Available Members column to the Group Members column.

Select **LDAP** and use the < < button to move the service from the Available Members column to the Group Members column.

CLI

```
set group service grp1
set group service grp1 add ike
set group service grp1 add ftp
set group service grp1 add ldap
save
```

NOTE: If you try to add a service to a service group that does not exist, the security device creates the group. Also, ensure that groups referencing other groups do not include themselves in the reference list.

Modifying a Service Group

In this example, you change the members in the service group named grp1 that you created in "Creating a Service Group" on page 175. You remove IKE, FTP, and LDAP services, and add HTTP, FINGER, and IMAP.

WebUI

Policy > Policy Elements > Services > Groups > Edit (for grp1): Move the following services, then click **OK**:

Select **IKE** and use the > > button to move the service from the Group Members column to the Available Members column.

Select **FTP** and use the > > button to move the service from the Group Members column to the Available Members column.

Select **LDAP** and use the > > button to move the service from the Group Members column to the Available Members column.

Select **HTTP** and use the < < button to move the service from the Available Members column to the Group Members column.

Select **Finger** and use the < < button to move the service from the Available Members column to the Group Members column.

Select **IMAP** and use the < < button to move the service from the Available Members column to the Group Members column.

CLI

```
unset group service grp1 clear
set group service grp1 add http
set group service grp1 add finger
set group service grp1 add imap
save
```

Removing a Service Group

In this example, you delete the service group named **grp1**.

WebUI

Policy > Policy Elements > Services > Groups: Click **Remove** (for grp1).

CLI

```
unset group service grp1
save
```

NOTE: The security device does not automatically delete a group from which you have removed all members.

Creating a Session Cache to Accelerate HTTP Traffic

In ScreenOS, especially in high-end platforms, creating a session and aging out a session, are both time-consuming. Since HTTP connections are short-lived, the CPU utilization for HTTP traffic is higher than other TCP traffic under the same traffic load. To accelerate HTTP traffic, you have to either speed up creating a session or make session aging-out faster. You can optimize the session-creation stage, because most of the elements for creating a session cache--source IP, destination IP and port, policy, and protocol--are fixed or semi-fixed.

Beginning with ScreenOS 6.3.0, you can create a session cache for HTTP-based protocols to minimize CPU utilization and to enhance performance. A session cache is a special structure that caches (stores) all the reusable information of both software and hardware sessions created by the first connection of an HTTP session bundle.

When an HTTP SYN packet arrives, it looks for a session cache in the session cache table. If a session cache exists, a duplicate software or hardware session is created from the session cache. If a session cache does not exist, ScreenOS creates it for the first connection, then duplicates it for all subsequent connections.

NOTE: A session cache supports other traffic but does not ensure performance enhancement.

You cannot create a session cache for the following conditions:

- When the session is synched from another security device
- When the session is created by an Application Layer Gateway (ALG)

You can create a session cache for both predefined and custom services using the WebUI or the CLI.

In this example, you create a session cache for a predefined service named **AOL**.

WebUI

Policy > Policy Elements > Service > Predefined > Edit: Enter the following, then click OK:

```
Service Name: AOL
Enable Session Cache: (select)
```

Policy > Policy Elements > Service > Session Cache: Enter the following, then click Apply:

```
Enable Session Cache: (select)
Session Cache Count: 18
```

CLI

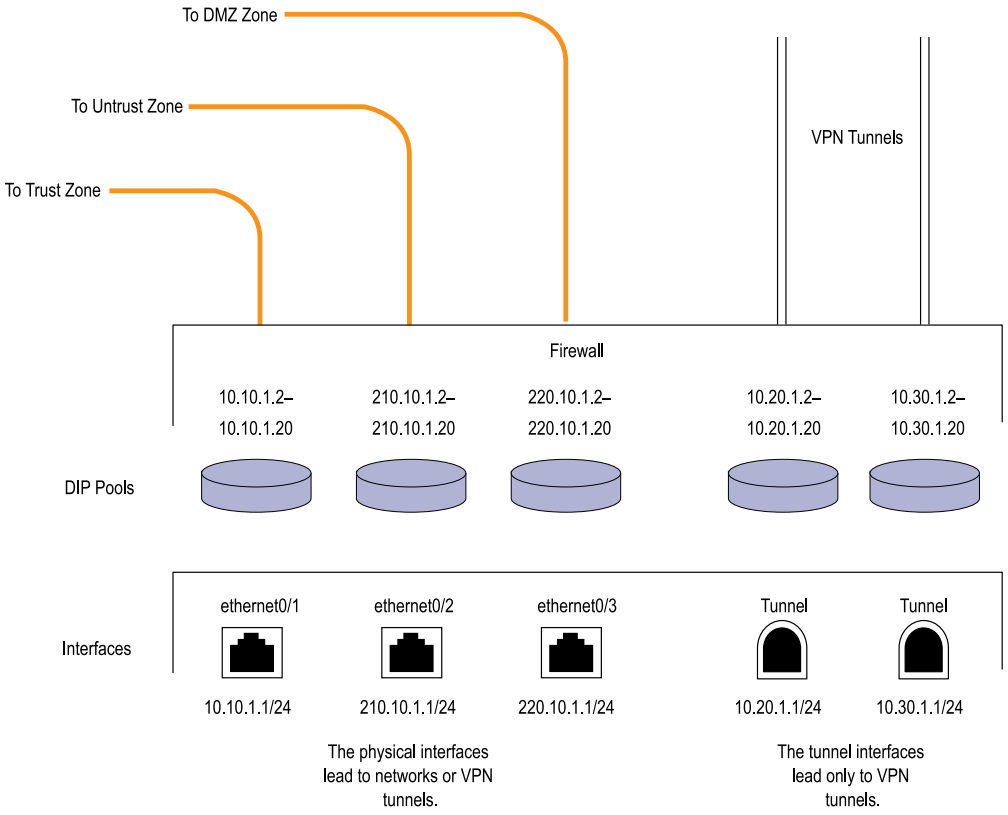
```
set session-cache enable
set service AOL session-cache
set session-cache count 18
```

Dynamic IP Pools

A Dynamic IP (DIP) pool is a range of IP addresses from which the security device can dynamically or deterministically take addresses to use when performing Network Address Translation (NAT) on the source IP address (NAT-src) in IP packet headers. (For information about deterministic source address translation, see “NAT-Src from a DIP Pool with Address Shifting” on page 1492.) If the range of addresses in a DIP pool is in the same subnet as the interface IP address, the pool must exclude the interface IP address, router IP addresses, and any Mapped IP (MIP) or Virtual IP (VIP) addresses that might also be in that subnet. If the range of addresses is in the subnet of an extended interface, the pool must exclude the extended interface IP address.

There are three kinds of interfaces that you can link to DIP pools: physical interfaces and subinterfaces for network and VPN traffic and tunnel interfaces for VPN tunnels only.

Figure 53: DIP Interfaces



Port Address Translation

Using Port Address Translation (PAT), multiple hosts can share the same IP address, the security device maintaining a list of assigned port numbers to distinguish which session belongs to which host. With PAT enabled, up to ~64,500 hosts can share a single IP address.

Some applications, such as NetBIOS Extended User Interface (NetBEUI) and Windows Internet Naming Service (WINS), require specific port numbers and cannot function properly if PAT is applied to them. For such applications, you can specify not to perform PAT (that is, to use a fixed port) when applying DIP. For fixed-port DIP, the security device hashes the original host IP address and saves it in its host hash table, thus allowing the security device to associate the right session with each host.

Creating a DIP Pool with PAT

In this example, you want to create a VPN tunnel for users at the local site to reach an FTP server at a remote site. However, the internal networks at both sites use the same private address space of 10.1.1.0/24. To solve the problem of overlapping

addresses, you create a tunnel interface in the Untrust zone on the local security device, assign it IP address 10.10.1.1/24, and associate it with a DIP pool with a range of one address (10.10.1.2–10.10.1.2) and Port Address Translation enabled.

The admin at the remote site, must also create a tunnel interface with an IP address in a neutral address space, such as 10.20.2.1/24, and set up a mapped IP (MIP) address to its FTP server, such as 10.20.2.5 to host 10.1.1.5.

NOTE: This example includes only the configuration of the tunnel interface and its accompanying DIP pool. For a complete example showing all the configuration steps necessary for this scenario, see "VPN Sites with Overlapping Addresses" on page 863.

WebUI

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Fixed IP: (select)
IP Address / Netmask: 10.10.1.1/24

Network > Interfaces > Edit (for tunnel.1) > DIP > New: Enter the following, then click **OK**:

ID: 5
IP Address Range: 10.10.1.2 ~ 10.10.1.2
Port Translation: (select)
In the same subnet as the interface IP or its secondary IPs: (select)

NOTE: You can use the ID number displayed, which is the next available number sequentially, or enter a different number.

CLI

```
set interface tunnel.1 zone untrust-tun
set interface tunnel.1 ip 10.10.1.1/24
set interface tunnel.1 dip 5 10.10.1.2 10.10.1.2
save
```

NOTE: Because PAT is enabled by default, there is no argument for enabling it. To create the same DIP pool as defined above but without PAT (that is, with fixed port numbers), do the following:
(WebUI) Network > Interfaces > Edit (for tunnel.1) > DIP > New: Clear the Port Translation check box, then click **OK**.
(CLI) set interface tunnel.1 dip 5 10.10.1.2 10.10.1.2 fix-port

You can add a maximum of three IP address ranges for a fixed-port DIP pool. The IP address ranges should not overlap. When the first address range is exhausted, the security device attempts to process the NAT request using the second address range. When the second address range is exhausted, the security device attempts to process the NAT request using the third address range. Note that the total range of all IP addresses defined in the fixed-port DIP pool must not exceed the permitted address scope of the subnet. For more information, see "NAT-Src from a DIP Pool with Address Shifting" on page 1492.

Modifying a DIP Pool

In this example, you change the range of addresses in an existing DIP pool (ID 5) from 10.20.1.2 – 10.20.1.2 to 10.20.1.2 – 10.20.1.10. This DIP pool is associated with tunnel.1. Note that to change the DIP pool range through the CLI, you must first remove (or unset) the existing DIP pool and then create a new pool.

NOTE: There are no policies using this particular DIP pool. If a policy uses a DIP pool, you must first delete the policy or modify it to not use the DIP pool before you can modify the DIP pool.

WebUI

Network > Interfaces > Edit (for tunnel.1) > DIP > Edit (for ID 5): Enter the following, then click **OK**:

IP Address Range: 10.20.1.2 ~ 10.20.1.10

CLI

```
unset interface tunnel.1 dip 5
set interface tunnel.1 dip 5 10.20.1.2 10.20.1.10
save
```

Sticky DIP Addresses

When a host initiates several sessions that match a policy requiring Network Address Translation (NAT) and is assigned an address from a DIP pool with port translation enabled, the security device assigns a different source IP address for each session. Such random address assignment can be problematic for services that create multiple sessions that require the same source IP address for each session.

NOTE: For DIP pools that do not perform port translation, the security device assigns one IP address for all concurrent sessions from the same host.

For example, it is important to have the same IP address for multiple sessions when using the AOL Instant Messaging (AIM) client. You create one session when you log in, and another for each chat. For the AIM server to verify that a new chat belongs to an authenticated user, it must match the source IP address of the login session with that of the chat session. If they are different—possibly because they were randomly assigned from a DIP pool during the NAT process—the AIM server rejects the chat session.

To ensure that the security device assigns the same IP address from a DIP pool to a host for multiple concurrent sessions, you can enable the "sticky" DIP address feature by entering the CLI command **set dip sticky**.

Using DIP in a Different Subnet

If circumstances require that the source IP address in outbound firewall traffic be translated to an address in a different subnet from that of the egress interface, you can use the extended interface option. This option allows you to graft a second IP address and an accompanying DIP pool onto an interface that is in a different subnet. You can then enable NAT for individual policies and specify the DIP pool built on the extended interface for the translation.

In this example, two branch offices have leased lines to a central office. The central office requires them to use only the authorized IP addresses it has assigned them. However, the offices receive different IP addresses from their ISPs for Internet traffic. For communication with the central office, you use the extended interface option to configure the security device in each branch office to translate the source IP address in packets it sends to the central office to the authorized address. The authorized and assigned IP addresses for branch offices A and B are as follows:

Table 27: Authorized Office IP Addresses

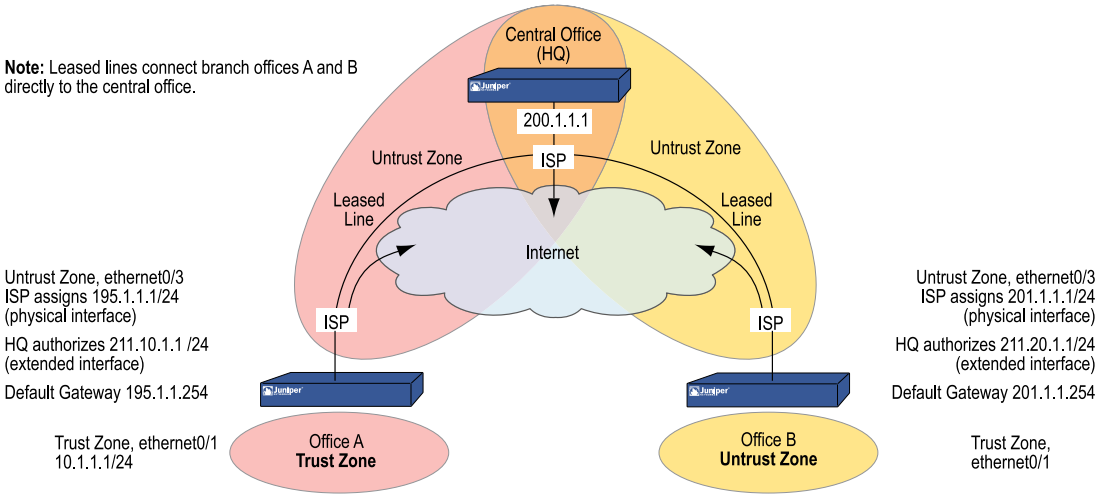
	Assigned IP Address (from ISP) Used for Untrust Zone Physical Interface	Authorized IP Address (from Central Office) Used for Untrust Zone Extended Interface DIP
Office A	195.1.1.1/24	211.10.1.1/24
Office B	201.1.1.1/24	211.20.1.1/24

The security devices at both sites have a Trust zone and an Untrust zone. All security zones are in the trust-vr routing domain. You bind ethernet0/1 to the Trust zone and assign it IP address 10.1.1.1/24. You bind ethernet0/3 to the Untrust zone and give it the IP address assigned by the ISPs: 195.1.1.1/24 for Office A and 201.1.1.1/24 for Office B. You then create an extended interface with a DIP pool containing the authorized IP address on ethernet0/3:

- Office A: extended interface IP 211.10.1.10/24; DIP pool 211.10.1.1 – 211.10.1.1; PAT enabled
- Office B: extended interface IP 211.20.1.10/24; DIP pool 211.20.1.1 – 211.20.1.1; PAT enabled

You set the Trust zone interface in NAT mode. It uses the Untrust zone interface IP address as its source address in all outbound traffic except for traffic sent to the central office. You configure a policy to the central office that translates the source address to an address in the DIP pool in the extended interface. (The DIP pool ID number is 5. It contains one IP address, which, with Port Address Translation (PAT), can handle sessions for ~64,500 hosts.) The MIP address that the central office uses for inbound traffic is 200.1.1.1, which you enter as "HQ" in the Untrust zone address book on each security device.

Figure 54: DIP Under Another Subnet



NOTE: Each ISP must set up a route for traffic destined to a site at the end of a leased line to use that leased line. The ISPs route any other traffic they receive from a local security device to the Internet.

WebUI (Branch Office A)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Interface Mode: NAT

Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 195.1.1.1/24
Interface Mode: Route

Network > Interfaces > Edit (for ethernet0/3) > DIP > New: Enter the following, then click **OK**:

ID: 5
IP Address Range: 211.10.1.1 ~ 211.10.1.1
    Port Translation: (select)
Extended IP/Netmask: 211.10.1.10/255.255.255.0

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: HQ
IP Address/Domain Name:
    IP/Netmask: (select), 200.1.1.1/32
Zone: Untrust

3. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
    Interface: ethernet0/3
    Gateway IP address: 195.1.1.254

4. **Policies**

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: ANY
Action: Permit

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), HQ
Service: ANY

Action: Permit
Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): 5 (211.10.1.1-211.10.1.1)/X-late

WebUI (Branch Office B)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 201.1.1.1/24
Interface Mode: Route

Network > Interfaces > Edit (for ethernet0/3) > DIP > New: Enter the following, then click **OK**:

ID: 5
IP Address Range: 211.20.1.1 ~ 211.20.1.1
Port Translation: (select)
Extended IP/Netmask: 211.20.1.10/255.255.255.0

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: HQ
IP Address/Domain Name:
IP/Netmask: (select), 200.1.1.1/32
Zone: Untrust

3. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0

Gateway: (select)
Interface: ethernet0/3
Gateway IP address: 201.1.1.254

4. **Policies**

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: ANY
Action: Permit
Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), HQ
Service: ANY
Action: Permit
Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
DIP On: (select), 5 (211.20.1.1-211.20.1.1)/X-late

CLI (Branch Office A)

1. **Interfaces**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 10.1.1.1/24
set interface ethernet0/1 nat
set interface ethernet0/3 zone untrust
set interface ethernet0/3 ip 195.1.1.1/24
set interface ethernet0/3 route
set interface ethernet0/3 ext ip 211.10.1.10 255.255.255.0 dip 5 211.10.1.1
```

2. **Address**

```
set address untrust hq 200.1.1.1/32
```

3. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet0/3 gateway 195.1.1.254
```

4. **Policies**

```
set policy from trust to untrust any any any permit
set policy top from trust to untrust any hq any nat src dip 5 permit
save
```

CLI (Branch Office B)

1. **Interfaces**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 10.1.1.1/24
set interface ethernet0/1 nat
set interface ethernet0/3 zone untrust
set interface ethernet0/3 ip 201.1.1.1/24
set interface ethernet0/3 route
set interface ethernet0/3 ext ip 211.20.1.10 255.255.255.0 dip 5 211.20.1.1
```

2. **Address**

```
set address untrust hq 200.1.1.1/32
```

3. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet0/3 gateway 201.1.1.254
```

4. **Policies**

```
set policy from trust to untrust any any any permit
set policy top from trust to untrust any hq any nat src dip 5 permit
save
```

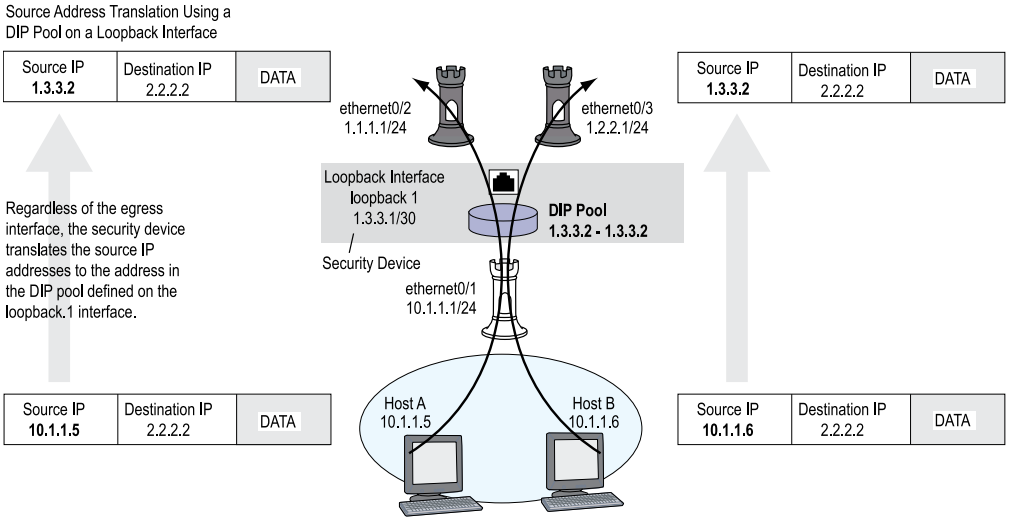
Using a DIP on a Loopback Interface

A loopback interface is a logical interface that is always in the up state as long as the device on which it resides is up. You can create a pool of Dynamic IP (DIP) addresses on a loopback interface so that it can be accessed by the group of interfaces belonging to its associated loopback interface group when performing source address translation. The addresses that the security device draws from such a DIP pool are in the same subnet as the loopback interface IP address, not in the subnet of any of the member interfaces. (Note that the addresses in the DIP pool must not overlap with the interface IP address or any MIP addresses also defined on the loopback interface.)

NOTE: For information about loopback interfaces, see “Loopback Interfaces” on page 75.

The primary application for putting a DIP pool on a loopback interface is to translate source addresses to the same address or range of addresses although different packets might use different egress interfaces.

Figure 55: Loopback DIP



In this example, the security device receives the following IP addresses for two Untrust zone interfaces from different Internet service providers (ISPs): ISP-1 and ISP-2:

- ethernet0/2, 1.1.1.1/24, ISP-1
- ethernet0/3, 1.2.2.1/24, ISP-2

You bind these interfaces to the Untrust zone and then assign them the above IP addresses. You also bind ethernet0/1 to the Trust zone and assign it IP address 10.1.1.1/24.

You want the security device to translate the source address in outbound traffic from the Trust zone to a remote office in the Untrust zone. The translated address must be the same IP address (1.3.3.2) because the remote office has a policy permitting inbound traffic only from that IP address. You have previously obtained the public IP addresses 1.3.3.1 and 1.3.3.2 and have notified both ISPs that you are using these addresses in addition to the addresses that they assign the device.

You configure a loopback interface loopback.1 with the IP address 1.3.3.1/30 and a DIP pool of 1.3.3.2 – 1.3.3.2 on that interface. The DIP pool has ID number 10. You then make ethernet0/1 and ethernet0/2 members of the loopback group for loopback.1.

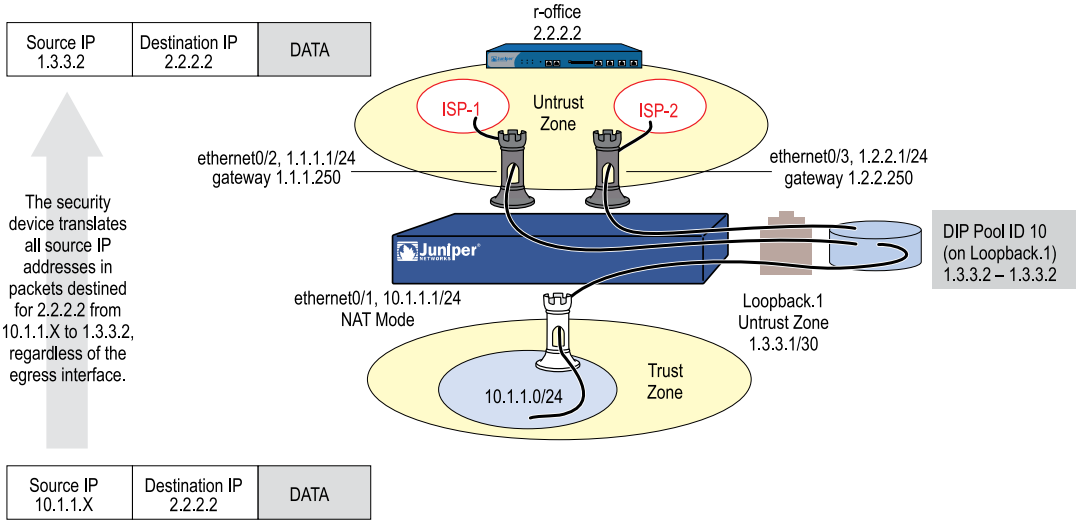
You define an address for the remote office named “r-office” with IP address 2.2.2.2/32. You also define default routes for both ethernet0/1 and ethernet0/2 interfaces pointing to the routers for ISP-1 and ISP-2, respectively.

You define routes to two gateways for outbound traffic to use. Because you do not prefer one route over the other, you do not include any metrics in the routes. Outbound traffic might follow either route.

NOTE: To indicate a route preference, include metrics in both routes, giving your preferred route a higher metric—that is, a value closer to 1.

Finally, you create a policy applying Source Network Address Translation (NAT-src) to outbound traffic to the remote office. The policy references DIP pool ID 10.

Figure 56: Loopback DIP Policy



WebUI

1. **Interfaces**

   Network > Interfaces > New Loopback IF: Enter the following, then click **OK**:

   Interface Name: loopback.1
   Zone: Untrust (trust-vr)
   IP Address/Netmask: 1.3.3.1/30

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

   As member of loopback group: loopback.1
   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet0/2): Enter the following, then click **OK**:

   As member of loopback group: loopback.1
   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

Interface Mode: Route

Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.2.2.1/24
Interface Mode: Route

2. **DIP Pool**

Network > Interfaces > Edit (for loopback.1) > DIP > New: Enter the following, then click **OK**:

ID: 5
IP Address Range: 1.3.3.2 ~ 1.3.3.2
Port Translation: (select)

3. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: r-office
IP Address/Domain Name:
IP/Netmask: (select), 2.2.2.2/32
Zone: Untrust

4. **Routes**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet0/2
Gateway IP address: 1.1.1.250

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet0/3
Gateway IP address: 1.2.2.250

5. **Policy**

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), r-office

Service: ANY
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
DIP On: (select), 10 (1.3.3.2-1.3.3.2)/port-xlate

CLI

1. **Interfaces**

```
set interface loopback.1 zone untrust
set interface loopback.1 ip 1.3.3.1/30
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 10.1.1.1/24
set interface ethernet0/1 nat
set interface ethernet0/2 zone untrust
set interface ethernet0/2 ip 1.1.1.1/24
set interface ethernet0/2 loopback-group loopback.1
set interface ethernet0/3 zone untrust
set interface ethernet0/3 ip 1.2.2.1/24
set interface ethernet0/3 loopback-group loopback.1
```

2. **DIP Pool**

```
set interface loopback.1 dip 10 1.3.3.2 1.3.3.2
```

3. **Address**

```
set address untrust r-office 2.2.2.2/32
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet0/2 gateway 1.1.1.250
set vrouter trust-vr route 0.0.0.0/0 interface ethernet0/3 gateway 1.2.2.250
```

5. **Policy**

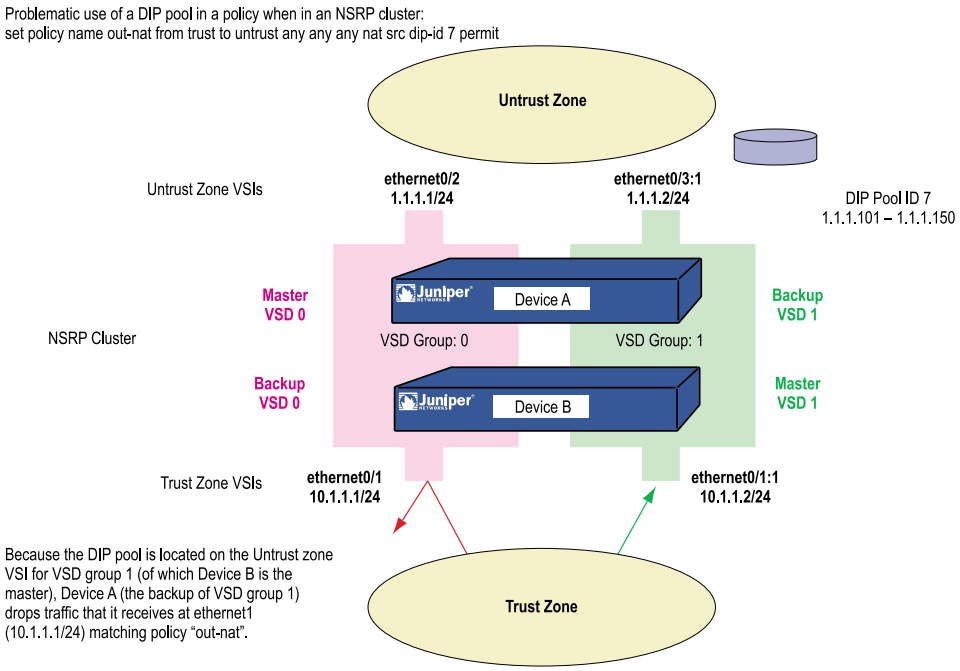
```
set policy from trust to untrust any r-office any nat src dip-id 10 permit
save
```

Creating a DIP Group

When you group two security devices into a redundant cluster to provide high availability (HA) services in an Active/Active configuration, both devices share the same configuration and both process traffic simultaneously. A problem can arise when you define a policy to perform Network Address Translation (NAT) using a dynamic IP (DIP) pool located on one VSI. Because that VSI is active only on the security device acting as the primary of the VSD group to which the VSI is bound,

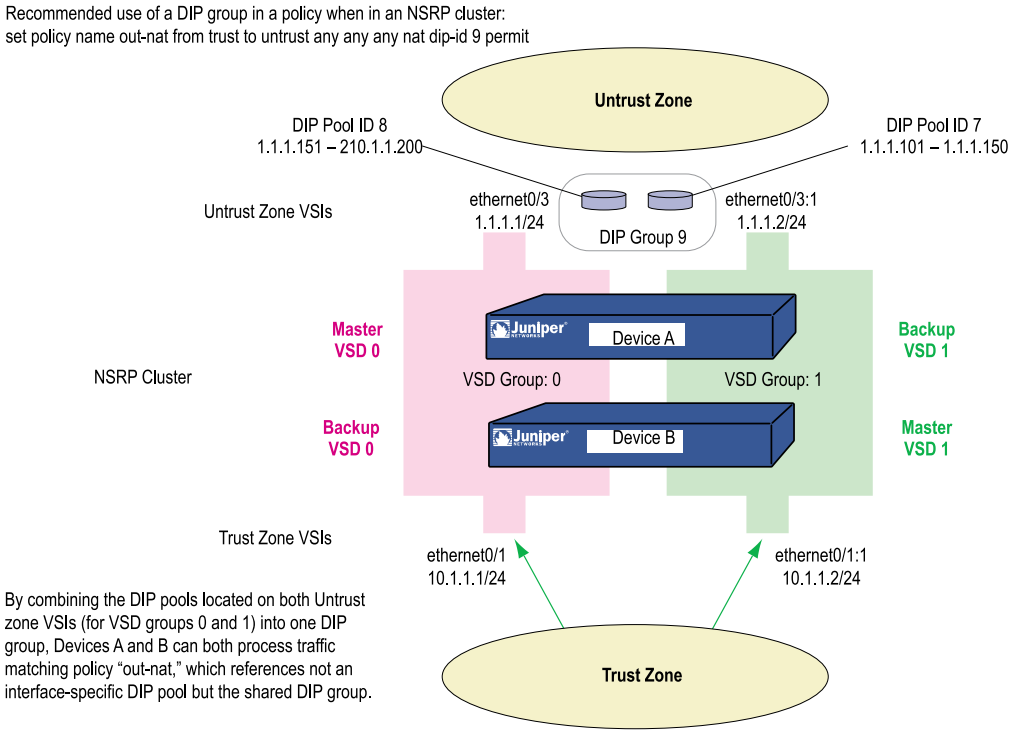
any traffic sent to the other security device—the one acting as the backup of that VSD group—cannot use that DIP pool and is dropped.

Figure 57: DIP Problems with NAT with One VSI



To solve this problem, you can create two DIP pools—one on the Untrust zone VSI for each VSD group—and combine the two DIP pools into one DIP group, which you reference in the policy. Each VSI uses its own VSD pool even though the policy specifies the DIP group.

Figure 58: Creating Two DIP Pools in One DIP Group



NOTE: For more information about setting up security devices for HA, see "High Availability" on page 1763.

In this example, you provide NAT services on two security devices (Devices A and B) in an Active/Active HA pair.

You create two DIP pools—DIP 5 (1.1.1.20 – 1.1.1.29) on ethernet0/3 and DIP 6 (1.1.1.30 – 1.1.1.39) on ethernet0/3:1. You then combine them into a DIP group identified as DIP 7, which you reference in a policy.

The VSIs for VSD groups 0 and 1 are as follows:

- Untrust zone VSI ethernet0/3 1.1.1.1/24 (VSD group 0)
- Untrust zone VSI ethernet0/3:1 1.1.1.2/24 (VSD group 1)
- Trust zone VSI ethernet0/1 10.1.1.1/24 (VSD group 0)
- Trust zone VSI ethernet0/1:1 10.1.1.2/24 (VSD group 1)

Let's assume that you have already set up Devices A and B in an NSRP cluster, created VSD group 1 (ScreenOS automatically creates VSD group 0 when you put a device in an NSRP cluster), and configured the above interfaces. (For information about configuring security devices for NSRP, see"High Availability" on page 1763.)

WebUI

1. **DIP Pools**

   Network > Interfaces > Edit (for ethernet0/3) > DIP > New: Enter the following, then click **OK**:

   ID: 5
   IP Address Range: 1.1.1.20 – 1.1.1.29
   Port Translation: (select)

   Network > Interfaces > Edit (for ethernet0/3:1) > DIP > New: Enter the following, then click **OK**:

   ID: 6
   IP Address Range: 1.1.1.30 – 1.1.1.39
   Port Translation: (select)

NOTE: At the time of this release, you can only define a DIP group through the CLI.

2. **Policy**

   Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: ANY
   Action: Permit

   > Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   NAT:
   Source Translation: (select)
   DIP On: (select), 7

CLI

1. **DIP Pools**

   ```
   set interface ethernet0/3 dip 5 1.1.1.20 1.1.1.29
   set interface ethernet0/3:1 dip 6 1.1.1.30 1.1.1.39
   ```

2. **DIP Groups**

   ```
   set dip group 7 member 5
   set dip group 7 member 6
   ```

3. **Policy**

   set policy from trust to untrust any any any nat src dip-id 7 permit
   save

Setting a Recurring Schedule

A schedule is a configurable object that you can associate with one or more policies to define when they are in effect. Through the application of schedules, you can control network traffic flow and enforce network security.

When you define a schedule, enter values for the following parameters:

- **Schedule Name:** The name that appears in the Schedule drop-down list in the Policy Configuration dialog box. Choose a descriptive name to help you identify the schedule. The name must be unique and is limited to 19 characters.
- **Comment:** Any additional information that you want to add.
- **Recurring:** Enable this when you want the schedule to repeat weekly.

  Start and End Times: You must configure both a start time and an end time. You can specify up to two time periods within the same day.
- **Once:** Enable this when you want the schedule to start and end only once.

  mm/dd/yyyy hh:mm: You must enter both start and stop dates and times.

In this example, there is a short-term employee named Tom who is using the company's Internet access for personal pursuits after work. You create a schedule for non-business hours that you can then associate with a policy to deny outbound TCP/IP traffic from that worker's computer (10.1.1.5/32) outside of regular business hours.

WebUI

1. **Schedule**

   Policy > Policy Elements > Schedules > New: Enter the following, then click **OK**:

   Schedule Name: After Hours
   Comment: For non-business hours
   Recurring: (select)
   Period 1:

Weekday	Start Time	End Time
Sunday	00:00	23:59
Monday	00:00	06:00
Tuesday	00:00	06:00

Weekday	Start Time	End Time
Wednesday	00:00	06:00
Thursday	00:00	06:00
Friday	00:00	06:00
Saturday	00:00	23:59

Period 2:

Weekday	Start Time	End Time
Sunday	17:00	23:59
Monday	17:00	23:59
Tuesday	17:00	23:59
Wednesday	17:00	23:59
Thursday	17:00	23:59
Friday	17:00	23:59
Saturday	17:00	23:59

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Tom
   Comment: Temp
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.5/32
   Zone: Trust

3. **Policy**

   Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Name: No Net
   Source Address:
   Address Book Entry: (select), Tom
   Destination Address:
   Address Book Entry: (select), Any
   Service: HTTP
   Action: Deny
   Schedule: After Hours

CLI

1. **Schedule**

```
set schedule “after hours” recurrent sunday start 00:00 stop 23:59
set schedule “ after hours” recurrent monday start 00:00 stop 06:00 start
17:00 stop 23:59
set schedule “ after hours” recurrent tuesday start 00:00 stop 06:00 start
17:00 stop 23:59
set schedule “ after hours” recurrent wednesday start 00:00 stop 06:00 start
 17:00 stop 23:59
set schedule “ after hours” recurrent thursday start 00:00 stop 06:00 start
17:00 stop 23:59
set schedule “ after hours” recurrent friday start 00:00 stop 06:00 start 17:00
 stop 23:59
set schedule “ after hours” recurrent saturday start 00:00 stop 23:59 comment
 “ for non-business hours”
```

2. **Address**

```
set address trust tom 10.1.1.5/32 “temp”
```

3. **Policy**

```
set policy from trust to untrust tom any http deny schedule “after hours”
save
```

Chapter 7

Policies

The default behavior of a security device is to deny all traffic between security zones (interzone traffic) and—except for traffic within the Untrust zone—allow all traffic between interfaces bound to the same zone (intrazone traffic). To permit selected interzone traffic to cross a security device, you must create interzone policies that override the default behavior. Similarly, to prevent selected intrazone traffic from crossing a security device, you must create intrazone policies.

This chapter describes what policies do and how the various elements that comprise a policy are related. It contains the following sections:

NOTE: If you configure multicast routing on a security device, you might have to configure multicast policies. For information about multicast policies, see "Multicast Policies" on page **7**-135.

Basic Elements

A policy permits, denies, or tunnels specified types of traffic unidirectionally between two points. The type of traffic (or "service"), the location of the two endpoints, and the invoked action compose the basic elements of a policy. Although there can be other components, the required elements, which together constitute the core section of a policy, are described in Table 27.

Table 28: Basic Policy Elements

Element	Description
Direction	The direction of traffic between two security zones (from a source zone to a destination zone)
Source Address	The address from which traffic initiates

Table 28: Basic Policy Elements *(continued)*

Element	Description
Destination Address	The address to which traffic is sent
Service	The type of traffic transmitted
Action	The action that the security device performs when it receives traffic meeting the first four criteria: deny, permit, reject, or tunnel Note: The "tunnel" action—VPN or L2TP tunnel—contains the concept of " permit" implicitly.

For example, the policy stated in the following CLI command permits FTP traffic from any address in the Trust zone to an FTP server named "server1" in the DMZ zone:

```
set policy from trust to untrust any server1 ftp permit
```

- **Direction**: **from trust to untrust** (that is, from the Trust zone to the Untrust zone)
- **Source Address**: **any** (that is, any address in the Trust zone. The term "any" stands for a predefined address that applies to any address in a zone)
- **Destination Address**: **server1** (a user-defined address in the Untrust zone address book)
- **Service**: **ftp** (File Transfer Protocol)
- **Action**: **permit** (the security device permits this traffic to traverse its firewall)

Three Types of Policies

You control the flow of traffic with the following three types of policies:

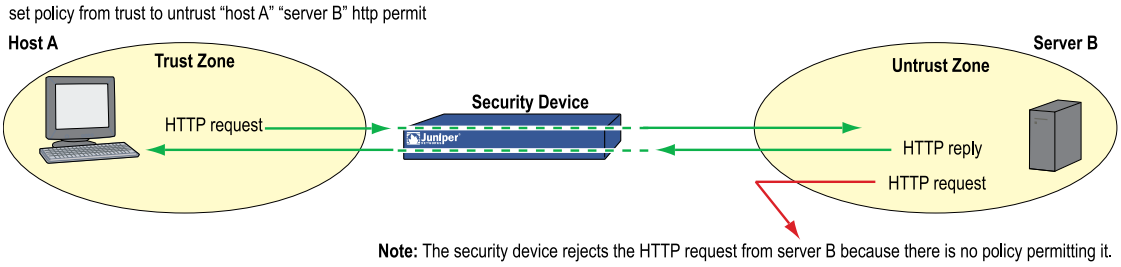
- **Interzone Policies**—Let you regulate the kind of traffic allowed to pass from one security zone to another.
- **Intrazone Policies**— Let you regulate the kind of traffic allowed to cross interfaces bound to the same zone.
- **Global Policies**—Let you regulate traffic between addresses, regardless of their security zones.

Interzone Policies

Interzone policies provide traffic control between security zones. You can set interzone policies to deny, permit, reject, or tunnel traffic from one zone to another. Using stateful inspection techniques, a security device maintains a table of active TCP sessions and active UDP "pseudo" sessions so that it can allow replies to service requests. For example, if you have a policy allowing HTTP requests from host A in the Trust zone to server B in the Untrust zone, when the security device receives

HTTP replies from server B to host A, the security device checks the received packet against its table. Finding the packet to be a reply to an approved HTTP request, the security device allows the packet from server B in the Untrust zone to cross the firewall to host A in the Trust zone. To permit traffic initiated by server B to host A (not just replies to traffic initiated by host A), you must create a second policy from server B in the Untrust zone to host A in the Trust zone.

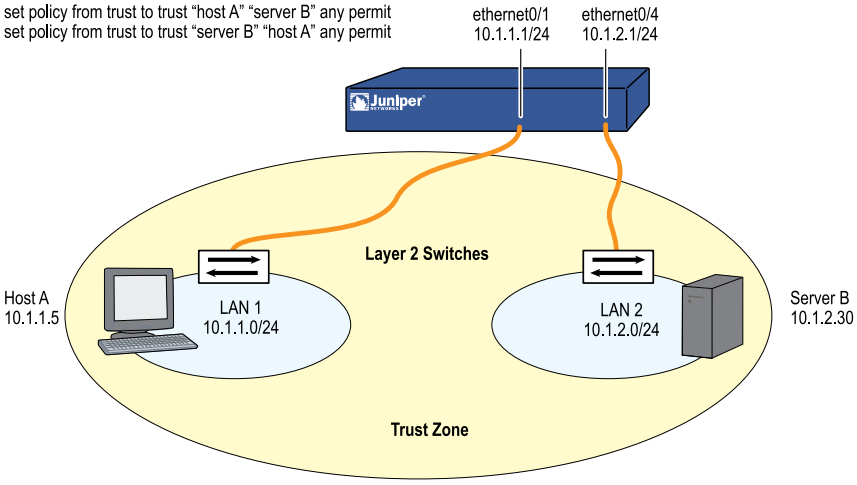
Figure 59: Interzone Policy



Intrazone Policies

Intrazone policies provide traffic control between interfaces bound to the same security zone. The source and destination addresses are in the same security zone but are reached via different interfaces on the security device. Like interzone policies, intrazone policies control traffic flowing unidirectionally. To allow traffic initiated at either end of a data path, you must create two policies—one policy for each direction.

Figure 60: Intrazone Policy



Intrazone policies do not support VPN tunnels or Source Network Address Translation (NAT-src) when it is set at the interface level (**set interface** interface **nat**). However, intrazone policies do support policy-based NAT-src and NAT-dst. They also support destination address translation when the policy references a mapped IP (MIP) as the destination address. (For information about NAT-src, NAT-dst, and MIPs, see *Address Translation*.)

Global Policies

Unlike interzone and intrazone policies, global policies do not reference specific source and destination zones. Global policies reference user-defined Global zone addresses or the predefined Global zone address “any”. These addresses can span multiple security zones. For example, if you want to provide access to or from multiple zones, you can create a global policy with the Global zone address “ any,” which encompasses all addresses in all zones.

NOTE: At the time of this release, global policies do not support Source Network Address Translation (NAT-src), VPN tunnels, or transparent mode. You can, however, specify a MIP or VIP as the destination address in a global policy.

Policy Set Lists

A security device maintains three different policy set lists, one each for interzone policies, intrazone policies, and global policies.

When the security device receives a packet initiating a new session, the device notes the ingress interface, and thereby learns the source zone to which that interface is bound. The security device then performs a route lookup to determine the egress interface, and thus determines the destination zone to which that interface is bound. Using the source and destination zones, the security device can perform a policy lookup, consulting the policy set lists in the following order:

1. If the source and destination zones are different, the security device performs a policy lookup in the interzone policy set list.

   (or)

   If the source and destination zones are the same, the security device performs a policy lookup in the intrazone policy set list.
2. If the security device performs the interzone or intrazone policy lookup and does not find a match, the security device then checks the global policy set list for a match.
3. If the security device performs the interzone and global policy lookups and does not find a match, the security device then applies the default permit/deny policy to the packet: **unset/set policy default-permit-all**.

   (or)

   If the security device performs the intrazone and global policy lookups and does not find a match, the security device then applies the intrazone blocking setting for that zone to the packet: **unset/set zone** zone **block**.

The security device searches each policy set list from top to bottom. Therefore, you must position more specific policies above less specific policies in the list. (For information about policy order, see “Reordering Policies” on page 190.)

Policies Defined

A security device provides a network boundary with a single point of entry and exit. Because all traffic must pass through this point, you can screen and direct that traffic by implementing policy set lists—for interzone policies, intrazone policies, and global policies.

Policies allow you to deny, permit, reject (deny and send a TCP RST or an ICMP port unreachable message to the source host), encrypt and decrypt, authenticate, prioritize, schedule, filter, have no hardware session, and monitor the traffic attempting to cross from one security zone to another. You decide which users and what data can enter and exit, and when and where they can go.

NOTE: For security devices that support virtual systems, policies set in the root system do not affect policies set in virtual systems.

Policies and Rules

A single user-defined policy produces one or more logical rules internally, and each logical rule consists of a set of components—source address, destination address, and service. The components consume memory resources. The logical rules that reference the components do not.

Depending on the use of multiple entries or groups for the source address, destination address, and service components in a policy, the number of logical rules can be much larger than is readily apparent from the creation of the single policy. For example, the following policy produces 125 logical rules:

1 policy: 5 source addresses x 5 destination addresses x 5 services = 125 logical rules

However, the security device does not duplicate components for each logical rule. The rules make use of the same set of components in various combinations. For example, the above policy that produces 125 logical rules results in only 15 components:

5 source addresses + 5 destination addresses + 5 services = 15 components

These 15 components combine in various ways to produce the 125 logical rules generated by the single policy. By allowing multiple logical rules to use the same set of components in different combinations, the security device consumes far fewer resources than if each logical rule had a one-to-one relationship with its components.

Because the installation time of a new policy is proportional to the number of components that the security device adds, removes, or modifies, policy installation becomes faster with fewer components. Also, by allowing a large number of logical rules to share a small set of components, ScreenOS allows you to create more policies—and the security device to create more rules—than would be possible if each rule required dedicated components.

Anatomy of a Policy

A policy must contain the following elements:

- ID (automatically generated, but can be user-defined in the CLI)
- Zones (source and destination)
- Addresses (source and destination)
- Services
- Action (deny, permit, reject, tunnel)

A policy can also contain the following elements:

- Application
- Name
- VPN tunneling
- L2TP tunneling
- Deep inspection (DI)
- Placement at the top of the policy list
- Source Network Address Translation (NAT-src)
- Destination Network Address Translation (NAT-dst)
- No hardware session
- User authentication
- High availability session backup
- Web filtering
- Logging
- Counting
- Traffic alarm threshold
- Schedules
- Antivirus scanning
- Traffic shaping

The remainder of this section examines each of the above elements.

ID

Every policy has an ID number, whether you define one or the security device automatically assigns it. You can only define an ID number for a policy through the set policy command in the CLI: **set policy id** number … After you know the ID number, you can enter the policy context to issue further commands to modify the policy. (For more information about policy contexts, see “Entering a Policy Context” on page 185.)

Zones

A zone can be a segment of network space to which security measures are applied (a security zone), a logical segment to which a VPN tunnel interface is bound (a tunnel zone), or either a physical or logical entity that performs a specific function (a function zone). A policy allows traffic to flow between two security zones (interzone policy) or between two interfaces bound to the same zone (intrazone policy). (For more information, see “Zones” on page 25, “ Interzone Policies” on page 161, and “ Intrazone Policies” on page 161.)

Addresses

Addresses are objects that identify network devices such as hosts and networks by their location in relation to the firewall—in one of the security zones. Individual hosts are specified using the mask 255.255.255.255, indicating that all 32 bits of the address are significant. Networks are specified using their subnet mask to indicate which bits are significant. To create a policy for specific addresses, you must first create entries for the relevant hosts and networks in the address book.

You can also create address groups and apply policies to them as you would to other address book entries. When using address groups as elements of policies, be aware that because the security device applies the policy to each address in the group, the number of available internal logical rules and the components that comprise those rules can become depleted more quickly than expected. This is a danger especially when you use address groups for both the source and destination. (For more information, see “Policies and Rules” on page 164.)

Wildcard Addresses

In addition to netmasks, Juniper Networks security devices allow you to define wildcard masks. A wildcard mask is similar to the subnet mask, but instructs the security device to consider the IP addresses corresponding to ‘0’s in wildcard mask. IP address that have a wildcard mask are called wildcard addresses. Wildcard addresses enable you to reduce the number of policies you create to control traffic.

However, a wildcard address cannot be a member of an address group or coexist with other normal addresses in the source or destination address domains in a policy. In addition, you can neither define a VPN policy or RPC policy with wildcard addresses nor shift IP addresses. Because the security device must check all the possible combinations on a matched policy during policy lookup, wildcard policies cause a performance penalty.

NOTE: The security device considers the IP address corresponding to '1's in a netmask or wildcard mask. However, netmask differs from wildcard mask on the rule that the '1's in the netmask must always be continuous and in the leading place, while the rule is not applicable for wildcard mask.

Services

Services are objects that identify application protocols using Layer 4 information such as standard and accepted TCP and UDP port numbers for application services like Telnet, FTP, SMTP, and HTTP. The ScreenOS includes predefined core Internet services. Additionally, you can define custom services.

You can define policies that specify which services are permitted, denied, encrypted, authenticated, logged, or counted.

Action

An action is an object that describes what the firewall does to the traffic it receives.

- **Deny** blocks the packet from traversing the firewall.
- **Permit** allows the packet to pass the firewall.
- **Reject** blocks the packet from traversing the firewall. The security device drops the packet and sends a TCP reset (RST) segment to the source host for TCP traffic and an ICMP "destination unreachable, port unreachable" message (type 3, code 3) for UDP traffic. For types of traffic other than TCP and UDP, the security device drops the packet without notifying the source host, which is also what occurs when the action is " deny."

NOTE: Juniper Networks security device with interfaces operating on Layer 2 (transparent mode) and Layer 3 (NAT or route mode) security zones support TCP RST.
The security device sends a TCP RST after receiving (and dropping) a TCP segment with any code bit set other than another RST.
When the ingress interface is operating at Layer 2 or Layer 3 and the protocol is TCP, the source IP address in the TCP RST is the destination IP address in the original (dropped) packet. When the ingress interface is operating at Layer 2 and the protocol is UDP, the source IP address in the ICMP message is also the destination IP address in the original packet. However, if the ingress interface is operating at Layer 3 and the protocol is UDP, then the source IP address in the ICMP message is that of the ingress interface.

- **Tunnel** encapsulates outgoing IP packets and decapsulates incoming IP packets. For an IPsec VPN tunnel, specify which VPN tunnel to use. For an L2TP tunnel, specify which L2TP tunnel to use. For L2TP-over-IPsec, specify both an IPsec VPN tunnel and an L2TP tunnel.

NOTE: For L2TP-over-IPsec, the source and destination addresses for the IPsec VPN tunnel must be the same as those for the L2TP tunnel.

The security device applies the specified action on traffic that matches the previously presented criteria: zones (source and destination), addresses (source and destination), and service.

Application

The application option specifies the Layer 7 application that maps to the Layer 4 service that you reference in a policy. A predefined service already has a mapping to a Layer 7 application. However, for custom services, you must link the service to an application explicitly, especially if you want the policy to apply an Application Layer Gateway (ALG) or deep inspection to the custom service.

NOTE: ScreenOS supports ALGs for numerous services, including DNS, FTP, H.323, HTTP, RSH, SIP, Telnet, and TFTP.

Applying an ALG to a custom service involves the following two steps:

- Define a custom service with a name, timeout value, transport protocol, and source and destination ports
- When configuring a policy, reference that service and the application type for the ALG that you want to apply

For information about applying deep inspection to a custom service, see “Mapping Custom Services to Applications” on page **4**-152.

Name

You can give a policy a descriptive name to provide a convenient means for identifying its purpose.

NOTE: For information regarding ScreenOS naming conventions—which apply to the names you create for policies—see “Naming Conventions and Character Types” on page 11.

VPN Tunneling

You can apply a single policy or multiple policies to any VPN tunnel that you have configured. In the WebUI, the VPN Tunnel option provides a drop-down list of all such tunnels. In the CLI, you can see all available tunnels with the **get vpn** command. (For more information, see “Site-to-Site Virtual Private Networks” on page **5**-79 and “ Dialup Virtual Private Networks” on page **5**-159.) When the VPN configurations at both ends of a VPN tunnel are using policy-based-NAT, then the administrators of both gateway devices each need to create an inbound and an outbound policy (four policies in total). When the VPN policies constitute a matching pair (that is, everything in the inbound and outbound policy configurations is the same except that the source and destination addresses are reversed), you can configure one policy and then select the Modify matching bidirectional VPN policy check box to create a second policy automatically for the opposite direction. For the configuration of a new policy, the

matching VPN policy check box is cleared by default. For the modification of an existing policy that is a member of a matching pair, the check box is selected by default, and any changes made to one policy are propagated to the other.

NOTE: This option is available only through the WebUI. It is not supported when there are multiple entries for any of the following policy components: source address, destination address, or service. In addition, you cannot use wildcard addresses in a VPN policy.

L2TP Tunneling

You can apply a single policy or multiple policies to any Layer 2 Tunneling Protocol (L2TP) tunnel that you have configured. In the WebUI, the L2TP option provides a drop-down list of all such tunnels. In the CLI, you can display status of active L2TP tunnels with the **get l2tp tunn_str active** command, and see all available tunnels with the **get l2tp all** command. You can also combine a VPN tunnel and an L2TP tunnel—if both have the same endpoints—to create a tunnel combining the characteristics of each. This is called L2TP-over-IPsec.

NOTE: A security device in transparent mode does not support L2TP.

Deep Inspection

Deep inspection (DI) is a mechanism for filtering the traffic permitted at the Network and Transport Layers by examining not only these layers but the content and protocol characteristics at the Application Layer. DI is designed to detect and prevent attacks or anomalous behavior present in traffic permitted by the security device

NOTE: In the Open Systems Interconnection (OSI) Model, the Network Layer is Layer 3, the Transport Layer is Layer 4, and the Application Layer is Layer 7. The OSI Model is a networking industry standard model of network protocol architecture. The OSI Model consists of seven layers.

To configure a policy for attack protection, you must make two choices: which attack group (or groups) to use and which attack action to take if an attack is detected. (For more information, see “Deep Inspection” on page **4**-115.)

Placement at the Top of the Policy List

By default, ScreenOS positions a newly created policy at the bottom of a policy set list. If you need to reposition the policy, you can use either of the policy reordering methods explained in “Reordering Policies” on page 192. To avoid the extra step of repositioning a newly created policy to the top of a policy set list, you can select the **Position at Top** option in the WebUI, or use the keyword **top** in the **set policy** command (**set policy top …**) in the CLI.

Session Limiting

When you configure or modify a policy, you can define a limit to the number of sessions from a source IP address. You can configure the device either to issue an alarm and allow the session to continue, or to drop any further traffic.

When you enforce session limiting, only the new sessions will be counted against the configured session limit for the source IP address to which the policy is applied. Similarly, in the case of NetScreen Redundancy Protocol (NSRP) or Virtual Router Redundancy Protocol (VRRP) clusters, the sessions that are reflected in the backup device will not be counted against the configured session limit. When a failover occurs, the backup device that takes over as the primary will allow any new sessions only within the limit, or after any existing sessions age out if the sessions count has reached the threshold.

Sending a TCP Session Close Notification

You can configure or modify a policy to send a TCP session close notification message when the session is closed by session timeout or with the **clear session** command. By default this option is disabled in a policy. When this option is enabled, the security device sends the notification ACKs to both the client and the server.

Before you enable a policy to send a TCP session close notification, you must enable the following options:

- TCP SYN checking, by using the **set flow tcp-syn-check** or **set flow tcp-syn-bit-check** command
- TCP reset option in client and server zones, by using the **set zone *name* tcp-rst** command
- TCP sequence check on ISG-1000/2000 and NS-5200/5400 devices, by using the **unset flow no-tcp-seq-check** command

Warning message appears if you try to enable **notify-conn-close** before enabling the above options.

NOTE: A warning message does not appear when you enable **notify-conn-close** in a policy with global or null zones.

The notification ACK and RST segments cannot be retransmitted.

To enable or disable a TCP session close notification within a policy, use the **set/unset notify-conn-close** command.

Source Network Address Translation

You can apply Source Network Address Translation (NAT-src) at the policy level. With NAT-src, you can translate the source address on either incoming or outgoing network and VPN traffic. The new source address can come from either a Dynamic IP (DIP)

pool or the egress interface. NAT-src also supports source port address translation (PAT). To learn about all the NAT-src options that are available, see “Source Network Address Translation” on page **8**-13.

NOTE: You can also perform NAT-src at the interface level, known as Network Address Translation (NAT). For information about both interface-level NAT-src and NAT, see “NAT Mode” on page 92.

Destination Network Address Translation

You can apply Destination Network Address Translation (NAT-dst) at the policy level. With NAT-dst, you can translate the destination address on either incoming or outgoing network and VPN traffic. NAT-dst can also support destination port mapping. To learn about all the NAT-dst options that are available, see “Destination Network Address Translation” on page **8**-27.

No Hardware Session

Using this option, you can disable the security device from creating a hardware session for a specific type of traffic. This option is useful for debugging and for when some types of traffic, such as PPTP, cannot be handled efficiently by the ASIC. In the CLI, use the no-hw-sess argument in the **set policy** command.

NOTE: Enabling or disabling this feature after a session has been created does not affect the session. For TCP traffic, you must create a dummy hardware session to pass the traffic to the CPU.

User Authentication

Selecting this option requires the auth user at the source address to authenticate his/her identity by supplying a username and password before traffic is allowed to traverse the firewall or enter the VPN tunnel. The security device can use the local database or an external RADIUS, SecurID, or LDAP auth server to perform the authentication check.

NOTE: If a policy requiring authentication applies to a subnet of IP addresses, authentication is required for each IP address in that subnet.
If a host supports multiple auth user accounts (as with a UNIX host running Telnet), then after the security device authenticates the first user, all other users from that host can pass traffic through the security device without being authenticated, having inherited the privileges of the first user.

ScreenOS provides following two authentication schemes:

- Run-time authentication, in which the security device prompts an auth user to log on when it receives HTTP, FTP, or Telnet traffic matching a policy that has authentication enabled

- WebAuth, in which a user must authenticate himself or herself before sending traffic through the security device

Run-Time Authentication

The run-time authentication process proceeds as follows:

1. When the auth user sends an HTTP, an FTP, or a Telnet connection request to the destination address, the security device intercepts the packet and buffers it.
2. The security device sends the auth user a login prompt.
3. The auth user responds to this prompt with his/her username and password.
4. The security device authenticates the auth user's login information.

   If the authentication is successful, a connection is established between the auth user and the destination address.

For the initial connection request, a policy must include one or all of the three following services: Telnet, HTTP, or FTP. Only a policy with one or all of these services is capable of initiating the authentication process. You can use any of the following services in a policy involving user authentication:

- Any (because "any" includes all three required services).
- Telnet, FTP, or HTTP.
- A service group that includes the service or services you want, plus one or more of the three services required to initiate the authentication process (Telnet, FTP, or HTTP). For example, you can create a custom service group named "Login" that supports FTP, NetMeeting, and H.323 services. Then, when you create the policy, specify **Login** as the service.

For any connection following a successful authentication, all services specified in the policy are valid.

NOTE: A policy with authentication enabled does not support DNS (port 53) as the service.

Pre-Policy Check Authentication (WebAuth)

The WebAuth authentication process proceeds as follows:

1. The auth user makes an HTTP connection to the IP address of the WebAuth server.
2. The security device sends the auth user a login prompt.
3. The auth user responds to this prompt with his/her username and password.
4. The security device or an external auth server authenticates the auth user's login information.

If the authentication attempt is successful, the security device permits the auth user to initiate traffic to destinations as specified in policies that enforce authentication via the WebAuth method.

NOTE: For more information about these two user authentication methods, see "Referencing Auth Users in Policies" on page **9**-46.

You can restrict or expand the range of auth users to which the policy applies by selecting a specific user group, local or external user, or group expression. (For information about group expressions, see "Group Expressions" on page **9**-5.) If you do not reference an auth user or user group in a policy (in the WebUI, select the **Allow Any** option), the policy applies to all auth users in the specified auth server.

NOTE: ScreenOS links authentication privileges with the IP address of the host from which the auth user logs on. If the security device authenticates one user from a host behind a NAT device that uses a single IP address for all NAT assignments, then users at other hosts behind that NAT device automatically receive the same privileges.

HA Session Backup

When two security devices are in an NSRP cluster for high availability (HA), you can specify which sessions to backup and which not to backup. For traffic whose sessions you do not want backed up, apply a policy with the HA session backup option disabled. In the WebUI, clear the HA Session Backup check box. In the CLI, use the **no-session-backup** argument in the **set policy** command. By default, security devices in an NSRP cluster back up sessions.

NOTE: IPv6 sessions also support Netscreen Redundancy Protocol (NSRP).

Web Filtering

Web filtering, also called *URL filtering*, enables you to manage Internet access and prevent access to inappropriate Web content. For more information, see "Web Filtering" on page **4**-97.

Logging

When you enable logging in a policy, the security device logs all connections to which that particular policy applies. You can view the logs through either the WebUI or CLI. In the WebUI, click **Reports > Policies > Logging** (for the policy whose log you want to see). In the CLI, use the **get log traffic policy** id_num command.

NOTE: For more information about viewing logs and graphs, see "Monitoring Security Devices" on page 3-55.

Counting

When you enable counting in a policy, the security device counts the total number of bytes of traffic to which this policy applies and records the information in historical graphs. To view the historical graphs for a policy in the WebUI, click **Reports > Policies > Counting** (for the policy whose traffic count you want to see).

Traffic Alarm Threshold

You can set a threshold that triggers an alarm when the traffic permitted by the policy exceeds a specified number of bytes per second, bytes per minute, or both. Because the traffic alarm requires the security device to monitor the total number of bytes, you must also enable the counting feature.

NOTE: For more information, see "Traffic Alarms" on page 3-68.

Schedules

By associating a schedule to a policy, you can determine when the policy is in effect. You can configure schedules to recur or as a one-time event. Schedules provide a powerful tool for controlling the flow of network traffic and enforcing network security. For an example of the latter, if you were concerned about employees transmitting important data outside the company, you might set a policy that blocked outbound FTP-Put and MAIL traffic after normal business hours.

In the WebUI, define schedules on the **Policy > Policy Elements > Schedules** page. In the CLI, use the **set schedule** command.

NOTE: In the WebUI, scheduled policies appear with a gray background to indicate that the current time is not within the defined schedule. When a scheduled policy becomes active, it appears with a white background.

Antivirus Scanning

Some Juniper Networks security devices support an internal AV scanner that you can configure to filter FTP, HTTP, IMAP, POP3, and SMTP traffic. If the embedded AV scanner detects a virus, it either substitutes the packet for a warning message or drops the packet. In both cases, a message reporting the virus is sent to the client that initiated the traffic.

Traffic Shaping

You can set parameters for the control and shaping of traffic for each policy. Table 28 describes the traffic-shaping parameters.

Table 29: Traffic-Shaping Parameters

Parameter	Description
Guaranteed Bandwidth	Guaranteed throughput in kilobits per second (kbps). Traffic below this threshold passes with the highest priority without being subject to any traffic management or shaping mechanism.
Maximum Bandwidth	Secured bandwidth available to the type of connection in kilobits per second (kbps). Traffic beyond this threshold is throttled and dropped. Note: It is advised that you do not use rates less than 10 Kbps. Rates below this threshold lead to dropped packets and excessive retries that defeat the purpose of traffic management.
Traffic Priority	When traffic bandwidth falls between the guaranteed and maximum settings, the security device passes higher priority traffic first, and lower priority traffic only if there is no other higher priority traffic. There are eight priority levels.
DiffServ Codepoint Marking	Differentiated Services (DiffServ) is a system for tagging (or "marking") traffic at a position within a hierarchy of priority. You can map the eight ScreenOS priority levels to the DiffServ system. By default, the highest priority (priority 0) in the ScreenOS system maps to the first three bits (111) in the DiffServ field (see RFC 2474), or the IP precedence field in the TOS byte (see RFC 1349), in the IP packet header. The lowest priority (priority 7) in ScreenOS maps to (000) in the TOS DiffServ system. When you enable DSCP, ScreenOS overwrites the first 3 bits in the ToS byte with the IP precedence priority. When you enable DSCP and set a *dscp-byte* **value**, ScreenOS overwrites the first 6 bits of the ToS byte with the DSCP value. Note: Some devices require that you explicitly enable DSCP marking by setting a system-wide environmental variable. Refer to the installation and configuration guide for your device to find out if it requires that you explicitly enable DSCP marking before using it in policies. If your device requires it, use the following command to enable DSCP marking system wide: **set envar ipsec-dscp-mark = yes**. This variable cannot be set using the WebUI. Use the **unset envar ipsec-dscp-mark** to disable DSCP marking system wide.

NOTE: For a more detailed discussion of traffic management and shaping, see "Traffic Shaping" on page 193.

To change the mapping between the ScreenOS priority levels and the DiffServ system, use the following CLI command:

```
set traffic-shaping ip_precedence number0 number1 number2 number3 number4
number5 number6 number7
```

where number0 is the mapping for priority 0 (the highest priority in the TOS DiffServ system), number1 is the mapping for priority 1, and so on.

To subsume IP precedence into class selector codepoints—that is, to zero out the second three bits in the DiffServ field and thus insure that priority levels you set with **ip_precedence** are preserved and handled correctly by downstream routers—use the following CLI command:

```
set traffic-shaping dscp-class-selector
```

Policies Applied

This section describes the management of policies: viewing, searching, creating, modifying, ordering and reordering, and removing policies.

Viewing Policies

To view policies through the WebUI, click **Policies**. You can sort the displayed policies by source and destination zones by selecting zone names from the From and To lists and then clicking **Go**.

In the CLI, use the **get policy** command, which allows you to display policies according to the following parameters: from zone, to zone, source IP or source address, destination IP or destination address, service, and action.

Keep in mind that source IP and source address are both used to designate the source address of the policy, so you can use only one of them. Furthermore, because the source address name is meaningless without zone, you must designate **the source zone (from** *zone1*) when using source address (**src-address** *addr_name*). The relationship of destination IP and destination address is the same as that of source IP and source address.

Searching Policies

To search policies through the WebUI, click **Policies**, and then click **Search** on the Policy List page. You can find policies by entering information into the Policy Search page, and then clicking **Go**. Customize your search by selecting or entering

- A specific source or destination zone or **All zones** to find policies without the restriction of a zone.
- A specific source or destination address name or the wildcard "*" to match polices without specifying the address.
- A specific service used by a policy or the wildcard "*" to match any policy without specifying a service.

  The **Service** list is sorted by category (Group, Predefined, and Custom), and each category is ordered alphabetically.
- The action of a policy (**Permit**, **Deny**, or **Tunnel**). You can select **Schedule** if the policy you are looking for has a schedule.

Creating Policies

To allow traffic to flow between two zones, you create policies to deny, permit, reject, or tunnel traffic between those zones. You can also create policies to control traffic within the same zone if the security device is the only network device that can route the intrazone traffic between the source and destination addresses referenced in the policy. You can also create global policies, which make use of source and destination addresses in the Global zone address book.

To allow bidirectional traffic between two zones—for example, between the Trust and Untrust zones—you need to create a policy that goes from Trust to Untrust, and then create a second policy from Untrust to Trust. Depending on your needs, the policies can use the same or different IP addresses, only the source and destination addresses are reversed.

You can define policies between any zones that are located within the same system—root or virtual. To define a policy between the root system and a vsys, one of the zones must be a shared zone. (For information about shared zones in relation to virtual systems, see *Virtual Systems*.)

Creating Interzone Policies Mail Service

In this example, you create three policies to control the flow of email traffic.

The first policy allows internal users in the Trust zone to send and retrieve email from a local mail server in the DMZ zone. This policy permits the services MAIL (that is, SMTP) and POP3 originating from the internal users to traverse the Juniper Networks firewall to reach the local mail server.

The second and third policies permit the service MAIL to traverse the firewall between the local mail server in the DMZ zone and a remote mail server in the Untrust zone.

However, before creating policies to control traffic between different security zones, you must first design the environment in which to apply those policies. First, you first bind interfaces to zones and assign the interfaces IP addresses:

- Bind ethernet0/1 to the Trust zone and assign it IP address 10.1.1.1/24.
- Bind ethernet0/2 to the DMZ zone and assign it IP address 1.2.2.1/24.
- Bind ethernet0/3 to the Untrust zone and assign it IP address 1.1.1.1/24.

  All security zones are in the trust-vr routing domain.

Second, you create addresses for use in the policies:

- Define an address in the Trust zone named “corp_net” and assign it IP address 10.1.1.0/24.
- Define an address in the DMZ zone named “mail_svr” and assign it IP address 1.2.2.5/32.
- Define an address in the Untrust zone named “r-mail_svr” and assign it IP address 2.2.2.5/32.

Third, you create a service group named "MAIL-POP3" containing the two predefined services MAIL and POP3.

Fourth, you configure a default route in the trust-vr routing domain pointing to the external router at 1.1.1.250 through ethernet0/3.

After completing steps 1 through 4, you can then create the policies necessary to permit the transmission, retrieval, and delivery of email in and out of your protected network.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet0/2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select this option when present)
   IP Address/Netmask: 1.2.2.1/24

   Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: corp_net
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: mail_svr
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.5/32
   Zone: DMZ

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: r-mail_svr
IP Address/Domain Name:
    IP/Netmask: (select), 2.2.2.5/32
Zone: Untrust

3. **Service Group**

Policy > Policy Elements > Services > Groups: Enter the following group name, move the following services, then click **OK**:

Group Name: MAIL-POP3

Select **MAIL** and use the < < button to move the service from the Available Members column to the Group Members column.

Select **POP3** and use the < < button to move the service from the Available Members column to the Group Members column.

4. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
    Interface: ethernet0/3
    Gateway IP Address: 1.1.1.250

5. **Policies**

Policy > Policies > (From: Trust, To: DMZ) > New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), corp_net
Destination Address:
    Address Book Entry: (select), mail_svr
Service: Mail-POP3
Action: Permit

Policy > Policies > (From: DMZ, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), mail_svr
Destination Address:
    Address Book Entry: (select), r-mail_svr
Service: MAIL
Action: Permit

Policy > Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), r-mail_svr
Destination Address:
    Address Book Entry: (select), mail_svr
Service: MAIL
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 10.1.1.1/24
set interface ethernet0/2 zone dmz
set interface ethernet0/2 ip 1.2.2.1/24
set interface ethernet0/3 zone untrust
set interface ethernet0/3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address trust corp_net 10.1.1.0/24
set address dmz mail_svr 1.2.2.5/32
set address untrust r-mail_svr 2.2.2.5/32
```

3. **Service Group**

```
set group service MAIL-POP3
set group service MAIL-POP3 add mail
set group service MAIL-POP3 add pop3
```

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet0/3 gateway 1.1.1.250
```

5. **Policies**

```
set policy from trust to dmz corp_net mail_svr MAIL-POP3 permit
set policy from dmz to untrust mail_svr r-mail_svr MAIL permit
set policy from untrust to dmz r-mail_svr mail_svr MAIL permit
save
```

Creating an Interzone Policy Set

A small software firm, ABC Design, has divided its internal network into two subnets, and both are in the Trust zone. These two subnets are:

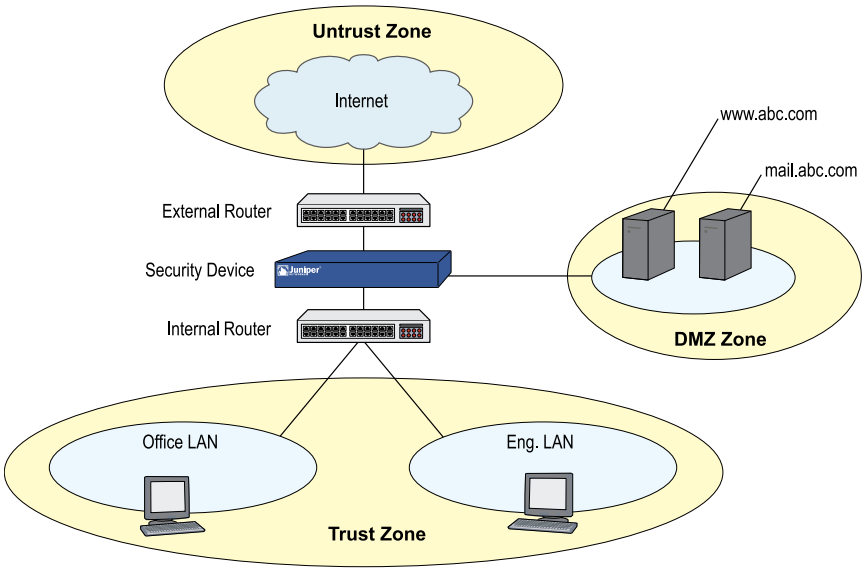
- Engineering (with the defined address "Eng")
- The rest of the company (with the defined address "Office")

ABC Design also has a DMZ zone for its Web and mail servers.

The following example presents a typical set of policies for the following users:

- “Eng” can use all the services for outbound traffic except FTP-Put, IMAP, MAIL, and POP3.
- “Office” can use email and access the Internet, provided they authenticate themselves via WebAuth. (For information about WebAuth user authentication, see “ Authentication Users” on page **9**-45.)
- Everyone in the Trust zone can access the Web and mail servers in the DMZ zone.
- A remote mail server in the Untrust zone can access the local mail server in the DMZ zone.
- There is also a group of system administrators (with the user-defined address “sys-admins”) who have complete user and administrative access to the servers in the DMZ zone.

Figure 61: Interzone Policy Set



It is assumed that you have already configured the interfaces, addresses, service groups, and routes that must be in place. For more information about configuring these, see “Interfaces” on page 35, “ Addresses” on page 103, “ Service Groups” on page 138, and *Routing*.

Table 30: Configured Policies

From Zone - Src Addr	To Zone - Dest Addr	Service	Action
Trust - Any	Untrust - Any	Com (service group: FTP-Put, IMAP, MAIL, POP3)	Reject
Trust - Eng	Untrust - Any	Any	Permit
Trust - Office	Untrust - Any	Internet (service group: FTP-Get, HTTP, HTTPS)	Permit (+ WebAuth)

Table 30: Configured Policies *(continued)*

From Zone - Src Addr	To Zone - Dest Addr	Service	Action
Untrust - Any	DMZ - mail.abc.com	MAIL	Permit
Untrust - Any	DMZ - www.abc.com	Web (service group: HTTP, HTTPS)	Permit
Trust - Any	DMZ - mail.abc.com	Email (service group: IMAP, MAIL, POP3)	Permit
Trust - Any	DMZ - www.abc.com	Internet (service group: FTP-Get, HTTP, HTTPS)	Permit
Trust - sys-admins	DMZ - Any	Any	Permit
DMZ - mail.abc.com	Untrust - Any	MAIL	Permit

NOTE: The default policy is to deny all.

WebUI

1. **From Trust to Untrust**

   Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Eng
   Destination Address:
   Address Book Entry: (select), Any
   Service: ANY
   Action: Permit

   Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Office
   Destination Address:
   Address Book Entry: (select), Any
   Service: Internet
   Action: Permit

   > Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   Authentication: (select)
   WebAuth: (select)

NOTE: "Internet" is a service group with the following members: FTP-Get, HTTP, and HTTPS.

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: Com
Action: Reject
Position at Top: (select)

NOTE: "Com" is a service group with the following members: FTP-Put, MAIL, IMAP, and POP3.
For traffic from the Untrust zone to the Trust zone, the default deny policy denies everything.

2. **From Untrust to DMZ**

Policy > Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), mail.abc.com
Service: MAIL
Action: Permit

Policy > Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), www.abc.com
Service: Web
Action: Permit

NOTE: "Web" is a service group with the following members: HTTP and HTTPS.

3. **From Trust to DMZ**

Policy > Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:

Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), mail.abc.com
Service: e-mail
Action: Permit

NOTE: "e-mail" is a service group with the following members: MAIL, IMAP, and POP3.

Policy > Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), www.abc.com
Service: Internet
Action: Permit

Policy > Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), sys-admins
Destination Address:
Address Book Entry: (select), Any
Service: ANY
Action: Permit

4. **From DMZ to Untrust**

Policy > Policies > (From: DMZ, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), mail.abc.com
Destination Address:
Address Book Entry: (select), Any
Service: MAIL
Action: Permit

CLI

1. **From Trust to Untrust**

```
set policy from trust to untrust eng any any permit
set policy from trust to untrust office any Internet permit webauth
set policy top from trust to untrust any any Com reject
```

NOTE: "Internet" is a service group with the following members: FTP-Get, HTTP, and HTTPS.
"Com" is a service group with the following members: FTP-Put, MAIL, IMAP, and POP3.

2. **From Untrust to DMZ**

```
set policy from untrust to dmz any mail.abc.com mail permit
set policy from untrust to dmz any www.abc.com Web permit
```

NOTE: "Web" is a service group with the following members: HTTP and HTTPS.

3. **From Trust to DMZ**

```
set policy from trust to dmz any mail.abc.com e-mail permit
set policy from trust to dmz any www.abc.com Internet permit
set policy from trust to dmz sys-admins any any permit
```

NOTE: "e-mail" is a service group with the following members: MAIL, IMAP, and POP3.
"Internet" is a service group with the following members: FTP-Get, HTTP, and HTTPS.

4. **From DMZ to Untrust**

```
set policy from dmz to untrust mail.abc.com any mail permit
save
```

Creating Intrazone Policies

In this example, you create an intrazone policy to permit a group of accountants access to a confidential server on the corporate LAN in the Trust zone. You first bind ethernet0/1 to the Trust zone and give it IP address 10.1.1.1/24. You then bind ethernet0/2 to the Trust zone and assign it IP address 10.1.5.1/24. You enable intrazone blocking in the Trust zone. Next, you define two addresses—one for a server on which the company stores its financial records (10.1.1.100/32) and another for the subnet on which hosts for the accounting department are located (10.1.5.0/24). You then create an intrazone policy to permit access to the server from those hosts.

WebUI

1. **Trust Zone—Interfaces and Blocking**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)

IP Address/Netmask: 10.1.1.1/24
Select the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.5.1/24
Select the following, then click **OK**:
Interface Mode: NAT

Network > Zones > Edit (for Trust): Enter the following, then click **OK**:

Block Intra-Zone Traffic: (select)

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Hamilton
IP Address/Domain Name:
    IP/Netmask: (select), 10.1.1.100/32
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: accounting
IP Address/Domain Name:
    IP/Netmask: (select), 10.1.5.0/24
Zone: Trust

3. **Policy**

Policy > Policies > (From: Trust, To: Trust) > New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), accounting
Destination Address:
    Address Book Entry: (select), Hamilton
Service: ANY
Action: Permit

CLI

1. **Trust Zone—Interfaces and Blocking**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 10.1.1.1/24
set interface ethernet0/1 nat
set interface ethernet0/2 zone trust
```

```
set interface ethernet0/2 ip 10.1.5.1/24
set interface ethernet0/2 nat
set zone trust block
```

2. **Addresses**

```
set address trust Hamilton 10.1.1.100/32
set address trust accounting 10.1.5.0/24
```

3. **Policy**

```
set policy from trust to trust accounting Hamilton any permit
save
```

Creating a Global Policy

In this example, you create a global policy so that every host in every zone can access the company website, which is www.juniper.net. Using a global policy is a convenient shortcut when there are many security zones. In this example, one global policy accomplishes what n interzone policies would have accomplished (where n = number of zones).

NOTE: To use a domain name instead of an IP address, be sure to have DNS service configured on the security device.

WebUI

1. **Global Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: server1
   IP Address/Domain Name:
   Domain Name: (select), www.juniper.net
   Zone: Global

2. **Policy**

   Policy > Policies > (From: Global, To: Global) > New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), server1
   Service: HTTP
   Action: Permit

CLI

1. **Global Address**

```
set address global server1 www.juniper.net
```

2. **Policy**

```
set policy global any server1 http permit
save
```

Entering a Policy Context

When configuring a policy through the CLI, after you first create a policy, you can then enter the context of the policy to make additions and modifications. For example, perhaps you first create the following policy:

```
set policy id 1 from trust to untrust host1 server1 HTTP permit attack HIGH:HTTP:SIGS
 action close
```

If you want to make some changes to the policy, such as adding another source or destination address, another service, or another attack group, you can enter the context for policy 1 and then enter the pertinent commands:

```
set policy id 1
device(policy:1)-> set src-address host2
device(policy:1)-> set dst-address server2
device(policy:1)-> set service FTP
device(policy:1)-> set attack CRITICAL:HTTP:SIGS
```

You can also remove multiple items for a single policy component as long as you do not remove them all. For example, you can remove server2 from the above configuration, but not server2 and server1 because then no destination address would remain.

Multiple Items per Policy Component

ScreenOS allows you to add multiple items to the following components of a policy:

- Source address
- Destination address
- Service
- Attack group

In ScreenOS releases prior to 5.0.0, the only way to have multiple source and destination addresses or services is to first create an address or service group with multiple members and then reference that group in a policy. You can still use address and service groups in policies in ScreenOS 5.0.0. In addition, you can simply add extra items directly to a policy component.

NOTE: If the first address or service referenced in a policy is “Any,” you cannot logically add anything else to it. ScreenOS prevents this kind of misconfiguration and displays an error message should it occur.

To add multiple items to a policy component, do either of the following:

WebUI

To add more addresses and services, click the **Multiple** button next to the component to which you want to add more items. To add more attack groups, click the **Attack Protection** button. Select an item in the “Available Members” column, and then use the < < key to move it to the “ Active Members” column. You can repeat this action with other items. When finished, click **OK** to return to the policy configuration page.

CLI

Enter the policy context with the following command:

```
set policy id number
```

Then use one of the following commands as appropriate:

```
device(policy:number)-> set src-address string
device(policy:number)-> set dst-address string
device(policy:number)-> set service string
device(policy:number)-> set attack string
```

Setting Address Negation

You can configure a policy so that it applies to all addresses except the one specified as either the source or destination. For example, you might want to create a policy that permits Internet access to everyone except the “P-T_contractors” address group. To accomplish this, you can use the address negation option.

In the WebUI, this option is available on the pop-up that appears when you click **Multiple** next to either **Source Address** or **Destination Address** on the policy configuration page.

In the CLI, you insert an exclamation point (!) immediately before source or destination address.

NOTE: Address negation occurs at the policy component level, applying to all items in the negated component. However, you cannot enforce address negation with a wildcard address.

In this example, you create an intrazone policy that allows all addresses in the Trust zone access to all FTP servers except to an FTP server named “vulcan” , which engineering uses to post functional specifications for one another.

However, before creating the policy, you must first design the environment in which to apply it. First, you enable intrazone blocking for the Trust zone. Intrazone blocking requires a policy lookup before the security device passes traffic between two interfaces bound to the same zone.

Second, you bind two interfaces to the Trust zone and assign them IP addresses:

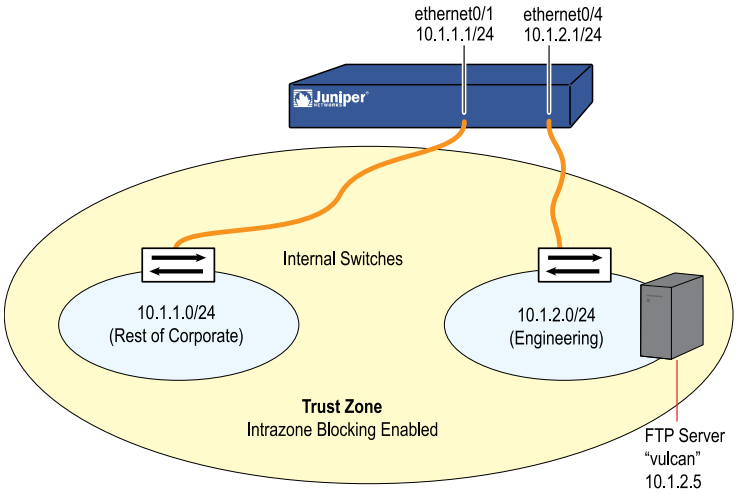
- You bind ethernet0/1 to the Trust zone and assign it IP address 10.1.1.1/24.
- You bind ethernet0/4 to the Trust zone and assign it IP address 10.1.2.1/24.

Third, you create an address (10.1.2.5/32) for the FTP server named "vulcan" in the Trust zone.

After completing these two steps, you can then create the intrazone policies.

NOTE: You do not have to create a policy for the engineering department to reach their FTP server because the engineers are also in the 10.1.2.0/24 subnet and do not have to cross the Juniper Networks firewall to reach their own server.

Figure 62: Intrazone Policies Negation



WebUI

1. **Intrazone Blocking**

   Network > Zones > Edit (for Trust): Enter the following, then click **OK**:

   Virtual Router Name: trust-vr
   Block Intra-Zone Traffic: (select)

2. **Trust Zone Interfaces**

Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24
Select the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet0/4): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.2.1/24
Select the following, then click **OK**:
Interface Mode: NAT

3. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: vulcan
IP Address/Domain Name:
    IP/Netmask: (select), 10.1.2.5/32
Zone: Trust

4. **Policy**

Policy > Policies > (From: Trust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), vulcan

> Click **Multiple**, select **Negate Following**, then click **OK** to return to the basic policy configuration page.

Service: FTP
Action: Permit

CLI

1. **Intrazone Blocking**

```
set zone trust block
```

2. **Trust Zone Interfaces**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 10.1.1.1/24
set interface ethernet0/1 nat
```

```
set interface ethernet0/4 zone trust
set interface ethernet0/4 ip 10.1.2.1/24
set interface ethernet0/1 nat
```

3. **Address**

```
set address trust vulcan 10.1.2.5/32
```

4. **Policy**

```
set policy from trust to trust any !vulcan ftp permit
save
```

Modifying and Disabling Policies

After you create a policy, you can always return to it to make modifications. In the WebUI, click the **Edit** link in the Configure column for the policy that you want to change. In the Policy configuration page that appears for that policy, make your changes, then click **OK**. In the CLI, use the **set policy** command.

ScreenOS also provides a means for enabling and disabling policies. By default, a policy is enabled. To disable it, do the following:

WebUI

Policies: Clear the Enable check box in the Configure column for the policy that you want to disable.

The row of text for a disabled policy appears as grey.

CLI

```
set policy id id_num disable
save
```

NOTE: To enable the policy again, select **Enable** in the Configure column for the policy that you want to enable (WebUI), or type **unset policy id** id_num **disable** (CLI).

Policy Verification

ScreenOS offers a tool for verifying that the order of policies in the policy list is valid. It is possible for one policy to eclipse, or "shadow," another policy. Consider the following example:

```
set policy id 1 from trust to untrust any any HTTP permit
set policy id 2 from trust to untrust any dst-A HTTP deny
```

Because the security device performs a policy lookup starting from the top of the list, when it finds a match for traffic received, it does not look any lower in the policy

list. In the above example, the security device never reaches policy 2 because the destination address “any” in policy 1 includes the more specific “ dst-A” address in policy 2. When an HTTP packet arrives at the security device from an address in the Trust zone bound for dst-A in the Untrust zone, the security device always first finds a match with policy 1.

To correct the above example, you can simply reverse the order of the policies, putting the more specific one first:

```
set policy id 2 from trust to untrust any dst-A HTTP deny
set policy id 1 from trust to untrust any any HTTP permit
```

Of course, this example is purposefully simple to illustrate the basic concept. In cases where there are dozens or hundreds of policies, the eclipsing of one policy by another might not be so easy to spot. To check if there is any policy shadowing in your policy list, you can use the following CLI command:

```
exec policy verify
```

This command reports the shadowing and shadowed policies. It is then the admin’s responsibility to correct the situation.

NOTE: The concept of policy shadowing refers to the situation where a policy higher in the policy list always takes effect before a subsequent policy. Because the policy lookup always uses the first policy it finds that matches the five-part tuple of source and destination zone, source and destination address, and service type, if another policy applies to the same tuple (or a subset of the tuple), the policy lookup uses the first policy in the list and never reaches the second one.

The policy verification tool cannot detect the case where a combination of policies shadows another policy. In the following example, no single policy shadows policy 3; however, policies 1 and 2 together do shadow it:

```
set group address trust grp1 add host1
set group address trust grp1 add host2
set policy id 1 from trust to untrust host1 server1 HTTP permit
set policy id 2 from trust to untrust host2 server1 HTTP permit
set policy id 3 from trust to untrust grp1 server1 HTTP deny
```

Reordering Policies

The security device checks all attempts to traverse the firewall against policies, beginning with the first one listed in the policy set for the appropriate list (see “Policy Set Lists” on page 163) and moving through the list. Because the security device applies the action specified in the policy to the first matching policy in the list, you must arrange them from the most specific to the most general. (Whereas a specific policy does not preclude the application of a more general policy located down the list, a general policy appearing before a specific one does.)

By default, a newly created policy appears at the bottom of a policy set list. There is an option that allows you to position a policy at the top of the list instead. In the

Policy configuration page in the WebUI, select **Position at Top**. In the CLI, add the key word **top** to the **set policy** command: **set policy top** …

To move a policy to a different position in the list, do either of the following:

WebUI

There are two ways to reorder policies in the WebUI: by clicking the circular arrows or by clicking the single arrow in the Configure column for the policy you want to move.

If you click the circular arrows:

A User Prompt dialog box appears.

To move the policy to the very end of the list, enter <-1>. To move it up in the list, enter the ID number of the policy above which you want to move the policy in question.

Click **OK** to execute the move.

If you click the single arrow:

A Policy Move page appears displaying the policy you want to move and a table displaying the other policies.

In the table displaying the other policies, the first column, Move Location, contains arrows pointing to various locations where you can move the policy. Click the arrow that points to the location in the list where you want to move the policy.

The Policy List page reappears with the policy you moved in its new position.

CLI

```
set policy move id_num { before | after } number
save
```

Removing a Policy

In addition to modifying and repositioning a policy, you can also delete it. In the WebUI, click **Remove** in the Configure column for the policy that you want to remove. When the system message prompts for confirmation to proceed with the removal, click **Yes**. In the CLI, use the **unset policy** id_num command.1

Chapter 8

Traffic Shaping

This chapter discusses the various ways you can use your Juniper Networks security device to manage limited bandwidth without compromising quality and availability of the network to all of your users. It contains the following sections:

Traffic shaping is the allocation of the appropriate amount of network bandwidth to every user and application on an interface. The appropriate amount of bandwidth is defined as cost-effective carrying capacity at a guaranteed Quality of Service (QoS). You use a security device to shape traffic by creating policies and by applying appropriate rate controls to each class of traffic going through the device.

Managing Bandwidth at the Policy Level

To classify traffic, you create policies and specify the amount of guaranteed bandwidth, maximum bandwidth, and the priority for each class of traffic. Guaranteed bandwidth and maximum bandwidth are not strictly policy-based. With multiple physical interfaces in the egress zone, bandwidth is based both on policy and on total available egress physical interface bandwidth. The physical bandwidth of every interface is allocated to the guaranteed bandwidth parameter for all traffic-shaping policies. If there is any bandwidth left over, it is sharable by any other traffic. In other words, each policy gets its guaranteed bandwidth and shares whatever is left over, according to its priority (up to the limit of its maximum bandwidth specification), with all other policies.

The traffic-shaping function applies to traffic from all policies. If you turn off traffic shaping for a specific policy while traffic shaping is still turned on for other policies, the system applies a default traffic-shaping policy to that particular policy, with the following parameters:

- Guaranteed bandwidth 0
- Unlimited maximum bandwidth
- Priority of 7 (the lowest priority setting)

NOTE: You can enable mapping of priority levels to the Differentiated Services Codepoint (DSCP) marking system. For more information about DSCP marking, see “Traffic Shaping” on page 212.

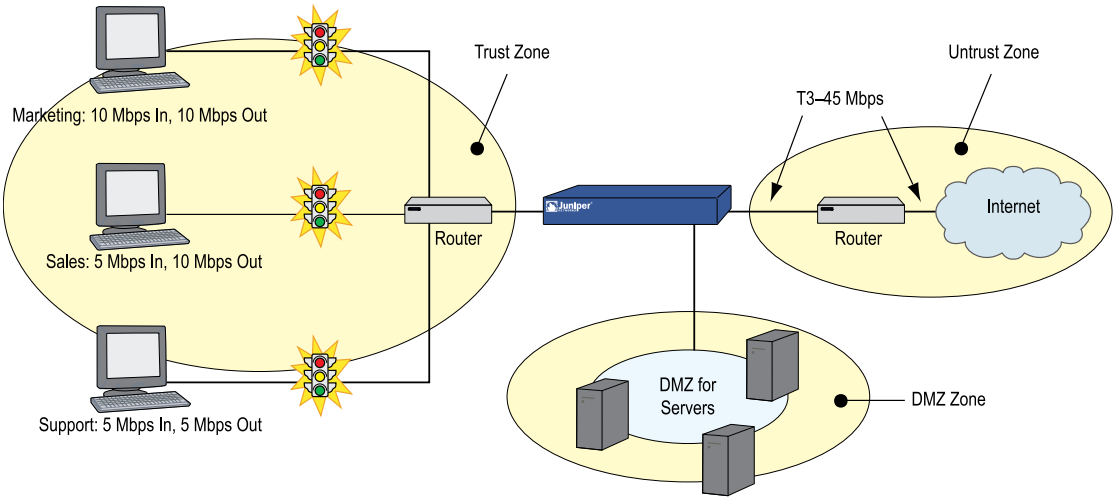
You can turn off traffic shaping system wide using the **set traffic-shaping mode off** command. Use the **set traffic-shaping mode on** command to turn on shaping on an interface. You can set traffic shaping to automatic in the WebUI: **Configuration > Advanced > Traffic Shaping**. In automatic mode, if traffic shaping in enabled on the policy, the device turns on traffic shaping when traffic hits the device and turns off traffic shaping when no traffic hits the device.

NOTE: ScreenOS supports traffic-shaping on non-ASIC devices. You cannot perform traffic-shaping on ASIC-based ISG1000, ISG2000, and NS5000 devices.

Setting Traffic Shaping

In this example, you partition 45Mbps of bandwidth on a T3 interface among three departments on the same subnet. The interface ethernet0/1 is bound to the Trust zone, and ethernet0/3 is bound to the Untrust zone.

Figure 63: Traffic Shaping



WebUI

1. **Bandwidth on Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

   Traffic Bandwidth: 45000

NOTE: If you do not specify bandwidth settings on an interface, the security device uses the available physical bandwidth.

   Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

   Traffic Bandwidth: 45000

2. **Bandwidth in Policies**

   Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Name: Marketing Traffic Shaping
   Source Address:
   Address Book Entry: (select), Marketing
   Destination Address:
   Address Book Entry: (select), Any
   Service: Any
   Action: Permit
   VPN Tunnel: None

NOTE: You can also enable traffic shaping in policies referencing VPN tunnels.

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Traffic Shaping: (select)
Guaranteed Bandwidth: 10000
Maximum Bandwidth: 15000

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: Sales Traffic Shaping Policy
Source Address:
    Address Book Entry: (select), Sales
Destination Address:
    Address Book Entry: (select), Any
Service: Any
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Traffic Shaping: (select)
Guaranteed Bandwidth: 10000
Maximum Bandwidth: 10000

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: Support Traffic Shaping Policy
Source Address:
    Address Book Entry: (select), Support
Destination Address:
    Address Book Entry: (select), Any
Service: Any
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Traffic Shaping: (select)
Guaranteed Bandwidth: 5000
Maximum Bandwidth: 10000

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Name: Allow Incoming Access to Marketing
Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Marketing
Service: Any

Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Traffic Shaping: (select)
Guaranteed Bandwidth: 10000
Maximum Bandwidth: 10000

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Name: Allow Incoming Access to Sales
Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Sales
Service: Any
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Traffic Shaping: (select)
Guaranteed Bandwidth: 5000
Maximum Bandwidth: 10000

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Name: Allow Incoming Access to Support
Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Support
Service: Any
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Traffic Shaping: (select)
Guaranteed Bandwidth: 5000
Maximum Bandwidth: 5000

CLI

To enable traffic shaping by policy:

1. **Bandwidth on Interfaces**

```
set interface ethernet0/1 bandwidth 45000
set interface ethernet0/3 bandwidth 45000
```

NOTE: If you do not specify bandwidth settings on an interface, the security device uses the available physical bandwidth.

2. **Bandwidth in Policies**

```
set policy name “Marketing Traffic Shaping” from trust to untrust marketing any
 any permit traffic gbw 10000 priority 0 mbw 15000
set policy name “ Sales Traffic Shaping Policy” from trust to untrust sales any
 any permit traffic gbw 10000 priority 0 mbw 10000
set policy name “ Support Traffic Shaping Policy” from trust to untrust support
 any any permit traffic gbw 5000 priority 0 mbw 10000
set policy name “ Allow Incoming Access to Marketing” from untrust to trust
any marketing any permit traffic gbw 10000 priority 0 mbw 10000
set policy name “ Allow Incoming Access to Sales” from untrust to trust any
sales any permit traffic gbw 5000 priority 0 mbw 10000
set policy name “ Allow Incoming Access to Support” from untrust to trust any
 support any permit traffic gbw 5000 priority 0 mbw 5000
save
```

Setting Service Priorities

The traffic-shaping feature on Juniper Networks security devices allows you to perform priority queuing on the bandwidth that is not allocated to guaranteed bandwidth, or unused guaranteed bandwidth. Priority queuing is a feature that allows all your users and applications to have access to available bandwidth as they need it, while ensuring that important traffic can get through, if necessary at the expense of less important traffic. Queuing allows the security device to buffer traffic in up to eight different priority queues. These eight queues are:

- High priority
- 2nd priority
- 3rd priority
- 4th priority
- 5th priority
- 6th priority
- 7th priority
- Low priority (default)

The priority setting for a policy means that the bandwidth not already guaranteed to other policies is queued on the basis of high priority first and low priority last. Policies with the same priority setting compete for bandwidth in a round robin fashion. The security device processes all of the traffic from all of the policies with high priority before processing any traffic from policies with the next lower priority setting, and so on, until all traffic requests have been processed. If traffic requests exceed available bandwidth, the lowest priority traffic is dropped.

CAUTION: Be careful not to allocate more bandwidth than the interface can support. The policy configuration process does not prevent you from creating unsupported policy configurations. You can lose data if the guaranteed bandwidth on contending policies surpasses the traffic bandwidth set on the interface.

If you do not allocate any guaranteed bandwidth, then you can use priority queuing to manage all of traffic on your network. That is, all high priority traffic is sent before any 2nd priority traffic is sent, and so on. The security device processes low priority traffic only after all other traffic has been processed.

Traffic Shaping for an ALG

In ScreenOS, when you create a policy to enable traffic shaping, the session created by that policy will perform traffic-shaping functions. All the sessions derived from the same policy will share the same quality-of-service (QoS) parameters for traffic shaping. There are four traffic-shaping parameters you can configure:

- Priority
- Guaranteed bandwidth (GBW)
- Maximum bandwidth (MBW)
- Policing bandwidth (PBW)

When you create a policy for traffic shaping, you should configure the GBW and MBW for the traffic in a manner that voice quality is guarnateed.

For example, consider the following cases where a SIP phone call requires 64 Kbps bandwidth for ensuirng voice quality.

Case1:

You configure the policy for a SIP phone with address 1.1.1.1:

```
set policy from trust to untrust 1.1.1.1 any sip permit traffic priority 0 gbw 64
```

The sip phone gets a guaranteed bandwidth of 64kbps and hence ensures voice quality.

Case 2:

You configure the policy for a group of five phones with a group address 1.1.1.0:

```
set policy from trust to untrust 1.1.1.0 any sip permit traffic priority 0 mbw 320
```

Here, we define the maximum bandwidth of the group to 320 kpbs. The available bandwidth is shared among the five phones. The voice quality can be guaranteed only if the number of concurrent calls is less than or equal to 5, since each phone requires a bandwidth of 64kpbs. If the number of calls exceed 5, voice quality cannot be guaranteed.

Refer to “Setting Traffic Shaping” on page 234 for more information on setting traffic shaping properties.

If you enable an ALG and create a traffic-shaping policy for that ALG's traffic, the policy creates a control session when the voice over Internet Protocol (VoIP) traffic reaches the ALG. The control session inherits the parameters of the traffic-shaping policy. However, when the voice data for that VoIP traffic reaches the ALG, the ALG creates a data/media session that does not have its own associated policy. Instead, the session uses the default policy defined in the security device. This slows down the VoIP traffic, because the data/media sessions will have the lowest priority settings in the default policy, and the traffic-shaping policy will be unable to perform traffic shaping on the data.

The solution for this is for the control/signal session and the data/media session to share the same policy, so that both sessions share the same QoS parameters as defined in the policy. In order to achieve this association, ScreenOS adds a policy pointer to the gate of the ALG when the gate is created which enables the data/media session to share the QoS parameters of the control/signal session.

Setting Priority Queuing

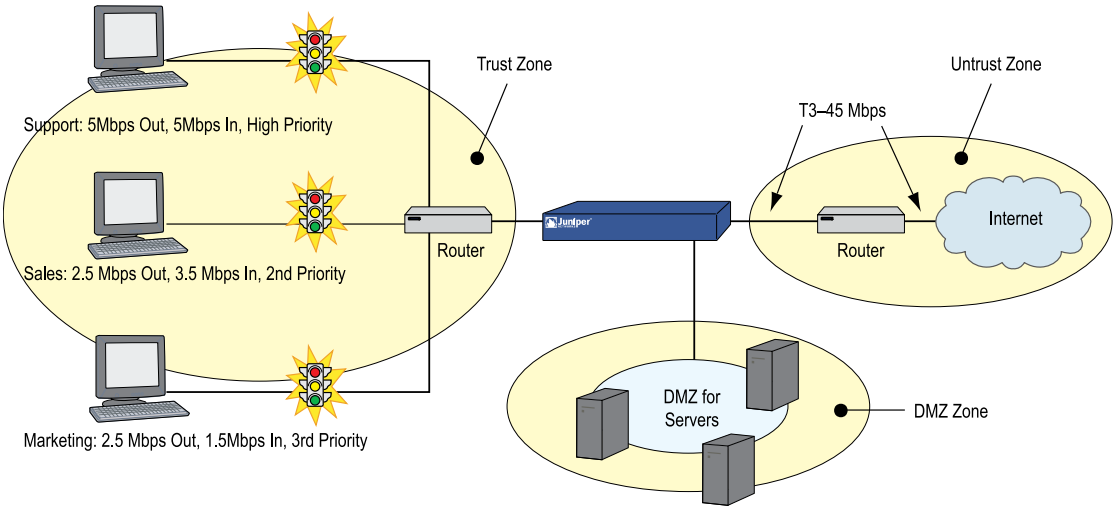
In this example, you configure the guaranteed and maximum bandwidth (in Mbps) for three departments—Support, Sales, and Marketing—as shown in Table 31 on page 240.

Table 31: Maximum Bandwidth Configuration

	Outbound Guaranteed	Inbound Guaranteed	Combined Guaranteed	Priority
Support	5	5	10	High
Sales	2.5	3.5	6	2
Marketing	2.5	1.5	4	3
Total	10	10	20	

If all three departments send and receive traffic concurrently through the firewall, the security device must allocate 20 Mbps of bandwidth to fulfill the guaranteed policy requirements. The interface ethernet0/1 is bound to the Trust zone, and ethernet0/3 is bound to the Untrust zone.

Figure 64: Priority Queuing



WebUI

1. **Bandwidth on Interfaces**

   Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

   Traffic Bandwidth: 40000

   Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

   Traffic Bandwidth: 40000

2. **Bandwidth in Policies**

   Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Name: Sup-out
   Source Address:
   Address Book Entry: (select), Support
   Destination Address:
   Address Book Entry: (select), Any
   Service: Any
   Action: Permit

   > Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   Traffic Shaping: (select)
   Guaranteed Bandwidth: 5000
   Maximum Bandwidth: 40000
   Traffic Priority: High priority
   DiffServ Codepoint Marking: (select)

NOTE: Differentiated Services (DS) is a system for tagging (or "marking") traffic at a position within a hierarchy of priority. DSCP marking maps the ScreenOS priority level of the policy to the first three bits of codepoint in the DS field in the IP packet header. For more information about DSCP marking, see "Traffic Shaping" on page 212.

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: Sal-out
Source Address:
    Address Book Entry: (select), Sales
Destination Address:
    Address Book Entry: (select), Any
Service: Any
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Traffic Shaping: (select)
Guaranteed Bandwidth: 2500
Maximum Bandwidth: 40000
Traffic Priority: 2nd priority
  DiffServ Codepoint Marking: Enable

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: Mar-out
Source Address:
    Address Book Entry: (select), Marketing
Destination Address:
    Address Book Entry: (select), Any
Service: Any
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Traffic Shaping: (select)
Guaranteed Bandwidth: 2500
Maximum Bandwidth: 40000
Traffic Priority: 3rd priority
DiffServ Codepoint Marking: (select)

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Name: Sup-in
Source Address:
    Address Book Entry: (select), Any
Destination Address:

Address Book Entry: (select), Support
Service: Any
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Traffic Shaping: (select)
Guaranteed Bandwidth: 5000
Maximum Bandwidth: 40000
Traffic Priority: High priority
DiffServ Codepoint Marking: (select)

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Name: Sal-in
Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Sales
Service: Any
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Traffic Shaping: (select)
Guaranteed Bandwidth: 3500
Maximum Bandwidth: 40000
Traffic Priority: 2nd priority
DiffServ Codepoint Marking: (select)

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Name: Mar-in
Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Marketing
Service: Any
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Traffic Shaping: (select)
Guaranteed Bandwidth: 1500
Maximum Bandwidth: 40000
Traffic Priority: 3rd priority
DiffServ Codepoint Marking: (select)

CLI

1. **Bandwidth on Interfaces**

```
set interface ethernet0/1 bandwidth 40000
set interface ethernet0/3 bandwidth 40000
```

2. **Bandwidth in Policies**

```
set policy name sup-out from trust to untrust support any any permit traffic gbw
 5000 priority 0 mbw 40000 enable
set policy name sal-out from trust to untrust sales any any permit traffic gbw
2500 priority 2 mbw 40000 dscp enable
```

NOTE: Some devices require that you explicitly enable DSCP marking by setting a system-wide environmental variable. Refer to the installation and configuration guide for your device to find out if it requires that you explicitly enable DSCP marking before using it in policies. If your device requires it, use the following command to enable DSCP marking system wide: **set envar ipsec-dscp-mark = yes**. This variable cannot be set using the WebUI. Use the **unset envar ipsec-dscp-mark** to disable DSCP marking system wide.

```
set policy name mar-out from trust to untrust marketing any any permit traffic
gbw 2500 priority 3 mbw 40000 dscp enable
set policy name sup-in from untrust to trust any support any permit traffic gbw
5000 priority 0 mbw 40000 dscp enable
set policy name sal-in from untrust to trust any sales any permit traffic gbw 3500
 priority 2 mbw 40000 dscp enable
set policy name mar-in from untrust to trust any marketing any permit traffic gbw
 1500 priority 3 mbw 40000 dscp enable
save
```

Ingress Policing

Ingress policing is traffic control at the ingress side of the security device. By constraining the flow of traffic at the point of ingress, traffic exceeding your bandwidth setting is dropped with minimal processing, conserving system resources. You can configure ingress policing at the interface level and in security policies.

You configure ingress policing on an interface by setting a maximum bandwidth (the **mbw** keyword). The following command, for example, limits bandwidth on ethernet0/1, the ingress interface, to 22 Mbps:

```
set interface ethernet0/1 bandwidth ingress mbw 22000
```

Incoming traffic on ethernet0/1 exceeding this bandwidth is dropped. If you set traffic shaping at the interface, you must also set traffic-shaping mode to on (**set traffic-shaping mode on**).

To apply ingress policing to a specific application, however, requires a policy. The following command creates a policy called *my_ftp* that limits FTP bandwidth on the ingress side of the security device to 10 Mbps:

```
set policy my_ftp from untrust to trust any any ftp permit traffic pbw 10000
```

Incoming FTP traffic exceeding the configured policing bandwidth (the **pbw** keyword) is dropped. You can also set **mbw** in the policy, but at the policy level **mbw** applies only to the egress side of traffic flow—traffic exceeding your configured rate is still processed, and is dropped only at the egress side (see Figure 66 on page 247). You can configure **mbw** or **pbw** in a policy, but not both.

Configuration and enforcement of ingress policing on virtual interfaces is the same as on physical interfaces, with the one exception that you can also configure guaranteed bandwidth (the **gbw** keyword) on virtual interfaces (see "Policy-Level Traffic Shaping" on page 247). On physical interfaces, guaranteed bandwidth is the same as maximum bandwidth.

NOTE: Ingress policing on tunnel interfaces is enforced after the encrypted packets are decrypted by the VPN engine.

Shaping Traffic on Virtual Interfaces

In the context of traffic shaping, the term *virtual interfaces* refers only to subinterfaces and tunnel interfaces—not to other types of virtual interfaces, such as virtual security interfaces (VSI), or aggregate or redundant interfaces. You cannot configure shaping parameters in policies created in a vsys. Similarly, bandwidth cannot be shaped on interfaces owned (inherited) by a user-created vsys. See "Virtual Systems" on page 1677 for more information.

Traffic shaping (as distinct from ingress policing) concerns traffic management at the egress side of the security device. As with physical interfaces, you shape traffic on virtual interfaces by setting bandwidth values at the interface level, and in policies.

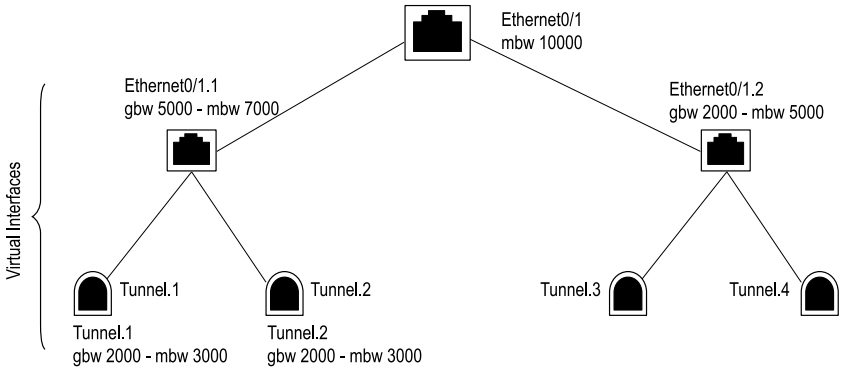
Interface-Level Traffic Shaping

Traffic shaping at the interface level is control of the minimum and maximum rate of traffic flow on a specific interface. You control minimum bandwidth by specifying a guaranteed bandwidth (**gbw**). This means that no matter what else happens on the device, this minimum rate is guaranteed to the appropriate traffic. The maximum bandwidth (**mbw**) you set establishes the rate traffic can never exceed. By default, guaranteed bandwidth on a physical interface is the carrying capacity (maximum bandwidth) of the interface; therefore, you cannot set guaranteed bandwidth on the physical interface.

In the context of traffic shaping, the term *virtual interfaces* refers to subinterfaces bound to physical interfaces and, by extension, tunnel interfaces bound to those subinterfaces—creating a hierarchy of interfaces. A subinterface bound to a physical interface is said to be the *child* of the physical interface, its *parent*. Accordingly, a

tunnel interface bound to a subinterface is the child of that subinterface, the physical interface being its *grandparent*. Figure 65 on page 246 illustrates these dependencies.

Figure 65: Interface Hierarchy



When working with virtual interfaces, bear in mind the following rules of interface hierarchy:

- Guaranteed bandwidth allocated to subinterfaces cannot be greater than the carrying capacity of the physical interface they are bound to. In Figure 65 on page 246, for example, the combined **gbw** of ethernet0/1.1 and ethernet0/1.2 is 7000 Kbps, 3000 Kbps below the **mbw** of ethernet0/1. Note, however, that the combined maximum bandwidth of these two subinterfaces exceeds the carrying capacity of the physical interface they are bound to by 2000 Kbps. This is acceptable because the **mbw** keyword is used only to limit traffic to a maximum rate. If traffic falls below a maximum setting on a subinterface, that bandwidth is available to any other subinterface bound to the same physical interface.
- Guaranteed bandwidth allocated to tunnel interfaces cannot be greater than the guaranteed bandwidth of the subinterface they are bound to.
- If guaranteed bandwidth is not configured for the immediate parent, bandwidth is taken from the grandparent interface.
- Total guaranteed bandwidth of children cannot exceed parent guaranteed bandwidth.
- Child maximum bandwidth cannot exceed parent maximum bandwidth.

As already stated, you cannot configure guaranteed bandwidth on physical interfaces because guaranteed bandwidth is the same as maximum bandwidth, which is the link speed of the interface. On virtual interfaces, however, you can configure egress **gbw** and **mbw**. You can also configure ingress **mbw**, which is ingress policing at the interface level. The following command guarantees a minimum out-going bit rate of 1000 Kbps on ethernet0/4.1, and a maximum rate, both incoming and outgoing, of 2000 Kbps:

```
set interface ethernet0/4.1 bandwidth egress gbw 1000 mbw 2000 ingress mbw
2000
```

You set bandwidth in the WebUI on the Network > Interfaces > Edit page.

After setting bandwidth, you use the **get traffic-shaping interface** command to see the actual bandwidth flowing through the security device. For example, you might have traffic entering on ethernet0/1 and exiting on ethernet0/3. If you set ingress bandwidth on ethernet0/1, the command **get traffic-shaping interface ethernet0/3** will show actual throughput on the device.

If you set traffic shaping at the interface, you must also set traffic-shaping mode to on (**set traffic-shaping mode on**).

Policy-Level Traffic Shaping

You shape traffic at the policy level to allocate bandwidth for particular types of traffic. The following command guarantees a minimum 1Mbps bandwidth to FTP traffic, and drops any traffic exceeding 2 Mbps:

```
set policy from trust to untrust any any ftp permit traffic gbw 1000 pbw 2000
```

Note that this command uses the policing bandwidth (**pbw**) keyword. You can use **pbw** or **mbw** in a policy, but not both. The advantage to using **pbw** is that traffic is dropped at the ingress side of the security device, reducing throughput processing and conserving system resources. (See "Ingress Policing" on page 244.)

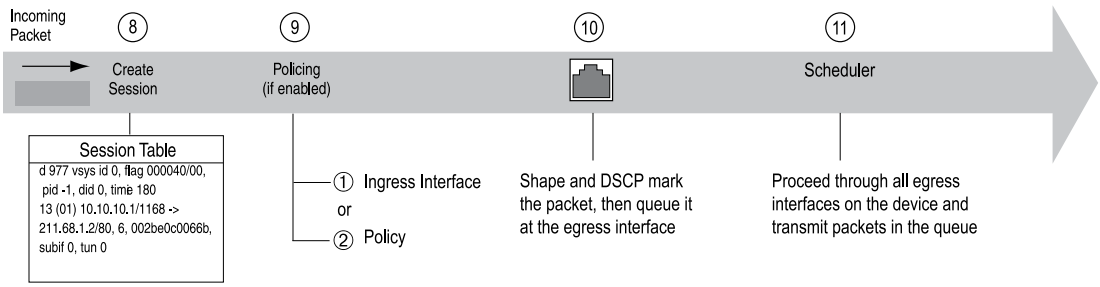
In the WebUI, after creating a policy, click the Advanced button to configure traffic-shaping parameters.

Although you must set traffic-shaping mode to **on** to shape traffic on interfaces, it is not necessary to turn on traffic shaping when shaping traffic in policies. This is because traffic-shaping mode is set to **auto** by default. When a session becomes active and policy lookup discovers traffic shaping, ScreenOS turns on traffic shaping for that session.

Packet Flow

Figure 66 on page 247 illustrates the part of the packet flow through the security device that is affected by traffic shaping and policing. (See "Packet-Flow Sequence" on page 27 for a complete picture of packet flow.) Packets exceeding **pbw** (or **mbw** configured at the interface) are dropped at step 9; shaping and DSCP marking occur at step 10, and packets exceeding **mbw** (configured in a policy) are dropped at step 11.

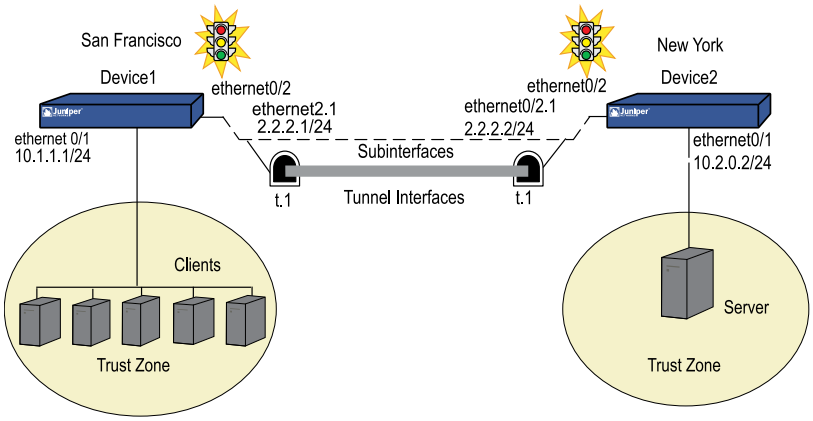
Figure 66: Traffic-Shaping Packet Flow



Example: Route-Based VPN with Ingress Policing

This example illustrates how to enforce ingress policing at the interface level for encrypted traffic. Ingress policing is configured on both the subinterface (ethernet0/2.1, maximum bandwidth:1200 Kbps) and the tunnel interface (tunnel.1, maximum bandwidth:1000 Kbps). You set the policing rate on the subinterface higher than on the tunnel interface bound to it to allow for the overhead of encryption (assuming, in this example, that all traffic received on the subinterface is meant for the tunnel interface). Policing on the subinterface is applied to the encrypted packets, while policing on the tunnel interface is applied to the decrypted inner packets. All encrypted traffic over 1200 Kbps on ethernet0/2.1 is dropped. And all decrypted (clear text) traffic over 1000 Kbps. on the tunnel.1 interface is dropped.

Figure 67: Route-Based VPN



WebUI (Configuration for Device1)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

   IP Address/Netmask: 10.1.1.1/24
   Zone: Trust

   Network > Interfaces > Sub-IF > New: Enter the following, then click **Apply**:

   Interface Name: (Select) ethernet0/2 and enter: 1
   Zone: Untrust
   IP Address/Netmask: 2.2.2.1/24
   VLAN Tag: 128

   Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

   Tunnel Interface Name: 1
   Zone: Untrust
   Unnumbered (select) ethernet0/2.1

Interface: ethernet0/2.1

2. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.0.0/24
Interface (select): Tunnel.1

3. **IKE**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: device1_ike
Security Level: Standard
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 2.2.2.2

Preshared Key

Preshared Key: secret
Outgoing Interface: ethernet0/2.1

VPNs > AutoKey IKE New: Enter the following, then click **OK**:

VPN Name: device1_vpn
Gateway Name: device1_ike

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: (Select) Tunnel Interface, (Select) tunnel.1

CLI (Configuration for the Device1)

1. **Interfaces**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 10.1.1.1/24
set interface ethernet0/2.1 tag 128 zone untrust
set interface tunnel.1 zone trust
set interface ethernet0/2.1 ip 2.2.2.1/24
set interface tunnel.1 ip unnumbered interface ethernet0/2.1
set route 10.2.0.0/24 int tunnel.1
```

2. **IKE**

```
set ike gateway device1_ike address 2.2.2.2 outgoing-interface ethernet0/2.1
 preshare sec-level standard
set vpn device1_vpn gateway 208a_ike sec-level standard
set vpn device1_vpn bind interface tunnel.1
save
```

WebUI (Configuration for Device2)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

   IP Address/Netmask: 10.2.0.2/24
   Zone: Trust

   Network > Interfaces > Sub-IF > New: Enter the following, then click **Apply**:

   Interface Name: (Select) ethernet0/2 and enter: 1
   Zone: Untrust
   IP Address/Netmask: 2.2.2.2/24
   VLAN Tag: 128

   Network > Interfaces > Tunnel IF > New: Enter the following, then click **Apply**:

   Tunnel Interface Name: 1
   Unnumbered: (select) ethernet0/2.1
   Interface: ethernet0/2.1

2. **Bandwidth on Interfaces**

   Network > Interfaces > Edit (for ethernet0/2.1): Enter the following, then click **OK**:

   Traffic Bandwidth, Ingress: 1200

   Network > Interfaces > Edit (for tunnel.1): Enter the following, then click **OK**:

   Traffic Bandwidth, Ingress: 1000

3. **Route**

   Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 10.1.1.0/24
   Interface (select): Tunnel.1

4. **IKE**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: device2_ike
   Security Level: Standard
   Remote Gateway Type:
   Static IP Address: (select), IP Address/Hostname: 2.2.2.2

   Preshared Key

   Preshared Key: secret

Outgoing Interface: ethernet0/2.1

VPNs > AutoKey IKE New: Enter the following, then click **OK**:

VPN Name: device2_vpn
Gateway Name: device2_ike

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: (Select) Tunnel Interface, (Select) tunnel.1

5. **Policies**

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Service: Any
Action: Permit

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Service: Any
Action: Permit

CLI (Configuration for the Device2)

1. **Interfaces**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 10.2.0.2/24
set interface ethernet0/2.1 tag 128 zone untrust
set interface ethernet0/2.1 ip 2.2.2.2/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet0/2.1
set route 10.1.1.0/24
```

2. **Bandwidth on interfaces**

```
set interface ethernet0/2.1 bandwidth ingress mbw 1200
set interface tunnel.1 bandwidth ingress mbw 1000
```

3. **IKE**

```
set ike gateway device2_ike address 2.2.2.1 preshare secret sec-level standard
set vpn device2_vpn gateway 208b_ike sec-level standard
set vpn device2_vpn bind interface tunnel.1
```

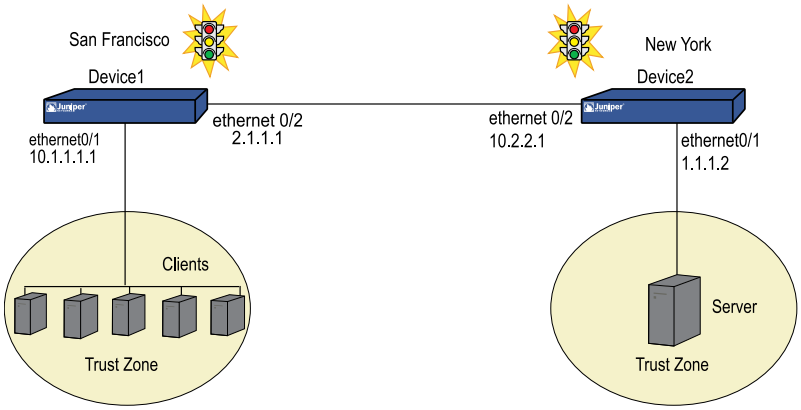
4. **Policy**

```
set policy from trust to untrust any any any permit
set policy from untrust to trust any any any permit
save
```

Example: Policy-Based VPN with Ingress Policing

This example illustrates how to enforce ingress policing at both the interface level and in policies. On the ethernet0/1 interface on *Device1*, you set the ingress maximum bandwidth at 20000 Kbps. With this setting, all traffic over 20000 Kbps from clients connected to *Device1* on the ethernet0/1 interface, is dropped. Ingress policing at the interface applies to all the traffic that arrives on that interface. For finer granularity, you can apply ingress policing at the policy level. In this example, you create policies to restrict all ingress FTP protocol traffic on *Device1* by creating policies between the trust and untrust zones, and set the policing bandwidth to 5000 Kbps. All FTP traffic over 5000 Kbps from the trust zone to the untrust zone is dropped.

Figure 68: Policy-Based VPN



WebUI (Configuration for Device1)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

   IP Address/Netmask: 10.1.1.1/24
   Zone: Trust
   Interface mode: (select) NAT

   Network > Interfaces > Edit (for ethernet0/2): Enter the following, then click **OK**:

   IP Address/Netmask: 2.1.1.1/24
   Zone: Untrust
   Interface mode: (select) Route

2. **IKE VPN**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: device2_ike

Security Level: Standard
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 2.2.2.2

Preshared Key

Preshared Key: secret
Outgoing Interface: ethernet0/2

> Advanced: Enter the following advanced settings, then click **OK** to return to basic Gateway configuration page:

Phase 1 Proposal: pre-g2-3des-sha

VPNs > AutoKey IKE New: Enter the following, then click **OK**:

VPN Name: device2_vpn
Gateway Name: device2_ike

3. **Interface-Based Policing**

Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

Traffic Bandwidth, Ingress: 20000

4. **Routing**

Network > Routing > Destination > New: Enter the following, then click **OK**:

Network IP Address/Netmask: 10.2.1.0/24
Interface: (select), ethernet0/2
Gateway IP Address: 2.2.2.2

5. **Policies**

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: 1
Service: FTP
Action: Tunnel
Tunnel VPN: (select), device2_vpn
Modify matching bidirectional VPN policy: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policies configuration page:

Traffic Shaping (select) Policing Bandwidth: 5000

CLI (Configuration for Device1)

1. **Interfaces**

```
set interface ethernet0/1 zone trust
```

```
set interface ethernet0/2 zone untrust
set interface ethernet0/1 ip 10.1.1.1/24
set interface ethernet0/1 nat
set interface ethernet0/2 ip 2.1.1.1/24
set interface ethernet0/1 route
```

2. **IKE VPN**

```
set ike gateway device2_ike address 2.2.2.2 main outgoing interface ethernet0/2
 preshare secret proposal pre-g2-3des-sha
set vpn device2_vpn gateway device2_ike no-replay tunnel idletime 0 sec-level
standard
```

3. **Routing**

```
set route 10.2.1.0/24 interface ethernet0/2 gateway 2.2.2.2
```

4. **Policies**

```
set policy from trust to untrust any any ftp tunnel vpn device2_vpn pair-policy 2
 traffic pbw 5000
set policy from untrust to trust any any ftp tunnel vpn netscreeen2_vpn pair-policy
 1 traffic pbw 5000
```

5. **Interface-Based Policing**

```
set interface ethernet0/1 bandwidth ingress mbw 20000
save
```

WebUI (Configuration for Device2)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

IP Address/Netmask: 1.1.1.1/24
Zone: Trust
Interface mode: (select) Route

Network > Interfaces > Edit (for ethernet0/2): Enter the following, then click **OK**:

IP Address/Netmask: 10.2.2.1/24
Zone: Untrust
Interface mode: (select) NAT

2. **IKE VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: device1_ike
Security Level: Standard
Remote Gateway Type:

Static IP Address: (select), IP Address/Hostname: 2.1.1.1

3. **Preshared Key**

   Preshared Key: secret
   Outgoing Interface: ethernet0/2

   > Advanced: Enter the following advanced settings, then click **OK** to return to basic Gateway configuration page:

   Phase 1 Proposal: pre-g2-3des-sha

   VPNs > AutoKey IKE New: Enter the following, then click **OK**:

   VPN Name: device1_vpn
   Gateway Name: device1_ike

4. **Routing**

   Network > Routing > Destination > New: Enter the following, then click **OK**:

   Network IP Address/Netmask: 10.1.1.0/24
   Interface: (select), ethernet0/2
   Gateway IP Address: 1.1.1.1

5. **Policies**

   Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Name: 1
   Service: FTP
   Action: Tunnel
   Tunnel VPN: (select), device1_vpn
   Modify matching bidirectional VPN policy: (select)

CLI (Configuration for Device2)

1. **Interfaces**

   ```
   set interface ethernet0/1 1.1.1.2/24
   set interface ethernet0/1 route
   set interface ethernet0/2 ip 10.2.2.1/24
   set interface ethernet0/2 nat
   ```

2. **IKE VPN**

   ```
   set ike gateway device1_ike address 2.1.1.1 main outgoing interface ethernet0/2
    preshare secret proposal pre-g2-3des-sha
   set vpn device1_vpn gateway device1_ike no-replay tunnel idletime 0 sec-level
   standard
   ```

3. **Routing**

   ```
   set route 10.1.1.0/24 interface ethernet0/1 gateway 1.1.1.1
   ```

4. **Policies**

```
set policy id 1 from trust to untrust any any ftp tunnel vpn device1_vpn pair-policy
 2
set policy id 2 from untrust to trust any any ftp tunnel vpn device1_vpn pair-policy
 1
save
```

Traffic Shaping Using a Loopback Interface

Traffic shaping is not supported on loopback interfaces, because no traffic is actually transmitted on a loopback interface. However, a loopback interface is often used as an anchor point (for example in the case of a VPN, to derive the source IP address), while the data is transmitted on an actual egress interface. When using a loopback interface in a VPN, therefore, you configure traffic shaping on the outgoing interface. ScreenOS then associates the session with the real outgoing interface, which it deduces from the routing table, dynamically updating the association as the routing table changes.

DSCP Marking and Shaping

As stated earlier in this chapter, Differentiated Services (DS) is a system for tagging (or "marking") traffic at a position within a hierarchy of priority. Differentiated Services codepoint (DSCP) marking maps the ScreenOS priority level of the policy to the first three bits of codepoint in the DS field in the IP packet header. (See "Setting Service Priorities" on page 238 for more information).

You can shape traffic in a policy that uses DSCP marking, or you can use DSCP marking independent of traffic shaping. Traffic shaping governs how traffic is processed on the security device and can be configured at the interface level or in policies. DSCP marking, which you set at the policy level, governs how traffic is processed by downstream routers.

NOTE: Some devices require that you explicitly enable DSCP marking by setting a system-wide environmental variable. Refer to the installation and configuration guide for your device to find out if it requires that you explicitly enable DSCP marking before using it in policies. If your device requires it, use the following command to enable DSCP marking system wide: **set envar ipsec-dscp-mark = yes**. This variable cannot be set using the WebUI. Use the **unset envar ipsec-dscp-mark** to disable DSCP marking system wide.

The DSCP marking feature is disabled in IDP-capable security devices. For information about IDP-capable security devices, see "Intrusion Detection and Prevention" on page 615.

If you specify DSCP marking in a policy but do not set a value, ScreenOS maps the policy priority to an equivalent IP precedence priority in the DSCP system. It does this by overwriting the first 3 bits in the ToS byte with the IP precedence priority. For example, if you create a policy that gives all traffic a priority of, for example, 2

(0 is the highest priority), and you enable DSCP marking, ScreenOS queues traffic for that policy with level 2 priority at the egress interface and marks it with an equivalent IP precedence priority. The following command creates a policy that gives priority 2 to all traffic, and enables DSCP marking:

```
set policy from trust to untrust any any any permit traffic priority 2 dscp enable
```

But if you give DSCP a *dscp-byte* **value** of, for example, 46 (the highest priority), the security device still queues traffic at the egress interface at priority 2 but overwrites the first 6 bits of the ToS byte with the DSCP value.

```
set policy from trust to untrust any any any permit traffic priority 2 dscp enable value
 46
```

DSCP marking is supported on all platforms and can be configured with traffic shaping or independently.

Enabling Differentiated Services Code Point

You can use the WebUI or CLI to enable Differentiated Services Code Point (DSCP) marking and specify the DSCP value for every route-based VPN. By default DSCP marking is disabled.

WebUI

VPNs > AutoKey IKE > Edit: Select the following options, then click **Apply**.

```
DSCP Mark: Enable(Select)
Dscp-value: Enter the DSCP value
```

You can disable DSCP marking by selecting the Disable option in the DSCP Mark field.

CLI

The following command enables DSCP-marking and sets the 6-bit DSCP in the IPSEC header to 52.

```
set vpn vpn1 dscp-marking 52
```

To disable DSCP marking, use the following command:

```
unset vpn vpn1 dscp-marking
```

Table 32 on page 258 shows how DSCP marking works for clear packets in policies. Table 33 on page 258 shows how DSCP marking works for clear packets in policy-based VPNs. Table 34 on page 258 shows how DSCP marking works for clear packets in route-based VPNs.

Table 32: DSCP Marking for Clear-Text Traffic

Description	Action
Clear packet with no marking on the policy.	No marking.
Clear packet with marking on the policy.	The packet is marked based on the policy.
Premarked packet with no marking on the policy.	Retain marking in the packet.
Premarked packet with marking on the policy.	Overwrite marking in the packet based on the policy.

Table 33: DSCP Marking for Policy-Based VPNs

Description	Action
Clear packet into policy-based VPN with no marking on the policy.	No marking.
Clear packet into policy-based VPN with marking on the policy.	The inner packet and IP header of the ESP are both marked, based on the policy.
Premarked packet into policy-based VPN with no marking on the policy.	Copy the inner packet marking to the IP header of the ESP, retain marking in the inner packet.
Premarked packet into policy-based VPN with marking on the policy.	Overwrite the marking in the inner packet based on the policy, and copy the inner packet marking to the IP header of the ESP.

Table 34: DSCP Marking for Route-Based VPNs

Description	Action
Clear packet into route-based VPN with no marking on the policy.	No marking.
Clear packet into route-based VPN with marking on the policy.	The inner packet and IP header of the ESP are both marked, based on the policy.
Premarked packet into route-based VPN with no marking on the policy.	Copy the inner packet marking to the IP header of the ESP, and retain marking in the inner packet.
Premarked packet into route-based VPN with marking on the policy.	Overwrite the marking in the inner packet based on the policy, and copy the inner packet marking to the IP header of the ESP.

Quality of Service Classification Based on Incoming Markings

In ScreenOS 6.3.0, traffic-shaping policies are enhanced to support quality of service (QoS) classification based on the IP precedence and Differentiated Services code point (DSCP) marking of incoming packets.

NOTE: The QoS classification feature on incoming traffic works only if traffic shaping mode is set to Auto or On.

There are three modes for QoS classification: Classic, IntServ, and DiffServ. In IntServ mode, IP precedence is used for QoS classification. In DiffServ mode, DSCP is used for QoS classification.

QoS profiles map the IP precedence and DSCP values of incoming packets with QoS parameters.

You can create two types of QoS profiles:

- A Precedence Profile (PP) contains entries for mapping IP precedence with QoS parameters.
- A DSCP Profile (DP) contains entries for mapping DSCP with QoS parameters.

Each QoS profile can have several entries, each representing a single mapping between one DSCP or IP precedence value and its preferred QoS parameters. IP precedence has 8 possible values, so a PP can have a maximum of 8 entries. DSCP has 64 possible values, so a DP can have a maximum of 64 entries. For a QoS profile, an existing entry can be overwritten with the same DSCP/IP precedence value. This will change the QoS parameters of that entry without interrupting the traffic.

In a PP, the three most significant bits in the type of service (TOS) field of each packet are mapped to the profile entries. In a DP, the six most significant bits in the TOS field of each IP packet are mapped to profile entries.

A single QoS profile is attached to a specified policy. However, the type of profile for a matched policy determines whether the matching packets operate under IntServ or DiffServ mode. After a policy is bound to a profile, the previous QoS parameters are overwritten.

For a flow in one direction only (incoming or outgoing), QoS parameters are determined by the IP precedence or DSCP value of the first packet that passes through the device. The QoS parameters for that flow do not change even if the subsequent packets carry other IP precedence or DSCP values.

For a flow with both incoming and outgoing packets that have different IP precedence or DSCP values, the packets are assigned different QoS parameters.

NOTE: The already existing QoS parameters are fully supported by the QoS classification. QoS classification is based on the original precedence and DSCP values of the incoming packets.

The following commands provide an example of a basic QoS profile configuration, where you are setting up a QoS profile of type DSCP.

WebUI

Policy > Policy Elements > Traffic Shaping: Enter the following, then click New Profile:

Traffic Shaping

Mode: (Select)

QoS Profile

Name: dscp_test
Type: (select) DSCP

Policy > Policy Elements > Traffic Shaping > Profile Edit: Enter the following, then click New:

TOS: 0
Priority: 3
PBW: 100
GBW: 100
MBW: 100

Policy > Policies > (From: UnTrust, To: Trust) New: Enter the following, then click OK:

Name: dscp_test
Source Address:
Address Book Entry: (select)
Destination Address:
Book Entry: (select) Any
Service: Any
Action: Permit

> Advanced: Enter the following, then click Return to set the advanced options and return to the basic configuration page:

Incoming Mark Based Configuration: (select)
Qos Profile: (select) DSCP

CLI

QoS Profile Settings

```
set qos-profile dscp_test dscp
```

```
set qos-profile dscp_test tos 0 priority 3 gbw 100 mbw 1000
```

Policy Settings

```
set policy id 1 from untrust to trust any any any permit qos-profile dscp_test
```

DSCP Marking for Self-initiated Traffic

The administrator can configure the DSCP value for the traffic initiated by the security device. Altogether, DSCP value can be configured for 11 services: BGP, OSPF, RIP, RIPNG, TELNET, SSH, WEB, TFTP, SNMP, SYSLOG and WEBTRENDS. The Web service contains the HTTP and HTTPS services.

The DSCP marking for self-initiated traffic is required. These self-initiated packets might be dropped by an intermediate device because of lower priority.

By default, this feature is disabled. The DSCP value of the BGP and the OSPF packet is set to 48; for all other services, the default value is 0. The value must be in the range of 0 to 63. The priority is lowest when the DSCP value is set to 0.

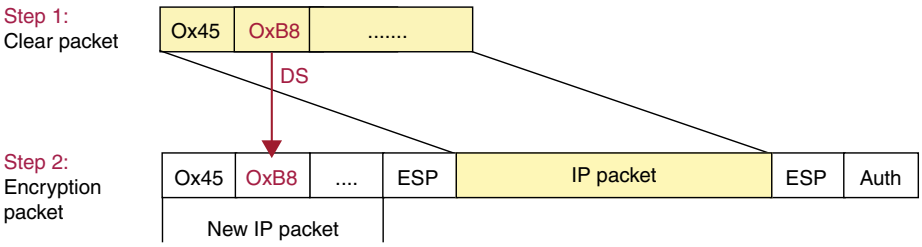
When the administrator sets the DSCP value for a specific service, the DSCP field of all the self-initiated packets which belong to that service are set to the specified value. For the self-initiated packets that go through the VPN tunnel, both the inner packet and ESP header are marked.

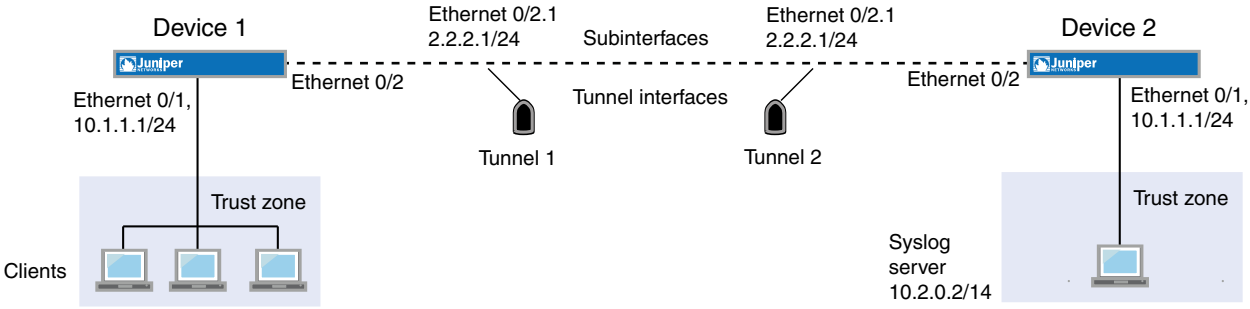
The following example illustrates how the DSCP marking works for the VPN traffic.

In the example, Device1 enables the syslog function and then admin executes the `set ip service syslog dscp 46`. Two steps are required for DSCP marking in this case.

1. The clear-text packet is marked based on the specified value 46. DSCP is the left-most 6 bit of DS. Hence, the DS field value is set to 0xB8 (46 < < 2).
2. ScreenOS copies the DS field of inner packet to ESP header.

Figure 69: DSCP Marking for VPN Traffic





In the following example, you set the DSCP value for the telnet service to 20.

CLI

```
set ip service telnet dscp 20
```

WebUI

Network > DSCP: In the Telnet text—box, enter 20, then click Apply.

Chapter 9
System Parameters

This chapter focuses on the concepts involved in configuring system parameters that control the following areas of a security device. It contains the following sections:

Domain Name System Support

The Juniper Networks security device incorporates Domain Name System (DNS) support, which allows you to use domain names as well as IP addresses for identifying locations. A DNS server keeps a table of the IP addresses associated with domain names. Using DNS makes it possible to reference locations by domain name (such as www.juniper.net) in addition to using the routable IP address, which for www.juniper.net is 207.17.137.68. DNS for IPv6 addresses also supports Netscreen Redundancy Protocol (NSRP). DNS translation is supported in all the following programs:

- Address Book
- Syslog
- Email
- WebTrends
- Websense
- LDAP
- SecurID
- RADIUS
- Network and Security Manager

Before you can use DNS for domain name/address resolution, you must enter the addresses for DNS servers in the security device.

NOTE: When enabling the security device as a Dynamic Host Configuration Protocol (DHCP) server (see "Dynamic Host Configuration Protocol" on page 271), you must also enter the IP addresses for DNS servers in the DHCP page on the WebUI or through the **set interface** *interface* **dhcp** command in the CLI.

DNS Lookup

The security device refreshes all the entries in its DNS table by checking them with a specified DNS server at the following times:

- After an HA failover occurs
- At a regularly scheduled time of day and at regularly scheduled intervals throughout the day
- When you manually command the device to perform a DNS lookup
  - WebUI: Network > DNS > Host: Click **Refresh**.
  - CLI: **exec dns refresh**

In addition to the existing method of setting a time for a daily automatic refresh of the DNS table, you can also define an interval of time from 4 to 24 hours.

NOTE: When you add a fully qualified domain name (FQDN) such as an address or IKE gateway through the WebUI, the security device resolves it when you click **Apply** or **OK**. When you type a CLI command that references an FQDN, the security device attempts to resolve it when you enter it.

When the security device connects to the DNS server to resolve a domain name/IP address mapping, it stores that entry in its DNS status table. The following list contains some of the details involved in a DNS lookup:

- When a DNS lookup returns multiple entries, the address book accepts all entries. The other programs listed on "Domain Name System Support" on page 263 accept only the first one.
- The security device reinstalls all policies if it finds that anything in the domain name table has changed when you refresh a lookup using the **Refresh** button in the WebUI or enter the **exec dns refresh** CLI command.
- If a DNS server fails, the security device looks up everything again.
- If a lookup fails, the security device removes it from the cache table.
- If the domain name lookup fails when adding addresses to the address book, the security device displays an error message stating that you have successfully added the address but the DNS name lookup failed.

The security device must do a new lookup once a day, which you can schedule it to do at a specified time.

WebUI

Network > DNS > Host: Enter the following, then click **Apply**:

DNS refresh every day at: Select check box and enter time <hh:mm>

CLI

```
set dns host schedule time_str
save
```

DNS Status Table

The DNS status table reports all the domain names looked up, their corresponding IP addresses, whether the lookup was successful, and when each domain name/IP address was last resolved.

Table 35: DNS Status Table

Name	IP Address	Status	Last Lookup
www.yahoo.com	204.71.200.74	Success	8/13/2000 16:45:33
www.hotbot.com	204.71.200.75	Success	8/13/2000 16:45:38
	204.71.200.67		
	204.71.200.68		
	209.185.151.28		
	209.185.151.210		
	216.32.228.18		

To view the DNS status table, do either of the following:

WebUI

Network > DNS > Host > Show DNS Lookup Table

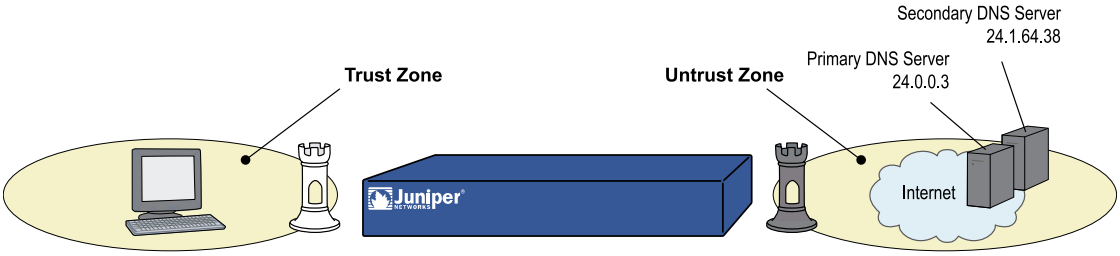
CLI

```
get dns host report
```

Setting the DNS Server and Refresh Schedule

To implement DNS functionality, the IP addresses for the DNS servers at 24.1.64.38 and 24.0.0.3 are entered in the security device, protecting a single host in a home office. The security device is scheduled to refresh the DNS settings stored in its DNS status table every day at 11:00 P.M.

Figure 70: DNS Refresh



WebUI

Network > DNS > Host: Enter the following, then click **Apply**:

Primary DNS Server: 24.0.0.3
Secondary DNS Server: 24.1.64.38
DNS Refresh: (select)
Every Day at: 23:00

CLI

```
set dns host dns1 24.0.0.3
set dns host dns2 24.1.64.38
set dns host schedule 23:00
save
```

Setting a DNS Refresh Interval

In this example, you configure the security device to refresh its DNS table every four hours beginning at 12:01 AM every day.

WebUI

Network > DNS > Host: Enter the following, then click **Apply**:

DNS Refresh: (select)
Every Day at: 00:01
Interval: 4

CLI

```
set dns host schedule 00:01 interval 4
save
```

Dynamic Domain Name System

Dynamic DNS (DDNS) is a mechanism that allows clients to dynamically update IP addresses for registered domain names. This is useful when an ISP uses PPP, DHCP, or XAuth to dynamically change the IP address for a CPE router (such as a security device) that protects a Web server. Internet clients can reach the Web server by using a domain name even if the IP address of the security device has previously changed

dynamically. This is made possible by a DDNS server (such as dyndns.org or ddo.jp), which maintains a list of the dynamically changed addresses and their associated domain names. The device updates these DDNS servers with this information periodically or in response to IP address changes.

To use DDNS, create an account (username and password) on the DDNS server. The server uses this account information to configure the client device.

Figure 71: Dynamic DNS

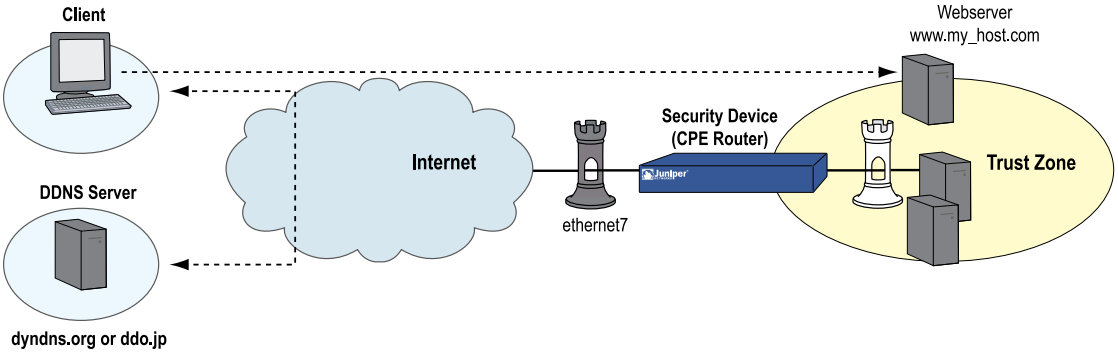


Figure 71 on page 267 shows how the DDNS concept works. When an IP address changes, the client can use the hostname www.my_host.com to reach the protected Web server, through either the dyndns.org server or the ddo.jp server. However, each of these servers requires a different configuration on the security device interface.

If you have configured the DDNS server type as dyndns.org in your security device, you can choose the following service options:

- dyndns—You can assign a dynamic IP (DIP) address to a static hostname in a fixed domain that dyndns.org provides. You can configure up to 5 hostnames. A dynamic DNS entry expires if it is not updated for 35 days.
- statdns—You can configure a hostname, such as yourname.dyndns.com, to point to your IP address.

  Unlike entries from a DDNS host, static DNS entries do not expire after 35 days without an update. However, static updates take longer to propagate through the DNS system and support a maximum of 5 hostnames.
- custom—You can control the domain names and dynamic IP addresses over the entire zone at the domain level, rather than at the domain-name level.

  Using a custom DNS service, you can configure unlimited hostnames and support for any domain purchased from dyndns.org. Changes you make to the DNS are propagated instantly across the DNS network. Dynamic DNS entries with the custom service never expire.

NOTE: Service options are available only with the dyndns.org service. For static and custom DNS services, you can configure a higher value for the minimum update interval than for dynamic DNS service, because the IP addresses change infrequently.

Setting Up DDNS for a Dynamic DNS Server

In the following example, you configure a security device for DDNS operation. The device uses the dyndns.org server to resolve changed addresses. For this server, you specify the protected host using the Host Name setting, which explicitly binds to the DNS interface (ethernet7).

WebUI

Network > DNS > DDNS > New: Enter the following, then click **OK**:

ID: 12
Server Settings
    Server Type: dyndns
    Server Name: members.dyndns.org
    Refresh Interval: 24
    Minimum Update Interval: 15
Account Settings
    Username: swordfish
    Password: ad93lvb
    Agent: Netscreen-6.0.0z-015608200600056
Bind to Interface: ethernet7
Host Name: www.my_host.com
Service: dyndns

NOTE: Minimum Update Interval specifies the minimum time interval (expressed in minutes) between DDNS updates. The default is 10 minutes, and the allowable range is 1-1440. In some cases, the device might not update the interval because the DNS server first needs to time out the DDNS entry from its cache. In addition, if you set the Minimum Update Interval to a low value, the security device might lock you out. The recommended minimum value is 10 minutes.

CLI

```
set dns ddns
set dns ddns enable
set dns ddns id 12 server members.dyndns.org server-type dyndns refresh-interval
24 minimum-update-interval 15
set dns ddns id 12 src-interface ethernet7 host-name myhost_dynamic.dyndns.org
service dyndns
set dns ddns id 12 username swordfish password ad93lvb
save
```

Setting Up DDNS for a DDO Server

In the following example, you configure a security device for DDNS. The device uses the ddo.jp server to resolve addresses. For the ddo.jp server, you specify the protected host FQDN as the Username setting for the DDNS entry instead of specifying the protected host using the Host Name setting. The service automatically derives the hostname from the Username value. For example, the ddo.jp server translates a username of my_host to my_host.ddo.jp. You need to make sure that the registered domain name on ddo.jp matches the derived DNS.

WebUI

Network > DNS > DDNS > New: Enter the following, then click **OK**:

ID: 25
Server Settings
    Server Type: ddo
    Server Name: juniper.net
    Refresh Interval: 24
    Minimum Update Interval: 15
Account Settings
    Username: my_host
    Password: ad93lvb
    Agent: Netscreen-6.0.0z-015608200600056
Host Name: www.my_host.com
Bind to Interface: ethernet7

CLI

```
set dns ddns
set dns ddns enable
set dns ddns id 25 server ddo.jp server-type ddo refresh-interval 24
minimum-update-interval 15
set dns ddns id 25 src-interface ethernet7
set dns ddns id 25 username my_host password ad93lvb
save
```

Proxy DNS Address Splitting

The proxy DNS feature provides a transparent mechanism that allows clients to make split DNS queries. Using this technique, the proxy selectively redirects the DNS queries to specific DNS servers, according to partial or complete domain names. This is useful when multiple VPN tunnels or PPPoE virtual links provide network connectivity and it is necessary to direct some DNS queries to one network and other queries to another network.

A DNS proxy has the following advantages:

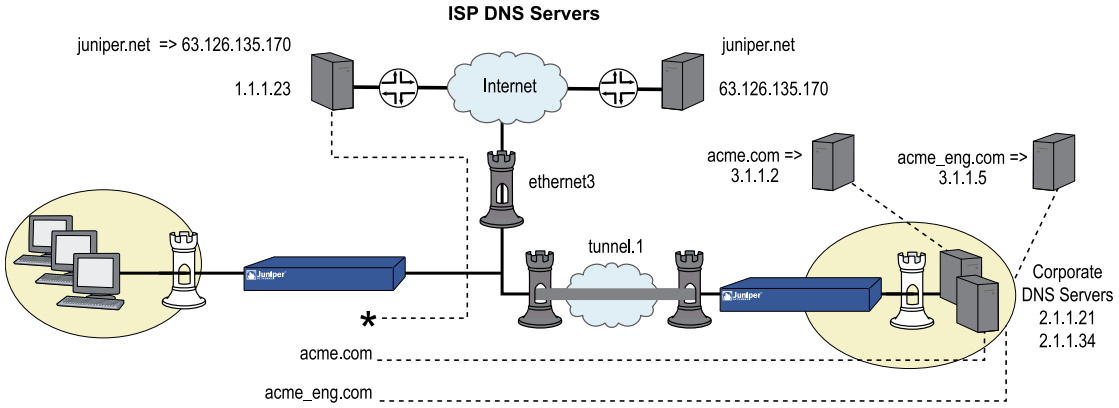
- Domain lookups are usually more efficient. For example, DNS queries meant for the corporate domain (such as acme.com) could go to the corporate DNS server exclusively, while all others go to the ISP DNS server, which reduces the

load on the corporate server. This can also prevent corporate domain information from leaking into the Internet.

- DNS proxy allows you to transmit selected DNS queries through a tunnel interface, which prevents malicious users from learning about the internal configuration of a network. For example, DNS queries bound for the corporate server can pass through a tunnel interface to use security features such as authentication, encryption, and anti-replay.

The following commands create two proxy-DNS entries that selectively forward DNS queries to different servers.

Figure 72: Splitting DNS Requests



- Any DNS query with a FQDN containing the domain name acme.com goes out through tunnel interface tunnel.1, to the corporate DNS server at IP address 2.1.1.21.

  For example, if a host sends a DNS query for www.acme.com, the device automatically directs the query to this server. (Let's assume that the server resolves the query to IP address 3.1.1.2.)

- Any DNS query with a FQDN containing the domain name acme_eng.com goes out through tunnel interface tunnel.1 to the DNS server at IP address 2.1.1.34.

  For example, if a host sends a DNS query for intranet.acme_eng.com, the device directs the query to this server. (Let's assume that the server resolves the query to IP address 3.1.1.5.)

- All other DNS queries (denoted by an asterisk) bypass the corporate servers and go out through interface ethernet0/3 to the DNS server at IP address 1.1.1.23.

  For example, if the host and domain name is www.juniper.net, the device automatically bypasses the corporate servers and directs the query to this server, which resolves the query to IP address 207.17.137.68.

WebUI

Network > DNS > Proxy: Enter the following, then click **Apply**:

Initialize DNS Proxy: Enable
Enable DNS Proxy: Enable

Network > DNS > Proxy > New: Enter the following, then click **OK**:

Domain Name: acme.com
Outgoing Interface: tunnel.1
Primary DNS Server: 2.1.1.21

Network > DNS > Proxy > New: Enter the following, then click **OK**:

Domain Name: acme_eng.com
Outgoing Interface: tunnel.1
Primary DNS Server: 2.1.1.34

Network > DNS > Proxy > New: Enter the following, then click **OK**:

Domain Name: *
Outgoing Interface: ethernet0/3
Primary DNS Server: 1.1.1.23

CLI

```
set dns proxy
set dns proxy enable
set interface ethernet0/3 proxy dns
set dns server-select domain acme.com outgoing-interface tunnel.1 primary-server
2.1.1.21
set dns server-select domain acme_eng.com outgoing-interface tunnel.1 primary-server
 2.1.1.34
set dns server-select domain * outgoing-interface ethernet0/3 primary-server 1.1.1.23
save
```

Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol (DHCP) reduces the demands on network administrators by automatically assigning the TCP/IP settings for the hosts on a network. Instead of requiring administrators to assign, configure, track, and change (when necessary) all the TCP/IP settings for every machine on a network, DHCP does it all automatically. Furthermore, DHCP ensures that duplicate addresses are not used, reassigns unused addresses, and automatically assigns IP addresses appropriate for the subnet on which a host is connected.

Different security devices support the different DHCP roles described in Table 36 on page 271.

Table 36: DHCP Roles

Role	Description
DHCP Client	Some security devices can act as DHCP clients, receiving a dynamically assigned IP address for any physical interface in any zone.

Table 36: DHCP Roles *(continued)*

Role	Description
DHCP Server	Some security devices can also act as DHCP servers, allocating dynamic IP addresses to hosts (acting as DHCP clients) on any physical or VLAN interface in any zone. Note: While using the DHCP server module to assign addresses to hosts such as workstations in a zone, you can still use fixed IP addresses for other machines such as mail servers and WINS servers.
DHCP Relay Agent	Some security devices can also act as DHCP relay agents, receiving DHCP information from a DHCP server and relaying that information to hosts on any physical or VLAN interface in any zone.
DHCP Client/Server/Relay Agent	Some security devices can simultaneously act as a DHCP client, server, and relay agent. You can only configure one DHCP role on a single interface. For example, you cannot configure the DHCP client and server on the same interface. Optionally, you can configure the DHCP client module to forward TCP/IP settings that it receives to the DHCP server module, for use when providing TCP settings to hosts in the Trust zone acting as DHCP clients.

NOTE: ScreenOS now supports DHCP consistently on all platforms.

DHCP consists of two components: a protocol for delivering host-specific TCP/IP configuration settings and a mechanism for allocating IP addresses. When the security device acts as a DHCP server, it provides the following TCP/IP settings to each host when that host starts up:

- Default gateway IP address and netmask. If you leave these settings as 0.0.0.0/0, the DHCP server module automatically uses the IP address and netmask of the default Trust zone interface.

NOTE: On devices that can have multiple interfaces bound to the Trust zone, the default interface is the first interface bound to that zone and assigned an IP address.

- The IP addresses of the following servers:
  - WINS servers (2): A Windows Internet Naming Service (WINS) server maps a NetBIOS name used in a Windows NT network environment to an IP address used on an IP-based network. The number in parentheses indicates the number of servers supported.
  - NetInfo servers (2): NetInfo is an Apple network service used for the distribution of administrative data within a LAN.
  - NetInfo tag (1): The identifying tag used by the Apple NetInfo database.
  - DNS servers (3): A Domain Name System (DNS) server maps a uniform resource locator (URL) to an IP address.

- SMTP server (1): A Simple Mail Transfer Protocol (SMTP) server delivers SMTP messages to a mail server, such as a POP3 server, which stores the incoming mail.
- POP3 server (1): A Post Office Protocol version 3 (POP3) server stores incoming mail. A POP3 server must work conjointly with an SMTP server.
- News server (1): A news server receives and stores postings for news groups.

NOTE: If a DHCP client to which the security device is passing the above parameters has a specified IP address, that address overrides all the dynamic information received from the DHCP server.

Configuring a DHCP Server

A security device can support up to eight DHCP servers on any physical or VLAN interface in any zone. When acting as a DHCP server, a security device allocates IP addresses and subnet masks in two modes:

- In dynamic mode, the security device, acting as a DHCP server, assigns (or "leases") an IP address from an address pool to a host DHCP client. The IP address is leased for a determined period of time or until the client relinquishes the address. (To define an unlimited lease period, enter 0.)
- In reserved mode, the security device assigns a designated IP address from an address pool exclusively to a specific client every time that client goes online.

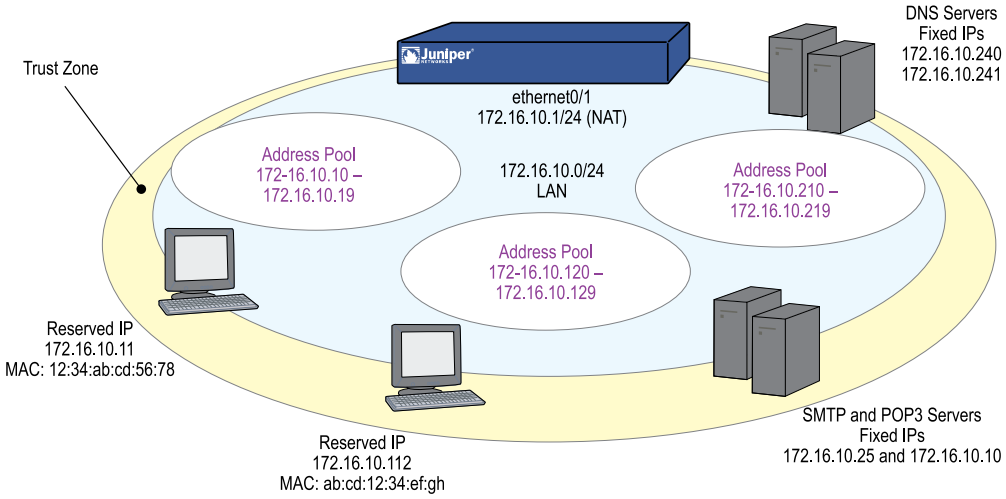
NOTE: An address pool is a defined range of IP addresses within the same subnet from which the security device can draw DHCP address assignments. You can group up to 255 IP addresses.
The DHCP server supports up to 64 entries, which can include both single IP addresses and IP address ranges, for dynamic and reserved IP addresses.
The security device saves every IP address assigned through DHCP in flash memory. Consequently, rebooting the security device does not affect address assignments.

In this example, using DHCP, the 172.16.10.0/24 network in the Trust zone is sectioned into three IP address pools.

- 172.16.10.10 through 172.16.10.19
- 172.16.10.120 through 172.16.10.129
- 172.16.10.210 through 172.16.10.219

The DHCP server assigns all IP addresses dynamically, except for two workstations with reserved IP addresses and four servers with static IP addresses. The interface ethernet0/1 is bound to the Trust zone, has IP address 172.16.10.1/24, and is in NAT mode. The domain name is dynamic.com.

Figure 73: Device as DHCP Server



WebUI

1. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: DNS#1
   Comment: Primary DNS Server
   IP Address/Domain Name:
   IP/Netmask: (select), 172.16.10.240/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: DNS#2
   Comment: Secondary DNS Server
   IP Address/Domain Name:
   IP/Netmask: (select), 172.16.10.241/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: SMTP
   Comment: SMTP Server
   IP Address/Domain Name:
   IP/Netmask: (select), 172.16.10.25/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: POP3

Comment: POP3 Server
IP Address/Domain Name:
IP/Netmask: (select), 172.16.10.110/32
Zone: Trust

2. **DHCP Server**

Network > DHCP > Edit (for ethernet0/1) > DHCP Server: Enter the following, then click **Apply**:

Lease: Unlimited (select)
WINS#1: 0.0.0.0
DNS#1: 172.16.10.240

NOTE: If you leave the Gateway and Netmask fields as **0.0.0.0**, the DHCP server module sends the IP address and netmask set for ethernet0/1 to its clients (172.16.10.1 and 255.255.255.0 in this example). However, if you enable the DHCP client module to forward TCP/IP settings to the DHCP server module (see "Propagating TCP/IP Settings" on page 286), then you must manually enter **172.16.10.1** and **255.255.255.0** in the Gateway and Netmask fields, respectively.

> Advanced Options: Enter the following, then click **OK** to set the advanced options and return to the basic configuration page:

WINS#2: 0.0.0.0
DNS#2: 172.16.10.241
DNS#3: 0.0.0.0
SMTP: 172.16.10.25
POP3: 172.16.10.110
NEWS: 0.0.0.0
NetInfo Server #1: 0.0.0.0
NetInfo Server #2: 0.0.0.0
NetInfo Tag: (leave field empty)
Domain Name: dynamic.com

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 172.16.10.10
IP Address End: 172.16.10.19

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 172.16.10.120
IP Address End: 172.16.10.129

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 172.16.10.210
IP Address End: 172.16.10.219

> Addresses > New: Enter the following, then click **OK**:

Reserved: (select)
IP Address: 172.16.10.11
Ethernet Address: 1234 abcd 5678

> Addresses > New: Enter the following, then click **OK**:

Reserved: (select)
IP Address: 172.16.10.112
Ethernet Address: abcd 1234 efgh

CLI

1. **Addresses**

```
set address trust dns1 172.16.10.240/32 “primary dns server”
set address trust dns2 172.16.10.241/32 “ secondary dns server”
set address trust snmp 172.16.10.25/32 “ snmp server”
set address trust pop3 172.16.10.110/32 “ pop3 server”
```

2. **DHCP Server**

```
set interface ethernet0/1 dhcp server option domainname dynamic.com
set interface ethernet0/1 dhcp server option lease 0
set interface ethernet0/1 dhcp server option dns1 172.16.10.240
set interface ethernet0/1 dhcp server option dns2 172.16.10.241
set interface ethernet0/1 dhcp server option smtp 172.16.10.25
set interface ethernet0/1 dhcp server option pop3 172.16.10.110
set interface ethernet0/1 dhcp server ip 172.16.10.10 to 172.16.10.19
set interface ethernet0/1 dhcp server ip 172.16.10.120 to 172.16.10.129
set interface ethernet0/1 dhcp server ip 172.16.10.210 to 172.16.10.219
set interface ethernet0/1 dhcp server ip 172.16.10.11 mac 1234abcd5678
set interface ethernet0/1 dhcp server ip 172.16.10.112 mac abcd1234efgh
set interface ethernet0/1 dhcp server service
save
```

NOTE: If you do not set an IP address for the gateway or a netmask, the DHCP server module sends its clients the IP address and netmask for ethernet0/1 (172.16.10.1 and 255.255.255.0 in this example). However, if you enable the DHCP client module to forward TCP/IP settings to the DHCP server module (see “Propagating TCP/IP Settings” on page 286), then you must manually set these options: **set interface ethernet0/1 dhcp server option gateway 172.16.10.1** and **set interface ethernet0/1 dhcp server option netmask 255.255.255.0**.

Customizing DHCP Server Options

When you specify DHCP servers for an interface, you might need to specify options that identify the servers or provide information used by the servers. For example, you can specify the IP address of the primary and secondary DNS servers, or set the IP address lease time.

The following are predefined DHCP services, as described in RFC 2132, *DHCP Options and BOOTP Vendor Extensions*.

Table 37: Predefined DHCP Services

Terminology	ScreenOS CLI Terminology	Option Code
Subnet Mask	netmask	1
Router Option	gateway	3
Domain Name System (DNS) server	dns1, dns2, dns3	6
Domain Name	domainname	15
NetBIOS over TCP/IP Name Server Option	wins1, wins2	44
IP Address Lease Time	lease	51
SMTP Server Option	smtp	69
POP3 Server Option	pop3	70
NNTP Server Option	news	71
(N/A)	nis1, nis2	112
(N/A)	nistag	113

In situations where the predefined server options are inadequate, you can define custom DHCP server options. For example, for certain Voice-over IP (VoIP) configurations, it is necessary send extra configuration information, which is not supported by predefined server options. In such cases, you must define suitable custom options.

In the following example, you create DHCP server definitions for IP phones which act as DHCP clients. The phones use the following custom options:

- Option code 444, containing string “Server 4”
- Option code 66, containing IP address 1.1.1.1
- Option code 160, containing integer 2004

CLI

1. **Addresses**

```
set address trust dns1 172.16.10.240/32 “primary dns server”
set address trust dns2 172.16.10.241/32 “ secondary dns server”
```

2. **DHCP Server**

```
set interface ethernet0/1 dhcp server option domainname dynamic.com
set interface ethernet0/1 dhcp server option lease 0
set interface ethernet0/1 dhcp server option dns1 172.16.10.240
set interface ethernet0/1 dhcp server option dns2 172.16.10.241
set interface ethernet0/1 dhcp server option custom 444 string “Server 4”
set interface ethernet0/1 dhcp server option custom 66 ip 1.1.1.1
set interface ethernet0/1 dhcp server option custom 160 integer 2004
set interface ethernet0/1 dhcp server ip 172.16.10.10 to 172.16.10.19
```

Placing the DHCP Server in an NSRP Cluster

When the primary unit in a redundant NSRP cluster functions as a DHCP server, all members in the cluster maintain all DHCP configurations and IP address assignments. In the event of a failover, the new primary unit maintains all the DHCP assignments. However, termination of HA communication disrupts synchronization of existing DHCP assignments among the cluster members. After restoring HA communication, you can resynchronize the DHCP assignments by using the following CLI command on both units in the cluster: **set nsrp rto-mirror sync**.

DHCP Server Detection

When a DHCP server on a security device starts up, the system can first check to see if there is already a DHCP server on the interface. ScreenOS automatically stops the local DHCP server process from starting if another DHCP server is detected on the network. To detect another DHCP server, the device sends out DHCP boot requests at two-second intervals. If the device does not receive any response to its boot requests, it then starts its local DHCP server process.

If the security device receives a response from another DHCP server, the system generates a message indicating that the DHCP service is enabled on the security device but not started because another DHCP server is present on the network. The log message includes the IP address of the existing DHCP server.

You can set one of three operational modes for DHCP server detection on an interface: auto, enable, or disable. Auto mode causes the security device to always check for an existing DHCP server at bootup. You can configure the device to not attempt to detect another DHCP server on an interface by setting the security DHCP server to enable or disable mode. In enable mode, the DHCP server is always on and the device does not check if there is an existing DHCP server on the network. In disable mode, the DHCP server is always off.

Enabling DHCP Server Detection

In this example, you set the DHCP server on the ethernet0/1 interface to check for an existing DHCP server on the interface first before starting up.

WebUI

Network > DHCP > Edit (for ethernet0/1) > DHCP Server: Enter the following, then click **OK**:

Server Mode: Auto (select)

CLI

```
set interface ethernet0/1 dhcp server auto
save
```

Disabling DHCP Server Detection

In this example, you set the DHCP server on the ethernet0/1 interface to start up without checking to see if there is an existing DHCP server on the network.

WebUI

Network > DHCP > Edit (for ethernet0/1) > DHCP Server: Enter the following, then click **OK**:

Server Mode: Enable (select)

CLI

```
set interface ethernet0/1 dhcp server enable
save
```

NOTE: Issuing the CLI command **set interface** interface **dhcp server service** command activates the DHCP server. If the DHCP server detection mode for the interface is set to Auto, the DHCP server on the security device starts only if it does not find an existing server on the network. Issuing the **unset interface** interface **dhcp server service** command disables the DHCP server on the security device and also deletes any existing DHCP configuration.

Assigning a Security Device as a DHCP Relay Agent

When acting as a DHCP relay agent, the security device forwards DHCP requests and assignments between DHCP clients directly attached to one interface and one or more DHCP servers accessible through another interface. The clients and servers may be in the same security zone or in separate security zones.

You can configure a DHCP relay agent on one or more physical or VLAN interfaces on a security device, but you cannot configure a DHCP relay agent and DHCP server or client functions on the same interface.

When the security device functions as a DHCP relay agent, its interfaces must be in either route mode or function as a Layer 3 device. For interfaces in Layer 3 mode (that is, that have IP addresses assigned to the interfaces), you must configure a security policy (from zone to zone or intrazone) to permit the predefined service DHCP-Relay before forwarding occurs.

You can configure up to three DHCP servers for each DHCP relay agent. The relay agent unicasts an address request from a DHCP client to all configured DHCP servers.

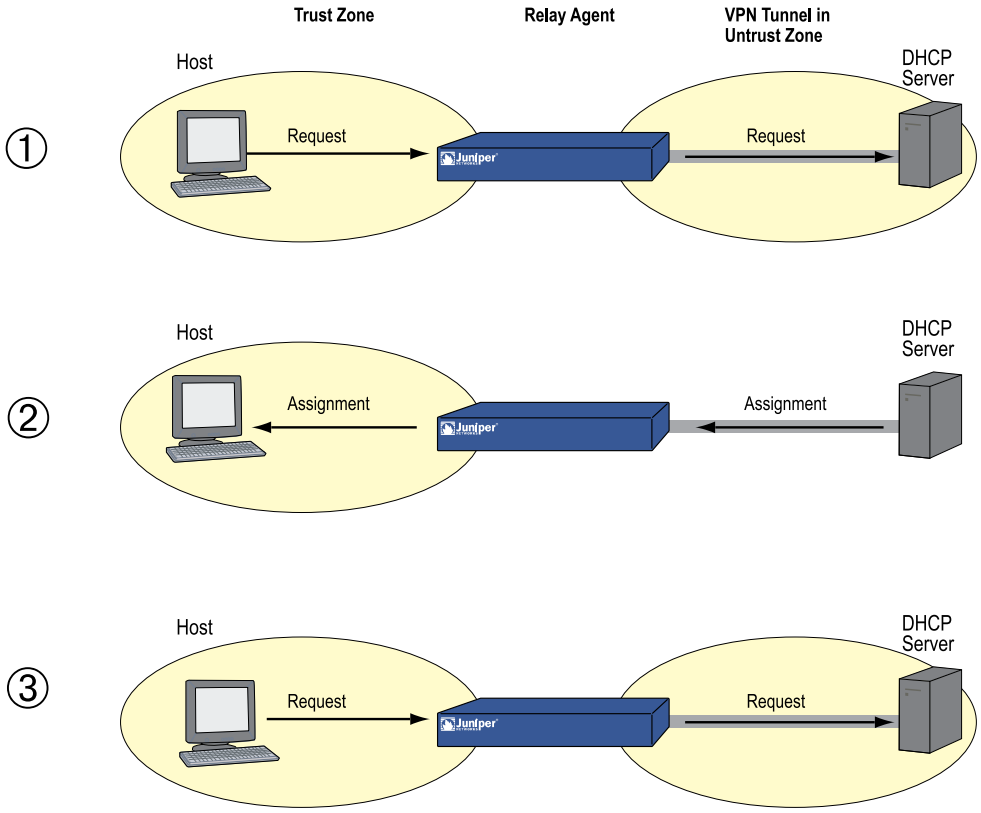
The relay agent forwards to the client all DHCP packets received from all servers. See “Forwarding All DHCP Packets” on page 283.

NOTE: When a security device acts as a DHCP relay agent, the device does not generate DHCP allocation status reports because the remote DHCP server controls all the IP address allocations.

ScreenOS supports DHCP relay in different vsys and for VLAN-tagged subinterfaces.

Figure 74 on page 280 illustrates the process involved in using a security device as a DHCP relay agent. To ensure security, the DHCP messages pass through a VPN tunnel when traveling through the untrusted network.

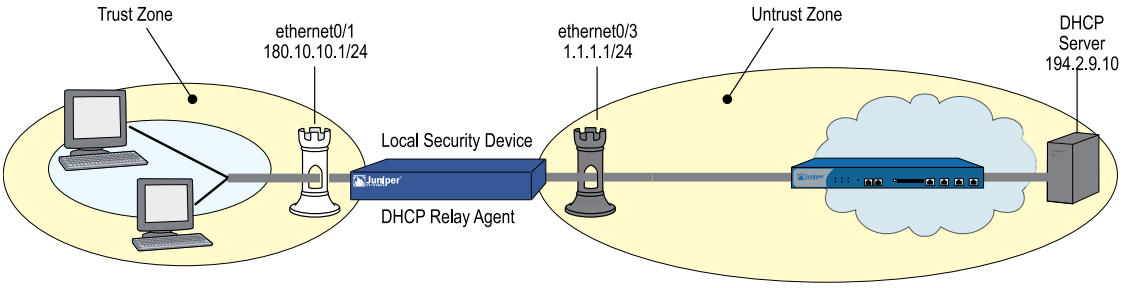
Figure 74: DHCP Relay Agent Traffic



In Figure 75 on page 281, a security device receives its DHCP information from a DHCP server at 194.2.9.10 and relays it to hosts in the Trust zone. The hosts receive IP addresses from an IP pool defined on the DHCP server. The address range is 180.10.10.2—180.10.10.254. The DHCP messages pass through a VPN tunnel between the local security device and the DHCP server, located behind a remote security device whose Untrust zone interface IP address is 2.2.2.2/24. The interface ethernet0/1 is bound to the Trust zone, has the IP address 180.10.10.1/24, and is in

route mode. The interface ethernet0/3 is bound to the Untrust zone and has the IP address 1.1.1.1/24. All security zones are in the trust-vr routing domain.

Figure 75: Device as DHCP Relay Agent



WebUI

1. **Interfaces**

   Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

   Zone: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 180.10.10.1/24

   Enter the following, then click **OK**:

   Interface Mode: Route

   Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

   Zone: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: DHCP Server
   IP Address/Domain Name:
   IP/Netmask: (select), 194.2.9.10/32
   Zone: Untrust

3. **VPN**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: dhcp server
   Security Level: Custom
   Remote Gateway Type:
   Static IP: (select), Address/Hostname: 2.2.2.2

Outgoing Interface: ethernet0/3

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Security Level:
User Defined: Custom (select)
Phase 1 Proposal: rsa-g2-3des-sha
Mode (Initiator): Main (ID Protection)

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: to_dhcp
Security Level: Compatible
Remote Gateway:
Predefined: (select), to_dhcp

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Bind to: None

4. **DHCP Relay Agent**

Network > DHCP > Edit (for ethernet0/1) > DHCP Relay Agent: Enter the following, then click **Apply**:

Relay Agent Server IP or Domain Name: 194.2.9.10
Use Trust Zone Interface as Source IP for VPN: (select)

5. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet0/3
Gateway IP Address: 1.1.1.250

NOTE: Setting a route to the external router designated as the default gateway is essential for both outbound VPN and network traffic. In this example, the security device sends encapsulated VPN traffic to this router as the first hop along its route to the remote security device. In Figure 75 on page 281, the concept is presented by depicting the tunnel passing through the router.

6. **Policies**

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:

Address Book Entry: (select), DHCP Server
Service: DHCP-Relay
Action: Tunnel
Tunnel VPN: to_dhcp
Modify matching outgoing VPN policy: (select)

CLI

1. **Interfaces**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 180.10.10.1/24
set interface ethernet0/1 route
set interface ethernet0/3 zone untrust
set interface ethernet0/3 ip 1.1.1.1/24
```

2. **Address**

```
set address untrust dhcp_server 194.2.9.10/32
```

3. **VPN**

```
set ike gateway “dhcp server”  ip 2.2.2.2 main outgoing-interface ethernet0/3
proposal rsa-g2-3des-sha
set vpn to_dhcp gateway “ dhcp server”  proposal g2-esp-3des-sha
```

4. **DHCP Relay Agent**

```
set interface ethernet0/1 dhcp relay server-name 194.2.9.10
set interface ethernet0/1 dhcp relay vpn
```

5. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet0/3 gateway 1.1.1.250
```

6. **Policies**

```
set policy from trust to untrust any dhcp_server dhcp-relay tunnel vpn to_dhcp
set policy from untrust to trust dhcp_server any dhcp-relay tunnel vpn to_dhcp
save
```

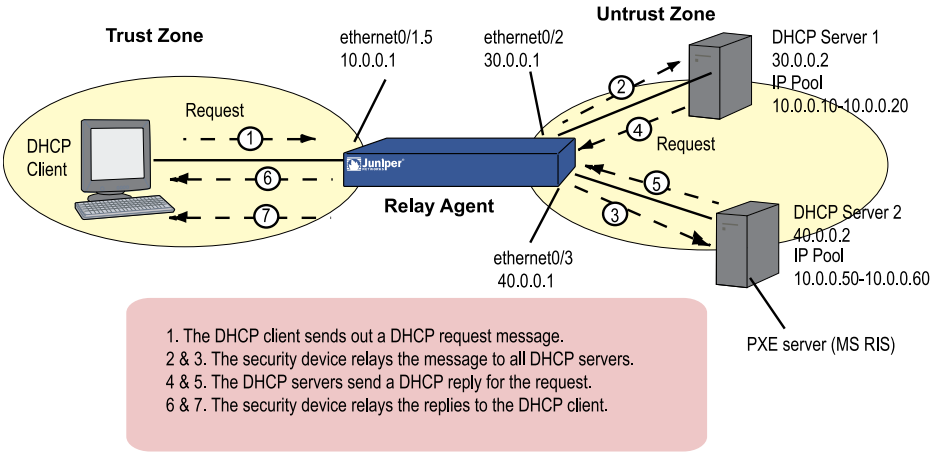
Forwarding All DHCP Packets

ScreenOS lets your security device relay all DHCP responses from multiple servers to a client. Some environments require multiple servers to respond with identical data and their clients only process the first-received response; other environments have multiple servers replying with unique data and the clients process appropriate data from each, for example in a Pre-Boot Execution Environment (PXE) scenario.

In common PXE cases (as shown in Figure 76 on page 284), at least two DHCP servers serve clients. When the DHCP servers receive a request from the DHCP client, one of the servers, DHCP Server1, provides DHCP address information to the client while

DHCP Server 2 (such as MS RIS) provides PXE information. This release of ScreenOS allows the security device to forward all DHCP packets to the client.

Figure 76: Relaying All DHCP Packets from Multiple DHCP Servers



Typically, a PXE server provides a boot-image-server for diskless PXE clients, which are diskless PC machine. When a PXE client powers on, it sends out a broadcast DHCP-DISCOVER (a kind of request), which means that the client requests the IP and boot-image path. In most cases, two kinds of servers serve the PXE: a PXE server (like a Microsoft RIS server) and a DHCP server. Both servers receive the DISCOVER request. The PXE server replies to the DISCOVER request with boot-image-server information. At the same time, the DHCP server replies to the DISCOVER request with an IP-assignment information. Both the responses from the two servers are forwarded to the DHCP client (diskless PC).

Configuring Next-Server-IP

If a security device receives conflicting or confusing information from the DHCP server, the device uses the IP address in the Next-Server-IP field. This DHCP configuration parameter has traditionally been used as the address of the TFTP server in the bootstrap process.

For example, in PXE scenarios, the first DHCP server serves the IP address, and the second DHCP server provides OS information. The Next-Server-IP field is configured to specify the next server in the chain. The chain and each member can vary from site to site. However, typically, it is a DHCP server chaining to a TFTP server. The chain is terminated either by supplying all zeroes (0.0.0.0) or by specifying the device interface IP into this field as shown in Table 38 on page 285.

This Next-Server-IP information is returned in the siaddr field of the DHCP header and is often used to chain several bootstrap servers together, with each serving a specific function. The siaddr field is mandatory, if available, because some DHCP servers will place their own IP address in this field when it is not configured.

Table 38: Specifying Next-Server-IP

Next Server IP	Description
None (default)	siaddr = 0.0.0.0 (default)
Interface	siaddr = the IP interface bound to the DHCP server
Option66	siaddr = option66 (identifies the TFTP server for supporting diskless PCs)
Input	siaddr = custom IP address

If the Next-Server-IP is non-zero and not equal to this server's address, then it is interpreted by the client as the address of the next server in a chain that supplies additional boot information. In the following example, the Next-Server-IP is configured for Option66.

WebUI

Network > DHCP > Edit (DHCP Server): Select one of the following, then click **Apply**:

Next Server IP: From Option66

CLI

```
set interface ethernet0/1 dhcp server enable
set interface ethernet0/1 dhcp server option custom 66 ip 10.10.10.1
set interface ethernet0/1 dhcp server config next-server-ip option66
save
```

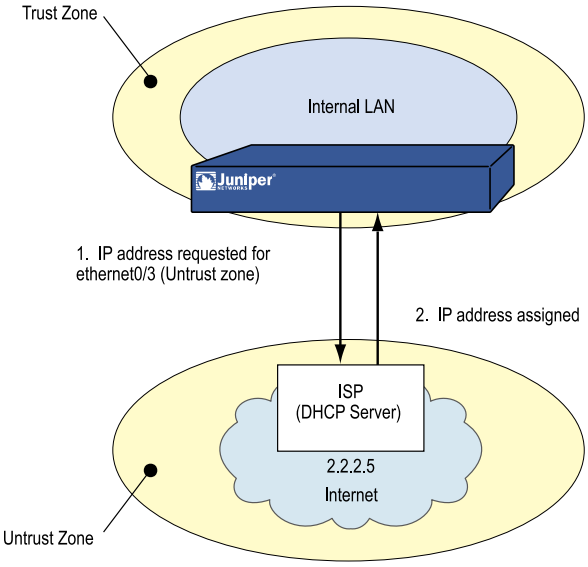
Using a Security Device as a DHCP Client

When acting as a DHCP client, the security device receives an IP address dynamically from a DHCP server for any physical interface in any security zone. If multiple interfaces bound to a single security zone exist, you can configure a DHCP client for each interface as long as each interface is not connected to the same network segment. If you configure a DHCP client for two interfaces that are connected to the same network segment, the first address assigned by a DHCP server is used. (If the DHCP client receives an address update to the same IP address, IKE is not rekeyed.)

NOTE: While some security devices can act as DHCP servers, a DHCP relay agents, or DHCP clients at the same time, you cannot configure more than one DHCP role on a single interface.

In this example, the interface bound to the Untrust zone has a dynamically assigned IP address. When the security device requests its IP address from its ISP, it receives its IP address, subnet mask, gateway IP address, and the length of its lease for the address. The IP address of the DHCP server is 2.2.2.5.

Figure 77: Device as DHCP Client



NOTE: Before setting up a site for DHCP service, you must have a Digital Subscriber Line DSL) and an account with an Internet Service Provider (ISP).

WebUI

Network > Interfaces > Edit (for ethernet0/3): Select **Obtain IP using DHCP**, then click **OK**.

NOTE: You cannot specify the IP address of the DHCP server through the WebUI; however, you can do so through the CLI.

CLI

```
set interface ethernet0/3 dhcp client
set interface ethernet0/3 dhcp settings server 2.2.2.5
save
```

Propagating TCP/IP Settings

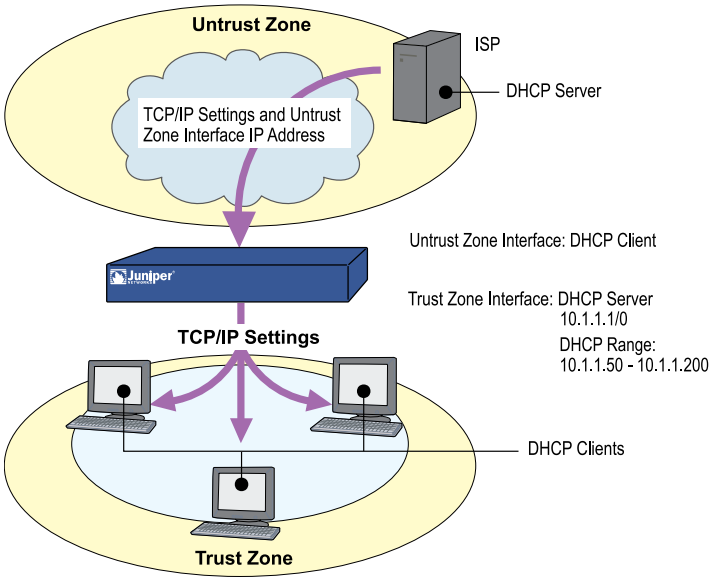
Some security devices can act as a Dynamic Host Control Protocol (DHCP) client, receiving its TCP/IP settings and the IP address for any physical interface in any security zone from an external DHCP server. Some security devices can act as a DHCP server, providing TCP/IP settings and IP addresses to clients in any zone. When a security device acts both as a DHCP client and a DHCP server simultaneously, it can transfer the TCP/IP settings learned through its DHCP client module to its default DHCP server module.

TCP/IP settings include the IP address of the default gateway and a subnet mask, and IP addresses for any or all of the following servers:

- DNS (3)
- WINS (2)
- NetInfo (2)
- SMTP (1)
- POP3 (1)
- News (1)

In Figure 78 on page 287, the security device is both a client of the DHCP server in the Untrust zone and a DHCP server to the clients in the Trust zone. The device takes the TCP/IP settings that it receives as a DHCP client and forwards them as a DHCP server to the clients in the Trust zone. The Untrust Zone Interface is the DHCP client and receives IP addresses dynamically from an ISP.

Figure 78: DHCP Propagation



You can configure the DHCP server module to propagate all TCP/IP settings that it receives from the DHCP client module using the **set interface** *interface* **dhcp-client settings update-dhcpserver** command. You can also override any setting with a different one.

In this example, you configure the security device to act both as a DHCP client on the ethernet0/3 interface and as a DHCP server on the ethernet0/1 interface. (The default DHCP server is on the ethernet0/1 interface.)

As a DHCP client, the security device receives an IP address for the ethernet0/3 interface and its TCP/IP settings from an external DHCP server at 211.3.1.6. You enable the DHCP client module in the security device to transfer the TCP/IP settings it receives to the DHCP server module.

You configure the DHCP server module to do the following with the TCP/IP settings that it receives from the DHCP client module:

- Forward the DNS IP addresses to its DHCP clients in the Trust zone.
- Override the default gateway, netmask, SMTP server, and POP3 server IP addresses with the following:
  - 10.1.1.1 (this is the IP address of the ethernet0/1 interface)
  - 255.255.255.0 (this is the netmask for the ethernet0/1 interface)
  - SMTP: 211.1.8.150
  - POP3: 211.1.8.172

NOTE: If the DHCP server is already enabled on the Trust interface and has a defined pool of IP addresses (which is default behavior for certain platforms), you must first delete the IP address pool before you can change the default gateway and netmask.

You also configure the DHCP server module to deliver the following TCP/IP settings that it does not receive from the DHCP client module:

- Primary WINS server: 10.1.2.42
- Secondary WINS server: 10.1.5.90

Finally, you configure the DHCP server module to assign IP addresses from the following IP Pool to the hosts acting as DHCP clients in the Trust zone: 10.1.1.50 – 10.1.1.200.

WebUI

NOTE: You can set this feature only through the CLI.

CLI

1. **DHCP Client**

```
set interface ethernet0/3 dhcp-client settings server 211.3.1.6
set interface ethernet0/3 dhcp-client settings update-dhcpserver
set interface ethernet0/3 dhcp-client settings autoconfig
set interface ethernet0/3 dhcp-client enable
```

2. **DHCP Server**

```
set interface ethernet0/1 dhcp server option gateway 10.1.1.1
set interface ethernet0/1 dhcp server option netmask 255.255.255.0
set interface ethernet0/1 dhcp server option wins1 10.1.2.42
set interface ethernet0/1 dhcp server option wins2 10.1.5.90
set interface ethernet0/1 dhcp server option pop3 211.1.8.172
```

```
set interface ethernet0/1 dhcp server option smtp 211.1.8.150
set interface ethernet0/1 dhcp server ip 10.1.1.50 to 10.1.1.200
set interface ethernet0/1 dhcp server service
save
```

Configuring DHCP in Virtual Systems

ScreenOS now fully supports DHCP client-server relay for virtual systems. You can configure DHCP relay for a specific vsys and relay all packets from multiple DHCP servers to a client.

NOTE: DHCP client-server relay is supported only on Ethernet-related interfaces.

Setting DHCP Message Relay in Virtual Systems

ScreenOS allows you to configure Dynamic Host Configuration Protocol (DHCP) message relay from one or multiple DHCP servers to clients within a virtual system (vsys). You can configure DHCP message relay on an interface that is available to a virtual system.

If you have two DHCP servers, server 1 and server 2, a security device sitting between the DHCP servers and a client individually passes DHCP requests to each DHCP server on different outgoing interfaces. As each DHCP reply is received, the security device passes them to the root vsys and then forwards them to the appropriate DHCP client within a vsys.

Figure 79: DHCP Relay Services Within a Vsys

To configure DHCP with vsys:

1. Create a virtual system.
2. Enable DHCP for that vsys.
3. Configure a static route to allow the DHCP server in the root system to access the vsys.
4. Set security policies in the virtual system.

Point-to-Point Protocol over Ethernet

PPP-over-Ethernet (PPPoE) merges the Point-to-Point Protocol (PPP), which is usually used for dialup connections, with the Ethernet protocol, which can connect multiple users at a site to the same customer premises equipment. Many users can share the same physical connection, but access control, billing, and type of service are handled for each user. Some security devices support a PPPoE client, allowing them to operate compatibly on DSL, Ethernet Direct, and cable networks run by ISPs using PPPoE for their clients' Internet access.

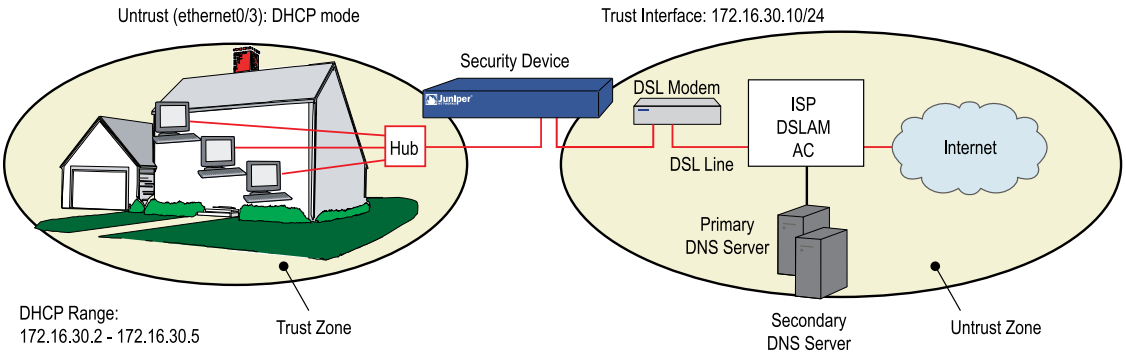
On devices that support PPPoE, you can configure a PPPoE client instance on any or all interfaces. You configure a specific instance of PPPoE with a username, password, and other parameters, and then you bind the instance to an interface. When two Ethernet interfaces (a primary and a backup) are bound to the Untrust zone, you can configure one or both interfaces for PPPoE.

Setting Up PPPoE

The following example illustrates how to define the untrusted interface of a security device for PPPoE connections and how to initiate PPPoE service.

In this example, the security device receives a dynamically assigned IP address for its Untrust zone interface (ethernet0/3) from the ISP, and the security device also dynamically assigns IP addresses for the three hosts in its Trust zone. In this case, the security device acts both as a PPPoE client and a DHCP server. The Trust zone interface must be in either NAT or route mode. In this example, it is in NAT mode.

Figure 80: PPPoE



Before setting up the site in this example for PPPoE service, you must have the following:

- Digital subscriber line (DSL) modem and line
- Account with ISP
- Username and password (obtained from the ISP)

WebUI

1. **Interfaces and PPPoE**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

   Zone: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 172.16.30.10/24

   Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

   Zone: Untrust
   Obtain IP using PPPoE: (select)
   User Name/Password: *name/password*

   Network > Interfaces > Edit (for ethernet0/3): To test your PPPoE connection, click **Connect**.

NOTE: When you initiate a PPPoE connection, your ISP automatically provides the IP addresses for the Untrust zone interface and for the Domain Name System (DNS) servers. When the security device receives DNS addresses by PPPoE, the new DNS settings overwrite the local settings by default. If you do not want the new DNS settings to replace the local settings, you can use the CLI command **unset pppoe dhcp-updateserver** to disable this behavior.
If you use a static IP address for the Untrust zone interface, you must obtain the IP addresses of the DNS servers and manually enter them on the security device and on the hosts in the Trust zone.

2. **DHCP Server**

   Network > Interfaces > Edit (for ethernet0/1) > DHCP: Select **DHCP Server**, then click **Apply**.

   Network > Interfaces > Edit (for ethernet0/1) > DHCP: Enter the following, then click **Apply**:

   Lease: 1 hour
   Gateway: 0.0.0.0
   Netmask: 0.0.0.0
   DNS#1: 0.0.0.0

   > Advanced: Enter the following, then click **Return**:

   DNS#2: 0.0.0.0
   Domain Name: (leave blank)

   Network > Interfaces > DHCP (for ethernet0/1) > New Address: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 172.16.30.2
IP Address End: 172.16.30.5

3. **Activating PPPoE on the Security Device**

1. Turn off the power to the DSL modem, the security device, and the three workstations.
2. Turn on the DSL modem.
3. Turn on the security device.

   The security device makes a PPPoE connection to the ISP and, through the ISP, gets the IP addresses for the DNS servers.

4. **Activating DHCP on the Internal Network**

   Turn on the workstations.

   The workstations automatically receive the IP addresses for the DNS servers. They get an IP address for themselves when they attempt a TCP/IP connection.

NOTE: When you use DHCP to assign IP addresses to hosts in the Trust zone, the security device automatically forwards the IP addresses of the DNS servers that it receives from the ISP to the hosts.

If the IP addresses for the hosts are not dynamically assigned through DHCP, you must manually enter the IP addresses for the DNS servers on each host.

Every TCP/IP connection that a host in the Trust zone makes to the Untrust zone automatically goes through the PPPoE encapsulation process.

CLI

1. **Interfaces and PPPoE**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 172.16.30.10/24
set interface ethernet0/3 zone untrust
set pppoe interface ethernet0/3
set pppoe username name_str password pswd_str
```

To test your PPPoE connection:

```
exec pppoe connect
get pppoe
```

2. **DHCP Server**

```
set interface ethernet0/1 dhcp server service
set interface ethernet0/1 dhcp server ip 172.16.30.2 to 172.16.30.5
set interface ethernet0/1 dhcp server option lease 60
```

save

3. **Activating PPPoE on the Security Device**

1. Turn off the power to the DSL modem, the security device, and the three workstations.
2. Turn on the DSL modem.
3. Turn on the security device.

4. **Activating DHCP on the Internal Network**

Turn on the workstations.

The workstations automatically receive the IP addresses for the DNS servers. They get an IP address for themselves when they attempt a TCP/IP connection.

Every TCP/IP connection that a host in the Trust zone makes to the Untrust zone automatically goes through the PPPoE encapsulation process.

Configuring PPPoE on Primary and Backup Untrust Interfaces

In the following example, you configure PPPoE for both the primary (ethernet0/3) and backup (ethernet0/2) interfaces to the Untrust zone.

WebUI

1. **PPPoE Configuration for ethernet0/3 Interface**

Network > PPP > PPPoE Profile > New: Enter the following, then click **OK**:

PPPoE instance: eth3-pppoe
Bound to interface: ethernet0/3 (select)
Username: user1
Password: 123456
Authentication: Any (select)
Access Concentrator: ac-11

2. **PPPoE Configuration for ethernet0/2 Interface**

Network > PPP > PPPoE Profile > New: Enter the following, then click **OK**:

PPPoE instance: eth2-pppoe
Bound to interface: ethernet0/2 (select)
Username: user2
Password: 654321
Authentication: Any (select)
Access Concentrator: ac-22

CLI

1. **PPPoE Configuration for ethernet0/3 Interface**

```
set pppoe name eth3-pppoe username user1 password 123456
set pppoe name eth3-pppoe ac ac-11
set pppoe name eth3-pppoe authentication any
set pppoe name eth3-pppoe interface ethernet0/3
```

2. **PPPoE Configuration for ethernet0/2 Interface**

```
set pppoe name eth2-pppoe username user2 password 654321
set pppoe name eth2-pppoe ac ac-22
set pppoe name eth2-pppoe authentication any
set pppoe name eth2-pppoe interface ethernet0/2
save
```

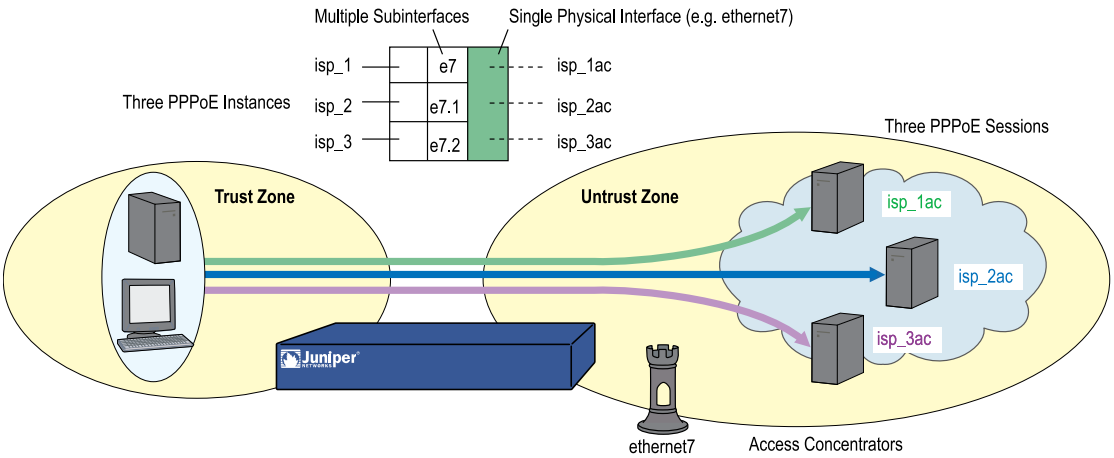
Configuring Multiple PPPoE Sessions over a Single Interface

Some security devices support creation of multiple PPPoE subinterfaces (each with the same MAC address) for a given physical interface. This support allows you to establish a private network connection with one ISP and connect to the Internet through a different ISP using the same physical interface. You can establish these connections using different user or domain names or be connected simultaneously to different ISPs.

The maximum number of concurrent PPPoE sessions on a physical interface is limited only by number of subinterfaces allowed by the device. There is no restriction on how many physical interfaces can support multiple sessions. You can specify username, static-ip, idle-timeout, auto-connect, and other parameters separately for each PPPoE instance or session.

To support a PPPoE session, a subinterface must be untagged. An untagged interface uses encap (not a VLAN tag) to identify a VLAN for a subinterface. Encap binds the subinterface to PPPoE encapsulation. By hosting multiple subinterfaces, a single physical interface can host multiple PPPoE instances. You can configure each instance to go to a specified Access Concentrator (AC), allowing separate entities such as ISPs to manage the PPPoE sessions through a single interface. For more information about VLANs and VLAN tags, see “Virtual Systems” on page 1677.

Figure 81: PPPoE with Multiple Sessions



In the following example, you define three PPPoE instances, specify an Access Concentrator (AC) for each, then initiate each instance.

- Instance isp_1, username **user1@domain1**, password **swordfish**, bound to interface ethernet7. The AC is **isp_1ac**.
- Instance isp_2, username **user2@domain2**, password **marlin**, bound to subinterface ethernet7.1. The AC is **isp_2ac**.
- Instance isp_3, username **user3@domain3**, password **trout**, bound to subinterface ethernet7.2. The AC is **isp_3ac**.

WebUI

1. **Interface and Subinterfaces**

   Network > Interfaces > Edit (for ethernet7):

   Enter the following, then click **OK**:

   Zone Name: Untrust

   Network > Interfaces > New (Sub-IF):

   Enter the following, then click **OK**:

   Interface Name: ethernet7.1
   Zone Name: Untrust

   Network > Interfaces > New (Sub-IF):

   Enter the following, then click **OK**:

   Interface Name: ethernet7.2
   Zone Name: Untrust

2. **PPPoE Instances and AC**

   Network > PPP > PPPoE Profile > New:

   Enter the following, then click **OK**:

   PPPoE Instance: isp_1
   Enable: Enable
   Bound to Interface: ethernet7
   Username: user1@domain1

   Network > PPP > PPPoE Profile > New:

   Enter the following, then click **OK**:

   PPPoE Instance: isp_2
   Enable: Enable
   Bound to Interface: ethernet7.1
   Username: user2@domain2
   Access Concentrator: isp_2ac

Network > PPP > PPPoE Profile > New:

Enter the following, then click **OK**:

PPPoE Instance: isp_3
Enable: Enable
Bound to Interface: ethernet7.2
Username: user3@domain3
Access Concentrator: isp_3ac

3. **PPPoE Initiation**

Network > PPP > PPPoE Profile > Connect (for isp_1)

Network > PPP > PPPoE Profile > Connect (for isp_2)

Network > PPP > PPPoE Profile > Connect (for isp_3)

CLI

1. **Interface and Subinterfaces**

```
set interface ethernet7 zone untrust
set interface ethernet7.1 encap pppoe zone untrust
set interface ethernet7.2 encap pppoe zone untrust
```

2. **PPPoE Instances and ACs**

```
set pppoe name isp_1 username user1@domain1 password swordfish
set pppoe name isp_1 interface ethernet7
set pppoe name isp_1 ac isp_1ac
set pppoe name isp_2 username user2@domain2 password marlin
set pppoe name isp_2 interface ethernet7.1
set pppoe name isp_2 ac isp_2ac
set pppoe name isp_3 username user3@domain3 password trout
set pppoe name isp_3 interface ethernet7.2
set pppoe name isp_3 ac isp_3ac
save
```

3. **PPPoE Initiation**

```
exec pppoe name isp_1 connect
exec pppoe name isp_2 connect
exec pppoe name isp_3 connect
```

PPPoE and High Availability

Two security devices that support PPPoE in Active/Active mode can handle failover of a PPPoE connection. Upon initiation of the connection, the primary device synchronizes its PPPoE state with the backup device. Because the passive device uses the same IP address as the primary device, it does not have to make a new PPPoE connection once it becomes the primary. Therefore, it can maintain communication with the Access Concentrator after failure of the primary. This is

necessary when the PPPoE interface supports VPN connections, and these connections must continue, using the same interface IP after failover. PPPoE for IPv6 also supports Netscreen Redundancy Protocol (NSRP). For more information about HA configurations, see *"High Availability" on page 1763*.

License Keys

The license key feature allows you to expand the capabilities of your Juniper Networks security device without having to upgrade to a different device or system image. You can purchase the following type of keys:

- Advanced
- Capacity
- Extended
- Virtualization
- GTP
- Vsys
- IDP

Each security device ships with a standard set of features enabled and might support the activation of optional features or the increased capacity of existing features. For information regarding which features are currently available for upgrading, see the latest marketing literature from Juniper Networks.

The procedure for obtaining and applying a license key is as follows:

1. Gather your authorization code and device serial number.

   Authorization Code: A pass key required to generate and activate the license key that you or your company have purchased for your Juniper Networks security device. Note: The Authorization Code is required to generate your license key—it is not the actual license key.

   Device Serial Number: A unique 16-character code Juniper Networks uses to identify your particular security device when generating license keys. You can find the device serial number at the bottom or back of the device. You can also find the serial number in the device information section in the GUI or by executing a "get system" command on the CLI.

2. Sign into the Juniper Networks License Management System (LMS) at www.juniper.net/generate_license/, select **Firewall/IPSec VPN and Intrusion Prevention**, then follow the instructions in the system user interface.
3. The LMS provides the license key in one of two ways:
   - Download the license key to your computer.
   - Receive an email that contains your license key.
4. Install the license key in one of the following ways:

WebUI

Configuration > Update > ScreenOS/Keys > Select **License Key Update (Features)** > click **Browse** > select the file with the license key, then click **Apply**.

CLI

```
exec license-key key_num
```

Configuration Files

A configuration file contains all of the information that administrators have configured on a security device, such as system parameters, access policies, VPN configurations, user-defined addresses and services, and user database settings. You can use this data to configure other security devices or as a backup in case of a failure.

CAUTION: A configuration file is only valid for security devices of the same model. Do not attempt to load the configuration file for a security device model onto a different device model.

Uploading Configuration Files

You can upload a configuration file through the WebUI or the CLI. You can either merge the new configuration to the existing configuration or replace the existing configuration with the new one. When you upload a new configuration file through the WebUI, the integrity of the configuration file is not guaranteed. To solve this problem, ScreenOS enables you to provide MD5 checksum of the uploaded configuration file. The MD5 hash should be in hexadecimal format with 32 hex. digits.

NOTE: If the MD5 hash you provide is invalid, the security device returns an error message.

When the device receives the new configuration file, it generates the MD5 checksum. This checksum is compared with the one provided by the user. If the checksums match, the device saves the new configuration file. If the checksums do not match, the device displays an error message stating that the MD5 hash value generated for the configuration file does not match the MD5 hash value provided.

CAUTION: Before upgrading a security device, save the existing configuration file to avoid losing any data.

WebUI

1. **Merging with Current Configuration**

   Configuration > Update > Config File: Select **Merge to Current Configuration**, click **Browse**, select the file, then click **Apply**.

NOTE: The new configuration overwrites the VLAN1 IP address of the existing configuration (if set) and any IP addresses for interfaces common to both configurations.

2. **Replacing the Current Configuration**

   Configuration > Update > Config File: Select **Replace Current Configuration**, click **Browse ,** select the file, then click **Apply**.

3. **Providing MD5 Hash Checksum (Optional)**

   Configuration > Update > Config File: Enter the MD5 hash, then click **Apply**.

CLI

save config from tftp to { flash | slot1 | tftp }...

Downloading Configuration Files

You can download a configuration file either through the WebUI or the CLI.

WebUI

Configuration > Update > Config File: Click **Save To File**.

CLI

save config to { flash | slot1 | tftp }...

NOTE: When the SSG device initializes, and if the administrator has configured envar properly, then ScreenOS can check if the USB device is connected to the port and loads the configuration file usb: auto_config.txt (if the file is stored in the USB device). To activate the security device, reset the device once uploading is complete.

For example, you can use the following command to check if the USB device is connected to the port:

```
set envar config=usb:my-config.txt
```

Registration and Activation of Subscription Services

Before your Juniper Networks security device can receive regular subscription service for antivirus (AV) patterns, Deep Inspection (DI) signatures, antispam, or Web filtering, you must do the following:

- Purchase a subscription
- Register the subscription
- Retrieve the subscription key

Retrieving the subscription license key activates your services on the device for the time period purchased. Your specific service-activation process depends upon which services you purchased and the method you used to purchase them.

Trial Service

To allow you to use AV, DI, antispam, or Web-filtering services, the security device provides a trial period. During this period, the device can obtain temporary services. To retrieve eligible trial license keys from the entitlement server, use the **exec license-key** update trials CLI command.

- No Juniper Networks security device comes with DI already enabled. To obtain trial DI service, you must start a WebUI session, then click **Retrieve Subscriptions Now** in Configuration > Update > ScreenOS/Keys. This action provides a one-time, one-day DI key.
- If your device has AV service bundled at the time of purchase, then the device has preinstalled trial service. This trial service lasts up to 60 days.
- Antispam
- No Juniper Networks security device comes with Web filtering already enabled. This feature does not have a trial service.

CAUTION: To avoid service interruption, you must register your Juniper Networks security device as soon as possible after purchasing your subscription. Registration ensures continuation of the subscription.

Updating Subscription Keys

If there is any software with an expiration date installed in the security device, the device periodically connects to the entitlement server to retrieve the subscription keys. The device connects to the entitlement server, the LMS, under all of following conditions:

- Key expires in two months
- Key expires in one month
- Key expires in two weeks

- Key expires
- 30 days after key expires

NOTE: To delete a single license key from the key file, use the **exec license-key delete** *name_str* CLI command.

Adding Antivirus, Web Filtering, Antispam, and Deep Inspection to an Existing or a New Device

After purchasing AV, Web filtering, antispam, or deep inspection (DI) subscriptions to add to your existing Juniper Networks security device, perform the following steps to activate the services:

1. After ordering the subscription, you should receive an authorization code, via email, from Juniper Networks or your authorized VAR. This code is a readable document that contains information you need to register your subscription.
2. Make sure the device is registered. If it is not currently registered, go to the following site:

   http://tools.juniper.net/subreg
3. Register the subscription authorization code to the device.
4. Confirm that your device has Internet connectivity.
5. Retrieve the subscription key on the device. You can do this in one of two ways:
   - In the WebUI, click **Retrieve Subscriptions Now** from Configuration > Update > ScreenOS/Keys.
   - Using the CLI, run the following command:

     ```
     exec license-key update
     ```
6. You must reset the device after the key has been loaded.

You can now configure the device to automatically or manually retrieve the signature services. For instructions on configuring your security device for these services, see the following sections:

- Fragment Reassembly on page 495
- Antivirus Scanning on page 211
- Web Filtering on page 210

System Clock

It is important that your Juniper Networks security device always be set to the right time. Among other things, the time on your device affects the set up of VPN tunnels and the timing of schedules. First, to ensure that the device always maintains the

proper time, you must set the system clock to the current time. Next, you can enable the daylight saving time (DST) option, and you can configure up to three Network Time Protocol (NTP) servers (one primary and two backups) from which the device can regularly update its system clock.

Date and Time

To set the clock to the current date and time, you can use the WebUI or the CLI. Through the WebUI, you do this by synchronizing the system clock with the clock on your computer:

1. Configuration > Date/Time: Click the **Sync Clock with Client** button.

   A pop-up message prompts you to specify if you have enabled the DST option on your computer clock.

2. Click **Yes** to synchronize the system clock and adjust it according to DST, or click **No** to synchronize the system clock without adjusting it for DST.

Through the CLI, you set the clock by manually entering the date and time using the following command:

```
set clock mm/dd/yyyy hh:mm:ss
```

Daylight Saving Time

Daylight saving time (DST) is a widely used system of adjusting the official local time forward in summer months in order to save energy and allow more daylight for work and school activities.

DST is observed differently in various countries. Accordingly, you can choose the appropriate DST settings that apply to your country.

You can set the DST adjustment to recur on a weekday schedule (for example, the first Sunday in April) or on a specific date. You also can set DST adjustment not to recur, in which case you can adjust DST settings only for a single year.

Time Zone

You set the time zone by specifying the number of hours by which the local time for the security device is behind or ahead of GMT (Greenwich Mean Time). For example, if the local time zone for the device is Pacific Standard Time, it is 8 hours behind GMT. Therefore, you have to set the clock to **-8**.

If you set the time zone using the WebUI:

Configuration > Date/Time > Set Time Zone_hours_minutes from GMT

If you set the time zone using the CLI:

```
device -> set clock timezone number (a number from -12 to 12)
```

or

device-> set ntp timezone *number* (a number from -12 to 12)

Network Time Protocol

To ensure that the security device always maintains the right time, it can use Network Time Protocol (NTP) to synchronize its system clock with that of an NTP server over the Internet. You can do this manually or configure the device to perform this synchronization automatically at time intervals that you specify.

Configuring Multiple NTP Servers

You can configure up to three NTP servers on a Juniper Networks security device: one primary server and two backup servers. When you configure the security device to synchronize its system clock automatically, it queries each configured NTP server sequentially. The device always queries the primary NTP server first. If the query is not successful, the device then queries the first backup NTP server and so on until it gets a valid reply from one of the NTP servers configured on the device. The device makes four attempts on each NTP server before it terminates the update and logs the failure.

When you manually synchronize the system clock, and you can only do this using the CLI, you can specify a particular NTP server or none at all. If you specify an NTP server, the security device queries that server only. If you do not specify an NTP server, the device queries each NTP server configured on the device sequentially. You can specify an NTP server using its IP address or its domain name.

Configuring a Backup NTP Server

You can specify an individual interface as the source address to direct NTP requests from the device (over a VPN tunnel to the primary NTP server) or to a backup server. You can also select a loopback interface to perform this function.

The security device sends NTP requests from a source interface and optionally uses an encrypted, preshared key when sending NTP requests to the NTP server. The key provides authentication.

In the following example, you configure the primary NTP server and two backup NTP servers by assigning an IP address to each server. Next, you set each server to send NTP requests from the Trust interface. After that, you set the key ID and preshared key for each server.

WebUI

Configuration > Date/Time: Enter the following, then click **Apply**:

Primary Server IP/Name: 1.1.1.1
Primary server Key ID: 10
Source interface: Select Trust from the list.
Preshared Key: !2005abc
Backup Server1 IP/Name: 1.1.1.2
Primary server Key ID: 10
Source interface: Select Trust from the list.

Preshared Key: !2005abc
Backup Server2 IP/Name: 1.1.1.3
Primary server Key ID: 10
Source interface: Select Trust from the list.
Preshared Key: !2005abc

CLI

```
set ntp server 1.1.1.1
set ntp server backup1 1.1.1.2
set ntp server backup2 1.1.1.3
set ntp server src-interface trust
set ntp server backup1 src-interface trust
set ntp server backup2 src-interface trust
set ntp server key-id 10 pre-share-key !2005abc
set ntp server backup1 key-id 10 pre-share-key !2005abc
set ntp server backup2 key-id 10 pre-share-key !2005abc
save
```

Device as an NTP Server

A security device can also work as an NTP server serving NTP requests from its subnet peers. To use this feature, you need to enable NTP service on any of the Layer 3 interface with an IP address.

NOTE: Currently, ScreenOS supports only unicast mode.

WebUI

Network > Interfaces > Edit (for ethernet0/0) > Basic: check the NTP Server check box, then click **Apply .**

CLI

```
set interface interface ntp-server
save
```

Maximum Time Adjustment

For automatic synchronization, you can specify a maximum time adjustment value (in seconds). The maximum time adjustment value represents the acceptable time difference between the security device system clock and the time received from an NTP server. The device only adjusts its clock with the NTP server time if the time difference between its clock and the NTP server time is within the maximum time adjustment value that you set. For example, if the maximum time adjustment value is 3 seconds, and the time on the device system clock is 4:00:00 and the NTP server sends 4:00:02 as the time, the difference in time between the two is acceptable and the device can update its clock. If the time adjustment is greater than the value you

set, the device does not synchronize its clock and proceeds to try the first backup NTP server configured on the device. If the device does not receive a valid reply after trying all the configured NTP servers, it generates an error message in the event log. The default value for this feature is 3 seconds and the range is 0 (no limit) to 3600 (one hour).

When you manually synchronize the system clock, and you can only do this using the CLI, the security device does not verify the maximum time adjustment value. Instead, if it receives a valid reply, the device displays a message informing you of which NTP server it reached, what the time adjustment is, and the type of authentication method used. The message also asks you to confirm or cancel the system clock update.

If the security device does not receive a reply, it displays a timeout message. This message appears only after the device unsuccessfully attempted to reach all NTP servers configured on the device.

NOTE: When issuing requests using the CLI, you can cancel the current request by pressing **Ctrl-C** on the keyboard.

NTP and NSRP

NetScreen Redundancy Protocol (NSRP) contains a mechanism for synchronizing the system clock of NSRP cluster members. Although the resolution for synchronization is in seconds, NTP has sub-second resolution. Because the time on each cluster member might differ by a few seconds due to processing delays, we recommend that you disable NSRP time synchronization when NTP is enabled on both cluster members (meaning that each member can updates its system clock from an NTP server). To disable the NSRP time synchronization function, enter the **set ntp no-ha-sync** CLI command.

Setting a Maximum Time Adjustment Value to an NTP Server

In the following example you configure the security device to update its clock at five-minute intervals from NTP servers at IP addresses 1.1.1.1, 1.1.1.2, and 1.1.1.3. You also set a maximum time adjustment value of 2 seconds.

WebUI

Configuration > Date/Time: Enter the following, then click **Apply**:

Automatically synchronize with an Internet Time Server (NTP): (select)
Update system clock every minutes: 5
Maximum time adjustment seconds: 2
Primary Server IP/Name: 1.1.1.1
Backup Server1 IP/Name: 1.1.1.2
Backup Server2 IP/Name: 1.1.1.3

CLI

```
set clock ntp
set ntp server 1.1.1.1
set ntp server backup1 1.1.1.2
set ntp server backup2 1.1.1.3
set ntp interval 5
set ntp max-adjustment 2
save
```

Securing NTP Servers

You can secure NTP traffic by using MD5-based checksum to provide authentication of NTP packets. You do not need to create an IPsec tunnel. This type of authentication ensures the integrity of NTP traffic. It does not prevent outside parties from viewing the data, but it prevents anyone from tampering with it.

To enable the authentication of NTP traffic, you must assign a unique key ID and preshared key to each NTP server you configure on a security device. The key ID and preshared key serve to create a checksum with which the security device and the NTP server can authenticate the data.

Table 39 on page 306 describes the two types of authentication for NTP traffic.

Table 39: NTP Traffic Authentication

Type	Description
Required	When selected, the security device must include the authentication information—key id and checksum—in every packet it sends to an NTP server and must authenticate all NTP packets it receives from an NTP server. Before authentication can occur between a security device and an NTP server, the administrators of the security device and the NTP server must first exchange a key id and a preshared key. They have to exchange these manually and can do so in different ways such as via email or telephone.
Preferred	When selected, the security device must first operate as in required mode by trying to authenticate all NTP traffic. If all attempts to authenticate fail, the security device then operates as if you selected no authentication. It sends out packets to an NTP server without including a key id and checksum. Essentially, although there is a preference for authentication, if authentication fails, the security device still permits NTP traffic.

Part 3
Administration

Juniper Networks security devices provide different ways for you to manage the devices, either locally or remotely. *Administration* contains the following chapters:

- "Administration" on page 309 explains the different means available for managing a security device both locally and remotely. This chapter also explains the privileges pertaining to each of the four levels of network administrators that can be defined.
- "Monitoring Security Devices" on page 371 explains various monitoring methods and provides guidance in interpreting monitoring output.

Chapter 10
Administration

This chapter describes management methods and tools, methods for securing administrative traffic, and the administrative privilege levels that you can assign to admin users. This chapter contains the following sections:

Federal Information Processing Standards (FIPS)

Federal Information Processing Standards (FIPS) specify the security requirements that a cryptographic module employed within a security system should comply with. FIPS requires that the system provide a self-test function for cryptographic algorithms at power on and conditional test. Juniper Networks security devices comply with FIPS by supporting this self-test on power-on.

Juniper Networks security devices support the self-test functions for the following situations:

- At power on
- On demand
- After key generation
- For periodic self-tests

Table 40 on page 310 lists the algorithms that the system tests as part of the FIPS requirements.

Table 40: Cryptographic Algorithms

DSA	Digital Signatures Algorithm
ECDSA	Elliptic Curve Digital Signatures Algorithm
RSA	Rivest, Shamir and Adleman Algorithm
DES	Data Encryption Standard
DRNG	Deterministic Random Number Generator
SHA	Deterministic Random Number Generator
DH	Diffie-Hellman
HMAC	Keyed-Hash Message Authentication Code
AES	Advanced Encryption Standard

Power-On Self-Test

When the device powers on, the system performs a set of cryptographic algorithm self-tests. These tests run after the hardware self-tests are complete and before the system loads the device configuration files. When the systemcompletes its power-on self-test (POST), it generates a message displaying the results. If the POST fails, the device halts; no traffic is passed and the status LED blinks red.

The system performs the following cryptographic algorithm tests as part of its POST:

- Advanced Encryption Standard (AES)
- Triple Data Encryption Standard (3DES)
- Secure Hash Algorithm 1 (SHA-1) and Secure Hash Algorithm 2 (SHA-256)
- Hashed Message Authentication Code (HMAC) SHA-1
- RSA known-answer test
- Digital Signature Algorithm (DSA) known-answer test
- Elliptical Curve Digital Signature Algorithm (ECDSA) known-answer test
- American National Standards Institute (ANSI) X9.32 DRNG
- Diffie-Hellman (DH) algorithm
- Elliptical Curve Diffie-Hellman (ECDH) algorithm

In addition to these tests, the system also performs the config-data integrity and firmware integrity tests as part of the POST.

Config-Data Integrity Test

In a config-data integrity test, the system calculates the SHA1 value for the configuration data and writes it in a new file. Whenever the system executes the config-data integrity test, it recalculates the hash value based on the configuration data and compares it with the hash value stored in the new file. If both values are the same, the device passes the config-data test.

Similarly, the system calculates the checksum of the public key infrastructure (PKI) database and stores it in flash memory. Whenever the PKI data changes, the checksum is recalculated and stored in the flash. When the config-data integrity test is executed, the system recalculates the checksum using the latest PKI database and compares it with the checksum stored in flash memory. If both checksums match, the PKI database is uncorrupted.

Firmware Integrity Test

Whenever the system administrator downloads the image using the save software command or through the boot loader, the system verifies the digital signature of the image against the original digital image that was signed using DSA. If the verification fails, the system does not write the image to the flash.

In the current release, you can configure a default gateway to download a boot loader or a new image from a TFTP server when upgrading using the boot loader method. After initialization, the boot loader prompts you to provide an input. To upgrade the boot loader from the TFTP server, press the X and A keys simultaneously. You can hit any key for downloading a new image from the TFTP server. For the boot loader to start the TFTP process, you should specify the IP address of your device, the mask of the subnet, the gateway to be used, and the IP address of your TFTP server. If the device address and the TFTP server address are not in the same subnet segment, the boot loader uses the specified gateway to initiate the TFTP process.

Self-Test on Demand by Administrator

The administrator can invoke the FIPS self-test at run time with the **exec fips-mode self-test** command. The cryptographic algorithms that are tested are similar to those tested at self-test on power up. In addition to those tests, the system also performs the config-data and firmware integrity tests as part of the self-test on demand. If the device fails the self-test, the system sends an error message to the console and the buffer and stores it in the event log. If the periodic self-test is running when the administrator invokes the self-test on demand, the system prompts the admin to try again later.

Any audit, cryptographic, or security administrator can execute the self-test on demand.

Self-Test After Key Generation

Administrators can configure the FIPS self-test to run immediately after the generation of a key by using the **set fips-mode self-test afterkeygen** command. This option is

available only for asymmetric cryptographic algorithms such as DSA, RSA, ECDSA, and ECDH. The system will run pair-wise consistency tests on these algorithms.

NOTE: Only cryptographic administrators can enable and disable the self-test after key generation feature.

Periodic Self-Test

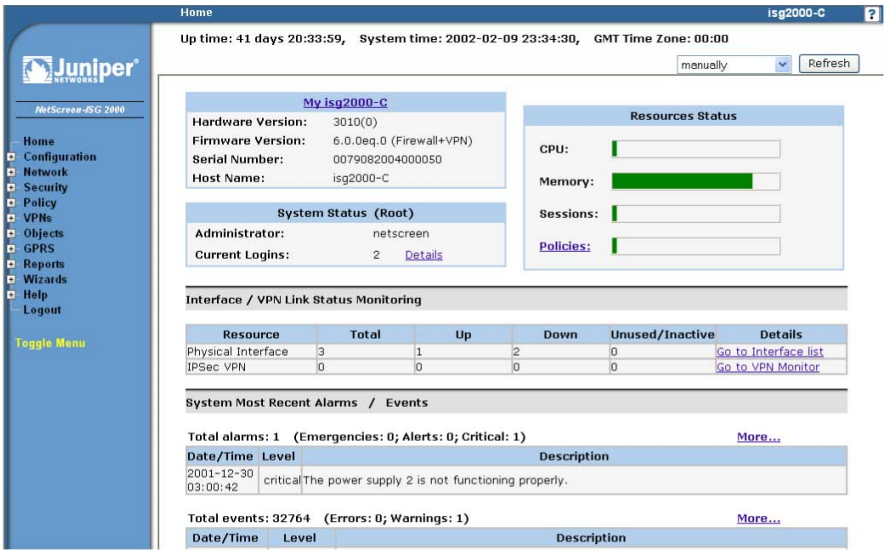
Administrators can also configure the system to run periodic self-tests by using the **set fips-mode self-test interval** command. Administrators can set the run interval for these tests from 1 to 24 hours. The cryptographic algorithms run during periodic self-tests are the same as those run at POST. Additionally, the system also performs the config-data and firmware integrity tests as part of the periodic self-test.

NOTE: Only security administrators can configure the periodic self-test feature.

Management with the Web User Interface

You can use the Web user interface (WebUI) to configure and manage the software for Juniper Networks security devices. Figure 82 on page 312 shows the WebUI window. The left pane contains the navigation menu, and the right pane displays the navigation window.

Figure 82: WebUI



To use the WebUI, you must have the following application and connection:

- Microsoft Internet Explorer (version 5.5 or later) or Netscape Communicator (version 4.7 or later)

- TCP/IP network connection to the security device

WebUI Help

You can view Help files for the WebUI at http://help.juniper.net/help/english/*screenos_version*/*filename*.htm (for example, http://help.juniper.net/help/english/6.2.0/620_Help.htm).

You also have the option of relocating the Help files. You might want to store them locally and point the WebUI to either the administrator's workstation or a secured server on the local network. In case you do not have Internet access, storing the Help files locally provides accessibility to them you otherwise would not have.

Copying the Help Files to a Local Drive

The Help files are available on the documentation CD. You can modify the WebUI to point to the Help files on the CD in your local CD drive. You can also copy the files from the CD to a server on your local network or to another drive on your workstation and configure the WebUI to invoke the Help files from that location.

NOTE: If you want to run the Help files directly from the documentation CD, you can skip this procedure. Proceed to "Pointing the WebUI to the New Help Location" on page 313.

1. Load the documentation CD in the CD drive of your workstation.
2. Navigate to the CD drive and copy the directory named **help**.
3. Navigate to the location where you want to store the Help directory and paste the Help directory there.

Pointing the WebUI to the New Help Location

You must now redirect the WebUI to point to the new location of the Help directory. Change the default URL to the new file path, where *path* is the specific path to the Help directory from the administrator's workstation.

1. Configuration > Admin > Management: In the Help Link Path field, replace the default URL:

   http://help.juniper.net/help/english/*screenos_version/filename*.htm

   with

   (for local drive) file://*path*…/help

   or

   (for local server) http://*server_name*…/*path*/help
2. Click **Apply**.

When you click the **help** link in the upper right corner of the WebUI, the device now uses the new path that you specified in the Help Link Path field to locate the appropriate Help file.

HyperText Transfer Protocol

With a standard browser, you can access, monitor, and control your network security configurations remotely using HyperText Transfer Protocol (HTTP).

You can secure HTTP administrative traffic by encapsulating it in a virtual private network (VPN) tunnel or by using the Secure Sockets Layer (SSL) protocol. You can further secure administrative traffic by completely separating it from network user traffic. To do this, you can run all administrative traffic through the MGT interface—available on some security devices—or bind an interface to the MGT zone and devote it exclusively to administrative traffic.

NOTE: For more information, see "Secure Sockets Layer" on page 315, "MGT and VLAN1 Interfaces" on page 340, and "VPN Tunnels for Administrative Traffic" on page 358.

Session ID

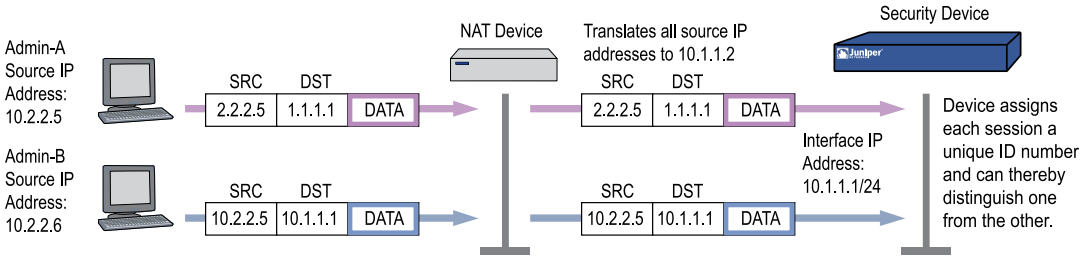
The security device assigns each HTTP administrative session a unique session ID. For security devices that support virtual systems (vsys), the ID is globally unique across all systems—root and vsys.

Each session ID is a 39-byte number resulting from the combination of five pseudo-randomly generated numbers. The randomness of the ID generation—versus a simple numerical incremental scheme—makes the ID nearly impossible to predict. Furthermore, the randomness combined with the length of the ID makes accidental duplication of the same ID for two concurrent administrative sessions extremely unlikely.

The following are two benefits that a session ID provides to administrators:

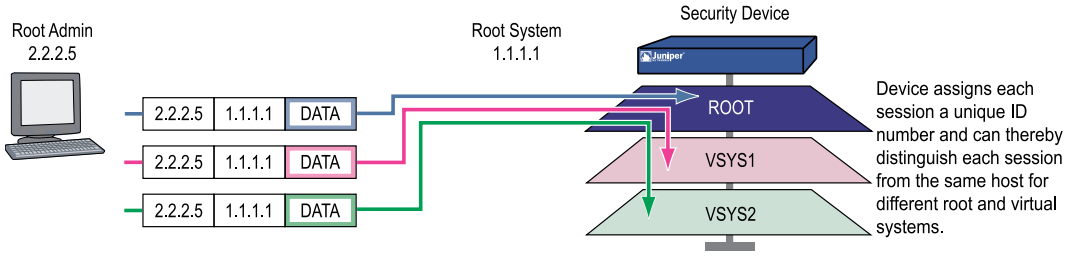
- Figure 83 on page 314 illustrates how the security device can distinguish concurrent sessions from multiple admins behind a NAT device that assigns the same source IP address to all outbound packets.

Figure 83: Session ID with a NAT device



- Figure 84 on page 315 illustrates how the security device can distinguish concurrent root-level admin sessions from the same source IP address to the root system and from there to different virtual systems.

Figure 84: Session ID with Source IP Address



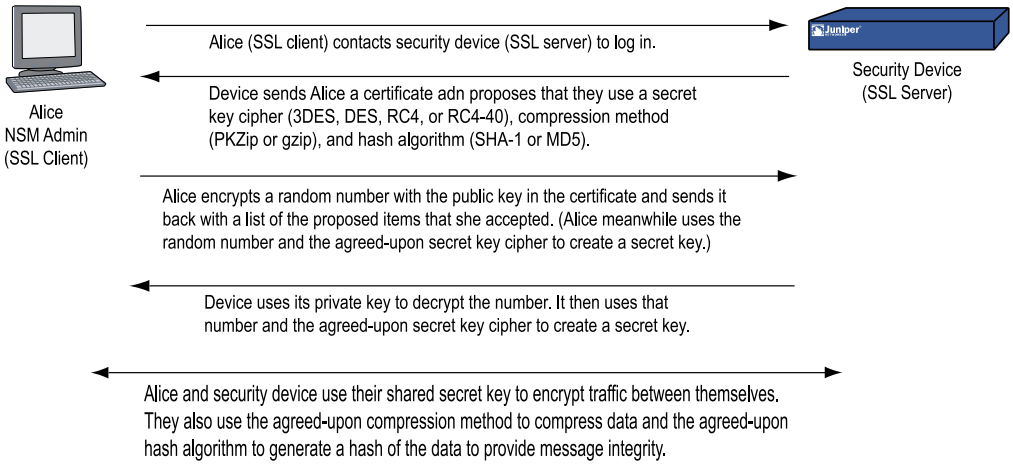
Secure Sockets Layer

Secure Sockets Layer (SSL) is a set of protocols that can provide a secure connection between a Web client and a Web server communicating over a TCP/IP network. SSL consists of the SSL Handshake Protocol (SSLHP), which can allow the client and server to authenticate each other and negotiate an encryption method, and the SSL Record Protocol (SSLRP), which provides basic security services to higher-level protocols such as HTTP. These two protocols operate at the following two layers in the Open Systems Interconnection (OSI) Model:

- SSLHP at the Application Layer (Layer 7)
- SSLRP at the Presentation Layer (Layer 6)

Independent of application protocol, SSL uses TCP to provide secure service (see Figure 85 on page 316). SSL uses certificates to authenticate first the server or both the client and the server, and then encrypt the traffic sent during the session. ScreenOS supports authentication only of the server (security device), not the client (administrator attempting to connect to the security device through SSL).

Figure 85: SSL Client to Server



A Juniper Networks security device can redirect administrative traffic using HTTP (default port 80) to SSL (default port 443). The default certificate for SSL is the automatically generated self-signed certificate, although you can later use a different certificate if you want. Because SSL is integrated with PKI key/certificate management, you can select the SSL certificate from any in the certificate list. You can also use the same certificate for an IPsec VPN.

NOTE: For information about redirecting administrative HTTP traffic to SSL, see “Redirecting HTTP to SSL” on page 318. For information about self-signed certificates, see “Self-Signed Certificates” on page 759. For information on obtaining certificates, see “Certificates and CRLs” on page 746.

The ScreenOS implementation of SSL provides the following capabilities, compatibilities, and integration:

- SSL server authentication (not SSL server and client authentication); that is, the security device authenticates itself to the administrator attempting to connect through SSL, but the administrator does not use SSL to authenticate himself to the device
- SSL version 3 compatibility (not version 2)
- Compatibility with Netscape Communicator 4.7x and later and Internet Explorer 5.x and later
- Public Key Infrastructure (PKI) key management integration (see “Public Key Cryptography” on page 741)
- The following encryption algorithms for SSL:
  - RC4-40 with 40-bit keys
  - RC4 with 128-bit keys

- DES: Data Encryption Standard with 56-bit keys
- 3DES: Triple DES with 168-bit keys

- The same authentication algorithms for SSL as for VPNs:
  - Message Digest version 5 (MD5)—128-bit keys
  - Secure Hash Algorithm version 1 (SHA-1)—160-bit keys
  - Secure Hash Algorithm version 2 (SHA-2)—256-bit keys

NOTE: The RC4 algorithms are always paired with MD5; DES and 3DES are always paired with SHA-1.

SSL Configuration

The basic steps for setting up SSL are as follows:

1. Make use of the self-signed certificate that the security device automatically generates during its initial bootup, or create another self-signed certificate, or obtain a CA-signed certificate and load it on the device.

NOTE: Check your browser to see how strong the ciphers can be and which ones your browser supports. (Both the security device and your browser must support the same kind and size of ciphers you use for SSL.) In Internet Explorer 5x, click **Help**, **About Internet Explorer**, and read the section about cipher strength. To obtain the advanced security package, click **Update Information**. In Netscape Communicator, click **Help**, **About Communicator**, and read the section about RSA. To change the SSL configuration settings, click **Security Info**, **Navigator**, **Configure SSL v3.**
For more information, see "Self-Signed Certificates" on page 759. For details on requesting and loading a certificate, see "Certificates and CRLs" on page 746.

2. Enable SSL management.

NOTE: SSL is enabled by default.

WebUI

Configuration > Admin > Management: Enter the following, then click **Apply**:

SSL: (select)
Port: Use the default port number (443) or change it to another.
Certificate: Select the certificate you intend to use from the drop-down list.
Cipher: Select the cipher you intend to use from the drop-down list.

NOTE: If you change the SSL port number, the admins need to specify the nondefault port number when entering the URL in their browser.

CLI

```
set ssl port num
set ssl cert id_num
set ssl encrypt { { 3des | des } sha-1 | { rc4 | rc4-40 } | md5 }
set ssl enable
save
```

NOTE: To learn the ID number for a certificate, use the following command: **get pki x509 list cert.**

3. Configure the interface through which you manage the security device to permit SSL management:

WebUI

Network > Interfaces > Edit (for the interface you want to manage): Select the SSL management service check box, then click **OK**.

CLI

```
set interface interface manage ssl
save
```

1. Connect to the security device through the SSL port. When you enter the IP address for managing the security device in the browser's URL field, change **http** to **https**, and follow the IP address with a colon and the HTTPS (SSL) port number if you have changed it from the default. For example:

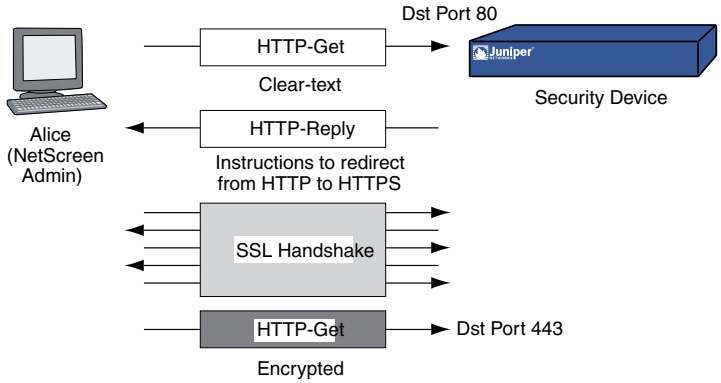
   https://123.45.67.89:1443).

Redirecting HTTP to SSL

The security device can redirect administrative traffic using HTTP (default port 80) to SSL (default port 443), as shown in Figure 86 on page 319.

During the SSL handshake, the security device sends Alice its certificate. Alice encrypts a random number with the public key contained in the certificate and sends it back to the device, which uses its private key to decrypt the number. Both participants then use the shared random number and a negotiated secret key cipher (3DES, DES, RC4, or RC4-40) to create a shared secret key, which they use to encrypt traffic between themselves. They also use an agreed-upon compression method (PKZip or gzip) to compress data and an agreed-upon hash algorithm (SHA-1 or MD-5) to generate a hash of the data to provide message integrity.

Figure 86: Redirection of HTTP to SSL



To enable the redirection and use the default automatically generated self-signed certificate for SSL, do either of the following:

WebUI

Configuration > Admin > Management: Enter the following, then click **Apply**:

Redirect HTTP to HTTPS**:** (select)
Certificate: Default – System Self-Signed Cert

CLI

```
set admin http redirect
save
```

NOTE: You do not have to enter a CLI command to apply the automatically generated self-signed certificate for use with SSL because the security device applies it to SSL by default. If you have previously assigned another certificate for use with SSL and you now want to use the default certificate instead, you must unset the other certificate with the **unset ssl cert** *id_num* command, in which *id_num* is the ID number of the previously assigned certificate.

Although HTTP does not provide the security that SSL does, you can configure the security device so that it does not redirect HTTP traffic. To disable the HTTP-to-SSL redirect mechanism, clear the Redirect HTTP to HTTPS option in the WebUI, or enter the **unset admin http redirect** CLI command.

Management with the Command Line Interface

Advanced administrators can attain finer control by using the command line interface (CLI). To configure a security device with the CLI, you can use any software that emulates a VT100 terminal. With a terminal emulator, you can configure the security device using a console from any Windows, UNIX, or Macintosh operating system. For remote administration through the CLI, you can use Telnet or Secure Shell (SSH). With a direct connection through the console port, you can use HyperTerminal.

NOTE: For a complete listing of the ScreenOS CLI commands, see the *ScreenOS CLI Reference Guide: IPv4 Command Descriptions*.

Telnet

Telnet is a login and terminal emulation protocol that uses a client/server relationship to connect to and remotely configure network devices over a TCP/IP network. You can create a connection with the Telnet client program on the security device by launching a Telnet server program on the admin workstation or other security device. After logging in, the administrator can issue CLI commands, which are sent to the Telnet program on the security device, effectively configuring the device as if operating through a direct connection. Using Telnet to manage security devices requires the following application and connection:

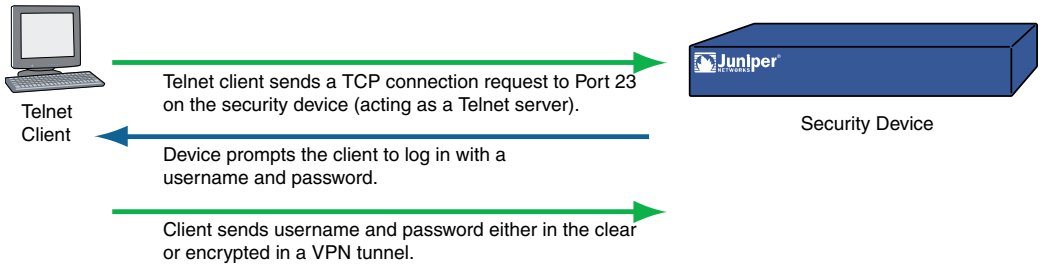
- Telnet software on the admin workstation or other security device
- Ethernet connection to the security device

NOTE: The Telnet client program is not available at the vsys level.

If you want to remotely check service availability using a telnet client, we recommend that you connect to a Juniper Networks security device using Telnet/SSH rather than console

Figure 87 on page 320 illustrates the setup procedure for establishing a Telnet connection.

Figure 87: Establishing a Telnet Connection



In this example, you telnet to a security device using a specific port and source interface:

```
telnet 192.168.2.1 port 23 src-interface ethernet0/0
```

You can close the Telnet connection by issuing **exit** or **clear socket** commands or by using the **ctrl + D** shortcut. Use the **set/unset telnet client enable** command to enable or disable the Telnet client feature on a security device.

To minimize an unauthorized user's chances of logging into a device, you can limit the number of unsuccessful login attempts allowed before the security device terminates a Telnet session. This restriction also protects against certain types of attacks, such as automated dictionary attacks.

By default, the device allows up to three unsuccessful login attempts before it closes the Telnet session. To change this number, enter the following command:

set admin access attempts *number*

NOTE: You must use the CLI to set this restriction.

Securing Telnet Connections

You can secure Telnet traffic by completely separating it from network user traffic. Depending upon your security device model, you can run all administrative traffic through the MGT interface or devote an interface such as the DMZ entirely to administrative traffic.

In addition, to ensure that admin users use a secure connection when they manage a security device through Telnet, you can require such users to Telnet only through a virtual private network (VPN) tunnel. After you have set this restriction, the device denies access if anyone tries to Telnet without going through a VPN tunnel.

NOTE: For information about VPN tunnels, see *"Virtual Private Networks" on page 705*.

To restrict Telnet access through a VPN:

set admin telnet access tunnel

NOTE: You must use the CLI to set this restriction.

Secure Shell

The built-in Secure Shell (SSH) server on a Juniper Networks security device provides a means by which administrators can remotely and securely manage the device in using applications that are SSH-aware. SSH allows you to open a remote command shell securely and execute commands. SSH provides protection from IP or DNS spoofing attacks and password or data interception.

You can choose to run either an SSH version 1 (SSHv1) or an SSH version 2 (SSHv2) server on the device. SSHv2 is considered more secure than SSHv1 and is currently being developed as the IETF standard. However, SSHv1 has been widely deployed and is commonly used. Note that SSHv1 and SSHv2 are not compatible. That is, you cannot use an SSHv1 client to connect to an SSHv2 server on the security device and you cannot use an SSHv2 client to connect to an SSHv1 server on the security device.

The client console or terminal application must run the same SSH version as the server. Figure 88 on page 322 illustrates SSH traffic flow.

Figure 88: SSH Traffic Flow

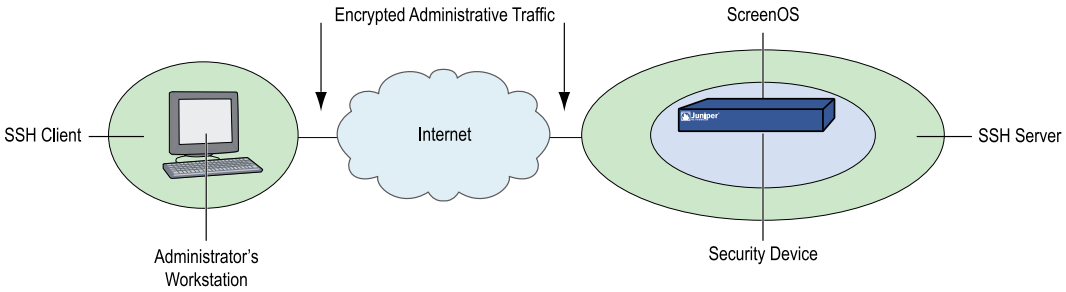
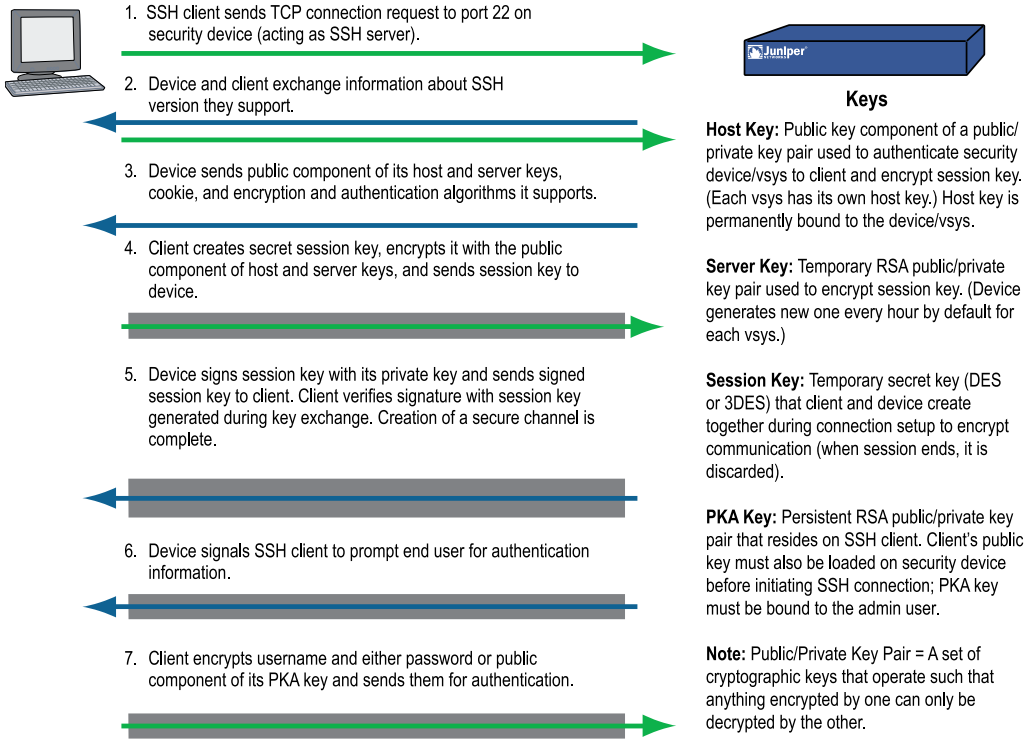


Figure 89 on page 322 illustrates the basic SSH connection procedure.

Figure 89: SSH Connection



A maximum of five SSH sessions is allowed on a Juniper Networks security device at any one time.

Client Requirements

As described in "Secure Shell" on page 321, the client application must run the same SSH version as the server on the security device. SSHv2 clients must be configured

to request the Diffie-Hellman (DH) key exchange algorithm and Digital Signature Algorithm (DSA) for PKA. SSHv1 clients must be configured to request the Rivest-Shamir-Adleman (RSA) for PKA.

Basic SSH Configuration on the Device

The following are the basic steps for configuring SSH on a Juniper Networks security device:

1. Determine whether you will use password authentication or PKA for SSH. If you are using PKA, the PKA certificates must be bound to an admin before SSH connections can be made. See "Authentication" on page 324 for more information about using passwords or PKA.
2. Determine which version of SSH you need to enable on the security device. By default, SSHv2 is enabled.(Remember that the client application and the SSH server on the device must run the same SSH version.) If you enabled SSH on the device in a previous ScreenOS version, SSHv2 runs when you enable SSH now. To see which version of SSH is active but not enabled on the device, enter the **get ssh** command:

   ```
   device> get ssh
   SSH V2 is active
   SSH is not enabled
   SSH is not ready for connections
   Maximum sessions: 8
   Active sessions: 0
   ```

   In the output shown above, SSHv2 is active and runs when you enable SSH. If you want to use a different SSH version, make sure that all keys created with the previous version are removed. For example, to clear SSHv2 keys and to use SSHv1:

   ```
   device> delete ssh device all
   ```

   The following messages appear:

   ```
   SSH disabled for vsys: 1
   PKA key deleted from device: 0
   Host keys deleted from device: 1
   Execute the 'set ssh version v2' command to activate SSH v1 for the device
   ```

   To use SSHv1

   device-> **set ssh version v1**

NOTE: Setting the SSH version does not enable SSH on the security device.

3. If you do not want to use port 22 (the default) for SSH client connections, you can specify a port number between 1024 and 32767.

NOTE: You can also use the WebUI to change the port number and enable SSHv2 and SCP on the Configuration > Admin > Management page.

```
device-> set admin ssh port 1024
```

4. Enable SSH for the root system or for the virtual system. See "SSH and Vsys" on page 326 for additional information about enabling and using SSH for a vsys.

   To enable SSH for the root system:

   ```
   device-> set ssh enable
   ```

   To enable SSH for a vsys, you need to first enter the vsys and then enable SSH:

   ```
   device-> set vsys v1
   device(v1)-> set ssh enable
   ```

5. Enable SSH on the interface on which the SSH client will connect.

   ```
   device-> set interface manage ssh
   ```

6. Distribute the host certificate to the SSH client. See "Host Certificate" on page 328 for more information.

Authentication

An administrator can connect to a Juniper Networks security device with SSH using one of two authentication methods:

- **Password Authentication:** This method is used by administrators who need to configure or monitor a security device. The SSH client initiates an SSH connection to the device. If SSH manageability is enabled on the interface receiving the connection request, the device signals the SSH client to prompt the user for a username and password. When the SSH client has this information, it sends it to the device, which compares it with the username and password in the admin user's account. If they match, the device authenticates the user. If they do not match, the device rejects the connection request.
- **Public Key Authentication with key (PKA key):** This method provides increased security over the password authentication method and allows you to run automated scripts. Instead of a username and password, the SSH client sends a username and the public key component of a public/private key pair. The device compares it with up to four public keys that can be bound to an admin. If one of the keys matches, the device authenticates the user. If none of them matches, the device rejects the connection request.
- **Public Key Authentication with certificate (PKA certificate):** This method is very similar to PKA with key, but instead of using raw keys, PKI certificates are used. The public key is embedded in the PKI certificate. This method requires that the certificate be loaded to the security device and then bound to an administrators account. This method provides increased security over PKA with keys in that the certificate contains the information identify the owner of the

certificate and the identity of the CA that has verified the identity. This authentication method is supported by SSHv2 only.

NOTE: The supported authentication algorithms are RSA for SSHv1 and DSA for SSHv2.

Both authentication methods require the establishment of a secure connection before the SSH client logs in. After an SSH client has established an SSH connection with the device, he must authenticate himself either with a username and password or with a PKA certificate.

Both password authentication and PKA require that you create an account for the admin user on the device and enable SSH manageability on the interface through which you intend to manage the device with an SSH connection. (For information about creating an admin user account, see "Defining Admin Users" on page 348.) The password authentication method does not require any further set up on the SSH client.

Binding a PKA key to administrator

To prepare for PKA, you must first perform the following tasks:

1. On the SSH client, generate a public and private key pair using a key generation program. (The key pair is either RSA for SSHv1 or DSA for SSHv2. See the SSH client application documentation for more information.)

NOTE: If you want to use PKA for automated logins, you must also load an agent on the SSH client to decrypt the private key component of the PKA public/private key pair and hold the decrypted version of the private key in memory.

2. Move the public key from the local SSH directory to a directory on your TFTP server, and launch the TFTP program.
3. To load the public key from the TFTP server to the device, enter one of the following CLI commands:

   For SSHv1:

   exec ssh tftp pka-rsa [**username** *name*] **file-name** *name_str* **ip-addr** *tftp_ip_addr*

   For SSHv2:

   exec ssh tftp pka-dsa [**user-name** *name*] **file-name** *name_str* **ip-addr** *tftp_ip_addr*

4. Bind the PKA key, a public key to the administrative account of the administrator that who processes the associated private key. The following CLI commands can be used to bind the PKA key to an administrators account:

   set ssh pka-dsa key *pka-key*
   set ssh pka-dsa user-name *login-id* key *pka-key*

The **user-name** option is only available to the root admin, so that only the root admin can bind to another admin. When you—as the root admin or as a read/write admin—enter the command without a username, the device binds the PKA certificate to your own admin account; that is, it binds the certificate to the admin who enters the command.

NOTE: The security device supports up to four PKA public keys per admin user.

When an administrator attempts to log in via SSH on an interface that has SSH manageability enabled, the device first checks if a public key is bound to that administrator. If so, the device authenticates the administrator using PKA. If a public key is not bound to the administrator, the device prompts for a username and password. (You can use the following command to force an admin to use only the PKA method: **set admin ssh password disable username** *name_str*.) Regardless of the authentication method you intend the administrator to use, when you initially define his or her account, you still must include a password, even though when you later bind a public key to this user, the password becomes irrelevant.

Binding a PKA certificate to administrator

1. Using the SSH client or a key generation utility generate a public key pair.
2. Using certificate request utility generate a certificate request with the public key pair embedded.
3. Submit the certificate request to a CA to generate a certificate.
4. Store the PKA certificate in the security devices PKI DB. The PKI system will assign a unique key-id to the certificate.
5. To bind the PKA certificate to an administrative account use the following commands:

```
set ssh pka-dsa cert cert-id
set ssh pka-dsa user-name login-id cert-id cert-id
```

The user-name option is only available to the ROOT privileged admin, since only the ROOT privileged admin is permitted to bind PKA certificates to the accounts belonging to other administrators will a non-ROOT privileged level. When no *login-id* is specified, the PKA certificate is bound to the account of the administrator executing the command.

NOTE: The security device supports up to four PKA public certificates per administrator

SSH and Vsys

For security devices that support vsys, you can enable and configure SSH for each vsys. SSH uses Host Keys or Host Certificates to provide a means to identify a server

side entity to the SSH client. Each device has its own host key (see "Host Key" on page 327) and maintains and manages a PKA key for the admin of the system.

In the case of Host Certificates (see "Host Certificate" on page 328), a single Host Certificate is used to identity the root system and all VSYS. This single Host Certificate must first be created and explicitly bound to the security devices SSH server/device.

The maximum number of SSH sessions is a device-wide limit and is between 2 and 24, depending upon the device. If the maximum number of SSH clients are already logged into the device, no other SSH client can log into the SSH server. The root system and the vsys share the same SSH port number. This means that if you change the SSH port from the default port 22, the port is changed for all vsys as well.

NOTE: When you deploy a large number of virtual systems on a single device, be aware that if many or all vsys admins use SSH, the storage reserved for PKI objects can fill up.

Host Key

The host key allows the security device to identify itself to an SSH client. On devices that support virtual systems (vsys), each vsys has its own host key. When SSH is first enabled on a vsys (for devices that support vsys) or on a device, a host key is generated that is unique to the vsys or device. The host key is permanently bound to the vsys or device and the same host key is used if SSH is disabled and then enabled again.

The host key on the device must be distributed to the SSH client in one of two ways:

- Manually—the root or vsys admin sends the host key to the client admin user with email, telephone, and so on. The receiving admin stores the host key in the appropriate SSH file on the SSH client system. (The SSH client application determines the file location and format.)
- Automatically—When the SSH client connects to the device, the SSH server sends the unencrypted public component of the host key to the client. The SSH client searches its local host key database to see if the received host key is mapped to the address of the device. If the host key is unknown (there is no mapping to the device address in the client's host key database), the Admin user might be able to decide whether to accept the host key. Otherwise, the connection is terminated. (See the appropriate SSH client documentation for information on accepting unknown host keys.)

To verify that the SSH client has received the correct host key, the Admin user on the client system can generate the SHA hash of the received host key. The client Admin user can then compare this SHA hash with the SHA hash of the host key on the device. On the device, you can display the SHA hash of the host key by executing the CLI command **get ssh host-key**.

Host Certificate

The host certificate allows the device to identify itself on the SSH client application. The SSH client application loads a local certificate to the device with the subject name **ssh-cert-dsa** along with the related CA certificate and CRL. The security device assigns a certificate ID to each host certificate and stores it in the PKI database.

Before binding the host certificate to the device, perform the following steps:

1. Disable SSH for all vsys and delete all host keys:

   ```
   delete ssh device all
   ```

2. Set SSH version to the active version:

   ```
   set ssh version 2
   ```

Once the certificate is loaded, the root-admin can force the device to use the host certificate by using the CLI command;

```
set ssh host-identity cert-dsa cert-id
```

where cert-id denotes the ID of the host certificate.

When the SSH client application connects to the device, the device sends the host certificate to the SSH client application. The application performs assured certification of the device based on the host certificate.

To unbind the host certificate from the device use the **delete ssh device all** command.

Binding a Host certificate to a device is not permitted in the following situations:

- No version of SSH is active
- SSH version1 is active
- SSH is enabled for any vsys
- The host key is still bound to any VSYS.
- The certificate-ID does not exist.
- The host certificate has an invalid subject name.

Example: SSHv1 with PKA for Automated Logins

In this example, you (as the root admin) set up SSHv1 public key authentication (PKA) for a remote host that runs an automated script. The sole purpose for this remote host to access the device is to download the configuration file every night. Because authentication is automated, no human intervention is necessary when the SSH client logs into the device.

You define an admin user account named **cfg**, with password **cfg** and read-write privileges. You enable SSH manageability on interface ethernet1, which is bound to the Untrust zone.

You have previously used a key generation program on your SSH client to generate an RSA public/private key pair, moved the public key file, which has the filename "idnt_cfg.pub", to a directory on your TFTP server, and launched the TFTP program. The IP address of the TFTP server is 10.1.1.5.

WebUI

Configuration > Admin > Administrators > New: Enter the following, then click OK:

```
Name: cfg
New Password: cfg
Confirm Password: cfg
Privileges: Read-Write (select)
SSH Password Authentication: (select)
```

Network > Interfaces > Edit (for ethernet1): Select **SSH** in Service Options, then click OK.

NOTE: You can only load a public key file for SSH from a TFTP server with the **exec ssh** command.

CLI

```
set admin user cfg password cfg privilege all
set interface ethernet1 manage ssh
exec ssh tftp pka-rsa username cfg file-name idnt_cfg.pub ip-addr 10.1.1.5
save
```

Secure Copy

Secure Copy (SCP) provides a way for a remote client to transfer files to or from the security device using the SSH protocol. (The SSH protocol provides authentication, encryption, and data integrity to the SCP connection.) The device acts as an SCP server to accept connections from SCP clients on remote hosts.

SCP requires that the remote client be authenticated before file transfer commences. SCP authentication is exactly the same process used to authenticate SSH clients. The SCP client can be authenticated with either a password or a PKA key. Once the SCP client is authenticated, one or more files can be transferred to or from the device. The SCP client application determines the exact method for specifying the source and destination filenames; see the SCP client application documentation.

SCP is disabled by default on the device. To enable SCP, you must also enable SSH.

WebUI

Configuration > Admin > Management: Select the following, then click **Apply**:

Enable SSH: (select)
Enable SCP: (select)

CLI

```
set ssh enable
set scp enable
save
```

The following is an example of an SCP client command to copy the configuration file from flash memory on a device (administrator name is "juniper" and the IP address is 10.1.1.1) to the file "ns_sys_config_backup" on the client system:

```
scp juniper@10.1.1.1:ns_sys_config ns_sys_config_backup
```

You can also copy a ScreenOS image to and from a device. To save an image named "ns.5.1.0r1" to a device from an SCP client, enter the following SCP client command, in which the administrator's login name is "juniper" and the IP address of the device is 10.1.1.1:

```
scp ns.5.1.0r1 juniper@10.1.1.1:image
```

Then enter the **reset** command to reboot the security device to load and run the new ScreenOS image.

To copy a ScreenOS image from a device to an SCP client and name the saved image "current_image_backup," enter the following SCP client command:

```
scp juniper@10.1.1.1:image current_image_backup
```

You need to consult your SCP client application documentation for information on how to specify the administrator name, device IP address, source file, and destination file.

Serial Console

You can manage a security device through a direct serial connection from the administrator's workstation to the device with the console port. Although a direct connection is not always possible, this is the most secure method for managing the device provided that the location around the device is secure.

NOTE: To prevent unauthorized users from logging in remotely as the root admin, you can require the root admin to log into the device through the console only. For additional information on this restriction, see "Restricting the Root Admin to Console Access" on page 356.

Depending on your Juniper Networks security device model, creating a serial connection requires one of the following cables:

- A female DB-9 to male DB-25 straight-through serial cable
- A female DB-9 to male DB-9 straight-through serial cable
- A female DB-9 to male MiniDIN-8 serial cable
- A female DB-9 to RJ-45 adapter with an RJ-45 to RJ-45 straight-through Ethernet cable

You will also need HyperTerminal software (or another kind of VT100 terminal emulator) on the management workstation, with the HyperTerminal port settings configured as follows:

- Serial communications 9600 bps
- 8 bit
- No parity
- 1 stop bit
- No flow control

NOTE: For more details on using HyperTerminal, see the *ScreenOS CLI Reference Guide: IPv4 Command Descriptions* or the documentation for your device.

Remote Console

You can remotely access the console interface on a security device by dialing into it. There are two ways of dialing into the console:

- "Remote Console Using V.92 Modem Port" on page 331
- "Remote Console Using an AUX Port" on page 332

Remote Console Using V.92 Modem Port

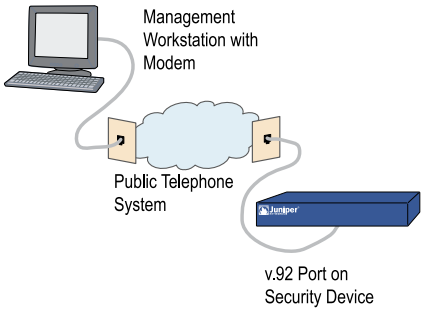
You can remotely manage the security devices that are equipped with v.92 modem ports by dialing into the port and accessing the console interface. To use remote console, you connect the v.92 modem port on the device to a telephone line, and dial into the device using a remote computer with a modem. You use a terminal program such as HyperTerminal to establish the console session.

In order to use remote console connection, you must first enable remote management with the following CLI command:

```
set interface serialx/0 modem auxenable
save
```

Figure 90 on page 332 shows how to connect the device for remote console management.

Figure 90: Remote Console Management Connection



You will also need HyperTerminal software (or another kind of VT100 terminal emulator) on the management workstation, with the HyperTerminal port settings configured as follows:

- Serial communications 9600 bps
- 8 bit
- No parity
- 1 stop bit
- No flow control

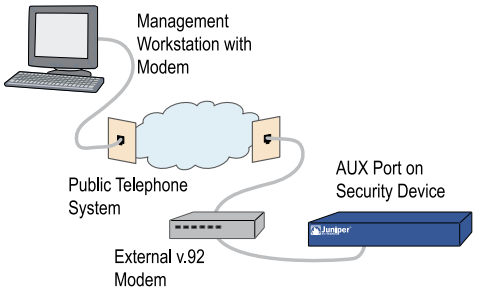
NOTE: For more details on using HyperTerminal, see the *ScreenOS CLI Reference Guide: IPv4 Command Descriptions* or the documentation for your device.

Remote Console Using an AUX Port

You can remotely manage the security devices that are equipped with AUX ports by dialing into a modem connected to the port and accessing the console interface. To use remote console, you connect the AUX modem port on the device to a telephone line using an external modem, and dial into the device using a remote computer with a modem. You use a terminal program such as HyperTerminal to establish the console session.

Figure 91 on page 333 shows how to connect the device for remote console management.

Figure 91: Remote Console Management Connection



You will also need HyperTerminal software (or another kind of VT100 terminal emulator) on the management workstation, with the HyperTerminal port settings configured as follows:

- Serial communications 9600 bps
- 8 bit
- No parity
- 1 stop bit
- No flow control

NOTE: For more details on using HyperTerminal, see the *ScreenOS CLI Reference Guide: IPv4 Command Descriptions* or the documentation for your device.

Modem Port

You can also manage a security device by connecting the administrator's workstation to the modem port on the device. The modem port functions similarly to the console port, except that you cannot define parameters for the modem port or use this connection to upload an image.

To prevent unauthorized users from managing the device through a direct connection to the console or modem port, you can disable both ports by entering the following commands:

```
set console disable
set console aux disable
```

Management with the Network and Security Manager

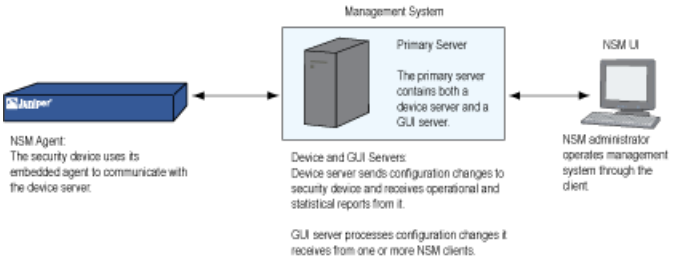
Network and Security Manager (NSM) is Juniper Networks' enterprise-level management software application that configures and monitors multiple Juniper Networks security devices over a local area network (LAN) or a wide area network (WAN) environment. The NSM user interface (UI) enables network administrators to deploy, configure, and manage multiple devices from central locations.

NSM uses three components to enable remote communication with security devices:

- The *NSM UI* is a java-based software application that you use to access and configure data on your network with the NSM management system. From the UI, you can view, configure, and manage your network.
- The *management system* is a set of services that resides on an external host. These services process, track, and store device management information exchanged between a device and the NSM UI. The management system is composed of two components:
  - The *GUI Server* receives and responds to requests and commands from the UI. It manages the system resources and configuration data required to manage your network. It also contains a local data store of information about your managed security devices, administrators, and configurations.
  - The *Device Server* acts as a collection point for all data generated by each of your network devices. It stores this data, primarily traffic logs, in the local data store.
- *NSM* Agent is a service that resides on each managed security device. NSM Agent receives configuration parameters from the external management system and forwards them to ScreenOS. NSM Agent also monitors each device and transmits reports back to the management system. NSM Agent can download signature packs, certificates, and entitlements between a security device and NSM.

Figure 92 on page 334 shows how NSM Agent communicates with the NSM UI.

Figure 92: Security Device with NSM Agent Enabled



For more information about these and other NSM components, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

Initiating Connectivity Between NSM Agent and the MGT System

Before NSM can access and manage a security device, it is necessary to initiate communications between NSM Agent (which resides on the device) and the management system (which resides on an external host). Initialization might require up to two users at two different sites, depending upon the current availability of the security device. These users might include the NSM administrator, who uses the NSM UI on a client host, and the on-site user, who executes CLI commands on a device with a console session. Possible initialization cases include the following:

- *Case 1:* A device already has a known IP address and is reachable over your network infrastructure.

  In this case, the NSM administrator adds the device using the NSM UI on the client host. (No on-site user is necessary.) The device automatically connects back to the management system and is ready to send configuration information to the NSM database that resides there.

- *Case 2:* The IP address is unreachable.

  In this case, both users perform initialization tasks. The administrator adds the device through the NSM UI. The administrator also determines which CLI commands the on-site user needs and delivers them to the user, who then executes them through the console. The device then automatically connects with the management system and is ready to send configuration information to the NSM database.

- *Case 3:* The device is a new appliance and contains the factory default settings.

  In this case, both users perform initialization tasks. The on-site user can use an encrypted configuration script, called a *configlet*, which the NSM administrator generates. The process is as follows:

  1. The administrator selects the device platform and ScreenOS version, using the Add Device wizard in the NSM UI.
  2. The administrator edits the device and enters the desired configuration.
  3. The administrator activates the device.
  4. The administrator generates and delivers the Configlet file (or the necessary CLI commands, as with Case 2) to the on-site user.
  5. The on-site user executes Configlet (or the CLI commands).

For more information, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

Enabling, Disabling, and Unsetting NSM Agent

Before a security device can communicate with the management system, you must enable Network and Security Manager (NSM) Agent residing on the device.

If you want to unset NSM, use the **unset nsmgmt all** command. This command sets NSM Agent to its initial defaults, so it acts as though it was never connected to NSM. Use the **unset nsmgmt all** command when you want to reconfigure the NSM settings.

To enable NSM Agent on the security device, do either of the following:

WebUI

Configuration > Admin > NSM: Select **Enable Communication with Network and Security Manager (NSM)**, then click **Apply**.

CLI

```
set nsmgt enable
save
```

To disable NSM Agent on the device, do either of the following:

WebUI

Configuration > Admin > NSM: Clear **Enable Communication with Network and Security Manager (NSM)**, then click **Apply**.

CLI

```
unset nsmgt enable
save
```

Setting the Primary Server IP Address of the Management System

The IP address by which NSM Agent identifies the external management system servers is a configurable parameter.

In the following example you set the primary server IP address to 1.1.1.100.

WebUI

Configuration > Admin > NSM: Enter the following, then click **Apply**:

```
Primary IP Address/Name: 1.1.1.100
```

CLI

```
set nsmgmt server primary 1.1.1.100
save
```

Setting Alarm and Statistics Reporting

NSM Agent monitors the device events and transmits reports back to the management system. This allows the NSM administrator to view the events from the NSM UI.

The categories of events tracked by NSM Agent are as follows:

- *Alarms* report potentially dangerous attacks or traffic anomalies, including attacks detected through deep inspection.
- *Log events* report changes in a device's configuration and non-severe changes that occur on a device.
- *Protocol distribution* events report messages generated by the following protocols:

  - Authentication Header (AH)
  - Encapsulating Security Payload (ESP)
  - Generic Routing Encapsulation (GRE)
  - Internet Control Message Protocol (ICMP)
  - Open Shortest Path First (OSPF)
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)
- *Statistics* messages report the following statistical information:
  - Attack statistics
  - Ethernet statistics
  - Traffic flow statistics
  - Policy statistics

In the following example, you enable transmission of all Alarm and Statistics messages to the Management System.

WebUI

Configuration > Admin > NSM: Enter the following, then click **Apply**:

```
Attack Statistics: (select)
Policy Statistics: (select)
Attack Alarms: (select)
Traffic Alarms: (select)
Flow Statistics: (select)
Ethernet Statistics: (select)
Deep Inspection Alarms: (select)
Event Alarms: (select)
```

CLI

```
set nsmgmt report statistics attack enable
set nsmgmt report statistics policy enable
set nsmgmt report alarm attack enable
set nsmgmt report alarm traffic enable
set nsmgmt report statistics flow enable
set nsmgmt report statistics ethernet enable
set nsmgmt report alarm idp enable
set nsmgmt report alarm other enable
save
```

Configuration Synchronization

If the ScreenOS configuration is changed from the last time it was synchronized with NSM, then the security device notifies the NSM administrator of the change. For example, the device sends a message when a device administrator uses console, telnet, SSH, or the WebUI to change a security device configuration. Changing the configuration with any application other than NSM causes it to be unsynchronized. The NSM configuration file must be synchronized with the security device configuration file for NSM to work correctly. The synchronization is achieved when you import the configuration file to NSM. For information about importing devices, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager,

The following example displays the command used to view the configuration status.

Example: Viewing the Configuration State

In the following example, you view the configuration synchronization state of a security device.

WebUI

NOTE: You must use the CLI to retrieve the running configuration state.

CLI

```
get config nsmgmt-dirty
```

NOTE: If applications other than NSM applications have not changed the configuration file, then the command returns a blank; otherwise, it returns a “yes.”

Example: Retrieving the Configuration Hash

NSM uses the configuration hash to verify the configuration synchronization of a security device. In the following example, you retrieve the running configuration hash for a specific virtual system.

WebUI

NOTE: You must use the CLI to retrieve the running configuration hash.

CLI

```
device-> enter vsys vsys1
```

```
device(vsys1)-> get config hash
a26a16cd6b8ef40dc79d5b2ec9e1ab4f
device(vsys1)->
device(vsys1)-> exit
```

Retrieving the Configuration Timestamp

A security device provides two configuration timestamps—running-config and saved-config. The running-config timestamp is when the set or unset command was last executed for each virtual system. The saved-config timestamp is when the device configuration was last saved.

In the following example, the security device retrieves the last running and saved configuration timestamps for the vsys1 virtual system:

WebUI

NOTE: You must use the CLI to retrieve the running and saved configuration timestamps.

CLI

```
get config timestamp vsys vsys1
get config saved timestamp
```

NOTE: If you omit **vsys vsys_name** from the command, the security device retrieves the configuration timestamp for the root system. If the timestamp is unavailable, then an “unknown” message is displayed.

Controlling Administrative Traffic

ScreenOS provides the following options for configuring and managing the security device:

- **WebUI:** Selecting this option allows the interface to receive HTTP traffic for management with the Web user interface (WebUI).
- **Telnet:** A terminal emulation program for TCP/IP networks such as the Internet, Telnet is a common way to remotely control network devices. Selecting this option enables Telnet manageability.
- **SSH:** You can administer the security device from an Ethernet connection or a dial-in modem using SSH. You must have an SSH client that is compatible with Version 1.5 of the SSH protocol. These clients are available for Windows 95 and later, Windows NT, Linux, and UNIX. The security device communicates with the SSH client through its built-in SSH server, which provides device configuration and management services. Selecting this option enables SSH manageability.

- **SNMP:** The security device supports both SNMPv1 and SNMPv2c, and all relevant Management Information Base II (MIB II) groups, as defined in RFC1213. Selecting this option enables SNMP manageability.
- **SSL:** Selecting this option allows the interface to receive HTTPS traffic for secure management of the security device with the WebUI.
- **Network and Security Manager:** Selecting this option allows the interface to receive NSM traffic.
- **Ping:** Selecting this option allows the security device to respond to an ICMP echo request, or ping, which determines whether a specific IP address is accessible over the network.
- **Ident-Reset:** Services like Mail and FTP send identification requests. If they receive no acknowledgement, they send the request again. While the request is processing, there is no user access. By enabling the Ident-reset option, the security device sends a TCP reset announcement in response to an IDENT request to port 113 and restores access that has been blocked by an unacknowledged identification request.

To use these options, you enable them on one or more interfaces, depending on your security and administrative needs.

MGT and VLAN1 Interfaces

Some Juniper Networks security devices have a physical interface—Management (MGT)—dedicated exclusively for management traffic. You can use this interface for management traffic when interfaces are in NAT, route, or transparent mode.

In transparent mode, you can configure all security devices to allow administration through the logical interface, VLAN1. To enable management traffic to reach the VLAN1 interface, you must enable the management options you want both on VLAN1 and on the Layer 2 zones—V1-Trust, V1-Untrust, V1-DMZ, user-defined Layer 2 zone—through which the management traffic passes to reach VLAN1.

In transparent mode, the VLAN1 logical interface supports the DHCP client with AUTOCONFIG feature. This feature is supported on SSG 5 and SSG 20 platforms.

The process of VLAN1 interface working as DHCP client to do AUTOCONFIG is as follows:

1. The environment variable is set by the set envar dhcp autocfg=yescommand. The DHCP client and AUTOCONFIG will be set on the UNTRUST interface.
2. The config file on TFTP server is loaded and saved to the flash. The config file turns the device into pure transparent mode and configure VLAN1 as DHCP client and set AUTOCONFIG on.
3. The VLAN1 interface again loads the config file as the DHCP client is set for the first time.

NOTE: The repeated config file loading will not impact the transparent mode.

4. If the SSG 5 or SSG 20 resets, at first, the device loads the configuration in the Flash. The VLAN1 then works as DHCP client to get the IP address of the device and loads the config file again. The config file is loaded from the TFTP server and saved again in the flash. The device is set to transparent mode.

NOTE: When the DHCP client with the AUTOCONFIG feature is enabled, you can access the device using WEB, Telnet, SSH, and Ping until the DHCP is done successfully and the you know the IP address that is allocated to the device.

To maintain the highest level of security, Juniper Networks recommends that you limit administrative traffic exclusively to the VLAN1 or MGT interface and user traffic to the security zone interfaces. Separating administrative traffic from network user traffic greatly increases administrative security and ensures constant management bandwidth.

Example: Administration Through the MGT Interface

In this example, you set the IP address of the MGT interface to 10.1.1.2/24 and enable the MGT interface to receive Web and SSH administrative traffic.

WebUI

Network > Interfaces > Edit (for mgt): Enter the following, then click **OK**:

IP Address/Netmask: 10.1.1.2/24
Management Services: WebUI, SSH: (select)

CLI

```
set interface mgt ip 10.1.1.2/24
set interface mgt manage web
set interface mgt manage ssh
save
```

Example: Administration Through the VLAN1 Interface

In this example, you set the IP address of the VLAN1 interface to 10.1.1.1/24 and enable the VLAN1 interface to receive Telnet and Web administrative traffic through the V1-Trust zone.

WebUI

Network > Interfaces > Edit (for VLAN1): Enter the following, then click **OK**:

IP Address/Netmask: 10.1.1.1/24
Management Services: WebUI, Telnet: (select)

Network > Zones > Edit (for V1-Trust): Select the following, then click **OK**:

Management Services: WebUI, Telnet: (select)

CLI

```
set interface vlan1 ip 10.1.1.1/24
set interface vlan1 manage web
set interface vlan1 manage telnet
set interface vlan1 dhcp client
set interface vlan1 dhcp client settings autoconfig
set zone v1-trust manage web
set zone v1-trust manage telnet
save
```

Setting Administrative Interface Options

On security devices that have multiple physical interfaces for network traffic, but no physical MGT interface, you might dedicate one physical interface exclusively for administration, separating management traffic completely from network user traffic. For example, you might have local management access the device through an interface bound to the Trust zone and remote management through an interface bound to the Untrust zone.

In this example, you bind ethernet1 to the Trust zone and ethernet3 to the Untrust zone. You assign ethernet1 the IP address 10.1.1.1/24 and give it the Manage IP address 10.1.1.2. (Note that the Manage IP address must be in the same subnet as the security zone interface IP address.) You also allow ethernet1 to receive Web and Telnet traffic. You then assign ethernet3 the IP address 1.1.1.1/24 and block all administrative traffic to that interface.

WebUI

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24
Manage IP: 10.1.1.2
Management Services:
WebUI: (select)
SNMP: (clear)
Telnet: (select)
SSL: (clear)
SSH: (clear)

Enter the following, then click **OK**:

Interface Mode: NAT
Network > Interfaces > Edit (for ethernet3):

Enter the following, then click **OK**:

Zone Name: Untrust

Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24
Management Services:
 WebUI: (clear)
 SNMP: (clear)
 Telnet: (clear)
SSL: (clear)
 SSH: (clear)

CLI

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 manage-ip 10.1.1.2
set interface ethernet1 manage web
unset interface ethernet1 manage snmp
set interface ethernet1 manage telnet
unset interface ethernet1 manage ssl
unset interface ethernet1 manage ssh
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
save
```

NOTE: When you bind an interface to any security zone other than the Trust and V1-Trust zones, all management options are disabled by default. Therefore, in this example, you do not have to disable the management options on ethernet3.

Setting Manage IPs for Multiple Interfaces

Any physical, redundant, or aggregate interface or sub-interface you bind to a security zone can have at least two IP addresses:

- An interface IP address, which connects to a network
- A logical Manage IP address for receiving administrative traffic

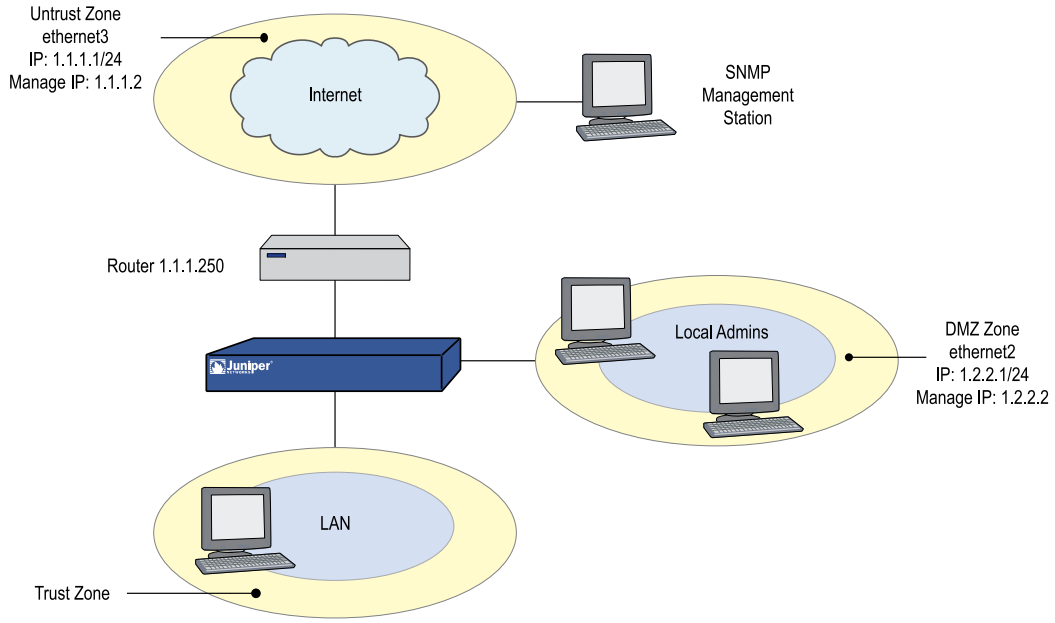
When a security device is a backup unit in a redundant group for high availability (HA), you can access and configure the unit through its Manage IP address (or addresses)

NOTE: The Manage IP address differs from the VLAN1 address in the following two ways:
When the security device is in transparent mode, the VLAN1 IP address can be the endpoint of a VPN tunnel, but the Manage IP address cannot.
You can define multiple Manage IP addresses—one for each network interface—but you can only define one VLAN1 IP address—for the entire system.

If you select the Manageable option on the interface configuration page in the WebUI, you can manage the security device either through the interface IP address or the Manage IP address associated with that interface.

Figure 93 on page 344 illustrates this example in which you bind ethernet2 to the DMZ zone and ethernet3 to the Untrust zone. You set the management options on each interface to provide access for the specific kinds of administrative traffic. You allow HTTP and Telnet access on ethernet2 for a group of local administrators in the DMZ zone, and SNMP access on ethernet3 for central device monitoring from a remote site. Ethernet2 and ethernet3 each have a Manage IP address, to which the administrative traffic is directed. You also set a route directing self-generated SNMP traffic out ethernet3 to the external router at 1.1.1.250.

Figure 93: Setting Management IPs for Multiple Interfaces



WebUI

Network > Interfaces > Edit (ethernet2): Enter the following, then click **OK**:

Zone Name: DMZ
Static IP: (select this option when present)
IP Address/Netmask: 1.2.2.1/24
Manage IP: 1.2.2.2
Management Services:
WebUI: (select)
Telnet: (select)

Network > Interfaces > Edit (ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)

```
IP Address/Netmask: 1.1.1.1/24
Manage IP: 1.1.1.2
Management Services:
 SNMP: (select)
```

CLI

```
set interface ethernet2 zone dmz
set interface ethernet2 ip 1.2.2.1/24
set interface ethernet2 manage-ip 1.2.2.2
set interface ethernet2 manage web
set interface ethernet2 manage telnet
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 manage-ip 1.1.1.2
set interface ethernet3 manage snmp
save
```

Levels of Administration

Juniper Networks security devices support multiple administrative users. For any configuration changes made by an administrator, the security device logs the following information:

- The name of the administrator making the change
- The IP address from which the change was made
- The time of the change

There are several levels of administrative user. The availability of some of these levels depends on the model of your Juniper Networks security device. The following sections list all the admin levels and the privileges for each level. These privileges are only accessible to an admin after he or she successfully logs in with a valid username and password.

Root Administrator

The root administrator has complete administrative privileges. There is only one root administrator per security device. The root administrator has the following privileges:

- Manages the root system of the security device
- Adds, removes, and manages all other administrators
- Assigns role attributes to all other administrators
- Establishes and manages virtual systems and assigns physical or logical interfaces to them
- Creates, removes, and manages virtual routers (VRs)
- Adds, removes, and manages security zones
- Assigns interfaces to security zones
- Performs asset recovery

- Sets the device to FIPS mode
- Resets the device to its default settings
- Updates the firmware
- Loads configuration files
- Clears all active sessions of a specified admin or of all active admins

Role Attributes

The Juniper Networks security device allows you—the root admin—to assign role attributes to nonroot read-write and read-only administrators. You cannot assign role attributes to root and vsys admins, however.

- You can assign one of the following role attributes to an admin user.
  - **Crypto**—Gives the user the ability to configure and monitor cryptographic data.
  - **Security**—Gives the user the ability to configure and monitor security data.
  - **Audit**—Gives the user the ability to configure and monitor audit data.
- You cannot assign two role attributes for the same admin user. However, you can change the role attribute for an admin user when the admin user is inactive.
- You can assign roles to admin users in local database. For admin users authenticated by external RADIUS or TACACS+ authentication servers, the role attribute is assigned in the remote server.

Table 41 on page 346 lists the privileges of admin users according to assigned role attribute.

Table 41: Privileges for Administrators According to Role Attribute

	Privileges					
Admin with Role	**Read Crypto Data**	**Write Crypto Data**	**Read Security Data**	**Write Security Data**	**Read Audit Data**	**Write Audit Data**
Read-write admin with crypto	x	x			x	
Read-only admin with crypto	x				x	
Read-write admin with security			x	x	x	
Read-only admin with security			x		x	
Read-write admin with audit					x	x

Table 41: Privileges for Administrators According to Role Attribute *(continued)*

Admin with Role	Privileges					
	Read Crypto Data	Write Crypto Data	Read Security Data	Write Security Data	Read Audit Data	Write Audit Data
Read-only admin with audit					x	

- Each role attribute has a scope. The security device keeps a check on the role of the admin user. An admin user with role attribute cannot make configuration changes outside the scope of the role attribute. For example, a cryptographic admin cannot access security data and a security admin cannot access cryptographic data.

NOTE: The security device does not check the role attribute when an administrator views the audit logs or executes self-tests. For information about self-tests, see "Federal Information Processing Standards (FIPS)" on page 309.

Read/Write Administrator

The read/write administrator has the same privileges as the root administrator, but cannot create, modify, or remove other admin users. The read/write administrator has the following privileges:

- Creates virtual systems and assigns a virtual system administrator for each one
- Monitors any virtual system
- Tracks statistics (a privilege that cannot be delegated to a virtual system administrator)

Read-Only Administrator

The read-only administrator has only viewing privileges using the WebUI, and can only issue the **get** and **ping** CLI commands. The read-only administrator has the following privileges:

- Read-only privileges in the root system, using the following four commands: **enter**, **exit**, **get**, and **ping**
- Read-only privileges in virtual systems

Virtual System Administrator

Some security devices support virtual systems. Each virtual system (vsys) is a unique security domain, which can be managed by virtual system administrators with privileges that apply only to that vsys. Virtual system administrators independently

manage virtual systems through the CLI or WebUI. On each vsys, the virtual system administrator has the following privileges:

- Creates and edits auth, IKE, L2TP, XAuth, and Manual Key users
- Creates and edits services
- Creates and edits policies
- Creates and edits addresses
- Creates and edits VPNs
- Modifies the virtual system administrator login password
- Creates and manages security zones
- Adds and removes virtual system read-only administrators

Virtual System Read-Only Administrator

A virtual system read-only administrator has the same set of privileges as a read-only administrator, but only within a specific virtual system. A virtual system read-only administrator has viewing privileges for his particular vsys through the WebUI, and can only issue the **enter**, **exit**, **get**, and **ping** CLI commands within his vsys.

NOTE: For more information on virtual systems, see *"Virtual Systems" on page 1677*.

Defining Admin Users

The root administrator is the only one who can create, modify, and remove admin users. In the following example, the one performing the procedure must be a root administrator.

Example: Adding a Read-Only Admin

In this example, you—as the root admin—add a read-only administrator named Roger with password 2bd21wG7.

WebUI

Configuration > Admin > Administrators > New: Enter the following, then click **OK**:

Name: Roger
New Password: 2bd21wG7
Confirm New Password: 2bd21wG7
Privileges: Read-Only (select)
Role: None (default)

NOTE: The password can be up to 31 characters long and is case sensitive.

CLI

```
set admin user Roger password 2bd21wG7 privilege read-only
save
```

Example: Modifying an Admin

In this example, you—as the root admin—change Roger's privileges from read-only to read/write and assign a security role attribute. Roger and the root admin share the same privilege in managing security-related configuration changes.

WebUI

Configuration > Admin > Administrators > Edit (for Roger): Enter the following, then click OK:

```
Name: Roger
New Password: 2bd21wG7
Confirm New Password: 2bd21wG7
Privileges: Read-Write (select)
Role: Security (select)
```

CLI

```
unset admin user Roger
set admin user Roger password 2bd21wG7 privilege all
set admin user Roger role security
save
```

Example: Deleting an Admin

In this example, you—as the root admin—delete the admin user Roger.

WebUI

Configuration > Admin > Administrators: Click **Remove** in the Configure column for Roger.

CLI

```
unset admin user Roger
save
```

Example: Configuring Admin Accounts for Dialup Connections

Some devices support a modem connection for outbound dialup disaster recovery situations. You can set up trustee accounts for the interface, for the modem or for both the interface and modem. This section describes the two types of trustees:

- Interface trustee

An interface trustee only has access to the WebUI and is restricted to the signaling methods and IP address assignment for the primary Untrust interface.

For devices with ADSL interfaces, an interface trustee has control over the following characteristics:

- Layer 1 characteristics: VPI/VCI, multiplexing mode, RFC1483 bridged or routed
- Layer 2 signaling methods (PPPoE or PPPoA, and their parameters)
- IP address assignment methods (statically defined by an administrator, or dynamically acquired from the circuit through PPPoE or PPPoA).

  For devices with only ethernet interfaces, an interface trustee can control how the interface IP address is assigned (statically defined by administrator, or dynamically acquired from the circuit with DHCP or PPPoE).

- Modem trustee

  A modem trustee only has access to the WebUI and is restricted to Modem and ISP settings for the serial interface. A modem trustee can created, modify, and delete modem definitions to suit their specific needs, and can create, modify, and delete the settings for ISP1 and ISP2. A modem trustee can view the configurations for ISP3 and ISP4, and can test connectivity for any defined ISP and phone number.

You can view all administrator accounts by entering the **get admin user** command, or you can view only the trustee accounts by entering the **get admin user trustee** command.

In the following example, you configure a Read/Write modem trustee account for Richard Brockie. You set his username to be sdonovan and his password to be !23fb.

WebUI

Configuration > Admin > Administrators

CLI

```
set admin user sdonovan password !23fb privilege all
set admin user sdonovan trustee modem
```

Example: Clearing an Admin's Sessions

In this example, you—as the root admin—terminate all active sessions of the admin user Roger. When you execute the following command, the security device closes all active sessions and automatically logs off Roger from the system.

WebUI

NOTE: You must use the CLI to clear an admin's sessions.

CLI

```
clear admin name Roger
save
```

Securing Administrative Traffic

To secure the security device during setup, perform the following steps:

1. On the WebUI, change the administrative port.

   See "Changing the Port Number" on page 352.

2. Change the username and password for administration access.

   See "Changing the Admin Login Name and Password" on page 352

3. Define the management client IP addresses for the admin users.

   See "Restricting Administrative Access" on page 355.

4. Turn off any unnecessary interface management service options.

   See "Controlling Administrative Traffic" on page 339.

5. Disable the ping and ident-reset service options on the interfaces, both of which respond to requests initiated by unknown parties and can reveal information about your network:

WebUI

Network > Interfaces > Edit (for the interface you want to edit): Disable the following service options, then click **OK**:

Ping: Selecting this option allows the security device to respond to an ICMP echo request, or "ping," which determines whether a specific IP address is accessible from the device.

Ident-Reset: When a service such as Mail or FTP sends an identification request and receives no acknowledgment, it sends the request again. While the request is in progress, user access is disabled. With the Ident-Reset check box enabled, the security device automatically restores user access.

CLI

```
unset interface interface manage ping
```

```
unset interface interface manage ident-reset
```

Changing the Port Number

Changing the port number to which the security device listens for HTTP management traffic improves security. The default setting is port 80, the standard port number for HTTP traffic. After you change the port number, you must then enter the new port number in the URL field in your browser when you next attempt to contact the security device. (In the following example, the administrator needs to enter http://188.30.12.2:15522.)

In this example, the IP address of the interface bound to the Trust zone is 10.1.1.1/24. To manage the security device with the WebUI on this interface, you must use HTTP. To increase the security of the HTTP connection, you change the HTTP port number from 80 (the default) to 15522.

WebUI

Configuration > Admin > Management: In the HTTP Port field, enter 15522, then click **Apply**.

CLI

```
set admin port 15522
save
```

Changing the Admin Login Name and Password

By default, the initial login name for security devices is **netscreen**. The initial password is also **netscreen**. Because these have been widely published, we recommend you change the login name and password immediately. The login name and password are both case-sensitive. They can contain any character that can be entered from the keyboard with the exception of ? and ". Record the new admin login name and password in a secure manner.

WARNING: Be sure to record your new password. If you forget it, you must reset the security device to its factory settings, and all your configurations will be lost. For more information, see “Resetting the Device to the Factory Default Settings” on page 354.

Admin users for the security device can be authenticated using the internal database or an external auth server. When the admin user logs into the security device, it first checks the local internal database for authentication. If there is no entry present and an external auth server is connected, it then checks for a matching entry in the external auth server database. After an admin user successfully logs into an external auth server, the security device maintains the admin’s login status locally.

NOTE: Juniper Networks supports RADIUS, SecurID, and LDAP servers for admin user authentication. (For more information, see "Admin Users" on page 1566.) Although the root admin account must be stored on the local database, you can store root-level read/write and root-level read-only admin users on an external auth server. To store root-level and vsys-level admin users on an external auth server and query their privileges, the server must be RADIUS and you must load the netscreen.dct file on it.
For more information about admin user levels, see "Levels of Administration" on page 345. For more about using external auth servers, see "External Authentication Servers" on page 1580.

When the root admin changes any attribute of an admin user's profile—username, password, or privilege—any administrative session that the admin currently has open automatically terminates. If the root admin changes any of these attributes for himself, or if a root-level read/write admin or vsys read/write admin changes his own password, all of that user's currently open admin sessions terminate, other than the one in which he made the change.

NOTE: The behavior of an HTTP or HTTPS session using the WebUI is different. Because HTTP does not support a persistent connection, any change that you make to your own user profile automatically logs you out of that and all other open sessions.

Example: Changing an Admin User's Login Name and Password

In this example, you—as the root admin—change a read/write administrator's login name from "John" to "Smith" and his password from xL7s62a1 to 3MAb99j2.

NOTE: Instead of using actual words for passwords, which might be guessed or discovered through a dictionary attack, you can use an apparently random string of letters and numbers. To create such a string that you can easily remember, compose a sentence and use the first letter from each word. For example, "Charles will be 6 years old on November 21" becomes "Cwb6yooN21."
For more information, see "Levels of Administration" on page 345.

WebUI

Configuration > Admin > Administrators > Edit (for John): Enter the following, then click **OK**:

```
Name: Smith
New Password: 3MAb99j2
Confirm New Password: 3MAb99j2
```

CLI

```
unset admin user John
set admin user Smith password 3MAb99j2 privilege all
```

```
save
```

Example: Changing Your Own Password

Admin users with read/write privileges can change their own administrator password, but not their login name. In this example, an administrator with read/write privileges and the login name “Smith” changes his password from 3MAb99j2 to ru494Vq5.

WebUI

Configuration > Admin > Administrators > Edit (for first entry): Enter the following, then click **OK**:

```
Name: Smith
New Password: ru494Vq5
Confirm New Password: ru494Vq5
```

CLI

```
set admin password ru494Vq5
save
```

Setting the Minimum Length of the Root Admin Password

In some corporations, one person might initially configure the device as the root admin, but another person later assumes the role of root admin and manages the device. To prevent the subsequent root admin from using short passwords that are potentially easier to decode, the initial root admin can set a minimum length requirement for the root admin’s password to any number from 1 to 31.

You can set the minimum password length only if you are the root admin and your own password meets the minimum length requirement you are attempting to set. Otherwise, the security device displays an error message.

To specify a minimum length for the root admin’s password, enter the following CLI command:

set admin password restrict length *number*

NOTE: You must use the CLI to set this restriction.

Resetting the Device to the Factory Default Settings

If the admin password is lost, you can use the following procedure to reset the security device to its default settings. The configurations will be lost, but access to the device will be restored. To perform this operation, you need to make a console connection, which is described in detail in *ScreenOS CLI Reference Guide: IPv4 Command Descriptions* and the documentation for your device.

NOTE: By default, the device recovery feature is enabled. You can disable it by entering the **unset admin device-reset** command. Also, if the security device is in FIPS mode, the recovery feature is automatically disabled.

1. At the login prompt, enter the serial number of the device.
2. At the password prompt, enter the serial number again.

   The following message appears:

   !!!! Lost Password Reset !!!! You have initiated a command to reset the device to factory defaults, clearing all current configuration, keys and settings. Would you like to continue? y/n
3. Press the **y** key.

   The following message appears:

   !! Reconfirm Lost Password Reset !! If you continue, the entire configuration of the device will be erased. In addition, a permanent counter will be incremented to signify that this device has been reset. This is your last chance to cancel this command. If you proceed, the device will return to factory default configuration, which is: System IP: 192.168.1.1; username: netscreen; password: netscreen. Would you like to continue? y/n
4. Press the **y** key to reset the device. You can now log in using **netscreen** as the default username and password.

Restricting Administrative Access

You can administer security devices from one or multiple addresses of a subnet. By default, any host on the trusted interface can administer a security device. To restrict this ability to specific workstations, you must configure management client IP addresses.

Example: Restricting Administration to a Single Workstation

In this example, the administrator at the workstation with the IP address 172.16.40.42 is the only administrator specified to manage the security device.

WebUI

Configuration > Admin > Permitted IPs: Enter the following, then click **Add**:

IP Address / Netmask: 172.16.40.42/32

CLI

```
set admin manager-ip 172.16.40.42/32
save
```

Example: Restricting Administration to a Subnet

In this example, the group of administrators with workstations in the 172.16.40.0/24 subnet are specified to manage a security device.

WebUI

Configuration > Admin > Permitted IPs: Enter the following, then click **Add**:

IP Address / Netmask: 172.16.40.0/24

CLI

```
set admin manager-ip 172.16.40.0 255.255.255.0
set admin manager-ip 3ffe:7777::1/32
set admin manager-ip 3ffe:7777::1/24
save
```

Restricting the Root Admin to Console Access

You can also require the root admin to log into the security device through the console only. This restriction requires the root admin to have physical access to the device to log in, thus preventing unauthorized users from logging in remotely as the root admin. After you have set this restriction, the device denies access if anyone tries to log in as the root admin through other means, such as the WebUI, Telnet, or SSH, even if these management options are enabled on the ingress interface.

To restrict the access of the root admin to the console only, enter the following command:

```
set admin root access console
```

NOTE: You must use the CLI to set this restriction.

Monitoring Admin access

ScreenOS provides the following features to monitor and control the admin access to the security devices:

Lock and unlock interactive session

Users will be able to lock and unlock the current active session by using Exit command.

Auto lock after inactivity

ScreenOS supports this feature by employing three components:

- When the administrator is inactive on the terminal for a specified period of time the device automatically locks the session. Root-admins can specify the time period of inactivity after which the device locks the session by using the set console time command.
- When the administrator locks a session by using the exit command or when the device auto-locks the session, the security device clears all contents or overwrites the contents on display devices making it unreadable. About 50 empty lines will be flushed out on the display devices.
- After locking the session, any activity other than unlocking the session will be disabled. Users will be denied access to data in ScreenOS without reauthentication.

Restrict admin access based on time

ScreenOS restricts the admin access to users based on location by using the **set admin manager** command. For example, the following command restricts management to a single host with IP address 10.1.10.100:

```
set admin manager-ip 10.1.10.100 255.255.255.255
```

To restrict admin access based on time, use the set admin user *user_name* access schedule *scheduler_name*. When a new connection initiated by certain admin comes at firewall, the scheduler bound to the admin is checked to see if the permit time window is open. If not, the connection request will be rejected. The scheduler will be checked every 10 seconds and once the stop time is due, the admin will be restricted access to the security device.

You can view details of admins who are currently active by using the **get active admin user login** command:

```
ssg5-serial-> get admin user login

No Name Vsys Date       Time     Source  IP Addr Auth Type Role  Time Remain
-- ---- ----- ---------- -------- ------- ------- --------- ----- ----------
1  kkk  Root  2008-04-23 17:53:01 console 0.0.0.0 local     -         N/A
2  kkk  Root  2008-04-23 17:49:26 web     2.2.2.2 local     -         N/A
```

To view details of all administrators use the get admin user command:

```
ssg5-serial-> get admin user

Name Privilege       Role Scheduler
---- --------- --------
kkk        Root           -        N/A
rrr       Read-Write      -        N/A
jjj       Read-Only       -        N/A
bbb       Read-Only       -        N/A
11        Read-Only       -        N/A
```

VPN Tunnels for Administrative Traffic

You can use virtual private network (VPN) tunnels to secure remote management of a security device from either a dynamically assigned or fixed IP address. Using a VPN tunnel, you can protect any kind of traffic, such as NSM, HTTP, Telnet, or SSH. (For information about creating a VPN tunnel to secure self-initiated traffic such as NSM reports, syslog reports, or SNMP traps, see "Configuring a MIB Filter in the SNMP Community" on page 402.)

Juniper Networks security devices support two types of VPN tunnel configurations:

- **Route-based VPNs**: The security device uses route table entries to direct traffic to tunnel interfaces, which are bound to VPN tunnels.
- **Policy-based VPNs**: The security device uses the VPN tunnel names specifically referenced in policies to direct traffic through VPN tunnels.

For each VPN tunnel configuration type, there are the following types of VPN tunnel:

- **Manual key**: You manually set the three elements that define a Security Association (SA) at both ends of the tunnel: a Security Parameters Index (SPI), an encryption key, and an authentication key. To change any element in the SA, you must manually enter it at both ends of the tunnel.
- **AutoKey IKE with pre-shared key**: One or two pre-shared secrets—one for authentication and one for encryption—function as seed values. Using them, the IKE protocol generates a set of symmetrical keys at both ends of the tunnel; that is, the same key is used to encrypt and decrypt. At predetermined intervals, these keys are automatically regenerated.
- **AutoKey IKE with certificates**: Using the Public Key Infrastructure (PKI), the participants at both ends of the tunnel use a digital certificate (for authentication) and an RSA public/private key pair (for encryption). The encryption is asymmetrical; that is, one key in a pair is used to encrypt and the other to decrypt.

NOTE: For a complete description of VPN tunnels, see *"Virtual Private Networks" on page 705*. For more information on NetScreen-Remote, see the *NetScreen-Remote VPN Client Administrator Guide*.

If you use a policy-based VPN configuration, you must create an address book entry with the IP address of an interface in any zone other than the one to which the outgoing interface is bound. You can then use that as the source address in policies referencing the VPN tunnel. This address also serves as the end entity address for the remote IPsec peer. If you are using a route-based VPN configuration, such an address book entry is unnecessary.

Administration Through a Route-Based Manual Key VPN Tunnel

Figure 94 on page 359 illustrates an example in which you set up a route-based Manual Key VPN tunnel to provide confidentiality for administrative traffic. The tunnel extends

from the NetScreen-Remote VPN client running on an admin's workstation at 10.1.1.56 to ethernet1 (10.1.1.1/24). The admin's workstation and ethernet1 are both in the Trust zone. You name the tunnel "tunnel-adm". You create an unnumbered tunnel interface, name it tunnel.1, and bind it to the Trust zone and to the VPN tunnel "tunnel-adm."

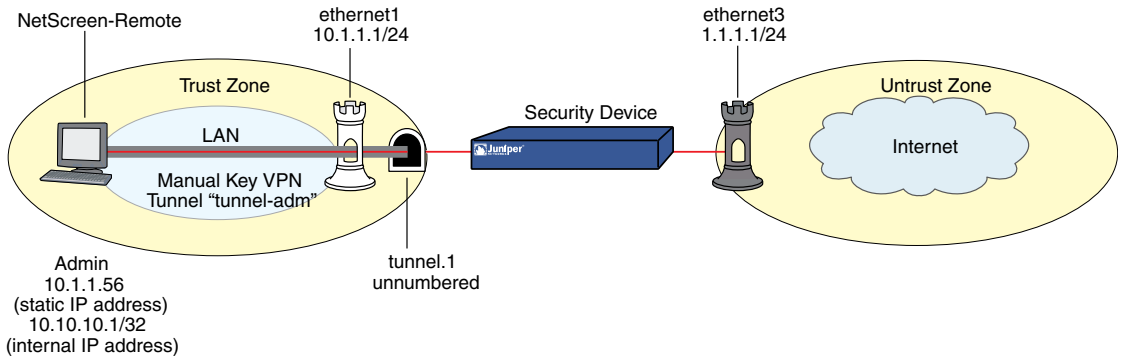
The security device uses the internal IP address configured on the NetScreen-Remote client—10.10.10.1—as the destination address to target beyond the peer gateway address of 10.1.1.56. You define a route to 10.10.10.1/32 through tunnel.1. A policy is unnecessary because of the following two reasons:

- The VPN tunnel protects administrative traffic that terminates at the security device itself instead of passing through the device to another security zone.
- This is a route-based VPN, meaning that the route lookup—not a policy lookup—links the destination address to the tunnel interface, which is bound to the appropriate VPN tunnel.

NOTE: Compare this example with "Administration Through a Policy-Based Manual Key VPN Tunnel" on page 362.

NetScreen-Remote uses the IP address of ethernet3—1.1.1.1—as the destination address to target beyond the remote gateway at 10.1.1.1. The NetScreen-Remote configuration specifies the remote party ID type as "IP address" and the protocol as "All."

Figure 94: Administration Through a Route-Based Manual Key VPN Tunnel



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

Select the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > Edit (ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: Tunnel.1
Zone (VR): Trust (trust-vr)
Unnumbered: (select)
Interface: ethernet1(trust-vr)

NOTE: The unnumbered tunnel interface borrows the IP address of the specified security zone interface.

2. **VPN**

VPNs > Manual Key > New: Enter the following, then click **OK**:

VPN Tunnel Name: tunnel-adm
Gateway IP: 10.1.1.56
Security Index (HEX Number): 5555 (Local) 5555 (Remote)
Outgoing Interface: ethernet1
ESP-CBC: (select)
Encryption Algorithm: DES-CBC
Generate Key by Password: netscreen1
Authentication Algorithm: MD5
Generate Key by Password: netscreen2

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Bind to Tunnel Interface: (select), Tunnel.1

NOTE: Because NetScreen-Remote processes passwords into keys differently than do other Juniper Networks products, after you configure the tunnel you need to do the following: (1) Return to the Manual Key Configuration dialog box (click **Edit** in the Configure column for "tunnel-adm"); (2) copy the generated hexadecimal keys; (3) use those hexadecimal keys when configuring the NetScreen-Remote end of the tunnel.

3. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.10.10.1/32
Gateway: (select)
Interface: Tunnel.1
Gateway IP Address: 0.0.0.0

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone trust
set interface tunnel.1 ip unnumbered interface ethernet1
```

NOTE: The unnumbered tunnel interface borrows the IP address of the specified security zone interface.

2. **VPN**

```
set vpn tunnel-adm manual 5555 5555 gateway 10.1.1.56 outgoing ethernet1
  esp des password netscreen1 auth md5 password netscreen2
set vpn tunnel-adm bind interface tunnel.1
```

NOTE: Because NetScreen-Remote processes passwords into keys differently than do other Juniper Networks products, after you configure the tunnel you need to do the following: (1) Enter **get vpn admin-tun**; (2) copy the hexadecimal keys generated by "netscreen1" and "netscreen2" ; (3) use those hexadecimal keys when configuring the NetScreen-Remote end of the tunnel.

3. **Route**

```
set vrouter trust-vr route 10.10.10.1/32 interface tunnel.1
save
```

NetScreen-Remote Security Policy Editor

1. Click **Options > Global Policy Settings**, and select the Allow to Specify Internal Network Address check box.
2. Click **Options > Secure > Specified Connections**.
3. Click **Add a new connection**, and enter **Admin** next to the new connection icon that appears.
4. Configure the connection options:

Connection Security: Secure

Remote Party Identity and Addressing:
ID Type: IP Address, 1.1.1.1
Protocol: All
Connect using Secure Gateway Tunnel: (select)
ID Type: IP Address, 10.1.1.1

5. Click the **PLUS** symbol, located to the left of the UNIX icon, to expand the connection policy.
6. Click **My Identity**, in the Select Certificate drop-down list, choose **None**, and in the Internal Network IP Address, enter **10.10.10.1**.
7. Click **Security Policy**, and select **Use Manual Keys**.
8. Click the **PLUS** symbol, located to the left of the Security Policy icon, then click the **PLUS** symbol to the left of Key Exchange (Phase 2) to expand the policy further.
9. Click **Proposal 1**, then select the following IPsec protocols:

   Encapsulation Protocol (ESP): (select)
   Encrypt Alg: DES
   Hash Alg: MD5
   Encapsulation: Tunnel

10. Click **Inbound Keys**, and in the Security Parameters Index field, enter 5555.
11. Click **Enter Key**, enter the following, then click **OK**:

    Choose key format: Binary
    ESP Encryption Key: dccbee96c7e546bc
    ESP Authentication Key: dccbe9e6c7e546bcb0b667794ab7290c

NOTE: These are the two generated keys that you copied after configuring the security device.

12. Click **Outbound Keys**, and, in the Security Parameters Index field, enter 5555.
13. Click **Enter Key**, enter the following, then click **OK**:

    Choose key format: Binary
    ESP Encryption Key: dccbee96c7e546bc
    ESP Authentication Key: dccbe9e6c7e546bcb0b667794ab7290c

14. Click **Save**.

Administration Through a Policy-Based Manual Key VPN Tunnel

Figure 95 on page 363 illustrates an example in which you set up a policy-based Manual Key VPN tunnel for administrative traffic. The tunnel extends from the NetScreen-Remote VPN client running on an admin's workstation at 10.1.1.56 to ethernet1 (10.1.1.1/24). The admin's workstation and ethernet1 are both in the Trust zone. You name the tunnel "tunnel-adm" and bind it to the Trust zone.

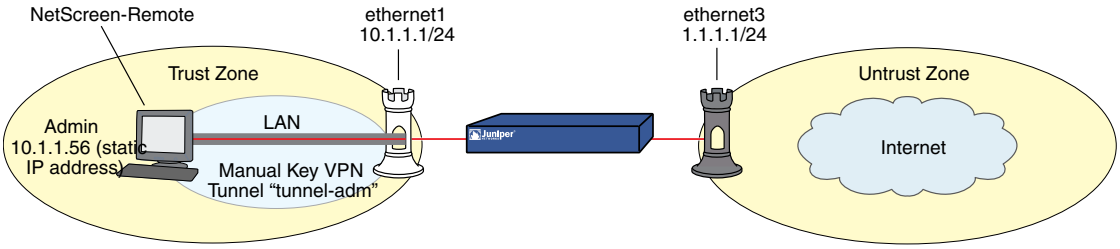
The security device uses the internal IP address configured on the NetScreen-Remote—10.10.10.1—as the destination address to target beyond the peer gateway address of 10.1.1.56. You define a Trust zone address book entry specifying 10.10.10.1/32, and an Untrust zone address book entry specifying the IP address of ethernet3. Although the address of the ethernet3 interface is 1.1.1.1/24, the address you create has a 32-bit netmask: 1.1.1.1/32. You use this address and the internal address of the admin's workstation in the policy you create referencing the tunnel "tunnel-adm". A policy is necessary because this is a policy-based VPN, meaning that the policy lookup—not a route lookup—links the destination address to the appropriate VPN tunnel.

You must also define a route to 10.10.10.1/32 through ethernet1.

NOTE: Compare this example with "Administration Through a Route-Based Manual Key VPN Tunnel" on page 358.

NetScreen-Remote uses the IP address 1.1.1.1 as the destination address to target beyond the remote gateway at 10.1.1.1. The NetScreen-Remote tunnel configuration specifies the remote party ID type as IP address and the protocol as "All."

Figure 95: Administration Through a Policy-Based Manual Key VPN Tunnel



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Addresses**

Policy > Policy Elements > Addresses > Lists > New: Enter the following, then click **OK**:

Address Name: Untrust-IF
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.1/32
Zone: Untrust

Policy > Policy Elements > Addresses > Lists > New: Enter the following, then click **OK**:

Address Name: admin
IP Address/Domain Name:
IP/Netmask: (select), 10.10.10.1/32
Zone: Trust

3. **VPN**

VPNs > Manual Key > New: Enter the following, then click **OK**:

VPN Tunnel Name: tunnel-adm
Gateway IP: 10.1.1.56
Security Index (HEX Number): 5555 (Local) 5555 (Remote)
Outgoing Interface: ethernet1
ESP-CBC: (select)
Encryption Algorithm: DES-CBC
Generate Key by Password: netscreen1
Authentication Algorithm: MD5
Generate Key by Password: netscreen2

NOTE: Because NetScreen-Remote processes passwords into keys differently than do other Juniper Networks products, after you configure the tunnel you need to do the following: (1) Return to the Manual Key Configuration dialog box (click **Edit** in the Configure column for "tunnel-adm"); (2) copy the generated hexadecimal keys; (3) use those hexadecimal keys when configuring the NetScreen-Remote end of the tunnel.

4. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.10.10.1/32
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 0.0.0.0

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), admin

Destination Address:
    Address Book Entry: (select), Untrust-IF
Service: Any
Action: Tunnel
Tunnel:
    VPN: tunnel-adm
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address trust admin 10.10.10.1/32
set address untrust Untrust-IF 1.1.1.1/32
```

3. **VPN**

```
set vpn tunnel-adm manual 5555 5555 gateway 10.1.1.56 outgoing ethernet1
 esp des password netscreen1 auth md5 password netscreen2
```

NOTE: Because NetScreen-Remote processes passwords into keys differently than do other Juniper Networks products, after you configure the tunnel you need to do the following: (1) Enter **get vpn admin-tun**; (2) copy the hexadecimal keys generated by "netscreen1" and "netscreen2" ; (3) use those hexadecimal keys when configuring the NetScreen-Remote end of the tunnel.

4. **Route**

```
set vrouter trust-vr route 10.10.10.1/32 interface ethernet1
```

5. **Policies**

```
set policy top from trust to untrust admin Untrust-IF any tunnel vpn tunnel-adm
set policy top from untrust to trust Untrust-IF admin any tunnel vpn tunnel-adm
save
```

NetScreen-Remote Security Policy Editor

1. Click **Options > Secure > Specified Connections**.
2. Click **Add a new connection**, and enter **Admin** next to the new connection icon that appears.

3. Configure the connection options:

   Connection Security: Secure
   Remote Party Identity and Addressing:
   ID Type: IP Address, 1.1.1.1
   Protocol: All
   Connect using Secure Gateway Tunnel: (select)
   ID Type: IP Address, 10.1.1.1

4. Click the **PLUS** symbol, located to the left of the UNIX icon, to expand the connection policy.
5. Click **My Identity**, and, in the **Select Certificate** drop-down list, choose **None**.
6. Click **Security Policy**, and select **Use Manual Keys**.
7. Click the **PLUS** symbol, located to the left of the Security Policy icon, and then the **PLUS** symbol to the left of Key Exchange (Phase 2) to expand the policy further.
8. Click **Proposal 1**, and select the following IPsec protocols:

   Encapsulation Protocol (ESP): (select)
   Encrypt Alg: DES
   Hash Alg: MD5
   Encapsulation: Tunnel

9. Click **Inbound Keys**, and in the Security Parameters Index field, enter 5555.
10. Click **Enter**, enter the following, and click **OK**:

    Choose key format: Binary
    ESP Encryption Key: dccbee96c7e546bc
    ESP Authentication Key: dccbe9e6c7e546bcb0b667794ab7290c

NOTE: These are the two generated keys that you copied after configuring the security device.

11. Click **Outbound Keys**, and in the Security Parameters Index field, enter 5555.
12. Click **Enter Key**, enter the following, then click **OK**:

    Choose key format: Binary
    ESP Encryption Key: dccbee96c7e546bc
    ESP Authentication Key: dccbe9e6c7e546bcb0b667794ab7290c

13. Click **Save**.

Password Policy

The password policy feature allows you to enforce a minimum length and a complexity scheme for administrator (admin) and authenticated (auth) user passwords. The password policy feature is intended for use in a local database, and therefore is useful in environments where the Windows directory or RADIUS are not available to provide centralized password policy enforcement.

Setting a Password Policy

You can create a password policy to require that admin and auth passwords fulfill one or both of the following:

- Minimum length
- Complexity

The range for password minimum length is 1 to 32 characters. Use the following command to create a password policy requiring a minimum length of 8 characters for admin passwords:

```
set password-policy user-type admin minimum-length 8
```

Password complexity means passwords must include at least two uppercase letters, two lowercase letters, and two alphanumeric and two non-alphanumeric characters; for example: AAbb12@#. To require that passwords contain complexity, you set complexity to 1. To unset the complexity requirement, set complexity to 0. Use the following command to create a password policy requiring that auth passwords contain complexity:

```
set password-policy user-type auth complexity 1
```

In the following example, you create a password policy for admin and auth accounts requiring complexity and a minimum length of 8 characters:

CLI

```
set password-policy user-type admin minimum-length 8
set password-policy user-type admin complexity 1
set password-policy user-type auth minimum-length 8
set password-policy user-type auth complexity 1
save
```

NOTE: You can configure a password policy only from the command line interface (CLI).

Removing a Password Policy

Use the **unset password-policy** command to delete a password policy. When you remove a password policy, the password requirement for the account reverts to the default settings. In the following example, you remove the minimum length requirement for auth passwords.

CLI

```
unset password-policy user-type auth minimum-length
```

Viewing a Password Policy

Use the **get password-policy** command to display the password policy for admin and auth users.

Recovering from a Rejected Default Admin Password

When you delete (unset) the root admin account on a device on which you have a password policy configured, you might need to set a new admin password before logging off the system. This is because ScreenOS reverts to the default password (*netscreen*) when you delete the root admin account. If you have a password policy requiring complexity, or a minimum length greater than 9 characters, your next login attempt will fail. If this happens, use the asset recovery procedure to gain access to the device. Refer to the installation and configuration guide for your device for details.

In the following example, you delete the admin account named **admin2005**, then display the current password policy. As shown, the policy specifies that passwords must have a minimum length of 8 characters, and use complexity (a minimum of two uppercase, two lowercase, two alphanumeric, and two non-alphanumeric characters). You then create a new admin account named **admin2006** and set a password for it that fulfills the minimum length and complexity requirements of the password policy.

CLI

```
unset admin admin2005
get password-policy

user-type: admin
password minimum length: 8
password complexity scheme: 1

user-type: auth
password minimum length: 8
password complexity scheme: 1

set admin admin2006 password AAbb12@#
save
```

NOTE: You can configure an admin account only from the command line interface (CLI).

Creating a Login Banner

The size of the login banner is increased to a maximum of 4Kbytes. This provides space for terms of use statements, which are presented before administrators and authenticated users log into the security device and into protected resources behind the device. The login banner is a clear text ASCII file you create and store on the security device, the file must be called usrterms.txt. You activate the banner by

restarting of the device. If the banner file is greater than 4Kbytes, the security device will not accept it and will continue using existing banners entered through the CLI and the WebUI.

When activated, the login banner is used globally by the root device and all virtual systems (vsys). You cannot differentiate or customize between or within a vsys. The login banner preempts all individually defined administrative access banners and firewall authentication banners. After entering a username and password, the user must click the Login button. Pressing the Enter key will not log the user into the device.

Use the SCP utility to securely copy the banner file to the security device. With the following command, an administrator with username netscreen copies the banner file my_large_banner.txt to a security device at IP address 1.1.1.2. The banner file must be saved on the security device as usrterms.txt.

You must restart the device to activate the new banner. To modify the banner file, create a new file and overwrite the existing one with the new one.

To remove the banner, issue the following command on the security device:

```
device-> delete file usrterms.txt
```

This disables the login banner feature after you restart the device.

Chapter 11

Monitoring Security Devices

This chapter discusses the following topics about monitoring Juniper Networks security devices. It contains the following sections:

Storing Log Information

All Juniper Networks security devices allow you to store event and traffic log data internally (in flash storage) and externally (in a number of locations). Although storing log information internally is convenient, the amount of device memory is limited. When the internal storage space completely fills up, the security device begins overwriting the oldest log entries with the latest ones. If this first-in-first-out (FIFO) mechanism occurs before you save the logged information, you can lose data. To mitigate such data loss, you can store event and traffic logs externally in a syslog or WebTrends server or in the NetScreen-Global PRO database. The security device sends new event and traffic log entries to an external storage location every second.

The following list provides the possible destinations for logged data:

- **Console:** A destination for all log entries to appear when you are troubleshooting a security device through the console. Optionally, you might elect to have only alarm messages (critical, alert, emergency) appear here to alert you immediately if you happen to be using the console at the time an alarm is triggered.
- **Internal:** Allows you store a limited number of log entries.
- **Email:** A method for sending event and traffic logs to remote administrators.

- **SNMP:** In addition to the transmission of SNMP traps, a security device can also send alarm messages (critical, alert, emergency) from its event log to an SNMP community.
- **Syslog:** All event and traffic log entries that a security device can store internally, it can also send to a syslog server. Because syslog servers have a much greater storage capacity than the internal flash storage on a security device, sending data to a syslog server can mitigate data loss that might occur when log entries exceed the maximum internal storage space. Syslog stores alert- and emergency-level events in the security facility that you specify, and all other events (including traffic data) in the facility you specify.
- **WebTrends:** Allows you to view log data for critical-, alert-, and emergency-level events in a more graphical format than syslog, which is a text-based tool.
- **CompactFlash (PCMCIA):** Allows you to store data on a CompactFlash card.
- **USB:** Allows you to store data on a USB flash drive. When you set USB as the log destination, the system sends log messages to a file on the USB flash drive. The log file is named *hostname_date*.evt_log, where *hostname* is the system hostname at boot time, and *date* is date on which the device was last started. Logging to USB is disabled by default; use the **set log usb enable** CLI command to enable USB logging.

Use the **set log... destination...** command to set the severity levels for all log destinations. The following example logs all system module messages of level critical or higher to the USB flash drive:

```
set log module system level critical destination usb
```

Event Log

ScreenOS provides an event log for monitoring system events such as admin-generated configuration changes, and self-generated messages and alarms regarding operational behavior and attacks. The security device categorizes system events by the following severity levels:

- **Emergency:** Messages on SYN attacks, Tear Drop attacks, and Ping of Death attacks. For more information on these types of attacks, see “Attack Detection and Defense Mechanisms” on page 431.
- **Alert:** Messages about conditions that require immediate attention, such as firewall attacks and the expiration of license keys.
- **Critical:** Messages about conditions that probably affect the functionality of the device, such as high availability (HA) status changes.
- **Error:** Messages about error conditions that probably affect the functionality of the device, such as a failure in antivirus scanning or in communicating with SSH servers.
- **Warning**: Messages about conditions that could affect the functionality of the device, such as a failure to connect to email servers or authentication failures, timeouts, and successes.
- **Notification:** Messages about normal events, including configuration changes initiated by an admin.

- **Information:** Messages that provide general information about system operations.
- **Debugging:** Messages that provide detailed information used for debugging purposes.

The event log displays the date, time, level and description of each system event. You can view system events for each category stored in flash storage on the security device through the WebUI or the CLI. You can also open or save the file to the location you specify, and then use an ASCII text editor (such as Notepad or WordPad) to view the file. Alternatively, you can send them to an external storage space (see "Storing Log Information" on page 371).

NOTE: For detailed information about the messages that appear in the event log, see the *ScreenOS Message Log Reference Guide*.

Viewing the Event Log by Severity Level and Keyword

You can view the event log stored in the device by using the CLI or the WebUI. You can display log entries by severity level and search the event log by keyword in both the WebUI and CLI.

To display the event log by severity level, do either of the following:

WebUI

Reports > System Log > Event: Select a severity level from the Log Level drop-down list.

CLI

```
get event level { emergency | alert | critical | error | warning | notification | information
 | debugging }
```

To search the event log by keyword, do either of the following:

WebUI

Reports > System Log > Event: Enter a word or phrase up to 15 characters in length in the search field, then click **Search**.

CLI

```
get event include word_string
```

In this example, you view event log entries with a "warning" severity level and do a search for the keyword AV.

WebUI

Reports > System Log > Event:

Log Level: Warning (select)

Search: Enter AV, then click **Search**.

CLI

```
get event level warning include av
```

Date	Time	Module Level	Level	Type	Description
2003-05-16	15:56:20	system warn	00547	AV	scanman is removed.
2003-05-16	09:45:52	system warn	00547	AV	test1 is removed.
Total entries matched = 2					

Sorting and Filtering the Event Log

Additionally, you can use the CLI to sort or filter the event log based on the following criteria:

- **Source or Destination IP Address:** Only certain events contain a source or destination IP address, such as authentication, land attacks, or ping flood attacks. When you sort event logs by source or destination IP address, the device sorts and displays only the event logs that contain source or destination IP addresses. It ignores all event logs with no source or destination IP address. The authentication log messages include the user's IP address.

  When you filter the event log by specifying a source or destination IP address or range of addresses, the device displays the log entries for the specified source or destination IP address, or range of addresses.

- **Date**: You can sort the event log by date only, or by date and time. When you sort log entries by date and time, the device lists the log entries in descending order by date and time.

  You can also filter event log entries by specifying a start date, an end date, or a date range. When you specify a start date, the device displays log entries with date/time stamps after the start date. When you specify an end date, the device displays log entries with date/time stamps before the end date.

- **Time:** When you sort logs by time, the device displays the log entries in descending order by time, regardless of the date. When you specify a start time, the device displays log entries with time stamps after the specified start time, regardless of the date. When you specify an end time, the device displays log entries with time stamps before the specified end time, regardless of the date.

When you specify both a start and end time, the device displays log entries with time stamps within the specified time period.

- **Message Type ID Number:** You can display event log entries for a specific message type ID number, or you can display log entries with message type ID numbers within a specified range. The device displays log entries with the message type ID number(s) you specified, in descending order by date and time.

In this example you view event log entries that contain source IP addresses within the range 10.100.0.0 to 10.200.0.0. The log entries are also sorted by source IP address.

WebUI

NOTE: You must use the CLI to sort the event log by address entries.

CLI

```
get event sort-by src-ip 10.100.0.0-10.200.0.0
```

Downloading the Event Log

You can open or save the event log to the location you specify, and then use an ASCII text editor (such as Notepad or WordPad) to view the file. Alternatively, you can send the log entries to an external storage space (see "Storing Log Information" on page 371). You can download the entire event log through the WebUI. You can download the event log by severity level through the CLI.

Example: Downloading the Entire Event Log

In this example, you download the event log to the local directory. Using the WebUI, you download it to *C:\netscreen\logs*. Using the CLI, you download it to the root directory of a TFTP server at the IP address 10.1.1.5. You name the file "evnt07-02.txt."

WebUI

1. Reports > System Log > Event: Click **Save**.

   The File Download dialog box prompts you to open the file (using an ASCII editor) or save it to disk.

2. Select the **Save** option, then click **OK**.

   The File Download dialog box prompts you to choose a directory.

3. Specify **C:\netscreen\logs**, name the file *evnt07-02.txt*, then click **Save**.

CLI

```
get event > tftp 10.1.1.5 evnt07-02.txt
```

Example: Downloading the Event Log for Critical Events

In this example, you download the critical events entered in the event log to the root directory of a TFTP server at the IP address 10.1.1.5. You name the file *crt_evnt07-02.txt*.

WebUI

NOTE: You must use the CLI to download entries by severity level.

CLI

```
get event level critical > tftp 10.1.1.5 crt_evnt07-02.txt
```

Traffic Log

The Juniper Networks security device can monitor and record traffic that it permits or denies based on previously configured policies. You can enable the logging option for each policy that you configure. When you enable the logging option for a policy that permits traffic, the device records the traffic allowed by that policy. When you enable the logging option for a policy that denies traffic, the device records traffic that attempted to pass through the device, but was dropped because of that policy.

A traffic log notes the following elements for each session:

- Date and time that the connection started
- Duration
- Source address and port number
- Translated source address and port number
- Destination address and port number
- The duration of the session
- The service used in the session

To log all traffic that a security device receives, you must enable the logging option for all policies. To log specific traffic, enable logging only on policies that apply to that traffic. To enable the logging option on a policy, do either of the following:

WebUI

Policies > (From: *src_zone*, To: *dst_zone*) New: Select **Logging** and then click **OK**.

CLI

set policy from *src_zone* to *dst_zone src_addr dst_addr service action* log

In addition to logging traffic for a policy, the device can also maintain a count in bytes of all network traffic to which the policy was applied. When you enable the counting option, the device includes the following information when it displays traffic log entries

- Bytes transmitted from a source to a destination
- Bytes transmitted from a destination to a source

You can enable counting on a policy from the WebUI and from the CLI.

WebUI

Policies > (From: src_zone, To: dst_zone) New > Advanced: Select **Counting**, click **Return**, then click **OK**.

CLI

set policy from *src_zone* to *dst_zone src_addr dst_addr service action* log count

Viewing the Traffic Log

You can view traffic log entries stored in flash storage on the security device using either the WebUI or the CLI.

WebUI

Policies > Logging (for policy ID *number*)

or

Reports > Policies > Logging (for policy ID *number*)

CLI

get log traffic policy *number*

Example: Viewing Traffic Log Entries

In this example, you view the traffic log details of a policy with ID number 3, and for which you have previously enabled logging:

WebUI

Policies: Click the Logging icon for the policy with ID number 3.

The following information appears:

- Date/Time: 2003-01-09 21:33:43
- Duration: 1800 sec.
- Source IP Address/Port: 1.1.1.1:1046
- Destination IP Address/Port: 10.1.1.5:80
- Service: HTTP
- Reason for Close: Age out
- Translated Source IP Address/Port: 1.1.1.1:1046
- Translated Destination IP Address/Port: 10.1.1.5:80
- Policy ID number: 3

CLI

```
get log traffic policy 3
```

Sorting and Filtering the Traffic Log

Similar to the event log, when you use the CLI to view the traffic log, you can sort or filter the log entries according to the following criteria:

- **Source or Destination IP Address:** You can sort the traffic log by source or destination IP address. You can also filter the traffic log by specifying a source or destination IP address or range of addresses.
- **Date**: You can sort the traffic log by date only, or by date and time. The device lists the log entries in descending order by date and time.

  You can also filter event log entries by specifying a start date, an end date, or a date range. When you specify a start date, the device displays log entries with date/time stamps after the start date. When you specify an end date, the device displays log entries with date/time stamps before the end date.
- **Time:** When you sort the traffic log by time, the device displays the log entries in descending order by time, regardless of the date. When you specify a start time, the device displays log entries with time stamps after the specified start time, regardless of the date. When you specify an end time, the device displays log entries with time stamps before the specified end time, regardless of the date. When you specify both a start and end time, the device displays log entries with time stamps within the specified time period.

Example: Sorting the Traffic Log by Time

In this example you view the traffic log sorted by time with a time stamp after 1:00 a.m.

WebUI

NOTE: The ability to sort the traffic log by time is available only through the CLI.

CLI

```
get log traffic sort-by time start-time 01:00:00
```

Downloading the Traffic Log

You can also open or save the log to the location you specify, and then use an ASCII text editor (such as Notepad or WordPad) to view the file.

Alternatively, you can send traffic log entries to an external storage space (see "Storing Log Information" on page 371). The security device makes an entry in the traffic log when a session terminates. When you enable the security device to send traffic log entries to an external storage location, it sends new entries every second. Because the security device makes a traffic log entry when a session closes, the security device sends traffic log entries for all sessions that have closed within the past second. You can also include traffic log entries with event log entries sent by email to an admin.

In this example, you download the traffic log for a policy with ID number 12. For the WebUI, you download it to the local directory "C:\netscreen\logs". For the CLI, you download it to the root directory of a TFTP server at the IP address 10.10.20.200. You name the file " traf_log11-21-02.txt."

WebUI

1. Reports > Policies > Logging (for policy ID 12): Click **Save**.

   The File Download dialog box prompts you to open the file (using an ASCII editor) or save it to disk.

2. Select the **Save** option, then click **OK**.

   The File Download dialog box prompts you to choose a directory.

3. Specify C:\netscreen\logs, name the file traf_log11-21-02.txt, then click **Save**.

CLI

```
get log traffic policy 12 > tftp 10.10.20.200 traf_log11-21-02.txt
```

Removing the Reason for Close Field

By default ScreenOS records and displays the reason for session close so that you can differentiate session creation messages from session close messages. If you do not want the reason to display, you can explicitly configure the device not to display the field.

Table 42 on page 380 lists the reasons for session close that ScreenOS identifies. Any session that cannot be identified is labeled OTHER.

Table 42: Reason Codes for Session Close

Logged Reason	Meaning
TCP FIN	TCP connection torn down due to FIN packet.
TCP RST	TCP connection torn down due to RST packet.
RESP	Special sessions, such as PING and DNS, close when response is received.
ICMP	ICMP error received.
AGE OUT	Connection aged out normally.
ALG	ALG forced session close either due to error or other reason specific to that ALG.
NSRP	NSRP session close message received.
AUTH	Auth failure.
IDP	Closed by IDP.
SYN PROXY FAIL	SYN Proxy failure.
SYN PROXY LIMIT	System limit for SYN proxy sessions reached.
TENT2NORM CONV	Failure of tentative to normal session conversion.
PARENT CLOSED	Parent session closed.
CLI	User command closed.
OTHER	Reason for close not identified.

Sample traffic log with reason for close listed:

```
device-> get log traffic
PID 1, from Trust to Untrust, src Any, dst Any, service ANY, action Permit
Total traffic entries matched under this policy = 2300
================================================================================
Date Time Duration Source IP Port Destination IP Port Service
Reason Xlated Src IP Port Xlated Dst IP Port ID
================================================================================
2001-10-25 07:08:51 0:00:59 10.251.10.25 137 172.24.16.10 137 NETBIOS (NS)
Close - AGE OUT 172.24.76.127 8946 172.24.16.10 137
2001-10-25 07:08:51 0:00:59 10.251.10.25 137 172.24.244.10 137 NETBIOS (NS)
Close - AGE OUT 172.24.76.127 8947 172.24.244.10 137
2001-10-25 07:07:53 0:00:01 10.251.10.25 1028 172.24.16.10 53 DNS
Close - RESP 172.24.76.127 8945 172.24.16.10 53
2001-10-25 07:06:29 0:01:00 10.251.10.25 138 172.24.244.10 138 NETBIOS (DGM)
Close - AGE OUT 172.24.76.127 8933 172.24.244.10 138
```

```
2001-10-25 07:06:11 0:03:16 10.251.10.25 2699 172.24.60.32 1357 TCP PORT 1357
Close - TCP FIN 172.24.76.127 8921 172.24.60.32 1357
```

Sample traffic log without reason for close listed:

```
device-> get log traffic
PID 1, from Trust to Untrust, src Any, dst Any, service HTTP, action Permit
Total traffic entries matched under this policy = 1538
================================================================================
Date Time Duration Source IP Port Destination IP Port Service
Xlated Src IP Port Xlated Dst IP Port ID
================================================================================
2002-07-19 15:53:11 0:01:33 10.251.10.25 2712 207.17.137.108 80 HTTP
10.251.10.25 2712 207.17.137.108 80
2002-07-19 15:51:33 0:00:12 10.251.10.25 2711 66.163.175.128 80 HTTP
10.251.10.25 2711 66.163.175.128 80
2002-07-19 15:41:33 0:00:12 10.251.10.25 2688 66.163.175.128 80 HTTP
10.251.10.25 2688 66.163.175.128 80
2002-07-19 15:31:39 0:00:18 10.251.10.25 2678 66.163.175.128 80 HTTP
10.251.10.25 2678 66.163.175.128 80
```

In the following example, you configure the device to not display the reason for closing sessions because it interferes with a script that you want to run on the traffic log. You must use the command line interface to change the log output style.

WebUI

Not available.

CLI

```
set log traffic detail 0
save
```

Self Log

ScreenOS provides a self log to monitor and record all packets terminated at the security device. For example, if you disable some management options on an interface—such as WebUI, SNMP, and ping—and HTTP, SNMP, or ICMP traffic is sent to that interface, entries appear in the self log for each dropped packet.

To activate the self log, do one of the following:

WebUI

Configuration > Report Settings > Log Settings: Select the **Log Packets Terminated to Self** check box, then click **Apply**.

CLI

```
set firewall log-self
```

When you enable the self log, the security device logs the entries in two places: the self log and the traffic log. Similar to the traffic log, the self log displays the date, time, source address/port, destination address/port, duration, and service for each dropped packet terminating at the security device. Self log entries typically have a source zone of Null and a destination zone of "self."

Viewing the Self Log

You can view the self log, which is stored in flash storage on the security device, through either the CLI or WebUI.

WebUI

Reports > System Log > Self

CLI

```
get log self
```

Sorting and Filtering the Self Log

Similar to the event and traffic logs, when you use the CLI to view the self log, you can sort or filter the log entries according to the following criteria:

- **Source or Destination IP Address:** You can sort the self log by source or destination IP address. You can also filter the self log by specifying a source or destination IP address or range of addresses.
- **Date**: You can sort the self log by date only, or by date and time. The device lists the log entries in descending order by date and time.

  You can also filter self log entries by specifying a start date, an end date, or a date range. When you specify a start date, the device displays log entries with date/time stamps after the start date. When you specify an end date, the device displays log entries with date/time stamps before the end date.
- **Time:** When you sort the self log by time, the security device displays the log entries in descending order by time, regardless of the date. When you specify a start time, the device displays log entries with time stamps after the specified start time, regardless of the date. When you specify an end time, the device displays log entries with time stamps before the specified end time, regardless of the date. When you specify both a start and end time, the device displays log entries with time stamps within the specified time period.

Example: Filtering the Self Log by Time

In this example, you filter self log entries by the end time. The security device displays log entries with time stamps before the specified end time:

WebUI

NOTE: The ability to filter the self log by time is available only through the CLI.

CLI

```
get log self end-time 16:32:57
```

Date	Time	Duration	Source IP	Port	Destination IP	Port	Service
2003-08-21	16:32:57	0:00:00	10.100.25.1	0	224.0.0.5	0	OSPF
2003-08-21	16:32:47	0:00:00	10.100.25.1	0	224.0.0.5	0	OSPF
Total entries matched = 2							

Storing Debug Information

The Juniper Networks security device allows you to store debug information on the debug buffer or a USB flash drive. By setting USB as the destination for debug information, you can increase the amount of debug information stored and reduce the time required to retrieve it.

The debug buffer size is limited to 4 MB. When the debug buffer fills up, the security device begins overwriting the oldest debug information. You can mitigate such data loss by setting a USB flash drive as the storage destination using the **set dbuf usb enable.** command.

When you set USB as the destination for debug information, the security device sends debug information to the debug buffer and to a file on the USB flash drive. The default file is named *hostname_date.def.txt*, where *hostname* is the system hostname at boot time and *date* is the date on which the device was last started. You use **set dbuf usb default-file** to activate the default file. Alternatively, you can use the **set dbuf usb filename** command to create and activate a new file for storing the debug information. You can create a maximum of 32 files, but only one file can be active at a time.

NOTE: Do not use special symbols or characters in filenames. Acceptable characters are A-Z, a-z, 0-9, and underline.

You use the **set dbuf usb filesize** command to set the maximum size of the files stored on a USB flash drive. When the file size exceeds 90 percent of the maximum file size, a warning message will be displayed. You can increase the amount of debug information stored in a file by defining a larger file size.

You can view file entry details by issuing the **get dbuf usb** command.

Downloading the Self Log

You can also save the log as a text file to a location you specify, and then use an ASCII text editor (such as Notepad or WordPad) to view it.

In this example, you download a self log to the local directory "C:\netscreen\logs" (WebUI) or to the root directory of a TFTP server at the IP address 10.1.1.5 (CLI). You name the file "self_log07-03-02.txt."

WebUI

1. Reports > System Log > Self: Click **Save**.

   The File Download dialog box prompts you to open the file (using an ASCII editor) or save it to disk.

2. Select the **Save** option, then click **OK**.

   The File Download dialog box prompts you to choose a directory.

3. Specify C:\netscreen\logs, name the file self_log07-03-02.txt, then click **Save**.

CLI

```
get log self > tftp 10.1.1.5 self_log07-03-02.txt
```

Downloading the Asset Recovery Log

A Juniper Networks security device provides an asset recovery log to display information about each time the device is returned to its default settings using the asset recovery procedure (see "Resetting the Device to the Factory Default Settings" on page 354). In addition to viewing the asset recovery log through the WebUI or CLI, you can also open or save the file to the location you specify. Use an ASCII text editor (such as Notepad) to view the file.

In this example, you download the asset recovery log to the local directory "C:\netscreen\logs" (WebUI) or to the root directory of a TFTP server at the IP address 10.1.1.5 (CLI). You name the file "sys_rst.txt,"

WebUI

1. Reports > System Log > Asset Recovery: Click Save.

   The File Download dialog box prompts you to open the file (using an ASCII editor) or save it to disk.

2. Select the **Save** option, then click **OK**.

   The File Download dialog box prompts you to choose a directory.

3. Specify C:\netscreen\logs, name the file sys_rst.txt, then click **Save**.

CLI

```
get log asset-recovery > tftp 10.1.1.5 sys_rst.txt
```

Traffic Alarms

The security device supports traffic alarms when traffic exceeds thresholds that you have defined in policies. You can configure the security device to alert you through one or more of the following methods whenever the security device generates a traffic alarm:

- Console
- Internal (Event Log)
- Email
- SNMP
- Syslog
- WebTrends
- NetScreen-Global PRO

You set alarm thresholds to detect anomalous activity. To know what constitutes anomalous activity, you must first establish a baseline of normal activity. To create such a baseline for network traffic, you must observe traffic patterns over a period of time. Then, after you have determined the amount of traffic that you consider as normal, you can set alarm thresholds above that amount. Traffic exceeding that threshold triggers an alarm to call your attention to a deviation from the baseline. You can then evaluate the situation to determine what caused the deviation and whether you need to take action in response.

You can also use traffic alarms to provide policy-based intrusion detection and notification of a compromised system. Examples of the use of traffic alarms for these purposes are provided below.

Example: Policy-Based Intrusion Detection

In this example, there is a Web server with IP address 211.20.1.5 (and name "web1") in the DMZ zone. You want to detect any attempts from the Untrust zone to access this Web server with Telnet. To accomplish this, you create a policy denying Telnet traffic from any address in the Untrust zone destined to the Web server named web1 in the DMZ zone, and you set a traffic alarm threshold at 64 bytes. Because the smallest size of IP packet is 64 bytes, even one Telnet packet attempting to reach the Web server from the Untrust zone will trigger an alarm.

WebUI

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: web1
IP Address/Domain Name:
IP/Netmask: (select), 211.20.1.5/32
Zone: DMZ

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), web1
Service: Telnet
Action: Deny

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Counting: (select)
Alarm Threshold: 64 Bytes/Sec, 0 Kbytes/Min

CLI

```
set address dmz web1 211.20.1.5/32
set policy from untrust to dmz any web1 telnet deny count alarm 64 0
save
```

Example: Compromised System Notification

In this example, you use traffic alarms to provide notification of a compromised system. You have an FTP server with IP address 211.20.1.10 (and name ftp1) in the DMZ zone. You want to allow FTP-get traffic to reach this server. You don't want traffic of any kind to originate from the FTP server. The occurrence of such traffic would indicate that the system has been compromised, perhaps by a virus similar to the NIMDA virus. You define an address for the FTP server in the Global zone, so that you can then create two global policies.

WebUI

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: ftp1
IP Address/Domain Name:
IP/Netmask: (select), 211.20.1.10/32
Zone: Global

Policies > (From: Global, To: Global) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), ftp1
Service: FTP-Get
Action: Permit

Policies > (From: Global, To: Global) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), ftp1
Destination Address:
    Address Book Entry: (select), Any
Service: ANY
Action: Deny

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Counting: (select)
Alarm Threshold: 64 Bytes/Sec, 0 Kbytes/Min

CLI

```
set address global ftp1 211.20.1.10/32
set policy global any ftp1 ftp-get permit
set policy global ftp1 any deny count alarm 64 0
save
```

Example: Sending Email Alerts

In this example, you set up notification by email alerts when there is an alarm. The mail server is at 172.16.10.254, the first email address to be notified is jharker@juniper.net, and the second address is driggs@juniper.net. The security device includes traffic logs with event logs sent with email.

WebUI

Configuration > Report Settings > Email: Enter the following information, then click **Apply**:

Enable E-Mail Notification for Alarms: (select)
Include Traffic Log: (select)
  SMTP Server Name: 172.16.10.254
E-Mail Address 1: jharker@juniper.net
E-Mail Address 2: driggs@juniper.net

NOTE: If you have DNS enabled, you can also use a hostname for the mail server, such as mail.juniper.net.

CLI

```
set admin mail alert
set admin mail mail-addr1 jharker@juniper.net
set admin mail mail-addr2 driggs@juniper.net
set admin mail server-name 172.16.10.254
set admin mail traffic-log
save
```

Security Alarms and Audit Logs

Juniper Networks security devices enable you to configure automatic security alarms. When the security device detects a security violation (event), it alerts the network security administrator and the users in the network with a security alarm. The security alarm is displayed on the console.

NOTE: You need to have security administrator privileges to configure and monitor security alarms and audit logs.

The security device provides an auditable event log for monitoring all security events. An audit log records the following elements for each event:

- Date and time the audit log was generated
- Module that generated the audit log
- Severity level of the event
- Type of the event
- Detailed description of each security alarm event
  - Acknowledgment ID (unique)
  - ID of the authenticated user
  - Source IP address
  - Destination IP address and port number
  - Exclude rule ID

ScreenOS supports persistent storage mechanisms such as the syslog server, which stores all audit logson a remote server. You can also save the audit log files to an external storage device. For more information about external storage space, see “Storing Log Information” on page 371.

NOTE: For detailed information about the messages that appear in the audit log, see the *ScreenOS Message Log Reference Guide*.

Enabling Security Alarms

You can configure the security device to generate an automatic alarm when it detects a security violation.

To log all alarms that a security device receives, you must enable the security alarm option. You can enable security alarms through the WebUI or the CLI.

WebUI

Configuration > Admin > Audit Setting: Enter the following, then click **Apply**:

Enable Alarm Security: (select)

CLI

set alarm security enable

In addition to enabling security alerts, you can configure the security device to send audible security alarms at regular intervals.

- **Audible Alarms**—The security device displays security alarm messages on consoles with an audible bell sound. The audible message is displayed on consoles that are logged in, logging out, or logging into the network when the admin is acknowledging the security alarm.
- **Local-force**—By default, the audible security alarm is visible to a local system only if the security administrator has logged in. By enabling the local-force feature, the security device displays the security alarm on a local system regardless of whether a security administrator is logged in or not.
- **Alarm Interval**—The security device sends the audible alert message at regular intervals. The default interval is 10 seconds. You can set a maximum interval of 3600 seconds.
- **Alarm Overwrite**—The maximum number of alarm events the security device can store is 100. When a new alarm event occurs, the security device drops the newest security alarm by default. However, you can configure the security device to overwrite the oldest event in the log with the new event.

WebUI

Configuration > Admin > Audit Setting: Enter the following, then click Apply:

Use Alarm Security Audible: (select)
Use Alarm Security Local-Force: (select)
Use Alarm Security Overwrite: (select)
Alarm Security Interval (seconds): 10

CLI

set alarm security audible
set alarm security local-force
unset alarm security overwrite disable
set alarm security interval *number*

Viewing Security Alarms

You can use the following CLI commands to view the details of a single security alarm and statistics for all security alarms.

CLI

To view the details of a security alarm:

```
get alarm security ack-id number
```

To view statistics for all security alarms:

```
get alarm security statistics
```

The output of the **get alarm security** command displays the following elements:

- **TOTAL**—Total number of security alarms
- **ACTIVE**—Number of security alarms currently not acknowledged
- **ACKED**—Number of security alarms manually acknowledged by the security administrator
- **AUTO ACKED**—Number of security alarms auto-acknowledged
- **OVERWRITTEN**—Number of security alarms overwritten by new events
- **LOG EXCLUDED**—Number of security alarms excluded by the exclude rules

Acknowledging Security Alarms

When a security alarm is triggered, you can execute the **exec alarm security** command to acknowledge it:

```
exec alarm security ack-id number
```

All consoles—connected to the network—then display the acknowledged security alarm, and the alarm status becomes inactive. Once the security alarm is acknowledged (manually or automatically), it becomes inactive and an audit log is generated. The security administrator can retrieve the details of each acknowledged security alarm from the audit log.

You can use the **exec alarm security all** command to acknowledge all active security alarms.

```
exec alarm security all
```

Setting Potential-Violation Security Alarms

Juniper Networks security devices enable you to configure a set of rules for monitoring events, including thresholds for each event type. If any of the rules is violated, the security device triggers a potential-violation security alarm.

A potential-violation security alarm is triggered if any of the following events exceeds its threshold value:

- Authentication violations
- Policy violations

- Replays of security attributes
- Encryption failures
- Decryption failures
- Key-generation failures
- Cryptographic and noncryptographic module self-test failures
- Internet Key Exchange (IKE) Phase 1 authentication failures
- IKE Phase 2 authentication failures

Once an event exceeds its threshold value, the security device triggers a potential-violation security alarm. The default threshold value is 3 for any event.

NOTE: The potential-violation security alarm does not support IPv6 traffic.

Example: Configuring a Device to Trigger a Potential-Violation Alarm

In this example, you configure the security device to trigger an alarm when the number of encryption, decryption, and IKE Phase 1/Phase 2 failures exceeds the potential-violation threshold value of 5. You also configure the security device to trigger an alarm when the number of policy failures exceeds 5 per minute.

When any of these threshold values is exceeded, the security device triggers a security alarm and alerts the security administrator.

NOTE: You must use the CLI to configure potential-violation security alarms.

CLI

```
set alarm security potential-violation encryption-failures 5
set alarm security potential-violation decryption-failures 5
set alarm security potential-violation ike-p1-failures 5
set alarm security potential-violation ike-p2-failures 5
set alarm security potential-violation policy-violation rate 5 per minute
save
```

For more information about potential-violation commands, see the *ScreenOS CLI Reference Guide: IPv4 Command Descriptions*.

Configuring Exclude Rules

You can set rules to exclude some audit logs from being generated. You must have security administrator privileges to include or exclude auditable events from the audit log.

You can configure exclude rules based on the following attributes:

- **Exclude ID**—Identity of the exclude rule
- **User ID**—Identity of the authenticated user
- **Event Type**—Log type or event type
- **Network Addresses**—Source IP address, destination IP address, and destination port

By default, no exclude rule is set and the security device generates all logs. You cannot set more than 10 exclude rules. However, you can modify the existing rules according to the requirements of your network. Excluded security alarms are not generated in the audit log.

Example: Setting an Exclude Rule to Exclude an Event for the Audit Log

In this example, you—as the root admin—configure an exclude rule to prevent a failure event from being generated in the audit log.

You can use the WebUI or the CLI to configure an exclude rule.

WebUI

Configuration > Admin > Exclude Rules: Enter the following, then click **Add**:

Rule ID: 1
User ID: admin
Event Type: 2
Source IP address: 2.2.2.0
Destination IP Address: 3.3.3.0
Destination Port: 80
Event Result: Failure (select)

The Configured Exclude Rules table displays all exclude rules configured on the security device.

CLI

```
set log exclude-id 1 user-id sam event-type 2 src-ip 2.2.2.0 dst-ip 3.3.3.0 dst-port
80 failure
```

To view the configured exclude rules:

```
get log exclude
```

The security device displays all active exclude rules that you have configured.

Syslog

A security device can generate syslog messages for system events at predefined severity levels (see the list of severity levels in “Syslog” on page 392) and, optionally, for traffic that policies permit across a firewall. It sends these messages to up to four

designated syslog hosts running on UNIX and Linux systems. For each syslog host, you can specify the following:

- Whether the security device includes traffic log entries or event log entries, or both.
- Whether to send traffic through a VPN tunnel to the syslog server and—if through a VPN tunnel—which interface to use as the source interface (see "Example: Self-Generated Traffic Through a Route-Based Tunnel" on page 408 and "Example: Self-Generated Traffic Through a Policy-Based Tunnel" on page 415).
- The port to which the security device sends syslog messages.
- The security facility, which classifies and sends emergency and alert level messages to the syslog host; and the regular facility, which classifies and sends all other messages for events unrelated to security.

By default, the security device sends messages to syslog hosts with User Datagram Protocol (UDP) on port 514. To increase the reliability of the message delivery, you can change the transport protocol for each syslog host to Transmission Control Protocol (TCP) on port 514.

You can use syslog messages to create email alerts for the system administrator or to display messages on the console of the designated host using UNIX syslog conventions.

NOTE: On UNIX and Linux platforms, modify the /etc/rc.d/init.d/syslog file so that the syslog retrieves information from the remote source (syslog -r).

Enabling Syslog on Backup Devices

In an Active/Passive NSRP configuration where one device acts as a primary and the other as its backup, only the primary device sends all the syslog messages to the syslog server. Whereas the backup devices in an NSRP cluster send only the event log messages. To allow an administrator to effectively monitor the backup devices, all the syslog messages need to be backed up on the syslog server. Hence in the current release, ScreenOS allows you to configure the backup devices to send all the syslog messages to the syslog server. This configuration is synced to the primary device in an NSRP Active/Passive cluster. You can verify the status of the syslog backup on a backup device by issuing the CLI command **get syslog backup**. By default, syslog backup is disabled.

To enable the backup device to send all the syslog messages:

WebUI

Configuration > Report Settings > Syslog: Enter the following, then click **Apply**:

Enable syslog backup: Select this option to send logs from the backup device to the specified syslog servers.

CLI

```
set syslog backup enable
save
```

Example: Enabling Multiple Syslog Servers

In this example, you configure the security device to send event and traffic logs with TCP to three syslog servers at the following IP addresses/port numbers:

- 1.1.1.1/1514
- 2.2.2.1/2514
- 3.3.3.1/3514

You set both the security and facility levels to **Local0**.

WebUI

Configuration > Report Settings > Syslog: Enter the following, then click **Apply**:

Enable syslog messages: Select this option to send logs to the specified syslog servers.
No.: Select 1, 2, and 3 to indicate you are adding 3 syslog servers.
IP/Hostname: 1.1.1.1, 2.2.2.1, 3.3.3.1
Port: 1514, 2514, 3514
Security Facility: Local0, Local0, Local0
Facility: Local0, Local0, Local0
Event Log: (select)
Traffic Log: (select)
TCP: (select)

CLI

```
set syslog config 1.1.1.1 port 1514
set syslog config 1.1.1.1 log all
set syslog config 1.1.1.1 facilities local0 local0
set syslog config 1.1.1.1 transport tcp
set syslog config 2.2.2.1 port 2514
set syslog config 2.2.2.1 log all
set syslog config 2.2.2.1 facilities local0 local0
set syslog config 2.2.2.1 transport tcp
set syslog config 3.3.3.1 port 3514
set syslog config 3.3.3.1 log all
set syslog config 3.3.3.1 facilities local0 local0
set syslog config 2.2.2.1 transport tcp
set syslog enable
save
```

WebTrends

NetIQ offers a product called the WebTrends Firewall Suite that allows you to create customized reports based on WELF logs generated by the security device. You can customize the reports to display the information you want in the format you specify. The security device sends the WELF logs to the WebTrends server through a source interface. You can create reports on all events and severity levels or focus on an area such as firewall attacks. (For additional information on WebTrends, see the WebTrends product documentation.)

WebTrends supports three kinds of logs: event, traffic, and IDP. However, you can view IDP logs only if the security device supports IDP.

If you enable backup for the logs, you can send them to multiple WebTrends servers (the maximum is 4). To send a log to a WebTrends server, you must specify the IP address or hostname of the server along with the destination port.

By default, WELF logs are sent using UDP, but you can also use TCP. If required, you can manually reset the IP connections.

The following table lists the log types along with the required heading prefix with which it must be sent.

Table 43: WELF Logs

Log Category	Log Types	Heading Prefix
Event Log	Configuration log	[Config Change]
Event Log	URL Filter Detection log	[URL filtering]
Event Log	AntiVirus Detection log	[AntiVirus]
Event Log	Antispam Detection log	[AntiSpam]
IDP Log	IPS/DI Detection log	[IPS/DI]
Event Log	Screen Attack log	[Attack]

You can also send WebTrends messages through a VPN tunnel. In the WebUI, use the **Use Trust Zone Interface as Source IP for VPN** option. In the CLI, use the **set webtrends vpn** command.

In the following example, you send an event log to the WebTrends host (172.10.16.25) through port 514.

WebUI

1. **WebTrends Settings**

   Configuration > Report Settings > WebTrends: Enter the following, then click **Apply**:

   Enable WebTrends Messages: (select)
   Use Trust Zone Interface as Source IP for VPN: (select)
   Enable WebTrends Backup: (select)
   Source Interface: (select)
   Enable: (select)
   IP/Hostname: 172.10.16.25
   Port: 514
   Event Log: (select)
   Traffic Log:
   IDP Log:
   TCP: (select)
   Reconnect:

2. **Severity Levels**

   Configuration > Report Settings > Log Settings: Enter the following, then click **Apply**:

   WebTrends Notification: (select)

NOTE: When you enable WebTrends on a security device running in transparent mode, you must set up a static route. See "Static Routing" on page 1221.

CLI

3. **WebTrends Settings**

   set webtrends VPN
   set webtrends enable
   set webtrends src-interface *interface-name*
   set webtrends config *IP address/host-name*
   set webtrends config *IP address/host-name* log { all | event | idp | traffic }
   set webtrends config*IP address/host-name* port *port_num*
   set webtrends config *IP address/host-name* transport tcp
   set webtrends backup enable
   exec webtrends reconnect *IP address/hostname*

4. **Severity Levels**

   set log module system level notification destination webtrends
   save

Simple Network Management Protocol

The Simple Network Management Protocol (SNMP) agent for the Juniper Networks security device provides network administrators with a way to view statistical data about the network and the devices on it and to receive notification of system events of interest.

Juniper Networks security devices support the SNMPv1, the SNMPv2c, and the SNMPv3 protocols, all described in the RFCs show in Table 44 on page 397:

Table 44: RFC List

Version	RFC List
SNMPv1	■ RFC1157, *A Simple Network Management Protocol*
SNMPv2c	■ RFC-1901, *Introduction to Community-based SNMPv2* ■ RFC-1905, *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)* ■ RFC-1906, *Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)*
SNMPv3	■ RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks* ■ RFC 3412, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)* ■ RFC 3413, *Simple Network Management Protocol (SNMP) Applications* ■ RFC 3414, *User-based Security Model (USM) for version 3 of Simple Network Management Protocol (SNMP)* ■ RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Managemetn Protocol (SNMP)* ■ RFC 3417, *Transport Mapping for the Simple Network Management Protocol (SNMP)* ■ RFC 3584, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*

Security devices also support all relevant Management Information Base II (MIB II) groups defined in RFC-1213, *Management Information Base for Network Management of TCP/IP-based Internets: MIB-II*. The devices also have private enterprise MIB files, which you can load into an SNMP MIB browser.

NOTE: Using an SNMP MIB browser, you can check the CPU, memory usage, and session usage counts on both the ScreenOS and the Intrusion Detection and Prevention (IDP) security modules.

The Juniper Networks SNMP agent generates the following traps (notifications) when specified events or conditions occur:

- **Cold Start Trap**: The security device generates a cold start trap when it becomes operational after you power it on.
- **Trap for SNMP Authentication Failure**: The SNMP agent in the security device triggers the authentication failure trap if someone attempts to connect to it using an incorrect SNMP community string or if the IP address of the host attempting the connection is not defined in an SNMP community. (This option is enabled by default.)
- **Traps for System Alarms**: Security device error conditions and firewall conditions trigger system alarms. Three enterprise traps are defined to cover alarms related to hardware, security, and software. (For more information on firewall settings and alarms, see "ICMP Fragments" on page 697 and "Traffic Alarms" on page 385.)
- **Traps for Traffic Alarms**: Traffic alarms are triggered when traffic exceeds the alarm thresholds set in policies. (For more information on configuring policies, see "Policies" on page 197.)

Table 45 on page 398 lists possible alarm types and their associated trap numbers.

Table 45: Trap Alarm Types

Trap Enterprise ID	Description
100	Hardware problems
200	Firewall problems
300	Software problems
400	Traffic problems
500	VPN problems
600	NSRP problems
800	DRP problems
900	Interface failover problems
1000	Firewall attacks

NOTE: The network administrator must have an SNMP manager application such as HP OpenView or SunNet Manager to browse the SNMP MIB II data and to receive traps from either the trusted or untrusted interface. Shareware and freeware SNMP manager applications are available from the Internet.

SNMPv1 and SNMPv2c

Security devices are not shipped with a default configuration for the SNMP manager. To configure your security device for SNMP, you must first create communities, define their associated hosts, and assign permissions (read/write or read-only).

When you create an SNMP community, you can specify whether the community supports SNMPv1, SNMPv2c, or both SNMP versions, as required by the SNMP management stations. (For backward compatibility with earlier ScreenOS releases that only support SNMPv1, security devices support SNMPv1 by default.) If an SNMP community supports both SNMP versions, you must specify a trap version for each community member.

For security reasons, an SNMP community member with read/write privileges can change only the following variables on a security device:

- **sysContact**—Contact information for the security device admin in the event that the SNMP admin needs to contact the admin for the security device. This can be the security admin's name, email address, telephone number, office location, or a combination of such information.
- **sysLocation**—The physical location of the security device. This can be the name of a country, city, building, or its exact location on a rack in a network operations center (NOC).
- **sysName**—The name that SNMP administrators use for the security device. By convention, this is a fully qualified domain name (FQDN), but it can be something else.
- **snmpEnableAuthenTraps**—This enables or disables the ability of the SNMP agent in the security device to generate a trap whenever someone attempts to contact the SNMP agent with an incorrect SNMP community name.
- **ipDefaultTTL**—The default value inserted into the time-to-live (TTL) field in the IP header of datagrams originating from the security device whenever the Transport Layer protocol does not supply a TTL value.
- **ipForwarding**—This indicates whether or not the security device forwards traffic—other than that destined for the security device itself. By default, the security device indicates that it does not forward traffic.

SNMPv3

Security devices are not shipped with a default configuration for SNMPv3. To configure your security device for SNMPv3 , you must first create a unique engine ID to identify an SNMP entity and a user-based security model (USM) with the respective privilege and password.

When you create a USM user, you can specify the authentication type (MD5, SHA, or None). The authentication type is used to compute identical message digests for the same block of data. It requires a password and uses Data Encryption Standard (DES) to encrypt and decrypt the SNMPv3 packets.

SNMPv1 and SNMPv2c Implementation Overview

Juniper Networks has implemented SNMP in its devices in the following ways:

- SNMP administrators are grouped in SNMP communities. A device can support up to three communities, with up to eight members in each community.
- A community member can be either a single host or a subnet of hosts, depending on the netmask you use when defining the member. By default, the security device assigns an SNMP community member with a 32-bit netmask (255.255.255.255), which defines it as a single host.
- If you define an SNMP community member as a subnet, any device on that subnet can poll the security device for SNMP MIB information. However, the security device cannot send an SNMP trap to a subnet, only to an individual host.
- Each community has either read-only or read/write permission for the MIB II data.
- Each community can support SNMPv1, SNMPv2c, or both. If a community supports both versions of SNMP, you must specify a trap version for each community member.
- You can allow or deny receipt of traps for each community.
- You can access the MIB II data and traps through any physical interface.
- Each system alarm (a system event classified with a severity level of critical, alert, or emergency) generates a single enterprise SNMP trap to each of the hosts in each community that is set to receive traps.
- The security device sends Cold Start/Link Up/Link Down traps to all hosts in communities that you set to receive traps.
- If you specify trap-on for a community, you also have the option to allow traffic alarms.
- You can send SNMP messages through a route-based or policy-based VPN tunnel. For more information, see "Configuring a MIB Filter in the SNMP Community" on page 402.

SNMPv3 Implementation Overview

Juniper Networks has implemented SNMPv3 in its devices in the following ways:

- An administrator can define a USM user in the SNMPv3 framework. A device can support up to 32 users.
- The security device administrator can create the view-based access control model (VACM) views. Each view is tagged with an object identifier and mask values. A device can support up to 32 VACM views.
- The security device administrator can create a VACM access group. For each access group you can define the security model, security level, and the privilege access. A device can support up to 32 VACM access groups.
- The security device administrator can create a group to map an SNMPv3 USM user, an SNMPv1 community, or an SNMPv2c community, in combination with a VACM access group.
- The security device administrator can configure an SNMPv1 or SNMPv2c community under an SNMPv3 framework. In an SNMPv3 community, the administrator can enforce access control to SNMPv1 or SNMPv2c requests as well. Each community has a tag name.

- The administrator can configure the trap details such as filters, target parameters, and target address.
- The SNMPv3 filters are configured by an administrator. A device can support up to 32 SNMPv3 filters.
- Each filter is attached to a target (host).
- The target parameter is used when sending a trap to a target. The administrator can configure an SNMPv3 target parameter. A device can support up to 32 target parameters.
- Each target has an IPv4 or IPv6 address/netmask.
- You can enter both the IPv4 and IPv6 addresses. The system sends the trap to the target if the mask is 32 for IPv4 addresses or 128 for IPv6 addresses.
- Each target has a trap port and a tag.
- You can specify 8 tags to a target.

Defining a Read/Write SNMP Community

In this example, you create an SNMP community named MAge11. You assign it read/write privileges and enable its members to receive MIB II data and traps. It has the members 1.1.1.5/32 and 1.1.1.6/32. Each of these members has an SNMP manager application running a different version of SNMP: SNMPv1 and SNMPv2c. The community name functions as a password and needs to be protected.

You provide contact information for the local admin of the security device in case an SNMP community member needs to contact him—name: al_baker@mage.com. You also provide the location of the security device—location: 3-15-2. These numbers indicate that the device is on the third floor, in the fifteenth row, in the second position in that row.

You also enable the SNMP agent to generate traps whenever someone illegally attempts an SNMP connection to the security device. Authentication failure traps is a global setting that applies to all SNMP communities and is disabled by default.

Finally, you enable SNMP manageability on ethernet1, an interface that you have previously bound to the Trust zone. This is the interface through which the SNMP manager application communicates with the SNMP agent in the security device.

WebUI

Configuration > Report Settings > SNMP: Enter the following settings, then click **Apply**:

System Contact: al_baker@mage.com
Location: 3-15-2
Enable Authentication Fail Trap: (select)

Configuration > Report Settings > SNMP > New Community: Enter the following settings, then click **OK**:

Community Name: MAge11

Permissions:
Write: (select)
Trap: (select)
Including Traffic Alarms: (clear)
Version: ANY (select)
Hosts IP Address/Netmask and Trap Version:
1.1.1.5/32 v1
1.1.1.6/32 v2c

Network > Interfaces > Edit (for ethernet1): Enter the following settings, then click **OK**:

Service Options:
Management Services: SNMP

CLI

```
set snmp contact al_baker@mage.com
set snmp location 3-15-2
set snmp auth-trap enable
set snmp community MAge11 read-write trap-on version any
set snmp host Mage 1.1.1.5/32 trap v1
set snmp host Mage 1.1.1.6/32 trap v2
set interface ethernet1 manage snmp
save
```

Configuring a MIB Filter in the SNMP Community

An SNMP agent can encounter problems when different devices from the same management domain report conflicting IP address ranges. This issue is widely observed in NAT deployments where the tendency is to use similar IP address ranges in the private domain. However, by configuring a MIB filter in the SNMP community, you can now resolve the issue of overlapping addresses by filtering conflicting private IP addresses that exist in the same domain.

As root admin, you can configure the MIB filter to either include or exclude an IP address entry from the address list of the following MIB tables:

- atTable
- ipAddrTable
- ipNetToMediaTable
- ipRouteTable

When you configure the MIB filter to exclude an IP address, all IP addresses are included except the one being filtered. Likewise, if you configure the MIB filter to include an IP address, only the filtered address is included in the MIB table and all other IP addresses are excluded. Each MIB filter you configure should be bound to its related SNMP community.

NOTE: You cannot configure a MIB filter to both include and exclude IP addresses at the same time.

Example

In this example, you (as the root admin) configure a MIB filter filter-private-address to exclude private network IP addresses 192.168.0.0/16 and 10.10.0.0/16. The MIB filter is bound to the SNMP community **comm_vzb**and the SNMP host IP address is 202.100.1.1/32.

You can use the WebUI and the CLI to create and bind a MIB filter to the SNMP community.

WebUI

Configuration > Report Settings > SNMP > MIB Filter Edit: Enter the following, then click **Apply**:

MIB Filter Name: filter-private-address
Type: IP (select)
Action: Exclude (select)

Configuration > Report Settings > SNMP > MIB Filter Edit: Enter the following, then click **Add**:

IP: 192.168.0.0
Netmask: 255.255.0.0
IP: 10.10.0.0
Netmask: 255.255.0.0

Configuration > Report Setting > SNMP > Community Edit: Enter the following, then click **OK**:

Community Name: comm_vzb
Permission:
Write: (select)
Trap: (select)
Including Traffic Alarms: (clear)
Version: ANY (select)
MIB Filter: filter_private_address (select)
Host IP Address: 202.100.1.1
Netmask: 255.255.255.255
Trap: V2C

CLI

1. **Creating a MIB Filter:**

   set snmp mib-filter name *filter-private-addres*s type ip action exclude

2. **Adding an Entry to the MIB Filter:**

```
set snmp mib-filter filter-private-address ip 192.168.0.0/255.255.0.0
set snmp mib-filter filter-private-address ip 10.10.0.0/255.255.0.0
```

3. **Binding a MIB Filter to the SNMP Community:**

```
set snmp community comm_vzb read-write version any
set snmp community comm_vzb mib-filter filter-private-address
```

4. **Assigning a Host to the SNMP Community:**

```
set snmp host comm_vzb 202.100.1.1/255.255.255.255 trap v2
save
```

Example: Configuring an SNMPv3 packet

In this example, you (as the root admin) configure an SNMPv3 packet.

WebUI

1. **Engine-ID**

NOTE: Local engine ID configuration is optional. A local-engine ID is to identify an SNMP entity. By default, the serial number of the device is assigned as the value of the local engine ID.

Configuration > Report Settings > SNMPv3: Enter the following settings, then click **Apply**:

Local-engine id: netscreen

2. **USM User**

Configuration > Report Settings > SNMPv3 > USM User > New User: Enter the following settings, then click **OK**:

User Name: netscreen
Authentication Type: (select)
Authentication Password: netscreen
Privacy Protocol: (select)
Privacy Password: netscreen

3. **View**

Configuration > Report Settings > SNMPv3 > VACM > New View: Enter the following settings, then click **OK**:

View Name: test-view

Configuration > Report Settings > SNMPv3 > VACM > View Databse Edit: Enter the following settings, then click **Add**:

Subtree OID: .1

Subtree Mask: FF
Type: (select)

4. **Access Group**

   Configuration > Report Settings > SNMPv3 > VACM > New Access Group: Enter the following settings, then click **OK**:

   Group Name: test-grp
   Security Model: (select)
   Security Level: (select)
   Read View: (select)
   Write View: (select)
   Notification View: (select)

5. **Group Mapping**

   Configuration > Report Settings > SNMPv3 > VACM > New Sec-to-group Mapping: Enter the following settings, then click **OK**:

   Security Model: (select)
   User Name: (select)
   Community: (read only)
   Group Name: (select)

6. **Community**

NOTE: The community name must be unique.

   Configuration > Report Settings > SNMPv3 > Community > New Community: Enter the following settings, then click **OK**:

   Community Name: public
   Tag: public

7. **Trap**

   Configuration > Report Settings > SNMPv3 > Trap > New Filter: Enter the following settings, then click **OK**:

   Filter Name: test-filter

   Configuration > Report Settings > SNMPv3 > Trap > Filter Database Edit: Enter the following settings, then click **Add**:

   Subtree OID: .1
   Subtree Mask: FF
   Type: (select)

8. **Target Parameter**

   Configuration > Report Settings > SNMPv3 > Trap > New Target Parameter: Enter the following settings, then click **OK**:

Target Parameter Name: test-param
Filter Name: (select)
Security Model: (select)
Security Level: (select)
User Name: (select)

9. **Target Address**

   Configuration > Report Settings > SNMPv3 > Trap > New Target Address: Enter the following settings, then click **OK**:

   Target Name: test-target
   Target IPv4 Address/Netmask: 192.168.1.1/32
   Trap Port: 162
   Target Parameter: (select)
   Taglist: (select)

CLI

1. **Engine ID**

   ```
   set snmpv3 local-engine id netscreen
   ```

2. **USM User**

   ```
   set snmpv3 user netscreen auth md5 auth-pass netscreen priv des priv-pass
     netscreen
   ```

3. **View**

   ```
   set snmpv3 view name test-view
   set snmpv3 view test-view oid .1 mask FF type include
   ```

4. **Access Group**

   ```
   set snmpv3 access group test-grp sec-model usm sec-level priv read test-view
   ```

5. **Group Mapping**

   ```
   set snmpv3 group-mapping sec-model usm user netscreen group test-grp
   ```

6. **Community**

   ```
   set snmpv3 community public tag public
   ```

7. **Trap**

   ```
   set snmpv3 filter name test-filter
   set snmpv3 filter test-filter oid .1 mask FF type include
   ```

8. **Target Parameter**

   ```
   set snmpv3 target-param test-param filter test-filter sec-model usm sec-level priv
     user netscreen
   ```

9. **Target Address**

```
set snmpv3 target test-target address 192.168.1.1/32 port 162 target-param
  test-param
```

VPN Tunnels for Self-Initiated Traffic

You can use virtual private network (VPN) tunnels to secure remote monitoring of a security device from a fixed IP address. Using a VPN tunnel, you can protect traffic addressed to and initiated from a security device. Types of traffic initiated from a security device can include NetScreen-Global PRO reports, event log entries sent to syslog and WebTrends servers, and SNMP MIB traps.

Juniper Networks security devices support two types of VPN tunnel configurations:

- **Route-Based VPNs**: The security device uses route table entries to direct traffic to tunnel interfaces, which are bound to VPN tunnels.

  To send traffic such as event log entries, NetScreen-Global PRO reports, or SNMP traps generated by the security device through a route-based VPN tunnel, you must manually enter a route to the proper destination. The route must point to the tunnel interface that is bound to the VPN tunnel through which you want the security device to direct the traffic. No policy is required.

- **Policy-Based VPNs**: The security device uses the VPN tunnel names specifically referenced in policies to direct traffic through VPN tunnels.

  To send self-initiated traffic through a policy-based VPN tunnel, you must include the source and destination addresses in the policy. For the source address, use the IP address of an interface on the security device. For the destination address, use the IP address of the storage server or SNMP community member's workstation, if it is located behind a remote security device. If the remote SNMP community member uses the NetScreen-Remote VPN client to make VPN connections to the local security device, use an internal IP address defined on the NetScreen-Remote as the destination address.

If either the remote gateway or the end entity has a dynamically assigned IP address, then the security device cannot initiate the formation of a VPN tunnel because these addresses cannot be predetermined, and thus you cannot define routes to them. In such cases, the remote host must initiate the VPN connection. After either a policy-based or route-based VPN tunnel is established, both ends of the tunnel can initiate traffic if policies permit it. Also, for a route-based VPN, there must be a route to the end entity through a tunnel interface bound to the VPN tunnel—either because you manually entered the route or because the local security device received the route through the exchange of dynamic routing messages after a tunnel was established. (For information about dynamic routing protocols, see *"Routing" on page 1219*.) You can also use VPN monitoring with the rekey option or IKE heartbeats to ensure that once the tunnel is established, it remains up regardless of VPN activity. (For more information about these options, see"VPN Monitoring" on page 971 and "Monitoring Mechanisms" on page 1027.)

For each VPN tunnel configuration type, you can use any of the following types of VPN tunnel:

- **Manual Key**: You manually set the three elements that define a Security Association (SA) at both ends of the tunnel: a Security Parameters Index (SPI), an encryption key, and an authentication key. To change any element in the SA, you must manually enter it at both ends of the tunnel.
- **AutoKey IKE with Pre-shared Key**: One or two pre-shared secrets—one for authentication and one for encryption—function as seed values. Using them, the IKE protocol generates a set of symmetrical keys at both ends of the tunnel; that is, the same key is used to encrypt and decrypt. At predetermined intervals, these keys are automatically regenerated.
- **AutoKey IKE with Certificates**: Using the Public Key Infrastructure (PKI), the participants at both ends of the tunnel use a digital certificate (for authentication) and an RSA public/private key pair (for encryption). The encryption is asymmetrical; that is, one key in a pair is used to encrypt and the other to decrypt.

NOTE: For a complete description of VPN tunnels, see *"Virtual Private Networks" on page 705*. For more information on NetScreen-Remote, see the *NetScreen-Remote VPN Client Administrator Guide*.

Example: Self-Generated Traffic Through a Route-Based Tunnel

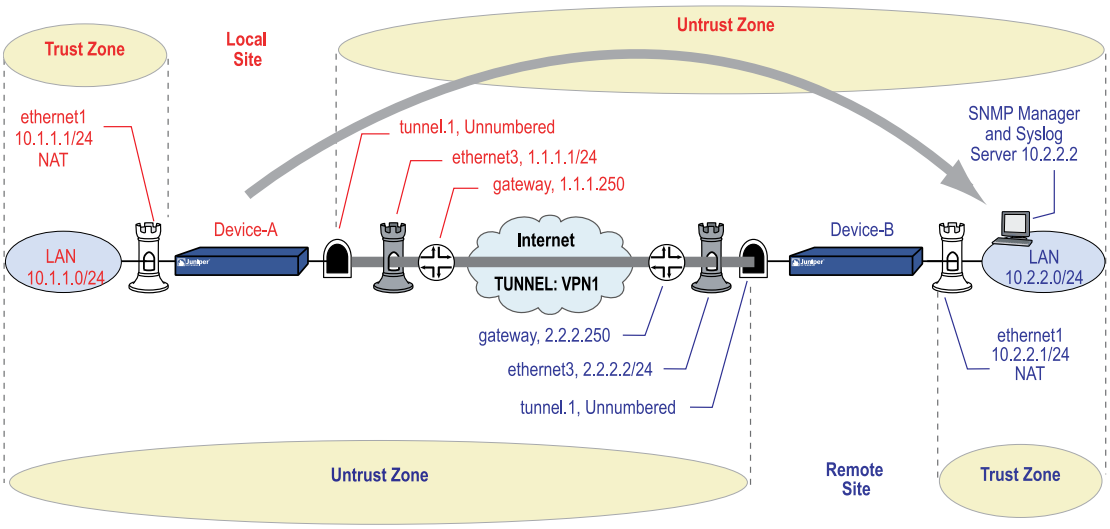
Figure 96 on page 409 illustrates an example in which you configure a local security device (Device-A) to send SNMPv1 MIB traps and syslog reports through a route-based AutoKey IKE VPN tunnel to an SNMP community member behind a remote security device (Device-B). The tunnel uses a pre-shared key (Ci5y0a1aAG) for data origin authentication and the security level predefined as "Compatible" for both Phase 1 and Phase 2 proposals. You, as the local admin for Device-A, create the tunnel.1 interface and bind it to vpn1. You and the admin for Device-B define the proxy IDs as shown in Table 46 on page 408.

Table 46: Proxy IDs for Route-Based Tunnel

Device-A		Device-B	
Local IP	10.1.1.1/32	Local IP	10.2.2.2/32
Remote IP	10.2.2.2/32	Remote IP	10.1.1.1/32
Service	Any	Service	Any

You bind ethernet1 to the Trust zone and ethernet3 to the Untrust zone. The default gateway IP address is 1.1.1.250. All zones are in the trust-vr routing domain.

Figure 96: Traffic Through a Route-Based Tunnel



The remote admin for Device-B uses similar settings to define that end of the AutoKey IKE VPN tunnel so that the pre-shared key, proposals, and proxy IDs match.

You also configure an SNMP community named "remote_admin" with read/write privileges, and you enable the community to receive traps. You define the host at 10.2.2.2/32 as a community member.

NOTE: This example assumes that the remote admin has already set up the syslog server and SNMP manager application that supports SNMPv1. When the remote admin sets up the VPN tunnel on his security device, he uses 1.1.1.1 as the remote gateway and 10.1.1.1 as the destination address.

WebUI (Device-A)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT (select)

NOTE: By default, any interface that you bind to the Trust zone is in NAT mode. Consequently, this option is already enabled for interfaces bound to the Trust zone. When the remote admin configures the SNMP manager, he must enter **10.1.1.1** in the Remote SNMP Agent field. This is the address to which the SNMP manager sends queries.

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24
Service Options:
Management Services: SNMP

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet1(trust-vr)

2. **Syslog and SNMP**

Configuration > Report Settings > Syslog: Enter the following, then click **Apply**:

Enable Syslog Messages: (select)
No.: Select 1 to indicate you are adding 1 syslog server.
IP / Hostname: 10.2.2.2
Port: 514
Security Facility: auth/sec
Facility: Local0

Configuration > Report Settings > SNMP > New Community: Enter the following, then click **OK**:

Community Name: remote_admin
Permissions:
Write: (select)
Trap: (select)
Including Traffic Alarms: (clear)
Version: V1
Hosts IP Address/Netmask: 10.2.2.2/32 V1
Trap Version: V1

3. **VPN**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: to_admin
Type: Static IP, Address/Hostname: 2.2.2.2
Preshared Key: Ci5y0a1aAG

Security Level: Compatible
Outgoing interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface: (select), tunnel.1
Proxy-ID: (select)
Local IP/Netmask: 10.1.1.1/32
Remote IP/Netmask: 10.2.2.2/32
Service: ANY

4. **Routes**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.2/32
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: (select) 1.1.1.250

CLI (Device-A)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 manage snmp
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet1
```

NOTE: When the remote admin configures the SNMP manager, he must enter 10.1.1.1 in the Remote SNMP Agent field. This is the address to which the SNMP manager sends queries.
By default, any interface that you bind to the Trust zone is in NAT mode. Consequently, this option is already enabled for interfaces bound to the Trust zone.

2. **VPN**

```
set ike gateway to_admin address 2.2.2.2 outgoing-interface ethernet3 preshare
 Ci5y0a1aAG sec-level compatible
set vpn vpn1 gateway to_admin sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 10.1.1.1/32 remote-ip 10.2.2.2/32 any
```

3. **Syslog and SNMP**

```
set syslog config 10.2.2.2 auth/sec local0
set syslog enable
set snmp community remote_admin read-write trap-on version v1
set snmp host remote_admin 10.2.2.2/32
```

4. **Routes**

```
set vrouter trust-vr route 10.2.2.2/32 interface tunnel.1
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
save
```

WebUI (Device-B)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.2.2.1/24

Select the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 2.2.2.2/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet1(trust-vr)

2. **Addresses**

Policy > Policy Elements > Addresses > Lists > New: Enter the following, then click **OK**:

Address Name: addr1
IP Address/Domain Name: IP/Netmask: 10.2.2.2/32

Zone: Trust

Policy > Policy Elements > Addresses > Lists > New: Enter the following, then click **OK**:

Address Name: ns-a
IP Address/Domain Name: IP/Netmask: 10.1.1.1/32
Zone: Untrust

3. **Service Group**

Policy > Policy Elements > Services > Groups > New: Enter the following group name, move the following services, then click **OK**:

Group Name: s-grp1

Select **Syslog** and use the < < button to move the service from the Available Members column to the Group Members column.

Select **SNMP** and use the < < button to move the service from the Available Members column to the Group Members column.

4. **VPN**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: to_admin
Type: Static IP, Address/Hostname: 1.1.1.1
Preshared Key: Ci5y0a1aAG
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click Return to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface: (select), tunnel.1
Proxy-ID: (select)
Local IP/Netmask: 10.2.2.2/32
Remote IP/Netmask: 10.1.1.1/32
Service: Any

5. **Routes**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask:10.1.1.1/32
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: (select) 2.2.2.250

6. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), addr1
Destination Address:
Address Book Entry: (select), ns-a
Service: s-grp1
Action: Permit
Position at Top: (select)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), ns-a
Destination Address:
Address Book Entry: (select), addr1
Service: s-grp1
Action: Permit
Position at Top: (select)

CLI (Device-B)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet1
```

2. **Addresses**

```
set address trust addr1 10.2.2.2/32
set address untrust ns-a 10.1.1.1/32
```

3. **Service Group**

```
set group service s-grp1
set group service s-grp1 add syslog
set group service s-grp1 add snmp
```

4. **VPN**

```
set ike gateway to_admin address 1.1.1.1 outgoing-interface ethernet3 preshare
 Ci5y0a1aAG sec-level compatible
set vpn vpn1 gateway to_admin sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 10.2.2.2/32 remote-ip 10.1.1.1/32 any
```

5. **Routes**

```
set vrouter trust-vr route 10.1.1.1/32 interface tunnel.1
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
```

6. **Policies**

```
set policy top from trust to untrust addr1 ns-a s-grp1 permit
set policy top from untrust to trust ns-a addr1 s-grp1 permit
save
```

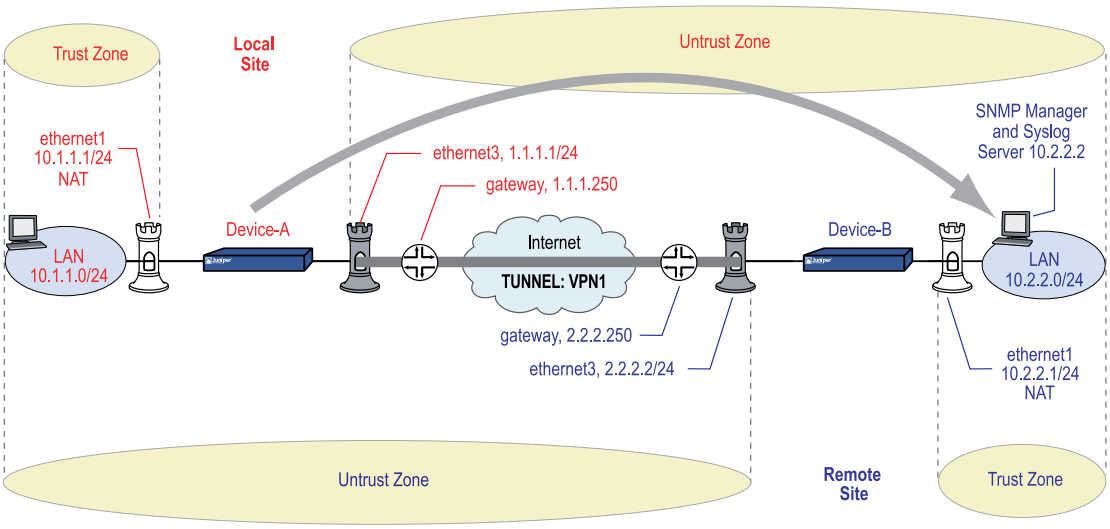
Example: Self-Generated Traffic Through a Policy-Based Tunnel

In this example (illustrated in Figure 97 on page 416), you configure a local security device (Device-A) to send SNMPv2c MIB traps and syslog reports through a policy-based AutoKey IKE VPN tunnel (vpn1) to an SNMP community member behind a remote security device (Device-B). The tunnel uses a preshared key (Ci5y0a1aAG) for data origin authentication and the security level predefined as “ompatible” for both Phase 1 and Phase 2 proposals.

NOTE: This example assumes that the remote admin has already set up the syslog server and an SNMP manager application that supports SNMPv2c. When the remote admin sets up the VPN tunnel on his security device, he uses 1.1.1.1 as the remote gateway and 10.1.1.1 as the destination address.

Both you and the remote admin bind ethernet1 to the Trust zone, and ethernet3 to the Untrust zone on Device-A and Device-B. The default gateway IP address for Device-A is 1.1.1.250. The default gateway IP address for Device-B is 2.2.2.250. All zones are in the trust-vr routing domain.

Figure 97: Traffic Through a Policy-Based Tunnel



You also configure an SNMP community named “remote_admin” with read/write privileges, and you enable the community to receive traps. You define the host at 10.2.2.2/32 as a community member.

The inbound and outbound policies on Device-A match the outbound and inbound policies on Device-B. The addresses and service used in the policies are as follows:

- 10.1.1.1/32, the address of the Trust zone interface on Device-A
- 10.2.2.2/32, the address of the host for the SNMP community member and syslog server
- Service group named “s-grp1,” which contains SNMP and syslog services

From the policies that you and the admin for Device-B create, the two security devices derive the following proxy IDs for vpn1:

Table 47: Proxy IDs for Policy-Based Tunnel

Device-A		Device-B	
Local IP	10.1.1.1/32	Local IP	10.2.2.2/32
Remote IP	10.2.2.2/32	Remote IP	10.1.1.1/32
Service	Any	Service	Any

NOTE: The security device treats a service group as “any” in proxy IDs.

WebUI (Device-A)

1. **Interfaces—Security Zones**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT (select)

NOTE: When the remote admin configures the SNMP manager, he must enter **10.1.1.1** in the Remote SNMP Agent field. This is the address to which the SNMP manager sends queries.
By default, any interface that you bind to the Trust zone is in NAT mode. Consequently, this option is already enabled for interfaces bound to the Trust zone.

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24
   Service Options:
   Management Services: SNMP

2. **Addresses**

   Policy > Policy Elements > Addresses > Lists > New: Enter the following, then click **OK**:

   Address Name: trust_int
   IP Address/Domain Name:
   IP/Netmask: 10.1.1.1/32
   Zone: Trust

   Policy > Policy Elements > Addresses > Lists > New: Enter the following, then click **OK**:

   Address Name: remote_admin
   IP Address/Domain Name:
   IP/Netmask: 10.2.2.2/32
   Zone: Untrust

3. **Service Group**

   Policy > Policy Elements > Services > Groups > New: Enter the following group name, move the following services, then click **OK**:

   Group Name: s-grp1

Select **Syslog** and use the < < button to move the service from the Available Members column to the Group Members column.

Select **SNMP** and use the < < button to move the service from the Available Members column to the Group Members column.

4. **VPN**

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: vpn1
   Security Level: Compatible
   Remote Gateway: Create a Simple Gateway: (select)
   Gateway Name: to_admin
   Type: Static IP, Address/Hostname: 2.2.2.2
   Preshared Key: Ci5y0a1aAG
   Security Level: Compatible
   Outgoing Interface: ethernet3

5. **Syslog and SNMP**

   Configuration > Report Settings > Syslog: Enter the following, then click **Apply**:

   Enable Syslog Messages: (select)
   Source Interface: ethernet1
   No.: Select 1 to indicate you are adding 1 syslog server.
   IP/Hostname: 10.2.2.2
   Port: 514
   Security Facility: auth/sec
   Facility: Local0

   Configuration > Report Settings > SNMP > New Community: Enter the following, then click **OK**:

   Community Name: remote_admin
   Permissions:
   Write: (select)
   Trap: (select)
   Including Traffic Alarms: (clear)
   Version: V2C
   Hosts IP Address/Netmask: 10.2.2.2/32 V2C
   Trap Version: V2C
   Source Interface:
   ethernet1 (select)

   Configuration > Report Settings > SNMP: Enter the following, then click **Apply**:

   Enable Authentication Fail Trap: (select)

6. **Route**

   Network > Routing > Destination > trust-vr New: Enter the following, then click OK:

   Network Address/Netmask: 0.0.0.0/0
   Gateway: (select)

Interface: ethernet3
Gateway IP Address: 1.1.1.250

7. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), trust_int
Destination Address:
Address Book Entry: (select), remote_admin
Service: s-grp1
Action: Tunnel
Tunnel VPN: vpn1
Modify matching outgoing VPN policy: (select)
Position at Top: (select)

CLI (Device-A)

1. **Interfaces—Security Zones**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 manage snmp
```

NOTE: By default, any interface that you bind to the Trust zone is in NAT mode. Consequently, this option is already enabled for interfaces bound to the Trust zone.

2. **Addresses**

```
set address trust trust_int 10.1.1.1/32
set address untrust remote_admin 10.2.2.2/32
```

3. **Service Group**

```
set group service s-grp1
set group service s-grp1 add syslog
set group service s-grp1 add snmp
```

4. **VPN**

```
set ike gateway to_admin address 2.2.2.2 outgoing-interface ethernet3 preshare
 Ci5y0a1aAG sec-level compatible
set vpn vpn1 gateway to_admin sec-level compatible
```

5. **Syslog and SNMP**

```
set syslog config 10.2.2.2 auth/sec local0
set syslog src-interface ethernet1
set syslog enable
```

```
set snmp community remote_admin read-write trap-on version v2c
set snmp host remote_admin 10.2.2.2/32 src-interface ethernet1
```

6. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

7. **Policies**

```
set policy top from trust to untrust trust_int remote_admin s-grp1 tunnel vpn
vpn1
set policy top from untrust to trust remote_admin trust_int s-grp1 tunnel vpn
vpn1
save
```

WebUI (Device-B)

1. **Interfaces—Security Zones**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.2.2.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 2.2.2.2/24

2. **Addresses**

   Policy > Policy Elements > Addresses > Lists > New: Enter the following, then click **OK**:

   Address Name: addr1
   IP Address/Domain Name:
   IP/Netmask: 10.2.2.2/32
   Zone: Trust

   Policy > Policy Elements > Addresses > Lists > New: Enter the following, then click **OK**:

   Address Name: ns-a
   IP Address/Domain Name:
   IP/Netmask: 10.1.1.1/32
   Zone: Untrust

3. **Service Groups**

Policy > Policy Elements > Services > Group: Enter the following group name, move the following services, then click **OK**:

Group Name: s-grp1

Select **Syslog** and use the < < button to move the service from the Available Members column to the Group Members column.

Select **SNMP** and use the < < button to move the service from the Available Members column to the Group Members column.

4. **VPN**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: to_admin
Type: Static IP, IP Address: 1.1.1.1
Preshared Key: Ci5y0a1aAG
Security Level: Compatible
Outgoing interface: ethernet3

5. **Route**

Network > Routing > Routing Table > trust-vr New: Enter the following, then click OK:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: (select) 2.2.2.250

6. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), addr1
Destination Address:
Address Book Entry: (select), ns-a
Service: s-grp1
Action: Tunnel
Tunnel VPN: vpn1
Modify matching outgoing VPN policy: (select)
Position at Top: (select)

CLI (Device-B)

1. **Interfaces—Security Zones**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
```

```
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
```

2. **Addresses**

```
set address trust addr1 10.2.2.2/32
set address untrust ns-a 10.1.1.1/32
```

3. **Service Group**

```
set group service s-grp1
set group service s-grp1 add syslog
set group service s-grp1 add snmp
```

4. **VPN**

```
set ike gateway to_admin address 1.1.1.1 outgoing-interface ethernet3 preshare
 Ci5y0a1sec-level compatible
set vpn vpn1 gateway to_admin sec-level compatible
```

5. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
```

6. **Policies**

```
set policy top from trust to untrust addr1 ns-a s-grp1 tunnel vpn vpn1
set policy top from untrust to trust ns-a addr1 s-grp1 tunnel vpn vpn1
save
```

Viewing Screen Counters

Juniper Networks security devices provide screen, hardware, and flow counters for monitoring traffic. Counters give processing information for specified zones and interfaces, and help you to verify configurations for desired policies.

Table 48 on page 422 shows the screen counters for monitoring general firewall behavior and for viewing the amount of traffic affected by specified policies.

Table 48: Screen Counters

Counter	Description
Bad IP Option Protection	Number of frames discarded because of malformed or incomplete IP options
Dst IP-based session limiting	Number of sessions dropped after the session threshold was reached
FIN bit with no ACK bit	Number of packets detected and dropped with an illegal combination of flags
Fragmented packet protection	Number of blocked IP packet fragments

Table 48: Screen Counters *(continued)*

Counter	Description
HTTP Component Blocked	Number of blocked packets with HTTP components
HTTP Component Blocking for ActiveX controls	Number of ActiveX components blocked
HTTP Component Blocking for .exe files	Number of blocked HTTP packets with .exe files
HTTP Component Blocking for Java applets	Number of blocked Java components
HTTP Component Blocking for .zip files	Number of blocked HTTP packets with .zip files
ICMP Flood Protection	Number of ICMP packets blocked as part of an ICMP flood
ICMP Fragment	Number of ICMP frames with the More Fragments flag set, or with offset indicated in the Offset field
IP Spoofing Attack Protection	Number of IP addresses blocked as part of an IP spoofing attack
IP Sweep Protection	Number of IP sweep attack packets detected and blocked
Land Attack Protection	Number of packets blocked as part of a suspected land attack
Large ICMP Packet	Number of ICMP frames detected with an IP length greater than 1024
Limit Session	Number of undeliverable packets because the session limit had been reached
Loose Src Route IP Option	Number of IP packets detected with the Loose Source Route option enabled
Malicious URL Protection	Number of suspected malicious URLs blocked
Ping-of-Death Protection	Number of suspected and rejected ICMP packets that are oversized or of an irregular size
Port Scan Protection	Number of port scans detected and blocked
Record Route IP Option	Number of frames detected with the Record Route option enabled
Security IP Option	Number of frames discarded with the IP Security option set
Src IP-based session limiting	Number of sessions dropped after the session threshold was reached
Source Route IP Option Filter	Number of IP source routes filtered
Stream IP Option	Number of packets discarded with the IP Stream identifier set
Strict Src Route IP Option	Number of packets detected with the Strict Source Route option enabled

Table 48: Screen Counters *(continued)*

Counter	Description
SYN-ACK-ACK-Proxy DoS	Number of blocked packets because of the SYN-ACK-ACK-proxy DoS SCREEN option
SYN and FIN bits set	Number of packets detected with an illegal combination of flags
SYN Flood Protection	Number of SYN packets detected as part of a suspected SYN flood
SYN Fragment Detection	Number of packet fragments dropped as part of a suspected SYN fragments attack
Timestamp IP Option	Number of IP packets discarded with the Internet Timestamp option set
TCP Packet without Flag	Number of illegal packets dropped with missing or malformed flags field
Teardrop Attack Protection	Number of packets blocked as part of a Teardrop attack
UDP Flood Protection	Number of UDP packets dropped as part of a suspected UDP flood
Unknown Protocol Protection	Number of packets blocked as part of an unknown protocol
WinNuke Attack Protection	Number of packets detected as part of a suspected WinNuke attack

Table 49 on page 424 shows the hardware counters for monitoring hardware performance and packets with errors.

Table 49: Hardware Counters

Counter	Description
drop vlan	Number of packets dropped because of missing VLAN tags, an undefined sub-interface, or because VLAN trunking was not enabled when the security device was in transparent mode
early frame	Number of counters used in an Ethernet driver buffer descriptor management
in align err	Number of incoming packets with an alignment error in the bit stream
in bytes	Number of bytes received
in coll err	Number of incoming collision packets
in crc err	Number of incoming packets with a cyclic redundancy check (CRC) error
in dma err	Number of incoming packets with a Direct Memory Access (DMA) error
in misc err	Number of incoming packets with a miscellaneous error

Table 49: Hardware Counters *(continued)*

Counter	Description
in no buffer	Number of unreceived packets because of unavailable buffers
in overrun	Number of transmitted overrun packets
in packets	Number of packets received
in short frame	Number of incoming packets with an Ethernet frame shorter than 64 bytes (including the frame checksum)
in underrun	Number of transmitted underrun packets
late frame	Number of counters used in an Ethernet driver buffer descriptor management
out bs pak	Number of packets held in back store while searching for an unknown MAC address When the security device forwards a packet, it first checks if the destination MAC address is in the ARP table. If it cannot find the destination MAC in the ARP table, the security device sends an ARP request to the network. If the security device receives another packet with the same destination MAC address before it receives a reply to the first ARP request, it increases the out bs pak counter by one.
out bytes	Number of bytes sent
out coll err	Number of outgoing collision packets
out cs lost	Number of dropped outgoing packets because the Carrier Sense Multiple Access/Collision Detect (CSMA/CD) protocol lost the signal
out defer	Number of deferred outgoing packets
out discard	Number of discarded outgoing packets
out heartbeat	Number of outgoing heartbeat packets
out misc err	Number of outgoing packets with a miscellaneous error
out no buffer	Number of unsent packets because of unavailable buffers
out packets	Number of packets sent
re xmt limit	Number of dropped packets when the retransmission limit was exceeded while an interface was operating at half-duplex

Table 50 on page 425 shows the flow counters for monitoring the number of packets inspected at the flow level.

Table 50: Flow Counters

Counter	Description
address spoof	Number of suspected address spoofing attack packets received

Table 50: Flow Counters *(continued)*

Counter	Description
auth deny	Number of times user authentication was denied
auth fail	Number of times user authentication failed
big bkstr	Number of packets that are too big to buffer in the ARP back store while waiting for MAC-to-IP address resolution
connections	Number of sessions established since the last boot
encrypt fail	Number of failed Point-to-Point Tunneling Protocol (PPTP) packets
***icmp broadcast**	Number of ICMP broadcasts received
icmp flood	Number of ICMP packets that are counted toward the ICMP flood threshold
illegal pak	Number of packets dropped because they do not conform to the protocol standards
in arp req	Number of incoming arp request packets
in arp resp	Number of outgoing arp request packets
in bytes	Number of bytes received
in icmp	Number of Internet Control Message Protocol (ICMP) packets received
in other	Number of incoming packets that are of a different Ethernet type
in packets	Number of packets received
in self	Number of packets addressed to the Management IP address
***in un auth**	Number of unauthorized incoming TCP, UDP, and ICMP packets
***in unk prot**	Number of incoming packets using an unknown Ethernet protocol
in vlan	Number of incoming vlan packets
in vpn	Number of IPsec packets received
invalid zone	Number of packets destined for an invalid security zone
ip sweep	Number of packets received and discarded beyond the specified ip sweep threshold
land attack	Number of suspected land attack packets received
loopback drop	Number of packets dropped because they cannot be looped back through the security device. An example of a loopback session is when a host in the Trust zone sends traffic to a MIP or VIP address that is mapped to a server that is also in the Trust zone. The security device creates a loopback session that directs such traffic from the host to the MIP or VIP server.
mac relearn	Number of times that the MAC address learning table had to relearn the interface associated with a MAC address because the location of the MAC address changed

Table 50: Flow Counters *(continued)*

Counter	Description
mac tbl full	Number of times that the MAC address learning table completely filled up
mal url	Number of blocked packets destined for a URL determined to be malicious
***misc prot**	Number of packets using a protocol other than TCP, UDP, or ICMP
mp fail	Number of times a problem occurred when sending a PCI message between the master processor module and the processor module
no conn	Number of packets dropped because of unavailable Network Address Translation (NAT) connections
no dip	Number of packets dropped because of unavailable Dynamic IP (DIP) addresses
no frag netpak	Number of times that the available space in the netpak buffer fell below 70%
***no frag sess**	The number of times that fragmented sessions were greater than half of the maximum number of NAT sessions
no g-parent	Number of packets dropped because the parent connection could not be found
no gate	Number of packets dropped because no gate was available
no gate sess	Number of terminated sessions because there were no gates in the firewall for them
no map	Number of packets dropped because there was no map to the trusted side
no nat vector	Number of packets dropped because the Network Address Translation (NAT) connection was unavailable for the gate
***no nsp tunnel**	Number of dropped packets sent to a tunnel interface to which no VPN tunnel is bound
no route	Number of unroutable packets received
no sa	The number of packets dropped because no Security Associations (SA) was defined
no sa policy	Number of packets dropped because no policy was associated with an SA
***no xmit vpnf**	Number of dropped VPN packets due to fragmentation
null zone	Number of dropped packets erroneously sent to an interface bound to the Null zone
nvec err	Number of packets dropped because of NAT vector error
out bytes	Number of bytes sent
out packets	Number of packets sent
out vlan	Number of outgoing vlan packets
ping of death	Number of suspected Ping of Death attack packets received

Table 50: Flow Counters *(continued)*

Counter	Description
policy deny	Number of packets denied by a defined policy
port scan	Number of packets that are counted as a port scan attempt
proc sess	Number of times that the total number of sessions on a processor module exceeded the maximum threshold
sa inactive	Number of packets dropped because of an inactive SA
sa policy deny	Number of packets denied by an SA policy
sessn thresh	the threshold for the maximum number of sessions
***slow mac**	Number of frames whose MAC addresses were slow to resolve
src route	Number of packets dropped because of the filter source route option
syn frag	Number of dropped SYN packets because of a fragmentation
tcp out of seq	Number of TCP segments received whose sequence number is outside the acceptable range
tcp proxy	Number of packets dropped from using a TCP proxy such as the SYN flood protection option or user authentication
teardrop	Number of packets blocked as part of a suspected Teardrop attack
tiny frag	Number of tiny fragmented packets received
trmn drop	Number of packets dropped by traffic management
trmng queue	Number of packets waiting in the queue
udp flood	Number of UDP packets that are counted toward the UDP flood threshold
url block	Number of HTTP requests that were blocked
winnuke	Number of WinNuke attack packets received
wrong intf	Number of session creation messages sent from a processor module to the master processor module
wrong slot	Number of packets erroneously sent to the wrong processor module

NOTE: For more information about the Carrier Sense Multiple Access/Collision Detect (CSMA/CD) protocol, see the IEEE 802.3 standard available at http://standards.ieee.org.

In this example, you view the device screen counters for the Trust zone.

WebUI

Reports > Counters > Zone Screen: Select **Trust** from the Zone drop-down list.

CLI

```
get counter screen zone trust
```

Part 4
Attack Detection and Defense Mechanisms

Attack Detection and Defense Mechanisms describes the Juniper Networks security options available in ScreenOS. You can enable many of these options at the security zone level. These options apply to traffic reaching the Juniper Networks security device through any interface bound to a zone for which you have enabled such options. These options offer protection against IP address and port scans, denial of service (DoS) attacks, and other kinds of malicious activity. You can apply other network security options, such as Web filtering, antivirus checking, and intrusion detection and prevention (IDP), at the policy level. These options only apply to traffic under the jurisdiction of the policies in which they are enabled.

NOTE: The subject of policies is presented only peripherally in this guide, as it applies to the network security options that you can enable at the policy level. For a complete examination of policies, see "Policies" on page 197.

This guide contains the following sections:

- "Protecting a Network" on page 433 outlines the basic stages of an attack and the firewall options available to combat the attacker at each stage.
- "Reconnaissance Deterrence" on page 439 describes the options available for blocking IP address sweeps, port scans, and attempts to discover the type of operating system (OS) of a targeted system.
- "Denial of Service Attack Defenses" on page 463 explains firewall, network, and OS-specific DoS attacks and how ScreenOS mitigates such attacks.
- "Content Monitoring and Filtering" on page 495 describes how to protect users from malicious uniform resource locators (URLs) and how to configure the Juniper Networks security device to work with third-party products to provide antivirus scanning, antispam, and Web filtering.
- "Deep Inspection" on page 559 describes how to configure the Juniper Networks security device to obtain Deep Inspection (DI) attack object updates, how to create user-defined attack objects and attack object groups, and how to apply DI at the policy level.
- "Intrusion Detection and Prevention" on page 615 describes Juniper Networks Intrusion Detection and Prevention (IDP) technology, which can both detect and stop attacks when deployed inline to your network. The chapter describes how to apply IDP at the policy level to drop malicious packets or connections before the attacks can enter your network.

- “Suspicious Packet Attributes” on page 697 presents several SCREEN options that protect network resources from potential attacks indicated by unusual IP and ICMP packet attributes.
- “Contexts for User-Defined Signatures” on page 2263provides descriptions of contexts that you can specify when defining a stateful signature attack object.

Chapter 12

Protecting a Network

There can be many reasons for invading a protected network. The following list contains some common objectives:

- Gathering the following kinds of information about the protected network:
  - Topology
  - IP addresses of active hosts
  - Numbers of active ports on active hosts
  - Operating systems of active hosts
- Overwhelming a host on a protected network with bogus traffic to induce a denial of service (DoS)
- Overwhelming the protected network with bogus traffic to induce a network-wide DoS
- Overwhelming a firewall with bogus traffic to induce a denial of service (DoS) for the network behind it
- Causing damage to and stealing data from a host on a protected network
- Gaining access to a host on a protected network to obtain information
- Gaining control of a host to launch other exploits
- Gaining control of a firewall to control access to the network that it protects

ScreenOS provides detective and defensive tools for uncovering and thwarting the efforts of attackers to achieve the above objectives when they attempt to target a network protected by a Juniper Networks security device.

This chapter presents an overview of the main stages of an attack and the various defense mechanisms that you can employ to thwart an attack at each stage:

Stages of an Attack

Each attack typically progresses in two major stages. In the first stage, the attacker gathers information, and in the second stage he or she launches the attack.

1. Perform reconnaissance.
   a. Map the network and determine which hosts are active (IP address sweep).
   b. Discern which ports are active (port scans) on the hosts discovered by the IP address sweep.
   c. Determine the operating system (OS), which might expose a weakness in the OS or suggest an attack to which that particular OS is susceptible.
2. Launch the attack.
   a. Conceal the origin of the attack.
   b. Perform the attack.
   c. Remove or hide evidence.

Detection and Defense Mechanisms

An exploit can be an information-gathering probe or an attack to compromise, disable, or harm a network or network resource. In some cases, the distinction between the two objectives of an exploit can be unclear. For example, a barrage of TCP SYN segments might be an IP address sweep with the intent of triggering responses from active hosts, or it might be a SYN flood attack with the intent of overwhelming a network so that it can no longer function properly. Furthermore, because an attacker usually precedes an attack by performing reconnaissance on the target, we can consider information-gathering efforts as a precursor to an impending attack—that is, they constitute the first stage of an attack. Thus, the term *exploit* encompasses both reconnaissance and attack activities, and the distinction between the two is not always clear.

Juniper Networks provides various detection methods and defense mechanisms at the zone and policy levels to combat exploits at all stages of their execution:

- SCREEN options at the zone level
- Firewall policies at the inter-, intra-, and super-zone policy levels (*super-zone* here means in global policies, where no security zones are referenced).

NOTE: Although the VLAN and MGT zones are function zones and not security zones, you can set SCREEN options for them. The VLAN zone supports the same set of SCREEN options as a Layer 3 security zone. (Layer 2 security zones support an additional SYN flood option that Layer 3 zones do not: Drop Unknown MAC). Because the following SCREEN options do not apply to the MGT zone, they are not available for that zone: SYN flood protection, SYN-ACK-ACK proxy flood protection, HTTP component blocking, and WinNuke attack protection.

To secure all connection attempts, Juniper Networks security devices use a dynamic packet-filtering method known as stateful inspection. Using this method, the security device notes various components in the IP packet and TCP segment headers— source and destination IP addresses, source and destination port numbers, and packet sequence numbers—and maintains the state of each TCP session and pseudo UDP session traversing the firewall. (The device also modifies session states based on changing elements such as dynamic port changes or session termination.) When a responding TCP packet arrives, the device compares the information reported in its header with the state of its associated session stored in the inspection table. If they match, the responding packet is allowed to pass the firewall. If the two do not match, the packet is dropped.

ScreenOS SCREEN options secure a zone by inspecting, then allowing or denying, all connection attempts that require crossing an interface bound to that zone. The security device then applies firewall policies, which can contain content filtering and intrusion detection and prevention (IDP) components, to the traffic that passes the SCREEN filters.

A Juniper Networks firewall provides the following sets of defense mechanisms:

- Reconnaissance deterrence
  - IP address sweep
  - Port scanning
  - Operating system probes
  - Evasion techniques
- Content monitoring and filtering
  - Fragment reassembly
  - Antivirus scanning
  - Antispam filtering
  - Web filtering
- Deep inspection
  - Stateful signatures
  - Protocol anomalies
  - Granular blocking of HTTP components
- Denial of service (DoS) attack defenses
  - Firewall DoS attacks
    - Session table flood
    - SYN-ACK-ACK proxy flood
  - Network DoS attacks
    - SYN flood
    - ICMP flood

        - UDP flood
    - OS-specific DoS attacks
        - Ping of death
        - Teardrop attack
        - WinNuke
- Suspicious packet attributes
    - ICMP fragments
    - Large ICMP packets
    - Bad IP options
    - Unknown protocols
    - IP packet fragments
    - SYN fragments

ScreenOS network-protection settings operate at two levels: security zone and policy. The Juniper Networks security device performs reconnaissance deterrence and DoS attack defenses at the security zone level. In the area of content monitoring and filtering, the security device applies fragment reassembly at the zone level and antivirus (AV) scanning and uniform resource locator (URL) filtering at the policy level. The device applies IDP at the policy level, except for the detection and blocking of HTTP components, which occurs at the zone level. Zone-level firewall settings are SCREEN options. A network protection option set in a policy is a component of that policy.

Exploit Monitoring

Although you typically want the security device to block exploits, there might be times when you want to gather intelligence about them. You might want to learn specifically about a particular exploit—to discover its intention, its sophistication, and possibly (if the attacker is careless or unsophisticated) its source.

If you want to gather information about an exploit, you can let it occur, monitor it, analyze it, perform forensics, and then respond according to a previously prepared incident response plan. You can instruct the security device to notify you of an exploit, but then, instead of taking action, you can have the device allow the exploit to transpire. You can then study what occurred and try to understand the attacker's methods, strategies, and objectives. Increased understanding of the threat to the network can then allow you to better fortify your defenses. Although a smart attacker can conceal his or her location and identity, you might be able to gather enough information to discover where the attack originated. You also might be able to estimate the attacker's capabilities. Gathering and analyzing this kind of information allows you to determine your response.

Example: Monitoring Attacks from the Untrust Zone

In this example, IP spoofing attacks from the Untrust zone have been occurring daily, usually between 21:00 and midnight. Instead of dropping the packets with the spoofed source IP addresses, you want the security device to notify you that the packets have arrived but allow them to pass, perhaps directing them to a honeypot (a decoy network server that is designed to lure attackers and then record their actions during an attack) that you have connected on the DMZ interface connection. At 20:55 PM, you change the firewall behavior from notification and rejection of packets belonging to a detected attack to notification and acceptance. When the attack occurs, you can then use the honeypot to monitor the attacker's activity after crossing the firewall. You might also work in cooperation with the upstream ISP to begin tracking the source of the packets back to their source.

WebUI

Screening > Screen (Zone: Untrust): Enter the following, then click **Apply**:

Generate Alarms without Dropping Packet: (select)
IP Address Spoof Protection: (select)

CLI

```
set zone untrust screen alarm-without-drop
set zone untrust screen ip-spoofing
save
```

NOTE: The alarm-without-drop option does not apply to the following:

- SYN-ACK-ACK proxy protection
- Source IP Based Session Limit
- Destination IP Based Session Limit
- Malicious URL protection

If this option is set, the device does not generate alarms and pass the packets. Instead, it drops or forwards the packet based on the inspection results.

Chapter 13

Reconnaissance Deterrence

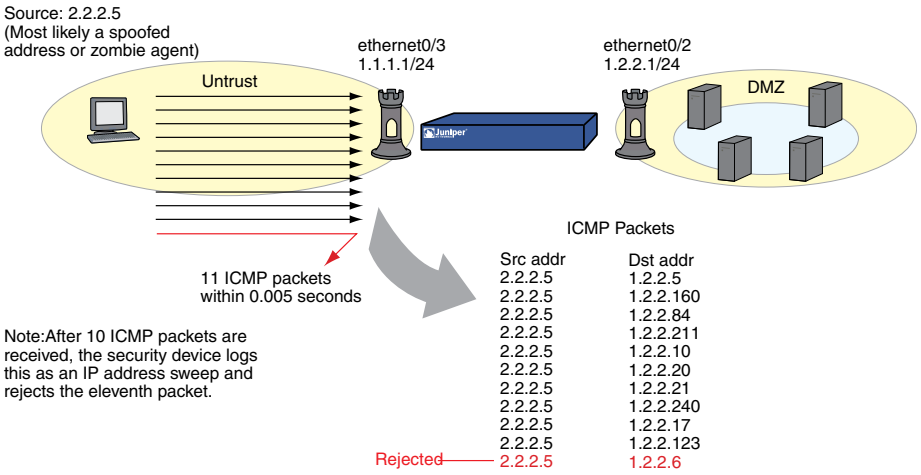
Attackers can better plan their attack when they first know the layout of the targeted network (which IP addresses have active hosts), the possible entry points (which port numbers are active on the active hosts), and the constitution of their victims (which operating system the active hosts are running). To gain this information, attackers must perform reconnaissance. Juniper Networks provides several SCREEN options to deter attackers' reconnaissance efforts and thereby hinder them from obtaining valuable information about the protected network and network resources.

IP Address Sweep

An address sweep occurs when one source IP address sends 10 ICMP packets to different hosts within a defined interval (5000 microseconds is the default). The purpose of this scheme is to send ICMP packets—typically echo requests—to various hosts in the hopes that at least one replies, thus uncovering an address to target. The security device internally logs the number of ICMP packets to different addresses from one remote source. Using the default settings, if a remote host sends ICMP traffic to 10 addresses in 0.005 seconds (5000 microseconds), the security device flags this as an address sweep attack, and rejects all further ICMP echo requests from that host for the remainder of the specified threshold time period. The device detects and drops the eleventh packet that meets the address sweep attack criterion. This is illustrated in Figure 98 on page 440.

In Figure 98 on page 440, the security device makes an entry in its session table for the first 10 ICMP packets from 2.2.2.5 and does a route lookup and policy lookup for these. If no policy permits these packets, the device tags these as invalid and removes them from the session table in the next "garbage sweep," which occurs every two seconds. After the eleventh packet, the device rejects all further ICMP traffic from 2.2.2.5.

Figure 98: Address Sweep



Consider enabling this SCREEN option for a security zone only if there is a policy permitting ICMP traffic from that zone. Otherwise, you do not need to enable it. The lack of such a policy denies all ICMP traffic from that zone, precluding an attacker from successfully performing an IP address sweep anyway.

To block IP address sweeps originating in a particular security zone:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

IP Address Sweep Protection: (select)
Threshold: (enter a value to trigger IP address sweep protection)

NOTE: The value unit is microseconds. The default value is 5000 microseconds.

CLI

```
set zone zone screen ip-sweep threshold number
set zone zone screen ip-sweep
```

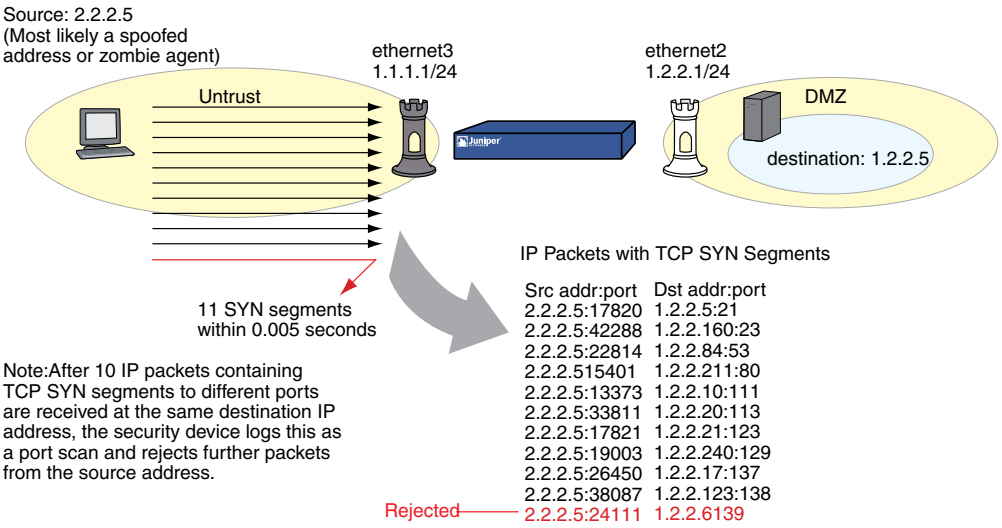
Port Scanning

A port scan occurs when one source IP address sends IP packets containing TCP SYN segments to 10 different ports at the same destination IP address within a defined interval (5000 microseconds is the default). The purpose of this scheme is to scan the available services in the hopes that at least one port will respond, thus identifying a service to target. The security device internally logs the number of different ports scanned from one remote source. Using the default settings, if a remote host scans 10 ports in 0.005 seconds (5000 microseconds), the device flags this as a port scan attack and rejects all further packets from the remote source for the remainder of

the specified timeout period. The device detects and drops the eleventh packet that meets the port scan attack criterion. This is illustrated in Figure 99 on page 441.

In Figure 99 on page 441, the security device makes an entry in its session table for the first 10 connection attempts from 2.2.2.5 to destination and does a route lookup and policy lookup for these. If no policy permits these connection attempts, the device tags these as invalid and removes them from the session table in the next "garbage sweep," which occurs every two seconds. After the eleventh attempt, the device rejects all further connection attempts.

Figure 99: Port Scan



To block port scans originating in a particular security zone:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

Port Scan Protection: (select)
Threshold: (enter a value to trigger protection against port scans)

NOTE: The value unit is microseconds. The default value is 5000 microseconds.

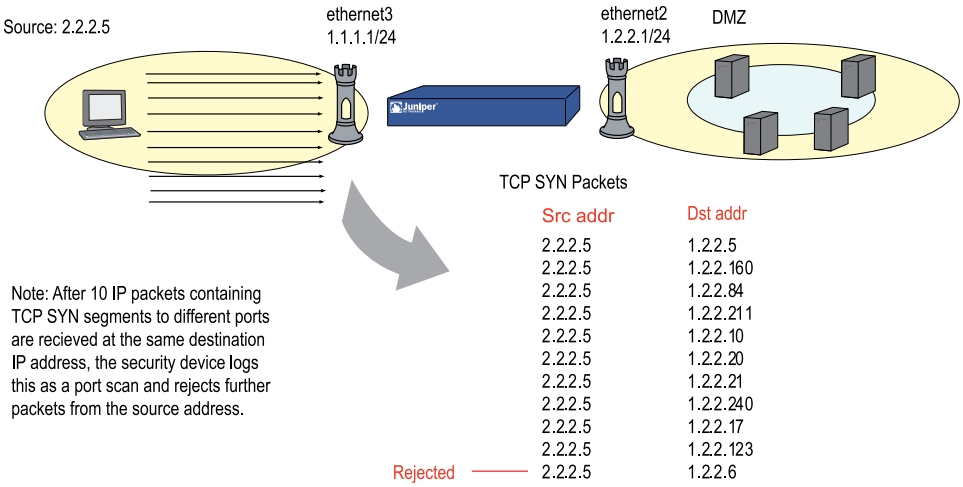
CLI

set zone *zone* screen port-scan threshold number
set zone *zone* screen port-scan

TCP/UDP Sweep Protection

In a TCP Sweep attack, an attacker sends TCP SYN packets to the target device as part of the TCP handshake. If the device responds to those packets, the attacker gets an indication that a port in the target device is open, which makes the port vulnerable to attack. Similarly, in a UDP Sweep attack, an attacker sends a UDP datagram to a UDP port. Depending on the reply, the attacker determines whether or not a port is open.

Figure 100: TCP/UDP Sweep Protection



To prevent these attacks, ScreenOS provides a TCP/UDP Sweep Protection SCREEN option at the security-zone level. This option limits the number of packets allowed from a source IP to a multiple IPs within a particular time frame. If the number of packets exceeds the threshold limit, the device does not establish the session.

The device maintains a source hash table for each initial packet destined for a different destination. The source hash table maintains a count of the number of attempts the source makes to reach each destination within a configured period.

If the rate of attacks from the source IP exceeds the configured threshold, the session-establishment attempts from that particular source IP are dropped and logged. The default threshold is 50 packets per second. You can enable or disable the TCP/UDP Sweep Protection SCREEN option and set the threshold rate with the WebUI or the CLI.

WebUI:

Security > Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply:**

TCP Sweep Protection: (select)
    Threshold: (enter a value to enable the TCP Sweep Protection SCREEN option)
UDP Sweep Protection: (select)

Threshold: (enter a value to enable the UDP Sweep Protection SCREEN option)

CLI:

To enable the TCP Sweep Protection SCREEN option:

set zone *zone-name* screen tcp-sweep

To set the threshold rate for TCP-SWEEP:

set zone *zone-name* screen tcp-sweep threshold *threshold rate*

To enable the UDP Sweep Protection SCREEN option and set the threshold rate:

set zone *zone-name* screen udp-sweep

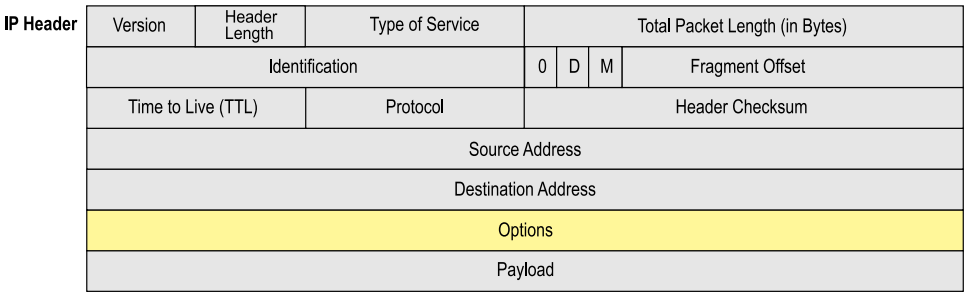
To set the threshold rate for UDP-SWEEP:

set zone *zone-name* screen udp-sweep threshold *threshold rate*

Network Reconnaissance Using IP Options

The Internet Protocol standard RFC 791, *Internet Protocol*, specifies a set of options to provide special routing controls, diagnostic tools, and security. These options appear after the destination address in an IP packet header, as shown in Figure 101 on page 443.

Figure 101: Routing Options



RFC 791 states that these options are “unnecessary for the most common communications” and, in reality, they rarely appear in IP packet headers. When they do appear, they are frequently being put to some illegitimate use. Table 51 on page 444 lists the IP options and their accompanying attributes.

Table 51: IP Options and Attributes

Type	Class	Number	Length	Intended Use	Nefarious Use
End of Options	0 Designed to provide extra packet or network control	0	0	Indicates the end of one or more IP options.	None.
No Options	0	1	0	Indicates there are no IP options in the header.	None.
Security	0	2	11 bits	Provides a way for hosts to send security, TCC (closed user group) parameters, and Handling Restriction Codes compatible with Department of Defense (DoD) requirements. (This option, as specified in RFC 791, *Internet Protocol,* and RFC 1038, *Revised IP Security Option,* is obsolete.)	Unknown. However, because it is obsolete, its presence in an IP header is suspect.
Loose Source Route	0	3	Varies	Specifies a partial route list for a packet to take on its journey from source to destination. The packet must proceed in the order of addresses specified, but it is allowed to pass through other routers in between those specified.	Evasion. The attacker can use the specified routes to hide the true source of a packet or to gain access to a protected network. (See "IP Source Route Options" on page 460.)
Record Route	0	7	Varies	Records the IP addresses of the network devices along the path that the IP packet travels. The destination machine can then extract and process the route information. (Due to the size limitation of 40 bytes for both the option and storage space, this can only record up to 9 IP addresses.)	Reconnaissance. If the destination host is a compromised machine in the attacker's control, he or she can glean information about the topology and addressing scheme of the network through which the packet passed.
Stream ID	0	8	4 bits	(Obsolete) Provided a way for the 16-bit SATNET stream identifier to be carried through networks that did not support the stream concept.	Unknown. However, because it is obsolete, its presence in an IP header is suspect.
Strict Source Route	0	9	Varies	Specifies the complete route list for a packet to take on its journey from source to destination. The last address in the list replaces the address in the destination field.	Evasion. An attacker can use the specified routes to hide the true source of a packet or to gain access to a protected network. (See "IP Source Route Options" on page 460.)

Table 51: IP Options and Attributes *(continued)*

Type	Class	Number	Length	Intended Use	Nefarious Use
Timestamp	2 Designed to provide diagnostics, debugging, and measurement.	4		Records the time (in Universal Time) when each network device receives the packet during its trip from the point of origin to its destination. The timestamp uses the number of milliseconds since midnight UT. The network devices are identified by IP number. This option develops a list of IP addresses of the routers along the path of the packet and the duration of transmission between each one.	Reconnaissance. If the destination host is a compromised machine in the attacker's control, he or she can glean information about the topology and addressing scheme of the network through which the packet passed.

The following SCREEN options detect IP options that an attacker can use for reconnaissance or for some unknown but suspect purpose:

- **Record Route:** The security device detects packets where the IP option is 7 (Record Route) and records the event in the SCREEN counters list for the ingress interface.
- **Timestamp:** The security device detects packets where the IP option list includes option 4 (Internet Timestamp) and records the event in the SCREEN counters list for the ingress interface.
- **Security:** The security device detects packets where the IP option is 2 (security) and records the event in the SCREEN counters list for the ingress interface.
- **Stream ID:** The security device detects packets where the IP option is 8 (Stream ID) and records the event in the SCREEN counters list for the ingress interface.

To detect packets with the above IP options set, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

IP Record Route Option Detection: (select)
IP Timestamp Option Detection: (select)
IP Security Option Detection: (select)
IP Stream Option Detection: (select)

CLI

set zone *zone* screen ip-record-route
set zone *zone* screen ip-timestamp-opt
set zone *zone* screen ip-security-opt
set zone *zone* screen ip-stream-opt

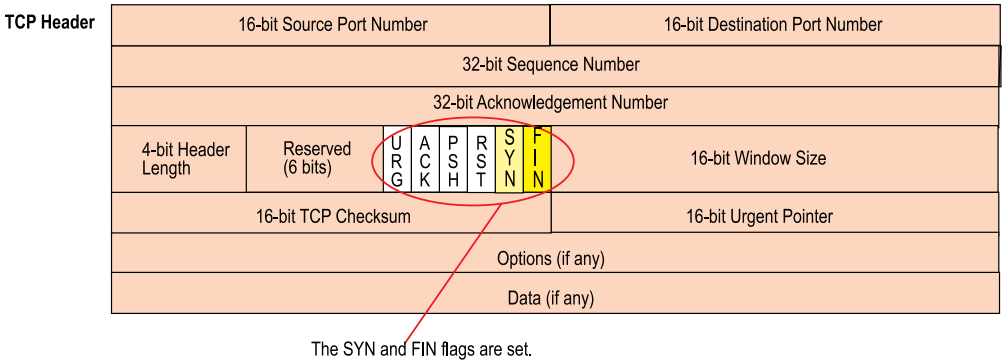
Operating System Probes

Before launching an exploit, an attacker might try to probe the targeted host to learn its operating system (OS). With that knowledge, he can better decide which attack to launch and which vulnerabilities to exploit. A Juniper Networks security device can block reconnaissance probes commonly used to gather information about OS types.

SYN and FIN Flags Set

Both the SYN and FIN control flags are not normally set in the same TCP segment header. The SYN flag synchronizes sequence numbers to initiate a TCP connection. The FIN flag indicates the end of data transmission to finish a TCP connection. Their purposes are mutually exclusive. A TCP header with the SYN and FIN flags set is anomalous TCP behavior, causing various responses from the recipient, depending on the OS. See Figure 102 on page 446.

Figure 102: TCP Header with SYN and FIN Flags Set



An attacker can send a segment with both flags set to see what kind of system reply is returned and thereby determine what kind of OS is on the receiving end. The attacker can then use any known system vulnerabilities for further attacks.

When you enable this SCREEN option, the security device checks if the SYN and FIN flags are set in TCP headers. If it discovers such a header, it drops the packet.

To block packets with both the SYN and FIN flags set, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **SYN and FIN Bits Set Protection**, then click **Apply**.

CLI

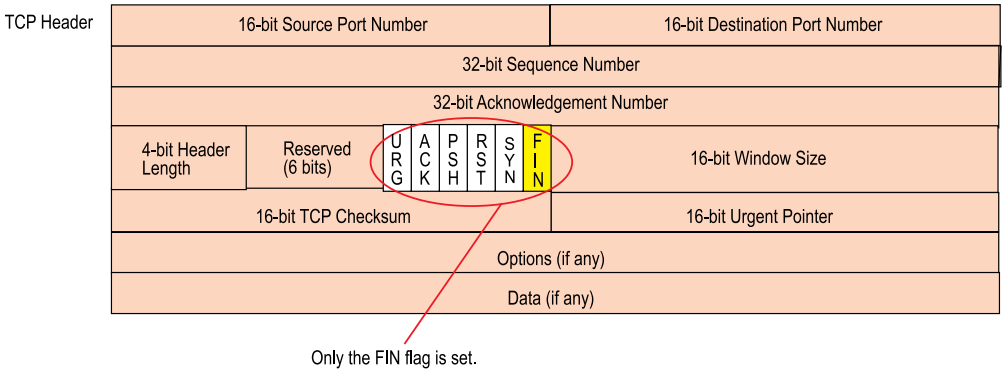
```
set zone zone screen syn-fin
```

FIN Flag Without ACK Flag

Figure 103 on page 447 shows TCP segments with the FIN control flag set (to signal the conclusion of a session and terminate the connection). Normally, TCP segments with the FIN flag set also have the ACK flag set (to acknowledge the previous packet received). Because a TCP header with the FIN flag set but not the ACK flag is anomalous TCP behavior, there is no uniform response to this. The OS might respond by sending a TCP segment with the RST flag set. Another might completely ignore it. The victim's response can provide the attacker with a clue as to its OS. (Other purposes for sending a TCP segment with the FIN flag set are to evade detection while performing address and port scans and to evade defenses on guard for a SYN flood by performing a FIN flood instead. For information about FIN scans, see "FIN Scan" on page 449.)

NOTE: Vendors have interpreted RFC 793, *Transmission Control Protocol*, variously when designing their TCP/IP implementations. When a TCP segment arrives with the FIN flag set but not the ACK flag, some implementations send RST segments. Some drop the packet without sending an RST.

Figure 103: TCP Header with FIN Flag Set



When you enable this SCREEN option, the security device checks if the FIN flag is set but not the ACK flag in TCP headers. If it discovers a packet with such a header, it drops the packet.

To block packets with the FIN flag set but not the ACK flag, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **FIN Bit with No ACK Bit in Flags Protection**, then click **Apply**.

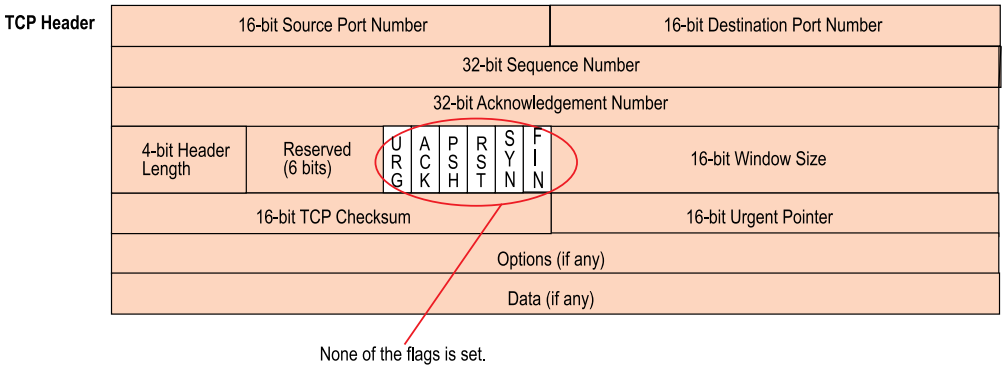
CLI

```
set zone zone screen fin-no-ack
```

TCP Header Without Flags Set

A normal TCP segment header has at least one flag control set. A TCP segment with no control flags set is an anomalous event. Because different operating systems respond differently to such anomalies, the response (or lack of response) from the targeted device can provide a clue as to the type of OS it is running. See Figure 104 on page 448.

Figure 104: TCP Header with No Flags Set



When you enable the security device to detect TCP segment headers with no flags set, the device drops all TCP packets with a missing or malformed flags field.

To block packets with no flags set, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **TCP Packet without Flag Protection**, then click **Apply**.

CLI

```
set zone zone screen tcp-no-flag
```

Evasion Techniques

Whether gathering information or launching an attack, it is generally expected that the attacker avoids detection. Although some IP address and port scans are blatant and easily detectable, more wily attackers use a variety of means to conceal their activity. Such techniques as using FIN scans instead of SYN scans—which attackers know most firewalls and intrusion detection programs detect—indicate an evolution of reconnaissance and exploit techniques to evade detection and successfully accomplish their tasks.

FIN Scan

A FIN scan sends TCP segments with the FIN flag set in an attempt to provoke a response (a TCP segment with the RST flag set) and thereby discover an active host or an active port on a host. An attacker might use this approach rather than perform an address sweep with ICMP echo requests or an address scan with SYN segments because he or she knows that many firewalls typically guard against the latter two approaches—but not necessarily against FIN segments. The use of TCP segments with the FIN flag set might evade detection and thereby help the attacker succeed in his or her reconnaissance efforts.

To thwart a FIN scan, you can do either or both of the following:

- Enable the SCREEN option that specifically blocks TCP segments with the FIN flag set but not the ACK flag, which is anomalous for a TCP segment:

  WebUI: Screening > Screen: Select the zone to which you want to apply this SCREEN option from the Zone drop-down list, then select **FIN Bit With No ACK Bit in Flags Protection**.

  CLI: Enter **set zone** name **screen fin-no-ack**, in which name is the name of the zone to which you want to apply this SCREEN option.

- Change the packet processing behavior to reject all non-SYN packets that do not belong to an existing session by entering the CLI command: **set flow tcp-syn-check**. (For more information about SYN flag checking, see "Non-SYN Flags" on page 449.)

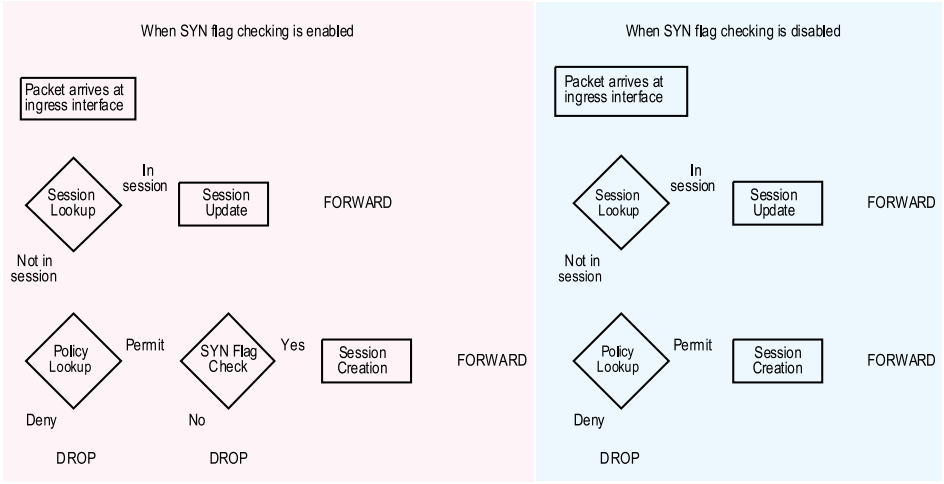
NOTE: Changing the packet flow to check that the SYN flag is set for packets that do not belong to existing sessions also thwarts other types of non-SYN scans, such as a null scan (when no TCP flags are set).

Non-SYN Flags

By default, the security device checks for SYN flags in the first packet of a session and rejects any TCP segments with non-SYN flags attempting to initiate a session. You can leave this packet flow as is or change it to so that the device does not enforce SYN flag checking before creating a session. Figure 105 on page 450 illustrates packet flow sequences when SYN flag checking is enabled and when it is disabled.

NOTE: By default, checking for the TCP SYN flag in the initial packet of a session is enabled when you install a Juniper Networks security device running ScreenOS 5.1.0 or higher. If you upgrade from a release prior to ScreenOS 5.1.0, SYN checking remains disabled by default—unless you have previously changed the default behavior. These packet flows are the same whether the ingress interface is operating at Layer 3 (route or NAT mode) or at Layer 2 (transparent mode).

Figure 105: SYN Flag Checking


When SYN flag checking is enabled
Packet arrives at ingress interface
Session Lookup
In session
Session Update
FORWARD
Not in session
Policy Lookup
Permit
SYN Flag Check
Yes
Session Creation
FORWARD
Deny
No
DROP
DROP
When SYN flag checking is disabled
Packet arrives at ingress interface
Session Lookup
In session
Session Update
FORWARD
Not in session
Policy Lookup
Permit
Session Creation
FORWARD
Deny
DROP

When the security device with SYN flag checking enabled receives a non-SYN TCP segment that does not belong to an existing session, it drops the packet and sends the source host to a TCP RST—unless the code bit of the initial non-SYN TCP packet is also RST. In that case, the security device simply drops the packet.

You can enable and disable SYN checking with the following CLI commands:

```
set flow tcp-syn-check
unset flow tcp-syn-check
```

In addition to normal SYN checking, you can configure the security device to do strict SYN checking on all the packets by using the **strict** option with the **set flow tcp-syn-check** command.

```
set flow tcp-syn-check strict
```

When the **strict** feature is enabled, the security device rejects or allows the packets depending on the phase and direction of the packets as explained in the following table:

Table 52: Strict SYN Checking Rules

phase	Phase direction	SYN	SYN+ACK	ACK	ACK+FIN	RST	Others
Phase 1: After receiving the first SYN packet from client	client —> server	Allow	Deny	Deny	Deny	Allow	Deny
	server —> client	Deny	Allow	Deny	Deny	Allow	Deny
Phase 2: After receiving the SYN + ACK packet from server	client —> server	Deny	Deny	Allow	Allow	Allow	Deny
	server —> client	Deny	Allow	Deny	Deny	Allow	Deny
Phase 3: After receiving the ACK or ACK + FIN packet from client	client —> server	Allow	Allow	Allow	Allow	Allow	Allow
	server —> client	Allow	Allow	Allow	Allow	Allow	Allow

When the strict check rule is enabled, two counters **pass-cnt** and **drop-cnt** log the number of packets allowed and denied respectively, by the strict check rule option. If the sum of the number of packets in **pass-cnt** and **drop-cnt** per session reaches 21, the time-out of that session will be set to 2 seconds. Packets that arrive within 2 seconds will be dropped and **drop-cnt** will work until session terminates.

When strict checking is enabled, the output of the following commands display details of the number of packets allowed and denied because of the **strict check**feature: **get session id**, **get session hardware**, **get counter statistics interface**, and **get asic ppu tcp3–way-check** For example, the output of **get asic ppu tcp-3way-check** displays the number of packets dropped because this feature is enabled.

```
get asic ppu tcp-3way-check
```

```
Show ASIC 1 PPU information:
total input: 0,          total fwd: 0
total drop: 0,           redirect to client: 0
packet from server: 0,   msg send to server: 0
msg rcv stage 4: 0,      msg rcv stage 5: 0
msg rcv stage 6: 0
Invalid session count: 0,        syn bit check drop: 0
strict syn check drop: 9
```

Not checking for the SYN flag in the first packets offers the following advantages:

- **NSRP with Asymmetric Routing:** In an Active/Active NSRP configuration in a dynamic routing environment, a host might send the initial TCP segment with the SYN flag set to one security device (Device-A) but the SYN/ACK might be routed to the other security device in the cluster (Device-B). If this asymmetric routing occurs after Device-A has synchronized its session with Device-B, all is well. On the other hand, if the SYN/ACK response reaches Device-B before Device-A synchronizes the session and SYN checking is enabled, Device-B rejects the SYN/ACK, and the session cannot be established. With SYN checking disabled, Device-B accepts the SYN/ACK response—even though there is no existing session to which it belongs—and creates a new session table entry for it.
- **Uninterrupted Sessions:** If SYN checking is enabled and you add a security device operating in transparent mode to a working network, it disrupts all existing sessions, which must then be restarted. For lengthy sessions, such as large data transfers or database backups, this can be a troublesome disruption. Similarly, if you reset the device or even change a component in the core section of a policy and SYN checking is enabled, all existing sessions or those sessions to which the policy change applies are disrupted and must be restarted. Disabling SYN checking avoids such disruptions to network traffic flows.

NOTE: A solution to this scenario is to install the security device with SYN checking disabled initially. Then, after a few hours—when established sessions are running through the device—enable SYN checking.
The core section in a policy contains the following main components: source and destination zones, source and destination addresses, one or more services, and an action.

However, note that the above advantages exact the following security sacrifices:

- **Reconnaissance Holes:** When an initial TCP segment with a non-SYN flag—such as ACK, URG, RST, FIN—arrives at a closed port, many operating systems (Windows, for example) respond with a TCP segment that has the RST flag set. If the port is open, then the recipient does not generate any response.

  By analyzing these responses or lack thereof, an intelligence gatherer can perform reconnaissance on the protected network and also on the ScreenOS policy set. If he sends a TCP segment with a non-SYN flag set and the policy permits it through, the destination host receiving such a segment might drop it and respond with a TCP segment that has the RST flag set. Such a response informs the perpetrator of the presence of an active host at a specific address and that the

targeted port number is closed. The intelligence gatherer also learns that the firewall policy permits access to that port number on that host.

By enabling SYN flag checking, the security device drops TCP segments without a SYN flag if they do not belong to an existing session. It does not return a TCP RST segment. Consequently, the scanner gets no replies regardless of the policy set or whether the port is open or closed on the targeted host.

- **Session Table Floods:** If SYN checking is disabled, an attacker can bypass the ScreenOS SYN flood protection feature by flooding a protected network with a barrage of TCP segments that have non-SYN flags set. Although the targeted hosts drop the packets—and possibly send TCP RST segments in reply—such a flood can fill up the session table of the security device. When the session table is full, the device cannot process new sessions for legitimate traffic.

  By enabling SYN checking and SYN flood protection, you can thwart this kind of attack. Checking that the SYN flag is set on the initial packet in a session forces all new sessions to begin with a TCP segment that has the SYN flag set. SYN flood protection then limits the number of TCP SYN segments per second so that the session table does not become overwhelmed.

NOTE: For information about session table floods, see "Session Table Flood" on page 463. For information about SYN floods, see "SYN Flood" on page 475.

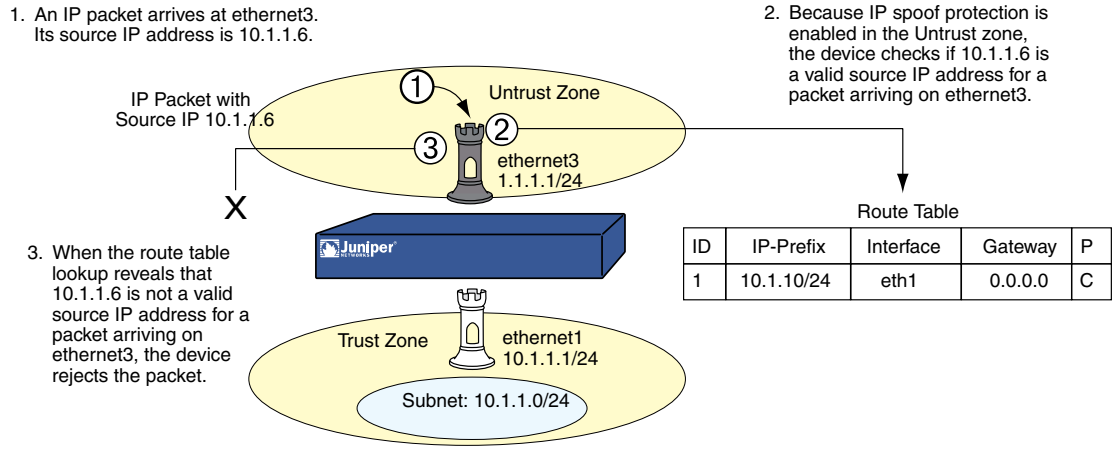
If you do not need SYN checking disabled, we strongly recommend that it be enabled (its default state for an initial installation of ScreenOS). You can enable it with the following command: **set flow tcp-syn-check**. With SYN checking enabled, the security device rejects TCP segments with non-SYN flags set unless they belong to an established session.

IP Spoofing

One method of attempting to gain access to a restricted area of the network is to insert a bogus source address in the packet header to make the packet appear to come from a trusted source. This technique is called IP spoofing. ScreenOS has two IP spoofing detection methods, both of which accomplish the same task: determining that the packet came from a location other than that indicated in its header. The method that a Juniper Networks security device uses depends on whether it is operating at Layer 3 or Layer 2 in the OSI Model.

- **Layer 3**—When interfaces on the security device are operating in route or NAT mode, the mechanism to detect IP spoofing relies on route table entries. If, for example, a packet with source IP address 10.1.1.6 arrives at ethernet3, but the security device has a route to 10.1.1.0/24 through ethernet1, IP spoof checking notes that this address arrived at an invalid interface—as defined in the route table, a valid packet from 10.1.1.6 can only arrive via ethernet1, not ethernet3. Therefore, the device concludes that the packet has a spoofed source IP address and discards it.

Figure 106: Layer 3 IP Spoofing

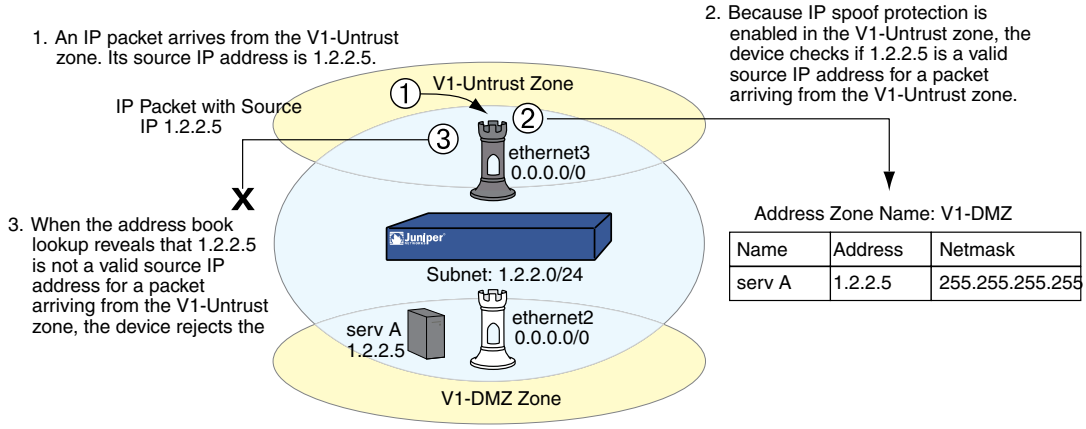


If the source IP address in a packet does not appear in the route table, by default the security device allows that packet to pass (assuming that a policy exists permitting it). Using the following CLI command—where the specified security zone is the one from which the packets originate—you can instruct the security device to drop any packet whose source IP address is not in the route table:

```
set zone zone screen ip-spoofing drop-no-rpf-route
```

- **Layer 2**—When interfaces on the security device are operating in transparent mode, the IP spoof checking mechanism makes use of the address book entries. For example, you define an address for "serv A" as 1.2.2.5/32 in the V1-DMZ zone. If a packet with source IP address 1.2.2.5 arrives at a V1-Untrust zone interface (ethernet3), IP spoof checking notes that this address arrived at an invalid interface. The address belongs to the V1-DMZ zone, not to the V1-Untrust zone, and is accepted only at ethernet2, which is bound to V1-DMZ. The device concludes that the packet has a spoofed source IP address and discards it.

Figure 107: Layer 2 IP Spoofing



Be careful when defining addresses for the subnet that straddles multiple security zones. In Figure 11, 1.2.2.0/24 belongs to both the V1-Untrust and V1-DMZ zones. If you configure the security device as follows, the device will block traffic from the V1-DMZ zone that you want it to permit:

- You define an address for 1.2.2.0/24 in the V1-Untrust zone.
- You have a policy permitting traffic from any address in the V1-DMZ zone to any address in the V1-Untrust zone (**set policy from v1-dmz to v1-untrust any any any permit**).
- You enable IP spoof checking.

  Because addresses in the V1-DMZ zone are also in the 1.2.2.0/24 subnet, when traffic from these addresses reaches ethernet2, the IP spoof check refers to the address book and finds 1.2.2.0/24 in the V1-Untrust zone. Consequently, the security device blocks the traffic.

Example: L3 IP Spoof Protection

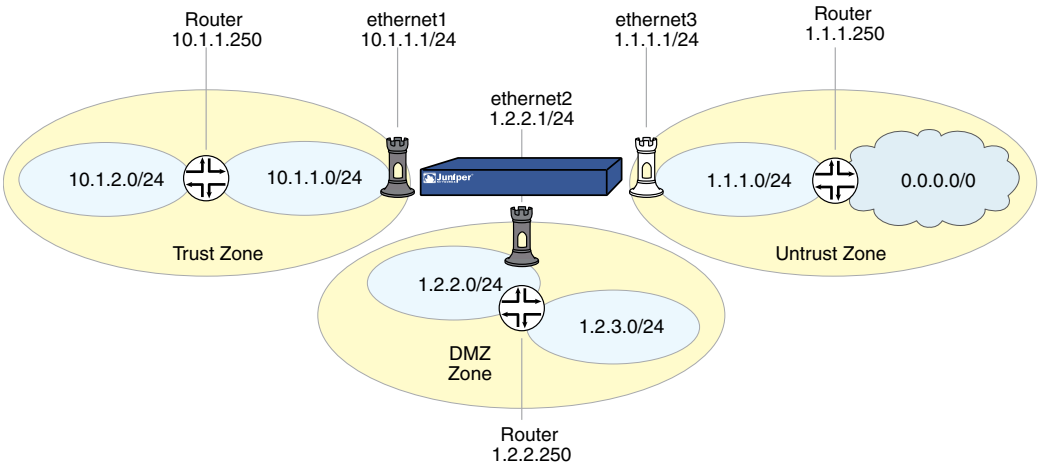
In this example, you enable IP spoof protection for the Trust, DMZ, and Untrust zones for a Juniper Networks security device operating at Layer 3. By default, the device automatically makes entries in the route table for the subnets specified in interface IP addresses. In addition to these automatic route table entries, you manually enter the three routes shown in the following table:

Destination	Egress Interface	Next Gateway
10.1.2.0/24	ethernet1	10.1.1.250
1.2.3.0/24	ethernet2	1.2.2.250
0.0.0.0/0	ethernet3	1.1.1.250

If you enable the IP spoof protection SCREEN option but do not enter the above three routes, the device drops all traffic from the addresses in the “Destination” column and enters alarms in the event log. For example, if a packet with the source address 10.1.2.5 arrives at ethernet1 and there is no route to the 10.1.2.0/24 subnet via ethernet1, the device determines that packet has arrived at an invalid interface and drops it.

All the security zones in this example are in the trust-vr routing domain.

Figure 108: Example of Layer 3 IP Spoofing



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Enter the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select this option when present)
   IP Address/Netmask: 1.2.2.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Routes**

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 10.1.2.0/24
   Gateway: (select)
   Interface: ethernet1
   Gateway IP Address: 10.1.1.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 1.2.3.0/24
Gateway: (select)
Interface: ethernet2
Gateway IP Address: 1.2.2.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

3. **IP Spoof Protection**

Screening > Screen (Zone: Trust): Select **IP Address Spoof Protection**, then click **Apply**.

Screening > Screen (Zone: DMZ): Select **IP Address Spoof Protection**, then click **Apply**.

Screening > Screen (Zone: Untrust): Select **IP Address Spoof Protection**, then click **Apply**.

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet2 zone dmz
set interface ethernet2 ip 1.2.2.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Routes**

```
set vrouter trust-vr route 10.1.2.0/24 interface ethernet1 gateway 10.1.1.250
set vrouter trust-vr route 1.2.3.0/24 interface ethernet2 gateway 1.2.2.250
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

3. **IP Spoof Protection**

```
set zone trust screen ip-spoofing
set zone dmz screen ip-spoofing
set zone untrust screen ip-spoofing
save
```

Example: L2 IP Spoof Protection

In this example, you protect the V1-DMZ zone from IP spoofing on traffic originating in the V1-Untrust zone. First, you define the following addresses for three Web servers in the V1-DMZ zone:

- servA: 1.2.2.10
- servB: 1.2.2.20
- servC: 1.2.2.30

You then enable IP spoofing in the V1-Untrust zone.

If an attacker in the V1-Untrust zone attempts to spoof the source IP address using any of the three addresses in the V1-DMZ zone, the security device checks the address against those in the address books. When it finds that the source IP address on a packet coming from the V1-Untrust zone belongs to a defined address in the V1-DMZ zone, the device rejects the packet.

WebUI

1. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: servA
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.10/32
   Zone: V1-DMZ

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: servB
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.20/32
   Zone: V1-DMZ

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: servC
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.30/32
   Zone: V1-DMZ

2. **IP Spoof Protection**

   Screening > Screen (Zone: V1-Trust): Select **IP Address Spoof Protection**, then click **Apply**.

CLI

1. **Addresses**

```
set address v1-dmz servA 1.2.2.10/32
set address v1-dmz servB 1.2.2.20/32
set address v1-dmz servC 1.2.2.30/32
```

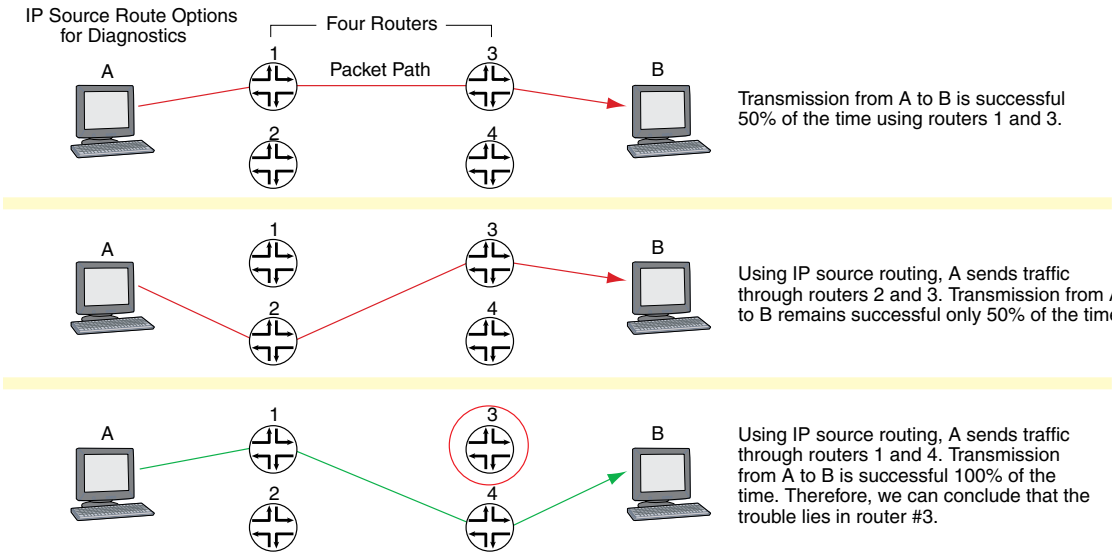
2. **IP Spoof Protection**

```
set zone v1-untrust screen ip-spoofing
save
```

IP Source Route Options

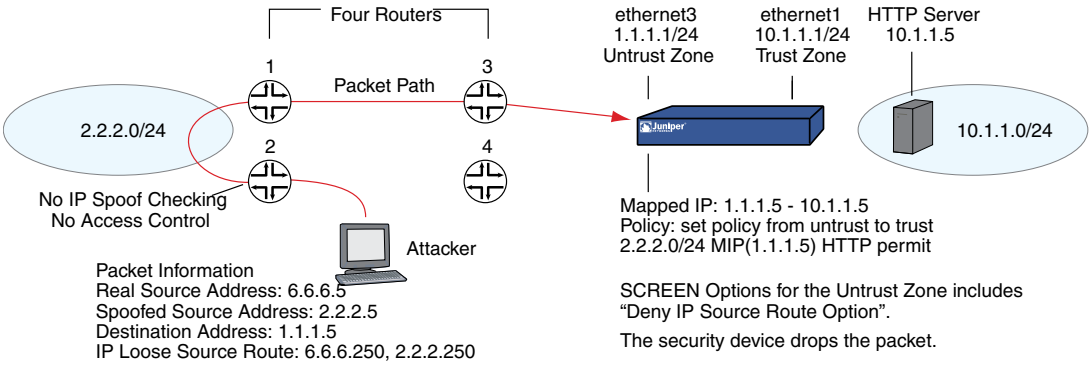
Source routing was designed to allow the user at the source of an IP packet transmission to specify the IP addresses of the routers (also referred to as "hops") along the path that he or she wants an IP packet to take on its way to its destination. The original intent of the IP source route options was to provide routing control tools to aid diagnostic analysis. If, for example, the transmission of a packet to a particular destination meets with irregular success, you might first use either the record route or timestamp IP option to discover the addresses of routers along the path or paths that the packet takes. You can then use either the loose or strict source route option to direct traffic along a specific path, using the addresses you learned from the results that the record route or timestamp options produced. By changing router addresses to alter the path and sending several packets along different paths, you can note changes that either improve or lessen the success rate. Through analysis and the process of elimination, you might be able to deduce where the trouble lies.

Figure 109: IP Source Routing



Although the uses of IP source route options were originally benign, attackers have learned to put them to more devious uses. They can use IP source route options to hide their true address and access restricted areas of a network by specifying a different path. For an example showing how an attacker can put both deceptions to use, consider the following scenario as illustrated in Figure 110 on page 461.

Figure 110: Loose IP Source Route Option for Deception



The Juniper Networks security device only allows traffic 2.2.2.0/24 if it comes through ethernet3, an interface bound to the Untrust zone. Routers 3 and 4 enforce access controls but routers 1 and 2 do not. Furthermore, router 2 does not check for IP spoofing. The attacker spoofs the source address, and by using the loose source route option, directs the packet through router 2 to the 2.2.2.0/24 network and from there out router 1. Router 1 forwards it to router 3, which forwards it to the security device. Because the packet came from the 2.2.2.0/24 subnet and has a source address from that subnet, it seems to be valid. However, one remnant of the earlier chicanery remains: the loose source route option. In this example, you have enabled the “Deny IP Source Route Option” SCREEN option for the Untrust zone. When the packet arrives at ethernet3, the device rejects it.

You can enable the security device to either block any packets with loose or strict source route options set or detect such packets and then record the event in the counters list for the ingress interface. The SCREEN options are as follows:

- **Deny IP Source Route Option:** Enable this option to block all IP traffic that employs the loose or strict source route option. Source route options can allow an attacker to enter a network with a false IP address.
- **Detect IP Loose Source Route Option:** The security device detects packets where the IP option is 3 (Loose Source Routing) and records the event in the SCREEN counters list for the ingress interface. This option specifies a partial route list for a packet to take on its journey from source to destination. The packet must proceed in the order of addresses specified, but it is allowed to pass through other routers in between those specified.
- **Detect IP Strict Source Route Option:** The security device detects packets where the IP option is 9 (Strict Source Routing) and records the event in the SCREEN counters list for the ingress interface. This option specifies the complete route list for a packet to take on its journey from source to destination. The last address in the list replaces the address in the destination field.

(For more information about all the IP options, see “Network Reconnaissance Using IP Options” on page 443.)

To block packets with either a loose or strict source route option set, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **IP Source Route Option Filter**, then click **Apply**.

CLI

```
set zone zone screen ip-filter-src
```

To detect and record (but not block) packets with a loose or strict source route option set, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

```
IP Loose Source Route Option Detection: (select)
IP Strict Source Route Option Detection: (select)
```

CLI

```
set zone zone screen ip-loose-src-route
set zone zone screen ip-strict-src-route
```

Chapter 14

Denial of Service Attack Defenses

The intent of a denial of service (DoS) attack is to overwhelm the targeted victim with a tremendous amount of bogus traffic so that the victim becomes so preoccupied processing the bogus traffic that it is unable to process legitimate traffic. The target can be the security device, the network resources to which the device controls access, or the specific hardware platform or operating system (OS) of an individual host.

If a DoS attack originates from multiple source addresses, it is known as a distributed denial of service (DDoS) attack. Typically, the source address of a DoS attack is spoofed. The source addresses in a DDoS attack might be spoofed or the actual addresses of hosts that the attacker has previously compromised and which he or she is now using as zombie agents from which to launch the attack.

The security device can defend itself and the resources it protects from DoS and DDoS attacks. The following sections describe the various defense options available:

Firewall DoS Attacks

If an attacker discovers the presence of the Juniper Networks security device, the attacker might launch a denial of service (DoS) attack against the security device instead of the network behind it. A successful DoS attack against a firewall amounts to a successful DoS attack against the protected network in that it thwarts attempts of legitimate traffic to traverse the firewall. This section explains the two methods an attacker might use to fill up the session table of a Juniper Networks security device and thereby produce a DoS: session table flood and SYN-ACK-ACK proxy flood.

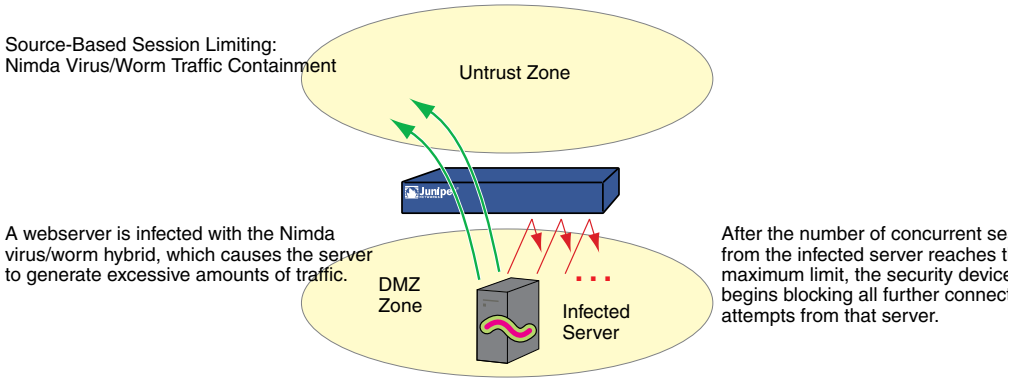
Session Table Flood

A successful DoS attack overwhelms its victim with such a massive barrage of false traffic that the victim becomes unable to process legitimate connection requests. DoS attacks can take many forms—SYN flood, SYN-ACK-ACK flood, UDP flood, ICMP flood, and so on—but they all have the same objective: to fill up their victim's session table. When the session table is full, that host cannot create any new sessions and begins rejecting new connection requests. The following SCREEN options help mitigate such attacks:

Source-Based and Destination-Based Session Limits

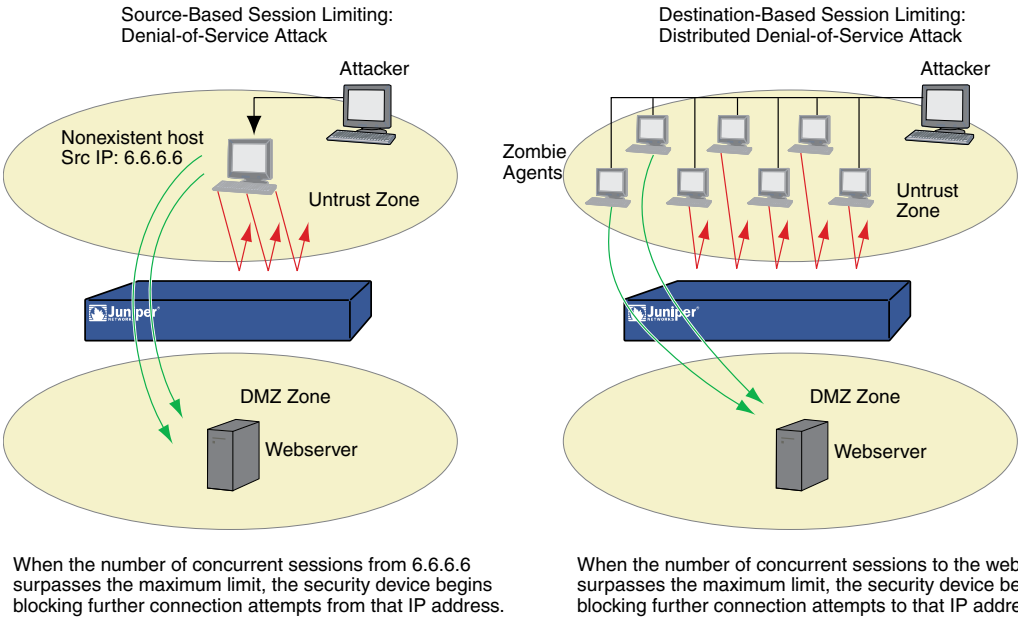
In addition to limiting the number of concurrent sessions from the same source IP address, you can also limit the number of concurrent sessions to the same destination IP address. One benefit of setting a source-based session limit is that it can stem an attack such as the Nimda virus (which is actually both a virus and a worm) that infects a server and then begins generating massive amounts of traffic from that server. Because all the virus-generated traffic originates from the same IP address, a source-based session limit ensures that the firewall can curb such excessive amounts of traffic.

Figure 111: Limiting Sessions Based on Source IP Address



Another benefit of source-based session limiting is that it can mitigate attempts to fill up the ScreenOS session table (if all the connection attempts originate from the same source IP address). However, a wily attacker can launch a distributed denial of service (DDoS) attack. In a DDoS attack, the malicious traffic can come from hundreds of hosts, known as zombie agents, that are surreptitiously under the control of an attacker. In addition to the SYN, UDP, and ICMP flood detection and prevention SCREEN options, setting a destination-based session limit can ensure that the security device allows only an acceptable number of concurrent connection requests—no matter what the source—to reach any one host.

Figure 112: Distributed DOS Attack



Determining what constitutes an acceptable number of connection requests requires a period of observation and analysis to establish a baseline for typical traffic flows. You also need to consider the maximum number of concurrent sessions required to fill up the session table of the particular Juniper Networks platform you are using. To see the maximum number of sessions that your session table supports, use the CLI command **get session**, and then look at the first line in the output, which lists the number of current (allocated) sessions, the maximum number of sessions, and the number of failed session allocations:

```
alloc 420/max 128000, alloc failed 0
```

The default maximum for both source- and destination-based session limits is 128 concurrent sessions, a value that might need adjustment to suit the needs of your network environment and the platform.

Example: Source-Based Session Limiting

In this example, you want to limit the amount of sessions that any one server in the DMZ and Trust zones can initiate. Because the DMZ zone only contains Web servers, none of which should initiate traffic, you set the source-session limit at the lowest possible value: 1 session. On the other hand, the Trust zone contains personal computers, servers, printers, and so on, many of which do initiate traffic. For the Trust zone, you set the source-session limit maximum to 80 concurrent sessions.

WebUI

Screening > Screen (Zone: DMZ): Enter the following, then click **OK**:

```
Source IP Based Session Limit: (select)
Threshold: 1 Sessions
```

Screening > Screen (Zone: Trust): Enter the following, then click **OK**:

Source IP Based Session Limit: (select)
Threshold: 80 Sessions

CLI

```
set zone dmz screen limit-session source-ip-based 1
set zone dmz screen limit-session source-ip-based
set zone trust screen limit-session source-ip-based 80
set zone trust screen limit-session source-ip-based
save
```

Example: Destination-Based Session Limiting

In this example, you want to limit the amount of traffic to a Web server at 1.2.2.5. The server is in the DMZ zone. After observing the traffic flow from the Untrust zone to this server for a month, you have determined that the average number of concurrent sessions it receives is 2000. Based on this information, you decide to set the new session limit at 4000 concurrent sessions. Although your observations show that traffic spikes sometimes exceed that limit, you opt for firewall security over occasional server inaccessibility.

WebUI

Screening > Screen (Zone: Untrust): Enter the following, then click **OK**:

Destination IP Based Session Limit: (select)
Threshold: 4000 Sessions

CLI

```
set zone untrust screen limit-session destination-ip-based 4000
set zone untrust screen limit-session destination-ip-based
save
```

Aggressive Aging

By default, an initial TCP session 3-way handshake takes 20 seconds to time out (that is, to expire because of inactivity). After a TCP session has been established, the timeout value changes to 30 minutes. For HTTP and UDP sessions, the session timeouts are 5 minutes and 1 minute, respectively. The session timeout counter begins when a session starts and is refreshed every 10 seconds if the session is active. If a session becomes idle for more than 10 seconds, the timeout counter begins to decrement.

On certain hardware platforms, ScreenOS provides a mechanism for accelerating the timeout process when the number of sessions in the session table surpasses a specified high-watermark threshold. This feature is not available on the high-end systems.

When the number of sessions dips below a specified low-watermark threshold, the timeout process returns to normal. During the period when the aggressive aging out

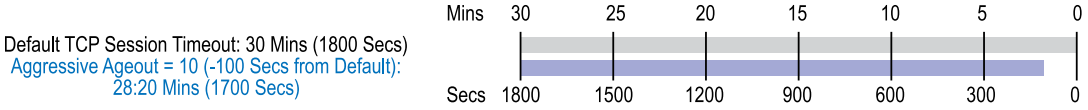
process is in effect, a security device ages out the oldest sessions first, using the aging out rate that you specify. These aged-out sessions are tagged as invalid and are removed in the next "garbage sweep," which occurs every 2 seconds.

The aggressive ageout option shortens default session timeouts by the amount you enter. When you set and enable the aggressive ageout option, the normal session timeout value displayed in the configuration remains unchanged—1800 seconds for TCP, 300 seconds for HTTP, and 60 seconds for UDP sessions. However, when the aggressive ageout period is in effect, these sessions time out earlier—by the amount you specify for early ageout—instead of counting down all the way to zero.

The aggressive ageout value can be between 2 and 10 units, where each unit represents a 10-second interval (that is, the aggressive ageout setting can be between 20 and 100 seconds). The default setting is 2 units, or 20 seconds. If you define the aggressive ageout setting at 100 seconds, for example, you shorten the TCP and HTTP session timeouts as follows:

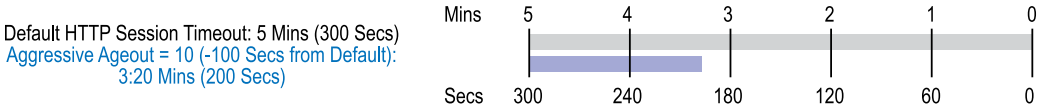
- TCP: The session timeout value shortens from 1800 seconds (30 minutes) to 1700 seconds (28:20 minutes) during the time when the aggressive aging process is in effect. During that period, the security device automatically deletes all TCP sessions whose timeout value has passed 1700 seconds, beginning with the oldest sessions first.

Figure 113: TCP Session Timeout



- HTTP: The session timeout value shortens from 300 seconds (5 minutes) to 200 seconds (3:20 minutes) during the time when the aggressive aging process is in effect. During that period, the security device automatically deletes all HTTP sessions whose timeout value has passed 200 seconds, beginning with the oldest sessions first.

Figure 114: HTTP Session Timeout



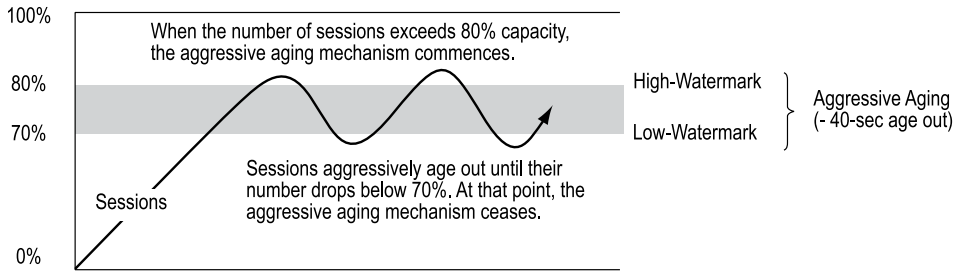
- UDP: Because the default UDP session timeout is 60 seconds, defining an early ageout setting at 100 seconds causes all UDP sessions to ageout and be marked for deletion in the next garbage sweep.

Example: Aggressively Aging Out Sessions

In this example, you set the aggressive aging out process to commence when traffic exceeds a high-watermark of 80 percent and cease when it retreats below a low-watermark of 70 percent. You specify 40 seconds for the aggressive age-out interval. When the session table is more than 80 percent full (the high-mark threshold), the security device decreases the timeout for all sessions by 40 seconds and begins

aggressively aging out the oldest sessions until the number of sessions in the table is under 70 percent (the low-mark threshold).

Figure 115: Aging Out Sessions Aggressively



WebUI

NOTE: You must use the CLI to configure the aggressive age-out settings.

CLI

```
set flow aging low-watermark 70
set flow aging high-watermark 80
set flow aging early-ageout 4
save
```

CPU Protection with Blacklisting DoS Attack Traffic

When a DoS attack occurs, the CPU recognizes the attack traffic and drops it. This can cause high CPU utilization and might make the security device drop all packets, including critical traffic such as management traffic. To prevent this, you can configure the security device to drop malicious packets within the device itself that processes them, after the CPU has recognized malicious traffic. In this mechanism, you create a blacklist of IP addresses from which malicious traffic reaches the security device, based on which the CPU instructs the device to drop the traffic. This saves significant processing load on the CPU during DoS attacks.

NOTE: The blacklist protection feature is not available for the following traffic conditions:

- IPV6 traffic
- IPV4 traffic when IPv6 is set as environment variable
- Traffic that has hardware session enabled

When a packet reaches the device, the packet processing hardware checks the packet against the list of blacklist entries. If a match occurs, the device drops that packet. If the packet does not match any blacklist entry, the device passes the packet to the

next stage that prioritizes the packet. For each entry in the blacklist, the security device maintains a drop counter to record the number of packets dropped against that entry.

The blacklist protection mechanism applies to the entire security device and is not limited to specific virtual systems that you may have created on the security device. In security devices that support virtual systems but do not support blacklist creation, CPU protection features such as rate limiting apply.

Creating a Blacklist

To implement blacklisting of DoS attack traffic, you create a blacklist. The security device CPU screens the traffic that reaches it and determines if a flow matches a DoS attack pattern. If a packet matches the blacklist entry, the device drops the packet.

You can set the timeout value for each of the blacklist entries. To permanently block specific traffic that has been identified as DoS attack traffic, set the timeout value for that blacklist entry to 0.

You create the blacklist with the following information:

Field	Description
Source IP Address	The source IP address from which the DoS attack traffic originated
Destination IP Address	The destination IP address.
Source Port	The source port in a TCP or UDP session. Setting this to 0 matches all ports
Destination Port	The destination port in a TCP or UDP session. Setting this to 0 matches all ports.
Protocol	Set this to 0 to match any protocol. The source port and destination port are valid only when you have set the protocol as UDP or TCP
Source IP Address Mask	Range is 0–32. Setting this to 0 matches all source IP addresses.
Destination IP Mask	Range is 0–32. Setting this to 0 matches all destination IP addresses.
Blacklist ID	The ID of the blacklists. Range is 0–31.
Timeout	The time out for the blacklist entry in the range 0 to 600 minutes. If you set the timeout for a blacklist entry to 0, the security device never times out that entry. The security device saves only the permanent entries in the blacklist configuration.

Example

In this example, you create a blacklist entry that times out after 90 minutes.

WebUI

Configuration > CPU Protection > Black List > New: Enter the following, then click **OK:**
ID: 1
Source IP/Netmask: 1.1.1.0/24
Source Port: 5
Destination IP/Netmask : 2.2.2.0/24
Destination Port: 7
Protocol: 17
Timeout: 90

CLI

```
set cpu-protection blacklist id 1 1.1.1.0/24 2.2.2.0/24 protocol 17 src-port 5 dst-port
 7 timeout 90
save
```

NOTE: You cannot create a blacklist entry with the source IP address mask, the destination IP address mask, and the protocol values set to 0.

Prioritizing Critical Traffic

In addition to dropping the malicious packets in the device, this mechanism provides prioritizing of traffic in high CPU utilization situations so that the security device allows critical traffic such as management traffic and drops noncritical traffic. For this mechanism to function, you configure a utilization threshold on the CPU. During a high-utilization situation, this mechanism compares the current CPU utilization

with the threshold value you have set, and then prioritizes the critical traffic. This can cause the security device to drop noncritical traffic.

Type	Class	Protocol
Critical	1	TELNET—device management
		SSH—device management
		HTTP/HTTPS—device management
		BGP—routing protocol updates
		OSPF—routing protocol updates
		RIP—routing protocol updates
		RIPNG—routing protocol updates
		PIM—multicast routing protocol updates
		NSRP—NSRP updates
		IKE/VPN Monitor—tunnel setup and VPN Monitor packets
		ARP—ARP responses, so that the device can move the session to hardware
		RADIUS—authentication protocol
		LDAP—authentication protocol
		SNMP/SNMP traps—SNMP updates
		NSM—communication with Network and Security Manager
		TFTP—Trivial File Transfer Protocol
		ICMP—Internet Control Message Protocol
Noncritical	2	Broadcast
	3	Non-first packet
	4	First packet
	5	Other

WebUI

Configuration > CPU Protection > General Settings: Enter the CPU Protection Threshold, then click **Apply:**

CLI

```
set cpu-protection threshold number
save
```

The following table shows the traffic statistics of the **get cpu-protection** command when the threshold is set to 70 percent.

```
Current usage: 80%  High CPU threshold: 70%
```

Class	Traffic	Dropped	Passed
1	Critical	0	16
2	ICMP/BC/ARP	3	6
3	Non-first	0	7
4	First	0	3
5	Other	1	2

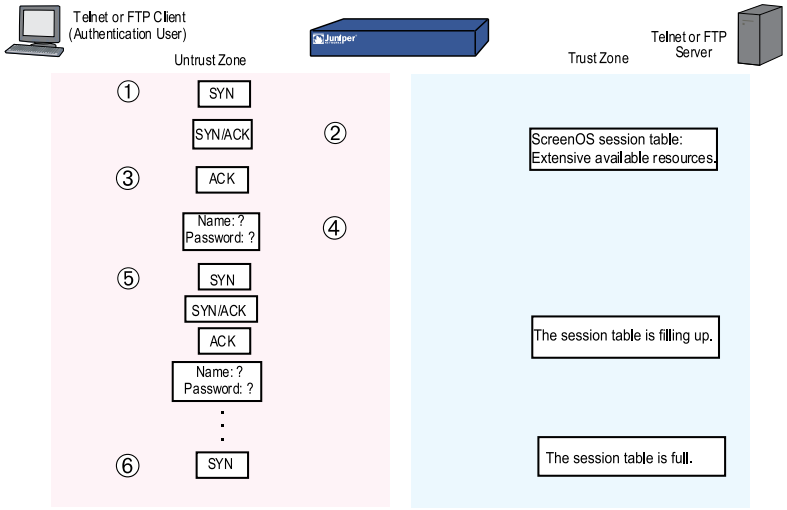
SYN-ACK-ACK Proxy Flood

When an authentication user initiates a Telnet or FTP connection, the user sends a SYN segment to the Telnet or FTP server. The security device intercepts the SYN segment, creates an entry in its session table, and proxies a SYN-ACK segment to the user. The user then replies with an ACK segment. At that point, the initial three-way handshake is complete. The device sends a login prompt to the user. If the user, with malicious intent, does not log in, but instead continues initiating SYN-ACK-ACK sessions, the ScreenOS session table can fill up to the point where the device begins rejecting legitimate connection requests.

See Figure 116 on page 473 for a step-by-step process:

1. The client sends a SYN segment to the server.
2. The security device proxies a SYN/ACK segment.
3. The client responds with an ACK segment.
4. The security device prompts the client (auth user) to log in.
5. The client ignores the login prompt and keeps repeating steps 1—4 until the session table is full.
6. Because the session table is full, the security device must reject all further connection requests.

Figure 116: SYN-ACK-ACK Proxy Flood


Telnet or FTP Client
(Authentication User)
Untrust Zone
Trust Zone
Telnet or FTP
Server
①
SYN
SYN/ACK
②
ScreenOS session table:
Extensive available resources.
③
ACK
Name: ?
Password: ?
④
⑤
SYN
SYN/ACK
ACK
Name: ?
Password: ?
The session table is filling up.
⑥
SYN
The session table is full.

To prevent such an attack, you can enable the SYN-ACK-ACK proxy protection SCREEN option. After the number of connections from the same IP address reaches the SYN-ACK-ACK proxy threshold, the security device rejects further connection requests from that IP address. By default, the threshold is 512 connections from any single IP address. You can change this threshold (to any number between 1 and 250,000) to better suit the requirements of your network environment.

NOTE: The alarm-without-drop option does not apply to this SCREEN option. For more information about this option, see "Exploit Monitoring" on page 436.

To enable protection against a SYN-ACK-ACK proxy flood, do either of the following, where the specified zone is that in which the attack originates:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

SYN-ACK-ACK Proxy Protection: (select)
Threshold: (enter a value to trigger SYN-ACK-ACK proxy flood protection)

NOTE: The value unit is connections per source address. The default value is 512 connections from any single address.

CLI

set zone *zone* screen syn-ack-ack-proxy threshold *number*
set zone *zone* screen syn-ack-ack-proxy

Network DoS Attacks

A denial of service (DoS) attack directed against one or more network resources floods the target with an overwhelming number of SYN, ICMP, or UDP packets, or with an overwhelming number of SYN fragments. Depending on the attacker's purpose and the extent and success of previous intelligence gathering efforts, the attacker might single out a specific host, such as a router or server; or he or she might aim at random hosts across the targeted network. Either approach has the potential of upsetting service to a single host or to the entire network, depending on how critical the role of the victim is to the rest of the network.

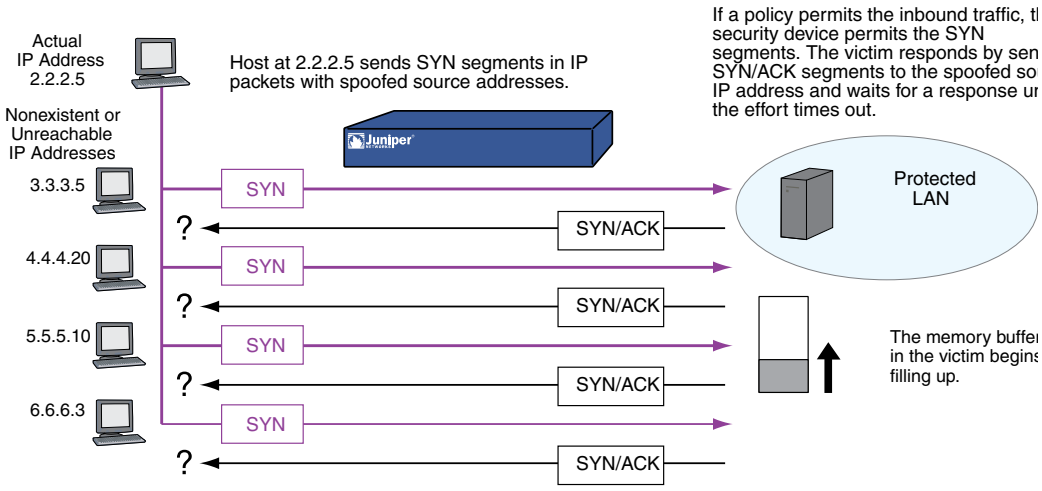
SYN Flood

A SYN flood occurs when a host becomes so overwhelmed by SYN segments initiating incomplete connection requests that it can no longer process legitimate connection requests.

Two hosts establish a TCP connection with a triple exchange of packets known as a *three-way handshake*: A sends a SYN segment to B; B responds with a SYN/ACK segment; and A responds with an ACK segment. A SYN flood attack inundates a site

with SYN segments containing forged (spoofed) IP source addresses with nonexistent or unreachable addresses. B responds with SYN/ACK segments to these addresses and then waits for responding ACK segments. Because the SYN/ACK segments are sent to nonexistent or unreachable IP addresses, they never elicit responses and eventually time out.

Figure 117: SYN Flood Attack

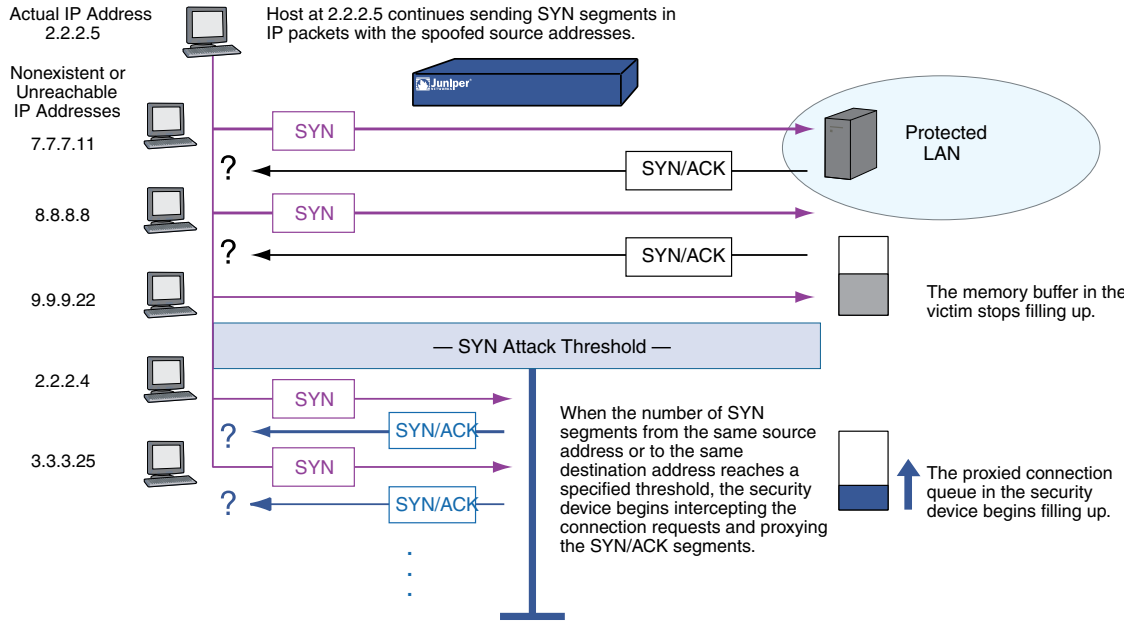


By flooding a host with incomplete TCP connections, the attacker eventually fills the memory buffer of the victim. Once this buffer is full, the host can no longer process new TCP connection requests. The flood might even damage the victim's operating system. Either way, the attack disables the victim and its normal operations.

SYN Flood Protection

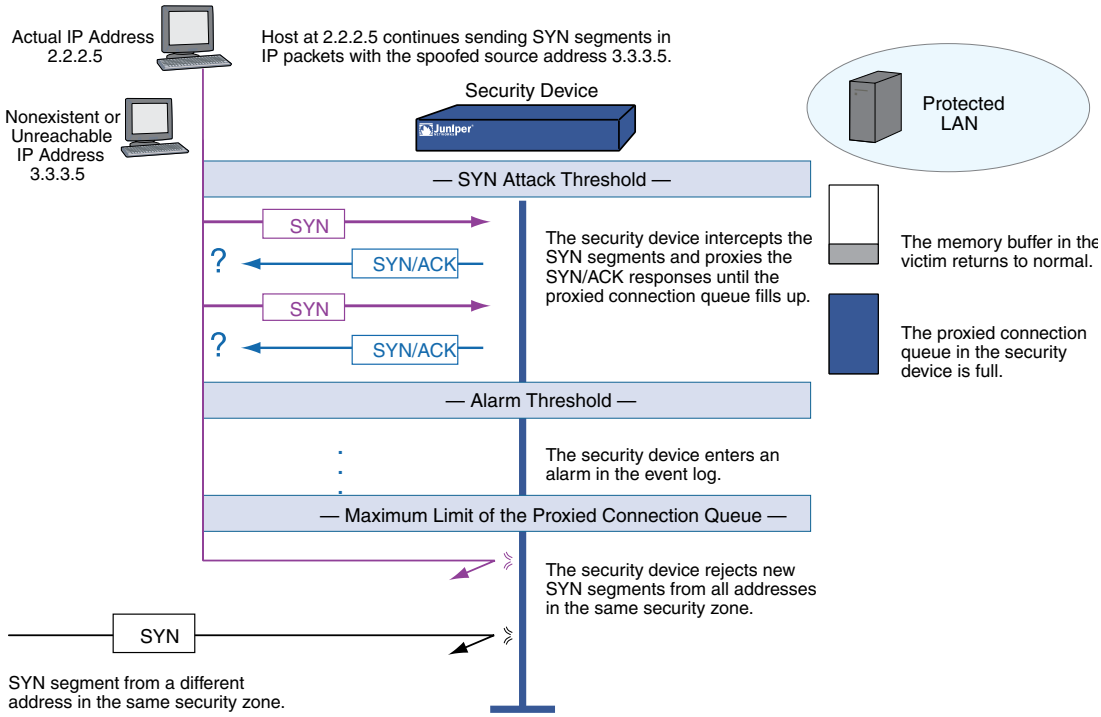
Juniper Networks security devices can impose a limit on the number of SYN segments permitted to pass through the firewall per second. You can base the attack threshold on the destination address and port, the destination address only, or the source address only. When the number of SYN segments per second exceeds one of these thresholds, the security device starts proxying incoming SYN segments, replying with SYN/ACK segments and storing the incomplete connection requests in a connection queue. The incomplete connection requests remain in the queue until the connection is completed or the request times out. In Figure 118 on page 477, the SYN attack threshold has been passed, and the device has started proxying SYN segments.

Figure 118: Proxying SYN Segments



In Figure 119 on page 478, the proxied connection queue has completely filled up, and the security device is rejecting new incoming SYN segments. This action shields hosts on the protected network from the bombardment of incomplete three-way handshakes.

Figure 119: Rejecting New SYN Segments



The security device starts receiving new SYN packets when the proxy queue drops below the maximum limit.

NOTE: The procedure of proxying incomplete SYN connections above a set threshold pertains only to traffic permitted by existing policies. Any traffic for which a policy does not exist is automatically dropped.

By default, the SYN Flood protection SCREEN option is enabled on the Untrust zone. To enable the SYN Flood protection SCREEN option and define its parameters, do either of the following, where the specified zone is that in which a SYN flood might originate:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

SYN Flood Protection: (select to enable)
Threshold: (enter the number of SYN packets–that is, TCP segments with the SYN flag set–per second required to activate the SYN proxying mechanism)
Alarm Threshold: (enter the number of proxied TCP connection requests required to write an alarm in the event log)
Source Threshold: (enter the number of SYN packets per second from a single IP address required for the security device to begin rejecting new connection requests from that source)

Destination Threshold: (enter the number of SYN packets per second to a single IP address required for the security device to begin rejecting new connection requests to that destination)
Timeout Value: (enter the length of time in seconds that the security device holds an incomplete TCP connection attempt in the proxied connection queue)
Queue Size: (enter the number of proxied TCP connection requests held in the proxied connection queue before the security device starts rejecting new connection requests)

NOTE: For more details about each of these parameters, see the descriptions in the following CLI section.

CLI

To enable SYN Flood protection:

set zone *zone* **screen syn-flood**

You can set the following parameters for proxying uncompleted TCP connection requests:

- **Attack Threshold:** The number of SYN segments (that is, TCP segments with the SYN flag set) to the same destination address and port number per second required to activate the SYN proxying mechanism. Although you can set the threshold at any number, you need to know the normal traffic patterns at your site to set an appropriate threshold for it. For example, if it is an e-business site that normally gets 2000 SYN segments per second, you might want to set the threshold at 3000/second. If a smaller site normally gets 20 SYN segments/second, you might consider setting the threshold at 40.

  set zone *zone* **screen syn-flood attack-threshold** *number*

- **Alarm Threshold:** The number of proxied, half-complete TCP connection requests per second after which the security device enters an alarm in the event log. The value you set for an alarm threshold triggers an alarm when the number of proxied, half-completed connection requests to the same destination address and port number per second exceeds that value. For example, if you set the SYN attack threshold at 2000 SYN segments per second and the alarm at 1000, then a total of 3001 SYN segments to the same destination address and port number per second is required to trigger an alarm entry in the log. More precisely:

  1. The firewall passes the first 2000 SYN segments per second that meet policy requirements.
  2. The firewall proxies the next 1000 SYN segments in the same second.
  3. The 1001st proxied connection request (or 3001st connection request in that second) triggers the alarm.

  set zone *zone* **screen syn-flood alarm-threshold** *number*

  For each SYN segment to the same destination address and port number in excess of the alarm threshold, the attack detection module generates a message. At the end of the second, the logging module compresses all similar messages

into a single log entry that indicates how many SYN segments to the same destination address and port number arrived after exceeding the alarm threshold. If the attack persists beyond the first second, the event log enters an alarm every second until the attack stops.

- **Source Threshold:** This option allows you to specify the number of SYN segments received per second from a single source IP address—regardless of the destination IP address and port number—before the security device begins dropping connection requests from that source.

  set zone *zone* **screen syn-flood source-threshold** *number*

  Tracking a SYN flood by source address uses different detection parameters from tracking a SYN flood by destination address and destination port number. When you set a SYN attack threshold and a source threshold, you put both the basic SYN flood protection mechanism and the source-based SYN flood tracking mechanism in effect.

- **Destination Threshold:** This option allows you to specify the number of SYN segments received per second for a single destination IP address before the security device begins dropping connection requests to that destination. If a protected host runs multiple services, you might want to set a threshold based on destination IP address only—regardless of the destination port number.

  set zone *zone* **screen syn-flood destination-threshold** *number*

  When you set a SYN attack threshold and a destination threshold, you put both the basic SYN flood protection mechanism and the destination-based SYN flood tracking mechanism in effect.

  Tracking a SYN flood by destination address uses different detection parameters from tracking a SYN flood by destination address and destination port number. Consider the following case where the security device has policies permitting FTP requests (port 21) and HTTP requests (port 80) to the same server. If the SYN flood attack threshold is 1000 packets per second (pps) and an attacker sends 999 FTP packets and 999 HTTP packets per second, neither set of packets (where a set is defined as having the same destination address and port number) activates the SYN proxying mechanism. The basic SYN flood attack mechanism tracks destination address and port number, and neither set exceeds the attack threshold of 1000 pps. However, if the destination threshold is 1000 pps, the device treats both FTP and HTTP packets with the same destination address as members of a single set and rejects the 1001st packet—FTP or HTTP—to that destination.

- **Timeout:** The maximum length of time before a half-completed connection is dropped from the queue. The default is 20 seconds, and you can set the timeout from 0–50 seconds. You might try decreasing the timeout value to a shorter length until you begin to see any dropped connections during normal traffic conditions. Twenty seconds is a very conservative timeout for a three-way handshake ACK response.

  set zone *zone* **screen syn-flood timeout** *number*

- **Queue size:** The number of proxied connection requests held in the proxied connection queue before the security device starts rejecting new connection requests. The longer the queue size, the longer the device needs to scan the

queue to match a valid ACK response to a proxied connection request. This can slightly slow the initial connection establishment; however, because the time to begin data transfer is normally far greater than any minor delays in initial connection setup, users would not see a noticeable difference.

set zone *zone* **screen syn-flood queue-size** *number*

- **Drop Unknown MAC:** When a security device detects a SYN attack, it proxies all TCP connection requests. However, a device in transparent mode cannot proxy a TCP connection request if the destination MAC address is not in its MAC learning table. By default, a device in transparent mode that has detected a SYN attack passes SYN packets containing unknown MAC addresses. You can use this option to instruct the device to drop SYN packets containing unknown destination MAC addresses instead of letting them pass.

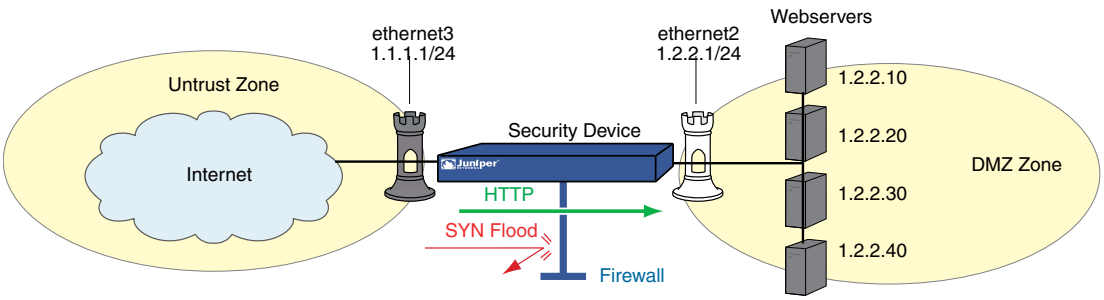
set zone *zone* **screen syn-flood drop-unknown-mac**

Example: SYN Flood Protection

In this example, you protect four Web servers in the DMZ zone from SYN flood attacks originating in the Untrust zone by enabling the SYN flood protection SCREEN option for the Untrust zone.

NOTE: We recommend that you augment the SYN flood protection that the security device provides with device-level SYN flood protection on each of the Web servers. In this example, the Web servers are running UNIX, which also provides some SYN flood defenses, such as adjusting the length of the connection request queue and changing the timeout period for incomplete connection requests.

Figure 120: Device-Level SYN Flood Protection



To configure the SYN flood protection parameters with appropriate values for your network, you must first establish a baseline of typical traffic flows. For one week, you run a sniffer on ethernet3—the interface bound to the Untrust zone—to monitor the number of new TCP connection requests arriving every second for the four Web servers in the DMZ. Your analysis of the data accumulated from one week of monitoring produces the following statistics:

- Average number of new connection requests per server: 250/second
- Average peak number of new connection requests per server: 500/second

NOTE: A sniffer is a network-analyzing device that captures packets on the network segment to which you attach it. Most sniffers allow you to define filters to collect only the type of traffic that interests you. Later, you can view and evaluate the accumulated information. In this example, you want the sniffer to collect all TCP packets with the SYN flag set arriving at ethernet3 and destined for one of the four Web servers in the DMZ.
You might want to continue running the sniffer at regular intervals to see if there are traffic patterns based on the time of day, days of the week, the time of month, or the season of the year. For example, in some organizations, traffic might increase dramatically during a critical event. Significant changes probably warrant adjusting the various thresholds.

Based on this information, you set the following SYN flood protection parameters for the Untrust zone as shown in Table 53 on page 482.

Table 53: SYN Flood Protection Parameters

Parameter	Value	Reason for Each Value
Attack Threshold	625 packets per second (pps)	This is 25% higher than the average peak number of new connection requests per second per server, which is unusual for this network environment. When the number of SYN packets per second for any one of the four Web servers exceeds this number, the device begins proxying new connection requests to that server. (In other words, beginning with the 626th SYN packet to the same destination address and port number in one second, the device begins proxying connection requests to that address and port number.)
Alarm Threshold	250 pps	250 pps is 1/4 of the queue size (1000 proxied, half-completed connection requests). When the device proxies 251 new connection requests in one second, it makes an alarm entry in the event log. By setting the alarm threshold somewhat higher than the attack threshold, you can avoid alarm entries for traffic spikes that only slightly exceed the attack threshold. Note: Half-completed connection requests are incomplete three-way handshakes. A three-way handshake is the initial phase of a TCP connection. It consists of a TCP segment with the SYN flag set, a response with the SYN and ACK flags set, and a response to that with the ACK flag set.
Source Threshold	25 pps	When you set a source threshold, the device tracks the source IP address of SYN packets, regardless of the destination address and port number. (Note that this source-based tracking is separate from the tracking of SYN packets based on destination address and destination port number that constitutes the basic SYN flood protection mechanism.) In the one week of monitoring activity, you observed that no more than 1/25 of new connection requests for all servers came from any one source within a one-second interval. Therefore, connection requests exceeding this threshold are unusual and provide sufficient cause for the device to execute its proxying mechanism. (25 pps is 1/25 of the attack threshold, which is 625 pps.) If the device tracks 25 SYN packets from the same source IP address, beginning with the 26th packet, it rejects all further SYN packets from that source for the remainder of that second and the next second as well.

Table 53: SYN Flood Protection Parameters *(continued)*

Parameter	Value	Reason for Each Value
Destination Threshold	0 pps	When you set a destination threshold, the device runs a separate tracking of only the destination IP address, regardless of the destination port number. Because the four Web servers only receive HTTP traffic (destination port 80)—no traffic to any other destination port number reaches them—setting a separate destination threshold offers no additional advantage.
Timeout	20 seconds	Because the queue size is relatively short (1000 proxied connection requests), the default value of 20 seconds is a reasonable length of time to hold incomplete connection requests in the queue for this configuration.
Queue Size	1000 proxied, half-completed connections	1000 proxied, half-completed connection requests is twice the average peak number of new connection requests (500 pps). The device proxies up to 1000 requests per second before dropping new requests. Proxying twice the average peak number of new connection requests provides a conservative buffer for legitimate connection requests to get through.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select this option when present)
   IP Address/Netmask: 1.2.2.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: ws1
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.10/32
   Zone: DMZ

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: ws2
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.20/32
   Zone: DMZ

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: ws3
IP Address/Domain Name:
IP/Netmask: (select), 1.2.2.30/32
Zone: DMZ

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: ws4
IP Address/Domain Name:
IP/Netmask: (select), 1.2.2.40/32
Zone: DMZ

Policy > Policy Elements > Addresses > Groups > (for Zone: DMZ) New: Enter the following group name, move the following addresses, then click **OK**:

Group Name: web_servers

Select **ws1** and use the < < button to move the address from the Available Members column to the Group Members column.

Select **ws2** and use the < < button to move the address from the Available Members column to the Group Members column.

Select **ws3** and use the < < button to move the address from the Available Members column to the Group Members column.

Select **ws4** and use the < < button to move the address from the Available Members column to the Group Members column.

3. **Policy**

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), web_servers
Service: HTTP
Action: Permit

4. **Screen**

Screening > Screen (Zone: Untrust): Enter the following, then click **Apply**:

SYN Flood Protection: (select)
    Threshold: 625
    Alarm Threshold: 250
    Source Threshold: 25
    Destination Threshold: 0
    Timeout Value: 20
    Queue Size: 1000

NOTE: Because 20 seconds is the default setting, you do not have to set the timeout to 20 seconds unless you have previously set it to another value.

CLI

1. **Interfaces**

```
set interface ethernet2 zone dmz
set interface ethernet2 ip 1.2.2.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address dmz ws1 1.2.2.10/32
set address dmz ws2 1.2.2.20/32
set address dmz ws3 1.2.2.30/32
set address dmz ws4 1.2.2.40/32
set group address dmz web_servers add ws1
set group address dmz web_servers add ws2
set group address dmz web_servers add ws3
set group address dmz web_servers add ws4
```

3. **Policy**

```
set policy from untrust to dmz any web_servers HTTP permit
```

4. **Screen**

```
set zone untrust screen syn-flood attack-threshold 625
set zone untrust screen syn-flood alarm-threshold 250
set zone untrust screen syn-flood source-threshold 25
set zone untrust screen syn-flood timeout 20
set zone untrust screen syn-flood queue-size 1000
set zone untrust screen syn-flood
save
```

NOTE: Because 20 seconds is the default setting, you do not have to set the timeout to 20 seconds unless you have previously set it to another value.

SYN Cookie

SYN Cookie is a stateless SYN Proxy mechanism you can use in conjunction with the defenses against a SYN Flood attack described in “SYN Flood” on page 475. Like traditional SYN proxying, SYN Cookie is activated when the SYN Flood attack threshold is exceeded, but because SYN Cookie is stateless, it does not set up a session or do policy and route lookups upon receipt of a SYN segment, and maintains no connection

request queues. This dramatically reduces CPU and memory usage and is the primary advantage of using SYN Cookie over the traditional SYN proxying mechanism.

When SYN Cookie is enabled on the security device and becomes the TCP-negotiating proxy for the destination server, it replies to each incoming SYN segment with a SYN/ACK containing an encrypted cookie as its Initial Sequence Number (ISN). The cookie is a MD5 hash of the original source address and port number, destination address and port number, and ISN from the original SYN packet. After sending the cookie, the device drops the original SYN packet and deletes the calculated cookie from memory. If there is no response to the packet containing the cookie, the attack is noted as an active SYN attack and is effectively stopped.

If the initiating host responds with a TCP packet containing the cookie +1 in the TCP ACK field, the device extracts the cookie, subtracts 1 from the value, and recomputes the cookie to validate that it is a legitimate ACK. If it is legitimate, the device starts the TCP proxy process by setting up a session and sending a SYN to the server containing the source information from the original SYN. When the device receives a SYN/ACK from the server, it sends ACKs to the sever and to the initiation host. At this point the connection is established and the host and server are able to communicate directly.

In high-end devices, the PPU ASIC in the security device performs the SYN Cookie mechanism instead of the security device CPU. When a valid user sends a SYN packet, the ASIC verifies if the IP address of the incoming packet is present in the whitelist. If the IP address is in the whitelist, the ASIC sends the packet to the security device CPU to create a session. The ASIC does not send the SYN-ACK packet to the user, thereby reducing the ramp-up rate for valid users.

To verify if a specific IP address (for example, 10.100.11.234) is in the ASIC whitelist:

```
get asic ppu syn-cookie  whitelist 10.100.11.234
```

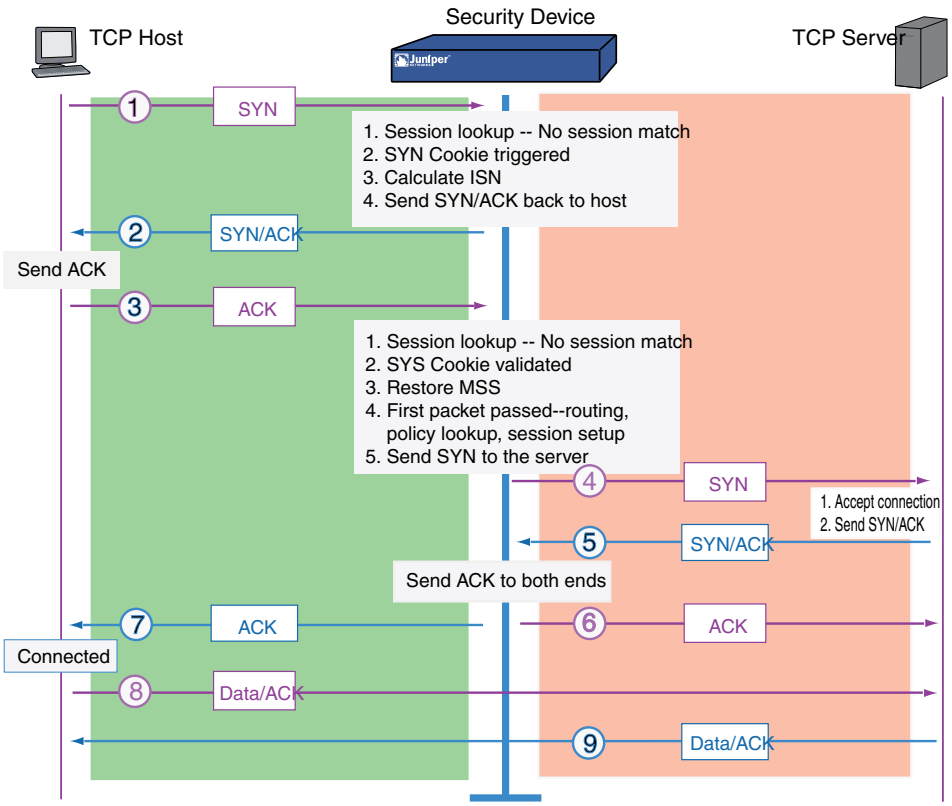
If the IP address is in the whitelist, the following message is displayed:

```
Found IP address 10.100.11.234 in white list, left time 5 second(s)
```

The time in seconds indicates the time after which the IP address ages out and is dropped from the whitelist.

Figure 121 on page 487 shows how a connection is established between an initiating host and a server when SYN Cookie is active on the security device.

Figure 121: Establishing a Connection with SYN Cookie Active



To enable SYN Cookie, set a SYN flood attack threshold (as described in "SYN Flood" on page 475), and do one of the following:

WebUI

Configuration > Advanced > Flow: Enter the following, then click **Apply**:

TCP SYN-proxy SYN-cookie: (select)

CLI

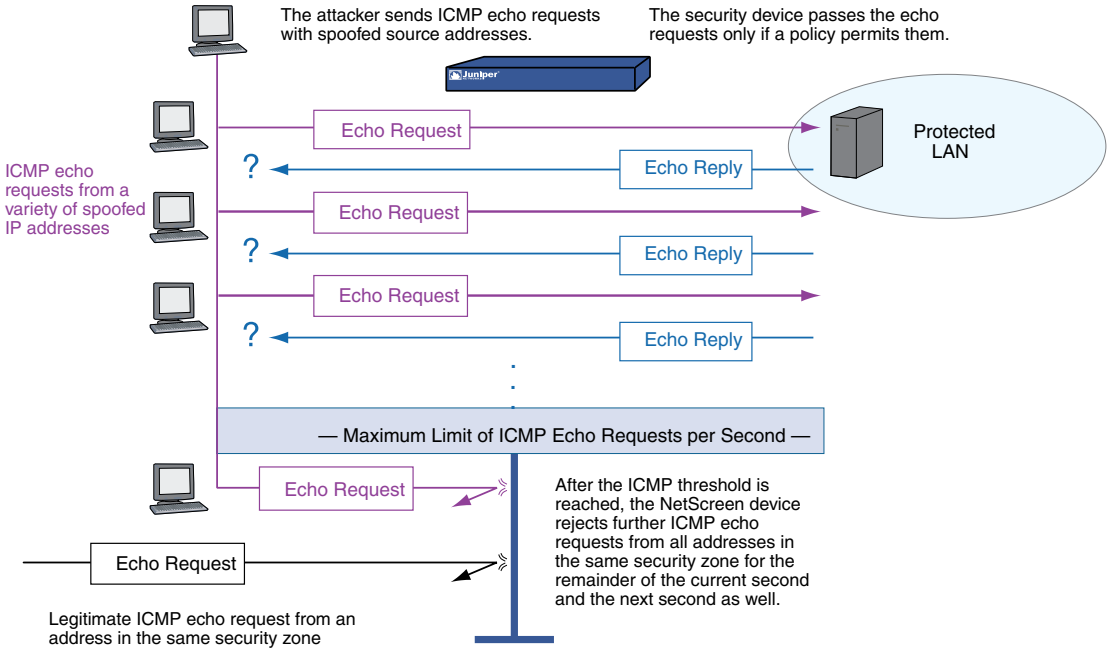
```
set flow syn-proxy syn-cookie
```

ICMP Flood

An ICMP flood typically occurs when ICMP echo requests overload its victim with so many requests that it expends all its resources responding until it can no longer process valid network traffic. When enabling the ICMP flood protection feature, you can set a threshold that once exceeded invokes the ICMP flood attack protection feature. (The default threshold value is 1000 packets per second.) If the threshold is exceeded, the security device ignores further ICMP echo requests for the remainder of that second plus the next second as well.

NOTE: An ICMP flood can consist of any type of ICMP message. Therefore, a Juniper Networks security device monitors all ICMP message types, not just echo requests.

Figure 122: ICMP Flooding



To enable ICMP flood protection, do either of the following, where the specified zone is that in which a flood might originate:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

ICMP Flood Protection: (select)
Threshold: (enter a value to trigger ICMP flood protection)

NOTE: The value unit is ICMP packets per second. The default value is 1000 packets per second.

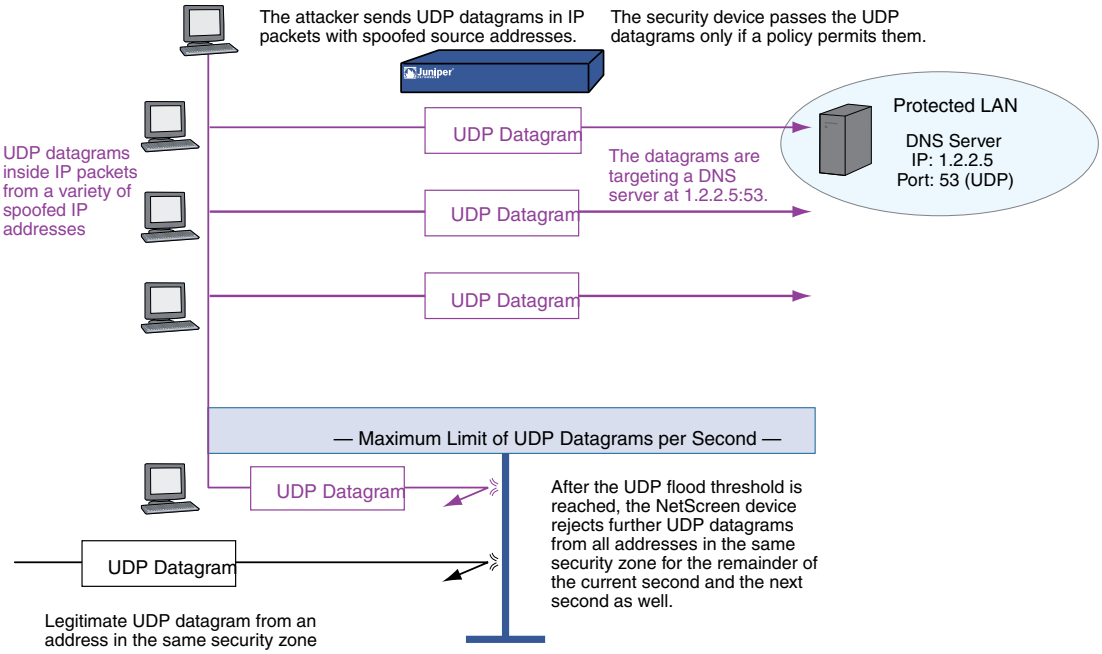
CLI

set zone *zone* screen icmp-flood threshold *number*
set zone *zone* screen icmp-flood

UDP Flood

Similar to the ICMP flood, UDP flooding occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that it can no longer handle valid connections. After enabling the UDP flood protection feature, you can set a threshold that, once exceeded, invokes the UDP flood attack protection feature. (The default threshold value is 1000 packets per second.) If the number of UDP datagrams from one or more sources to a single destination exceeds this threshold, the security device ignores further UDP datagrams to that destination for the remainder of that second plus the next second as well.

Figure 123: UDP Flooding



To enable UDP flood protection, do either of the following, where the specified zone is that in which a flood might originate:

WebUI

Screening > Screen (Zone: select a zone name): Enter the following, then click **Apply**:

UDP Flood Protection: (select)
Threshold: (enter a value to trigger UDP flood protection)

NOTE: The value unit is UDP packets per second. The default value is 1000 packets per second.

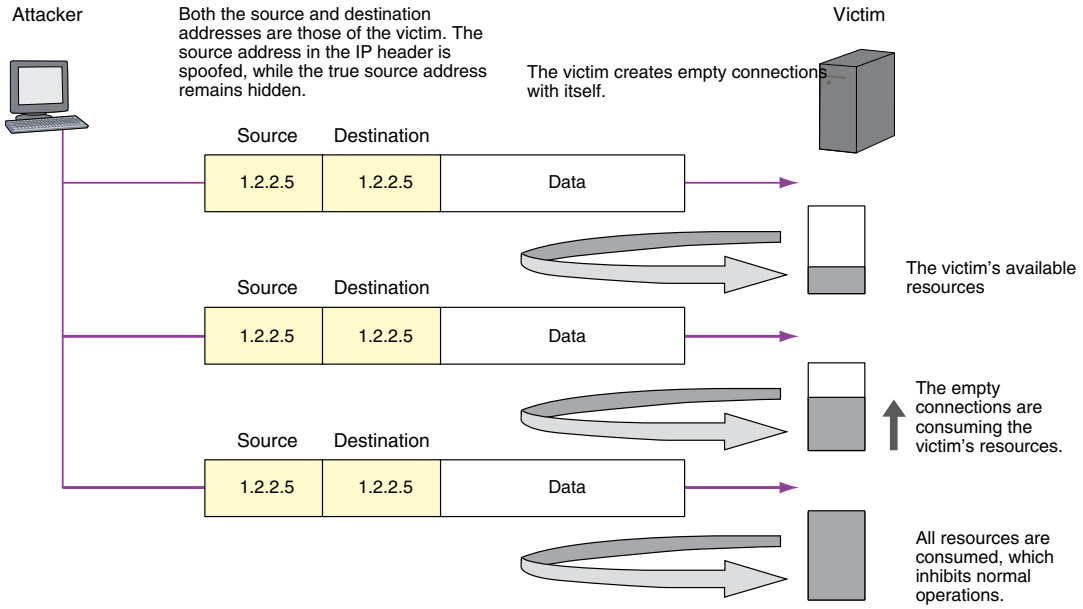
CLI

set zone *zone* screen udp-flood threshold *number*
set zone *zone* screen udp-flood

Land Attack

Combining a SYN attack with IP spoofing, a land attack occurs when an attacker sends spoofed SYN packets containing the IP address of the victim as both the destination and source IP address. The receiving system responds by sending the SYN-ACK packet to itself, creating an empty connection that lasts until the idle timeout value is reached. Flooding a system with such empty connections can overwhelm the system, causing a denial of service.

Figure 124: Land Attack



When you enable the SCREEN option to block land attacks, the security device combines elements of the SYN flood defense and IP spoofing protection to detect and block any attempts of this nature.

To enable protection against a land attack, do either of the following, where the specified zone is that in which the attack originates:

WebUI

Screening > Screen (Zone: select a zone name): Select **Land Attack Protection**, then click **Apply**.

CLI

set zone *zone* screen land

OS-Specific DoS Attacks

If an attacker not only identifies the IP address and responsive port numbers of an active host but also its operating system (OS), instead of resorting to brute-force attacks, he or she can launch more elegant attacks that can produce one-packet or two-packet kills. The attacks presented in this section can cripple a system with minimum effort. If your Juniper Networks security device is protecting hosts susceptible to these attacks, you can enable the security device to detect these attacks and block them before they reach their target.

Ping of Death

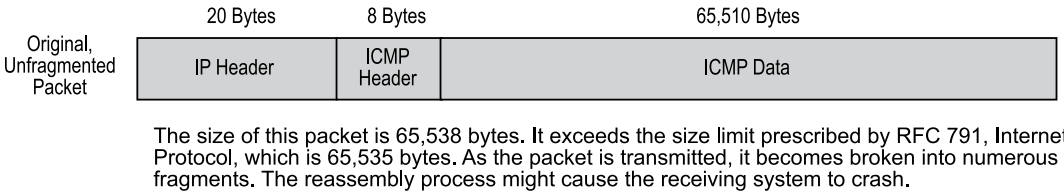
The maximum allowable IP packet size is 65,535 bytes, including the packet header, which is typically 20 bytes long. An ICMP echo request is an IP packet with a pseudo header, which is 8 bytes long. Therefore, the maximum allowable size of the data area of an ICMP echo request is 65,507 bytes (65,535 - 20 - 8 = 65,507).

However, many ping implementations allow the user to specify a packet size larger than 65,507 bytes. A grossly oversized ICMP packet can trigger a range of adverse system reactions such as denial of service (DoS), crashing, freezing, and rebooting.

When you enable the Ping of Death SCREEN option, the security device detects and rejects such oversized and irregular packet sizes even when the attacker hides the total packet size by purposefully fragmenting it.

NOTE: For information about IP specifications, see RFC 791, *Internet Protocol*.
For information about ICMP specifications, see RFC 792, *Internet Control Message Protocol*.
For information about ping of death attacks, see
http://insecure.org/sploits/ping-o-death-html

Figure 125: Ping of Death



To enable protection against a ping of death attack, do either of the following, where the specified zone is that in which the attack originates:

WebUI

Screening > Screen (Zone: select a zone name): Select **Ping of Death Attack Protection**, then click **Apply**.

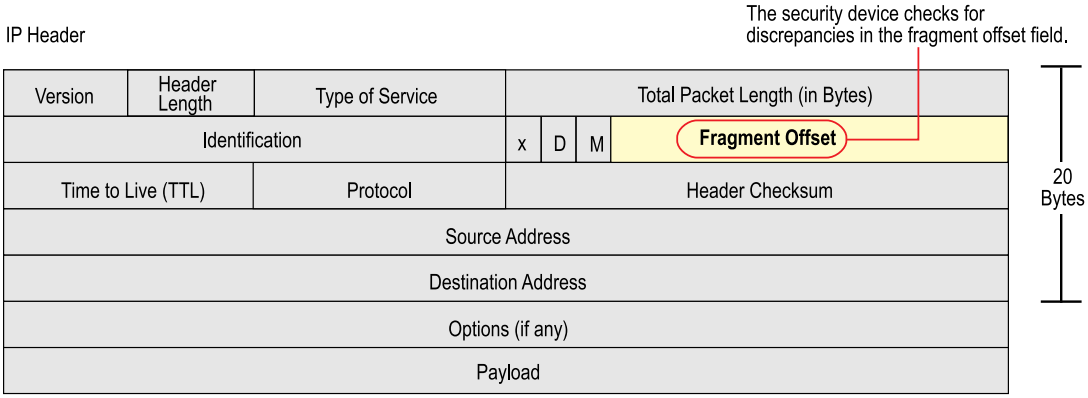
CLI

set zone *zone* screen ping-death

Teardrop Attack

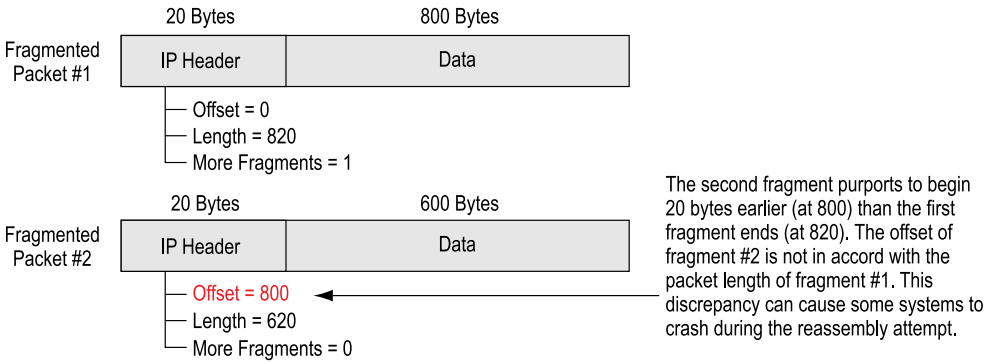
Teardrop attacks exploit the reassembly of fragmented IP packets. In the IP header, one of the fields is the fragment offset field, which indicates the starting position, or offset, of the data contained in a fragmented packet relative to the data of the original unfragmented packet.

Figure 126: Teardrop Attacks



When the sum of the offset and size of one fragmented packet differ from that of the next fragmented packet, the packets overlap, and the server attempting to reassemble the packet can crash, especially if it is running an older operating system that has this vulnerability.

Figure 127: Fragment Discrepancy



After you enable the Teardrop Attack SCREEN option, whenever the device detects this discrepancy in a fragmented packet, it drops it.

To enable protection against a Teardrop attack, do either of the following, where the specified zone is that in which the attack originates:

WebUI

Screening > Screen (Zone: select a zone name): Select **Teardrop Attack Protection**, then click **Apply**.

CLI

```
set zone zone screen tear-drop
```

WinNuke

WinNuke is a DoS attack targeting any computer on the Internet running Windows. The attacker sends a TCP segment—usually to NetBIOS port 139 with the urgent (URG) flag set—to a host with an established connection. This introduces a NetBIOS fragment overlap, which causes many machines running Windows to crash. After rebooting the attacked machine, the following message appears, indicating that an attack has occurred:

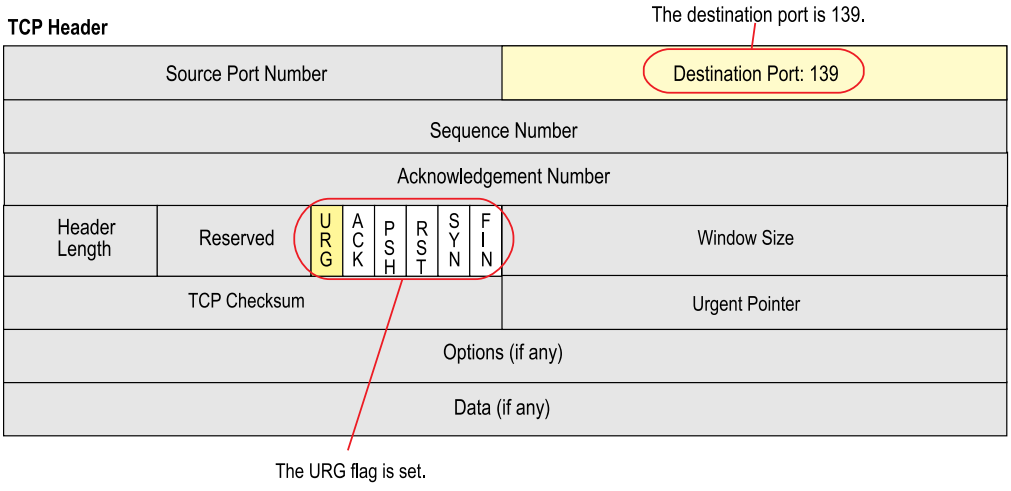
```
An exception OE has occurred at 0028:[address] in VxD MSTCP(01) +
000041AE. This was called from 0028:[address] in VxD NDIS(01) +
00008660. It may be possible to continue normally.

Press any key to attempt to continue.

Press CTRL+ALT+DEL to restart your computer. You will lose any unsaved information
 in all applications.

Press any key to continue.
```

Figure 128: WinNuke Attack Indicators



If you enable the WinNuke attack defense SCREEN option, the security device scans any incoming Microsoft NetBIOS session service (port 139) packets. If the device observes that the URG flag is set in one of those packets, it unsets the URG flag, clears the URG pointer, forwards the modified packet, and makes an entry in the event log noting that it has blocked an attempted WinNuke attack.

To enable protection against a WinNuke attack, do either of the following, where the specified zone is that in which the attack originates:

WebUI

Screening > Screen (Zone: select a zone name): Select **WinNuke Attack Protection**, then click **Apply**.

CLI

```
set zone zone screen winnuke
```

Chapter 15

Content Monitoring and Filtering

Juniper Networks provides broad protection and control of network activity through ScreenOS features and the pairing of ScreenOS with Websense, SurfControl, and Kaspersky Lab products.

This chapter describes how to configure the device to perform segment and packet reassembly, monitor HTTP traffic for malicious URLs, and communicate with other devices to perform antivirus (AV) scanning and Web filtering. This chapter contains the following sections:

Fragment Reassembly

Typically, a network-forwarding device such as a router or switch does not reassemble the fragmented packets that it receives. Usually the destination host reconstructs the fragmented packets when they all arrive. The main function of a forwarding device is the efficient delivery of traffic. If the forwarding device also needs to queue, reassemble, and refragment all packets, its efficiency is adversely affected. However, passing fragmented packets through a firewall is insecure. An attacker can intentionally break up packets to conceal traffic strings that the firewall otherwise would detect and block.

ScreenOS allows you to enable fragment reassembly for individual zones. This method allows the security device to expand its ability to detect and block malicious URL strings. Fragment reassembly occurs on Application Layer Gateway (ALG)-enabled traffic only if a device is configured for NAT.

Malicious URL Protection

In addition to the Web-filtering feature (explained in “Redirect Web Filtering” on page 551), you can define up to 48 malicious URL string patterns per zone, each of which can be up to 64 characters long, for malicious URL protection at the zone level. With the Malicious URL blocking feature enabled, the security device examines the data payload of all HTTP packets. If it locates a URL and detects that the beginning of its string—up to a specified number of characters—matches the pattern you defined, the device blocks that packet from passing through the firewall.

A resourceful attacker, realizing that the string is known and might be guarded against, can deliberately fragment the IP packets or TCP segments to make the pattern unrecognizable during a packet-by-packet inspection. For example, if the malicious URL string is **120.3.4.5/level/50/exec**, IP fragmentation might break up the string into the following sections:

- First packet: **120**
- Second packet: **3.4.5/level/50**
- Third packet: **/exec**

Individually, the fragmented strings can pass undetected through the security device, even if you have the string defined as **120.3.4.5/level/50/exec** with a length of 20 characters. The string in the first packet—"**120**." — matches the first part of the defined pattern, but it is shorter than the required length of 20 matching characters. The strings in the second and third packets do not match the beginning of the defined pattern, so they would also pass without being blocked.

However, if the packets are reassembled, the fragments combine to form a recognizable string that the device can block. Using the Fragment Reassembly feature, the device can buffer fragments in a queue, reassemble them into a complete packet, and then inspect that packet for a malicious URL. Depending on the results of this reassembly process and subsequent inspection, the device performs one of the following actions:

- If the device discovers a malicious URL, it drops the packet and enters the event in the log.
- If the device cannot complete the reassembly process, a time limit is imposed to age out and discard fragments.
- If the device determines that the URL is not malicious but the reassembled packet is too big to forward, the device fragments that packet into multiple packets and forwards them.
- If the device determines that the URL is not malicious and does not need to fragment it, it forwards the packet.

NOTE: The device can drop or forward the packets based on the reassembly process and subsequent inspection. When using the malicious URL protection feature, you cannot cannot make the device notify you about malicious traffic while allowing the traffic to pass. The alarm-without-drop option does not apply to this feature. For more information about this option, see "Exploit Monitoring" on page 436.

Application Layer Gateway

ScreenOS provides an Application Layer Gateway (ALG) for a number of protocols such as DNS, FTP, H.323, and HTTP. Of these, fragment reassembly can be an important component in the enforcement of policies involving FTP and HTTP services. The ability of the Juniper Networks firewall to screen packets for protocols such as FTP-Get and FTP-Put requires it to examine not only the packet header but also the data in the payload.

For example, there might be a policy denying FTP-Put from the Untrust to the DMZ zone:

```
set policy from untrust to dmz any any ftp-put deny
```

If the security device finds **STOR** *filename*, the client has sent a request to store the specified file on the server, and the device blocks the packet.

NOTE: For a deny policy, FTP-Put, FTP-Get, and FTP service behave the same way by blocking all packets.

In a permit policy, FTP-Get, FTP-Put, and FTP are all different services. For example, there might be a policy permitting FTP-Get from the Untrust to the DMZ zone.

```
set policy from untrust to dmz any any ftp-get permit
```

If the security device reads **RETR** *filename*, the FTP client has sent a request to retrieve the specified file from the FTP server, and the device allows the packet to pass.

If you have two policies, one denying FTP-Put from the Untrust to the DMZ zone and another permitting FTP-Get from the Untrust to the DMZ zone, then the device blocks the packet.

```
set policy from untrust to dmz any any ftp-put deny
set policy from untrust to dmz any any ftp-get permit
```

To thwart this defense, an attacker can deliberately fragment a single FTP-Put packet into two packets that contain the following text in their respective payloads:

- packet 1: **ST**
- packet 2: **OR** *filename*

When the security device inspects each packet individually, it does not find the string **STOR** *filename*, so would consequently allow both fragments to pass.

However, if the packets are reassembled, the fragments combine to form a recognizable string upon which the security device can act. Using the Fragment Reassembly feature, the device buffers the FTP fragments in a queue, reassembles them into a complete packet, and then inspects that packet for the complete FTP request. Depending on the results of this reassembly process and subsequent inspection, the device performs one of the following actions:

- If the device discovers an FTP-Put request, it drops the packet and enters the event in the log.
- If the device cannot complete the reassembly process, a time limit is imposed to age out and discard fragments.
- If the device discovers an FTP-Get request but the reassembled packet is too big to forward, the device fragments that packet into multiple packets and forwards them.
- If the device discovers an FTP-Get request and does not need to fragment it, the device then forwards the packet.

Example: Blocking Malicious URLs in Packet Fragments

In this example, you define the following three malicious URL strings and enable the malicious URL blocking option:

- Malicious URL 1
  - ID: Perl
  - Pattern: scripts/perl.exe
  - Length: 14
- Malicious URL 2
  - ID: CMF
  - Pattern: cgi-bin/phf
  - Length: 11
- Malicious URL 3
  - ID: DLL
  - Pattern: 210.1.1.5/msadcs.dll
  - Length: 18

The values for length indicate the number of characters in the pattern that must be present in a URL—starting from the first character—for a positive match. Note that for 1 and 3, not every character is required.

You then enable fragment reassembly for the detection of the URLs in fragmented HTTP traffic arriving at an Untrust zone interface.

WebUI

Security > Screening > Mal-URL (Zone: Untrust): Enter the following, then click **OK**:

ID: perl
Pattern: /scripts/perl.exe
Length: 14

Security > Screening > Mal-URL (Zone: Untrust): Enter the following, then click **OK**:

ID: cmf
Pattern: cgi-bin/phf
Length: 11

Security > Screening > Mal-URL (Zone: Untrust): Enter the following, then click **OK**:

ID: dll
Pattern: 210.1.1.5/msadcs.dll
Length: 18

Network > Zones > Edit (for Untrust): Select the TCP/IP Reassembly for ALG check box, then click **OK**.

CLI

```
set zone untrust screen mal-url perl “get /scripts/perl.exe” 14
set zone untrust screen mal-url cmf “get /cgi-bin/phf” 11
set zone untrust screen mal-url dll “get /210.1.1.5/msadcs.dll” 18
set zone untrust reassembly-for-alg
save
```

Antivirus Scanning

A virus is executable code that infects or attaches itself to other executable code in order to reproduce itself. Some malicious viruses erase files or lock up systems, while other viruses merely infect files and can overwhelm the target host or network with bogus data.

Juniper Networks supports internal and external antivirus (AV) scanning on select security devices. Refer to the *ScreenOS Release Notes* for a list of security devices and the supported AV scan engine.

You have the following two antivirus solutions for the ISG series of products:

- Internet Content Adaptation Protocol (ICAP) AV

  Use this solution for lower speeds, such as in T-3 or fractional T-3 deployments. For more details, see “External AV Scanning” on page 499.

- Policy based routing (PBR)

  Use this solution for higher speeds, such as in OC-3 deployments. In this scenario, PBR on the ISG offloads specific traffic to a high-end security device running the embedded AV scanner (internal AV scanner). For more details on this configuration, see “Advanced PBR Example” on page 1384. For more details on the embedded AV scanner, see “Internal AV Scanning” on page 501.

External AV Scanning

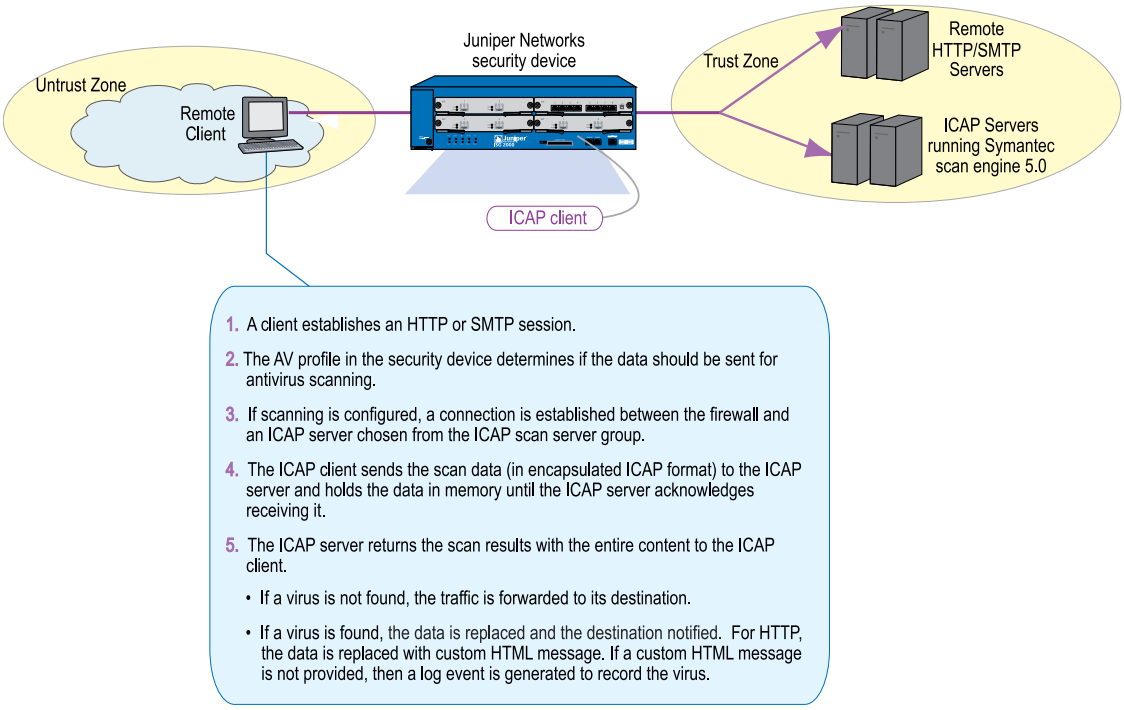
External AV scanning occurs when the security device redirects traffic to an external ICAP AV scan server. The ICAP client on the security device communicates with the external ICAP scan server to provide the following features:

- Supports ICAP v1.0 and is fully compliant with RFC 3507
- Supports Symantec Scan Engine 5.0 ICAP server
- Scalable antivirus scanning (add additional ICAP scan servers)
- Multiple security devices (firewalls) share the same ICAP scan server
- Load balancing traffic among a set of ICAP servers
- Encapsulation of HTTP and SMTP traffic
- Supports custom HTML message for HTTP traffic

- Supports custom x-response header
- Supports persistent connection to the same ICAP server
  Persistent connection reduces processing overhead and enhances AV scanning throughput.

Figure 129 on page 500 illustrates how external AV scanning works with the security device.

Figure 129: How External Scanning Works



Scanning Modes

After the traffic undergoes AV scanning, the ICAP server running Symantec Scan Engine 5.0 provides one of the following results:

- **Scan only**. Denies access to the infected file but does nothing to the file.
- **Scan and delete.** Deletes all infected files, including files that are embedded in archive files, without attempting to repair.
- **Scan and repair files.** Attempts to repair infected files but does nothing to files that cannot be repaired.
- **Scan and repair or delete.** Attempts to repair infected files and deletes any unrecoverable files from archive files.

Refer to your ICAP server documentation for more information about scanning behavior and results.

Load-Balancing ICAP Scan Servers

ScreenOS external AV scanning allows you to load-balance ICAP scan servers configured in an ICAP server group. The load-balancing algorithm used among the ICAP scan servers in the group is least request. The ICAP servers are load-balanced based upon the server's health and traffic volume (number of pending requests). Unhealthy servers are bypassed, and traffic is reduced automatically to the overloaded server.

A configured ICAP server can be in either an enabled or a disabled state. The status of an enabled ICAP server can be *in-service* or *out-of-service*. When an ICAP server is configured as disabled, then the server is not used to serve new requests.

ICAP servers are monitored through a probing mechanism. For example, if the probe interval is set to 30, then an enabled ICAP server is automatically probed every 30 seconds to determine its status (in-service or out-of-service).

An auto probe returns an out-of-service result for the following conditions:

- Firewall cannot establish a successful TCP connection to an ICAP server
- Invalid ICAP server AV license
- Client-side error response for ICAP options request
- Server-side error response for ICAP options request

The server goes into an out-of-service state when three consecutive probes fail.

Internal AV Scanning

Internal AV scanning is performed when the scan engine in the security device scans traffic for viruses. The internal or embedded scan engine is a Juniper-Kaspersky scan engine.

NOTE: The internal AV scanner requires you to install an AV license on your security device. An AV license is not required if you are using an external AV scanner.

The embedded scan engine allows you to do the following:

- Enable/disable scanning based on file extension and content type

  For example, you can set up a profile that supports scanning of executable files (.exe) but not documentation files (.doc or .pdf).

- Configure decompression layers for specific application protocols

  In each profile, you can configure different decompression levels for each protocol (HTTP/SMTP/POP3/IMAP/FTP). Based on your network environment, for example, you might want to specify the number of embedded zip files to unpack for each protocol.

- Configure AV scanning for Instant Messaging (IM) traffic

For more information about scanning IM traffic, see "AV Scanning of IM Traffic" on page 502.

- Configure email notification to sender/receiver on detected virus and scanning errors
- Configure scanning levels to provide spyware and phishing protection

  The Juniper-Kaspersky scan engine, by default, provides the highest level of security. In addition to stopping all viruses (including polymorphic and other advanced viruses), this scan engine also provides inbound spyware and phishing protection.

  Spyware protection. The spyware-protection feature adds another layer of protection to Juniper Networks anti-spyware and anti-adware solutions by letting you block incoming spyware, adware, keyloggers, and related malware to prevent the malicious traffic from penetrating your enterprise.

  This solution complements Juniper Networks Intrusion Detection and Prevention (IDP) products, which provide spyware phone-home protection (that is, stopping spyware from sending sensitive data if your laptops/desktops/servers are infected).

  Phishing protection. The phishing protection feature lets you block emails that try to entice users to fake (phishing) sites that steal sensitive data.

  You can change the default security level of scanning by choosing one of the following two options:

AV Scanning of IM Traffic

An Instant Messaging (IM) network is composed of clients and servers, along with the protocols needed to connect them.

IM Clients

Each IM client has three major components:

- A buddy list or other roster of friends with whom you wish to communicate.
- A separate window that shows the text chats in progress—Users type their messages and view their correspondents' responses in this window.
- Additional features for video and audio chats and for file transfers between users.

  All major IM clients are moving beyond simple text chats to more integrated and sophisticated communications, including real-time voice and video calls.

ScreenOS supports scanning of popular public IM applications such as:

- AOL Instant Messenger (AIM)
- I Seek You (ICQ)
- Yahoo! Messenger (YMSG)
- MSN Messenger

The AV scanning features in this release of ScreenOS apply to the following IM services:

- Text chat message
- Group chat message
- File transfer/file sharing

IM Server

The IM server maintains the directory of user accounts, keeps track of who is online, and, in most cases, routes messages among users. The IM server operates in real time, sending messages back and forth between two users as they finish typing each line of text. The servers also pass real-time information about the availability of various users in the directory, such as when they come online and change their status message.

Each IM server communicates with its clients over an assigned port number across the Internet. But IM clients however, can login using other ports when the default port is blocked by a deny policy. Typical port numbers include those shown in the following table:

IM Application	Service Port Numbers	Proxies
AIM	5190	SOCKS 4, SOCKS 5, HTTP, HTTPS
ICQ	5190	
YMSG	5050[1] (443 and 80)	SOCKS 4, SOCKS 5, HTTP
MSN Messenger	1863	SOCKS 4, SOCKS 5, HTTP

1.In addition to port 5050, make sure traffic is permitted on ports 443 (HTTPS) and 80 (HTTP).

NOTE: AV scanning is not supported for AIM or ICQ traffic communicating in encrypted format.

IM Protocols

The IM network employs a client-server model for authentication to the service and for communication with other clients using the protocols shown in the following table:

IM Application	Supported Protocol
AIM/ICQ	Open System for Communication in Realtime protocol (OSCAR)
YMSG	Yahoo Messenger Service Gateway Protocol (YMSG)
MSN Messenger	Mobile Status Notification Protocol (MSNP)

Because the proprietary protocol for the respective IM applications is constantly being updated, ScreenOS provides a configurable parameter to control the firewall behavior. Refer to the software release notes for the supported client and protocol version. ScreenOS, however, processes traffic for unsupported versions of the protocol in one of the following two ways:

- Best Effort: Uses the existing protocol knowledge to process the traffic
- Pass: Passes the traffic without scanning it

Instant Messaging Security Issues

Generally, worms spread over instant messaging services and appear as a URL. These URLs are accessed because they appear from someone on your buddy list. If the URL is clicked, the worm infects your PC and spreads to everyone on the buddy list.

The buddy list also leads to social engineering. Social engineering occurs when people obtain information from legitimate users of a computer system—specifically, information that will allow them to gain unauthorized access to a particular system.

The file transfer service is another security risk where instant messaging applications can send Trojans and viruses.Update for Qian: This appears in 6.0 and 6.1 but was hidden because it was not supported at 6.0. Was it supported for 6.1? If so add. Add for 6.2 if it is supported in 6.2.

NOTE: Unsolicited email (SPAM), referred to as SPIM in the Instant Messaging network, often contains links to offensive websites. These messages are more intrusive than SPAM email, because IM clients alert users when new instant messages arrive.

IM Security Issues

Instant messaging (IM) services are vulnerable to attacks such as viruses, worms, and Trojans via the following methods:

- Buddy lists

  A worm can spread through IM services because it generally appears as URL in an instant message. These URLS often appear to come from someone on your buddy list. If you click such a malicious URL, the worm infects your PC and can easily spread to everyone on your buddy list.

- Social engineering

  Social engineering occurs when an attacker illegally obtains sensitive information (such as a buddy list) from legitimate users of a system or service—information the attacker then uses to gain unauthorized access.

- File transfers

  Trojans and viruses can easily spread when files are sent from one user to another via an IM session.

Scanning Chat Messages

When the device is enabled for AV, the firewall processes the data packets sent between the IM client and the server. The firewall detects the beginning of an individual chat message in a data packet and retains the data packets that follow until the chat message is complete. The complete message is sent to the AV scan engine for virus scanning using the procedure shown in the following table.

If...	The Chat Message	Result
Virus is found	Is dropped.	A virus drop notification message is forwarded to the original message's destination.
Scanning error occurs (scan error permit is disabled)	Is dropped.	A scan-error drop notification message is forwarded to the original message's destination.
AV scanning finishes with no virus or scanning errors	Is forwarded to its destination.	Message reaches destination.

NOTE: In an AOL Instant Message (AIM) session, if a group chat message includes a virus, the drop message is sent back to the client, after which the client is unable to send any more messages.

Scanning File Transfers

The firewall processes the data packets communicated between the IM client and the server. Typically, file sharing means get file, but AIM file transfer includes send file, get file, and send directory. When the firewall detects file transfer commands, the following occurs:

If File Size Is...	File Transfer/File Sharing	Result
< = AV max_content_size	AV scanning occurs	■ Virus found. File content is replaced by virus notification message. ■ Scanning error (AV scan error permit is disabled). File content is replaced by scan-error notification message.
> AV max_content_size (max_content_size drop is enabled)	Skips AV scanning	Drops the file and forwards drop-notification message to original message's destination.
> AV max_content_size (max_content_size drop is disabled)	Skips AV scanning	Forwards file to its destination.

NOTE: This release of ScreenOS does not support instant messaging P2P traffic through the firewall.

AV Scanning Results

AV scanning may not occur for several reasons. When your device is configured for external scanning, the device simply redirects the traffic to the external ICAP server. Refer to your ICAP server documentation for information about AV scanning behavior and results.

If your device is configured for internal AV scanning, the **get av stat** command displays scanning failures. In addition to the following scan-code results, this command generates an event log that contains more information about scanning results.

Scan Code: Clear
Scan Code: Infect
Scan Code: Psw Archive File
Scan Code: Decompress Layer
Scan Code: Corrupt File
Scan Code: Out Of Resource
Scan Code: Internal Error
Scan Code: Error
Scan Eng: Error:

Fail Mode:

In the following scenarios, traffic is dropped and replacement data is forwarded to its destination:

- AV scan engine returns one of the following scan-errors and the corresponding configuration drop setting is enabled.
  - Max-decompress-layer setting is exceeded.
  - Password protected file.
  - Corrupted file.
  - Out of resource.

    When AV scan results in an out of resource error condition, the file is dropped or passed based on the max-content-size setting, but the out-of-resource counter is incremented.
- AV scan engine returns any of the above scan-errors and fail-mode permit is disabled.
- Size of file transfer exceeds the AV max-content-size setting and max-content-size drop is enabled.

NOTE: If the file to be sent exceeds the max-content-size and the fail mode is drop, both ends will receive the message of File transfer error.

NOTE: If the max-decompress-layer setting is set to drop the data packets on exceeding the decompress layer setting, a replacement file is sent to the receiver. The MSN Server however, does not send a replacement file or an error message about the download failure.

See “Scanning Application Protocols” on page 509 for information about AV-scanning failure, including those instances when data cannot be successfully scanned.

Refer to the *ScreenOS Message Log Reference Guide* for a list of error messages generated from AV scanning.

Policy-Based AV Scanning

AV scanning profiles increase the flexibility and granularity of AV scans. Profile-based scanning allows you to configure a profile to scan traffic and assign the profile to a policy. Policy-based scanning allows you to:

- Select specific data traffic for AV scanning
- Enhance performance and control the AV scan engine

To configure policy-based scanning, you must configure AV profiles for use in policies by doing the following:

1. Initiate an AV profile context. For more information, see “Initiating an AV Profile for Internal AV” on page 528.
2. Configure a profile (*ns-profile* is predefined for internal AV) to examine network traffic for the protocols shown in the following table.

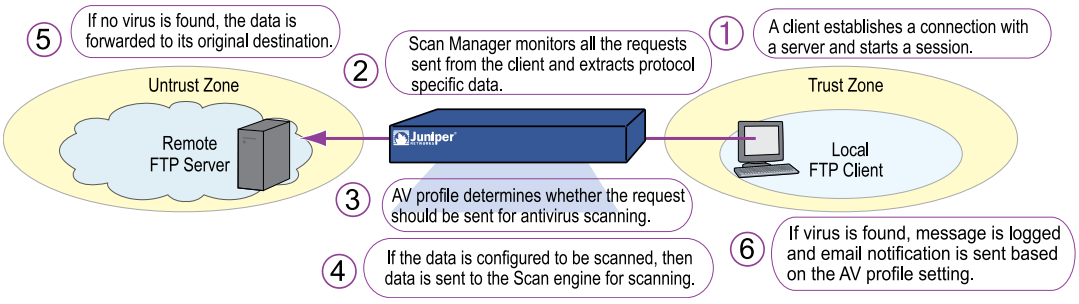
Protocols	See
File Transfer Protocol (FTP)	“Scanning FTP Traffic” on page 509
HyperText Transfer Protocol (HTTP)	“Scanning HTTP Traffic” on page 511
Internet Mail Access Protocol (IMAP)	“Scanning IMAP and POP3 Traffic” on page 513
Post Office Protocol, version 3 (POP3)	“Scanning IMAP and POP3 Traffic” on page 513
Simple Mail Transfer Protocol (SMTP)	“Scanning SMTP Traffic” on page 514
Internet Content Adaptation Protocol (ICAP)	“Redirecting Traffic to ICAP AV Scan Servers” on page 516

3. Exit the AV profile context.
4. Assign the AV profile to a firewall policy. (Only one AV profile can be linked to a policy.)

   To apply AV protection, you reference the AV profile in a security policy. When the security device receives traffic to which a policy requiring AV scanning applies, it directs the content it receives to the AV scanner (internal or external).
5. Save your profile.

Figure 130 on page 508 shows how the AV profile works with the AV scanner (internal or external).

Figure 130: How the AV Profile Works with the AV Scanner How the AV Profile Works with the AV Scanner



Scanning Application Protocols

The internal embedded AV scan engine supports scanning for specific Application Layer transactions allowing you to select the content (FTP, HTTP, IMAP, POP3, or SMTP traffic) to scan. For example, scan performance can be enhanced by not scanning certain content. Similarly, external AV scanning is supported for HTTP and SMTP protocols only.

NOTE: You need to assess the risk and determine the best trade-off between security and performance.

This section discusses how to configure the following application protocols for AV scanning:

- "Scanning FTP Traffic" on page 509
- "Scanning HTTP Traffic" on page 511
- "Scanning IMAP and POP3 Traffic" on page 513
- "Scanning SMTP Traffic" on page 514

Each of the above applications can be configured for one or more of the following:

Command	Description
decompress-layer	Specifies how many layers of nested compressed files the internal AV scanner can decompress before it executes the virus scan.
extension list	Specifies the extension list name (*string*) to include or exclude defined extensions.
scan-mode	Specifies how the scan engine scans traffic for a specific protocol.
timeout	Specifies the timeout value for an AV session for a specific protocol.
http skipmime	Skips the specified MIME list from AV scanning. **Note:** Disabling **skipmime** allows the security device to scan all kinds of HTTP traffic regardless of MIME content types.
email-notify	Notifies sender or recipient of detected virus or scanning errors for IMAP, POP3, and SMTP traffic only.
virus-detection-notify-method	Specifies how the scan engine notifies the sender or the recipient about a detected virus for FTP, HTTP, IMAP, POP3, and SMTP traffic.

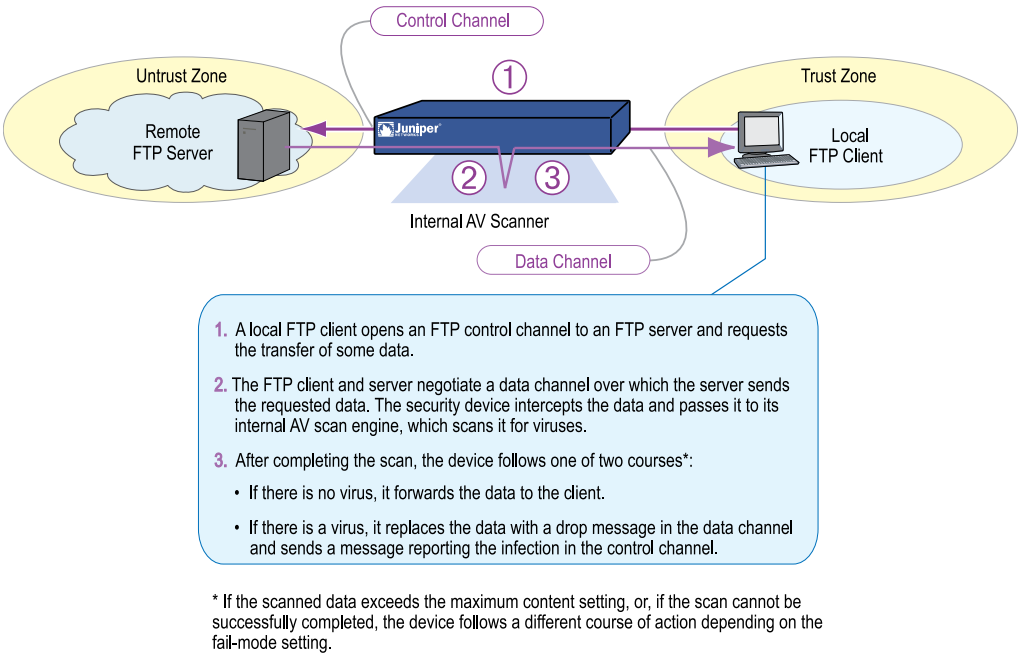
Scanning FTP Traffic

For File Transfer Protocol (FTP) traffic, the security device monitors the control channel and, when it detects one of the FTP commands for transferring data (RETR, STOR, STOU, APPE, or NLST), it scans the data sent over the data channel.

Depending on the results of the scan and how you have configured the fail-mode behavior, the security device takes one of the following actions:

If the Data	And	The Security Device
is uncontaminated		passes the data to the FTP client through the data channel
contains a virus		replaces the data with a warning message or drops the data. In both cases, a message with a URL link that describes the virus appears in the event log.
exceeds the maximum content size	**drop** is set	drops the data from the data channel and sends a "file too large" message to the FTP client through the control channel
exceeds the maximum content size	**drop** is unset	passes the unexamined data to the FTP client through the data channel
cannot successfully be scanned	**fail mode** is unset (drop)	drops the data from the data channel and sends a "scan error" message to the FTP client through the control channel
cannot successfully be scanned	**fail mode** is permit (**traffic permit** is set)	passes the data to the FTP client through the data channel
exceeds the maximum concurrent messages	**drop** is set	drops the data from the data channel and sends an "exceeding maximum message setting" message to the FTP client through the control channel
exceeds the maximum concurrent messages	**drop** is unset	passes the data to the FTP client through the data channel

Figure 131: Antivirus Scanning for FTP Traffic



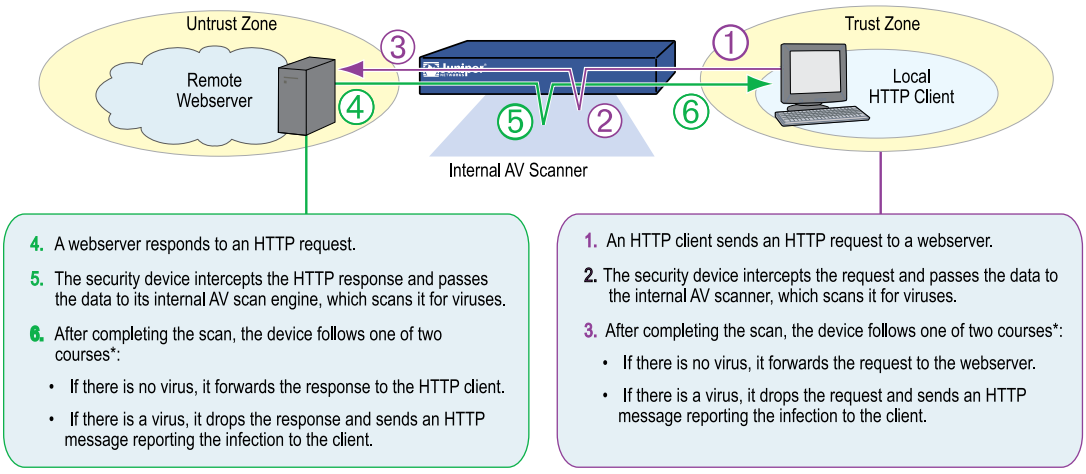
Scanning HTTP Traffic

For HTTP traffic, the security device scans both HTTP responses and requests (**get**, **post**, and **put** commands). The internal AV scanner examines HTTP downloads, that is, HTTP data contained in responses from a Web server to HTTP requests from a client. The internal AV scanner also scans uploads, such as when an HTTP client completes a questionnaire on a Web server or when a client writes a message in an email originating on a Web server.

Depending on the results of the scan and how you have configured the fail-mode behavior, the security device takes one of the following actions:

If the Data	And	The Security Device
is uncontaminated		passes the data to the HTTP client
contains a virus		replaces the data with a warning message or drops the data. In both cases, a message with a URL link that describes the virus appears in the event log.
exceeds the maximum content size	**drop** is set	drops the data and sends a "file too large" message to the HTTP client
exceeds the maximum content size	**drop** is unset	passes the data to the HTTP client
cannot successfully be scanned	**fail mode** is unset (drop)	drops the data and sends a "scan error" message to the HTTP client
cannot successfully be scanned	**traffic permit** is set (**fail mode** is permit)	passes the data to the HTTP client
exceeds the maximum concurrent messages	**drop** is set	drops the data from the data channel and sends an "exceeding maximum message setting" message to the HTTP client through the control channel
exceeds the maximum concurrent messages	**drop** is unset	passes the data to the HTTP client through the data channel

Figure 132: Antivirus Scanning for HTTP Traffic



* If the scanned data exceeds the maximum content setting, or, if the scan cannot be successfully completed, the security device follows a different course of action depending on the fail-mode setting.

HTTP MIME Extensions

By default, HTTP scanning does not scan HTTP entities composed of any of the following Multipurpose Internet Mail Extensions (MIME) content types and subtypes (when present following a slash):

- Application/x-director
- Application/pdf
- Image/
- Video/
- Audio/
- Text/css
- Text/html

To improve performance, Juniper Networks security devices do not scan the above MIME content types. Because most HTTP entities are made up of the above content types, HTTP scanning only applies to a small subset of HTTP entities that are most likely to contain viruses, such as the application/zip and application/exe content types.

To change HTTP scanning behavior so that the security device scans all HTTP traffic regardless of its MIME content type:

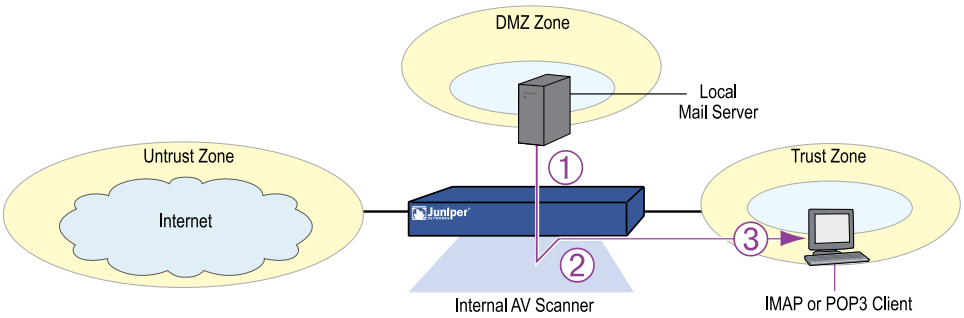
```
set av profile
jnpr-profile
(av:jnpr-profile)-> unset av http skipmime
(av:jnpr-profile-> exit
save
```

Scanning IMAP and POP3 Traffic

For IMAP and POP3 traffic, the security device redirects traffic from a local mail server to the internal AV scanner before sending it to the local IMAP or POP3 client. Depending on the results of the scan and how you have configured the fail-mode behavior, the security device takes one of the following actions:

If the Data	And	The Security Device
is uncontaminated		passes the message to the IMAP or POP3 client.
contains a virus		replaces the data with a warning message or drops the data. In both cases, a message with a URL link that describes the virus appears in the event log.
	email notification is set	changes the content type to text/plain, replaces the body of the message with a default warning message along with the following notice and notifies the sender by email: virus description http://www.viruslist.com/en/search?VN=<*name of a malware/virus*> For more information about email notification, see "AV Notify Mail" on page 523.
exceeds the maximum content size or cannot successfully be scanned or exceeds the maximum concurrent messages	**drop** is set *fail mode* is unset (drop) *email notification* is set	changes the content type to text/plain, replaces the body of the message with the following notice, and sends it to the IMAP or POP3 client: Content was not scanned for viruses because reason_text_str (code number), and it was dropped. reason_text_str can be one of the following: ■ The file was too large. ■ An error or a constraint was found. ■ The maximum content size was exceeded. ■ The maximum number of messages was exceeded. notifies the sender/recipient of detected virus or scanning errors.
exceeds the maximum content size or cannot successfully be scanned or exceeds the maximum concurrent messages	**drop** is unset **traffic permit** is set (**fail mode** is permit) **drop** is unset *email notification* is set	passes the original message to the IMAP or POP3 client with the original subject line modified as follows: *original_subject_text_str* (No virus check because *reason_text_str*, code *number*) notifies the sender/recipient of detected virus or scanning errors.

Figure 133: Antivirus Scanning for IMAP and POP3 Traffic



Scanning SMTP Traffic

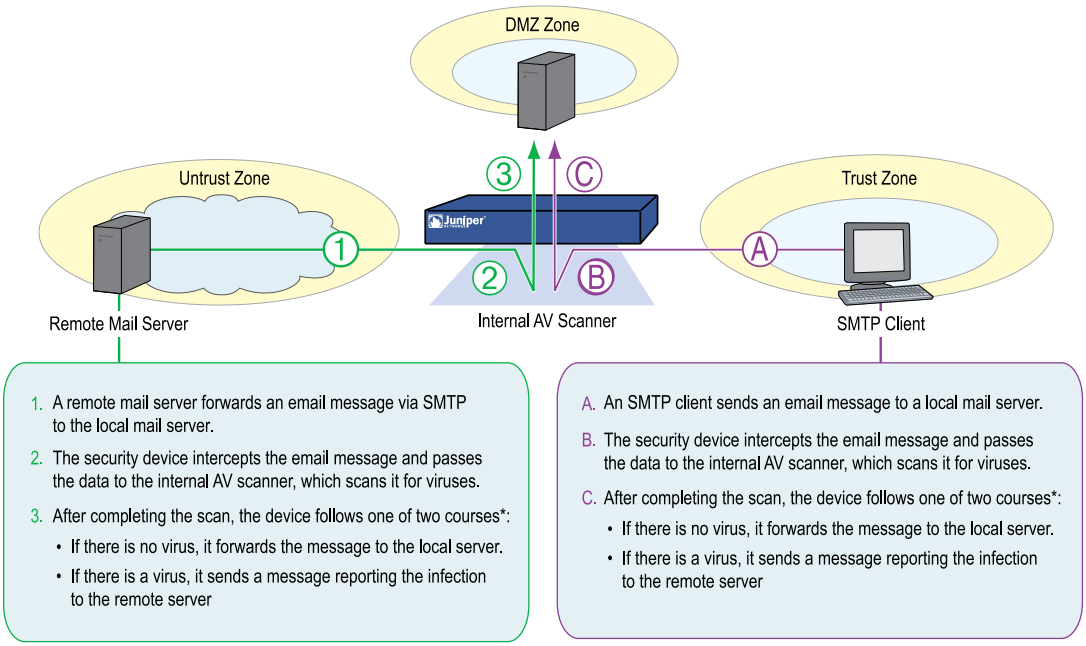
For SMTP traffic, the security device redirects traffic from local SMTP clients to the internal AV scanner before sending it to the local mail server. Depending on the results of the scan and how you have configured the fail-mode behavior, the security device takes one of the following actions:

If the Data	And	The Security Device
is uncontaminated		passes the message to the SMTP recipient.
contains a virus	**email notification** is set	replaces the data with a warning message or drops the data. In both cases, a message with a URL link that describes the virus appears in the event log. changes the content type to text/plain, replaces the body of the message with a default warning message along with the following notice and notifies the sender by email: virus description http://www.viruslist.com/en/search?VN=<*name of a malware/virus*> For more information about email notification, see “AV Notify Mail” on page 523.

If the Data	And	The Security Device
exceeds the maximum content size or cannot successfully be scanned or exceeds the maximum concurrent messages	**drop** is set **fail mode** is unset (drop) **drop** is set **email notification** is set	changes the content type to text/plain, replaces the body of the message with the following notice, and sends it to the SMTP recipient: Content was not scanned for viruses because *reason_text_str* (code *number*), and it was dropped. *reason_text_str* can be one of the following: ■ The file was too large. ■ An error or a constraint was found. ■ The maximum content size was exceeded. ■ The maximum number of messages was exceeded. notifies the sender/recipient of detected virus or scanning errors.
exceeds the maximum content level or cannot successfully be scanned or exceeds the maximum concurrent messages	**drop** is disabled **traffic permit** is set (fail mode is permit) **drop** is unset **email notification** is set	passes the original message to the SMTP recipient with the original subject line modified as follows: *original_subject_text_str* (No virus check because *reason_text_str*, code *number*) notifies the sender/recipient of detected virus or scanning errors.

NOTE: Because an SMTP *client* refers to the entity that sends email, a client could, in fact, be another SMTP server.

Figure 134: Antivirus Scanning for SMTP Traffic



* If the scanned message exceeds the maximum content setting, or if the scan cannot be completed successfully, the security device follows a different course of action depending on the fail-mode setting.

Redirecting Traffic to ICAP AV Scan Servers

Your Juniper Networks security device communicates with an external AV scan engine using the Internet Content Adaptation Protocol (ICAP). ScreenOS 6.2 supports redirection of HTTP and SMTP traffic only.

To configure the security device to support external ICAP AV scanning, perform the following steps:

1. Use the **set icap** command to configure the external ICAP scan server.
2. Configure an ICAP profile and specify one or more of the following:

Command	Description
timeout	Specifies the timeout value for an AV session for a specific protocol (HTTP or SMTP).
http skipmime	Skips the specified files in the MIME list from AV scanning. **Note:** Disabling the skipmime list allows the security device to scan all kinds of HTTP traffic regardless of MIME content types.
email-notify	Notifies sender or recipient of detected virus or scanning errors for SMTP traffic only.

WebUI

Objects > Antivirus > ICAP Server > New: Enter the following, then click **Apply**:

ICAP AV Server Name: **ICAP _ Server1**
Enable: (select), Scan Server Name/IP: 1.1.1.1
Scan Server Port: 1344, Scan URL: /SYMCScan-Resp-AV
Probe Interval: 10, Max Connections:

CLI

```
set icap server icap_server1 host 1.1.1.1
save
```

The ICAP server is automatically enabled when it is set up.

Updating the AV Pattern Files for the Embedded Scanner

Internal AV scanning requires that you load a database of AV patterns onto the Juniper Networks security device and periodically update the pattern file.

Before you start updating the AV pattern files, make sure your device supports the following:

Prerequisites	Description
Valid AV license key	av_v2_key
Access to the Internet	Your device has a route to the internet
DNS and port settings	Verify your DNS setting and port 80
AV signature service	See "Subscribing to the AV Signature Service" on page 517

Subscribing to the AV Signature Service

To purchase a subscription for the AV signature service you must first register your device. For the life of the subscription, you can load the current version of the database and update it as newer versions become available. The procedure for initiating the AV signature service varies depending on one of the following:

- If you purchased a security device with AV functionality, you can load an AV pattern file for a short period after the initial purchase. You must, however, register the device and purchase a subscription for AV signatures in order to continue receiving pattern updates.
- If you are upgrading a current security device to use internal AV scanning, you must register the device and purchase a subscription for AV signatures before you can begin loading the AV pattern file. After completing the registration process, you must wait up to four hours before initiating the AV pattern file download.

NOTE: For more information about the AV signature service, see “Registration and Activation of Subscription Services” on page 300.

Updating AV Patterns from a Server Updating AV Patterns from a Server

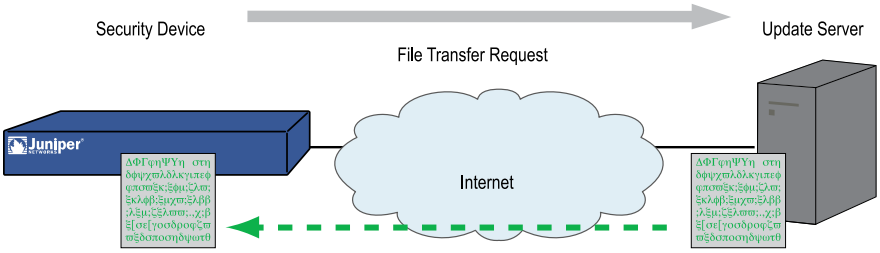
Figure 135 on page 518 and Figure 136 on page 519 illustrate how the pattern file is updated from the Juniper Networks server.

Update the AV pattern file as follows:

1. On the security device, specify the URL of the pattern update server:

   http://update.juniper-updates.net/av/5gt/

Figure 135: Updating Pattern Files—Step 1

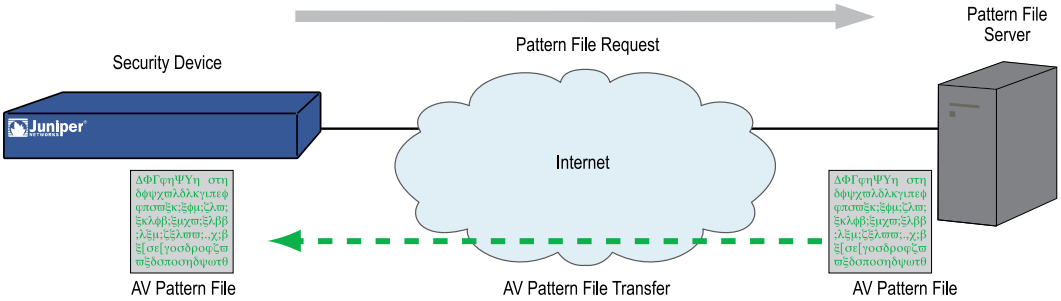


2. After the security device downloads the server-initialization file, the device checks that the pattern file is valid. The device then parses the file to obtain information about it, including the file version, size, and location of the pattern update server.

NOTE: ScreenOS contains a CA certificate for authenticating communications with the pattern update server.

3. If the pattern file on the security device is out of date (or nonexistent because this is the first time you are loading it), and, if the AV pattern update service subscription is still valid, the device automatically retrieves an updated pattern file from the pattern update server.

Figure 136: Updating Pattern Files—Step 2



4. The device saves the new pattern file to flash memory and RAM and, if there is an existing file, overwrites it.
5. If the **mail-to-admin** command is set, the device notifies the administrator via e-mail when an updated pattern file is available. You can enable this command in the WebUI (Security > Antivirus > Scan Manager: Click **Send Admin E-mail after Pattern Update** in the Pattern Type pane) or with the CLI command **set av scan-mgr pattern-up mail-to-admin**. By default, this feature is disabled.

NOTE: You can edit the source address of an e-mail. For more information, see "AV Notify Mail" on page 523.

Updates to the pattern file are added as new viruses propagate. You can configure the security device to regularly update the pattern file automatically, or you can update the file manually.

NOTE: Once your subscription expires, the update server no longer permits you to update the AV pattern file.

Example: Automatic Update

In this example, you configure the security device to update the pattern file automatically every 120 minutes. (The default AV pattern-update interval is 60 minutes.) For example, if the pattern-update server is located at the URL http://update.juniper-updates.net/av/5gt/, you configure automatic update as follows:

WebUI

Security > Antivirus > Scan Manager: Enter the following, then click **Apply**:

Pattern Update Server: http://update.juniper-updates.net/av/5gt

Auto Pattern Update: (select), Interval: 120 minutes (10~10080)

CLI

```
set av scan-mgr pattern-update-url http://update.juniper-updates.net/av/5gt
interval 120
save
```

Example: Manual Update

In this example, you update the pattern file manually. The pattern update server is located at the following URL:

http://update.juniper-updates.net/av/5gt/

WebUI

Security > Antivirus > Scan Manager: Enter the following, then click **Apply**:

Pattern Update Server: http://update.juniper-updates.net/av/5gt
Update Now: (select)

CLI

```
exec av scan-mgr pattern-update
```

The **set** command is not required because the URL is the default.

Updating AV Patterns from a Proxy Server Updating AV Patterns from a Proxy Server

You can update the AV patterns from the proxy server. This update does not require Internet access and is done offline.

To configure a proxy server:

WebUI

Security > Proxy: Set the HTTP and SSL proxy addresses, then click **Apply:**

HTTP Proxy: 10.0.0.5:8080
SSL Proxy: 10.0.0.5:443

CLI

```
set pattern-update proxy http 10.0.0.5:8080
save
```

NOTE: You cannot configure an HTTPS proxy, because you cannot cache an HTTPS proxy.

AV Scanner Global Settings

You can modify AV scanner settings to serve the needs of your network environment. The global **scan-mgr** command in the CLI configures the Scan Manager, which is the AV component that interacts with the scan engine. For example, the **set** or **get av scan-mgr** CLI command sets the global commands that control parameters such as max-content-size, max-msgs, pattern-type, pattern-update, and queue-size.

The following sections explain the global settings for your AV scanner:

- "AV Resource Allotment" on page 521
- "Fail-Mode Behavior" on page 522
- "Maximum Content Size and Maximum Messages (Internal AV Only)" on page 524
- "HTTP Keep-Alive" on page 525
- "HTTP Trickling (Internal AV Only)" on page 525
- "AV Warning Message" on page 522
- "AV Notify Mail" on page 523

Use the **get av all** or **get av scan-mgr** to see the global antivirus settings on the device.

AV Resource Allotment

A malicious user might generate a large amount of traffic all at once in an attempt to consume all available resources and hinder the ability of the AV scanner to scan other traffic. To prevent such activity from succeeding, the Juniper Networks security device can impose a maximum percentage of AV resources that traffic from a single source can consume at one time. The default maximum percentage is 70 percent. You can change this setting to any value between 1 and 100 percent, where 100 percent does not impose any restriction on the amount of AV resources that traffic from a single source can consume.

WebUI

NOTE: You must use the CLI to configure this option.

CLI

```
set av all resources number
```

```
unset av all resources
```

The above **unset av** command returns the maximum percentage of AV resources per source to the default (70 percent).

Fail-Mode Behavior

Fail-mode is the behavior that the security device applies when it cannot complete a scan operation—either to permit the unexamined traffic or to block it. By default, if a device cannot complete a scan, it blocks the traffic that a policy with antivirus checking enabled permits. You can change the default behavior from block to permit.

When the AV scan engine is scanning a file and runs out of memory (typically, when decompressing files), the content is either dropped or passed based on the out of resource (set av scan-mgr out-of-resource) setting, instead of the fail-mode setting.

WebUI

Security > Antivirus > Global: Select **Fail Mode Traffic Permit** to permit unexamined traffic, or clear it to block unexamined traffic, then click **Apply**.

CLI

```
set av all fail-mode traffic permit
unset av all fail-mode traffic
```

The above **unset av** command returns the fail mode behavior to the default (block unexamined traffic).

AV Warning Message

When the AV scanner detects a virus, it can send an AV warning message to the client that initiated the traffic. ScreenOS allows you to customize the warning message for FTP/HTTP/POP3/IMAP/SMTP protocols.

When a virus is detected, the AV scanner appends the customized warning message to the default message and the device sends the message to the client. If you do not set a customized message, the AV scanner sends only the default warning message.

WebUI

Security > Antivirus > Global: Enter or edit the warning message that the AV scanner sends to the client, then click **Apply**.

CLI

```
set av warning-message string
```

```
unset av warning-message
```

AV Notify Mail

The AV scanner sends an AV notification mail to the client when it detects a virus. By default, the AV scanner uses the IP address of the security device and the default mail subject. You can configure the AV scanner to use a customized source address and mail subject by using the **notify-mail-source** and **notify-mail-subject** commands, respectively.

If the notify mail includes Japanese or other characters, you can specify the character set to be used to display the notification mail. You configure the character set using the CLI or WebUI. For example, if the notify mail includes Japanese characters, you would set the charset to **shift_jis**.

WebUI

1. **Editing Source Address**

   Security > Antivirus > Global: Enter or edit the source address that the AV scanner uses to send the notification mail, then click **Apply**.

2. **Editing Mail Subject**

   Security > Antivirus > Global: Enter or edit the mail subject that the AV scanner uses to send the notification mail, then click **Apply**. The default mail subject is Mail Delivery Failure.

3. **Specifying Charset**

   Security > Antivirus > Global: Enter the character set for the notification mail, then click **Apply**.

CLI

1. **Editing Source Address**

   ```
   set av notify-mail-subject
   unset av notify-mail-subject
   ```

2. **Editing Mail Subject**

   ```
   set av notify-mail-source
   unset av notify-mail-source
   ```

3. **Specifying Charset**

   ```
   set av notify-mail-charset string
   unset av notify-mail-charset
   ```

Maximum Content Size and Maximum Messages (Internal AV Only)

Scan Manager settings for maximum content size and maximum messages are supported on internal AV only. ICAP AV does not support the maximum content size and maximum messages settings.

On some devices, the internal AV scanner examines a maximum of 256 messages and 30 megabytes (MB) of decompressed file content at a time. The values for Maximum Content Size and Maximum Number of Messages depend on the device (see the *ScreenOS Release Notes*).

If the total number of messages or the size of the content received concurrently exceeds the device limits, by default the scanner drops the content without checking for viruses. For slow links, such as ISDN, decrease the max-content-size to a lesser value (set av scan-mgr max-content-size 20), so that AV scanning does not time out.

NOTE: On some security devices, the default for Maximum Content Size is 10 MB. However, if DI is enabled, we recommend that you configure a value of 6 MB.

For example, the scanner can receive and examine four 4-megabyte messages concurrently. If the scanner receives nine 2-megabyte messages concurrently, it drops the contents of the last two files without scanning it. You can change this default behavior so that the scanner *passes* the traffic instead of dropping it by doing the following:

WebUI

Security > Antivirus > Scan Manager

Content Oversize: Select **Permit** to pass traffic if the file size exceeds 30,000 KB

Or

Msgs Overflow: Select **Permit** if the number of files exceeds the maximum number of messages on the device, then click **Apply**.

CLI

```
unset av scan-mgr max-content-size drop
unset av scan-mgr max-msgs drop
```

When the AV scan engine is scanning a file and runs out of memory (typically, when decompressing files), the content is either dropped or passed based on the out of resource (set av scan-mgr out-of-resource) setting, instead of the fail-mode (set av all failmode) setting.

HTTP Keep-Alive

By default, the security device uses the HTTP " keep-alive" connection option, which does not send a TCP FIN to indicate the termination of data transmission. The HTTP server must indicate that it has sent all the data in another way, such as by sending the content length in the HTTP header or by some form of encoding. (The method that a server uses varies by server type.) This method keeps the TCP connection open while the antivirus examination occurs, which decreases latency and improves processor performance.

You can change the default behavior of the security device to use the HTTP "close" connection option for indicating the end of data transmission. (If necessary, the device changes the token in the connection header field from "keep-alive" to "close.") With this method, when the HTTP server completes its data transmission, it sends a TCP FIN to close the TCP connection and indicate that the server has finished sending data. When the device receives a TCP FIN, it has all the HTTP data from the server and can instruct the AV scanner to begin scanning it.

NOTE: The "keep-alive" not as secure as the " close" connection method. You can change the default behavior if you find that HTTP connections are timing out during the antivirus examination.

WebUI

Security > Antivirus > Global: Select **Keep Alive** to use the "keep-alive" connection option, or clear it to use the "close" connection option, then click **Apply**.

CLI

```
set av http keep-alive
unset av http keep-alive
```

HTTP Trickling (Internal AV Only)

HTTP trickling is the forwarding of specified amounts of unscanned HTTP traffic to the requesting HTTP client to prevent the browser window from timing out while the scan manager examines downloaded HTTP files. (The security device forwards small amounts of data in advance of transferring an entire scanned file.) By default, HTTP trickling is disabled. To enable it and use the default HTTP trickling parameters:

WebUI

Security > Antivirus > Global: Select the Trickling Default check box, then click **Apply**.

CLI

set av http trickling default

NOTE: HTTP trickling is supported on internal AV only. For YMSG however, trickling is disabled for chat and file transfer. ICAP AV does not support HTTP trickling.

With the default parameters, the security device employs trickling if the size of an HTTP file is 3MB or larger. The device forwards 500 bytes of content for every 1MB sent for scanning.

ScreenOS allows you to configure more granular trickling options if your browser times out during AV scanning. The browser times out if the security device requires more time to scan traffic or when the traffic is slow. Based on your environment, customize the values for time and data to trigger HTTP trickling as follows:

WebUI

Security > Antivirus > Global: Enter the following, then click **Apply**:

Trickling:
Custom: (select)
Minimum Length to Start Trickling:
***number1*.**
Trickle Size: ***number2*.**
Trickle for Every KB Sent for Scanning:***number3*.**
Trickle Timeout: ***number4*.**

CLI

set av http trickling threshold *number1* segment-size *number3* trickle-size *number2* timeout *number4*

The four *number* variables have the following meanings:

- *number1*: The minimum size (in kilobytes) of an HTTP file required to trigger trickling. If your browser times out because of a slow download, then reduce this value to trigger trickling sooner.
- *number2*: The size (a nonzero value) in bytes of unscanned traffic that the security device forwards.
- *number3*: The size (in kilobytes) of a block of traffic to which the security device applies trickling.
- *number4*: The time (in seconds) to trigger the trickling event. Time-based trickling begins when the initial size (*number1*) is reached. The value 0 indicates that time-based trickling is disabled.

NOTE: Data trickled to the client's browser appears as a small, unusable file. Because trickling works by forwarding a small amount of data to a client without scanning it, virus code might be among the data that the security device has trickled to the client. We advise users to delete such files.

You can disable HTTP trickling in the WebUI (Security > Antivirus: Click **Disable** in the Trickling section) or with the CLI command **unset av http trickling enable**. However, if a file being downloaded is larger than 8MB and HTTP trickling is disabled, the browser window will probably time out.

AV Profiles

Policies use AV profiles to determine which traffic undergoes AV examination and the actions to take as a result of this examination. ScreenOS supports the following types of profiles:

- Predefined AV Profiles

  ScreenOS supports two predefined profiles: the default, **ns-profile** (read only) and **scan-mgr profile** (read and write). Both profiles are supported for internal embedded AV only.

  The **scan-mgr profile** is automatically generated for backward compatibility, when you upgrade from ScreenOS 5.2 or earlier to ScreenOS 5.3 or later. The **scan-mgr profile** is generated to migrate the global Scan Manager commands.

  The Scan Manager is the AV component that manages the scan engine. For more information about the Scan Manager options, see "AV Scanner Global Settings" on page 521.

  The **scan-mgr profile** executes the following commands, so that the commands are now entered from within a profile context:

```
set ftp decompress-layer 2
set http decompress-layer 2
set imap decompress-layer 2
set pop3 decompress-layer 2
set smtp decompress-layer 2
set http skipmime enable
set http skipmime mime-list ns-skip-mime-list
```

For example, the **get av profile ns-profile** or **get av profile scan-mgr** command displays profile settings for the supported protocols:

```
device->get av profile ns-profile
ftp Setting:
status: enable
mode: scan-intelligent
decompress layer: 3
timeout: 180 seconds
include ext list: N/A
```

```
exclude ext list: N/A
http Setting:
status: enable
mode: scan-intelligent
decompress layer: 2
timeout: 180 seconds
include ext list: N/A
exclude ext list: N/A
skip scanning:text/html;text/css;audio/;video/;image/;
application/x-director
---
```

- Custom AV profiles

  Create your own AV profiles to customize the settings for each protocol. You can define a maximum of 8 AV profiles for each vsys (and root).

Assigning an AV Profile to a Firewall Policy

Only one AV profile can be linked to a firewall policy. Do the following to link the AV profile to a firewall policy.

WebUI

Policy > Policies: Click **Edit** on the policy to which you want to link the AV profile and select the profile under **Antivirus Profile**. Click **OK**.

CLI

device->**set policy id** *policy_num* **av ns-profile**

The following sections explain how to initiate an AV profile and configure the profile settings:

- "Initiating an AV Profile for Internal AV" on page 528
- "Example: (Internal AV) Scanning for All Traffic Types" on page 529
- "Example: AV Scanning for SMTP and HTTP Traffic Only" on page 529
- "AV Profile Settings" on page 530

Initiating an AV Profile for Internal AV

The following commands initiate a custom AV profile named *jnpr-profile,* which by default is configured to scan FTP, HTTP, IMAP, POP3, and SMTP traffic.

WebUI

Security > Antivirus > Profile: Select **New** and enter the profile name, *jnpr-profile,* then click **OK**.

CLI

```
set av profile jnpr-profile
device(av:jnpr-profile)->
```

```
device-> set av profile jnpr-profile
device(av:jnpr-profile)->
```

After you enter an AV profile context, all subsequent command executions modify the specified AV profile (*jnpr-profile*).

Example: (Internal AV) Scanning for All Traffic Types

In this example, you configure the AV scanner to examine FTP, HTTP, IMAP, POP3, IM, and SMTP traffic. Because you anticipate that the scanner will be processing a lot of traffic, you also increase the timeout from 180 seconds (the default setting) to 300 seconds.

WebUI

Security > Antivirus > Profile: Enter *profile_name*, then click **OK**.

By default, traffic for all six protocols is scanned.

NOTE: To change the timeout value, you must use the CLI.

CLI

```
set av profile jnpr-profile
(av:jnpr-profile)-> set http enable
(av:jnpr-profile)-> set http timeout 300
(av:jnpr-profile)-> set ftp enable
(av:jnpr-profile)-> set ftp timeout 300
(av:jnpr-profile)-> set imap enable
(av:jnpr-profile)-> set imap timeout 300
(av:jnpr-profile)-> set pop3 enable
(av:jnpr-profile)-> set pop3 timeout 300
(av:jnpr-profile)-> set smtp enable
(av:jnpr-profile)-> set smtp timeout 300
(av:jnpr-profile)-> exit
save
```

Example: AV Scanning for SMTP and HTTP Traffic Only

By default, the AV scanner examines FTP, HTTP, IMAP, POP3, and SMTP traffic. You can change the default behavior so that the scanner examines specific types of network traffic only.

You can also change the timeout value for each protocol. By default, an AV scan operation times out after 180 seconds if the security device does not start scanning after it receives all the data. The range is 1 to 1800 seconds.

In this example, you configure the AV scanner to examine all SMTP and HTTP traffic. You return the timeout value for both protocols to their defaults: 180 seconds.

NOTE: The internal AV scanner examines specific HTTP Webmail patterns only. The patterns for Yahoo!, Hotmail, and AOL mail services are predefined.

WebUI

Security > Antivirus > Select **New** and enter the profile name *jnpr-profile.*

Enter the following, then click **OK**.

Protocols to be scanned:
HTTP: (select)
SMTP: (select)
POP3: (clear)
FTP: (clear)
IMAP: (clear)

NOTE: To change the timeout value, you must use the CLI.

CLI

```
set av profile jnpr-profile
(av:jnpr-profile)-> set smtp timeout 180
(av:jnpr-profile)-> set http timeout 180
(av:jnpr-profile)-> unset pop3 enable
(av:jnpr-profile)-> unset ftp enable
(av:jnpr-profile)-> unset imap enable
(av:jnpr-profile)-> exit
save
```

AV Profile Settings

The following scanning options are configured for each application protocol:

DecompressingFile Attachments

When the security device receives content, the internal AV scanner decompresses any compressed files. You can configure the internal AV scanner to decompress up to four nested compressed files before executing a virus scan.

For example, if a message contains a compressed .zip file that contains another compressed .zip file, there are two compression layers, and decompressing both files requires a decompress-layer setting of 2.

WebUI

Security > Antivirus > Profile: Select **New** or **Edit** to edit an existing profile. Update the Decompress Layer to 2, then click **Apply**.

CLI

```
set av profile jnpr-profile
(av:jnpr-profile)-> set smtp decompress-layer 2
```

When transmitting data, some protocols use content encoding. The AV scan engine needs to decode this layer, which is considered as a decompression level before it scans for viruses.

AV Scanning Based on File Extensions

ScreenOS supports three modes of scanning:

- **scan-all**. The AV engine forwards all files to the scan engine for virus scanning.
- **scan-intelligent**. The AV engine uses a built-in algorithm to decide if files need to be scanned.

  This default scan-intelligent option specifies that the AV engine uses an algorithm that scans the traffic for the most common and prevalent viruses, including ensuring the file type is true and that it doesn't infect other files directly. Although this option is not as broad in coverage as scan-all, it provides better performance
- **scan-extension**. The AV engine forwards files for scanning based on extensions, exclusive and inclusive list (see the following section).

File-extension lists are used to determine which files undergo AV scanning for a specific protocol. You can select to *Include* a file-extension list and *Exclude* a file-extension list for each protocol.

A message is scanned when the file extension of a message is in the inclusion file-extension list. A message is not scanned if the file extension is in the exclusion file-extension list. If the file extension is not in either file-extension list, then the scanning decision depends on the default file-extension-scan setting. The default file extension is in the scan engine database, so it is read-only. Use the **get av scan-mgr** command to view the Scan engine default file extension list. There is no predefined file extension list for each protocol.

Configure the AV scanner to scan IMAP traffic by extensions and exclude files with the following extensions: .ace, .arj, and .chm.

WebUI

Security > Antivirus > Ext-list > New > Enter an extension-list name (elist1), and enter the list of extensions (ace;arj;chm). Click **OK**.

Security > Antivirus > Profile > Click **Edit** to select the profile > Select **IMAP** > Select the following options, then click **OK**:

Enable
Scan Mode: Scan by Extension
Exclude Extension List: elist1

CLI

```
set av extension-list elist1 ace;arj;chm
set av profile test1
(av:test1)-> set imap scan-mode scan-ext
(av:test1)-> set imap extension-list exclude elist1
```

AV Scanning Based on HTTP Content Type

You may use this option to determine which HTTP traffic must undergo AV scanning. The HTTP traffic is categorized into default predefined Multipurpose Internet Mail Extensions (MIME) types such as application/x-director, application/pdf, image, and so on.

You can configure the AV profile to skip MIME lists containing specific MIME types. The default predefined MIME list is ns-skip-mime-list. Yahoo Messenger file transfer ignores the MIME extensions specified in the MIME list because it uses the HTTP protocol. As part of the HTTP GET/PUT operation, the content-type header is specified as text or html for all files.

In this example, you configure the security device to scan all kinds of HTTP traffic regardless of MIME content type:

WebUI

Security > Antivirus > Profile > Click **Edit** to select the profile > Select HTTP and clear the Enable box under **Profile Name**. Click **OK**.

CLI

```
set av profile jnpr-profile
(av:jnpr-profile)-> unset av http skipmime
(av:jnpr-profile)-> exit
save
```

For more information about MIME types, see the *ScreenOS CLI Reference Guide: IPv4 Command Descriptions*.

Virus Detection Notification Method

When ScreenOS detects a virus, it provides the following methods for notifying the client that initiated the traffic:

- Content—Replaces the data with a warning message about the virus and sends it to the client
- Protocol—Drops the data and sends a warning message about the virus to the client

By default, a security device notifies the client about the detected virus using the content method. You can configure the AV profile to use the protocol method with the virus-detection-notify-method option. This option applies only to the FTP, HTTP, IMAP, and POP3 protocols.

When a virus is detected, the AV scanner can either replace the infected packet with a warning message or drop the packet. In both cases, a warning message is sent to the client that initiated the traffic.

The AV scanner uses the default warning messages and their respective protocol codes to notify the client when a virus is detected.

NOTE: You can customize the warning messages. For more information, see "AV Warning Message" on page 522.

In this example, you configure the security device to use the protocol method:

WebUI

Security > Antivirus > Profile > Click **Edit** to select the profile > Select **FTP**, then select **Virus Detection Notify with Protocol Code**, and click **OK**.

CLI

```
set av profile jnpr-profile
(av:jnpr-profile)-> set ftp virus-detection-notify-method protocol
save
```

Notifying Sender and Recipient via Email

The email-notification option applies only to the IMAP, POP3, and SMTP protocols. You can configure the AV profile to notify senders or recipients scanning errors or virus information.

When a virus is found in an email message, the content of the warning message (virus name, source/destination IP) is included in a notification-level message. The warning-level message is sent via an email through the SMTP protocol.

When a scanning error occurs in a message, the content of the scanning error message should be included in a warning-level message. This message is sent via an email through the SMTP protocol.

In this example, you configure the security device to do the following:

- Notify the sender when a virus is detected
- Notify the sender and recipients if scanning errors occur

WebUI

Security > Antivirus > Profile > Select the Profile to **Edit** > Select IMAP, then click **OK**.

Enter the following, then click **OK:**
Protocols to be scanned:
Email Notify > Select Virus Sender
Email Notify > Select Scan-error Sender
Email Notify > Select Scan-error Recipient

CLI

```
set av profile jnpr-profile
(av:jnpr-profile)-> set imap email-notify virus sender
(av:jnpr-profile)-> set imap email-notify scan-error sender
(av:jnpr-profile)-> set imap email-notify scan-error recipient
(av:jnpr-profile)-> exit
save
```

Example: Dropping Large Files

In this example, you configure the AV scanner to decompress HTTP traffic of up to three files layered within one another. You also configure the scanner to drop content either if the total number of messages received concurrently exceeds four messages or if the total decompressed size of the content exceeds the configured value. The total decompressed file content size that ScreenOS can handle is device-specific with a minimum of 10 MB.

NOTE: The default value for decompressed file content size is per message and not the total number of concurrent messages being examined.

The default values for Maximum Concurrent Messages and Maximum Queue size indicate that the AV scanner can examine a total of 16 concurrent messages at any specific time. The 17th message is dropped or passed as configured.

WebUI

Security > Antivirus > Scan Manager: Enter the following, then click **OK**:

Content Oversize: Drop Max Content Size: 3000 KB (20~10000)
Msg Overflow: Drop Max Concurrent messages is 256

Security > Antivirus > Profile: Select Edit > HTTP: Enter the following, then click **OK**:

File decompression: 3 layers (1~4)

CLI

```
set av scan-mgr max-msgs drop
set av scan-mgr max-content-size 3000
set av scan-mgr max-content-size drop
set av profile jnpr-profile
(av:jnpr-profile)-> set http decompress-layer 3
(av:jnpr-profile)-> exit
save
```

Antispam Filtering

Spam consists of unwanted email messages, usually sent by commercial, malicious, or fraudulent entities. The antispam feature examines transmitted messages to identify spam. When the device detects a message deemed to be spam, it either drops the message or tags the message field with a preprogrammed string.

This antispam feature is not meant to replace your antispam server, but to complement it. Configuring this command prevents an internal corporate email server from receiving and distributing spams. Corporate users retrieve emails from an internal email server without going through the firewall. This should be a typical configuration in an enterprise environment.

Juniper Networks antispam uses a constantly updated, IP-based, spam-blocking service that uses information gathered worldwide. Because this service is robust and yields very few false positives, you are not required to tune or configure blacklists. However, you have the option of adding specific domains and IPs to local whitelists or blacklists, which the device can enforce locally.

NOTE: This release supports antispam for the SMTP protocol only.

To prevent or reduce the volume of spam messages you receive, you can configure an antispam profile. You can use the profile in policies to detect and filter out suspected spam messages. An antispam profile allows you to designate lists of IP addresses, emails, hostnames, or domain names identified as malicious (spam) or benign (non-spam). The antispam profile can include lists of the following types:

- Public-based blacklists or whitelists

  If the connection is from a mail-forwarding agent, the device can filter the connection's source IP address using lists of devices deemed to be benign (whitelist) or malicious (blacklist).

- Custom defined blacklists or whitelists
- Domain-name-based blacklists or whitelists

  The device can use such lists to filter connections that use domain names deemed to be benign or malicious.

- Address-book-based blacklists or whitelists

  The device can use such lists to base filtering on the sender's email address or domain. By default, any email server should accept its own user's email.

Blacklists and Whitelists

The antispam feature requires that the firewall have Internet connectivity with the Spam Block List (SBL) server. Domain Name Service (DNS) must be available to access the SBL server. The firewall performs reverse DNS lookups on the source of the SMTP sender (or relaying agent), adding the name of the SBL server (such as sbl-server) as the authoritative domain. The DNS server then forwards each request to the SBL server, which returns a value to the firewall.

You can configure both local whitelists and blacklists. By default, the device checks first against the local database of whitelists and then the blacklists. If it does not find the hostname, the device proceeds to query the Spamhaus Block List (SBL) server located on the Internet.

Basic Configuration

The following commands provide an example of basic antispam configuration, where you are protecting an SMTP server (or relay server) from receiving spam e-mails.

```
set anti-spam profile ns-profile
set policy from untrust to trust any mail-server SMTP permit log anti-spam
ns-profile
```

In the following example, the device tests spammer.org to see if it resides on either the whitelist or the blacklist.

```
exec anti-spam testscan spammer.org
```

If the blacklist contains spammer.org, the device might produce the following output:

```
AS: anti spam result: action Tag email subject, reason: Match local blacklist
```

Alternatively, if the whitelist contains spammer.org, the device might produce the following output:

```
AS: anti spam result: action Pass, reason: Match local whitelist
```

For information about creating blacklists or whitelists, see "Defining a Blacklist" on page 537 and "Defining a Whitelist" on page 538.

NOTE: The whitelist takes precedence over the blacklist, so an IP address match in both will mean that the IP address is on the whitelist.

Filtering Spam Traffic

In the following examples, SMTP traffic that includes spam traverses the security device. However, ScreenOS checks for spam by either DNS name or IP address.

The following commands provide an example of filtering spam traffic:

```
device-> exec anti-spam test 2.2.2.2
AS: anti spam result: action Tag email subject, reason: Match local black list
exec anti-spam testscan spammer.org
AS: anti spam result: action Tag email subject, reason: Match local black list
```

Dropping Spam Messages Dropping Spam Messages

Executing the **set anti-spam profile ns-profile** command without specifying further options places the CLI within the context of a new or an existing antispam profile. For example, the following commands define a profile named **ns-profile** and then enter the **ns-profile** context to instruct the device to drop suspected spam messages:

```
device-> set anti-spam profile ns-profile
device(ns-profile)-> set default action drop
```

After you enter an antispam context, all subsequent command executions modify the specified antispam profile (**ns-profile** in this example). To save your changes, you must first exit the antispam context and then enter the **save** command:

```
device(ns-profile)-> exit
device-> save
```

Defining a Blacklist

Use the blacklist commands to add or remove an IP address with or without a netmask, an e-mail address, a hostname, or a domain name from the local antispam blacklist. Each entry in a blacklist can identify a possible spammer.

NOTE: All IP addresses with netmask are stored in a list and sorted by the number of zeros in each netmask. Netmasks with more zeros are matched first.

To define a blacklist:

1. Initiate a profile context (**ns-profile**).
2. Give the profile a blacklist entry that prevents connections with the hostname **www.wibwaller.com**.
3. Exit the spam context and apply the profile to an existing policy (**id 2**).

```
device-> set anti-spam profile ns-profile
device(anti-spam:ns-profile)-> set blacklist www.wibwaller.com
device(anti-spam:ns-profile)-> exit
device-> set policy id 2 anti-spam ns-profile
```

Defining a Whitelist

Use the whitelist commands to add or remove an IP or e-mail address, a hostname, or a domain name from the local whitelist. Each entry in a whitelist can identify an entity that is not a suspected spammer. The following table shows some possible entries.

To define a whitelist:

1. Initiate a profile context (**ns-profile**).
2. Give the profile a whitelist entry that allows connections with the hostname **www.fiddwicket.com**.
3. Exit the spam context and apply the profile to an existing policy (**id 2**).

```
device-> set anti-spam profile ns-profile
device(anti-spam:ns-profile)-> set whitelist www.fiddwicket.com
device(anti-spam:ns-profile)-> exit
device-> set policy id 2 anti-spam ns-profile
```

Defining a Default Action

Use the default commands to specify how the device handles messages deemed to be spam. The device can either drop a spam message or identify it as spam by tagging it.

You can place the tag either in the message header or the subject line.

To define the default action for spam, perform the following tasks:

1. Initiate a profile context (**ns-profile**).
2. Specify that email messages deemed to be spam will have the string “This is spam” added to the message header.
3. Exit the spam context and apply the profile to an existing policy (**id 2**).

```
device-> set anti-spam profile ns-profile
device(anti-spam:ns-profile)-> set default action tag header “This is spam”
device(anti-spam:ns-profile)-> exit
 device-> set policy id 2 anti-spam ns-profile
```

Enabling a Spam-Blocking List Server

Use the **sbl** command to enable use of the external spam-blocking SBL service, which uses a blacklist to identify known spam sources. The service replies to queries from the device about whether an IP address belongs to a known spammer.

Example: These commands perform the following tasks:

1. Initiate a profile context (**ns-profile**).
2. Enable use of the default antispam service.
3. Exit the spam context and apply the profile to an existing policy (**id 2**).

```
device-> set anti-spam profile ns-profile
device(anti-spam:ns-profile)-> set sbl default-server-enable
device(anti-spam:ns-profile)-> exit
device-> set policy id 2 anti-spam ns-profile
```

Testing Antispam

Use the command, **exec anti-spam testscan** < IP addr > to cause the security device to scan for known spammer IP addresses.

Example: These commands tests the IP address 12.13.2.3 for spam:

```
device-> set console dbuf
device-> exec anti-spam testscan 12.13.2.3
device-> get dbuf stream
 anti spam result: action Pass, reason: No match
```

Web Filtering

Web filtering enables you to manage Internet access by preventing access to inappropriate Web content. ScreenOS provides two Web-filtering solutions:

- Integrated

  Select security devices support an integrated Web-filtering solution that employs Content Portal Authority (CPA) servers from WebSense.

NOTE: In order to access all integrated Web-filtering features, you must install a license key.

The following table specifies the Web filtering features that the security device still supports if the license key is not installed or has expired.

Feature	License Key Not Installed	License Key Expired
Define Web-filtering profiles and bind them to policies	Yes	Yes
Retrieve category information for HTTP requests	Yes	Yes
Check cache for categories	No	Yes
Define static whitelist or blacklist category and bind them to policies.	Yes	Yes

Integrated Web filtering allows you to permit or block access to a requested site by binding a Web-filtering profile to a firewall policy. A Web-filtering profile specifies URL categories and the action the security device takes (permit or block) when it receives a request to access a URL in each category. URL categories are predefined or are user-defined. For information about configuring the integrated Web-filtering feature, see "Integrated Web Filtering" on page 541.

- Redirect

  Select security devices support a Web-filtering solution that employs SurfControl and Websense services to a SurfControl or Websense server.

  In redirect Web filtering, the security device sends the HTTP request in a TCP connection to either a Websense server or a SurfControl server, enabling you to block or permit access to different sites based on their URLs, domain names, and IP addresses. For information about configuring the redirect Web-filtering feature, see "Redirect Web Filtering" on page 551.

Using the CLI to Initiate Web-Filtering Modes

You can use the WebUI or CLI to configure your security device for Web filtering. If you are using the CLI, perform the following steps to configure either of the Web-filtering solutions:

1. Select the protocol.

   For example, the **set url protocol type { sc-cpa | scfp | websense }** command selects the protocol.

2. Initiate the Web-filtering mode.

   Executing the **set url protocol { sc-cpa | scfp | websense }** command places the CLI in the Web-filtering context. Once you initiate the Web-filtering context, all subsequent command executions apply to that Web-filtering mode.

Table 54 on page 540 shows the commands for entering and exiting the three different Web-filtering modes.

Table 54: Entering and Exiting Web-Filtering Modes

		Integrated Web Filtering	Redirecting to SurfControl Server	Redirecting to Websense Server
1.	Select the protocol	**set url protocol type sc-cpa**	**set url protocol type scfp**	**set url protocol type websense**
1.	Initiate the Web-filtering context	**set url protocol sc-cpa** **(url:sc-cpa)-> :**	**set url protocol scfp** **(url:scfp)-> :**	**set url protocol websense** **(url:websense)-> :**
1.	Exit the Web-filtering mode	**(url:sc-cpa)-> :exit**	**(url:scfp)-> :exit**	**(url:websense)-> :exit**

Integrated Web Filtering

To enable Web filtering, you first bind a Web-filtering profile to a firewall policy. With integrated Web filtering, the Juniper Networks security device intercepts each HTTP request, determines whether to permit or block access to a requested site by categorizing its URL, then matches the URL category to a Web-filtering profile. A Web-filtering profile defines the action the security device takes (permit or block) when it receives a request to access a URL.

A URL category is a list of URLs organized by content. Security devices use the SurfControl predefined URL categories to determine the category of the requested URL. SurfControl Content Portal Authority (CPA) servers maintain the largest database of all types of Web content classified into about 40 categories. A partial list of the URL categories is shown in "Define URL Categories (Optional)" on page 543.

For a complete list of SurfControl URL categories, visit the Websense website at http://www.websense.com/global/en/scwelcome.

In addition to the SurfControl predefined URL categories, you can also group URLs and create categories based on your needs. For information about creating user-defined categories, see "Define URL Categories (Optional)" on page 543.

Following is the basic sequence of events when a host in the Trust zone tries an HTTP connection to a server in the Untrust zone:

1. The security device checks for a firewall policy that applies to the traffic:
   - If there is no firewall policy for the traffic, the device drops the traffic.
   - If there is a firewall policy and if Web filtering is enabled on that policy, the device intercepts all HTTP requests.
2. The device checks for a user-defined profile bound to the firewall policy. If there is none, the device then uses the default profile, **ns-profile**.
3. The device determines if the category of the requested URL is already cached. If it is not, the device sends the URL to the SurfControl CPA server for categorization and caches the result.
4. Once the device determines the category of the URL, it checks for the category in the Web-filtering profile bound to the firewall policy.
   - If the category is in the profile, the device blocks or permits access to the URL as defined in the profile.
   - If the category is not in the profile, the device performs the configured default action.

This section addresses the following integrated Web-filtering topics:

- "SurfControl Servers" on page 542
- "Redirect Web Filtering" on page 551
- "Web-Filtering Cache" on page 542

- "Configuring Integrated Web Filtering" on page 543
- "Example: Integrated Web Filtering" on page 549

SurfControl Servers

SurfControl has three server locations, each of which serves a specific geographic area: the Americas, Asia Pacific, and Europe/MiddleEast/Africa. The default primary server is the Americas, and the default backup server is Asia Pacific. You can change the primary server, and the security device automatically selects a backup server, based on the primary server. (The Asia Pacific server is the backup for the Americas server, and the Americas server is the backup for the other two servers.)

The SurfControl CPA server periodically updates its list of categories. Since the CPA server does not notify its clients when the list is updated, the security device must periodically poll the CPA server. By default, the device queries the CPA server for category updates every two weeks. You can change this default to support your networking environment. You can also manually update the category list by entering the Web-filtering context and executing the **exec cate-list-update** command. To manually update the category list, do the following:

```
device-> set url protocol sc-cpa
device(url:sc-cpa)-> exec cate-list-update
```

Web-Filtering Cache

By default, the security device caches the URL categories. This action reduces the overhead of accessing the SurfControl CPA server each time the device receives a new request for previously requested URLs. You can configure the size and duration of the cache, according to the performance and memory requirements of your networking environment. The default cache size is platform-dependent, and the default timeout is 24 hours.

In the following example, you change the cache size to 500 kilobytes (KB) and the timeout value to 18 hours.

WebUI

Security > Web Filtering > Protocol Selection > Select **Integrated (SurfControl)**, then click **Apply**.

```
Enable Cache: (select)
Cache Size: 500 (K)
Cache Timeout: 18 (Hours)
```

CLI

```
device-> set url protocol sc-cpa
device(url:sc-cpa)-> set cache size 500
device(url:sc-cpa)-> set cache timeout 18
device(url:sc-cpa)-> exit
device-> save
```

Configuring Integrated Web Filtering

To configure a security device for Web filtering, perform the following steps:

1. "Set Up a Domain Name Server" on page 543
2. "Enable Web Filtering" on page 543
3. "Define URL Categories (Optional)" on page 543
4. "Define Web-Filtering Profiles (Optional)" on page 545
5. Prioritize User Groups on page 547
6. "Enable Web-Filtering Profile and Policy" on page 548

Each step is described in detail in the following sections.

Set Up a Domain Name Server

The Juniper Networks security device incorporates Domain Name System (DNS) support, allowing you to use domain names as well as IP addresses for identifying locations. You must configure at least one DNS server to enable the security device to resolve the CPA server name to an address. For more information about DNS, see"Domain Name System Support" on page 263.

Enable Web Filtering

You can use the Web UI or CLI commands to enable integrated Web filtering on a security device. If you use the CLI, you must enter the Web-filtering context before entering the commands specific to integrated Web filtering.

WebUI

Security > Web Filtering > Protocol Selection: Select **Integrated (SurfControl)**, then click **Apply**. Then select **Enable Web Filtering via CPA Server**, and click **Apply** again.

CLI

```
device-> set url protocol type sc-cpa
device-> set url protocol sc-cpa
device(url:sc-cpa)-> set enable
device(url:sc-cpa)-> exit
device-> save
```

The *device* **(url:sc-cpa)->** prompt indicates that you have entered the integrated Web-filtering context and can now configure integrated Web-filtering parameters.

Define URL Categories (Optional)

A category is a list of URLs grouped by content. There are two types of categories: predefined and user-defined. SurfControl maintains about 40 predefined categories. A partial list of the URL categories is shown in Table 55 on page 544. For a complete

list and description of each URL category developed by SurfControl, visit the Websense website at http://www.websense.com/global/en/scwelcome.

To view the list of SurfControl predefined URL categories:

WebUI

Security > Web Filtering > Profiles > Predefined category

CLI

```
device-> set url protocol type sc-cpa
device-> set url protocol sc-cpa
device(url:sc-cpa)-> get category pre
```

The URL category list displayed is similar to that shown in Table 55 on page 544.

Table 55: Partial List of SurfControl URL Categories

Type	Code	Category Name
Predefine	76	Advertisements
Predefine	50	Arts & Entertainment
Predefine	3001	Chat
Predefine	75	Computing & Internet

The predefined categories list displays the categories and their SurfControl internal codes. Although you cannot list the URLs within a category, you can determine the category of a website by using the Test A Site feature on the Websense website at http://www.websense.com/global/en/scwelcome.

In addition to the SurfControl predefined URL categories, you can group URLs and create categories specific to your needs. Each category can have a maximum of 20 URLs. When you create a category, you can add either the URL or the IP address of a site. When you add a URL to a user-defined category, the device performs DNS lookup, resolves the hostname into IP addresses, and caches this information. When a user tries to access a site with the IP address of the site, the device checks the cached list of IP addresses and tries to resolve the hostname.

Many sites have dynamic IP addresses, meaning that their IP addresses change periodically. A user attempting to access a site can type an IP address that is not in the cached list on the device. Therefore, if you know the IP addresses of sites you are adding to a category, enter both the URL and the IP address(es) of the site.

NOTE: If a URL appears in both a user-defined category and a predefined category, the device matches the URL to the user-defined category.

In the following example, you create a category named **Competitors** and add the following URLs: **www.games1.com** and **www.games2.com**.

WebUI

Security > Web Filtering > Profiles > Custom > New: Enter the following, then click **Apply**:

```
Category Name: Competitors
URL: www.games1.com
```

Enter the following, then click **OK**:

```
URL: www.games2.com
```

CLI

```
device-> set url protocol sc-cpa
device(url:sc-cpa)-> set category competitors url www.games1.com
device(url:sc-cpa)-> set category competitors url www.games2.com
device(url:sc-cpa)-> exit
device-> save
```

Define Web-Filtering Profiles (Optional)

A Web-filtering profile consists of a group of URL categories assigned with one of the following actions:

- Permit - The security device always allows access to the websites in this category.
- Block - The security device blocks access to the websites in this category. When the device blocks access to this category of websites, it displays a message in your browser indicating the URL category.

  You can edit an existing message or create a new message (up to 500 characters) to be sent from the security device. To create or edit a deny message:

WebUI

Security > Web Filtering > Protocol > Selection: Select **Integrated (SurfControl)**.

Enter the message in the **Web Filter Deny Message** text area, then click **Apply**.

The device displays the following default message:

Your page is blocked due to a security policy that prohibits access to $URL-CATEGORY

CLI

```
device-> set deny-message deny-message-str
```

- Blacklist—The security device always blocks access to the websites in this list. You can create a user-defined category or use a predefined category.
- Whitelist—The security device always allows access to the websites in this list. You can create a user-defined category or use a predefined category.

Juniper Networks security devices provide a default profile called **ns-profile**. This profile lists the SurfControl predefined URL categories and their actions. You cannot edit the default profile. To view the predefined profile, use the following command:

WebUI

Security > Web Filtering > Profiles > Predefined

CLI

```
device-> set url protocol sc-cpa
device(url:sc-cpa)-> get profile ns-profile
```

The security device displays the predefined profile as illustrated below:

```
Profile Name: ns-profile
Black-List category: None
White-List category: None
Default Action: Permit

Category                Action
Advertisements          block
Arts & Entertainment    permit
Chat    permit
Computing & Internet    permit
-
-
-
Violence                block
Weapons                 block
Web-based Email         permit
other                   permit
```

If the URL in an HTTP request is not in any of the categories listed in the default profile, the default action of the security device is to permit access to the site.

You can create a custom profile by cloning an existing profile, saving it with a new name, and then editing the profile. Perform the following step in the WebUI to clone **ns-profile**.

WebUI

Security > Web Filtering > Profiles > Custom: ns-profile: Select **Clone**.

NOTE: You must use the webUI to clone **ns-profile**.

You can also create your own Web-filtering profile. When you create a web-filtering profile, you can:

- Add both user-defined and SurfControl predefined URL categories
- Specify a category for the blacklist and/or the whitelist
- Change the default action

In the following example, you create a custom profile called **my-profile** with a default action of **permit**. Then, you take the category you created in the previous example and add it to **my-profile** with an action of **block**.

WebUI

Security > Web Filtering > Profiles > Custom > New: Enter the following, then click **Apply**:

Profile Name: my-profile
Default Action: Permit
Select the following, then click **OK**:
Subscribers Identified by:
Category Name: Competitors (select)
Action: Block (select)
Configure: Add (select)

NOTE: To configure the default action using the CLI, specify the action for the Other category.

CLI

```
device-> set url protocol type sc-cpa
device-> set url protocol sc-cpa
device(url:sc-cpa)-> set profile my-profile other permit
device(url:sc-cpa)-> set profile my-profile competitors block
device(url:sc-cpa)-> exit
device-> save
```

Prioritize User Groups

User groups are created to manage authentication users collectively. See Referencing Auth User Groups in Policies on page 1526. A user can belong to more than one user group. Therefore, user groups must be prioritized so that the UF Manager selects the

user group with the highest priority and blocks or permits the HTTP/HTTPS request according to the profile bound to that user-group. The priority can be set to a value between 1 and 65535. No two user groups can have the same priority. You can create user groups and prioritize them using the WebUI or the CLI

NOTE: If profile and priority are not configured for the user group, the UF Manager accepts the profile bound to the policy.

In the following example, you set the priority to 4.

WebUI

Security > Web Filtering > User Group Binding: Enter the following, then click **Apply**:

User Group: (select)
Priority: 4
Profile: (select)

CLI

device-> set url protocol sc-cpa
device(url:sc-cpa)-> set user-group *group_name* priority *priority_number*
device(url:sc-cpa)-> set user-group *group_name* profile *profile_name*

Enable Web-Filtering Profile and Policy

Firewall policies permit or deny specified types of unidirectional traffic between two points. (For information about firewall policies, see Policies.)You can enable both antivirus (AV) scanning and integrated Web filtering in a policy. (For information about AV scanning, see “Antivirus Scanning” on page 499.)

If user–group–based filtering is enabled, the UF Manager extracts the URL from the HTTP/HTTPS request and identifies the profile assigned to the user. For example, the UF Manager identifies the username and the user group associated with the IP address. If the user belongs to multiple user groups, the UF Manager selects the user group that has highest priority. Then the UF Manager identifies the category of the URL and permits or blocks the HTTP/HTTPS request according to the group-based profile.

NOTE: By default, user group–based–filtering is disabled.

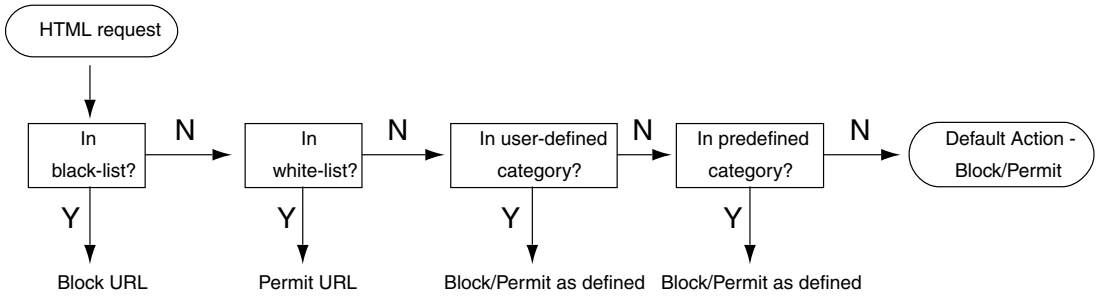
If the user group or profile related to the user is not found, the security device intercepts all HTTP requests. If there is a Web-filtering profile bound to the policy, the device matches the URL in the incoming HTTP request to the categories in the profile in the following sequence:

1. Blacklist
2. Whitelist

3. User-defined categories
4. SurfControl predefined URL categories

If the device is unable to determine the category of the requested URL, it blocks or permits access based on the default configuration in the profile.

Figure 137: Web-Filtering Profiles and Policies Flowchart



If the device determines that the URL is in a permitted category, and if AV scanning is enabled for that policy, the device scans the contents for viruses. If the device determines that the URL is in a blocked category, it closes the TCP connection, sends a message alerting the user, and does not perform AV scanning.

Example: Integrated Web Filtering

In this example, you perform the following steps to enable integrated Web filtering on the security device and block access to competitor sites.

1. Create a category called **Competitors**.
2. Add the following URLS to the category: **www.comp1.com** and **www.comp2.com**.
3. Create a profile called **my-profile**, and add the **Competitors** category.
4. Apply **my-profile** to a firewall policy.

To determine whether the group–based filtering is enabled, use the **get url** CLI command. The following output appears:

```
device-> get url
UF Key Expire Date: 11/21/2009 00:00:00
SC-CPA Web filtering:  enabled
Primary:   america    usi.SurfCPA.com       port:9020
Secondary: asia       Asiai.SurfCPA.com     port:9020
Cache: enabled
Cache Size: 500(K)
Cache Timeout: 24 Hour(s)
Cache Host Count: 0
Cache URL  Count: 0
Category list query interval: 2 Week(s)
Fail Mode: Fail-Block
```

Log: blocked
Group-based filtering: disabled

WebUI

1. **Web Filtering**

   Security > Web Filtering > Protocol > Select **Integrated (SurfControl)** then click **Apply**. Then select **Enable Web Filtering via CPA Server** and select **User Group Based Filtering**. Click **User Group Binding** to set the priority, and click **Apply** again.

2. **URL Category**

   Security > Web Filtering > Profile > Custom List > New: Enter the following, then click **Apply**:

   Category Name: Competitors
   URL: www.comp1.com

   Enter the following, then click **OK**:

   URL: www.comp2.com

3. **Web-Filtering Profile**

   Security > Web Filtering > Profile > Custom Profile > New: Enter the following, then click **Apply**:

   Profile Name: my-profile
   Default Action: Permit
   Category Name: Competitors (select)
   Action: Block (select)
   Configure: Add (select)

4. **Policy**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: HTTP
   Web Filtering: (select), my-profile
   Action: Permit

CLI

1. **Web Filtering**

```
get url
device->set url protocol sc-cpa
device(url:sc-cpa)-> set enable
```

```
device(url:sc-cpa)-> set group-based-filtering
device(url:sc-cpa)-> get user-group group_name | all
```

2. **URL Category**

```
device(url:sc-cpa)-> set category competitors url www.comp1.com
device(url:sc-cpa)-> set category competitors url www.comp2.com
```

3. **Web-Filtering Profile**

```
device(url:sc-cpa)-> set profile my-profile other permit
device(url:sc-cpa)-> set profile my-profile competitors block
device(url:sc-cpa)-> exit
```

4. **Firewall Policy**

```
device-> set policy id 23 from trust to untrust any any http permit url-filter
device-> set policy id 23
device(policy:23)-> set url protocol sc-cpa profile my-profile
device(policy:23)-> exit
device-> save
```

Redirect Web Filtering

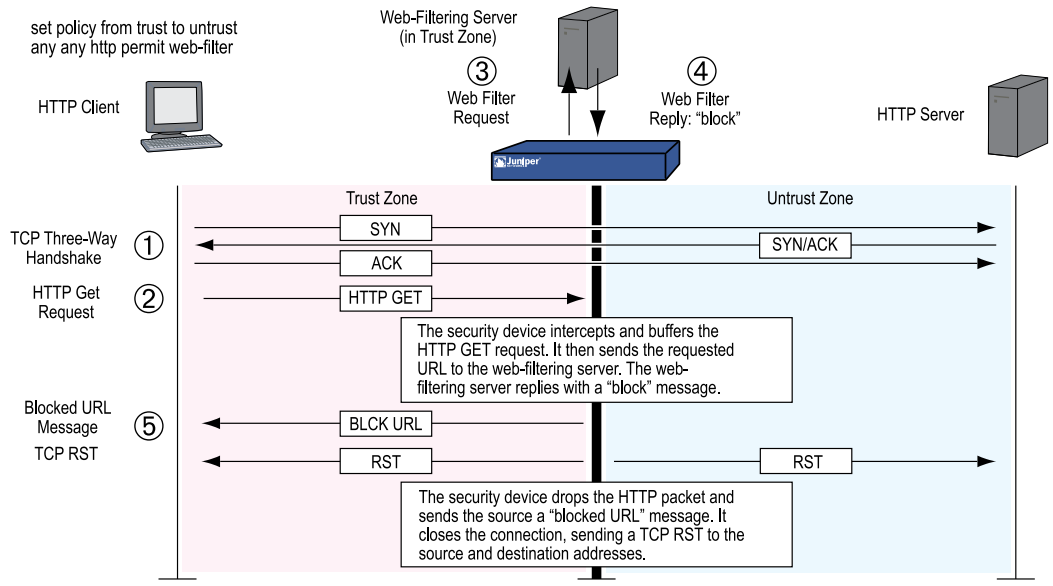
Juniper Networks security devices support redirect Web filtering using either the Websense Enterprise Engine or the SurfControl Web Filter, both of which enable you to block or permit access to different sites based on their URLs, domain names, and IP addresses. The security device can link directly to a Websense or SurfControl Web-filtering server.

NOTE: For additional information about Websense, visit http://www.websense.com/global/en. For additional information about SurfControl, visit http://www.websense.com/global/en/scwelcome.

Figure 138 on page 552 shows the basic sequence of events when a host in the Trust zone attempts an HTTP/HTTPS connection to a server in the Untrust zone. However, Web filtering determines that the requested URL is prohibited.

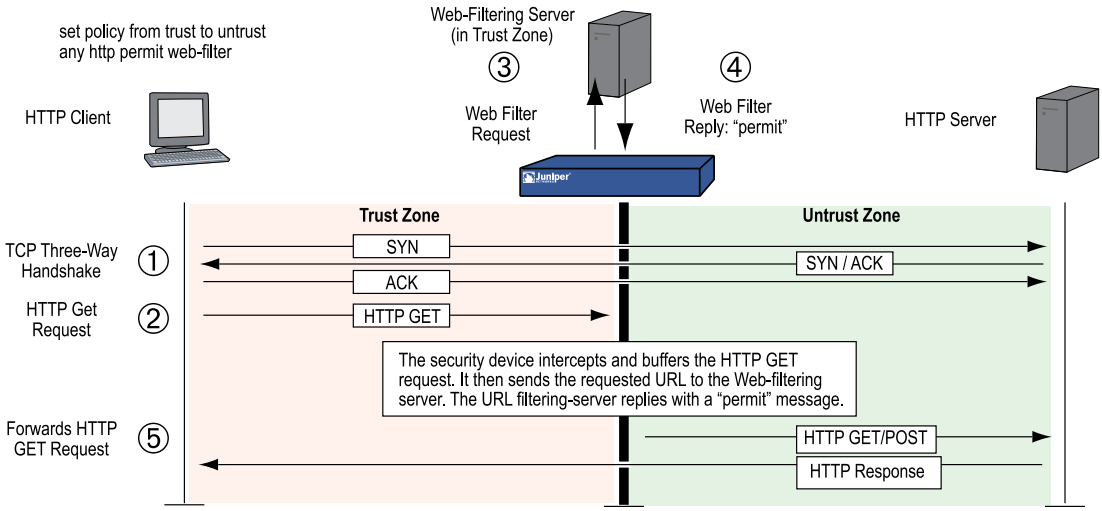
NOTE: The examples in Figure 138 on page 552 and Figure 139 on page 552 use HTTP; however, HTTPS is also supported.

Figure 138: A Blocked URL from Trust Zone to Untrust Zone



If the server permits access to the URL, the sequence of events in the HTTP connection attempt proceeds as shown in Figure 139 on page 552.

Figure 139: A Permitted URL from Trust Zone to Untrust Zone



See the following sections for more details on redirect Web filtering:

- "Virtual System Support" on page 553
- "Configuring Redirect Web Filtering" on page 553
- "Example: Redirect Web Filtering" on page 556

Virtual System Support

Security devices with virtual systems (vsys) support up to eight Web-filtering servers—one server reserved for the root system, which can be shared with an unrestricted number of virtual systems, and seven Web-filtering servers for private use by the virtual systems. A root-level administrator can configure the Web-filtering module at the root and vsys levels. A vsys-level administrator can configure the URL module for his or her own vsys if that vsys has its own dedicated Web-filtering server. If the vsys-level administrator uses the root Web-filtering server settings, that administrator can view—but not edit—the root-level Web-filtering settings.

Alternatively, devices with virtual systems that use Websense Web-filtering servers can share all eight Websense servers, not just the root server. Each Websense server can support an unrestricted number of virtual systems, allowing you to balance the traffic load among the eight servers.

To configure multiple virtual systems to connect to a Websense Web-filtering server, the root-level or vsys administrator must perform the following steps:

1. Create an account name for each vsys. Use the following CLI command:

   ```
   device-> set url protocol type websense
   device-> set url protocol websense
   device(url:websense)-> set account name
   ```

   When a host in a vsys sends out a URL request, it includes the account name. This name enables the Websense server to identify which vsys sent the URL request.

2. Configure the same Web-filtering server settings and system-level parameters for each vsys that shares a Websense Web-filtering server. The next section contains information about configuring Web-filtering settings and parameters.

Configuring Redirect Web Filtering

To configure a security device to perform redirected Web filtering, follow these steps:

1.Set Up a Domain Name System Server

The Juniper Networks security device incorporates Domain Name System (DNS) support, allowing you to use domain names as well as IP addresses for identifying locations. You must configure at least one DNS server to enable the security device to resolve the CPA server name to an address. For more information about DNS, see “Domain Name System Support” on page 263.

2.Set Up Communication with the Web-Filtering Servers

Configure the security device to communicate with one of the following servers:

- Websense server
- SurfControl server using the SurfControl Content Filtering Protocol (SCFP)

You can set up communications with up to eight Web-filtering servers.

WebUI

Security > Web Filtering > Protocol > Select **Redirect (Websense) or Redirect (SurfControl),** then click **Apply**.

CLI

Enter the Web-filtering context for SurfContol (scfp) or Websense (websense) redirect filtering. For more information, see "Using the CLI to Initiate Web-Filtering Modes" on page 540.

```
device-> set url protocol type { websense | scfp }
device-> set url protocol { websense | scfp }
device(url:scfp)-> set server { ip_addr | dom_name } port_num timeout_num
```

Configure the following Web-filtering settings at the system level for Web-filtering server communication:

- **Source Interface:** The source from which the device initiates Web-filter requests to a Web-filtering server.
- **Server Name:** The IP address or Fully Qualified Domain Name (FQDN) of the computer running the Websense or SurfControl server.
- **Server Port:** If you have changed the default port on the server, you must also change it on the security device. (The default port for Websense is 15868, and the default port for SurfControl is 62252.) Please see your Websense or SurfControl documentation for full details.
- **Communication Timeout:** The time interval, in seconds, that the device waits for a response from the Web-filtering server. If the server does not respond within the time interval, the device either blocks or allows the request. For the time interval, enter a value from 10 through 240.

If a device with multiple virtual systems connects to a Websense server, the virtual systems can share the server. To configure multiple virtual systems to share a Websense server, use the following CLI commands to create an account name for each vsys:

```
device-> set url protocol type websense
device-> set url protocol websense
device(url:websense)-> set account name
```

Once you have configured the vsys names, you define the settings for the Web-filtering server and the parameters for the behavior that you want the security device to take when applying Web filtering. If you configure these settings in the root system, they also apply to any vsys that shares the Web-filtering configuration with the root system. For a vsys, the root and vsys administrators must configure the settings separately. Virtual systems that share the same Websense Web-filtering server must have the same Web-filtering settings.

3.Enable Web Filtering at the Root and Vsys Levels

You must enable Web filtering at the system level. For a device that is hosting virtual systems, enable Web filtering for each system that you want to apply it. For example, if you want the root system and a vsys to apply Web filtering, enable Web filtering in both the root system and that vsys.

To enable Web filtering:

WebUI

Security > Web Filtering > Protocol > Select **Redirect (Websense) or Redirect (SurfControl),** then click **Apply.**
Enable Web Filtering check box.

CLI

```
device-> set url protocol type { websense | scfp }
device-> set url protocol { websense | scfp }
device(url:scfp)-> set config enable
```

When Web filtering is enabled at the system level, HTTP/HTTPS requests are redirected to a Websense or SurfControl server. This action allows the device to check all HTTP/HTTPS traffic for policies (defined in that system) that require Web filtering. If you disable Web filtering at the system level, the device ignores the Web-filtering component in policies and treats the policies as " permit" policies.

4.Define the System-Level Behavioral Parameters

Define the parameters that you want the system—root or vsys—to use when applying Web filtering. One set of parameters can apply to the root system and any vsys that shares the Web-filtering configuration with the root system. Other sets can apply to virtual systems that have a dedicated Web-filtering server.

The options are as follows:

- **If connectivity to the server is lost:** If the security device loses contact with the Web-filtering server, you can specify whether to **Block** or **Permit** all HTTP/HTTPS requests.
- **Blocked URL Message Type:** If you select **NetPartners Websense/SurfControl**, the security device forwards the message it receives in the " block" response from the Websense or SurfControl server. When you select **Juniper Networks**, the device sends the message that you have previously entered in the Juniper Networks Blocked URL Message field.

NOTE: If you select **Juniper Networks**, some of the functions that Websense provides, such as redirection, are suppressed.

- **Juniper Networks Blocked URL Message:** This is the message the security device returns to the user after blocking a site. You can use the message sent from the

Websense or SurfControl server, or you can create a message (up to 500 characters) to be sent from the device.

To configure these settings, use either of the following:

WebUI

Security > Web Filtering > Protocol > Select **Redirect (Websense) or Redirect (SurfControl),** then click **Apply**.

CLI

```
device-> set url protocol type { websense | scfp }
device-> set url protocol { websense | scfp }
device(url:scfp)-> set fail-mode permit
device(url:scfp)-> set deny-message use-server
```

5.Enable Web Filtering in Individual Policies

Configure the device to contact the Web-filtering server based on the policy.

To enable Web filtering in a policy:

WebUI

Policy > Policies > Click **Edit** (edit the policy that you want Web filtering to apply), then select the **Web Filter** check box.
Select the Web-filtering profile from the drop-down list.

CLI

set policy from zone to zone *src_addr dst_addr service* permit url-filter

NOTE: The device reports the status of the Websense or SurfControl server. To update the status report, click the **Server Status** icon in the WebUI:

Security > Web Filtering > Protocol > Select **Redirect (Websense) or Redirect (SurfControl),** then click **Apply**.

Example: Redirect Web Filtering

In this example, you configure the security device to do the following:

1. Set the interfaces to work with a SurfControl server at IP address 10.1.2.5, with port number 62252 (default), and have the Web-filtering server in the Trust security zone.
2. Enable Web filtering on all outbound HTTP/HTTPS traffic from hosts in the Trust zone to hosts in the Untrust zone. If the device loses connectivity with the Web-filtering server, the device permits outbound HTTP/HTTPS traffic. When an

HTTP/HTTPS client requests access to a prohibited URL, the device sends the following message: "We're sorry, but the requested URL is prohibited. Contact ntwksec@mycompany.com."

3. Set both security zones to be in the trust-vr routing domain with the interface for the Untrust zone as ethernet3, IP address 1.1.1.1/24, and the interface for the Trust zone as ethernet1, IP address 10.1.1.1/24. Because the Web-filtering server is not in the immediate subnet of one of the device interfaces, a route is added to it through ethernet1 and the internal router at 10.1.1.250.
4. Configure the policy to enable Web filtering so that Trust to Untrust permits HTTP/HTTPS service from any source address to any destination address.

WebUI

1. **Interfaces**

   Network > Interfaces > List > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Enter the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > List > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Web-Filtering Server**

   Security > Web Filtering > Protocol: Select **Redirect (SurfControl)**, then click **Apply**. Then enter the following, and click **Apply** again:

   Enable Web Filtering: (select)
   Server Name: 10.1.2.5
   Server Port: 62252
   Communication Timeout: 10 (seconds)
   If connectivity to the server is lost ... all HTTP requests: Permit
   Blocked URL Message Type: Juniper Networks
   Juniper Blocked URL Message: We're sorry, but the requested URL is prohibited. Contact ntwksec@mycompany.com.

3. **Routes**

   Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 0.0.0.0/0

Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.2.0/24
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 10.1.1.250

4. **Policy**

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: (select), HTTP
Action: Permit
Web Filtering: (select)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Web-Filtering Server**

```
device-> set url protocol type scfp
device-> set url protocol scfp
device(url:scfp)-> set server 10.1.2.5 62252 10
device(url:scfp)-> set fail-mode permit
device(url:scfp)-> set deny-message “We’re sorry, but the requested URL is
prohibited. Contact ntwksec@mycompany.com.”
device(url:scfp)-> set config enable
```

3. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
set vrouter trust-vr route 10.1.2.0/24 interface ethernet1 gateway 10.1.1.250
```

4. **Policy**

```
set policy from trust to untrust any any http permit url-filter
save
```

Chapter 16

Deep Inspection

You can enable deep inspection (DI) in policies to examine permitted traffic and take action if the DI module in ScreenOS finds attack signatures or protocol anomalies. The following sections present the DI elements that appear in policies and explains how to configure them:

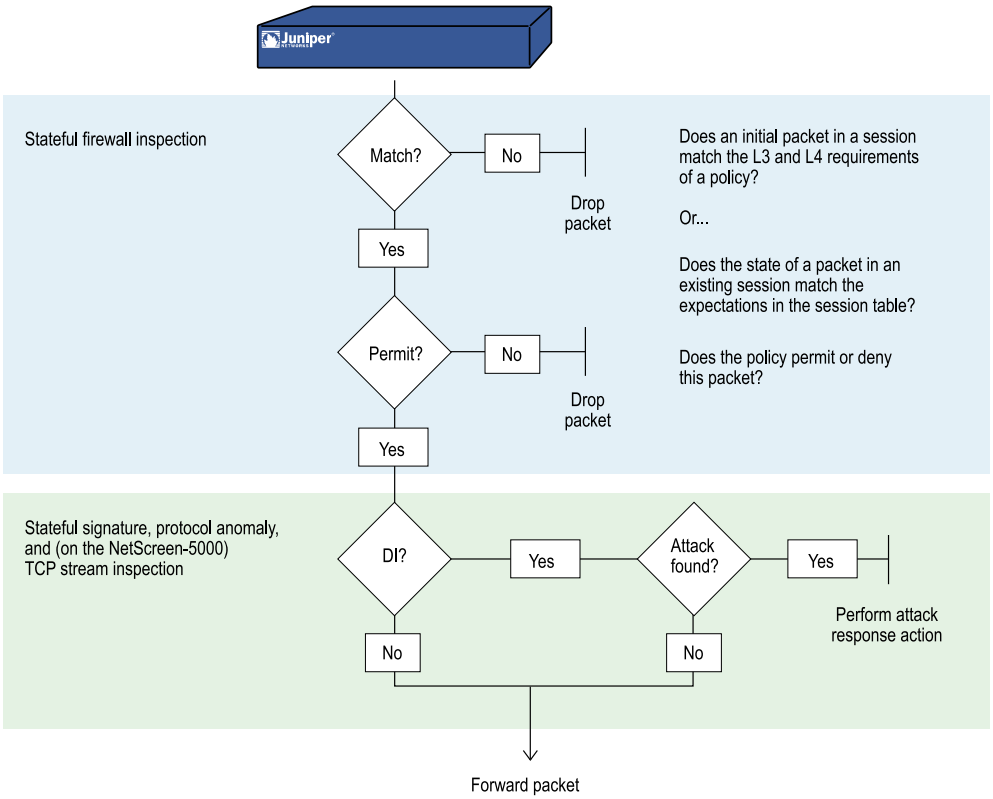
You can also enable DI at the security zone level for HTTP components. These SCREEN options are explained in the final section of this chapter:

Overview

Deep inspection (DI) is a mechanism for filtering the traffic permitted by the Juniper Networks firewall. DI examines Layer 3 and Layer 4 packet headers and Layer 7 application content and protocol characteristics in an effort to detect and prevent any attacks or anomalous behavior that might be present. Figure 140 on page 560 shows how a packet undergoes Layer 3 inspection.

NOTE: Juniper Networks security devices detect anomalous traffic patterns at Layer 3 and Layer 4 (IP and TCP) via SCREEN options set at the zone level, not the policy level. Examples of IP and TCP traffic-anomaly detection are “IP Address Sweep” on page 439, “Port Scanning” on page 440, and the various flood attacks described in “Network DoS Attacks” on page 475.

Figure 140: Stateful Firewall Inspection



When the security device receives the first packet of a session, it inspects the source and destination IP addresses in the IP packet header (Layer 3 inspection) and the source and destination port numbers and protocol in the TCP segment or UDP datagram header (Layer 4 inspection). If the Layer 3 and 4 components match the criteria specified in a policy, the device then performs the specified action on the packet—permit, deny, or tunnel. When the device receives a packet for an established session, it compares it with the state information maintained in the session table to determine if it belongs to the session.

NOTE: If the specified action is tunnel, the notion of permission is implied. Note that if you enable DI in a policy whose action is tunnel, the security device performs the specified DI operations before encrypting an outbound packet and after decrypting an inbound packet.

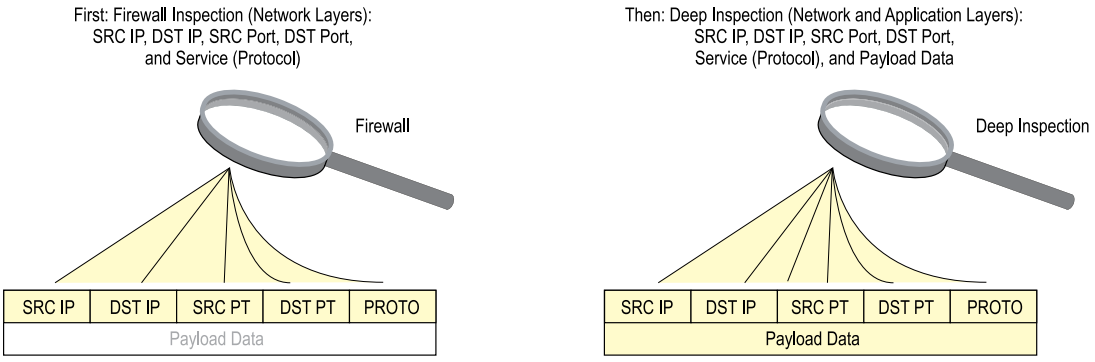
If you have enabled DI in the policy that applies to this packet and the policy action is “permit” or “tunnel,” then the security device further inspects it and its associated data stream for attacks. It scans the packet for patterns that match those defined in one or more groups of attack objects. Attack objects can be attack signatures or protocol anomalies, which you can either define yourself or download to the security device from a database server. (For more information, see “Attack Objects and Groups” on page 574 and “Customized Attack Objects and Groups” on page 599.)

NOTE: The deep inspection (DI) feature is available after you have obtained and loaded an advanced mode license key. (If you upgrade from a pre-5.0.0 version of ScreenOS, the mode automatically becomes "advanced." In this case, an advanced-mode license key is not required.) The ability to download signature packs from the database server requires that you first subscribe for the service. For more information, see "Registration and Activation of Subscription Services" on page 300.

Based on the attack objects specified in the policy, the security device might perform the following inspections (see Figure 141 on page 561):

- Examine header values and payload data for stateful attack signatures
- Compare the format of the transmitted protocol with the standards specified in the RFCs and RFC extensions for that protocol to determine if someone has altered it, possibly for malicious purposes

Figure 141: Firewall Inspection Versus Deep Inspection



If the security device detects an attack, it performs the action specified for the attack object group to which the matching attack object belongs: close, close-client, close-server, drop, drop-packet, ignore, or none. If it does not find an attack, it forwards the packet. (For more information about attack actions, see "Attack Actions" on page 582.)

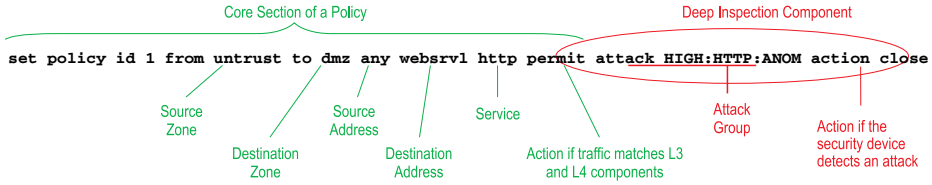
You can conceptually separate a **set policy** command into two parts—the core section and the DI component:

- The core section contains the source and destination zones, source and destination addresses, one or more services, and an action.
- The DI component instructs the security device to inspect traffic permitted by the core section of the policy for patterns matching the attack objects contained in one or more attack object groups. If the security device detects an attack object, it then performs the action defined for the corresponding group.

NOTE: You can optionally add other extensions to the core component of a **set policy** command: VPN and L2TP tunnel references, a schedule reference, address translation specifications, user authentication specifications, antivirus checking, logging, counting, and traffic management settings. Whereas these extensions are optional, the elements that constitute the core of a policy—source and destination zones, source and destination addresses, service (or services), and action—are required. (An exception to this is a global policy, in which no source and destination zones are specified: set policy global *src_addr dst_addr service action*. For more information about global policies, see "Global Policies" on page 200.)

The following **set policy** command includes a DI component:

Figure 142: DI Component in the Set Policy Command


Core Section of a Policy
Deep Inspection Component
set policy id 1 from untrust to dmz any websrvl http permit attack HIGH:HTTP:ANOM action close
Source Zone
Destination Zone
Source Address
Destination Address
Service
Action if traffic matches L3 and L4 components
Attack Group
Action if the security device detects an attack

The above command directs the security device to permit HTTP traffic from any address in the Untrust zone to the destination address “websrv1” in the DMZ zone. It also instructs the device to inspect all HTTP traffic permitted by this policy. If any pattern in the traffic matches an attack object defined in the attack object group “HIGH:HTTP:ANOM”, the device closes the connection by dropping the packet and sending TCP RST notifications to the hosts at the source and destination addresses.

It is possible to enter the context of an existing policy by using its ID number. For example:

```
device-> set policy id 1
device(policy:1)->
```

NOTE: The command prompt changes to signal that the subsequent command will be within a particular policy context.

Entering a policy context is convenient if you want to enter several commands related to a single policy. For example, the following set of commands creates a policy that permits HTTP and HTTPS traffic from the any address in the Untrust to websrv1 and websrv2 in the DMZ zone and looks for high and critical HTTP stateful signature and protocol anomaly attacks:

```
device-> set policy id 1 from untrust to dmz any websrv1 http permit attack
CRITICAL:HTTP:ANOM action close
device-> set policy id 1
device(policy:1)-> set dst-address websrv2
device(policy:1)-> set service https
device(policy:1)-> set attack CRITICAL:HTTP:SIGS action close-server
device(policy:1)-> set attack HIGH:HTTP:ANOM action drop
device(policy:1)-> set attack HIGH:HTTP:SIGS action close-server
device(policy:1)-> exit
device-> save
```

The above configuration permits both HTTP and HTTPS traffic, but only looks for attacks in HTTP traffic. To be able to add attack object groups within a policy context, you must first specify a DI attack and action in the top-level command. In the above example, you can add CRITICAL:HTTP:SIGS, HIGH:HTTP:ANOM, and HIGH:HTTP:SIGS attack object groups because you first configured the policy for DI with the CRITICAL:HTTP:ANOM group.

NOTE: You can specify a different attack action for each attack object group in a policy. If the security device simultaneously detects multiple attacks, it applies the most severe action, which in the above example is “close.” For information about the seven attack actions, including their severity levels, see “Attack Actions” on page 582.

Attack Object Database Server

The attack object database server contains all the predefined attack objects, organized into attack object groups by protocol and severity level. Juniper Networks stores the attack object database on a server at https://services.netscreen.com/restricted/sigupdates.

Predefined Signature Packs

The attack object database is organized into four signature packs: base, server protection, client protection, and worm mitigation. This approach is ideal because of the limited device memory and increased protocol support in the signature packs. Table 56 on page 566 describes each of the predefined signature packs and the threat coverage.

Table 56: Predefined Signature Packs

Signature Pack	Description	Threat Coverage
Base	A selected set of signatures for client/server and worm protection optimized for remote and branch offices along with small/medium businesses. *Note:* As a result of the memory allocation required for new enhancements, only DI signatures of critical severity are provided for NS-5XT/GT devices.	Includes a sample of worm, client-to-server, and server-to-client signatures for Internet-facing protocols and services, such as HTTP, DNS, FTP, SMTP, POP3, IMAP, NetBIOS/SMB, MS-RPC, and IM (AIM, YMSG, MSN, and IRC).
Server protection	For small/medium enterprises and remote and branch offices of large enterprises needing perimeter defense and compliance for server infrastructure, such as IIS, and Exchange.	Primarily focuses on protecting a server farm. It includes a comprehensive set of server-oriented protocols, such as HTTP, DNS, FTP, SMTP, IMAP, MS-SQL, and LDAP. Also includes worm signatures that target servers.
Client protection	For small/medium enterprises and remote and branch offices of large enterprises needing perimeter defense and compliance for hosts (desktops, laptops, and so on).	Primarily focuses on protecting users from getting malware, Trojans, and so on while surfing the Internet. Includes a comprehensive set of client-oriented protocols, such as HTTP, DNS, FTP, IMAP, POP3, and IM (AIM, YMSG, MSN, and IRC). Also includes worm signatures that target clients.
Worm Mitigation	For remote and branch offices of large enterprises along with small/medium businesses to provide the most comprehensive defense against worm attacks.	Includes stream signatures and primarily focuses on providing comprehensive worm protection. Detects server-to-client and client-to-server worm attacks for all protocols.

Table 57 on page 567 lists the predefined signatures packs with the corresponding URLs.

Table 57: URLs for Predefined Signature Packs

Signature Pack	URL
Base (default)	https://services.netscreen.com/restricted/sigupdates The security device uses this URL by default.
Server	https://services.netscreen.com/restricted/sigupdates/server
Client	https://services.netscreen.com/restricted/sigupdates/client
Worm-mitigation	https://services.netscreen.com/restricted/sigupdates/worm

Updating Signature Packs

Juniper Networks stores the four predefined signature packs on an attack object database server at https://services.netscreen.com/restricted/sigupdates. To gain access to this database server, you must first subscribe to the DI signature service for your device as described in the next section.

There are four ways to update the database:

- "Immediate Update" on page 568
- "Automatic Update" on page 569
- "Automatic Notification and Immediate Update" on page 570
- "Manual Update" on page 571

NOTE: You can also use NSM to download the signature packs. For more information, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

Before You Start Updating Attack Objects

Before you start downloading and updating attack objects, you must do the following:

1. Register your security device and obtain an authorization code.
2. Purchase a license key and activate a subscription for DI.
3. Verify that the system clock and the Domain Name System (DNS) settings on your device are accurate.

   For more information, see "Registration and Activation of Subscription Services" on page 300.

WebUI

Configuration > Date/Time

Network > DNS > Host

4. Click the **Update Now** button.

   Note that this option is only available after you retrieve a DI subscription key.

   The security device then attempts to contact the server at the default URL: https://services.netscreen.com/restricted/sigupdates; or, if you have entered a different URL in the Database Server field, it attempts to contact the URL that you entered. Table 57 on page 567 lists the predefined signatures packs and the corresponding URLs.

   After a few moments, a message appears indicating whether the update was successful. If the update was unsuccessful, then check the event log to determine the cause of the failure.

NOTE: After you download the signature pack the first time, you must reset the security device. Following each download thereafter, resetting the device is unnecessary.

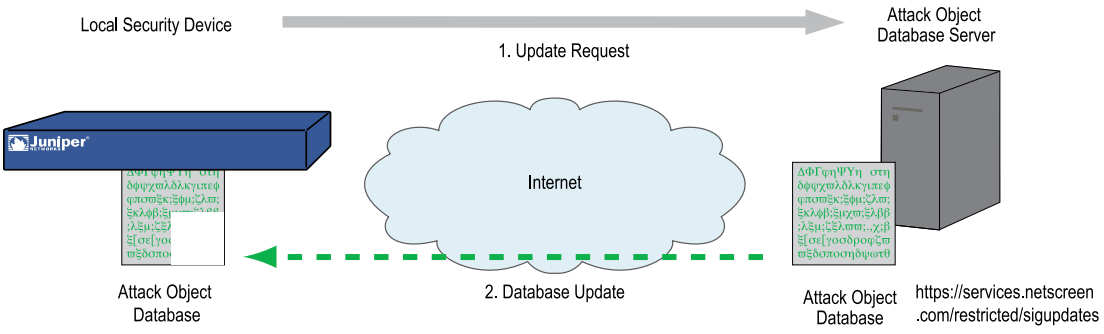
Immediate Update

The **Immediate Update** option allows you to update the signature pack on the security device immediately with the signature pack stores on the database server. For this operation to work, you must first configure the attack object database server settings.

In this example (see Figure 143 on page 568), you save a predefined signature pack from the attack object database server to the security device immediately.

You do not set a schedule for updating the database on the security device. Instead, you save the database from the server to the security device immediately.

Figure 143: Updating DI Signatures Immediately



WebUI

Configuration > Update > Attack Signature:
Signature Pack: Client

Click the **Update Now** button.

CLI

```
set attack db sigpack client
exec attack-db update
Loading attack database.....
Done.
Done.
Switching attack database...Done
Saving attack database to flash...Done.
```

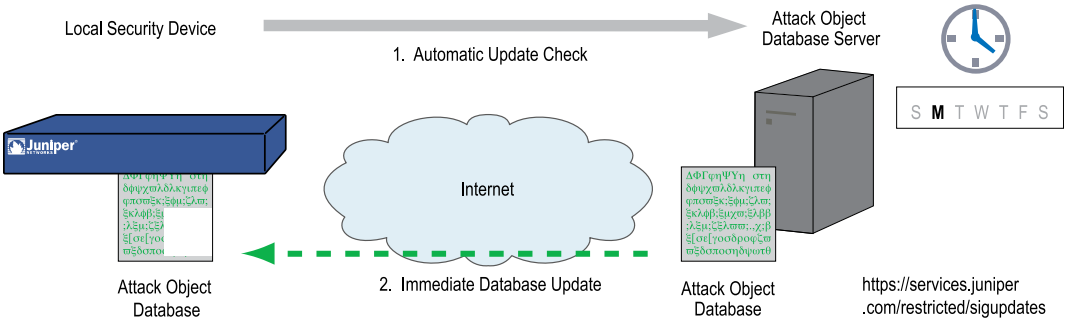
Automatic Update

The **Automatic Update** option, downloads the signature pack at user-scheduled times if the database on the server is a newer version than that previously loaded on the device. Juniper Networks regularly updates the signature pack with newly discovered attack patterns. Therefore, because of its changing nature, you must also update the signature pack on your security device regularly. For this operation to work, you must first configure the attack object database server settings.

In this example (see Figure 144 on page 569), you set a schedule to update the database on the security device every Monday at 04:00 AM. At that scheduled time, the device compares the version of the database on the server with that on the device. If the version on the server is more recent, the security device automatically replaces its database with the newer version.

For example, select Server to update the server signature pack. See Table 57 on page 567 for a list of predefined signatures packs and the corresponding URLs.

Figure 144: Updating DI Signatures Automatically



WebUI

Configuration > Update > Attack Signature: Enter the following, then click **OK**:

```
Signature Pack: Server
Update Mode: Automatic Update
Schedule:
      Weekly on: Monday
Time (hh:mm): 04:00
```

NOTE: If you schedule monthly updates and the date you choose does not occur in a particular month (for example, 31), the security device uses the last possible date of the month in its place.

CLI

```
set attack db sigpack server
set attack db mode update
set attack db schedule weekly monday 04:00
save
```

Automatic Notification and Immediate Update

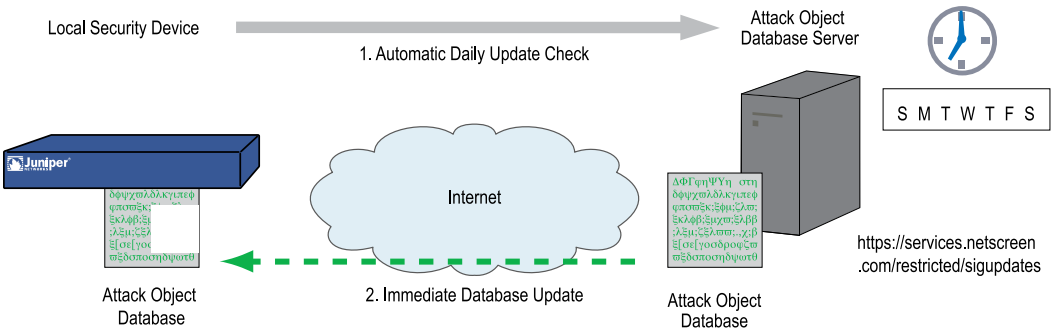
The **Automatic Notification** and I**mmediate Update** option allows you to check at user-scheduled times if the data on the database server is more recent than that on the security device. If the data on the server is more recent, a notice appears on the Home page in the WebUI, and in the CLI after you log into the security device. You can then enter the exec attack-db update command or click the Update Now button on the Configuration > Update > Attack Signature page in the WebUI to save the signature pack from the server to the device. For the server-checking operation semi-automatic procedure to work, you must first configure the attack object database server settings.

In this example (see Figure 145 on page 570), you set a schedule to check the database on the security device every day at 07:00 AM.

When you receive a notice that the database on the server has been updated, you click the **Update Now** button on the Configuration > Update > Attack Signature page in the WebUI or enter the **exec attack-db update** command to save the database from the server to the device.

For example, do the following to update the Client signature pack. See Table 57 on page 567 for a list of predefined signatures packs and the corresponding URLs.

Figure 145: Notifying Signature Updates



WebUI

1. **Scheduled Database Checking**

   Configuration > Update > Attack Signature: Enter the following, then click **OK**:

   Signature Pack: Client
   Update Mode: Automatic Notification
   Schedule:
   Daily
   Time (hh:mm): 07:00

2. **Immediate Database Update**

   When you receive a notice that the attack database on the server is more current than the one on the security device, do the following:

   Configuration > Update > Attack Signature

   Signature pack: Client
   Click the **Update Now** button.

CLI

1. **Scheduled Database Checking**

   ```
   set attack db sigpack client
   set attack db mode notification
   set attack db schedule daily 07:00
   ```

2. **Immediate Database Update**

   When you receive a notice that the attack database on the server is more current than the one on the security device, do the following:

   ```
   exec attack-db update
   ```

Manual Update

The **Manual Update** option, allows you to first use a browser to download the signature pack to a local directory or TFTP server directory. You can then load the signature pack on the security device using either the WebUI (from the local directory) or CLI (from the TFTP server directory).

NOTE: Before performing an immediate database update, you can use the **exec attack-db check** command to check if the attack object database on the server is more recent than the one on the security device.

In this example (see Figure 146 on page 572), you manually save the latest signature pack to the local directory “C:\netscreen\attacks-db” (if you want to use the WebUI to load the database) or C:\Program Files\TFTP Server (if you want to use the CLI to load it). You then load the database on the security device from your local directory.

NOTE: After downloading the signature pack, you can also post it on a local server and set it up for other security devices to access. The admins for the other devices must then change the database server URL to that of the new location. They can either enter the new URL in the Database Server field on the Configuration > Update > Attack Signature page or use the following CLI command: **set attack db server** url_string.

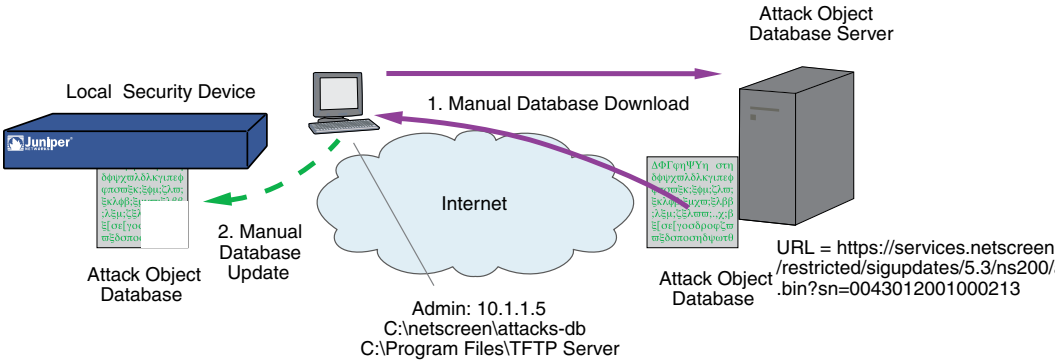
For an automatic update, the security device automatically adds the following elements to the URL:

- Serial number of the security device
- Number of the major ScreenOS version running on the device
- Platform type

When you manually update the DI Signatures, you must add these elements yourself. In this example, the serial number is 0043012001000213, the ScreenOS version is 5.4, and the platform is NetScreen-208 (ns200). Consequently, the resulting URL is:

```
https://services.netscreen.com/restricted/sigupdates/5.4/ns200/
  attacks.bin?sn=0043012001000213
```

Figure 146: Updating DI Signatures Manually



1.Downloading the Signature Pack

To save the signature pack to your local server, enter the following URL in the address field of your browser. See Table 6 on page 125 for a list of predefined signatures packs and the corresponding URLs.

```
https://services.netscreen.com/restricted/sigupdates/5.4/ns200/
  attacks.bin?sn=0043012001000213
```

Save attacks.bin to the local directory “C:\netscreen\attacks-db” (for loading via the WebUI) or to your TFTP server directory C:\Program Files\TFTP Server (when you want to use the CLI to load it).

2.Updating the Signature Pack

WebUI

Configuration > Update > Attack Signature: Enter the following, then click **OK**:

Deep Inspection Signature Update:
Load File: Enter **C:\netscreen\attacks-db\attacks.bin**

Or

Click **Browse** and navigate to that directory, select **attacks.bin**, then click **Open**.

If you downloaded the server, client, or worm protection signature packs, then enter the appropriate filename.

CLI

```
save attack-db from tftp 10.1.1.5 attacks.bin to flash
```

Updating DI Patterns from a Proxy Server

You can update the DI patterns from a proxy server. This update does not require Internet connectivity and is done offline.

To configure a proxy server:

WebUI

Security > Proxy: Set the HTTP and SSL proxy addresses, then click **Apply:**

HTTP Proxy: 10.0.0.5:8080
SSL Proxy: 10.0.0.5:443

CLI

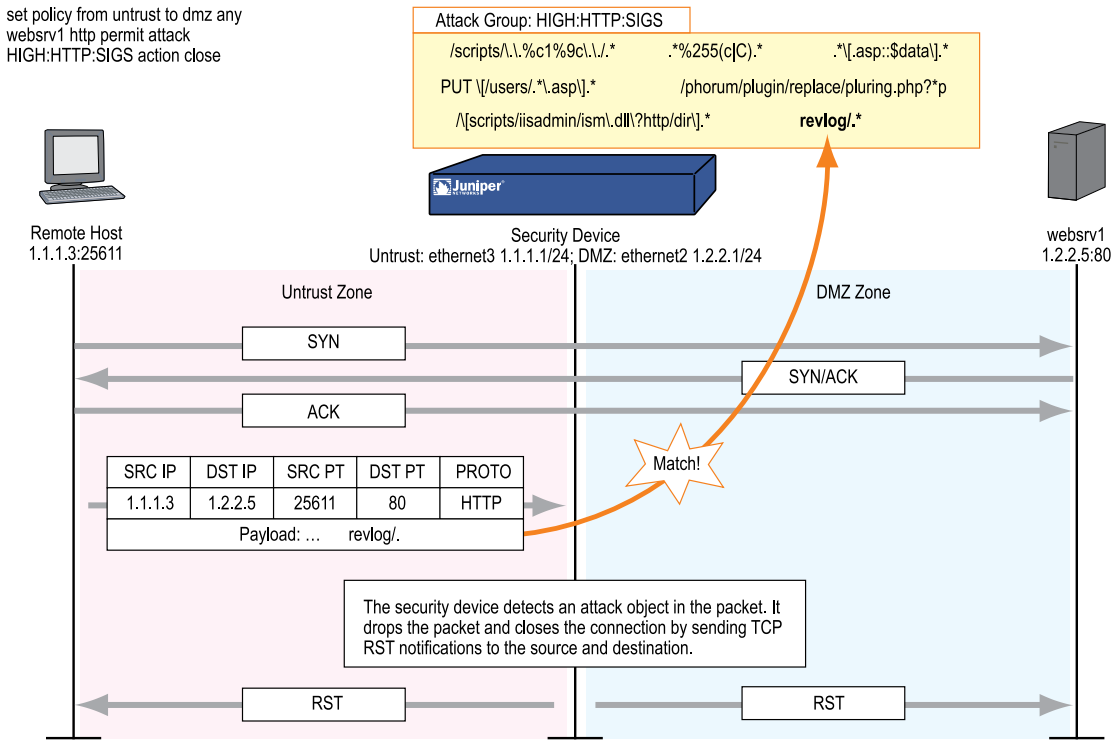
```
set pattern-update proxy http 10.0.0.5:8080
save
```

NOTE: You cannot configure an HTTPs proxy, because you cannot cache an HTTPs proxy.

Attack Objects and Groups

Attack objects are stateful signatures, stream signatures (on the NetScreen-5000 series), and protocol anomalies that a security device uses to detect attacks aimed at compromising one or more hosts on a network. Attack objects are in groups organized by protocol type and then by severity. When you add deep inspection (DI) to a policy, the device inspects the traffic that the policy permits for any patterns matching those in the referenced attack object group (or groups).

Figure 147: Attack Objects and Groups



The attack object groups that you reference in the DI component of a policy must target the same service type that the policy permits. For example, if the policy permits SMTP traffic, the attack object group must aim at attacks on SMTP traffic. The following policy exemplifies a valid configuration:

set policy id 2 from trust to untrust any any **smtp** permit attack **CRIT:SMTP:SIGS** action close

The next policy is erroneous because the policy permits SMTP traffic, but the attack object group is for POP3 traffic:

set policy id 2 from trust to untrust any any **smtp** permit attack **CRIT:POP3:SIGS** action close

The second policy is configured incorrectly and, if implemented, would cause the security device to expend unnecessary resources inspecting SMTP traffic for POP3 attack objects that it could never find. If policy 2 permits both SMTP and POP3 traffic, you can configure the DI component to check for SMTP attack objects, POP3 attack objects, or for both.

```
set group service grp1
set group service grp1 add smtp
set group service grp1 add pop3
set policy id 2 from trust to untrust any any grp1 permit attack CRIT:SMTP:SIGS
action close
set policy id 2 attack CRIT:POP3:SIGS action close
```

Supported Protocols

The deep inspection (DI) module supports stateful signature attack objects and protocol anomaly attack objects for the following protocols and applications:

Table 58: Basic Network Protocols

Protocol	Stateful Signature	Protocol Anomaly	Definition
DNS	Yes	Yes	Domain Name System (DNS) is a database system for translating domain names to IP addresses, such as www.juniper.net = 207.17.137.68.
FTP	Yes	Yes	File Transfer Protocol (FTP) is a protocol for exchanging files between computers across a network.
HTTP	Yes	Yes	HyperText Transfer Protocol (HTTP) is a protocol primarily used to transfer information from Web servers to Web clients.
IMAP	Yes	Yes	Internet Mail Access Protocol (IMAP) is a protocol that provides incoming email storage and retrieval services, with the option that users can either download their email or leave it on the IMAP server.
NetBIOS	Yes	Yes	NetBIOS (Network Basic Input Output System) is an application interface that allows applications on users' workstations to access network services provided by network transports such as NetBEUI, SPX/IPX, and TCP/IP.
POP3	Yes	Yes	Post Office Protocol, version 3 (POP3) is a protocol that provides incoming email storage and retrieval services.
SMTP	Yes	Yes	Simple Mail Transfer Protocol (SMTP) is a protocol for transferring email between mail servers.
Chargen	Yes	Yes	Character generator protocol
DHCP	Yes	Yes	Dynamic Host Configuration Protocol is used to control vital networking parameters of hosts (running clients) with the help of a server. DHCP is backward compatible with BOOTP.
Discard	Yes	Yes	Discard protocol is a useful debugging and measurement tool. A discard service simply throws away any data it receives.

Table 58: Basic Network Protocols *(continued)*

Protocol	Stateful Signature	Protocol Anomaly	Definition
Echo	Yes	Yes	Echo protocol is an internet protocol intended for testing and measurement purposes. A host may connect to a server that supports the ECHO protocol, on either TCP or UDP port 7. The server then sends back any data it receives.
Finger	Yes	Yes	Finger User Information protocol is a simple protocol that provides an interface to a remote user information program.
Gopher	Yes	Yes	Gopher is an internet protocol designed for distributed document search and retrieval.
ICMP	Yes	Yes	Internet Control Message Protocol is a required protocol tightly integrated with IP. ICMP messages, delivered in IP packets, are used for out-of-band messages related to network operation.
IDENT	Yes	Yes	Identification protocol provides a means to determine the identity of a user of a particular TCP connection.
LDAP	Yes	Yes	Lightweight Directory Access Protocol is a set of protocols for accessing information directories.
LPR	Yes	Yes	Line Printer spooler
NFS	Yes	Yes	Network File System (NFS) protocol provides transparent remote access to shared files across networks. The NFS protocol is designed to be portable across different machines, operating systems, network architectures, and transport protocols.
NNTP	Yes	Yes	Network News Transfer Protocol specifies a protocol for the distribution, inquiry, retrieval, and posting of news articles using a reliable stream-based transmission of news.
NTP	Yes	Yes	Network Time Protocol and Simple Network Time Protocol is used to synchronize the time of a computer client or server to another server or reference time source, such as a radio or satellite receiver or modem.
Portmapper	Yes	Yes	Port Mapper Program Protocol maps RPC program and version numbers to transport- specific port numbers.
RADIUS	Yes	Yes	Remote Authentication Dial In User Service, an authentication and accounting system used by many Internet Service Providers (ISPs).
rexec	Yes	Yes	Remote Execution
rlogin	Yes	Yes	Remote Login occurs when a user connects to an Internet host to use its native user interface.
rsh	Yes	Yes	Remote shell
RTSP	Yes	Yes	Real Time Streaming Protocol is a client-server application-level protocol for controlling the delivery of data with real-time properties. It establishes and controls either a single or several time-synchronized streams of continuous media, such as audio and video.
SNMPTRAP	Yes	Yes	Simple Network Management Protocol is an SNMP application that uses the SNMP TRAP operation to send information to a network manager.

Table 58: Basic Network Protocols *(continued)*

Protocol	Stateful Signature	Protocol Anomaly	Definition
SSH	Yes	Yes	Secure Shell Protocol is a protocol for secure remote login and other secure network services over an insecure network.
SSL	Yes	Yes	Secure Sockets Layer is a protocol used for transmitting private documents via the Internet using a cryptographic system.
syslog	Yes	Yes	System Logging Protocol is used for the transmission of event notification messages across networks.
Telnet	Yes	Yes	Telnet protocol is a terminal emulation program for TCP/IP networks. This protocol enables you to communicate with other servers on the network.
TFTP	Yes	Yes	Trivial File Transfer Protocol is a simple protocol used to transfer files. TFTP uses the User Datagram Protocol (UDP) and provides no security features.
VNC	Yes	Yes	Virtual Network Computing is a desktop protocol to remotely control another computer.
Whois	Yes	Yes	Network Directory Service Protocol is a TCP transaction based query/response server that provides network-wide directory service to internet users.

Table 59: Instant Messaging Applications

Protocol	Stateful Signature	Protocol Anomaly	Definition
AIM	Yes	Yes	America Online Instant Messaging (AIM) is the instant messaging application for America Online.
MSN Messenger	Yes	Yes	Microsoft Network Messenger (MSN Messenger) is the instant messaging service provided by Microsoft.
Yahoo! Messenger	Yes	Yes	Yahoo! Messenger is the instant messaging service provided by Yahoo!.
IRC	Yes	Yes	Internet Relay Chat is a text-based protocol, with the simplest client being any socket program capable of connecting to the server.

Table 60: Application Layer Gateways (ALGs)

Protocol	Stateful Signature	Protocol Anomaly	Definition
MSRPC	Yes	Yes	MSRPC (Microsoft-Remote Procedure Call) is a mechanism for running processes on a remote computer.

If the security device has access tohttp://help.juniper.net/sigupdates/english , you can see the contents of all the predefined attack object groups and descriptions of the

predefined attack objects. Open your browser, and enter one of the following URLs in the Address field:

http://help.juniper.net/sigupdates/english/AIM.html
http://help.juniper.net/sigupdates/english/DNS.html
http://help.juniper.net/sigupdates/english/FTP.html
http://help.juniper.net/sigupdates/english/GNUTELLA
http://help.juniper.juniper.net/sigupdates/english/HTTP.html
http://help.juniper.net/sigupdates/english/IMAP.html
http://help.juniper.net/sigupdates/english/MSN.html
http://help.juniper.net/sigupdates/english/NBDS.html
http://help.juniper.net/sigupdates/english/NBNAME.html
http://help.juniper.net/sigupdates/english/POP3.
juniper.net/sigupdates/english/SMTP.html
http://help.juniper.net/sigupdates/english/MSRPC.html
http://help.juniper.net/sigupdates/english/SMB.html
http://help.juniper.net/sigupdates/english/YMSG.html

Each of the above URLs links to an HTML page containing a list of all the predefined attack objects—organized in groups by severity—for a particular protocol. To see a description of an attack object, click its name.

Stateful Signatures

An attack signature is a pattern that exists when a particular exploit is in progress. The signature can be a Layer 3 or 4 traffic pattern, such as when one address sends lots of packets to different port numbers at another address (see “Port Scanning” on page 440), or a textual pattern, such as when a malicious URL string appears in the data payload of a single HTTP or FTP packet. The string can also be a specific segment of code or a specific value in the packet header. However, when searching for a textual pattern, the deep inspection (DI) module in a security device looks for more than just a signature in a packet; it looks for the signature in a particular portion of the packet (even if fragmented or segmented), in packets sent at a particular time in the life of the session, and sent by either the connection initiator or the responder.

NOTE: Because the DI module supports regular expressions, it can use wildcards when searching for patterns. Thus, a single attack signature definition can apply to multiple attack pattern variations. For information about regular expressions, see “Regular Expressions” on page 600.

When the DI module checks for a textual pattern, it considers the roles of the participants as client or server and monitors the state of the session to narrow its search to just those elements relevant to the exploit for which attackers use the pattern. Using contextual information to refine packet examination greatly reduces false alarms—or “false positives” —and avoids unnecessary processing. The term “stateful signatures” conveys this concept of looking for signatures within the context of the participants’ roles and session state.

To see the advantage of considering the context in which a signature occurs, note the way the DI module examines packets when enabled to detect the EXPN Root attack. Attackers use the EXPN Root attack to expand and expose mailing lists on a

mail server. To detect the EXPN Root attack, the security device searches for the signature "expn root" in the control portion of a Simple Mail Transfer Protocol (SMTP) session. The device examines only the control portion because that is only where the attack can occur. If "expn root" occurs in any other portion of the session, it is not an attack.

Using a simple textual packet signature detection technique, the signature "expn root" triggers an alarm even if it appears in the data portion of the SMTP connection; that is, in the body of an email message. If, for example, you were writing to a colleague about EXPN Root attacks, a simple packet signature detector would regard this as an attack. Using stateful signatures, the DI module can distinguish between text strings that signal attacks and those that are harmless occurrences.

NOTE: For a list of protocols for which there are predefined stateful signature attack objects, see "Supported Protocols" on page 575.

TCP Stream Signatures

Like a stateful signature, a TCP stream signature is a pattern that exists when an exploit is in progress. However, when the DI module examines traffic for stateful signatures, it searches only within specific contexts. When the DI module examines traffic for TCP stream signatures, it does so without regard for contexts. Another distinction between the two types of signatures is that although stateful signatures can be predefined or user-defined, TCP stream signatures must be user-defined. After you add a stream signature attack object to an attack object group, you can then reference that group in a policy that applies DI. (For more about TCP stream signatures, see "TCP Stream Signature Attack Objects" on page 604.)

NOTE: You can define TCP stream signatures on the high-end systems only.

Stream signatures are independent of protocols and are therefore more flexible in matching traffic. Stream signatures can examine traffic where protocols decoders can't inspect. However, this flexibility affects performance and resource consumption.

Stream signatures consume resources and affect performance, so they must be used sparingly. Stream256 signatures however, operate the same way, but rather than matching over the entire stream, they only match on the first 256 bytes of the stream. Therefore, they consume fewer resources and are less of a performance hit.

Protocol Anomalies

Attack objects that search for protocol anomalies detect traffic that deviates from the standards defined in RFCs and common RFC extensions. With signature attack objects, you must use a predefined pattern or create a new one; therefore, they can only detect known attacks. Protocol anomaly detection is particularly useful for catching new attacks or those attacks that cannot be defined by a textual pattern.

NOTE: For a list of protocols for which there are predefined protocol anomaly attack objects, see "Supported Protocols" on page 575.

Attack Object Groups

Predefined attack object groups contain attack objects for a specific protocol. For each protocol, the groups are separated into protocol anomalies and stateful signatures, and then roughly organized by severity. The five attack object group severity levels are critical, high, medium, low, and info:

- **Critical:** Contains attack objects matching exploits that attempt to evade detection, cause a network device to crash, or gain system-level privileges.
- **High:** Contains attack objects matching exploits that attempt to disrupt a service, gain user-level access to a network device, or activate a Trojan horse previously loaded on a device.
- **Medium:** Contains attack objects matching exploits that detect reconnaissance efforts attempting to access vital information through directory traversal or information leaks.
- **Low:** Contains attack objects matching exploits that attempt to obtain non-critical information or scan a network with a scanning tool.
- **Info:** Contains attack objects matching normal, harmless traffic containing URLs, DNS lookup failures, and SNMP public community strings. You can use informational attack objects to obtain information about your network.

Changing Severity Levels

Although attack object groups are classified by protocol and severity level (critical, high, medium), each attack object has its own specific severity level: critical, high, medium, low, info. These attack object severity levels map to severity levels for event log entries as follows:

Table 61: Attack Object Severity Levels

Attack Object Severity Level – Maps to –	Event Log Entry Severity Level
Critical	Critical
High	Error
Medium	Warning
Low	Notification
Info	Information

For example, if the security device detects an attack with the severity level "Medium," the corresponding entry that appears in the event log then has the severity level "Warning."

It is possible to override the default severity level of all attack objects in one or more attack object groups referenced in a policy. You do this at the policy level by entering the context of an existing policy and then assigning a new severity level to all the attack object groups that the policy references.

The following shows how to change the severity level of the attack object groups referenced in a policy through the WebUI and CLI:

WebUI

Policies > Edit (for an existing policy): Do the following, then click **OK**:

> Deep Inspection: Select a severity option in the Severity drop-down list, then click **OK**.

CLI

```
device-> set policy id number
device(policy:number)-> set di-severity { info | low | medium | high | critical }
```

To return the severity level for each attack object to its original setting, you again enter the context of a policy and do either of the following:

WebUI

Policies > Edit (for an existing policy): Do the following, then click **OK**:

> Deep Inspection: Select **Default** in the Severity drop-down list, then click **OK**.

CLI

```
device-> set policy id number
device(policy:number)-> unset di-severity
```

Disabling Attack Objects

When you reference an attack object group in a policy, the security device checks the traffic to which the policy applies for patterns matching any of the attack objects in that group. At some point, you might not want to use a particular attack object if it repeatedly produces false-positives; that is, if it erroneously interprets legitimate traffic as an attack. If the problem stems from a custom attack object, you can simply remove it from its custom attack object group. However, you cannot remove a predefined attack object from a predefined attack object group. In that case, the best course of action is to disable the object.

Note that a predefined attack object is disabled only within the root system or virtual system (vsys) in which you disable it. For example, disabling a predefined attack object in the root system does not automatically disable it in any virtual systems.

Likewise, disabling an attack object in one vsys does not affect that object in any other vsys.

NOTE: Disabling attack objects does not improve throughput performance.

To disable an attack object:

WebUI

Security > Deep Inspection > Attacks > Predefined: Clear the check box in the **Configure** column for the attack object that you want to disable.

CLI

```
set attack disable attack_object_name
```

To re-enable a previously disabled attack object:

WebUI

Security > Deep Inspection > Attacks > Predefined: Select the check box in the **Configure** column for the attack object that you want to enable.

CLI

```
unset attack disable attack_object_name
```

Attack Actions

When the security device detects an attack, it performs the action that you specify for the attack group containing the object that matches the attack. The seven actions are as follows, from most to least severe:

- **Close** (severs connection and sends RST to client and server)

NOTE: The client is always the initiator of a session; that is, the source address in a policy. The server is always the responder, or the destination address.

  Use this option for TCP connections. The security device drops the connection and sends a TCP RST to both the client (source) and server (destination). Because the delivery of RST notifications is unreliable, by sending a RST to both client and server, there is a greater chance that at least one gets the RST and closes the session.

- **Close Server** (severs connection and sends RST to server)

  Use this option for inbound TCP connections from an untrusted client to a protected server. If the client tries to launch an attack, the security device drops

the connection and sends a TCP RST only to the server for it to clear its resources while the client is left hanging.

- **Close Client** (severs connection and sends RST to client)

  Use this option for outbound TCP connections from a protected client to an untrusted server. If, for example, the server sends a malicious URL string, the security device drops the connection and sends a RST only to the client for it to clear its resources while the server is left hanging.

- **Drop** (severs connection without sending anyone a RST)

  Use this option for UDP or other non-TCP connections, such as DNS. The security device drops all packets in a session, but does not send a TCP RST.

- **Drop Packet** (drops a particular packet, but does not sever connection)

  This option drops the packet in which an attack signature or protocol anomaly occurs but does not terminate the session itself. Use this option to drop malformed packets without disrupting the entire session. For example, if the security device detects an attack signature or protocol anomaly from an AOL proxy, dropping everything would disrupt all AOL service. Instead, dropping just the packet stops the problem packet without stopping the flow of all the other packets.

- **Ignore** (after detecting an attack signature or anomaly, the security device makes a log entry and stops checking—or ignores—the remainder of the connection)

  If the security device detects an attack signature or protocol anomaly, it makes an event log entry but does not sever the session itself. Use this option to tweak false positives during the initial setup phase of your deep inspection (DI) implementation. Also, use this option when a service uses a standard port number for nonstandard protocol activities; for example, Yahoo Messenger uses port 25 (SMTP port) for non-SMTP traffic. The security device logs the occurrence once per session (so that it does not fill the log with false positives), but takes no action.

- **None** (no action)

  It is useful when first identifying attack types during the initial setup phase of your DI implementation. When the security device detects an attack signature or protocol anomaly, it makes an entry in the event log but takes no action on the traffic itself. The security device continues to check subsequent traffic in that session and make log entries if it detects other attack signatures and anomalies.

You can create a policy referencing multiple attack object groups, each group having a different action. If the security device simultaneously detects multiple attacks that belong to different attack object groups, it applies the most severe action specified by one of those groups.

Example: Attack Actions—Close Server, Close, Close Client

In this example, there are three zones: Trust, Untrust, and DMZ. You have finished analyzing attacks and have concluded you need the following three policies:

- **Policy ID 1:** Permit HTTP, HTTPS, PING, and FTP-GET traffic from any address in the Untrust zone to the Web servers (websrv1 and websrv2) in the DMZ.

  Attack Settings for Policy ID 1:

- CRITICAL:HTTP:ANOM, CRITICAL:HTTP:SIGS
- HIGH:HTTP:ANOM, HIGH:HTTP:SIGS
- MEDIUM:HTTP:ANOM, MEDIUM:HTTP:SIGS
- CRITICAL:FTP:SIGS
- Action for all attack object groups: Close Server
- Logging: Enabled (default setting)

  You choose to drop the connection and send a TCP RST notification only to the protected Web servers so they can terminate sessions and clear resources. You anticipate attacks coming from the Untrust zone.

- **Policy ID 2:** Permit HTTP, HTTPS, PING, and FTP traffic from any address in the Trust zone to the Web servers (websrv1 and websrv2) in the DMZ

  Attack Settings for Policy ID 2:

- CRITICAL:HTTP:ANOM, CRITICAL:HTTP:SIGS
- HIGH:HTTP:ANOM, HIGH:HTTP:SIGS
- MEDIUM:HTTP:ANOM, MEDIUM:HTTP:SIGS
- CRITICAL:FTP:SIGS
- Action for all attack object groups: Close
- Logging: Enabled (default setting)

  You choose to drop the connection and send a TCP RST notification to both the protected clients and servers so they both can terminate their sessions and clear their resources regardless of the severity level of the attack.

- **Policy ID 3:** Permit FTP-GET, HTTP, HTTPS, PING traffic from any address in the Trust zone to any address in the Untrust zone

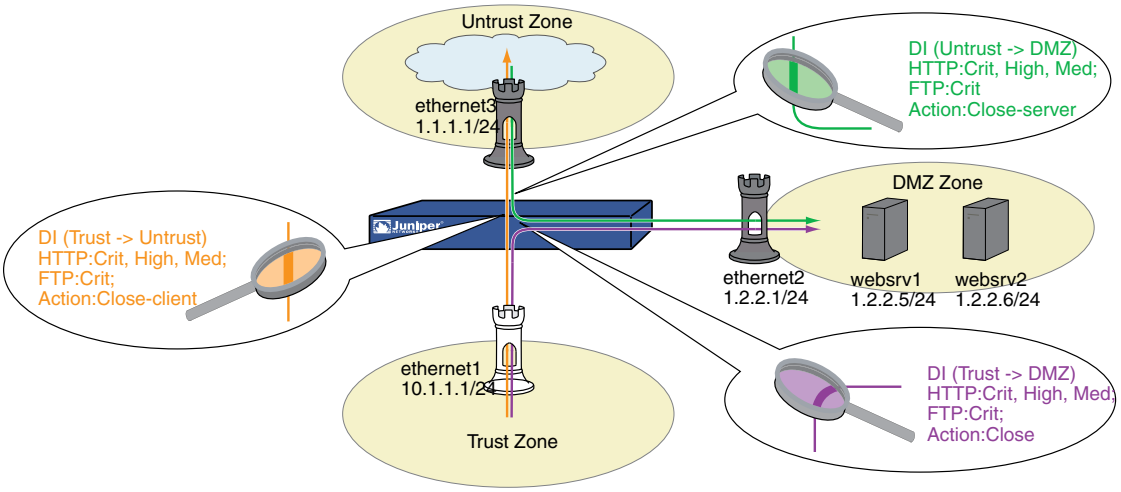
  Attack Settings for Policy ID 3:

- CRITICAL:HTTP:ANOM, CRITICAL:HTTP:SIGS
- HIGH:HTTP:ANOM, HIGH:HTTP:SIGS
- MEDIUM:HTTP:ANOM, MEDIUM:HTTP:SIGS
- CRITICAL:FTP:SIGS
- Action for all attack object groups: Close Client
- Logging: Enabled (default setting)

  You choose to drop the connection and send a TCP RST notification to the protected clients so they both can terminate their sessions and clear their resources. In this case, you anticipate an attack coming from an untrusted HTTP or FTP server.

Although the policies permit HTTP, HTTPS, Ping, and FTP-Get or FTP, the security device activates DI only for HTTP and FTP traffic. All zones are in the trust-vr routing domain.

Figure 148: DI Attack Actions



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Enter the following, then click **OK**:

Interface Mode: NAT
Service Options:
Management Services: (select all)
Other services: Ping

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: DMZ
Static IP: (select this option when present)
IP Address/Netmask: 1.2.2.1/24

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: websrv1
IP Address/Domain Name:
IP/Netmask: (select), 1.2.2.5/32
Zone: DMZ

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: websrv2
IP Address/Domain Name:
IP/Netmask: (select), 1.2.2.6/32
Zone: DMZ

3. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

4. **Policy ID 1**

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), websrv1

> Click **Multiple**, select **websrv2**, then click **OK** to return to the basic policy configuration page.

Service: HTTP

> Click **Multiple**, select **FTP-GET**, **HTTPS**, **PING**, then click **OK** to return to the basic policy configuration page.

Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Group: CRITICAL:HTTP:ANOM
Action: Close Server
Log: (select)
Group: CRITICAL:HTTP:SIGS
Action: Close Server
Log: (select)
Group: HIGH:HTTP:ANOM
Action: Close Server
Log: (select)
Group: HIGH:HTTP:SIGS
Action: Close Server
Log: (select)
Group: MEDIUM:HTTP:ANOM
Action: Close Server
Log: (select)
Group: MEDIUM:HTTP:SIGS
Action: Close Server
Log: (select)
Group: CRITICAL:FTP:SIGS
Action: Close Server
Log: (select)

5. **Policy ID 2**

Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), websrv1

> Click **Multiple**, select **websrv2**, then click **OK** to return to the basic policy configuration page.

Service: HTTP

> Click **Multiple**, select **FTP-GET**, **HTTPS**, **PING**, then click **OK** to return to the basic policy configuration page.

Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Group: CRITICAL:HTTP:ANOM
Action: Close

Log: (select)
Group: CRITICAL:HTTP:SIGS
Action: Close
Log: (select)
Group: HIGH:HTTP:ANOM
Action: Close
Log: (select)
Group: HIGH:HTTP:SIGS
Action: Close
Log: (select)
Group: MEDIUM:HTTP:ANOM
Action: Close
Log: (select)
Group: MEDIUM:HTTP:SIGS
Action: Close
Log: (select)
Group: CRITICAL:FTP:SIGS
Action: Close
Log: (select)

6. **Policy ID 3**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: HTTP

> Click **Multiple**, select **FTP-GET**, **HTTPS**, **PING**, then click **OK** to return to the basic policy configuration page.

Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Group: CRITICAL:HTTP:ANOM
Action: Close Client
Log: (select)
Group: CRITICAL:HTTP:SIGS
Action: Close Client
Log: (select)
Group: HIGH:HTTP:ANOM
Action: Close Client
Log: (select)
Group: HIGH:HTTP:SIGS
Action: Close Client
Log: (select)
Group: MEDIUM:HTTP:ANOM
Action: Close Client
Log: (select)
Group: MEDIUM:HTTP:SIGS
Action: Close Client
Log: (select)

Group: CRITICAL:FTP:SIGS
Action: Close Client
Log: (select)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 manage
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet2 zone dmz
set interface ethernet2 ip 2.1.1.1/24
```

2. **Addresses**

```
set address dmz websrv1 1.2.2.5/32
set address dmz websrv2 1.2.2.6/32
```

3. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

4. **Policy ID 1**

```
set policy id 1 from untrust to dmz any websrv1 http permit attack
CRITICAL:HTTP:ANOM action close-server
set policy id 1
device(policy:1)-> set dst-address websrv2
device(policy:1)-> set service ftp-get
device(policy:1)-> set service https
device(policy:1)-> set service ping
device(policy:1)-> set attack CRITICAL:HTTP:SIGS action close-server
device(policy:1)-> set attack HIGH:HTTP:ANOM action close-server
device(policy:1)-> set attack HIGH:HTTP:SIGS action close-server
device(policy:1)-> set attack MEDIUM:HTTP:ANOM action close-server
device(policy:1)-> set attack MEDIUM:HTTP:SIGS action close-server
device(policy:1)-> set attack CRITICAL:FTP:SIGS action close-server
device(policy:1)-> exit
```

5. **Policy ID 2**

```
set policy id 2 from trust to dmz any websrv1 http permit attack
CRITICAL:HTTP:ANOM action close
set policy id 2
device(policy:2)-> set dst-address websrv2
device(policy:2)-> set service ftp
device(policy:2)-> set service https
device(policy:2)-> set service ping
device(policy:2)-> set attack CRITICAL:HTTP:SIGS action close
device(policy:2)-> set attack HIGH:HTTP:ANOM action close
device(policy:2)-> set attack HIGH:HTTP:SIGS action close
device(policy:2)-> set attack MEDIUM:HTTP:ANOM action close
```

```
device(policy:2)-> set attack MEDIUM:HTTP:SIGS action close
device(policy:2)-> set attack CRITICAL:FTP:SIGS action close
device(policy:2)-> exit
```

6. **Policy ID 3**

```
set policy id 3 from trust to untrust any any http permit attack
CRITICAL:HTTP:ANOM action close-client
set policy id 3
device(policy:3)-> set service ftp-get
device(policy:3)-> set service https
device(policy:3)-> set service ping
device(policy:3)-> set attack CRITICAL:HTTP:SIGS action close-client
device(policy:3)-> set attack HIGH:HTTP:ANOM action close-client
device(policy:3)-> set attack HIGH:HTTP:SIGS action close-client
device(policy:3)-> set attack MEDIUM:HTTP:ANOM action close-client
device(policy:3)-> set attack MEDIUM:HTTP:SIGS action close-client
device(policy:3)-> set attack CRITICAL:FTP:SIGS action close-client
device(policy:3)-> exit
save
```

Brute Force Attack Actions

A typical brute force attack is accomplished by sending lots of traffic with varying source ports or IP in an attempt to obtain network access. In order to effectively prevent future attempts, ScreenOS allows you to associate an IP action for each attack group in a policy.

Brute force attack is detected based on the threshold values set for the DI supported protocols. For example,

set di service *protocol-namevalue*

Apart from a DI action, brute force attack actions are configured with the **IP action** command for a configured amount of time for a specified target. If your security device detects a brute force attack, then select one of the following actions to perform:

- Notify: The security device logs the event but does not take any action against further traffic matching the target definition for the period of time specified in the timeout setting.
- Block: The security device logs the event and drops all further traffic matching the target definition for the period of time specified in the timeout setting.
- Close: The security device logs the event and drops all further traffic matching the target definition for the period of time specified in the timeout setting, and sends a Reset (RST) for TCP traffic to the source and destination addresses.

Brute Force Attack Objects

Table 62 on page 591 lists the brute force attack objects in ScreenOS 5.4 and the threshold parameters that can be used with the IP actions.

Table 62: Brute Force Attack Objects

Brute Force Attack Name	Parameter
HTTP Brute Force Login Attempt	failed_logins
HTTP Brute Search Attempt	brute_search
IMAP Brute Force Login Attempt	failed_logins
LDAP Brute Force Login Attempt	failed_logins
MS-RPC IsSystemActive request flood	Not configurable—32 attempts
MS-SQL Login Brute Force	Not configurable—4 attempts
POP3 Brute Force Login Attempt	failed_login
RADIUS Brute Force Authentication Attempt	failed_auth
SMB Brute Force Directory Create/Delete	Not configurable—200 attempts
SMB Brute Force Login Attempt	failed_login
FTP Brute Force Login Attempt	failed_login
Telnet Brute Force Login Attempt	failed_login
VNC Brute Force Login Attempt	failed_login

Brute Force Attack Target

The target option specifies a set of elements that must match for the security device to consider a packet part of a brute force attack. The specified set of elements in an IP packet arriving during a specified timeout period must match that in the packet that the security device detected as part of a brute force attack for the subsequent packet to be considered part of the same attack. The default target definition is Serv. You can select any of the following target definitions shown in Table 63 on page 591.

Table 63: Target Options

Target option	Matching elements
Serv	source IP, destination IP, destination port, and protocol
Src-IP	source IP address
Zone-Serv	source security zone, destination IP, destination port number, and protocol
Dst-IP	destination IP address

Table 63: Target Options *(continued)*

Target option	Matching elements
Zone	source security zone (The security zone to which the ingress interface is bound; that is, the source security zone from which the attacking packets originate)

Brute Force Attack Timeout

Timeout is a period of time following brute force attack detection during which the security device performs an IP action on packets matching specified target parameters. The default timeout is 60 seconds.

Example 1

In this example, you configure an IP action along with the existing DI action for each group in a policy. The following CLI commands block brute force attack object—HTTP Brute Force Login Attempt or HTTP Brute Force Search for 45 seconds. All other attacks in the HIGH:HTTP:ANOM attack group are configured with a DI action of **close**.

In this release of ScreenOS, S2C HTTP protocol decoding is performed only if you configure server-to-client signature attacks. HTTP:Brute-Force, a server-to-client anomaly attack is detected if you configure an HTTP server-to-client signature attack in the policy. In the following example, HTTP:HIGH:SIGS has server-to-client signature attacks, so add HTTP:HIGH:SIGS along with HTTP:HIGH:ANOM in a policy.

CLI

```
device>get attack group HIGH:HTTP:ANOM
GROUP "HIGH:HTTP:ANOM" is pre-defined. It has the following members
ID     Name
1674 HTTP:INVALID:INVLD-AUTH-CHAR
1675 HTTP:INVALID:INVLD-AUTH-LEN
1711 HTTP:OVERFLOW:HEADER
1713 HTTP:OVERFLOW:INV-CHUNK-LEN
1717 HTTP:OVERFLOW:AUTH-OVFLW
5394 HTTP:EXPLOIT:BRUTE-FORCE
5395 HTTP:EXPLOIT:BRUTE-SEARCH
device> set policy id 1 from Untrust to DMZ Any Any Any permit attack
MEDIUM:HTTP:ANOM action none
device> set policy id 1 from Untrust to DMZ Any Any Any permit attack
MEDIUM:HTTP:HIGH:SIGS action none
device> set policy id 1
device(policy:1)-> set attack HIGH:HTTP:ANOM action close ip-action block
target dst-ip timeout 45
```

If the configured attack group does not have any brute force attack protocol anomalies, IP action is not enforced.

Example 2

In this example, you associate an IP action for each attack group for a configured amount of time from a specified host.

```
set policy id 1 from trust to untrust any any any permit attack HIGH:POP3:ANOM
action close ip-action notify target serv timeout 60
```

Example 3

In this example, the default threshold value of FTP brute force login attempt is 8 attempts per minute. If a user at IP address 192.168.2.2 is launching a FTP brute force login attempt to FTP server at 10.150.50.5 in order to figure out a user account name and password, the attempt is detected when the attacker makes 8 FTP login attempts within a minute.

If an IP action is configured to "Block" for 120 seconds for target of "serv", any traffic coming from 192.168.2.2 (src IP) to 10.150.50.5 (dst IP) over TCP (protocol) port 21 (dst port) is blocked for 120 seconds.

Note that some IP action targets may affect traffic matching another policy.

Attack Logging

You can enable the logging of detected attacks per attack group per policy. In other words, within the same policy, you can apply multiple attack groups and selectively enable the logging of detected attacks for just some of them.

By default, logging is enabled. You might want to disable logging for attacks that are lower priority for you and about which you do not give much attention. Disabling logging for such attacks helps prevent the event log from becoming cluttered with entries that you do not plan to look at anyway.

Example: Disabling Logging per Attack Group

In this example, you reference the following five attack groups in a policy and enable logging only for the first two:

- HIGH:IMAP:ANOM
- HIGH:IMAP:SIGS
- MEDIUM:IMAP:ANOM
- LOW:IMAP:ANOM
- INFO:IMAP:ANOM

The policy applies to IMAP traffic from all hosts in the Trust zone to a mail server named "mail1" in the DMZ. If any of the predefined IMAP attack objects in the above five groups match an attack, the security device closes the connection. However, it only creates log entries for detected attacks matching attack objects in the first two groups.

WebUI

1. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: mail1
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.10/32
   Zone: DMZ

2. **Policy**

   Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), mail1
   Service: IMAP
   Action: Permit

   > Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

   Group: HIGH:IMAP:ANOM
   Action: Close
   Log: (select)
   Group: HIGH:IMAP:SIGS
   Action: Close
   Log: (select)
   Group: MEDIUM:IMAP:ANOM
   Action: Close
   Log: (clear)
   Group: LOW:IMAP:ANOM
   Action: Close
   Log: (clear)
   Group: INFO:IMAP:ANOM
   Action: Close
   Log: (clear)

CLI

1. **Address**

   ```
   set address dmz mail1 1.2.2.10/32
   ```

2. **Policy**

   ```
   device-> set policy id 1 from trust to dmz any mail1 imap permit attack
   HIGH:IMAP:ANOM action close
   device-> set policy id 1
   ```

```
device(policy:1)-> set attack HIGH:IMAP:SIGS action close
device(policy:1)-> set attack MEDIUM:IMAP:ANOM action close
device(policy:1)-> unset attack MEDIUM:IMAP:ANOM logging
device(policy:1)-> set attack LOW:IMAP:ANOM action close
device(policy:1)-> unset attack LOW:IMAP:ANOM logging
device(policy:1)-> set attack INFO:IMAP:ANOM action close
device(policy:1)-> unset attack INFO:IMAP:ANOM logging
device(policy:1)-> exit
device-> save
```

Mapping Custom Services to Applications

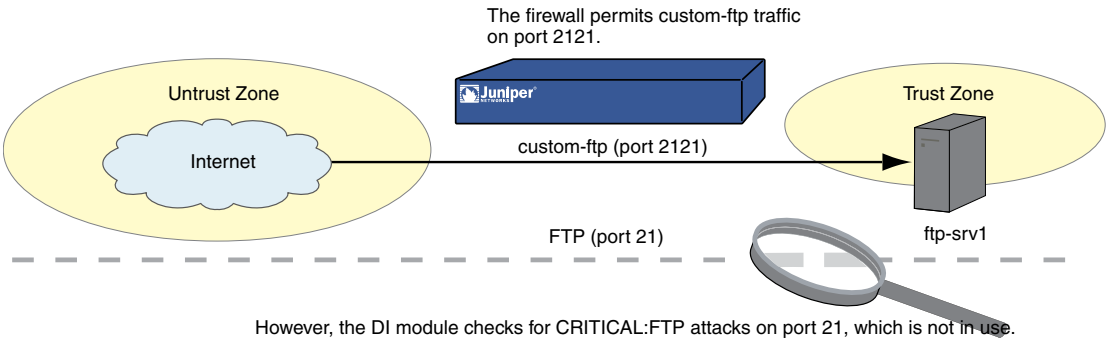
When using a custom service in a policy with a deep inspection (DI) component, you must explicitly specify the application that is running on that service so that the DI module can function properly. For example, if you create a custom service for FTP running on a nonstandard port number such as 2121 (see Figure 149 on page 595), you can reference that custom service in a policy as follows:

```
set service custom-ftp protocol tcp src-port 0-65535 dst-port 2121-2121
set policy id 1 from untrust to trust any ftp-srv1 custom-ftp permit
```

However, if you add a DI component to a policy that references a custom service, the DI module cannot recognize the application because it is using a nonstandard port number.

```
set policy id 1 from untrust to trust any ftp-srv1 custom-ftp permit attack
CRITICAL:FTP:SIGS action close-server
```

Figure 149: Mapping Custom Service



To avoid this problem, you must inform the DI module that the FTP application is running on port 2121 (see Figure 150 on page 596). Essentially, you must map the protocol in the Application Layer to a specific port number in the Transport Layer. You can do this binding at the policy level:

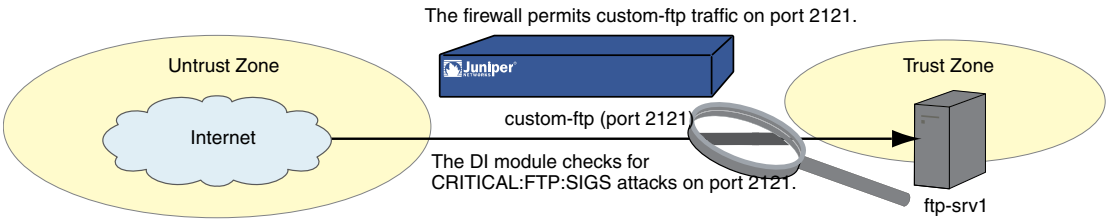
```
set policy id 1 application ftp
```

When you map the FTP application to the custom service "custom-ftp" and configure DI to examine FTP traffic for the attacks defined in the CRITICAL:FTP:SIGS attack

object group in a policy that references custom-ftp, the DI module perform its inspection on port 2121.

```
set policy id 1 from untrust to trust any ftp-srv1 custom-ftp permit attack
CRITICAL:FTP:SIGS action close-server
set policy id 1 application ftp
```

Figure 150: Mapping Custom Service to Attack Object Group



Example: Mapping an Application to a Custom Service

In this example, you define a custom service named "HTTP1" that uses destination port 8080. You map the HTTP application to the custom service for a policy permitting HTTP1 traffic from any address in the Untrust zone to a Web server named "server1" in the DMZ zone. You then apply deep inspection (DI) to the permitted HTTP traffic running on port 8080. The DI settings for this policy are as follows:

- Attack Object Groups:
  - CRITICAL:HTTP:ANOM, CRITICAL:HTTP:SIGS
  - HIGH:HTTP:ANOM, HIGH:HTTP:SIGS
  - MEDIUM:HTTP:ANOM, MEDIUM:HTTP:SIGS
- Action for all attack object groups: Close Server
- Logging: Enabled (default setting)

WebUI

1. **Custom Service**

   Policy > Policy Elements > Services > Custom > New: Enter the following, then click **OK**:

   ```
   Service Name: HTTP1
   Transport Protocol: TCP (select)
   Source Port Low: 0
   Source Port High: 65535
   Destination Port Low: 8080
   Destination Port High: 8080
   ```

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: server1
IP Address/Domain Name:
    IP Address/Netmask: 1.2.2.5/32
Zone: DMZ

3. **Policy**

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), server1
Service: HTTP1
Application: HTTP
Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Group: CRITICAL:HTTP:ANOM
Action: Close Server
Log: (select)

Group: CRITICAL:HTTP:SIGS
Action: Close Server
Log: (select)

Group: HIGH:HTTP:ANOM
Action: Close Server
Log: (select)

Group: HIGH:HTTP:SIGS
Action: Close Server
Log: (select)

Group: MEDIUM:HTTP:ANOM
Action: Close Server
Log: (select)

Group: MEDIUM:HTTP:SIGS
Action: Close Server
Log: (select)

CLI

1. **Custom Service**

set service HTTP1 protocol tcp src-port 0-65535 dst-port 8080-8080

2. **Address**

set address dmz server1 1.2.2.5/32

3. **Policy**

```
device-> set policy id 1 from untrust to dmz any server1 HTTP1 permit attack
CRITICAL:HTTP:ANOM action close-server
device-> set policy id 1
device(policy:1)-> set attack CRITICAL:HTTP:SIGS action close-server
device(policy:1)-> set attack HIGH:HTTP:ANOM action close-server
device(policy:1)-> set attack HIGH:HTTP:SIGS action close-server
device(policy:1)-> set attack MEDIUM:HTTP:ANOM action close-server
device(policy:1)-> set attack MEDIUM:HTTP:SIGS action close-server
device(policy:1)-> exit
device-> set policy id 1 application http
save
```

Example: Application-to-Service Mapping for HTTP Attacks

Some known HTTP attacks use TCP port 8000. At the time of this writing, there are currently two such attacks in the deep inspection (DI) attack object database:

- 3656, App: HP Web JetAdmin Framework Infoleak

  DOS:NETDEV:WEBJET-FW-INFOLEAK (in the attack object group MEDIUM:DOS:SIGS)

- 3638, App: HP Web JetAdmin WriteToFile Vulnerability,

  DOS:NETDEV:WEBJET-WRITETOFILE (in the attack object group CRITICAL:HTTP:SIGS)

However, by default, ScreenOS considers only TCP traffic on port 80 to be HTTP. Therefore, if the security device receives TCP traffic using port 8000, it does not recognize it as HTTP. Consequently the DI engine does not scan such HTTP traffic for these attacks and cannot detect them if they occur—unless you map HTTP as an application to a custom service using port 8000.

In this example, you associate traffic using the nonstandard port of 8000 with HTTP to detect the above attacks.

NOTE: In general, if you are running some services using nonstandard port numbers in your network and you want the DI engine to scan that traffic, you must associate the nonstandard port number to the service.

WebUI

1. **Custom Service**

   Policy > Policy Elements > Services > Custom > New: Enter the following, then click **OK**:

   Service Name: HTTP2
   Transport Protocol: TCP (select)

Source Port Low: 0
Source Port High: 65535
Destination Port Low: 8000
Destination Port High: 8000

2. **Policy**

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: HTTP2
Application: HTTP
Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Group: CRITICAL:HTTP:SIGS
Action: Close
Log: (select)

Group: MEDIUM:DOS:SIGS
Action: Close
Log: (select)

CLI

1. **Custom Service**

set service HTTP2 protocol tcp *src-port 0-65535* dst-port 8000-8000

2. **Policy**

device-> set policy id 1 from untrust to dmz any any HTTP2 permit attack
CRITICAL:HTTP:SIGS action close
device-> set policy id 1
device(policy:1)-> set attack MEDIUM:DOS:SIGS action close
device(policy:1)-> exit
device-> set policy id 1 application http
save

Customized Attack Objects and Groups

You can define new attack objects and object groups to customize the deep inspection (DI) application to best meet your needs. User-defined attack objects can be stateful signatures or—on the NetScreen-5000—TCP stream signatures. You can also adjust various parameters to modify predefined protocol anomaly attack objects.

User-Defined Stateful Signature Attack Objects

You can create a stateful signature attack object for FTP, HTTP, and SMTP. (For a complete list of supported protocols, see “Contexts for User-Defined Signatures” on page 2263.) When creating an attack object, you perform the following steps:

- Name the attack object. (All user-defined attack objects must begin with “CS:”.)
- Set the context for the DI search. (For a complete list of all the contexts that you can use when creating attack objects, see “Contexts for User-Defined Signatures” on page 2263.)
- Define the signature. (“Regular Expressions” on page 600 examines the regular expressions that you can use when defining signatures.)
- Assign the attack object a severity level. (For information about severity levels, see “Changing Severity Levels” on page 580.)

You must then put a user-defined attack object in a user-defined attack object group for use in policies.

NOTE: A user-defined attack object group can only contain user-defined attack objects. You cannot mix predefined and user-defined attack objects in the same attack object group.

Regular Expressions

When entering the text string for a signature, you can enter an alphanumeric string of ordinary characters to search for an exact character-to-character match, or you can use regular expressions to broaden the search for possible matches to sets of characters. ScreenOS supports the following regular expressions as shown in Table 64 on page 600.

Table 64: ScreenOS Supported Regular Expressions

Purpose	Meta characters	Example	Meaning
Direct binary match (octal)	\Ooctal_number	\0162 Matches: 162	Exactly match this octal number: 162
Direct binary match (hexadecimal)	\Xhexadecimal_number\X	\X01 A5 00 00\X Matches: 01 A5 00 00	Exactly match these four hexadecimal numbers: 01 A5 00 00

Table 64: ScreenOS Supported Regular Expressions *(continued)*

Purpose	Meta characters	Example	Meaning
Case-insensitive matches	\[characters\]	\[cat\] Matches: ■ Cat, cAt, caT ■ CAt, CaT, CAT ■ cat, cAt	Match the characters in cat regardless of the case of each character
Match any character	.	c . t Matches: ■ cat, cbt, cct, ... czt ■ cAt, cBt, cCt, ... cZt ■ c1t, c2t, c3t, ... c9t	Match c-any character-t
Match the previous character zero or more times, instead of only once	*	a*b + c Matches: ■ bc ■ bbc ■ abc ■ aaabbbbc	Match zero, one, or multiple occurrences of a, followed by one or more occurrences of b, followed by one occurrence of c
Match the previous character one or more times	+	a + b + c Matches: ■ abc ■ aabc ■ aaabbbbc	Match one or more occurrences of a, followed by one or more occurrences of b, followed by one occurrence of c
Match the previous character zero times or one time	?	drop-?packet Matches: ■ drop-packet ■ droppacket	Match either drop-packet or droppacket
Group expressions	( )		
Either the previous or the following character—typically used with ( )	|	(drop | packet) Matches: ■ drop ■ packet	Match either drop or packet
Character range	[start-end]	[c-f]a(d | t) Matches: ■ cad, cat ■ dad, dat ■ ead, eat ■ fad, fat	Match everything that begins with c, d, e, or f and that has the middle letter a and the last letter d or t

Table 64: ScreenOS Supported Regular Expressions *(continued)*

Purpose	Meta characters	Example	Meaning
Negation of the following character	[^character]	[^0-9A-Z] Matches: a, b, c, d, e, ... z	Match lowercase letters

NOTE: Octal is a base-8 number system that uses only the digits 0 through 7. Hexadecimal is a base-16 number system that uses the digits 0 through 9 as usual and then the letters A through F representing hexadecimal digits with decimal values of 10 through 15.

Example: User-Defined Stateful Signature Attack Objects

In this example, you have an FTP server, a Web server, and a mail server in the DMZ zone. You define the following attack objects for the use-defined signature objects as shown in Table 65 on page 602.

Table 65: User-Defined Stateful Signature Attack Objects

Object Name	Usage
cs:ftp-stor	Block someone from putting files on an FTP server
cs:ftp-user-dm	Deny FTP access to the user with the login name dmartin
cs:url-index	Block HTTP packets with a defined URL in any HTTP request
cs:spammer	Block email from any email address at “spam.com”

You then organize them into a user-defined attack object group named “DMZ DI”, which you reference in a policy permitting traffic from the Untrust zone to the servers in the DMZ zone.

WebUI

1. **Attack Object 1: ftp-stor**

   Security > Deep Inspection > Attacks > Custom > New: Enter the following, then click **OK**:

   Attack Name: CS:ftp-stor
   Attack Context: FTP Command
   Attack Severity: Medium
   Attack Pattern: STOR

2. **Attack Object 2: ftp-user-dm**

Security > Deep Inspection > Attacks > Custom > New: Enter the following, then click **OK**:

Attack Name: CS:ftp-user-dm
Attack Context: FTP User Name
Attack Severity: Low
Attack Pattern: dmartin

3. **Attack Object 3: url-index**

   Security > Deep Inspection > Attacks > Custom > New: Enter the following, then click **OK**:

   Attack Name: CS:url-index
   Attack Context: HTTP URL Parsed
   Attack Severity: High
   Attack Pattern: .*index.html.*

4. **Attack Object 4: spammer**

   Security > Deep Inspection > Attacks > Custom > New: Enter the following, then click **OK**:

   Attack Name: CS:spammer
   Attack Context: SMTP Mail From
   Attack Severity: Info
   Attack Pattern: .*@spam.com

5. **Attack Object Group**

   Security > Deep Inspection > Attacks > Custom Groups > New: Enter the following group name, move the following custom attack objects, then click **OK**:

   Group Name: CS:DMZ DI

   Select **cs:ftp-stor** and use the < < button to move the address from the Selected Members column to the Selected Members column.

   Select **cs:ftp-user-dm** and use the < < button to move the address from the Available Members column to the Selected Members column.

   Select **cs:url-index** and use the < < button to move the address from the Available Members column to the Selected Members column.

   Select **cs:spammer** and use the < < button to move the address from the Available Members column to the Selected Members column.

6. **Policy**

   Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: HTTP

> Click **Multiple**, select **FTP**, then click **OK** to return to the basic policy configuration page.

Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Group: CS:DMZ DI
Action: Close Server
Log: (select)

CLI

1. **Attack Object 1: ftp-stor**

```
set attack CS:ftp-stor ftp-command STOR severity medium
```

2. **Attack Object 2: ftp-user-dm**

```
set attack CS:ftp-user-dm ftp-username dmartin severity low
```

3. **Attack Object 3: url-index**

```
set attack CS:url-index http-url-parsed index.html severity high
```

4. **Attack Object 4: spammer**

```
set attack CS:spammer smtp-from .*@spam.com severity info
```

5. **Attack Object Group**

```
set attack group "CS:DMZ DI"
set attack group "CS:DMZ DI" add CS:ftp-stor
set attack group "CS:DMZ DI" add CS:ftp-user-dm
set attack group "CS:DMZ DI" add CS:url-index
set attack group "CS:DMZ DI" add CS:spammer
```

6. **Policy**

```
set policy id 1 from untrust to dmz any any http permit attack "CS:DMZ DI" action
 close-server
set policy id 1
device(policy:1)-> set service ftp
device(policy:1)-> exit
save
```

TCP Stream Signature Attack Objects

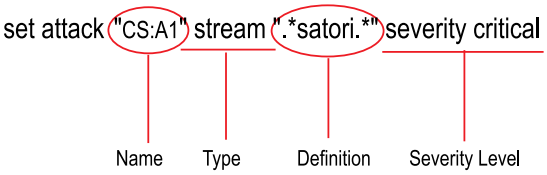
The stateful signatures are context-based within specific applications, such as an FTP username or an SMTP header field. TCP stream signatures look for patterns anywhere in any kind of TCP traffic regardless of the application protocol in use.

NOTE: You can define TCP stream signatures on NetScreen-5000 series systems only.

Because there are no predefined TCP stream signature attack objects, you must define them. When creating a signature attack object, you define the following components:

- Attack object name (All user-defined attack objects must begin with “CS:”.)
- Object type (“stream”)
- Pattern definition
- Severity level

Figure 151: Example of a TCP Stream Signature Attack Object



Example: User-Defined Stream Signature Attack Object

In this example, you define a stream signature object “.*satori.*”. You name it “CS:A1” and define its severity level as critical. Because a policy can reference only attack object groups, you create a group named “CS:Gr1”, and then add this object to it. Finally, you define a policy that references CS:Gr1 and that instructs the security device to sever the connection and send TCP RST to the client if the pattern appears in any traffic to which the policy applies.

WebUI

1. **Stream Signature Attack Object**

   Security > Deep Inspection > Attacks > Custom > New: Enter the following, then click **OK**:

   Attack Name: CS:A1
   Attack Context: Stream
   Attack Severity: Critical
   Attack Pattern: .*satori.*

2. **Stream Signature Attack Object Group**

   Security > Deep Inspection > Attacks > Custom Groups > New: Enter the following, then click **OK**:

   Group Name: CS:Gr1

Select **CS:A1** in the Available Members column and then click **< <** to move it to the Selected Members column.

3. **Policy**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: ANY
   Action: Permit

   > Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group; and then click **OK** to return to the basic policy configuration page:

   Group: CS:Gr1
   Action: Close Client
   Log: (select)

CLI

1. **Stream Signature Attack Object**

   ```
   set attack "CS:A1" stream ".*satori.*" severity critical
   ```

2. **Stream Signature Attack Group**

   ```
   set attack group "CS:Gr1"
   set attack group "CS:Gr1" add "CS:A1"
   ```

3. **Policy**

   ```
   set policy from trust to untrust any any any permit attack CS:Gr1 action
   close-client
   save
   ```

Configurable Protocol Anomaly Parameters

You can modify certain parameters of a protocol anomaly attack object. Although Juniper defines protocol anomaly attack objects to find deviations from protocol standards defined in RFCs and common RFC extensions, not all implementations adhere to these standards. If you find that the application of a certain protocol anomaly attack object is producing numerous false positives, you can modify its parameters to better match the accepted use of that protocol in your network.

NOTE: For a complete list of all configurable parameters, see the **di** command in *ScreenOS CLI Reference Guide: IPv4 Command Descriptions*.

Example: Modifying Parameters

In this example, you set higher values for the following parameters to reduce the number of false positives that occurred with the default settings:

Protocol Parameter	Default	New
SMB—Maximum number of login failures per minute	8 failures	10 failures
Gnutella—Maximum number of time-to-live (TTL) hops	8 hops	10 hops

For the following parameters, you set lower values to detect anomalous behavior that the security device missed with the default settings:

Protocol Parameter	Default	New
AOL Instant Messenger (AIM)—Maximum OSCAR File Transfer (OFT) filename length. OSCAR = Open System for Communication in Real-time, the protocol that AIM clients use.	10,000 bytes	5,000 bytes
AOL Instant Messenger—Maximum length of a FLAP frame (FLAP header, which is always 6 bytes, plus data). OSCAR makes use of a the FLAP protocol to make connections and open channels between AIM clients.	10,000 bytes	5,000 bytes

WebUI

Security > Deep Inspection > Service Limits: Enter the following, then click **Apply**:

Service: AIM (select)
Maximum Bytes in FLAP Length: 5000
Maximum Bytes in OFT Length: 5000

Service: GNUTELLA (select)
Maximum TTL Hops: 10

Service: SMB (select)
Maximum Number of Login Failures per Minute: 10

CLI

```
set di service smb failed_logins 10
set di service gnutella max_ttl_hops 10
set di service aim max_flap_length 5000
set di service aim max_oft_frame 5000
```

```
save
```

Negation

Typically, you use attack objects to match patterns that are indicative of malicious or anomalous activity. However, you can also use them to match patterns indicative of benign or legitimate activity. WIth this approach, something is amiss only if a type of traffic does not match a particular pattern. To use attack objects in this way, you apply the concept of negation.

A useful application of attack object negation would be to block all login attempts other than those with the correct username and password. It would be difficult to define all invalid usernames and passwords, but quite easy to define the correct ones and then apply negation to reverse what the security device considers an attack; that is, everything except the specified attack object.

Example: Attack Object Negation

In this example (see Figure 152 on page 609), you define two attack objects: one specifying the correct username required to log into an FTP server, and another the correct password. You then apply negation to both attack objects, so that the security device blocks any login attempt to that server that uses any other username or password than those defined in the attack objects.

The example uses the following settings:

- The correct username and password are admin1 and pass1.
- The FTP server is at 1.2.2.5 in the DMZ zone. Its address name is ftp1.
- You apply DI on FTP traffic to the server from all hosts in the Untrust and Trust zones.
- All security zones are in the trust-vr routing domain.

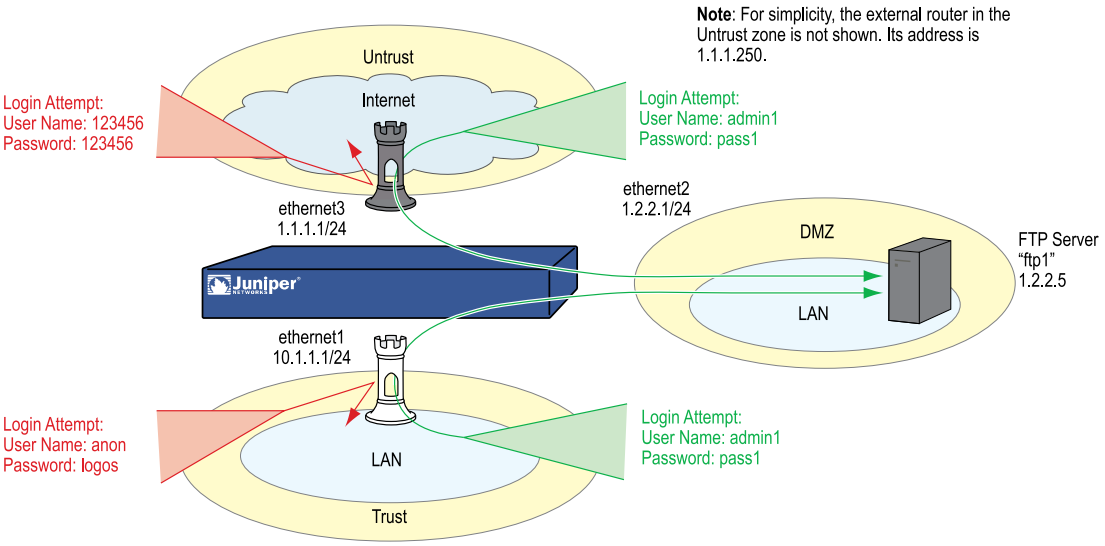
You create the following two attack objects:

- Attack Object #1:
  - Name: CS:FTP1_USR_OK
  - Negation: enabled
  - Context: ftp-username
  - Pattern: admin1
  - Severity: high
- Attack Object #2:
  - Name: CS:FTP1_PASS_OK
  - Negation: enabled
  - Context: ftp-password

- Pattern: pass1
- Severity: high

You then put both objects into an attack object group named CS:FTP1_LOGIN and reference that attack object group in two policies permitting FTP traffic from the Trust and Untrust zones to ftp1 in the DMZ.

Figure 152: Attack Object Negation



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT (select)

NOTE: By default, any interface that you bind to the Trust zone is in NAT mode. Consequently, this option is already enabled for interfaces bound to the Trust zone.

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: DMZ
Static IP: (select this option when present)

IP Address/Netmask: 1.2.2.1/24

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: ftp1
IP Address/Domain Name:
IP/Netmask: (select), 1.2.2.5/32
Zone: DMZ

3. **Attack Object 1: CS:FTP1_USR_OK**

Security > Deep Inspection > Attacks > Custom > New: Enter the following, then click **OK**:

Attack Name: CS:FTP1_USR_OK
Attack Context: FTP Username
Attack Severity: High
Attack Pattern: admin1
Pattern Negation: (select)

4. **Attack Object 2: CS:FTP1_PASS_OK**

Security > Deep Inspection > Attacks > Custom > New: Enter the following, then click **OK**:

Attack Name: CS:FTP1_PASS_OK
Attack Context: FTP Password
Attack Severity: High
Attack Pattern: pass1
Pattern Negation: (select)

5. **Attack Object Group**

Security > Deep Inspection > Attacks > Custom Groups > New: Enter the following group name, move the following custom attack objects, then click **OK**:

Group Name: CS:FTP1_LOGIN

Select **CS:FTP1_USR_OK** and use the < < button to move the address from the Available Members column to the Selected Members column.

Select **CS:FTP1_PASS_OK** and use the < < button to move the address from the Available Members column to the Selected Members column.

6. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **ok:**

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: (select) 1.1.1.250

7. **Policies**

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), ftp1
Service: FTP
Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group; and then click **OK** to return to the basic policy configuration page:

Group: CS:FTP1_LOGIN
Action: Drop
Log: (select)

Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), ftp1
Service: FTP
Action: Permit

> Click **Deep Inspection**, enter the following, click **Add** to enter each attack object group, then click **OK** to return to the basic policy configuration page:

Group: CS:FTP1_LOGIN
Action: Drop
Log: (select)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet2 zone dmz
set interface ethernet2 ip 1.2.2.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Address**

```
set address dmz ftp1 1.2.2.5/32
```

3. **Attack Objects**

```
set attack CS:FTP1_USR_OK ftp-username not admin1 severity high
set attack CS:FTP1_PASS_OK ftp-password not pass1 severity high
set attack group CS:FTP1_LOGIN
set attack group CS:FTP1_LOGIN add CS:FTP1_USR_OK
set attack group CS:FTP1_LOGIN add CS:FTP1_PASS_OK
```

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

5. **Policies**

```
set policy from untrust to dmz any ftp1 ftp permit attack CS:FTP1_LOGIN action
 drop
set policy from trust to dmz any ftp1 ftp permit attack CS:FTP1_LOGIN action
drop
save
```

Granular Blocking of HTTP Components

A Juniper Networks security device can selectively block ActiveX controls, Java applets, .zip files, and .exe files sent via HTTP. The danger that these components pose to the security of a network is that they provide a means for an untrusted party to load and then control an application on hosts in a protected network.

When you enable the blocking of one or more of these components in a security zone, the security device examines every HTTP header that arrives at an interface bound to that zone. It checks if the content type listed in the header indicates that any of the targeted components are in the packet payload. If the content type is Java, .exe, or .zip and you have configured the security device to block those HTTP component types, the device blocks the packet. If the content type lists only "octet stream" instead of a specific component type, then the device examines the file type in the payload. If the file type is Java, .exe, or .zip and you have configured the device to block those component types, the device blocks the packet.

When you enable the blocking of ActiveX controls, the device blocks all HTTP packets containing any type of HTTP component in its payload—ActiveX controls, Java applets, .exe files, or .zip files.

NOTE: When ActiveX-blocking is enabled, the security device blocks Java applets, .exe files, and .zip files whether or not they are contained within an ActiveX control.

ActiveX Controls

Microsoft ActiveX technology provides a tool for web designers to create dynamic and interactive web pages. ActiveX controls are components that allow different programs to interact with each other. For example, ActiveX allows your browser to

open a spreadsheet or display your personal account from a backend database. ActiveX components might also contain other components such as Java applets, or files such as .exe and .zip files.

When you visit an ActiveX-enabled website, the site prompts you to download ActiveX controls to your computer. Microsoft provides a pop-up message displaying the name of the company or programmer who authenticated the ActiveX code that is offered for download. If you trust the source of the code, you can proceed to download the controls. If you distrust the source, you can refuse them.

If you download an ActiveX control to your computer, it can then do whatever its creator designed it to do. If it is malicious code, it can now reformat your hard drive, delete all your files, send all your personal email to your boss, and so on.

Java Applets

Serving a similar purpose as ActiveX, Java applets also increase the functionality of web pages by allowing them to interact with other programs. You download Java applets to a Java Virtual Machine (VM) on your computer. In the initial version of Java, the VM did not allow the applets to interact with other resources on your computer. Starting with Java 1.1, some of these restrictions were relaxed to provide greater functionality. As a result, Java applets can now access local resources outside the VM. Because an attacker can program Java applets to operate outside the VM, they pose the same security threat as ActiveX controls do.

EXE Files

If you download and run an executable file (that is, a file with a .exe extension) obtained off the Web, you cannot guarantee that the file is uncontaminated. Even if you trust the site from which you downloaded it, it is possible that somebody sniffing download requests from that site has intercepted your request and responded with a doctored .exe file that contains malicious code.

ZIP Files

A zip file (that is, a file with a .zip extension) is a type of file containing one or more compressed files. The danger of downloading a .exe file presented in the previous section about .exe files applies to .zip files, because a .zip file can contain one or more .exe files.

Example: Blocking Java Applets and .exe Files

In this example, you block any HTTP traffic containing Java applets and .exe files in packets arriving at an Untrust zone interface.

WebUI

Screening > Screen (Zone: Untrust): Select **Block Java Component** and **Block EXE Component**, then click **Apply**.

CLI

```
set zone untrust screen component-block jar
set zone untrust screen component-block exe
save
```

Chapter 17

Intrusion Detection and Prevention

An Intrusion Prevention System (IPS), more commonly known as a *firewall*, is used to detect and prevent attacks in network traffic. While firewalls provide perimeter and boundary protection, allowed traffic can hide attacks that firewalls are not designed to detect.

Juniper Networks Intrusion Detection and Prevention (IDP) technology can detect and then stop attacks when deployed inline to your network. Unlike an IPS alone, IDP uses multiple methods to detect attacks against your network and prevent attackers from gaining access and doing damage. IDP can drop malicious packets or connections before the attacks can enter your network. It is designed to reduce false positives and ensure that only actual malicious traffic is detected and stopped. You can also deploy IDP as a passive sniffer, similar to a traditional IPS but with greater accuracy and manageability.

This chapter contains the following sections:

IDP-Capable Security Devices

ScreenOS supports IDP capabilities on some security devices. The security module (SM), an optional component installed on these devices, provides IDP functionality. For more information about how the security modules detects malicious packets, see “Traffic Flow in an IDP-Capable Device” on page 616.

If you purchased a security device with only firewall or virtual private network (VPN) capabilities, you can upgrade the device to an IDP-capable system by performing the following steps:

- Installing the Advanced and IDP license keys
- Upgrading the boot loader
- Installing an IDP-capable version of ScreenOS
- Upgrading the system memory
- Installing security module(s)

NOTE: Installing the IDP license key disables the ScreenOS deep inspection (DI) feature.

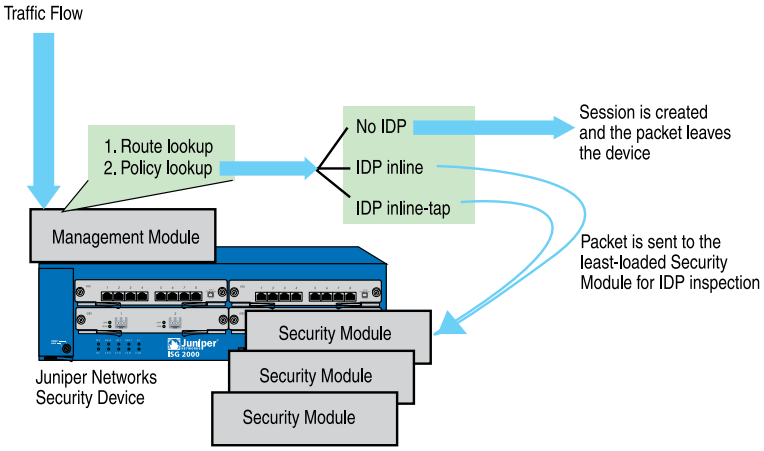
Refer to the *ISG 2000 Field Upgrade Guide* and the *NetScreen-ISG 1000 Field Upgrade Guide* for instructions on how to upgrade the devices to include IDP capabilities.

You can use the IDP-capable security device as a fully integrated firewall/VPN/IDP security system that not only screens traffic between the Internet and your private network but also provides application-level security. You can also use this device as a standalone IDP system to protect critical segments of your private network. For more information, see “Configuring the Device as a Standalone IDP Device” on page 683.

Traffic Flow in an IDP-Capable Device

This section describes the packet flow on the IDP-capable security device. The ASIC processor in the device receives the packet, checks the session table and if it doesn’t find a match, forwards the packet to the management module. Figure 153 on page 617 illustrates the packet flow from the management module to the security module. The management module checks the packet against firewall policies. If IDP is enabled, traffic is redirected to the least-loaded security module, which performs IDP analysis on the packet.

Figure 153: Traffic Flow in the Security Device



Each security module, consisting of a dual processor, maintains its own session table. The dual CPU allows each security module to run two instances of IDP per module. The **get sm status** command shows the dual CPU for each security module

```
device->get sm status
SM CPU aval ena Sess_cnt
1  1    1      10    \
2  1    1      8     / Security module 1
3  0    1      0     \
4  0    1      0     / Security module 2
5  0    1      0     \
6  0    1      0     / Security module 3
```

You can view details of sessions that are running on a particular slot in the security device by using the get session sm-slot *slot-id* command. Similarly, you can view details of sessions that are running on a specific CPU by using the get session sm-cpu *cpu-id* command. The sm-slot and sm-cpu options are supported only for ISG1000/2000 devices with a security module.

The inline-tap mode configuration enables the IDP security modules to monitor traffic passively. In inline-tap mode, an internal tap sends a copy of every packet in the traffic flow to the security module for IDP processing; at the same time, the ASIC module performs firewall/VPN processing on the inline traffic.

If an attack object is detected, TCP reset occurs with “close” option to clear the session table. For each attack that matches a rule, you can choose to ignore, drop, or close the current attacking packets or connection. For more information about the action to perform when an attack object is found, see “Defining Actions” on page 644.

NOTE: The security module inspects traffic within IPsec tunnels that use Null encryption method. It does not inspect the traffic that uses any other encryption. For more information about configuring this feature in the security module using NSM, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

When the PPU passes the IPsec ESP tunneled packet that uses Null-encryption method to the IDP security device, the IDP security device checks if ESP-NULL encrypted packet inspection feature is enabled. If the feature is enabled, IDP performs packet sanity test and ESP-NULL packet check on the ESP-NULL packet. If the packet does not conform to either of the tests, the security device drops the packet. If the packet passes both the tests, IDP decapsulates the payload data of the original IP ESP-NULL packet. It then checks if a session already exists for the decapsulated data. If a session does not exist, it creates a session for the decapsulated packet based on the decapsulated packet data header. The IDP device then passes the packet to the IDP engine for inspection.

IDP limits the total number of internal sessions that is generated for the ESP-NULL encrypted traffic. By default, the maximum value is set to 20000 sessions and the user can define a value up to 40000 sessions. If the maximum session limit is reached the IDP bypasses the packet without inspection and will generate an event log.

NOTE: IDP supports inspection of both transparent mode and tunnel mode ESP-NULL encrypted traffic. Transport mode is not currently supported.

For tunnel mode ESP, IDP uses the new IP packet header's source IP address, destination IP address, internal payload data's protocol id, source port and destination port for policy look up.

IDP follows the normal packet processing on the decapsulated ESP-NULL packets. But when an attack happens, IDP drops the corresponding internal session and does not forward it to the outer session. The following table explains the actions taken by the ESP-NULL encrypted traffic against the actions taken by the normal traffic for the rule actions defined by IDP.

Table 66: IDP Actions for ESP-NULL Traffic

Rule Action	Normal Traffic	ESP-NULL Traffic (Inner Session)	ESP-NULL Traffic (Outer session)
NONE	NONE	NONE	NONE
RECOMMENDED	Attack action	Attack action	NONE
IGNORE	IGNORE	IGNORE	IGNORE
MARK_DIFFSERV	MARK_DIFFSERV	NONE	MARK_DIFFSERV
DROP_PACKET	DROP_PACKET	DROP_PACKET	DROP_PACKET
DROP	drop packet /teject flow	drop packet /reject flow	drop packet
CLOSE_CLIENT	reset client	drop packet /reject flow	none
CLOSE_SERVER	reset server	drop packet /reject flow	none
CLOSE	reset	drop packet /reject flow	none

When attacks happen on the ESP-NULL traffic, appropriate actions are taken on the internal sessions based on the rule actions as explained in Table 16. When there are no attack objects, the traffic passes successfully through IDP.

Users can enable the IDP security device to inspect multicast traffic by using the set flow multicast idp command.

NOTE: For multicast traffic inspection, all the outgoing interfaces should belong to the same zone.

To disable the multicast traffic inspection feature, use the unset flow multicast idp command.

Configuring Intrusion Detection and Prevention

This section presents three basic examples for configuring IDP on your security device:

- "Example 1: Basic IDP Configuration" on page 620
- "Example 2: Configuring IDP for Active/Passive Failover" on page 622
- "Example 3: Configuring IDP for Active/Active Failover" on page 624

Preconfiguration Tasks

Before you start configuring IDP on the device, you need to ensure the following:

- Your security device is IDP-capable. For more information, see "IDP-Capable Security Devices" on page 615.
- You have installed and configured a Network and Security Manager (NSM) system on a management station.

NOTE: Although you can perform basic device configuration using the ScreenOS CLI or WebUI, you need NSM to configure and manage IDP on the security device.

NSM provides integrated policy management, where each security device is linked to one security policy that contains rules defining the types of traffic permitted on the network and the way that traffic is treated inside the network.

- You have a security policy for the device. You can use the default security policy provided in NSM, or you can create a custom security policy for the firewall/VPN functions on the device.

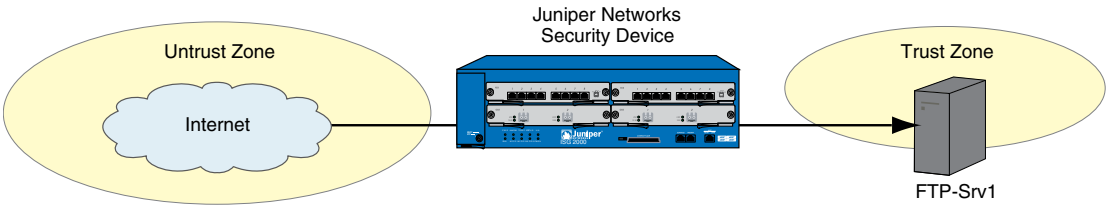
NOTE: You cannot use the DSCP marking feature if you have enabled IDP on the security device. The DSCP marking feature is disabled in IDP-capable security devices.

Example 1: Basic IDP Configuration

In this example, a Juniper Networks device is deployed with firewall/VPN/IDP functionality. Before you start configuring, make sure your device is IDP-capable as described in "IDP-Capable Security Devices" on page 615. Set up the device as shown in Figure 154 on page 620, then do the following:

1. Physically connect the network components.
2. Add the network components that you want IDP to protect using the CLI, WebUI, or NSM UI.

Figure 154: Setting Up the Device for Basic IDP



These components can be routers, servers, workstations, subnetworks, or any other objects connected to your network. In NSM, these network components are represented as *address objects*. You can also create address object *groups*, which represent multiple address objects. For more information about creating address objects, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

3. Enable IDP (the default is inline mode) in the appropriate firewall rule for the device.

   This step can be performed using the CLI or NSM UI. The CLI commands are shown below (to configure using NSM, see "Enabling IDP in Firewall Rules" on page 629):

```
device-> get policy
Total regular policies 5, Default deny.
ID From     To       Src-address  Dst-address  Service  Action State  ASTLCB
9 Trust     Untrust  Any          Any          MAIL     Permit       enabled –X-X
4 blade1    dmz2     Any          Any          ANY      Permit       enabled –X-X
6 dmz2      blade1   Any          Any          ANY      Permit       enabled –X-X
10 Untrust  Trust    Any          MIP(172.24.~ HTTP     Permit       enabled –X-X
                                    HTTPS
device-> get policy id 4
name:"none" (id 4), zone blade1 -> dmz2,action Permit, status "enable
src "Any", dst "Any", serv "ANY"
Policies on this vpn tunnel: 0
nat off, Web filtering : disabled
vpn unknown vpn, policy flag 00010000, session backup: on
policy IDP mode : disable
```

```
traffic shapping off, scheduler n/a, serv flag 00
log yes, log count 0, alert no, counter no(0) byte rate(sec/min) 0/0
total octets 0, counter(session/packet/octet) 0/0/0
No Authentication
No User, User Group or Group expression set
device-> set policy id 4
device (policy:4)-> set idp
device (policy:4)-> exit
device-> get policy id 4
policy IDP mode : inline
```

4. Add the device using the NSM UI.

   To **add the device**, do the following:

   a. Select **Device Manager > Security Devices >** + (**Device**).

   b. Enter a device name, then and click **Next**.

   c. Enter the management IP address of the device, then click **Finish**.

      The new device appears in the list of security devices.

5. Validate the security policy on your device.

   Make sure you have a security policy for the device. You can use the default security policy; or, if the device is deployed as an integrated firewall/VPN/IDP device, you can create a custom security policy for the firewall/VPN functions on the device. For more information, see “Configuring Security Policies” on page 626. For more information about configuring security polices using NSM, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

6. Import the device.

   To **import the device**, right-click on the device that you added, then select **Import device**. Importing the device copies the security-policy information from the device to the NSM server so that the device can be managed. The imported policy is displayed in NSM under **Security Policy**.

   For more information about adding and configuring devices using NSM, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

   Other configuration settings include operational mode, administrative password, zone interface assignments, and default route configurations.

7. Add and configure IDP rules in the security policy for the device.

   You configure a security policy on the device to include IDP rules. When you update the configuration on the device, the entire security policy, including IDP rule additions or changes, is installed on the device. For more information about enabling and configuring IDP rules, see “Configuring IDP Rules” on page 631.

NOTE: If you are using the device as a standalone IDP system, you need to configure a simple firewall rule that directs all traffic to be checked against the IDP rules. For more information, see "Configuring the Device as a Standalone IDP Device" on page 683.

8. Assign the security policy to the security device.
9. Allow traffic to flow and view the IDP sessions with the following command:

```
device->get sm status
SM CPU aval ena Sess_cnt
1  1   1     10   \
2  1   1     8    / Security module 1
3  0   1     0    \
4  0   1     0    / Security module 2
5  0   1     0    \
6  0   1     0    / Security module 3
```

The above command shows one security module (SM1 and SM2) installed in the device. The CPU column indicates that security modules 2 and 3 are not installed in the device. The status on the two CPUs on each security module is displayed in separate rows.

The management module in the device processes the traffic and then forwards it for IDP inspection to the security modules. The traffic is load-balanced between the two CPUs in the security module (see the **Sess_Cnt** column). If your device has more than one security module, then the management module load-balances the traffic between the security modules.

NOTE: When you have more than one security module installed in the device and one module fails, then the IDP sessions are automatically transferred to the next security module.

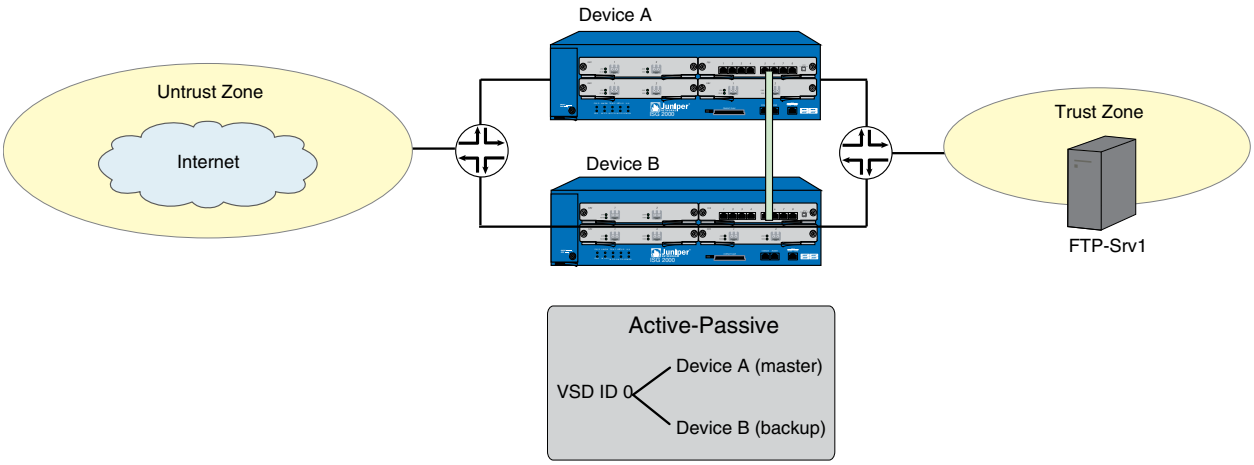
10. Periodically update the attack object database on the NSM server.

    See "Managing IDP" on page 687 for more information.

Example 2: Configuring IDP for Active/Passive Failover

In this example, set up your security device in high availability (HA) pairs to remove a potential point of failure from your network design. Figure 155 on page 623 illustrates device setup for configuring IDP for Active/Passive failover. The two devices are in an Active/Passive failover configuration; that is, the primary device is active, handling all firewall and VPN activities; and the backup device is passive, waiting to take over when the primary device fails.

Figure 155: Configuring IDP for Active/Passive Failover



Set up the device as shown in Figure 155 on page 623, then do the following:

1. Configure Device A and Device B for IDP as described in "Example 1: Basic IDP Configuration" on page 620.
2. To ensure continuous traffic flow, cable and configure two security devices in a redundant cluster with Device A acting as a primary device and Device B acting as its backup.

   Cable e1/x on Device A to e1/x on Device B. Similarly cable the e2/x interfaces. For more information about cabling the two devices together, setting up managed IP addresses to manage a backup device, or removing other potential points of failure by setting up redundant switches on either side of the HA pair, see *"High Availability" on page 1763*.
3. Configure the HA interfaces.

   Specify the zones with HA interfaces. Bind e1/x and e2/x to the HA zone. Set Manage IP addresses for the Trust zone interfaces on both devices.
4. Configure an Active/Passive NetScreen Redundancy Protocol (NSRP) cluster.

   Assign each device to NSRP cluster ID 0. When the devices become members of the NSRP cluster, the IP addresses of their physical interfaces automatically become the IP addresses of the Virtual Security Interfaces (VSIs) for Virtual Security Device (VSD) group ID 0. Each VSD member has a default priority of 100. The device with the higher unit ID becomes the VSD group's primary device. For more information about VSDs, see *"High Availability" on page 1763*.

   For example, enter the following on each of the devices to configure an NSRP cluster:

   a. Add the device to an NSRP cluster and a VSD group.

      ```
      set nsrp cluster id 0
      ```

   b. Enable automatic Run-Time Object (RTO) synchronization.

      ```
      set nsrp rto sync all
      ```

c. Select the ports that you want the devices to monitor.

If the device detects a loss of network connectivity on one of the monitored ports, then it triggers a device failover.

```
set nsrp rto-mirror sync
set nsrp monitor interface ethernet1
set nsrp monitor interface ethernet3
set nsrp cluster id 0
save
```

Upon initial NSRP configuration, the VSD group members with the priority number closest to 0 becomes the primary device. (The default is 100.) If Device A and B have the same priority value, the device with the highest MAC address becomes primary device.

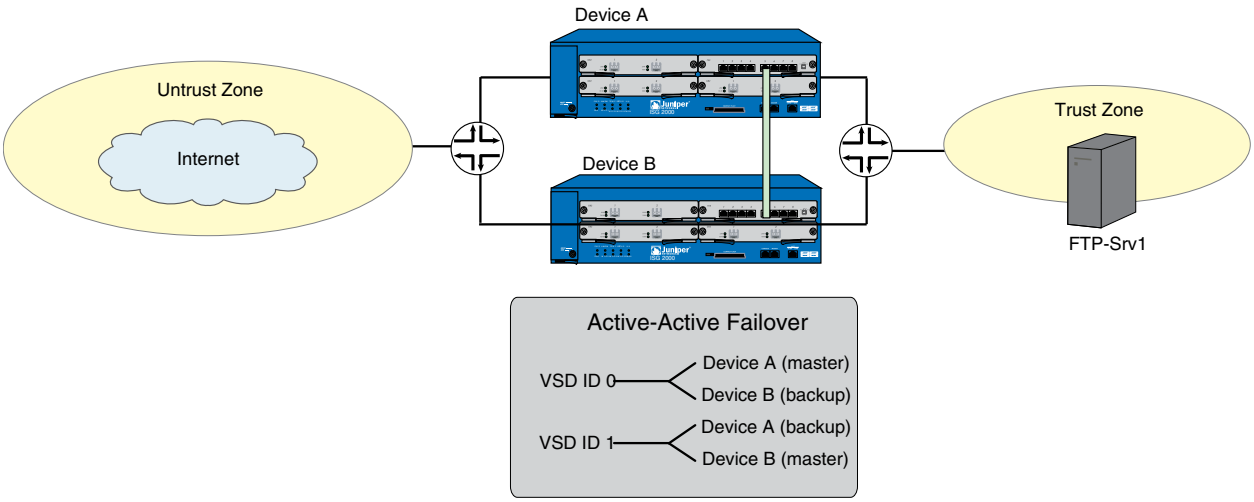
The primary device propagates all its network and configuration settings and the current session information to the backup device. Although the backup device is passive, it is maintaining its synchronization with the information it receives from the primary device. If the primary device fails, the backup device is promoted to primary and takes over the traffic processing.

NOTE: Synchronization is maintained for firewall sessions only. Stateful failover does not occur for IDP sessions.

Example 3: Configuring IDP for Active/Active Failover

In this example, set up your security devices in route or Network Address Translation (NAT) mode and configure them in a redundant cluster to be active, sharing the traffic distributed between them. This is accomplished using NSRP to create two VSD groups as shown in Figure 156 on page 625. Device A acts as the primary device in VSD group 1 and as the backup of VSD group 2. Device B acts as the primary device in VSD group 2 and as the backup of VSD group 1. No single point of failure exists in an Active/Active setup.

Figure 156: Configuring IDP for Active/Active Failover



Set up the device as shown in Figure 156 on page 625, then do the following:

NOTE: We recommend that the same number of security modules be installed on both devices.

1. Configure Device A and Device B for IDP as described in "Example 1: Basic IDP Configuration" on page 620.
2. To ensure continuous traffic flow, cable and configure two security devices in a redundant cluster.

   Cable e1/x on Device A to e1/x on Device B. Similarly cable the e2/x interfaces. For more information about how to cable the two devices together, setting up managed IP addresses to manage a backup device, or to remove other potential points of failure by setting up redundant switches on either side of the HA pair, see *"High Availability" on page 1763*.
3. Configure the HA interfaces.

   Specify the zones with HA interfaces. Bind e1/x and e2/x to the HA zone. Set Manage IP addresses for the trust zone interfaces on both devices.
4. Configure an Active/Active NSRP cluster.

   Devices A and B are members of the same NSRP cluster and VSD group 0. For Active/Active failover, create a second VSD group—group 1.

1. Assign Device A priority 1 in group 0 and the default priority (100) in group 1.
2. Assign Device B priority 1 in group 1 and the default priority (100) in group 0.

   In both VSD groups, enable the preempt option on the primary device and set the preempt hold-down time to 10 seconds. If both devices are active, Device A is always the primary device of group 1 and Device B is always the primary device of group 2.

Device A

```
set nsrp vsd-group id 0 priority 1
set nsrp vsd-group id 0 preempt hold-down 10
set nsrp vsd-group id 0 preempt
set nsrp vsd-group id 1
save
```

Device B

```
set nsrp vsd-group id 1 priority 1
set nsrp vsd-group id 1 preempt hold-down 10
set nsrp vsd-group id 1 preempt
save
```

For more information about creating two VSD groups, see "High Availability" on page 1763.

Devices A and B each receive 50 percent of the network and VPN traffic. When Device A fails, Device B becomes the primary device of VSD group 1, as well as continuing to be the primary device of VSD group 2, and handles 100 percent of the traffic.

NOTE: Synchronization is maintained for firewall sessions only. Stateful failover does not occur for IDP sessions.

Configuring Security Policies

A security policy defines how your managed devices handle network traffic. You can configure multiple security policies in NSM, but a device can only have one active security policy at a time. You can install the same security policy on multiple devices, or you can install a unique security policy on each device in your network.

About Security Policies

Each instruction in a security policy is called a *rule*. Security policies can contain multiple rules. You create rules in *rulebases*, sets of rules that combine to define a security policy. Each security policy contains the Zone and Global firewall rulebases, which cannot be deleted. You can add or delete any other rulebase—Multicast, IDP, Exempt, and Backdoor—in a security policy; however, a single policy can only contain one instance of any type of rulebase. Each security policy (all rulebases combined) can contain a maximum of 40,000 rules.

This section describes the IDP, Exempt, and Backdoor rulebases. For more information about Zone and Global firewall rulebases and the Multicast rulebase, see the information about configuring security policies in the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

NOTE: In the ScreenOS WebUI and CLI, a security policy is a single statement that defines a source, destination, zone, direction, and service. In NSM, those same statements are known as *rules*, and a security policy is a collection of rules.

Managing Security Policies

Within security policies, you can manage individual rules in each rulebase, including:

- Determining the order in which rules are applied to network traffic
- Disabling a rule
- Negating source or destination addresses (ScreenOS 5.x devices only)
- Verifying the security policy
- Merging security policies

NOTE: The IDP, Exempt, and Backdoor rulebases are not included when you merge two policies into a single policy.

For detailed information about managing your security policy, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

Installing Security Policies

After you create a security policy by building rules in one or more rulebases, you can assign, validate, and install that policy on specific managed devices. For detailed information about installing security policies, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

Using IDP Rulebases

After a firewall rule (intrazone or global) has permitted the network traffic, you can direct the device to further inspect the traffic for known attacks. NSM supports the following IDP rulebases:

- **IDP:** This rulebase protects your network from attacks by using attack objects to detect known and unknown attacks. Juniper Networks provides predefined attack objects that you can use in IDP rules. You can also configure your own custom attack objects. For more information, see "Configuring IDP Attack Objects" on page 663.

NOTE: Juniper Networks regularly updates predefined attack objects to keep current with newly discovered attacks. For more information about updating attack objects, see "Managing IDP" on page 687.

- **Exempt:** This rulebase works in conjunction with the IDP rulebase to prevent unnecessary alarms from being generated. You configure rules in this rulebase to exclude known false positives or to exclude a specific source, destination, or source/destination pair from matching an IDP rule. If traffic matches a rule in the IDP rulebase, IDP attempts to match the traffic against the Exempt rulebase before performing the action specified.
- **Backdoor Detection:** This rulebase protects your network from mechanisms installed on a host computer that facilitate unauthorized access to the system. Attackers who have already compromised a system typically install backdoors (such as Trojans) to make future attacks easier. When attackers send information to and retrieve information from a backdoor program, they generate interactive traffic that IDP can detect.

The rules in all rulebases, including the Zone, Global, and Multicast rulebases, combine to create a security policy. To direct the device to process and execute rules in the IDP rulebases, you need to enable IDP in a firewall rule. See "Enabling IDP in Firewall Rules" on page 629.

NOTE: If you import the device into NSM, the imported device configuration does not include the IDP, Exempt, or Backdoor rulebases.

Role-Based Administration of IDP Rulebases

NSM's role-based administration (RBA) allows you to create custom roles for individual administrators to give them authority to view or edit IDP rulebases. For more information about RBA, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

You can assign view or edit capabilities for a role based on a IDP rulebase. For example, an administrator who can view and edit a firewall rulebase may be able to only view IDP and Backdoor rulebases.

By default, the predefined roles System Administrator and Domain Administrator can view and edit all rulebases, and the Read-Only System Administrator and Read-Only Domain Administrator can only view rulebases. When you create a new role, the New Role dialog box allows you to specify whether an administrator can view or edit IDP or Backdoor rulebases.

Configuring Objects for IDP Rules

Objects are reusable logical entities that you can apply to rules. Each object that you create is added to a database for the object type. You can use the following types of objects:

- **Address objects** represent components of your network, such as host machines, servers, and subnets. You use address objects in security policy rules to specify the network components that you want to protect.

NOTE: You must create each object in the Address Object database. There are no default address objects.

For information about creating address objects, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

- **Service objects** represent network services that use Transport layer protocols such as TCP, UDP, RPC, and ICMP. You use service objects in rules to specify the service an attack uses to access your network. NSM includes predefined service objects, a database of service objects that are based on industry-standard services. If you need to add service objects that are not included in the predefined service objects, you can create custom service objects. For more information about creating service objects, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.
- **IDP attack objects** represent known and unknown attacks. IDP includes a predefined attack object database that is periodically updated by Juniper Networks (see "Managing IDP" on page 687). You can also add custom attack objects to detect attacks that are unique to your network (see "Configuring IDP Attack Objects" on page 663.)

Using Security Policy Templates

When you create a new security policy, you have the following options:

- Create a security policy that contains a default firewall rule.
- Select a predefined template.
- Copy an existing security policy into a new policy, which you can then modify.

A template is a set of rules of a specific rulebase type that you can use as a starting point when creating a security policy. For a list of templates, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

Enabling IDP in Firewall Rules

The rules in all rulebases combine to create a security policy. Security devices process and execute rules in each rulebase in the following order:

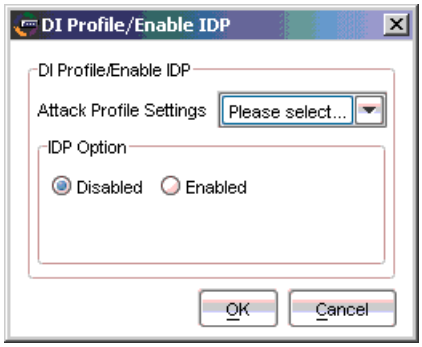
1. Zone-based firewall
2. Global firewall
3. Multicast
4. IDP
5. Exempt
6. Backdoor

Enabling IDP in a zone-based or global firewall rule directs traffic that matches the firewall rule to be checked against the IDP rulebases.

NOTE: The firewall action must be **permit**. You cannot enable IDP for traffic that the device denies or rejects.

To enable IDP in a firewall rule, right-click in the Rule Options column for the zone-based or global firewall rule, then select **DI Profile/Enable IDP**. The DI Profile/Enable IDP dialog box appears, as shown in Figure 157 on page 630.

Figure 157: DI Profile/Enable IDP Dialog Box



NOTE: These attack-profile settings apply only to the Deep Inspection (DI) feature on firewall/VPN devices. When you install the IDP license on the device, DI is disabled on the device.

Enabling IDP

By default, the IDP option is disabled. Select **Enable** to enable IDP for traffic that matches the firewall rule. When you enable IDP, you can also select whether the IDP function is to operate inline or in inline tap mode on the device on which the security policy is installed.

NOTE: If you do not enable IDP in a firewall rule for a target device, you can still configure IDP rules for the device. However, you will not be able to apply the IDP rules when you update the security policy on the device.

Specifying Inline or Inline Tap Mode

IDP on the target device can operate in one of two modes:

- In **inline** mode, IDP is directly in the path of traffic on your network and can detect and block attacks. For example, you can deploy the security device with

integrated firewall/VPN/IDP capabilities between the Internet and an enterprise LAN, WAN, or special zone such as the DMZ. This is the default mode.

- In **inline tap** mode, IDP receives a copy of a packet while the original packet is forwarded on the network. IDP examines the copy of the packet and flags any potential problems. IDP's inspection of packets does not affect the forwarding of the packet on the network.

NOTE: You must deploy the IDP-capable device inline. You cannot connect a device that is in inline tap mode to an external TAP or SPAN port on a switch.

You specify the IDP mode as part of the security policy for the device.

Configuring IDP Rules

The IDP rulebase protects your network from attacks by using attack objects to identify malicious activity and then by taking action to thwart the attacks. Avoid creating too large number of IDP rules. It is important to balance the complexity and number of IDP rules you configure against their potential to cause issues with performance. For more information about performance issues, see the Network and Security Manager documentation athttp://www.juniper.net/techpubs/software/management/security-manager .

When you create an IDP rule, you must specify the following:

- The type of network traffic you want IDP to monitor for attacks, using the following characteristics:
  - **From Zone/To Zone:** All traffic flows from a source to a destination zone. You can select any zone for the source or destination; however, the zone must be valid for the security devices you select in the Install On column of the rule. You can also use zone exceptions to specify unique **to** and **from** zones for each device.
  - **Source IP** The source IP address from which the network traffic originates. You can set this to "any" to monitor network traffic originating from any IP address. You can also specify "negate" to specify all sources except the specified addresses.
  - **User Role**The role of the user. You can select specific user roles to monitor the traffic they initiate, irrespective of IP address. To support use of role-based IDP policies, you must select both Infranet Auth and IDP Enabled in firewall Rule Options. For information on how to configure firewall rule options, see < to be inserted > .
  - When you select the user role, NSM sets Source-IP to any. For a session, the device first searches role-based rules, and, if any matching role is found, it will not try to match IP-based rules.
  - **Destination IP:** The destination IP address to which the network traffic is sent. You can set this to "any" to monitor network traffic sent to any IP

address. You can also specify negate to specify all destinations except the specified addresses.

- **Service:** The Application Layer protocols supported by the destination IP address.
- **Terminate Match:** By default, rules in the IDP rulebase are *nonterminal*, meaning that IDP examines all rules in the rulebase and executes all matches. You can specify that a rule is *terminal*; if IDP encounters a match for the source, destination, and service specified in a terminal rule, it does not examine any subsequent rules for that connection. Note that the traffic does not need to match the attacks specified in the terminal rule. Terminal rules should appear near the top of the rulebase, before other rules that would match the same traffic. Use caution when specifying terminal rules.

  See Figure 169 on page 644. Note that, if you check **Terminate Match**, rules below the Terminate Match Rule (Rule Shadowing) are not evaluated.

  If you do not check **Terminate Match**, multi-event logging/matching occur, which results in one attack creating multiple entries in the logs and multiple actions.

- The attack(s) you want IDP to match in the monitored network traffic. Each attack is defined as an *attack object*, which represents a known pattern of attack. Whenever this known pattern of attack is encountered in the monitored network traffic, the attack object is matched. You can add attack objects individually or by category, operating system, or severity.
- The action you want IDP to take when the monitored traffic matches the rule's attack objects. You can specify the following:
  - **Action:** The action you want IDP to perform against the current connection.
  - **IP Actions:** The action you want IDP to perform against future connections that use the same IP address.
  - **Notification:** Choose **none**; or enable logging, then select the appropriate logging options for your network.
  - **Severity:** Use the default severity settings of the selected attack objects, or choose a specific severity for your rule.

NOTE: The ISG 1000 and ISG 2000 devices with IDP can inspect traffic that is encapsulated in GPRS tunneling protocol (GTP) and generic routing encapsulation (GRE).

IDP inspects only the following two GRE protocol types:

- IP
- PPP for CDMA A10 channel

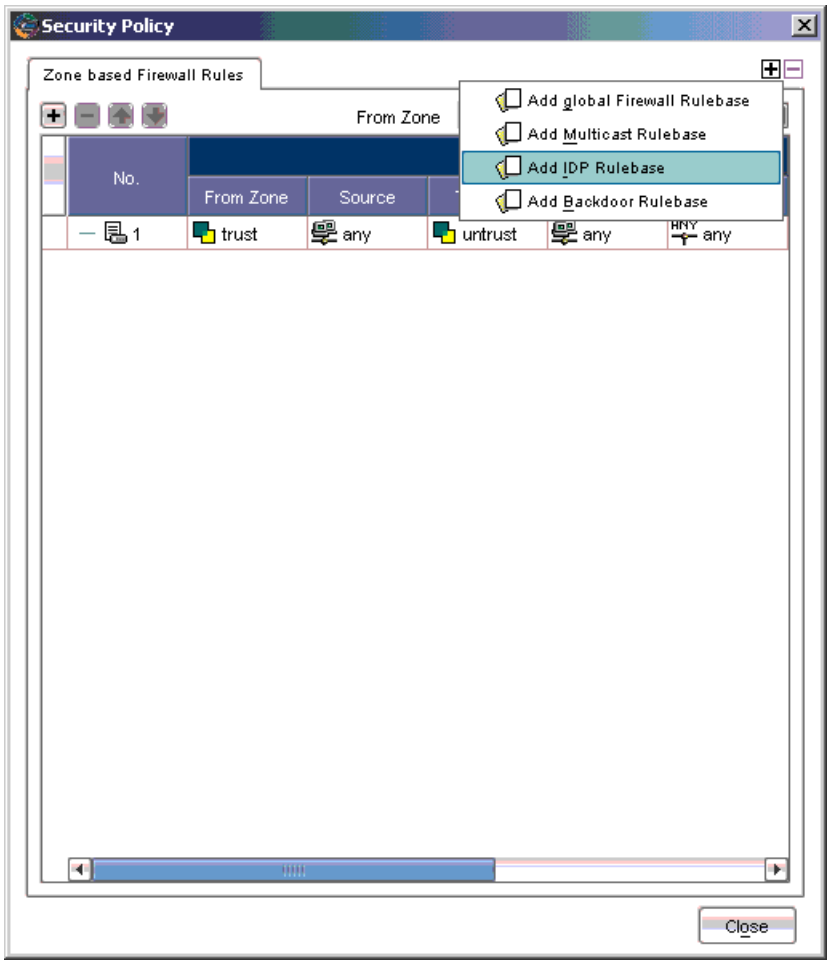
For more information about how IDP inspects GRE and GTP packets, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

Adding the IDP Rulebase

Before you can configure a rule in the IDP rulebase, you need to add the IDP rulebase to a security policy using the following steps:

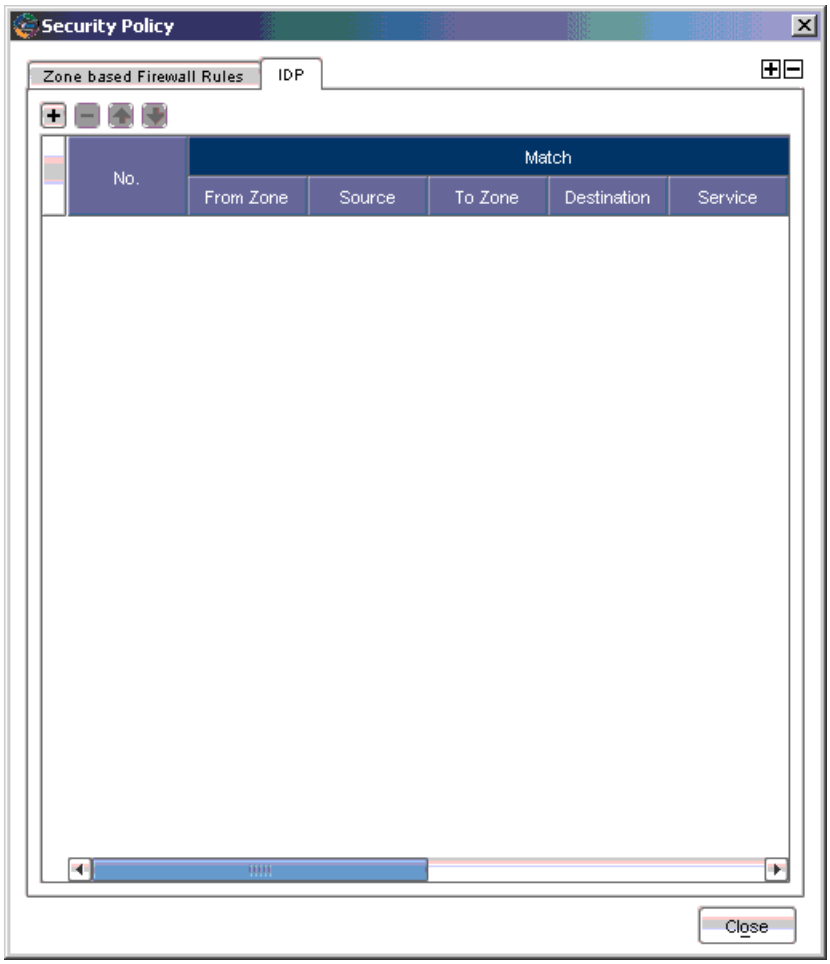
1. In the main navigation tree, select **Security Policies**. Open a security policy either by double-clicking on the policy name in the Security Policy window or by clicking on the policy name and then selecting the Edit icon.
2. Click the Add icon in the upper right corner of the Security Policy window, then select **Add IDP Rulebase.** See Figure 158 on page 634.

Figure 158: Adding an IDP Rulebase



The IDP rulebase tab appears. See Figure 159 on page 635.

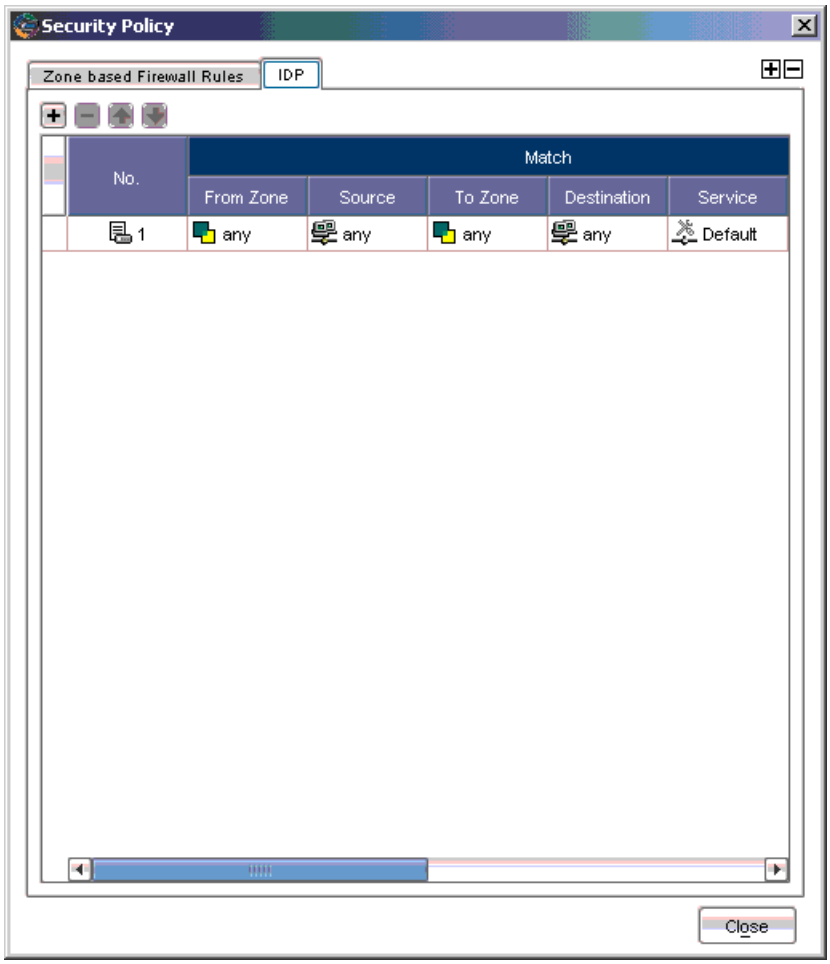
Figure 159: IDP Rulebase Added



3. To configure an IDP rule, click the Add icon on the left side of the Security Policy window.

   A default IDP rule appears. You can modify this rule as needed. See Figure 160 on page 636.

Figure 160: IDP Rule Added



Matching Traffic

When creating your IDP rules, you must specify the type of network traffic that you want IDP to monitor for attacks. These characteristics include the network components that originate and receive the traffic and the firewall zones the traffic passes through.

The Match columns From Zone, Source, To Zone, Destination, and Service are required for all rules in the IDP rulebase. The Terminate Match selection allows you to designate a rule as terminal; if IDP encounters a match for the other Match columns in a terminal rule, no other rules in the rulebase are examined. The matching traffic does not need to match the attacks specified in a terminal rule. (For more information, see “Terminal Rules” on page 642.)

The following sections detail the Match columns of an IDP rule.

Source and Destination Zones

You can select multiple zones for the source and destination; however, these zones must be available on the security devices on which you will install the policy. You can specify "any" for the source or destination zones to monitor network traffic originating from or destined for any zone.

NOTE: You can create custom zones for some security devices. The list of zones from which you can select source and destination zones includes the predefined and custom zones that have been configured for all devices managed by NSM. Therefore, you should only select zones that are applicable for the device on which you will install the security policy.

Source and Destination Address Objects

In the NSM system, address objects are used to represent components on your network: hosts, networks, servers, and so on. Typically, a server or other device on your network is the destination IP for incoming attacks and can sometimes be the source IP for interactive attacks (see "Configuring Backdoor Rules" on page 657 for more information about interactive attacks). You can specify "any" to monitor network traffic originating from any IP address. You can also "negate" the address object(s) listed in the Source or Destination column to specify all sources or destinations except the excluded object(s).

You can create address objects either before you create an IDP rule (see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager) or while creating or editing an IDP rule. To select or configure an address object, right-click on either the Source or the Destination column of a rule, then select **Select Address**. In the **Select Source Addresses** dialog box, you can either select an already created address object or click the Add icon to create a new host, network, or group object.

NOTE: You can create IPv4 and Ipv6 address objects using NSM.

Example: Setting Source and Destination

You want to detect incoming attacks that target your internal network. Set the From Zone to **Untrust** and the Source IP to **any**. Set the To Zone to **dmz** and **trust**. Select the address object that represents the host or server you want to protect from attacks as the Destination IP.

Your rule looks similar to the example shown in Figure 161 on page 638.

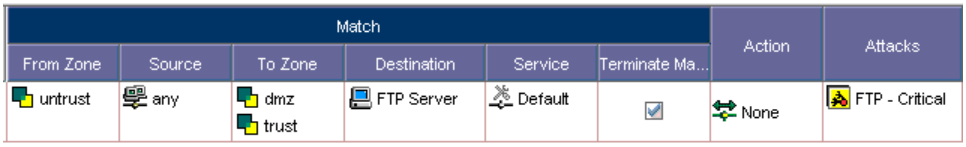
Figure 161: Set Source and Destination

Example: Setting Multiple Sources and Destinations

You want to detect attacks between two networks. Select multiple address objects for the Source and Destination.

Your rule looks similar to the example in Figure 162 on page 638

Figure 162: Set Multiple Source and Destination Networks



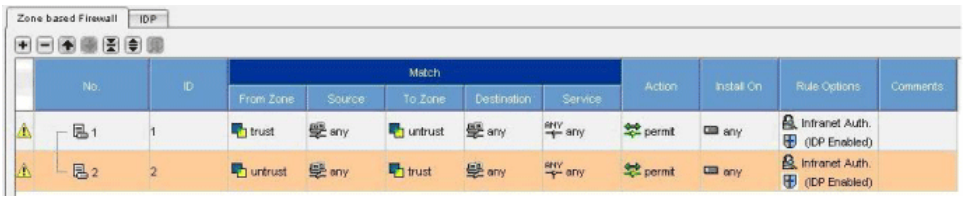
The more specific you are in defining the source and destination of an attack, the more you reduce false positives.

User Role

User-role-based policy management enables the administrator to apply a policy to particular user roles regardless of the location the user logs in.

In order to support role-based IDP policies, you must select Infranet Auth and IDP Enabled in Rule Options while configuring zone-based firewall rules. You firewall rule looks similar to the example in Figure 163 on page 638

Figure 163: Firewall configuration for user-role based policies



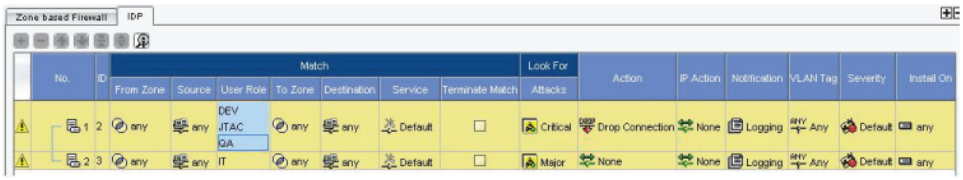
In role-based IDP, the security device calls the UAC module first and finds the role names before forwarding the packet to IDP.

NOTE: The IDP policy engine partitions the policy into two different parts—one with role-based rules and another with IP-based rules. If there are roles for a session, it will search role based rules first, and only if there is no matching role it will check IP based rules.

Example : Setting user-roles

In the example shown in Figure 164 on page 639, there are 2 rules defined. In Rule 2, the security device searches traffic from user roles QA , DEV or JTAC for critical attacks. If it identifies a match, it drops the session and logs the attack. In rule 3, it searches traffic from user role IT for major attacks. If it identifies a match, it logs the attack, but does not take any action.

Figure 164: Setting user-roles



Services

Services are Application Layer protocols that define how data is structured as it travels across the network. Because the services you support on your network are the same services that attackers must use to attack your network, you can specify which services are supported by the destination IP to make your rule more efficient.

NOTE: All services rely on a Transport Layer protocol to transmit data. IDP includes services that use the TCP, UDP, RPC, and ICMP Transport Layer protocols.

Service objects represent the services running on your network. NSM includes predefined service objects that are based on industry-standard services. You use these service objects in rules to specify the service an attack uses to access your network. You can also create custom service objects to represent protocols that are not included in the predefined services. For more information about configuring service objects, see the information about object configuration in the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

In the Service column, you select the service of the traffic you want IDP to match:

- Select **Default** to accept the service specified by the attack object you select in the Attacks column. When you select an attack object in the Attack column, the service associated with that attack object becomes the default service for the rule. To see the exact service, view the attack object details.
- Select **Any** to set any service.

- Select **Select Service** to choose specific services from the list of defined service objects.

Example: Setting Default Services

You want to protect your FTP server from FTP attacks. Set the service to Default, and add an attack object that detects FTP buffer overflow attempts. The Service column in the rule still displays “Default”, but the rule actually uses the default service of TCP-FTP, which is specified in the attack object.

Your rule looks similar to the example shown in Figure 165 on page 640.

Figure 165: Set Default Services

Match						Action	Attacks
From Zone	Source	To Zone	Destination	Service	Terminate Ma...		
untrust	any	dmz trust	FTP Server	Default	☑	None	FTP - Critical

Example: Setting Specific Services

Your mail server supports POP3 and SMTP connections but does not support IMAP. Set POP3 and SMTP service objects as services that can be used to attack the mail server. Because IMAP is not supported, you do not need to add the IMAP service object.

Your rule looks similar to the example in Figure 166 on page 640.

Figure 166: Set Specific Services

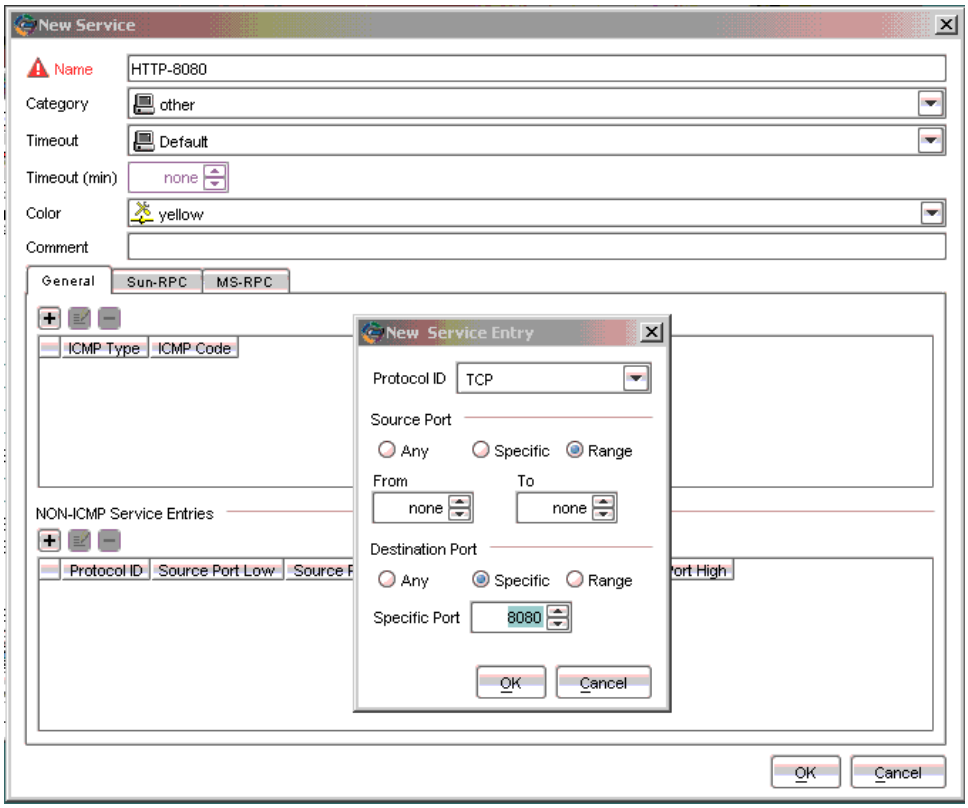
Match				
From Zone	Source	To Zone	Destination	Service
untrust	any	dmz trust	Web Server	SMTP POP3

If you are supporting services on nonstandard ports, you should choose a service other than the default.

Example: Setting Nonstandard Services

You use a nonstandard port (8080) for your HTTP services. Use the Object Manager to create a custom service object on port 8080.

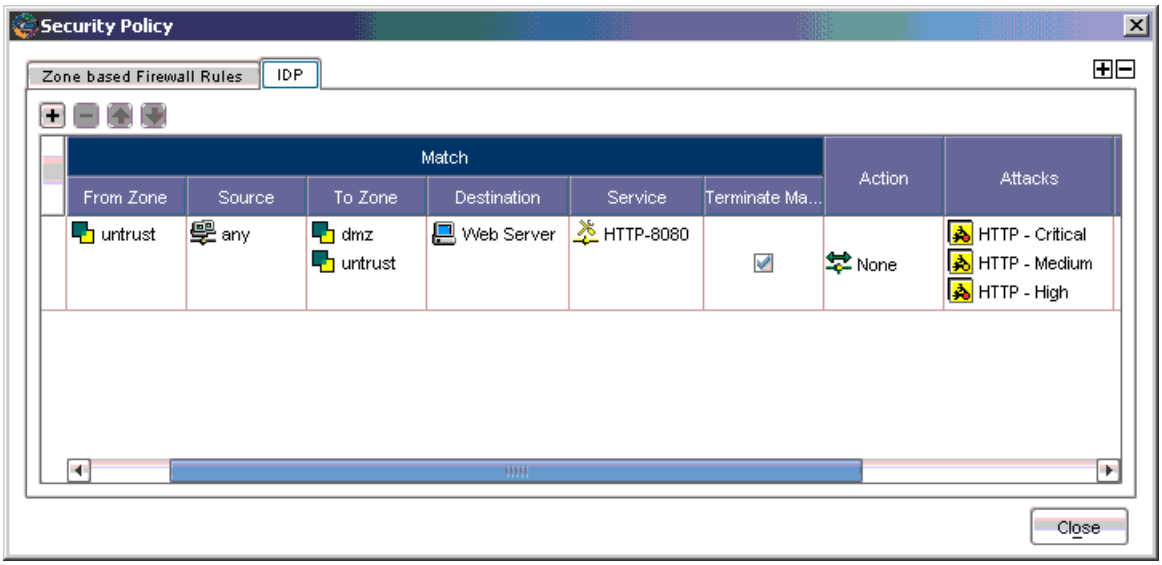
Figure 167: Add Nonstandard Services Object



Add this service object to your rule, then add several HTTP attack objects, which have a default service of TCP/80. IDP uses the specified service, HTTP-8080, instead of the default and looks for matches to the HTTP attacks in TCP traffic on port 8080.

Your rule looks similar to the example in Figure 168 on page 642.

Figure 168: Set Nonstandard Service



You can create your own service objects to use in rules, such as service objects for protocols that use nonstandard ports. However, you cannot match attack objects to protocols they do not use.

Terminal Rules

The normal IDP rule-matching algorithm starts from the top of the rulebase and checks traffic against all rules in the rulebase that match the source, destination, and service. A terminal rule is an exception to this normal rule-matching algorithm. When a match is discovered in a terminal rule for the source, destination, and service, IDP does not continue to check subsequent rules for the same source, destination, and service. It does not matter whether or not the traffic matches the attack objects in the matching rule.

You can use a terminal rule for the following purposes:

- To set different actions for different attacks for the same Source and Destination. This is illustrated by rules 3 and 6 in the following section, “Example: Setting Terminal Rules” on page 643.
- To disregard traffic that originates from a known trusted source. Typically, the action is None for this type of terminal rule. This is illustrated by rule 1 in the following section, “Example: Setting Terminal Rules” on page 643.
- To disregard traffic sent to a server that is only vulnerable to a specific set of attacks. Typically, the action is Drop Connection for this type of terminal rule.

Use caution when defining terminal rules. An inappropriate terminal rule can leave your network open to attacks. Remember that traffic matching the source, destination, and service of a terminal rule is not compared to subsequent rules, even if the traffic does not match an attack object in the terminal rule. Use a terminal rule only when you want to examine a certain type of traffic for one specific set of attack objects.

Be particularly careful about terminal rules that use “any” for both the source and destination.

Terminal rules should appear near the top of the rulebase before other rules that would match the same traffic. You set a rule as terminal by selecting the box in the Terminate Match column of the Security Policy window when you create or modify the rule.

NOTE: In many cases, you can use an exempt rule instead of a terminal rule. You might find it easier and more straightforward to configure an exempt rule than a terminal rule. See “Configuring Exempt Rules” on page 650.

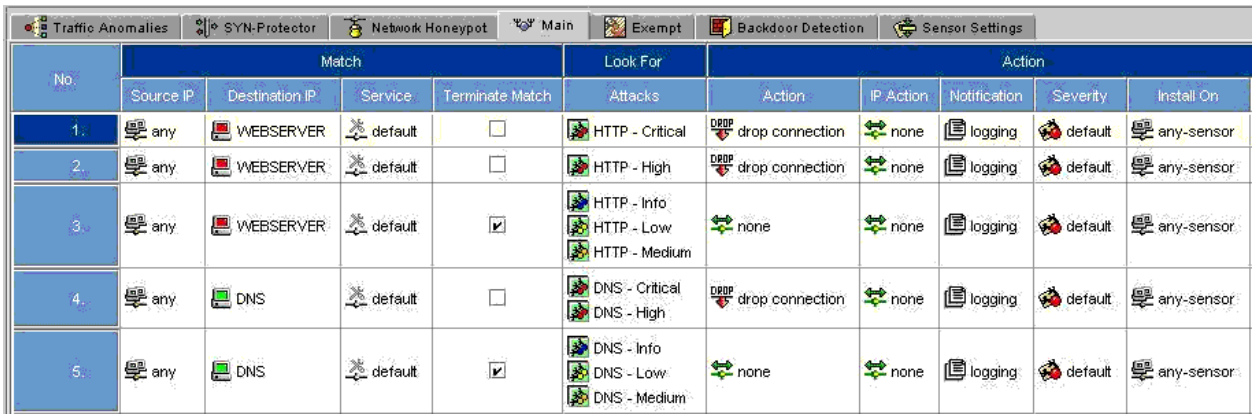
Example: Setting Terminal Rules

In the example IDP rulebase shown below, rules 1, 3, 4, and 5 are configured as terminal rules:

- Rule 1 terminates the match algorithm if the source IP of the traffic originates from the security network, a known trusted network. If this rule is matched, IDP disregards traffic from the security network and does not continue monitoring the session for malicious data.
- Rules 3 and 6 set different actions for different attacks when the destination IP is the Corporate or Europe email server. Rule 3 terminates the match algorithm when the attack is an email that uses the SMTP context Confidential. Rule 6 closes the server when the attack is an SMTP attack.
- Rule 4 terminates the match algorithm when the destination is the Web server and the attack is a Critical or High HTTP attack. The rule ensures that IDP drops the most important HTTP attacks against the Web server and does not continue to match the connection.
- Rule 5 terminates the match algorithm when the source is the internal network and the attack is a critical, high, or medium trojan backdoor. The rule ensures that IDP closes both the client and server and does not continue to match the connection.

The default in the Service Column (see Figure 169 on page 644) means the rule is dynamically built based on the service bindings of the attack objects of that rule. To see the service bindings for a rule, right click on the attacks and select **View Services**. Even if you select a broad category like HTTP Critical, use the View Services for more details.

Figure 169: Set Terminal Rules



Defining Actions

You can specify which actions IDP is to perform against attacks that match rules in your security policy. For each attack that matches a rule, you can choose to ignore, drop, or close the current attacking packets or connection. If the rule is triggered, IDP can perform actions against the connection.

When IDP is configured for Drop Packet and finds a TCP attack, the security module informs the management module that successive packets are attacks; consequently, the IDP action is updated to a higher severity, Drop Connection.

If a packet triggers multiple rule actions, the device will apply the most severe action. For example, if the rules dictate that a packet will receive a diffserv marking and be dropped, then the packet will be dropped.

Table 67 on page 644 shows the actions you can specify for IDP rules.

Table 67: IDP Rule Actions

Action	Description
None	IDP inspects for attacks but IDP does not take action against the connection. If a rule that contains None action is matched, the corresponding log record displays “accept” in the action column of the Log Viewer.
Ignore	IDP does not inspect for attacks and ignores this connection.

Table 67: IDP Rule Actions *(continued)*

Action	Description
Diffserv Marking	Assigns the indicated service-differentiation value to packets in an attack, then passes them on normally. Set the service-differentiation value in the dialog box that appears when you select this action in the rulebase. Note that diffserv marking is not applied to the first packet that is detected as an attack but is applied to subsequent packets. The marking has no effect in tap mode or when using NSRP. If there is a conflict in DSCP specified by the IDS rulebase and the firewall policy, the setting in the IDS rulebase has priority.
Drop Packet	IDP drops a matching packet before it can reach its destination but does not close the connection. Use this action to drop packets for attacks in traffic that is prone to spoofing, such as UDP traffic. Closing a connection for such traffic could result in a denial of service(Dos), which will prevent you from receiving traffic from legitimate source IP addresses.
Drop Connection	IDP drops all packets associated with the connection. Use this action to drop connections for traffic that is not prone to spoofing.
Close Client and Server	IDP closes the connection and sends an RST packet to both the client and the server. If IDP is operating in inline tap mode, IDP sends an RST packet to both the client and the server but does not close the connection.
Close Client	IDP closes the connection to the client but not to the server.
Close Server	IDP closes the connection to the server but not to the client.
Recommended	IDP takes the action recommended in individual attack objects. If IDP detects multiple attacks in one packet and the rule action is Recommended for all the attacks, IDP applies the most severe action. For example, if two attacks match in the same packet and have Drop Packet and Drop as their recommended actions, respectively, Drop is applied. Note: You cannot set Diffserv Marking as the recommended action. Recommended actions in individual attack objects can be overwritten by specifying explicit actions in policy rules. If you specify an action within a policy rule, it will take precedence over the recommended action.

For more information about IDP Rule actions, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

Setting Attack Objects

Attack objects represent specific patterns of malicious activity within a connection, and are a method for detecting attacks. Each attack object detects a known or an

unknown attack that can be used to compromise your network. From more information about attack objects, see “Configuring IDP Attack Objects” on page 663.

You can add attack objects to your rule individually or in groups. Attack objects are organized as follows:

- **Attack List** is an alphabetical list of all attack objects, including custom attack objects.
- **Dynamic Attack Group** contains predefined and custom attack groups.

To add attack objects for a rule, right-click the Attacks column of the rule, then select **Select Attacks**. The Add Attacks dialog box appears.

Adding Attack Objects Individually

The Attack List allows you to select one or more specific attack objects for your rule. The Attack List contains attack objects displayed in alphabetical order. You can also use the integrated search function in NSM to locate a specific word or string in the attack object name. For more information about using the search feature, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

For more information about attack objects and creating custom attack objects and groups, see “Configuring IDP Attack Objects” on page 663.

Adding Attack Objects by Category

IDP groups attack objects into predefined service category groups. Services are Application Layer protocols which define how data is structured as it travels across the network.

To attack a system, an attacker must use a protocol supported on that system. Therefore, When you create a rule to protect a system, you must select only the categories that are used by the address objects you are protecting with the rule.

Example: Adding Attack Objects by Service

You rely on FTP and HTTP for extensive file transfer on your Web server. Choose the FTP and HTTP category groups to carefully monitor all traffic that uses these services.

If you do not want to choose an entire category group for a rule, you can select your attack objects by severity.

Adding Attack Objects by Operating System

IDP groups attack objects for several predefined operating systems to help you choose the attack objects that are the most dangerous to specific devices on your network. You can choose BSD, Linux, Solaris, or Windows.

If you do not want to choose an entire operating system group for a rule, you can select your attack objects by severity.

Adding Attack Objects by Severity

IDP defines five severity levels, each with a recommended set of IDP actions and notifications (see Table 68 on page 647). You can add a severity level to the Attacks column in your rule, then choose the recommended actions for the severity level in the Action column. (For more information about the actions you can select, see "Defining Actions" on page 644.) You can also choose the recommended notifications for the severity level in the Notifications column. (For more information about the notifications you can select, see "Setting Notification" on page 649.)

NOTE: To protect critical address objects or popular targets for attack, such as your mail server, use multiple severity levels to ensure maximum protection.

Table 68 on page 647 shows the IDP severity levels, along with their recommended actions and notifications.

Table 68: Severity Levels with Recommended Actions and Notifications

Severity Level	Description	Recommended Action	Recommended Notification
Critical	Attacks attempt to evade an IPS, crash a machine, or gain system-level privileges.	Drop Packet Drop Connection	Logging Alert
Major	Attacks attempt to crash a service, perform a denial of service, install or use a Trojan, or gain user-level access to a host.	Drop Packet Drop Connection	Logging Alert
Minor	Attacks attempt to obtain critical information through directory traversal or information leaks.	None	Logging
Warning	Attacks attempt to obtain noncritical information or scan the network. They can also be obsolete attacks (but probably harmless) traffic.	None	Logging
Info	Attacks are normal, harmless traffic containing URLs, DNS lookup failures, and SNMP public community strings. You can use informational attack objects to obtain information about your network.	None	None

Setting IP Actions

The IP Action column appears only when you view the security policy in expanded mode. To change the security policy view from compact to expanded mode, select **View > Expanded Mode**.

If the current network traffic matches a rule, IDP can perform an IP action against future network traffic that uses the same IP address. IP actions are similar to other actions; they direct IDP to drop or close the connection. However, because you now

also have the attacker's IP address, you can choose to block the attacker for a specified time. If attackers cannot immediately regain a connection to your network, they might try to attack easier targets.

Use IP actions in conjunction with actions and logging to secure your network. In a rule, first configure an action to detect and prevent current malicious connections from reaching your address objects. Then, right-click in the IP Action column of the rule and select **Configure** to bring up the Configure IP Action dialog box. Enable and configure an IP action to prevent future malicious connections from the attacker's IP address.

Choosing an IP Action

For each IP action option, an IP action is generated by the IDP system. The IP action instructs IDP to perform the specified task. Select from the following options:

- **IDP Notify**. IDP does not take any action against future traffic, but logs the event. This is the default.
- **IDP Drop**. IDP drops blocks future connections that match the criteria in the Blocking Options box.
- **IDP Close**. IDP closes future connections that match the criteria in the Blocking Options box.

Choosing a Blocking Option

Each blocking option follows the criteria you set in the Actions box. Blocking options can be based on the following matches of the attack traffic:

- **Source, Destination**, **Destination Port** and **Protocol**. IDP blocks future traffic based on the source, destination, destination port, and protocol of the attack traffic. This is the default.
- **Source**. IDP blocks future traffic based on the source of the attack traffic.
- **Destination**. IDP blocks future traffic based on the destination of the attack traffic.
- **From Zone**, **Destination**, **Destination Port** and **Protocol**. IDP blocks future traffic based on the source zone, destination, destination port, and protocol of the attack traffic.
- **From Zone**. IDP blocks future traffic based on the source zone of the attack traffic.

Setting Logging Options

When IDP detects attack traffic that matches a rule and triggers an IP action, IDP can log information about the IP action or create an alert in the Log Viewer. By default, no logging options are set.

Setting Timeout Options

You can set the number of seconds that you want the IP action to remain in effect after a traffic match. For permanent IP actions, the default timeout value is 0.

Setting Notification

The first time you design a security policy, you might be tempted to log all attacks and let the policy run indefinitely. Do not do this! Some attack objects are informational only, and others can generate false positives and redundant logs. If you become overloaded with data, you can miss something important. Remember that security policies that generate too many log records are hazardous to the security of your network, as you might discover an attack too late or miss a security breach entirely if you have to sift through hundreds of log records. Excessive logging can also affect IDP throughput, performance, and available disk space. A good security policy generates enough logs to fully document only the important security events on a network.

NOTE: The Juniper Networks security devices support packet capture and packet logs for IPV6 traffic.

Setting Logging

In the Configure Notification dialog box, select **Logging**, then click **OK**. Each time the rule is matched, the IDP system creates a log record that appears in the Log Viewer.

Logging an attack creates a log record that you can view in realtime in the Log Viewer. For more critical attacks, you can also set an alert flag to appear in the log record.

To log an attack for a rule, right-click the Notification column of the rule, then select **Configure**. The Configure Notification dialog box appears.

Setting an Alert

In the Configure Notification dialog box, select **Alert**, then click **OK**. If **Alert** is selected and the rule is matched, IDP places an alert flag in the Alert column of the Log Viewer for the matching log record.

Logging Packets

You can record individual packets in network traffic that match a rule by capturing the packet data for the attack. Viewing the packets used in an attack on your network can help you determine the extent of the attempted attack and its purpose, whether or not the attack was successful, and any possible damage to your network.

NOTE: To improve IDP performance, log only the packets received after the attack.

If multiple rules with packet capture enabled match the same attack, IDP captures the maximum specified number of packets. For example, you configure rule 1 to capture 10 packets before and after the attack, and you configure rule 2 to capture 5 packets before and after the attack. If both rules match the same attack, IDP attempts to capture 10 packets before and after the attack.

NOTE: Packet captures are restricted to 256 packets before and after an attack.

Setting Severity

The Severity column appears only when you view the security policy in expanded mode. To change the security policy view from compact to expanded mode, from the menu bar, select **View > Expanded Mode**.

You can override the inherent severity for an attack in a rule within the IDP rulebase. You can set the severity level to Default, Info, Warning, Minor, Major, or Critical.

To change the severity for a rule, right-click the Severity column of the rule, then select a severity.

Setting Targets

For each rule in the IDP rulebase, you can select the security device that will use that rule to detect and prevent attacks. When you install the security policy to which the rule belongs, the rule becomes active only on the device(s) you selected in the Install On column of the rulebase.

Entering Comments

You can enter notations about the rule in the Comments column. The information in the Comments column is not pushed to the target device(s). To enter a comment, right-click on the Comments column, then select **Edit Comments**. The Edit Comments dialog box appears. You can enter a comment of up to 1024 characters.

Configuring Exempt Rules

The Exempt rulebase works in conjunction with the IDP rulebase. Before you can create exempt rules, you must first create rules in the IDP rulebase. If traffic matches a rule in the IDP rulebase, IDP attempts to match the traffic against the Exempt rulebase before performing the specified action or creating a log record for the event.

NOTE: If you delete the IDP rulebase, the Exempt rulebase is also deleted.

You might want to use an exempt rule under the following conditions:

- When an IDP rule uses an attack object group that contains one or more attack objects that produce false positives or irrelevant log records.
- When you want to exclude a specific source, destination, or source/destination pair from matching an IDP rule. This prevents IDP from generating unnecessary alarms.

You can also use an exempt rule if the IDP rulebase uses static or dynamic attack-object groups containing one or more attack objects that produce false positives or irrelevant log records.

When you create an exempt rule, you must specify the following:

- Source and destination for traffic you want to exempt. You can set the source or destination to "any" to exempt network traffic originating from any source or sent to any destination. You can also specify "negate" to specify all sources or destinations except specified addresses.
- The attack(s) you want IDP to exempt for the specified source/destination addresses. You must include at least one attack object in an exempt rule.

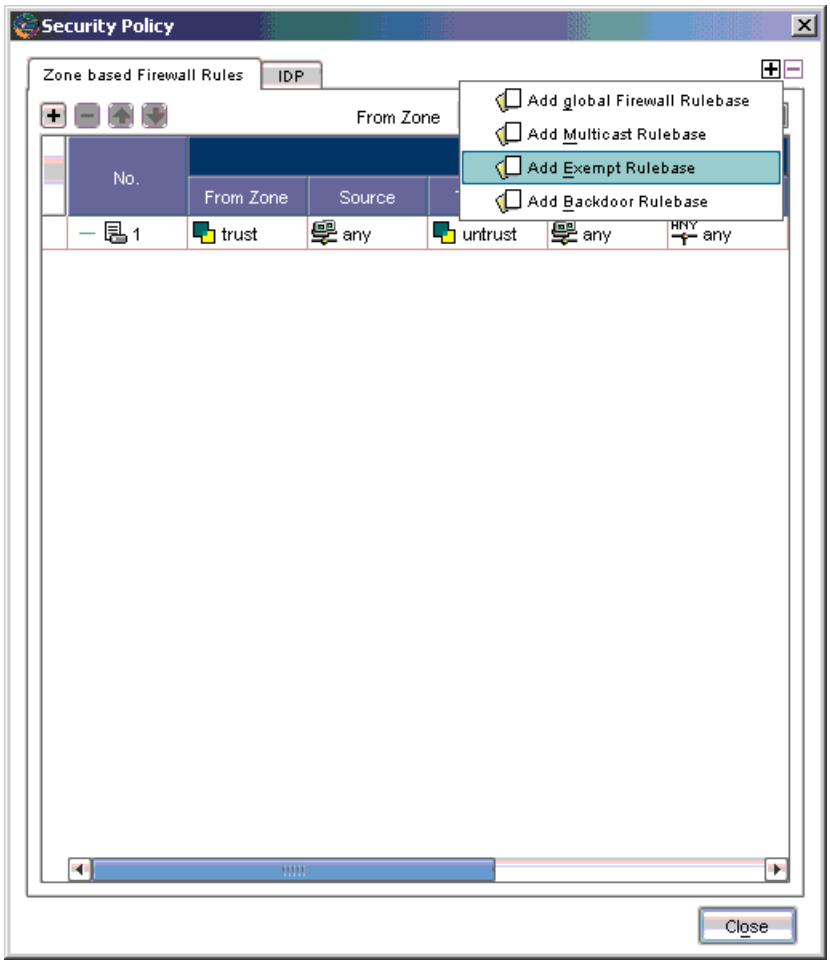
NOTE: The Exempt rulebase is a nonterminal rulebase. That is, IDP attempts to match traffic against all rules in the Exempt rulebase and executes all matches.

Adding the Exempt Rulebase

Before you can configure a rule in the Exempt rulebase, you need to add the Exempt rulebase to a security policy with the following steps:

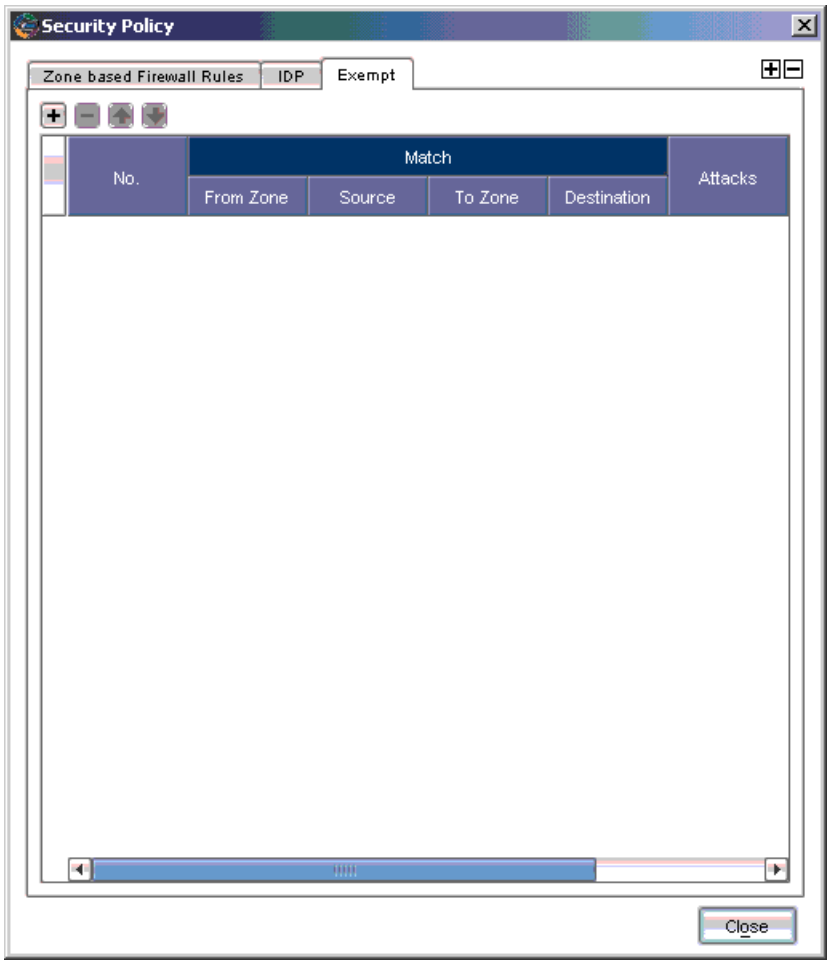
1. In the main navigation tree, select **Security Policies**. Open a security policy either by double-clicking on the policy name in the Security Policies window, or by clicking on the policy name, then selecting the Edit icon.
2. Click the Add icon in the upper right corner of the Security Policy window, then select **Add Exempt Rulebase**.

Figure 170: Adding an Exempt Rulebase



The Add Exempt Rulebase tab appears.

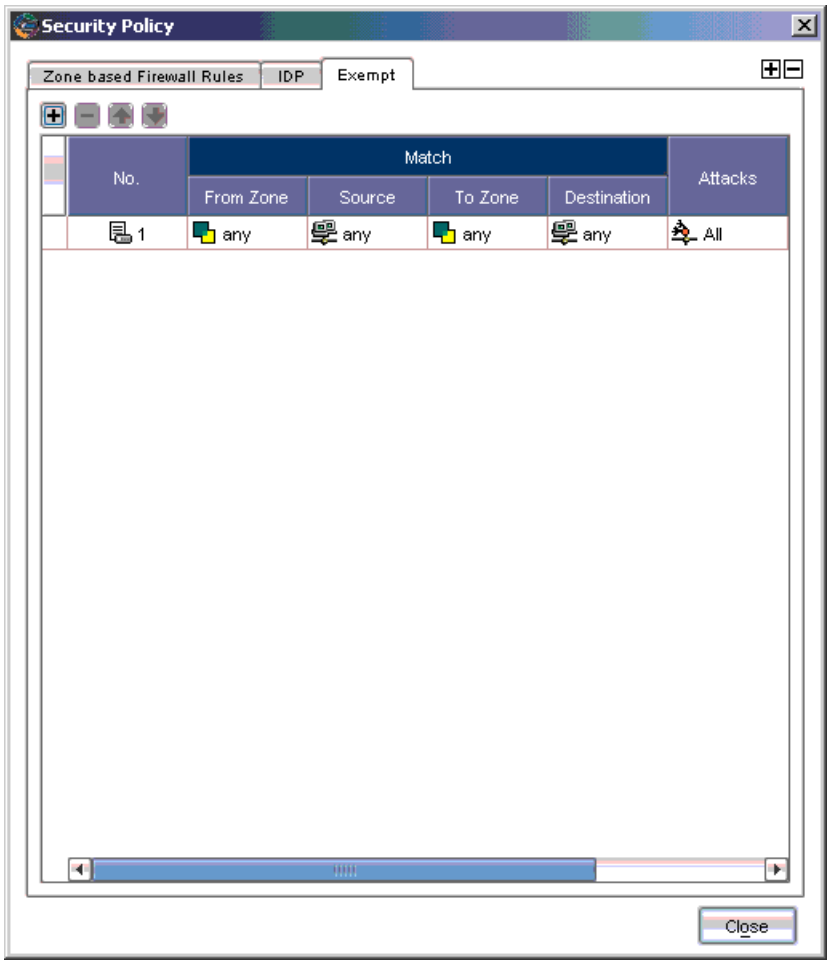
Figure 171: Exempt Rulebase Added



3. To configure an exempt rule, click the Add icon on the left side of the Security Policy window.

   A default exempt rule appears. You can modify this rule as needed.

Figure 172: Exempt Rule Added



Defining a Match

Specify the traffic you want to exempt from attack detection. The Match columns From Zone, Source, To Zone, and Destination are required for all rules in the exempt rulebase.

The following sections detail the Match columns of an exempt rule.

Source and Destination Zones

You can select multiple zones for the source and destination, however these zones must be available on the devices on which you will install the policy. You can specify “any” for the source or destination zones to monitor network traffic originating from or destined for any zone.

NOTE: You can create custom zones for some security devices. The list of zones from which you can select source and destination zones includes the predefined and custom zones that have been configured for all devices managed by NSM. Therefore, you should only select zones that are applicable for the device on which you will install the security policy.

Source and Destination Address Objects

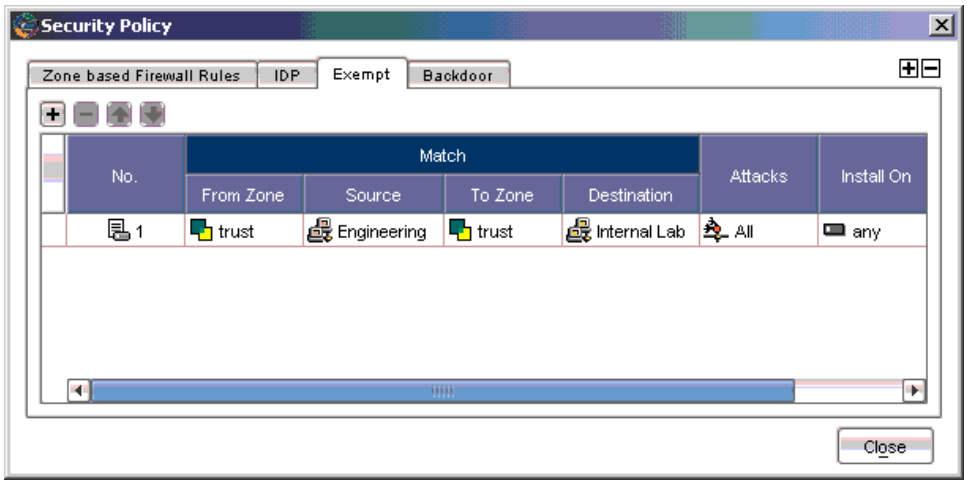
In the NSM system, address objects are used to represent components on your network: hosts, networks, servers, and so on. You can specify "any" to monitor network traffic originating from any IP address. You can also negate the address object(s) listed in the Source or Destination column of a rule to specify all sources or destinations except the excluded object.

You can create address objects either before you create an exempt rule (see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager) or while creating or editing an exempt rule. To select or configure an address object, right-click on either the Source or Destination column of a rule, then select **Select Address**. In the Select Source Addresses dialog box, you can either select an already created address object, or you can click the Add icon to create a new host, network, or group object.

Example: Exempting a Source/Destination Pair

To improve performance and eliminate false positives between your Internal Lab devices and your Engineering desktops, you want to exempt attack detection. Your exempt rule looks similar to Figure 173 on page 655:

Figure 173: Exempting Source and Destination



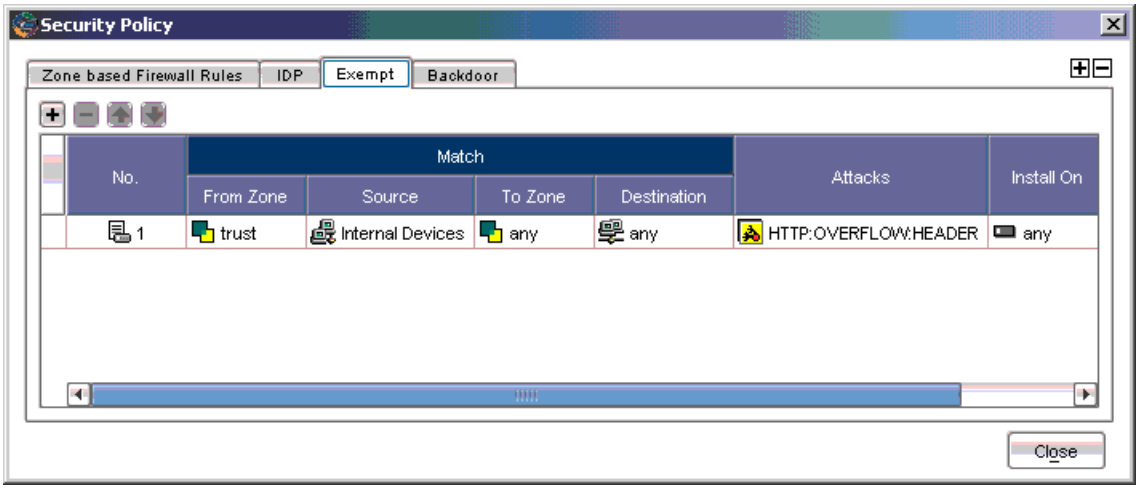
Setting Attack Objects

You specify the attack(s) you want IDP to exempt for the specified source/destination addresses. You must include at least one attack object in an exempt rule.

Example: Exempting Specific Attack Objects

You consistently find that your security policy generates false positives for the attack HTTP Buffer Overflow: Header on your internal network. You want to exempt attack detection for this attack when the source IP is from your internal network. Your exempt rule looks similar to Figure 174 on page 656:

Figure 174: Exempting Attack Object



Setting Targets

For each rule in the Exempt rulebase, you can select the IDP-capable device that will use that rule to detect and prevent attacks. When you install the security policy to which the rule belongs, the rule becomes active only on the device(s) you select in the Install On column of the rulebase.

Entering Comments

You can enter notations about the rule in the Comments column. The information in the Comments column is not pushed to the target device(s). To enter a comment, right-click on the Comments column, then select **Edit Comments**. The Edit Comments dialog box appears. You can enter a comment of up to 1024 characters.

Creating an Exempt Rule from the Log Viewer

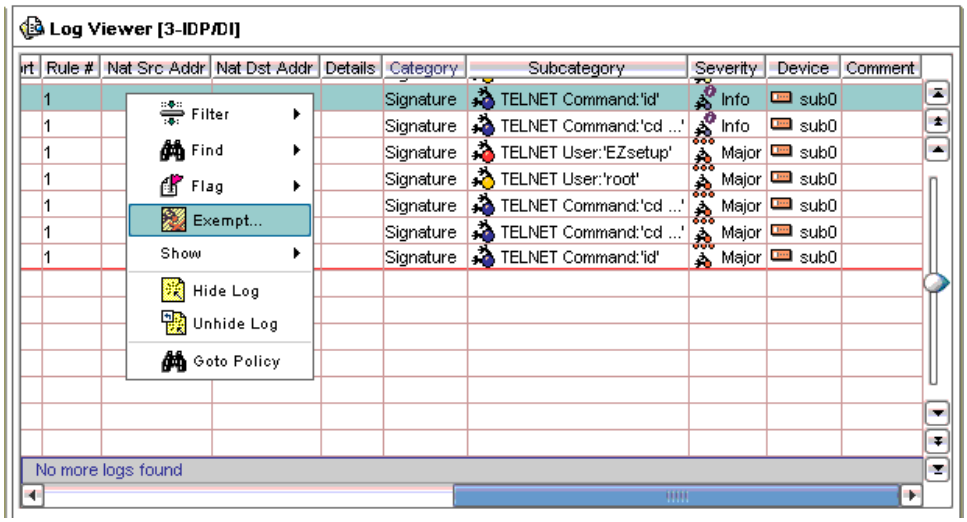
You can also create a rule in the Exempt rulebase directly from the NSM Log Viewer. You might want to use this method to quickly eliminate rules that generate false positive log records. (For more information about viewing IDP logs, see “Managing

IDP" on page 687. For more information about using the Log Viewer, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager).

To create an exempt rule from the Log Viewer, perform the following steps:

1. View the IDP/DI logs in the Log Viewer.
2. Right-click a log record that contains an attack you want to exempt, then select **Exempt**.

Figure 175: Exempting a Log Record Rule



The Exempt rulebase for the security policy that generated the log record is displayed, with the exempt rule that is associated with the log entry. The source, destination, and attack settings for the rule are automatically filled in based on the information in the log record.

NOTE: If the Exempt rulebase does not already exist when you create an exempt rule from the Log Viewer, the rulebase is automatically created and the rule is added.

You can modify, reorder, or merge an exempt rule created from the Log Viewer in the same manner as any other exempt rule that you create directly in the Exempt rulebase.

Configuring Backdoor Rules

A backdoor is a mechanism installed on a host computer that facilitates unauthorized access to a system. Attackers who have already compromised a system can install a backdoor to make future attacks easier. When attackers type commands to control a backdoor, they generate interactive traffic.

Unlike antivirus software, which scans for known backdoor or executable files on the host system, IDP detects the interactive traffic that is produced when backdoors are used. If interactive traffic is detected, IDP can perform IP actions against the connection to prevent an attacker from further compromising your network.

When you configure a backdoor rule, you must specify the following:

- Source and destination addresses for traffic you want to monitor. To detect incoming interactive traffic, set the Source to "any" and the Destination to the IP address of network device you want to protect. To detect outgoing interactive traffic, set the Source to the IP address of the network device you want to protect and the Destination to "any."
- Services that are offered by the Source or Destination as well as interactive services that can be installed and used by attackers.

NOTE: Do not include Telnet, SSH, RSH, NetMeeting, or VNC, as these services are often used to legitimately control a remote system and their inclusion might generate false positives.

- Action that the IDP is to perform if interactive traffic is detected. Set the Operation to "detect." If you are protecting a large number of network devices from interactive traffic, you can create a rule that ignores accepted forms of interactive traffic from those devices, then create another rule that detects all interactive traffic from those devices.

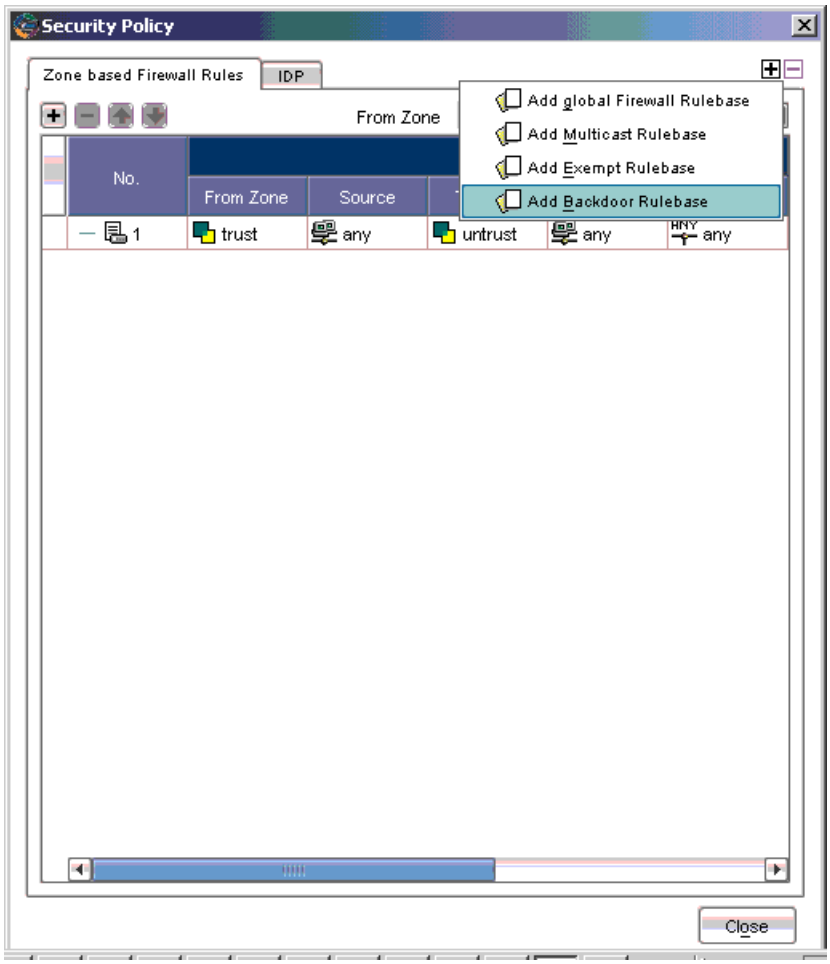
NOTE: The Backdoor rulebase is a terminal rulebase. That is, when IDP finds a match on a rule in the Backdoor rulebase, it does not execute successive rules.

Adding the Backdoor Rulebase

Before you can configure a rule in the Backdoor rulebase, you need to add the Backdoor rulebase to a security policy with the following steps:

1. In the main navigation tree, select **Security Policies**. Open a security policy either by double-clicking on the policy name in the Security Policy window, or by clicking on the policy name, then selecting the Edit icon.
2. To configure a backdoor rule, click the Add icon in the upper right corner of the Security Policy window (see Figure 176 on page 659).

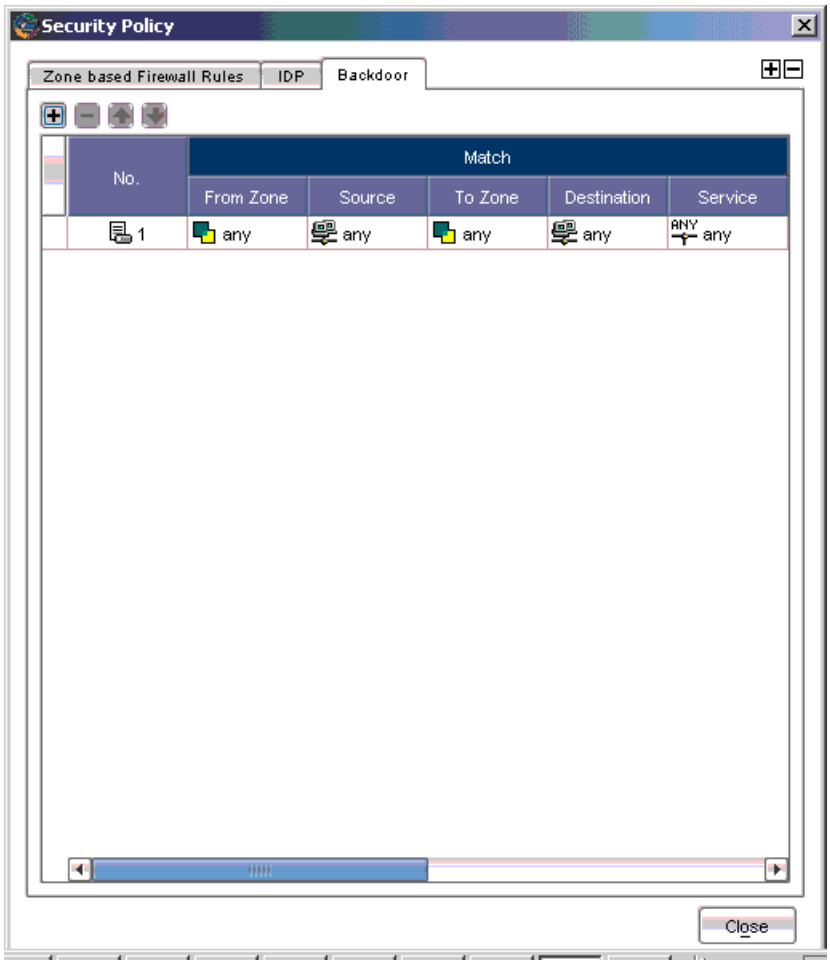
Figure 176: Adding the Backdoor Rulebase



3. Select **Add Backdoor Rulebase**.

   A default backdoor rule appears as shown in Figure 177 on page 660. You can modify this rule as needed.

Figure 177: Backdoor Rule Added



Defining a Match

You specify the traffic you want IDP to monitor for indications of backdoors or Trojans. The Match columns From Zone, Source, To Zone, Destination, and Service are required for all rules in the Backdoor rulebase.

The following sections detail the Match columns of a backdoor rule.

Source and Destination Zones

You can select multiple zones for the source and destination. However, these zones must be available on the security devices on which you will install the policy. You can specify “any” for the source or destination zones to monitor network traffic originating from or destined for any zone.

NOTE: You can create custom zones for some security devices. The list of zones from which you can select source and destination zones includes the predefined and custom zones that have been configured for all devices managed by NSM. Therefore, you should only select zones that are applicable for the device on which you will install the security policy.

Source and Destination Address Objects

In the NSM system, address objects are used to represent components on your network: hosts, networks, servers, and so on. Typically, a server or other device on your network is the destination IP for incoming attacks and it can sometimes be the source IP for interactive attacks. You can specify "any" to monitor network traffic originating from any IP address. You can also negate the address object(s) listed in the Source or Destination column to specify all sources or destinations except the excluded address object.

You can create address objects either before you create a backdoor rule (see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager) or while creating or editing a backdoor rule. To select or configure an address object, right-click on either the Source or Destination column of a rule, then select **Select Address**. In the Select Source Addresses dialog box, you can either select an already created address object or you can click the **Add** icon to create a new host, network, or group object.

Services

Select interactive service objects. Be sure to include services that are offered by the source or destination IP as well as interactive services that are not; attackers can use a backdoor to install any interactive service. Do not include Telnet, SSH, RSH, NetMeeting, or VNC, as these services are often used to control a remote system legitimately, and their inclusion might generate false positives.

Setting the Operation

Set the Operation to **detect** or **ignore**. If you select **detect**, choose an action to perform if backdoor traffic is detected. If you are protecting a large number of address objects from interactive traffic, you can create a rule that ignores accepted forms of interactive traffic from those objects, then create a succeeding rule that detects all interactive traffic from those objects.

Setting Actions

Use the following steps to configure an action to perform if IDP detects interactive traffic:

Table 69: Actions for Backdoor Rule

Action	Description
Accept	IDP accepts the interactive traffic.
Drop Connection	IDP drops the interactive connection without sending an RST packet to the sender, preventing the traffic from reaching its destination. Use this action to drop connections for traffic that is not prone to spoofing.
Close Client and Server	IDP closes the interactive connection and sends an RST packet to both the client and the server. If the IDP is in sniffer mode, IDP sends an RST packet to both the client and server but does not close the connection.
Close Client	IDP closes the interactive connection to the client but not to the server.
Close Server	IDP closes the interactive connection to the server but not to the client.

Setting Notification

The first time you design a security policy, you might be tempted to log all attacks and let the policy run indefinitely. Do not do this! Some attack objects are informational only, and others can generate false positives and redundant logs. If you become overloaded with data, you might miss something important. Remember that security policies that generate too many log records are hazardous to the security of your network, as you might discover an attack too late or miss a security breach entirely if you have to sift through hundreds of log records. Excessive logging can also affect IDP throughput, performance, and available disk space. A good security policy generates enough logs to fully document only the important security events on a network.

Setting Logging

In the Configure Notification dialog box, select **Logging**, then click **OK**. Each time the rule is matched, the IDP system creates a log record that appears in the Log Viewer.

Logging an attack creates a log record that you can view real time in the Log Viewer. For more critical attacks, you can also set an alert flag to appear in the log record, notify you immediately by email, have IDP run a script in response to the attack, or set an alarm flag to appear in the log record. Your goal is to fine tune the attack notifications in your security policy to your individual security needs.

To log an attack for a rule, right-click the Notification column of the rule, then select **Configure**. The Configure Notification dialog box appears.

Setting an Alert

In the Configure Notification dialog box, select **Alert**, then click **OK**. If Alert is selected and the rule is matched, IDP places an alert flag in the Alert column of the Log Viewer for the matching log record.

Logging Packets

You can record the individual packets in network traffic that matched a rule by capturing the packet data for the attack. Viewing the packets used in an attack on your network can help you determine the extent of the attempted attack and its purpose, whether or not the attack was successful, and any possible damage to your network.

NOTE: To improve IDP performance, log only the packets following the attack.

If multiple rules with packet capture enabled match the same attack, IDP captures the maximum specified number of packets. For example, you configure rule 1 to capture 10 packets before and after the attack, and you can configure rule 2 to capture 5 packets before and after the attack. If both rules match the same attack, IDP attempts to capture 10 packets before and after the attack.

NOTE: Packet captures are restricted to 256 packets before and after the attack.

Setting Severity

You can override the inherent attack severity for a rule within the Backdoor rulebase. You can set the severity to Default, Info, Warning, Minor, Major, or Critical.

To change the severity for a rule, right-click the Severity column, then select a severity.

Setting Targets

For each rule in the Backdoor rulebase, you can select the security device that will use that rule to detect and prevent attacks. When you install the security policy to which the rule belongs, the rule becomes active only on the devices you select in the Install On column of the rulebase.

Entering Comments

You can enter notations about the rule in the Comments column. The information in the Comments column is not pushed to the target device(s). To enter a comment, right-click on the Comments column, then select **Edit Comments**. The Edit Comments dialog box appears. You can enter a comment of up to 1024 characters.

Configuring IDP Attack Objects

Attack objects contain patterns for known attacks that attackers can use to compromise your network. Attack objects do not work on their own—they need to be part of a rule before they can start detecting known attacks and preventing malicious traffic from entering your network. To use attack objects in your IDP rules,

add the IDP rulebase in your security policy, then add an IDP rule to the rulebase. See “Configuring Security Policies” on page 626.

NOTE: IDP is supported only on security devices with IDP capabilities.

About IDP Attack Object Types

In a security policy, you can select the attack object that a device uses to detect attacks against your network. If an attack is detected, the device generates an attack log entry that appears in the Log Viewer. For more information, see “Configuring IDP Rules” on page 631.

NSM supports three types of IDP attack objects:

- Signature attack objects
- Protocol anomaly attack objects
- Compound attack objects

The following sections detail each attack object type.

Signature Attack Objects

An attack *signature* is a pattern that always exists within an attack; if the attack is present, so is the attack signature. IDP uses stateful *signatures* to detect attacks. Stateful signatures are more specific than regular signatures. With stateful signatures, IDP can look for the specific protocol or service used to perpetrate the attack, the direction and flow of the attack, and the context in which the attack occurs. ScreenOS combines the attack pattern with service, context, and other information into the attack object. Stateful signatures produce few false positives because the context of the attack is defined, eliminating huge sections of network traffic in which the attack would not occur.

Protocol Anomaly Attack Objects

Protocol anomaly attack objects detect abnormal or ambiguous messages within a connection according to the set of rules for the particular protocol being used. Protocol anomaly detection works by finding deviations from protocol standards, most often defined by RFCs and common RFC extensions. Most legitimate traffic adheres to established protocols. Traffic that does not produces an anomaly, which may be created by attackers for specific purposes, such as evading an IPS.

Compound Attack Objects

A compound attack object combines multiple signatures and/or protocol anomalies into a single object. Traffic must match all of the combined signatures and/or protocol anomalies to match the compound attack object; you can specify the order in which signatures or anomalies must match. Use compound attack objects to refine your security policy rules, reduce false positives, and increase detection accuracy.

A compound attack object enables you to be very specific about the events that need to occur before IDP identifies traffic as an attack. For example, you might want to take action only if an FTP session includes a failed login attempt for specific users.

Viewing Predefined IDP Attack Objects and Groups

We provide predefined attack objects and attack object groups that you can use in security policies to match traffic against known attacks. we regularly update the predefined attack objects and groups. While you cannot create, edit, or delete predefined attack objects, you can update the list of attack objects that you can use in security policies. The revised attack objects and groups are available as part of an attack database update, which is downloaded to the NSM GUI Server. See "Managing IDP" on page 687 for information about attack-database updates.

To view predefined attack objects and groups perform the following steps:

1. In the Object Manager, click **Attack Objects > IDP Objects**. The IDP Objects dialog box appears.
2. Click the Predefined Attacks or Predefined Attack Groups tab to view predefined attack objects or groups.

Viewing Predefined Attacks

The Predefined Attacks tab displays all attacks in a table format and includes the following information:

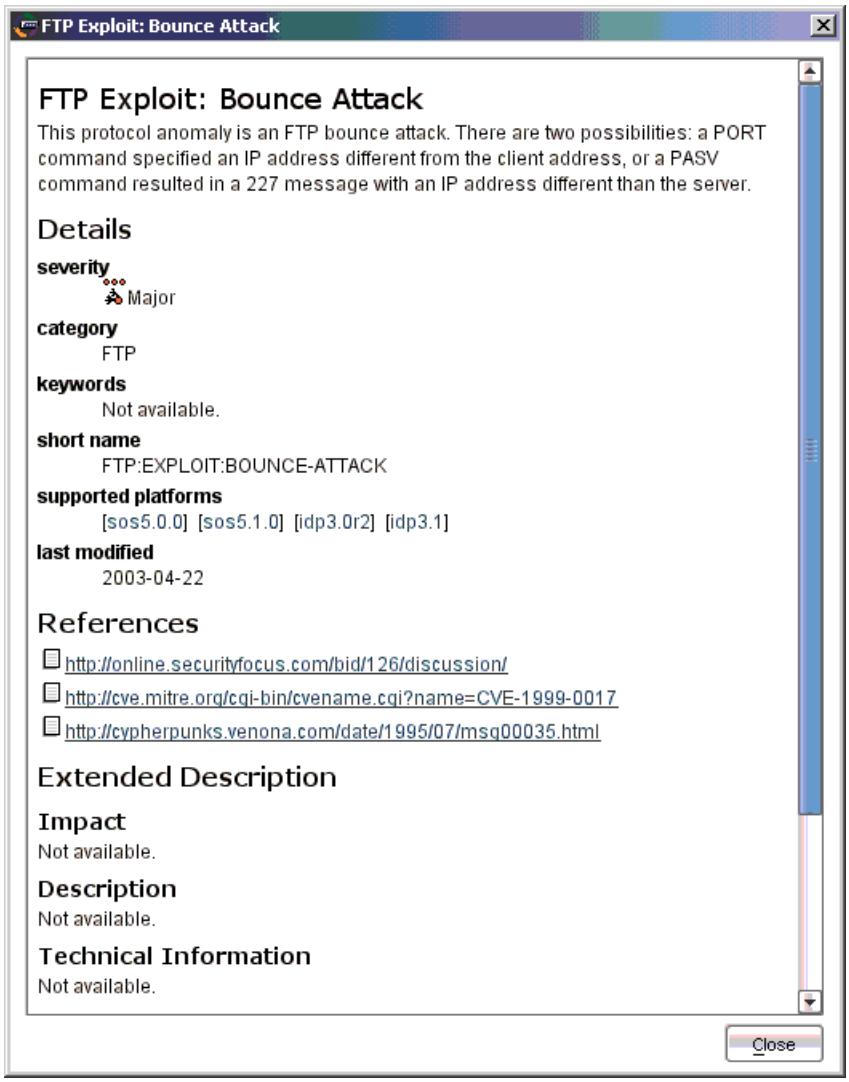
- Name of the attack object
- Severity of the attack: critical, major, minor, warning, info
- Category
- Keywords for the attack
- CVE number, which identifies the attack's number in the Common Vulnerabilities and Exposures database
- Bugtraq number, which identifies the equivalent attack in the Security Focus Bugtraq database

Initially, attack objects are listed alphabetically by Category name but you can view attacks in different orders by clicking on a column heading.

To locate all rules that use a predefined attack object, right-click the attack object, then select **View Usages**.

To display a detailed description of an attack object, double-click on the attack.

Figure 178: Attack Viewer



Viewing Predefined Groups

The Predefined Attack Group tab displays the following predefined attack groups:

- **Recommended**—a list of all attack object objects that we consider serious threats, organized into categories.
  - **Attack** Type groups attack objects by type (anomaly or signature). Within each type, attack objects are grouped by severity.
  - **Category** groups attack objects by predefined categories. Within each category, attack objects are grouped by severity.

- **Operating System** groups attack objects by the operating system to which they apply: BSD, Linux, Solaris, or Windows. Within each operating system, attack objects are grouped by services and severity.
- **Severity** groups attack objects by the severity assigned to the attack. IDP has five severity levels: Info, Warning, Minor, Major, Critical. Within each severity, attack objects are grouped by category.

To locate all rules that use a predefined attack object group, right-click the attack object group, then select **View Usages**.

To display a detailed description of an attack object, double-click on the attack.

For more information about predefined groups, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

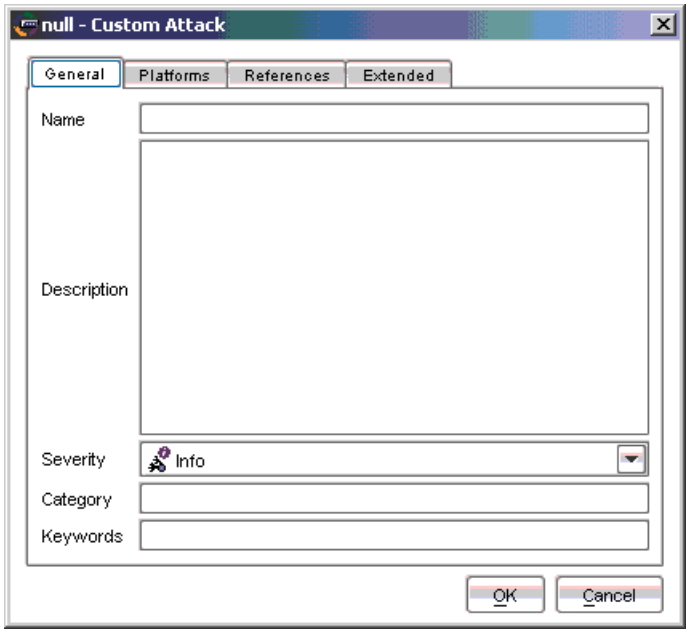
Creating Custom IDP Attack Objects

You can create custom attack objects to detect new attacks or otherwise meet the unique needs of your network.

To create a custom attack object, perform the following steps:

1. In the Object Manager, click **Attack Objects > IDP Objects**. The IDP Objects dialog box appears.
2. Click the **Custom Attacks** tab.
3. Click the **Add** icon. The Custom Attack dialog box appears with the General tab selected.

Figure 179: Custom Attack Dialog Box



a. Enter a name for the attack. The name is used to display the attack object in the UI. You might want to include the protocol the attack uses as part of the attack name.

b. Enter a description for the attack. The description provides details about the attack. Entering a description is optional when creating a new attack object, but it can help you remember important information about the attack. View the attack descriptions for predefined attacks for examples. To display details about a predefined attacks, see "Viewing Predefined Attacks" on page 665.

c. Select a severity for this attack: Info, Warning, Minor, Major, or Critical. Critical attacks are the most dangerous—typically these attacks attempt to crash your server or gain control of your network. Informational attacks are the least dangerous and typically are used by network administrators to discover holes in their own security system.

d. Enter a category for this attack. You can use a predefined category or define a new category

e. Enter one or more keywords for this attack that can help you find it later. A keyword is a unique identifier used to display the attack object in log records. Keywords indicate the important words that relate to the attack and the attack object.

f. Check the check box if you want this attack object to be part of your highest-risk set of attack objects. Later, when you add this attack object to dynamic groups, you can specify whether only recommended attack objects will be included. For more information about recommended attack objects, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

4. Click the Platforms tab in the Custom Attack dialog box to specify the security platform on which the attack detection is to occur.

   a. Click the Add icon. The Custom Attack dialog box appears.

   b. Select the platform on which the attack detection is to occur.

   c. Select the type of attack—Signature, Protocol Anomaly, or Compound Attack—then click **Next**.

If you are configuring a new attack object, the attack object editor leads you through the screens to configure the specific type of attack:

- For signature attack objects, see the following section, "Creating a Signature Attack Object" on page 668.
- For protocol anomaly attack objects, see "Protocol Anomaly Attack Objects" on page 664.
- For compound attack objects, see "Compound Attack Objects" on page 664.

Creating a Signature Attack Object

To configure a signature attack in the Custom Attack dialog box, perform the following steps:

1. Configure general parameters for the attack perform the following steps:
   - **False-Positives** indicates the frequency (Unknown, Rarely, Occasionally, Frequently) that the attack object produces a false positive when used in a security policy. By default, all compound attack objects are set to Unknown, as you fine tune your IDP system to your network traffic, you can change this setting to help you track false positives.
   - **Service Binding** allows you to select a protocol that the attack uses to enter your network. Depending upon the protocol you select, additional fields might appear. You can select the following protocol types:
     - **Any** allows IDP to match the signature in all services (attacks can use multiple services to attack your network).
     - **IP** (specify protocol number) allows IDP to match the signature for a specified IP protocol type.
     - **TCP** (specify port ranges) allows IDP to match the signature for specified TCP port(s).
     - **UDP** (specify port ranges) allows IDP to match the signature for specified UDP port(s).
     - **ICMP** (specify ID) allows IDP to match the signature for specified ICMP ID.
     - **RPC** (specify program number) allows IDP to match the signature for a specified remote procedure call program number.
     - **Service** (specify service) allows IDP to match the signature for a specified service.
   - **Time Binding** allows IDP to detect a sequence of the same attacks over a period of time. If you select **Time Binding**, you can specify the following attributes which are bound to the attack object for one minute:
     - **Scope** specifies whether the counting of the attack is from the same source IP address, the same destination IP address, or a peer. If you select **Source**, IDP detects attacks from a given source IP address for the specified number of times, regardless of the destination IP address. If you select **Destination**, IDP detects attacks to a given destination IP address for the specified number of times, regardless of the source IP address. If you select **Peer**, IDP detects attacks between source and destination IP addresses of the sessions for the specified number of times.
     - **Count** specifies the number of times that IDP detects the attack within the specified scope before triggering an event.
2. Click **Next**.
3. Configure detection parameters for the signature attack:

- The attack pattern is the signature of the attack you want to detect. The signature is a pattern that always exists within an attack; if the attack is present, so is the signature. When creating a new signature attack object, you must analyze the attack to detect a pattern (such as a segment of code, a URL, or a value in a packet header) and then use that pattern to create a signature. You can also negate an entered pattern.

  Table 70 on page 670 shows the regular expressions you can use in the attack pattern:

Table 70: Attack Pattern Expressions

Regular Expression	Description
\0 < *octal-number* >	For a direct binary match
\X < *hexadecimal-number* > \X	For a direct binary match
\[< *character-set* > \]	For case-insensitive matches
.	To match any symbol
*	To match 0 or more symbols
+	To match 1 or more symbols
?	To match 0 or 1 symbols
( )	Grouping of expressions
|	Denotes alternate, typically used with ( )
[< *start* > - < *end* >]	Character range
[^ < *start* > - < *end* >]	Negation of range

- **Offset** is the starting place from the specified service context where IDP should look for the attack. If there is no offset, you can specify **None**; otherwise, you can specify a decimal value.
- **Context** defines the location where IDP should look for the attack in a specific Application Layer protocol. When creating a signature attack object, you should choose a service context, if possible. Because the service context is very specific, your chances of detecting a false positive are greatly reduced. However, choosing a service context overrides any protocol you previously specified in the Service Binding general parameter.

  Table 71 on page 671 lists the service contexts you can use for the attacks.

Table 71: Service Context for Signature Attacks

Service	Description	RFC
AIM	AOL Instant Messenger	
DHCP	Dynamic Host Configuration Protocol	2131, 2132
DNS	Domain Name System	1034, 1035
Finger	Finger Information Protocol	1128
FTP	File Transfer Protocol	959
Gnutella	Gnutella	
Gopher	Gopher	1436
HTTP	Hypertext Transfer Protocol	2616
IMAP	Internet Message Access Protocol	2060
IRC	Internet Relay Chat	2810, 2811, 2812, 2813
LDAP	Lightweight Directory Access Protocol	2251, 2252, 2253, 3377
LPR	Line Printer Protocol	1179
MSN	Microsoft Instant Messenger	
NBNAME/ NBDS	NetBios Name Service	1001, 1002
NFS	Network File System	
NNTP	Network News Transfer Protocol	977
NTP	Network Time Protocol	1305
POP3	Post Office Protocol, version 3	1081
RADIUS	Remote Authentication Dial-In User Service	2865, 2866, 2867, 2868, 3575
REXEC		
RLOGIN	Remote Login (rlogin)	1258, 1282
RSH	Remote Shell (rsh)	
RUSERS		
SMB	Server Message Block	
SMTP	Simple Mail Transfer Protocol	821
SNMP	Simple Network Management Protocol	1067

Table 71: Service Context for Signature Attacks *(continued)*

Service	Description	RFC
SNMPTRAP	SNMP trap	1067
SSH	Secure Shell	Proprietary
SSL	Secure Sockets Layer	
Telnet	Telnet TCP protocol	854
TFTP	Trivial File Transfer Protocol	783
VNC	Virtual Network Computing	
YMSG	Yahoo! Messenger	

- **Direction** defines the connection direction of the attack:
  - **Client to Server** detects the attack only in client-to-server traffic.
  - **Server to Client** detects the attack only in server-to-client traffic.
  - **Any** detects the attack in either direction.
- **Flow** defines the connection flow of the attack: Control, Auxiliary, or Both.

4. Click **Next**.
5. Configure the header match information for the signature attack. You can configure header match information for Internet Protocol version 4 and version 6 (IPv4 and IPv6), Transmission Control protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP) and ICMP version 6 (ICMPv6). The header match configuration allows you to specify that IDP search a packet for a pattern match only for certain header information for the following protocols:
   - Internet Protocol (IPv4)
     - **Type-of-service**. Specify an operand (none, =, !, >, <) and a decimal value.
     - **Total length**. Specify an operand (none, =, !, >, <) and a decimal value.
     - **ID**. Specify an operand (none, =, !, >, <) and a decimal value.
     - **Time-to-live**. Specify an operand (none, =, !, >, <) and a decimal value.
     - **Protocol**. Specify an operand (none, =, !, >, <) and a decimal value.
     - **Source**. Specify the IP address of the attacking device.
     - **Destination**. Specify the IP address of the attack target.

- **Reserved Bi**t. Specifies that IDP looks for a pattern match whether or not the IP flag is set (**none**), only if the flag is set (**set**), or only if the flag is not set (**unset**).
- **More Fragments**. Specifies that IDP looks for a pattern match whether or not the IP flag is set (**none**), only if the flag is set (**set**), or only if the flag is not set (**unset**).
- **Don't Fragment**. Specifies that IDP looks for a pattern match whether or not the IP flag is set (**none**), only if the flag is set (**set**), or only if the flag is not set (**unset**).

- IPv6
  - Configure the IPv6 header match information. See "Internet Protocol (IPv4)" for information on the different parameters.

- Transmission Control Protocol (TCP)
  - **Source Port**. The port number on the attacking device.
  - **Destination Port**. The port number of the attack target.
  - **Sequence Number**. The sequence number of the packet. This number identifies the location of the data in relation to the entire data sequence.
  - **ACK Number**. The ACK number of the packet. This number identifies the next sequence number; the ACK flag must be set to activate this field.
  - **Header Length**. The number of bytes in the TCP header.
  - **Window Size**. The number of bytes in the TCP window size.
  - **Urgent Pointer**. Indicates that the data in the packet is urgent; the URG flag must be set to activate this field.
  - **Data Length**. The number of bytes in the data payload. For SYN, ACK, and FIN packets, this field should be empty.

    You can also specify the following TCP flag options as **none**, **set**, or **unset**:
    - **URG**. When set, the urgent flag indicates that the packet data is urgent.
    - **ACK**. When set, the acknowledgment flag acknowledges receipt of a packet.
    - **PSH**. When set, the push flag indicates that the receiver should push all data in the current sequence to the destination application (identified by the port number) without waiting for the remaining packets in the sequence.

        - **RST**. When set, the reset flag resets the TCP connection, discarding all packets in an existing sequence.
        - **FIN**. When set, the final flag indicates that the packet transfer is complete and the connection can be closed.

    - User Datagram Protocol (UDP)
        - **Source Port**. The port number on the attacking device. Specify an operand (none, =, !, >, <) and a decimal value.
        - **Destination Port**. The port number of the attack target. Specify an operand (none, =, !, >, <) and a decimal value.
        - **Data Length**. The number of bytes in the data payload. Specify an operand (none, =, !, >, <) and a decimal value.
    - Internet Control Message Protocol
        - **ICMP Type**. The primary code that identifies the function of the request/reply.
        - **ICMP Code**. The secondary code that identifies the function of the request/reply within a given type.
        - **Sequence Number**. The sequence number of the packet. This number identifies the location of the request/reply in relation to the entire sequence.
        - **ICMP ID**. The identification number is a unique value used by the destination system to associate requests and replies.
        - **Data Length**. The number of bytes in the data payload.
    - ICMPv6
        - Configure the ICMPv6 header match information. See "Internet Control Message Protocol" for information on configuring the protocol parameters.

6. Click **Finish**.

Creating a Protocol Anomaly Attack

Perform the following steps to configure a protocol anomaly attack in the Custom Attack dialog box:

1. Configure general parameters for the attack:
    - **False-Positives** indicates the frequency (**Unknown**, **Rarely**, **Occasionally**, **Frequently**) that the attack object produces a false positive when used in a security policy. As you finetune your IDP system to your network traffic, you can change this setting to help you track false positives.

- **Anomaly** allows you to select a protocol anomaly from a list of known protocol anomalies. NSM detects anomalies for the following protocols:

AIM	DHCP	IDENT	RUSERS	TFTP
FINGER	CHARGEN	IMAP	Gnutella	RLOGIN
FTP	DISCARD	IP Packet	Gopher	RPC
HTTP	DNS	POP3	IRC	RSH
ICMP	ECHO	REXEC	MSN	RTSP
MSN	LPR	NFS	VNC	NNTP
SNMP	SMTP	SMB	SNMP TRAP	YMSG
TCP segment	SYSLOG	SSH	TELNET	

- **Time Binding** allows IDP to detect a sequence of the same attacks over a specified period. If you select **Time Binding**, you can specify the following attributes that are bound to the attack object for one minute:
  - **Scope** specifies whether the counting of the attack is from the same source IP address, the same destination IP address, or a peer. If you select **Source**, IDP detects attacks from a given source IP address for the specified number of times, regardless of the destination IP address. If you select **Destination**, IDP detects attacks to a given destination IP address for the specified number of times, regardless of the source IP address. If you select **Peer**, IDP detects attacks between source and destination IP addresses of the sessions for the specified number of times.
  - **Count** specifies the number of times that IDP detects the attack within the specified scope before an event is triggered.

2. Click **Finish**.

NOTE: For ICMP, a predefined protocol anomaly action generates a log if the number of time-exceed ICMP messages exceeds 4. This action also supports ICMPv6 packets.

Creating a Compound Attack

Note the following when creating a custom compound attack object:

- All members of the compound attack object must use the same service setting or service binding, such as FTP, Telnet, YMSG, TCP/80, and so on.
- You cannot add predefined or custom signature attack objects to a compound attack object. Instead, you specify the signature directly within the compound attack object, including such details as service (or service binding), service context,

attack pattern, and direction. You can add protocol anomaly attack objects to a compound attack object.

- You can add between 2 and 32 protocol anomaly attack objects and/or signatures as members of the compound attack object. However, all members must use the same service setting or service binding.

Perform the following steps to configure a compound attack in the Custom Attack dialog box:

1. Configure general parameters for the attack:
   - **False-Positives** indicates the frequency (**Unknown**, **Rarely**, **Occasionally**, **Frequently**) that the attack object produces a false positive when used in a security policy. By default, all compound attack objects are set to **Unknown**. As you finetune IDP to your network traffic, you can change this setting to help you track false positives.
   - **Service Binding** allows you to select a protocol that the attack uses to enter your network. Depending upon the protocol you select, additional fields may appear. You can select the following protocol types:
     - **Any** allows IDP to match the signature in all services (attacks can use multiple services to attack your network).
     - **IP** (specify protocol number) allows IDP to match the signature in a specified IP protocol type.
     - **TCP** (specify port ranges) allows IDP to match the signature for specified TCP port(s).
     - **UDP** (specify port ranges) allows IDP to match the signature for specified UDP port(s).
     - **ICMP** (specify ID) allows IDP to match the signature for specified ICMP ID.
     - **RPC** (specify program number) allows IDP to match the signature for a specified remote procedure call program number.
     - **Service** (specify service) allows IDP to match the signature for a specified service.
   - **Time Binding** allows IDP to detect a sequence of the same attacks over a specified period. If you select **Time Binding**, you can specify the following attributes that are bound to the attack object for one minute:
     - **Scope** specifies whether the counting of the attack is from the same source IP address, the same destination IP address, or a peer. If you select **Source**, IDP detects attacks from a given source IP address for the specified number of times, regardless of the destination IP address. If you select **Destination**, IDP detects attacks to a given destination IP address for the specified number of times, regardless of the source IP address. If you select **Peer**, IDP detects attacks between source and destination IP addresses of the sessions for the specified number of times.

- **Count** specifies the number of times that IDP detects the attack within the specified scope before an event is triggered.

2. Click **Next**.
3. Perform the following steps to configure the compound attack members:
   - **Scope** specifies whether the match should occur over a single session or can be made across multiple transactions within a session:
     - Select **Session** to allow multiple matches for the object within the same session.
     - Select **Transaction** to match the object across multiple transactions that occur within the same session.
   - Select **Reset** if the compound attack should be matched more than once within a single session or transaction. If **Reset** is selected, multiple matches can be made within a single session or transaction.
   - Select **Ordered Match** to create a compound attack object that must match each member signature or protocol anomaly in the order you specify. If you do not specify an ordered match, the compound attack object still must match all members, but the attacks or protocol anomalies can appear in random order.

     You can now add signature or protocol anomaly attack objects to the compound attack, as described in the following sections.

Adding a Signature to the Compound Attack Object

1. To add an attack pattern to the compound attack object, click the Add icon, then select **Signature**. The New Member dialog box appears.
2. Double-click the newly created signature member of the compound attack object. Configure the attack pattern settings:
   - **DFA Pattern**. Specify the pattern IDP should match. You construct the attack pattern just as you would when creating a new signature attack object.

     To exclude the specified pattern from being matched, select the Negate check box.
   - **Context**. Specify the context in which the IDP should look for the pattern. The context displays only contexts that are appropriate for the specified service. If you selected a service binding of **any**, you are restricted to the service contexts packet and first packet.
   - **Direction**. Specify whether IDP should match the pattern in traffic flowing in any direction, from client-to-server, or from server-to-client.

     Examine the traffic before you determine the direction. We recommend client-to-server direction for better performance. There is a performance hit on the device if you select server-to-client and the risk of attack objects is lower with client-to-server.
3. Click **OK**.

Adding a Protocol Anomaly to the Compound Attack Object

1. To add a protocol anomaly to the compound attack object, click the Add icon, then select **Anomaly**. The New Member dialog box appears.
2. Select an anomaly.
3. Click **OK**.

Deleting a Member from the Compound Attack Object

To remove a member signature or an anomaly, select the member in the list, then click the Delete icon. A confirmation window asks you to verify that you want to delete the item. Click **OK**.

Editing a Custom Attack Object

To modify a custom attack object, double-click the object in the Custom Attacks tab in the IDP Objects dialog box. The Custom Attacks dialog box appears with the previously configured information in the General and Platforms tabs. You can enter optional information in the References and Extended tabs. Enter any changes you want to make, then click **Apply**. To close the dialog box, click **OK**.

Deleting a Custom Attack Object

To delete a custom attack object, right-click the object in the Custom Attacks tab in the IDP Objects dialog box, then select **Delete**. A confirmation window asks you to verify that you want to delete the item. Click **OK**.

Creating Custom IDP Attack Groups

The IDP system contains hundreds of predefined attack objects, and you can create additional custom attack objects. When you create your security policy rules, you can add attack objects individually or by the predefined or the custom attack group. To help keep your security policies organized, you can organize attack objects into groups.

You can create *static groups*, which contain only the groups or attack objects you specify, or *dynamic groups*, which contain attack objects based on criteria you specify.

Configuring Static Groups

A static group contains a specific, finite set of attack objects or groups. There are two types of static groups: *predefined* static groups and *custom* static groups.

A predefined static group can include the following members:

- Predefined attack objects
- Predefined static groups
- Predefined dynamic groups

A custom static group can include the same members as a predefined static group, plus the following members:

- Custom attack objects
- Custom dynamic groups
- Other custom static groups

You use static groups to do the following:

- Define a specific set of attacks to which you know your network is vulnerable
- Group custom attack objects
- Define a specific set of informational attack objects that you use to keep you aware of what is happening on your network

Static groups require more maintenance than dynamic groups because you must manually add or remove attack objects in a static group in order to change the members. However, you can include a dynamic group within a static group to automatically update some attack objects. For example, the predefined attack object group Operating System is a static group that contains four predefined static groups: BSD, Linux, Solaris, and Windows. The BSD group contains the predefined dynamic group BSD-Services-Critical, to which attack objects can be added during an attack database update.

To create a custom static group:

1. In Object Manager, click **Attack Objects > IDP Objects**. The IDP Objects dialog box appears.
2. Click the Custom Attack Groups tab.
3. Click the Add icon, then select **Add Static Group**. The New Static Group dialog box appears.
4. Enter a name and description for the static group. Select a color for the group icon.
5. To add an attack or a group to the static group, select the attack or group from the Attacks/Group list, then click **Add**.
6. Click **OK**.

Configuring Dynamic Groups

A dynamic group contains a dynamic set of attack objects that are automatically added or deleted based on specified criteria for the group. For example, an attack database update can add or remove attack objects from a dynamic group based on the group criteria. This eliminates the need for you to review each new signature to determine if you need to use it in your existing security policy.

A predefined or custom dynamic group can only contain attack objects, not attack groups. Dynamic group members can be either predefined or custom attack objects.

Perform the following steps to create a custom dynamic group:

1. In the Object Manager, click **Attack Objects > IDP Objects**. The IDP Objects dialog box appears.
2. Click the Custom Attack Groups tab.
3. Click the Add icon, then select **Add Dynamic Group**. The New Dynamic Group dialog box appears.
4. Enter a name and description for the static group. Select a color for the group icon.
5. In the Filters tab, click the Add icon, then select one of the following:
   - **Add Products Filter** adds attack objects based on the application that is vulnerable to the attack
   - **Add Severity Filter** adds attack objects based on the attack severity.

NOTE: We assign all predefined attack objects a severity level. However, you can edit this setting to match the needs of your network.

   - **Add Category Filter** adds attack objects based on category
   - **Add Last Modified Filter** adds attack objects based on their last modification date
   - **Add Recommended** Filter includes only attacks designated to be the most serious threats to the dynamic group. In the future, Juniper Networks will designate only attacks it considers to be serious threats as recommended. These settings will be updated with new attack object updates. In addition, you can designate custom attack objects as recommended or not. For more information about recommended actions, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

You create filters one at a time; as you add each criteria, IDP compares it to the attributes for each attack object and immediately filters out any attack object that does not match. If you create a filter with attributes that no attack object can match, a message appears warning you that your dynamic group has no members.

From the resulting list of matching attack objects, you can then exclude any attack objects that produce false positives on your network or that detect an attack to which your network is not vulnerable.

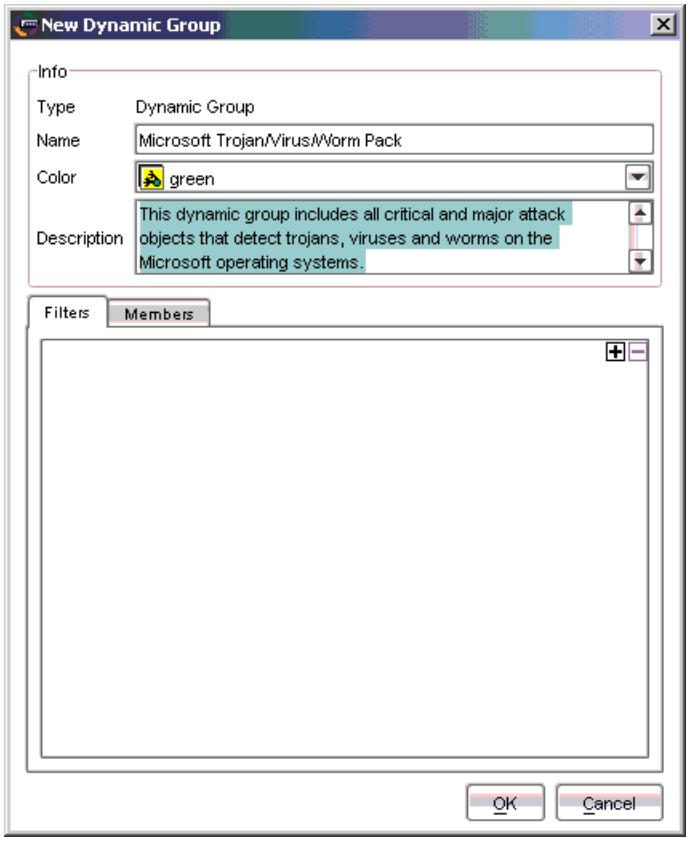
NOTE: A dynamic group cannot contain another group (predefined, static, or dynamic). However, you can include a dynamic group as a member of a static group.

Example: Creating a Dynamic Group

Perform the following steps to create a dynamic group:

1. In the Custom Attack Groups tab, click the Add icon, then select **Add Dynamic Group**. The New Dynamic Group dialog box appears.
2. Enter a name and description for the group. Select a color for the group icon.

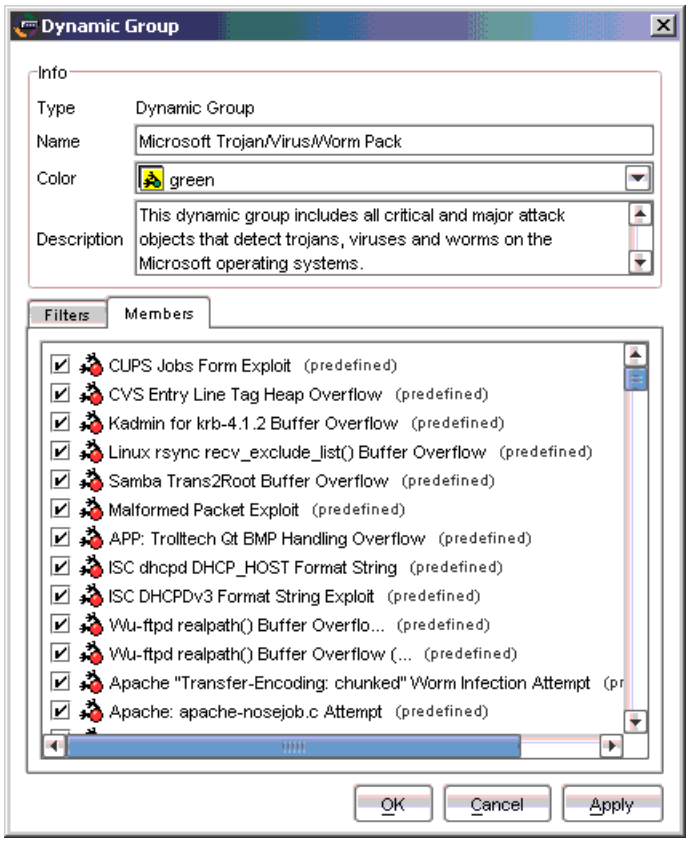
Figure 180: New Dynamic Group



3. In the Filters tab, click the Add icon, then add the filters that determine which attack objects should be in the group:
   a. Add a Products filter to add attack objects that detect attacks against all Microsoft Windows operating systems.
   b. Add a Severity filter to add attack objects that have a severity level of Critical or Major.

      IDP automatically applies all filters to the entire attack object database, identifies the attack objects that meet the defined criteria, and adds the matching objects as members of the group.
4. View the members of the group by clicking on the Members tab as shown in Figure 181 on page 682:

Figure 181: New Dynamic Group Members



5. Click **OK** to save the dynamic group.

Updating Dynamic Groups

When you are satisfied with the group criteria and its members, use the group in a security policy. The next time you update your attack objects, the following tasks are performed automatically:

- For all new attack objects, the update compares the predefined attributes of each attack object to each dynamic group criteria and adds the attack objects that match.
- For all updated attack objects, the update removes attack objects that no longer meet their dynamic group criteria. The update also reviews updated attack objects to determine if they now meet any other dynamic group criteria and adds them to those groups as necessary.
- For all deleted attack objects, the update removes the attack objects from their dynamic groups.

You can also edit a dynamic group manually, adding new filters or adjusting existing filters to get the type of attack objects you want. You can also edit a dynamic group from within a security policy (see “Configuring Security Policies” on page 626).

Editing a Custom Attack Group

To modify a custom attack group, double-click the group in the Custom Attack Groups tab in the IDP Objects dialog box. The Static Group or Dynamic Group dialog box appears with the previously configured information displayed. Enter any changes you want to make, then click **Apply**; to close the dialog box, click **OK**.

Deleting a Custom Attack Group

To delete a custom attack group, right-click the group in the Custom Attack Groups tab in the IDP Objects dialog box, then select **Delete**. A confirmation window asks you to verify that you want to delete the item. Click **OK**.

Configuring the Device as a Standalone IDP Device

You can deploy the IDP-capable device as a standalone IDP security system protecting critical segments of your private network. For example, you might already have a security device actively screening traffic between the Internet and your private network (some devices can optionally use Deep Inspection to inspect this traffic). But you still need to protect internal systems, such as mail servers, from attacks that might originate from user machines in an otherwise trusted network. In this case, you need a security system that provides IDP instead of firewall functions.

This section describes how to configure the security device to provide standalone IDP functions.

NOTE: Juniper Networks offers standalone IDP appliances that provide IDP functionality without integrated firewall/VPN capabilities. You can use the NSM system to manage these appliances as well as IDP-capable firewall/VPN devices.

Enabling IDP

To enable IDP, you need to configure a firewall rule in a security policy that directs traffic between the applicable zones to be checked against IDP rulebases. You can make this firewall rule very simple in that it can match all traffic from all sources to all destinations for all services.

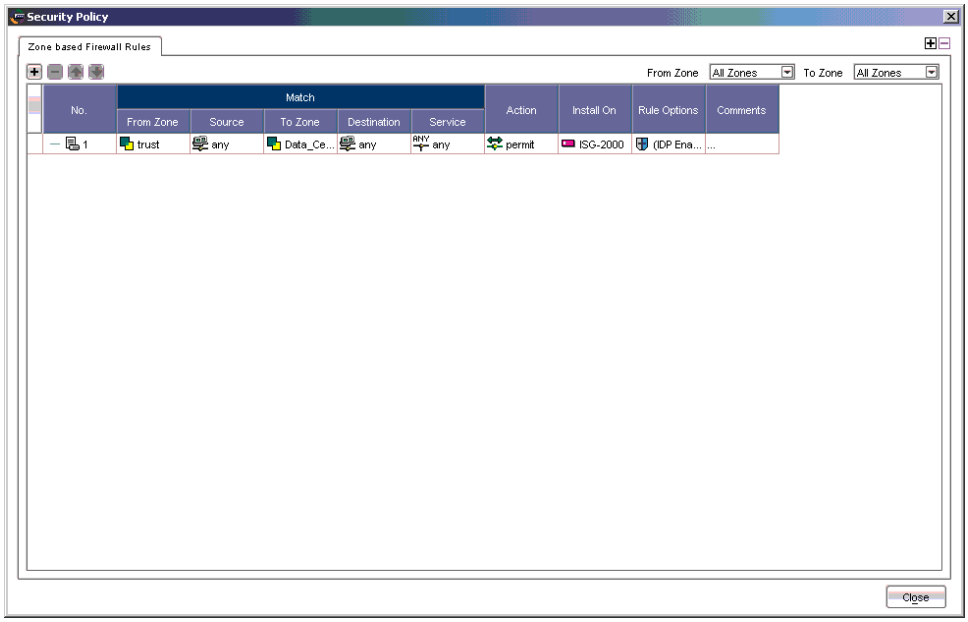
1. Create a firewall rule that permits traffic from any source to any destination for any service.
2. Right-click in the Rule Options column for the firewall rule, then select **DI Profile/Enable IDP**.
3. In the DI Profile/Enable IDP dialog box, click the button to enable IDP, then select **OK**.
4. Configure IDP rules, creating IDP rulebases as needed.

For more information about configuring security policies that include IDP rules, see “Configuring Security Policies” on page 626.

Example: Configuring a Firewall Rule for Standalone IDP

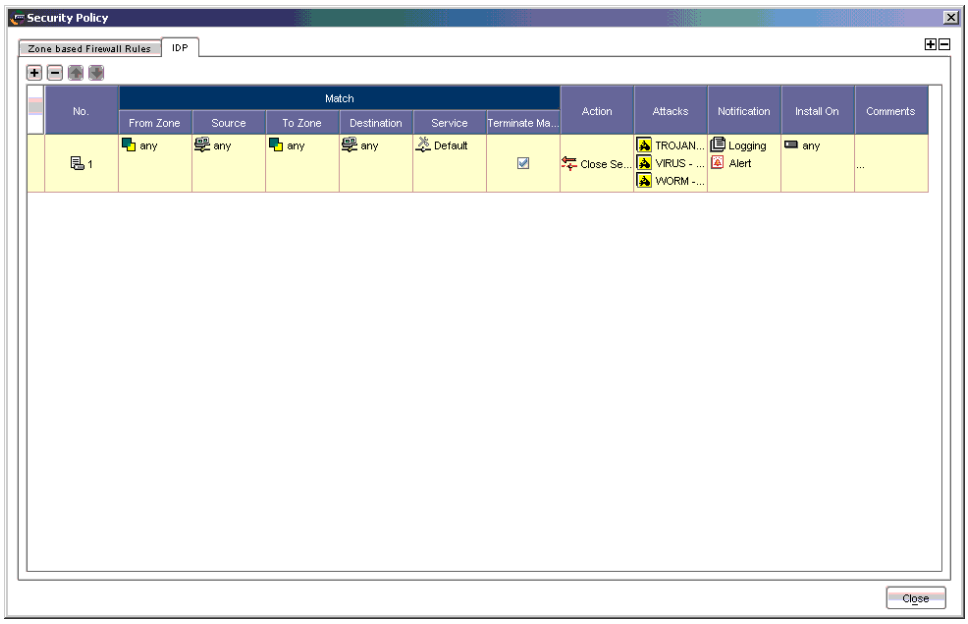
In this example, you are deploying an IDP/firewall/VPN device as a standalone IDP security system between the Trust zone and the custom Data_Center zone in your network. Your company's file, mail, and database servers reside in the Data_Center zone. While you want to allow users in the Trust zone to be able to access the servers in the Data_Center zone, you also need to protect the servers from attacks that inadvertently might have been introduced into a user machine in the Trust zone. You create a firewall rule from the Trust to the Data_Center zone that allows traffic from any source to any destination for any service, then enable IDP in the Rule Options column, as shown in Figure 182 on page 684.

Figure 182: Firewall Rule for Standalone IDP



You would then add and configure IDP rulebases for the security policy to detect possible attacks against servers in the Data_Center zone, as shown in Figure 183 on page 685.

Figure 183: IDP Rules for Standalone IDP



Configuring Role-Based Administration

NSM's role-based administration (RBA) allows the super administrator (superadmin) to create a custom role and administrator for the standalone IDP device. This gives the IDP administrator permission to perform only those tasks that are specific to configuring and administering IDP functions; the IDP administrator does not need to create, edit, delete, view, or update device configurations. When the IDP administrator logs into the NSM UI, he or she only sees the menus and options that are applicable to IDP.

Example: Configuring an IDP-Only Administrator

In this example, you (the superadmin) create a custom role and an IDP administrator who can only perform tasks that are specific to configuring and administering IDP on the standalone IDP device.

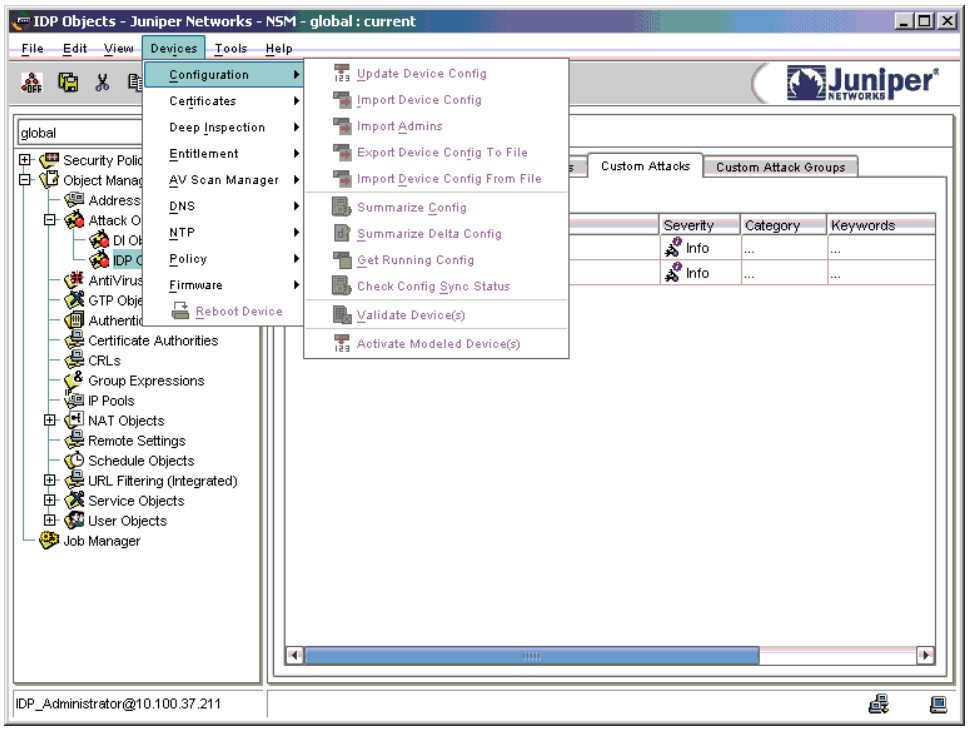
1. Log into the global domain as the superadmin. From the menu bar, select **Tools > Manage Administrators and Domains**.
2. Click the Roles tab, then click the Add icon to create a role called **IDP_Only**. Select tasks that are specific for IDP configuration and administration, such as:
   - Attack Update
   - Create/View/Edit/Delete Policies
   - Create/View/Edit/Delete Backdoor and IDP Rulebases
   - View Firewall Rulebases
   - Create/Edit/Delete Shared Objects and Groups

Select any other tasks that might be helpful for the IDP administrator; for example, you can select the options to view Jobs and the System Status Monitor.

3. Click **OK** in the New Role dialog box to return to the Manage Administrators and Domains dialog box.
4. Click the Administrators tab, then click the Add icon to create an administrator called **IDP_Administrator**. The New Admin dialog box appears with the General tab selected.
5. In the Name field, enter **IDP_Administrator**. You can enter contact information for the administrator.
6. Click the Authorization tab. Select the authorization method and the local password for the administrator.
7. Click the Permissions tab, then click the Add icon to select the role **IDP_Only** for this administrator.
8. Click **OK** to close the New Select Role and Domains dialog box. Click **OK** to close the New Admin dialog box. Click **OK** to close the Manage Administrators and Domains dialog box.

The administrator for the standalone IDP device can now log into NSM as **IDP_Administrator**. Upon login, the NSM UI displays a limited navigation tree and menu options for this user, as shown in Figure 184 on page 687. Note that the UI displays only the security policy and Object Manager options in the navigation tree; the Devices > Configuration options are not available for this user.

Figure 184: UI Display for IDP_Administrator



Managing IDP

This section describes IDP management on the IDP-capable device.

About Attack Database Updates

Juniper Networks periodically provides attack database updates, in the form of a download file, on the Juniper Web site. Attack database updates can include the following:

- New or modified predefined IDP attack objects and groups
- New or modified signatures used by the Deep Inspection (DI) feature
- Updates to the IDP engine, which runs in the security device

In a new attack database update, the version number increments by 1. When you download a version of an attack database update from the Juniper Networks website, NSM stores the version number of the attack database update. You can check to see if there is a more recent update available than the last one you downloaded.

Downloading Attack Database Updates

The attack database updates are downloaded to the NSM GUI server. Perform the following steps to download an attack database update file:

1. From the menu bar, select **Tools > View/Update NSM Attack Database**. The Attack Update Manager wizard appears.
2. Follow the instructions in the Attack Update Manager to download the attack database update file to the NSM GUI server.

NOTE: The Juniper Networks website is set by default in the New Preferences dialog box, which you access by selecting **Tools > Preferences**. The GUI Server must have Internet access.

Using Updated Attack Objects

You cannot create, edit, or delete predefined IDP attack objects or groups, but you can update the attack object database installed in the NSM GUI server. Updates to predefined IDP attack objects and groups can include the following:

- New descriptions or severities for existing attack objects
- New attack objects or groups
- Deletion of obsolete attack objects

When you download updated IDP attack objects and groups to the GUI server, any new attack objects in the update are available for selection in an IDP rulebase in a security policy. When you install a security policy on your managed device, only the attack objects that are used in IDP rules for the device are pushed from the GUI server to the device.

NOTE: For the DI feature, all updated signatures are pushed to your managed device. For more information about updating the attack object database for DI on your managed device, see the Network and Security Manager documentation at http://www.juniper.net/techpubs/software/management/security-manager.

Updating the IDP Engine

The IDP engine is dynamically changeable firmware that runs on the firewall/VPN device. There are two ways that the IDP engine can be updated on the device:

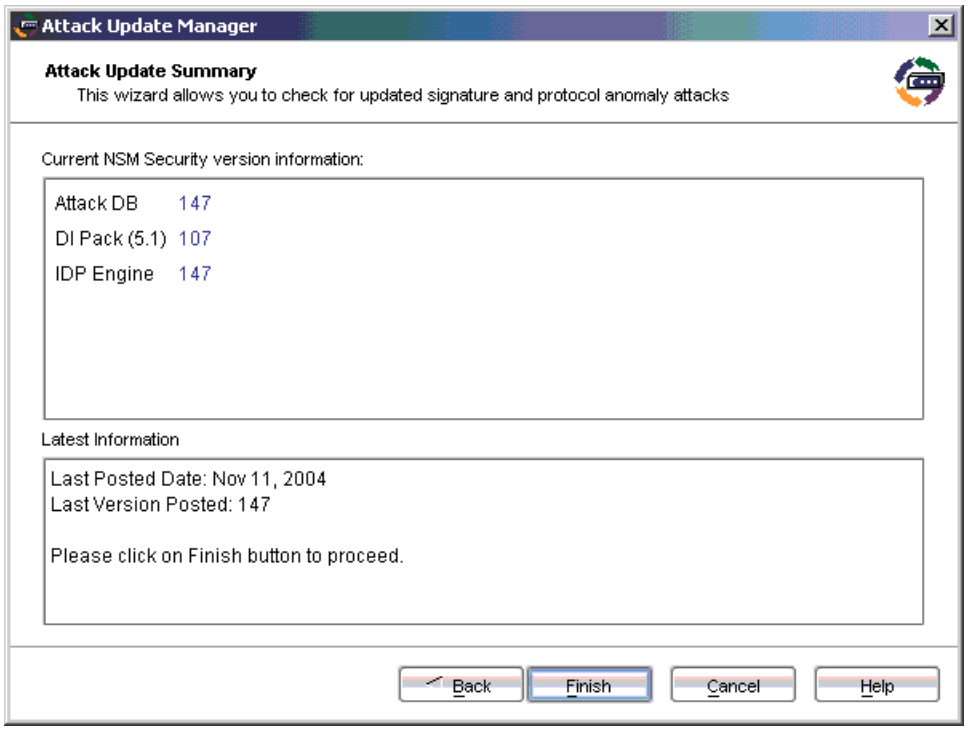
- When you upgrade the firmware on an IDP/firewall/VPN device, the upgraded firmware will typically include a recent version of the IDP engine as well as a new version of ScreenOS. (For information about upgrading the firmware on a security device, see the Network and Security Manager documentation athttp://www.juniper.net/techpubs/software/management/security-manager.
- You can update the IDP engine on a managed device from an attack database update on the GUI server. Because attack database updates are available more often than firmware releases, an attack database update may include a more recent version of the IDP engine than is available on the latest firmware release. For example, an attack database update might contain updated IDP attack objects that can only be used with an updated version of the IDP engine.

Perform the following steps to see the version of the IDP engine that is currently running on the device:

1. Select **Tools > View/Update NSM Attack Database**. The Attack Update Manager wizard appears.
2. Click **Next**.

   The Attack Update Summary, as shown in Figure 185 on page 689, displays information about the current version downloaded on the GUI server and the latest version available from Juniper Networks.

Figure 185: Attack Update Summary



3. Click **Finish** to continue downloading the latest attack database update, or click **Cancel** to exit the Attack Update Manager.

To update the IDP engine on the device:

1. Select **Devices > IDP Detector Engine > IDP Detector Engine**. The Change Device Sigpack dialog box appears.

NOTE: The IDP engine version you install on the security device must be compatible with the version of the firmware that is running in the device. You cannot downgrade the IDP engine version on the device.

2. Click **Next**, then select the managed devices on which you want to install the IDP engine update.
3. Follow the instructions in the Change Device Manager to update the IDP engine on the selected device.

NOTE: Updating the IDP engine on a device does not require a reboot of the device.

Viewing IDP Logs

When attack objects are matched in an IDP rule, IDP log entries appear in the NSM Log Viewer. Perform the following steps to receive IDP log entries in the Log Viewer:

NOTE: The Log Viewer supports both IPv4 and IPv6 addresses.

1. Enable the device to send log entries with the desired severity settings to NSM:
   a. In Device Manager, open the device configuration for the device.
   b. In the device navigation tree, select **Report Settings > General > NSM**.
   c. Select the severity settings you want logged to NSM.
   d. Click **OK**.
2. Enable IDP detection and logging in the security policy installed on the device. For detailed information about configuring IDP logging in the security policy, see "Configuring Security Policies" on page 626.

   IDP alarm log entries appear in the Log Viewer and display the following columns of information:
   - Source and Destination Address
   - Action
   - Protocol
   - Category (Anomaly, Custom, or Signature)
   - Subcategory
   - Severity
   - Device

ISG-IDP Devices

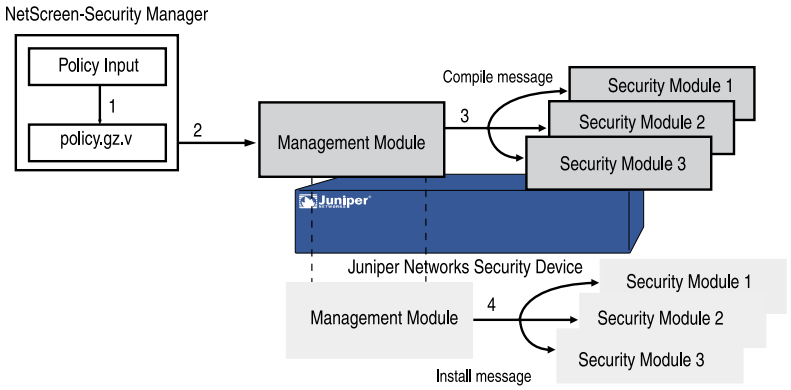
Juniper Networks Integrated Security Gateway–Intrusion Detection and Prevention (ISG-IDP) devices protect your networks from traffic anomalies and malicious attacks. You can configure IDP policies for the traffic flowing through your network based

on your network's security requirements. The ISG-IDP device loads the IDP policy data and applies the policy to the network traffic.

Compiling a Policy

You use NSM to remotely manage ISG-IDP devices. NSM compiles the policies you define and transfers them to the ISG-IDP device. The following section explains the steps involved in compiling and loading a policy on the ISG-IDP device. Figure 186 on page 691 illustrates an ISG-IDP policy compilation.

Figure 186: ISG-IDP Policy Compilation



1. NSM compiles the policy you defined and generates a policy.gz.v file.
2. NSM transfers the policy.gz.v file to the management module of the ISG-IDP device.
3. The management module sends a compile message along with the policy.gz.v file to each ISG-IDP security module.
4. After all the security modules compile the policy file, the management module sends an install message to each security module.

The management module sends the policy to the security module and waits for a reply with a timeout of 60 seconds. If the security module does not respond with a reply within the 60 seconds, the management module treats it as a policy push failure and notifies NSM. NSM sends an error message to the user reporting the policy push failure.

Policy Size Multiplier

When the security module attempts to compile the policy.gz.v file but does not have enough memory to do so, the security module runs out of memory and enters an irrecoverable state.

To avoid entering the irrecoverable state, the security module estimates the memory required to compile the policy by multiplying the size of the policy.gz.v file by the configurable parameter **sc_policy_size_multiplier**. If the security module determines that the available memory is less than the estimated memory, the security module does not compile the file and the policy push operation fails.

By default, **sc_policy_size_multiplier** is set to 100. In this case, a security module requires 500 MB of memory to compile a 5 MB file.

You need to configure **sc_policy_size_multiplier** for each security module by using the CLI command:

```
exec sm 2 ksh "scio const set sc_policy_size_multiplier 300"
```

This command **sets sc_policy_size_multiplier** for security module 2 to 300. The security module requires 900 MB of memory to compile a 3 MB file. If the memory available is less than 900 MB, the security module does not compile the file and the policy push fails.

User-Role-Based IDP Policies

Juniper Networks security devices now support user-role-based IDP policies, which administrators can use to define actions on traffic originated by users in specific roles (for example, developer, QA, admin) irrespective of the user's source IP address.

When the user-role-based policy is enabled, the device checks role-based rules first and, if it finds a matching role,it does not try to match IP-based rules. Only when it does not find a matching role will it search an IP-based IDP policy.

You can enable or disable this feature by using the sc_enable_uac_policy parameter.

To enable a role-based policy:

```
exec sm 2 ksh scio const set sc_enable_uac_policy 1
```

However, if this parameter is set to 0, the security device searches only IP-based IDP policies for the new sessions.

To view the list of role names defined in NSM, use the scio policy role socommand:

```
scio policy role so
  For cpu 0
  role id                    role name
       1                        dev
       2                         qa
       3                         ce
  For cpu 1
  role id                    role name
       1                        dev
       2                         qa
       3                         ce
```

For information on enabling this feature using the NSM UI, see "Configuring IDP Rules" on page 631

Unloading Existing Policies

The ISG-IDP device cannot share memory for a new policy when its CPU usage is high for a long time. In such cases, the policy compilation fails by default. You can configure the security module to unload the current active policy whenever memory

is required to load a new policy. Use the **sc_pcomp_unload_cur_on_low_mem** parameter to unload the active policy and load the new policy.

By default, **sc_pcomp_unload_cur_on_low_mem** is set to 0. In this case, the security module does not unload the existing policy when loading a new policy. Therefore, the policy compilation fails if free memory available is insufficient for compiling a new policy.

If you set **sc_pcomp_unload_cur_on_low_mem** to 1, the security module unloads the current active policy when free memory available is insufficient for compiling the new policy. The security module uses the freed memory to compile and load the new policy.

Configure the **sc_pcomp_unload_cur_on _low_mem** parameter with the following CLI command:

```
exec sm 2 ksh "scio const set sc_pcomp_unload_cur_on_low_mem 1"
```

This command sets **sc_pcomp_unload_cur_on_low_mem** of security module 2 to 1.

CPU Usage Monitoring and Event Log

The ISG-IDP security devices provide the following features for allowing users to monitor CPU usage:

- Calculating and viewing CPU usage
- Logging events if CPU utilization parameters exceed the user-defined threshold.

CPU Usage

Users can enable the security device to compute CPU usage by using the **sc_enable_cpu_usage** parameter. If the value is set to 0, the security device does not compute the CPU usage. If the value is set to 1, the security device computes the CPU usage. By default, this value is set to 0.

To set the sc_enable_cpu_usage parameter value to 1

```
exec sm 2 ksh "scio const set sc_enable_cpu_usage 1"
```

This command enables the second security module in the security device to calculate the CPU usage.

When the **sc_enable_cpu_usage** parameter is set to 1, to view the CPU usage for the last 60 seconds, last 60 minutes, and last 24 hours:

```
exec sm num ksh "scio cpu"
```

The output of the command will be in the following format:

```
For cpu 0
Last 60 seconds:
59:  2   58:  2   57:  2   56:  80**   55:  2   54:  2
```

```
53:  2    52:  2    51:  2    50:  2    49:  2    48:  2
47:  2    46:  2    45:  2    44:  2    43:  2    42:  2
41:  2    40:  2    39:  2    38:  2    37:  2    36:  2
35:  2    34:  2    33:  50*    32:  2    31:  2    30:  2
29:  2    28:  2    27:  2    26:  2    25:  2    24:  2
23:  2    22:  2    21:  2    20:  2    19:  2    18:  2
17:  2    16:  2    15:  2    14:  2    13:  2    12:  2
11:  2    10:  2     9:  2     8:  90***     7:  2     6:  2
 5:  2     4:  2     3:  2     2:  2     1:  2     0:  2

Last 60 minutes:
59:  2    58:  2    57:  2    56:  2    55:  2    54:  2
53:  2    52:  2    51:  2    50:  2    49:  2    48:  2
47:  2    46:  2    45:  2    44:  2    43:  2    42:  2
41:  2    40:  2    39:  2    38:  2    37:  2    36:  2
35:  2    34:  2    33:  2    32:  2    31:  2    30:  2
29:  2    28:  2    27:  2    26:  2    25:  2    24:  2
23:  2    22:  2    21:  2    20:  2    19:  2    18:  2
17:  2    16:  2    15:  2    14:  2    13:  2    12:  2
11:  2    10:  2     9:  2     8:  2     7:  2     6:  2
 5:  2     4:  2     3:  2     2:  2     1:  2     0:  2

Last 24 hours:
23:  2    22:  2    21: N/A   20: N/A   19: N/A   18: N/A
17: N/A   16: N/A   15: N/A   14: N/A   13: N/A   12: N/A
11: N/A   10: N/A    9: N/A    8: N/A    7: N/A    6: N/A
 5: N/A    4: N/A    3: N/A    2: N/A    1: N/A    0: N/A
```

"*" indicates that the CPU usage is between 50% and 70%.
"**" Indicates that the CPU usage is between 70% and 85%.
"**"Indicates that the CPU usage is above 85%.

Event Log

The CPU in the ISG-IDP security device informs the management module about the CPU utilization, memory utilization, and session count on the CPU for every 1 minute. If any of these values exceeds the user-defined threshold, the management module generates a log and an SNMP trap.

If the management module does not receive any information from the CPU for 5 minutes, it generates a log and an SNMP trap indicating that the CPU is not responsive.

The management module in the security device generates log messages under the following conditions:

- CPU utilization reaches the user-defined threshold.
- Memory utilization reaches the user-defined threshold.
- Session count per CPU reaches the user-defined threshold.
- CPU is unresponsive.

Use the **sc_log_threshold_memory**, **sc_log_threshold_cpu**, and **sc_log_threshold_session** parameters to set the respective threshold values.

The following table lists default values for the threshold parameters and their configurable range:

Parameter	Definition	Default Threshold Value	Range
sc_log_threshold_memory	Memory utilization by CPU	90	100
sc_log_threshold_cpu	CPU Utilization by the security device	95	0–100
sc_log_threshold_session	Session count per CPU	150000	0–175000

NOTE: When the security device restarts, it sets the threshold to the default value.

To configure the **sc_log_threshold_memory** parameter:

```
exec sm 2 ksh "scio const set sc_log_threshold_memory 100"
```

To configure the **sc_log_threshold_cpu** parameter:

```
exec sm 2 ksh "scio const set sc_log_threshold_cpu 100"
```

To configure the **sc_log_threshold_session** parameter:

```
exec sm 2 ksh "scio const set sc_log_threshold_session 16000"
```

Core dump files

Whenever IDP-related programs (such as engine, pcid, and scio) crash, ScreenOS generates a core dump file for each program and saves it to the /idp/log directory in the RAM of the security device. These core dump files help admin users identify the reason for a crash.

Because RAM is volatile, core dump files might be lost when the security device is turned off. For this reason, the default behavior of the security device is to transfer the core dump files from the security device RAM to the management module flash memory. The transfer is done through a PCI bus. You enable or disable this feature with the sm-ctx coresave command.

To disable this feature:

```
set sm-ctx coresave
```

To enable this feature:

```
unset sm-ctx coresave
```

When the security device executes the idp_run script, a core dump monitor program in the security device starts checking the "/idp/log" directory at one second interval and transfers the core dump files to the management module. The program also changes the status of the core dump files to indicate that they have been transferred.

NOTE: If the core monitoring program in the security module crashes, the core dump files generated after that will not be sent to the management module until the idp_run script is executed.

In the management module, when the core dump files arrive, the core dump saver program stores the files in the flash memory in the database file sm_coredump.gz present within the flash memory. The database holds up to 10 core dump files for each program in the security module.

NOTE: Use the get file command to determine if the sm_coredump.gz file exists in the flash memory. If the file exists it will be listed along with the other files residing in the flash.

In the database file in flash memory, a core filename contains the program name, the security module ID, and a timestamp in the format YYMMDDHHMMSSNN,where NN identifies core files generated in the same second (SS). The core file engine_091000823445500_SM3.core indicates that the file was generated in security module 3.

Chapter 18

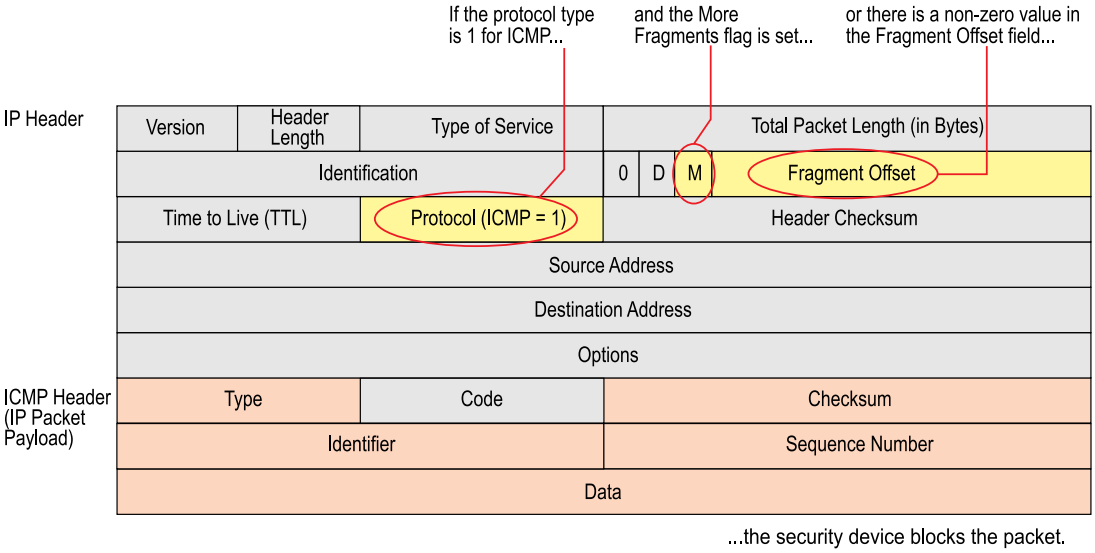
Suspicious Packet Attributes

As shown in the other chapters in this guide, attackers can craft packets to perform reconnaissance or launch denial of service (DoS) attacks. Sometimes it is unclear what the intent of a crafted packet is, but the very fact that it is crafted suggests that its being put to some kind of insidious use. All of the SCREEN options presented in this chapter block suspicious packets that might contain hidden threats:

ICMP Fragments

Internet Control Message Protocol (ICMP) provides error reporting and network probe capabilities. Because ICMP packets contain very short messages, there is no legitimate reason for ICMP packets to be fragmented. If an ICMP packet is so large that it must be fragmented, something is amiss. When you enable the ICMP Fragment Protection SCREEN option, the security device blocks any ICMP packet that has the More Fragments flag set or that has an offset value indicated in the offset field.

Figure 187: Blocking ICMP Fragments



To block fragmented ICMP packets, do either of the following, where the specified security zone is the one from which the fragments originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **ICMP Fragment Protection**, then click **Apply**.

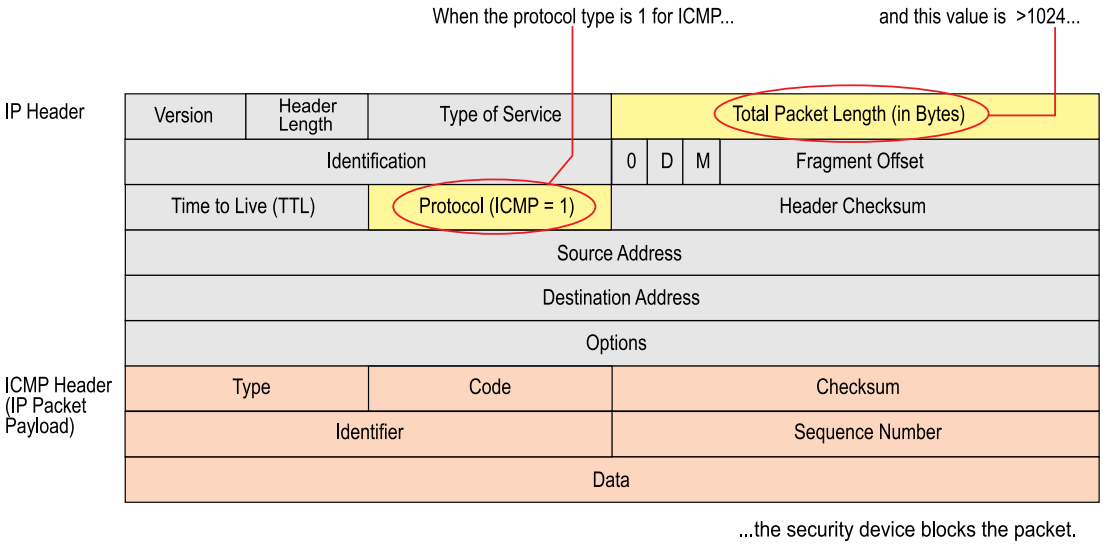
CLI

```
set zone zone screen icmp-fragment
```

Large ICMP Packets

As stated in "ICMP Fragments" on page 697, Internet Control Message Protocol (ICMP) provides error reporting and network probe capabilities. Because ICMP packets contain very short messages, there is no legitimate reason for large ICMP packets. If an ICMP packet is unusually large, something is wrong. For example, the Loki program uses ICMP as a channel for transmitting covert messages. The presence of large ICMP packets might expose a compromised machine acting as a Loki agent. It also might indicate some other kind of questionable activity.

Figure 188: Blocking Large ICMP Packets



When you enable the Large Size ICMP Packet Protection SCREEN option, the security device drops ICMP packets with a length greater than 1024 bytes.

To block large ICMP packets, do either of the following, where the specified security zone is the one from which the ICMP packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **Large Size ICMP Packet (Size > 1024) Protection**, then click **Apply**.

CLI

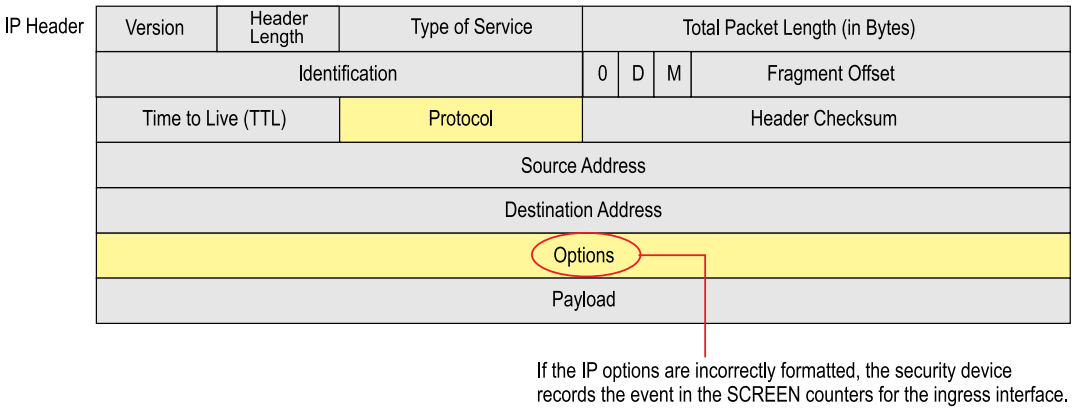
```
set zone zone screen icmp-large
```

Bad IP Options

The Internet Protocol standard RFC 791, *Internet Protocol*, specifies a set of eight options that provide special routing controls, diagnostic tools, and security. Although the original, intended uses for these options served worthy ends, people have figured out ways to twist these options to accomplish less commendable objectives. (For a summary of the exploits that attackers can initiate from IP options, see “Network Reconnaissance Using IP Options” on page 443.)

Either intentionally or accidentally, attackers sometimes configure IP options incorrectly, producing either incomplete or malformed fields. Regardless of the intentions of the person who crafted the packet, the incorrect formatting is anomalous and potentially harmful to the intended recipient.

Figure 189: Incorrectly Formatted IP Options



When you enable the Bad IP Option Protection SCREEN option, the security device blocks packets when any IP option in the IP packet header is incorrectly formatted. The security device records the event in the event log.

To detect and block IP packets with incorrectly formatted IP options, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **Bad IP Option Protection**, then click **Apply**.

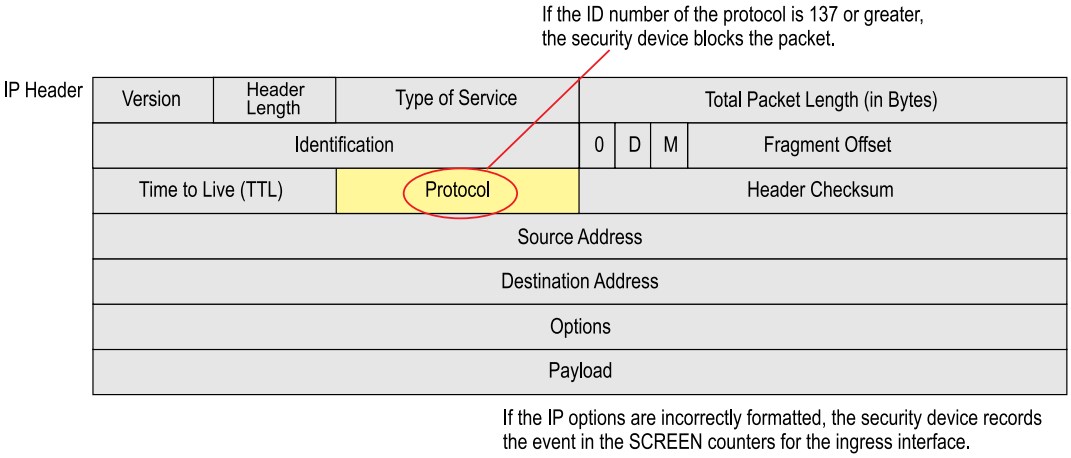
CLI

```
set zone zone screen ip-bad-option
```

Unknown Protocols

These protocol types with ID numbers of 137 or greater are reserved and undefined at this time. Precisely because these protocols are undefined, there is no way to know in advance if a particular unknown protocol is benign or malicious. Unless your network makes use of a nonstandard protocol with an ID number of 137 or greater, a cautious stance is to block such unknown elements from entering your protected network.

Figure 190: Unknown Protocols



When you enable the Unknown Protocol Protection SCREEN option, the security device drops packets when the protocol field is contains a protocol ID number of 137 or greater.

To drop packets using an unknown protocol, do either of the following, where the specified security zone is the one from which the packets originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **Unknown Protocol Protection**, then click **Apply**.

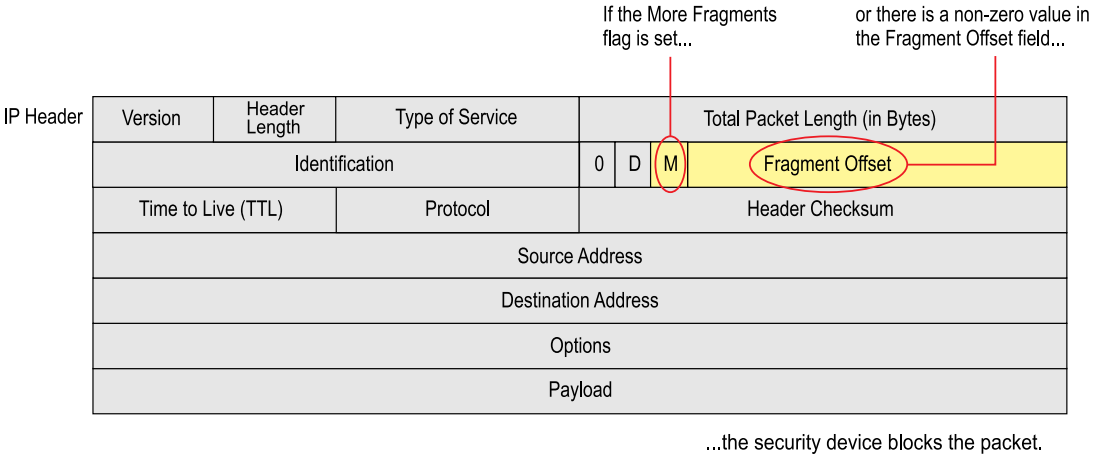
CLI

```
set zone zone screen unknown-protocol
```

IP Packet Fragments

As packets traverse different networks, it is sometimes necessary to break a packet into smaller pieces (fragments) based upon the maximum transmission unit (MTU) of each network. IP fragments might contain an attacker's attempt to exploit the vulnerabilities in the packet reassembly code of specific IP stack implementations. When the victim receives these packets, the results can range from processing the packets incorrectly to crashing the entire system.

Figure 191: IP Packet Fragments



When you enable the security device to deny IP fragments on a security zone, the device blocks all IP packet fragments that it receives at interfaces bound to that zone.

To drop fragmented IP packets, do either of the following, where the specified security zone is the one from which the fragments originate:

WebUI

Screening > Screen (Zone: select a zone name): Select **Block Fragment Traffic**, then click **Apply**.

CLI

```
set zone zone screen block-frag
```

SYN Fragments

The Internet Protocol (IP) encapsulates a Transmission Control Protocol (TCP) SYN segment in the IP packet that initiates a TCP connection. Because the purpose of this packet is to initiate a connection and invoke a SYN/ACK segment in response, the SYN segment typically does not contain any data. Because the IP packet is small, there is no legitimate reason for it to be fragmented. A fragmented SYN packet is anomalous, and as such suspect. To be cautious, block such unknown elements from entering your protected network.

When you enable the SYN Fragment Detection SCREEN option, the security device detects packets when the IP header indicates that the packet has been fragmented and the SYN flag is set in the TCP header. The security device records the event in the SCREEN counters list for the ingress interface.

To drop IP packets containing SYN fragments, do either of the following, where the specified security zone is the one from which the packets originate:

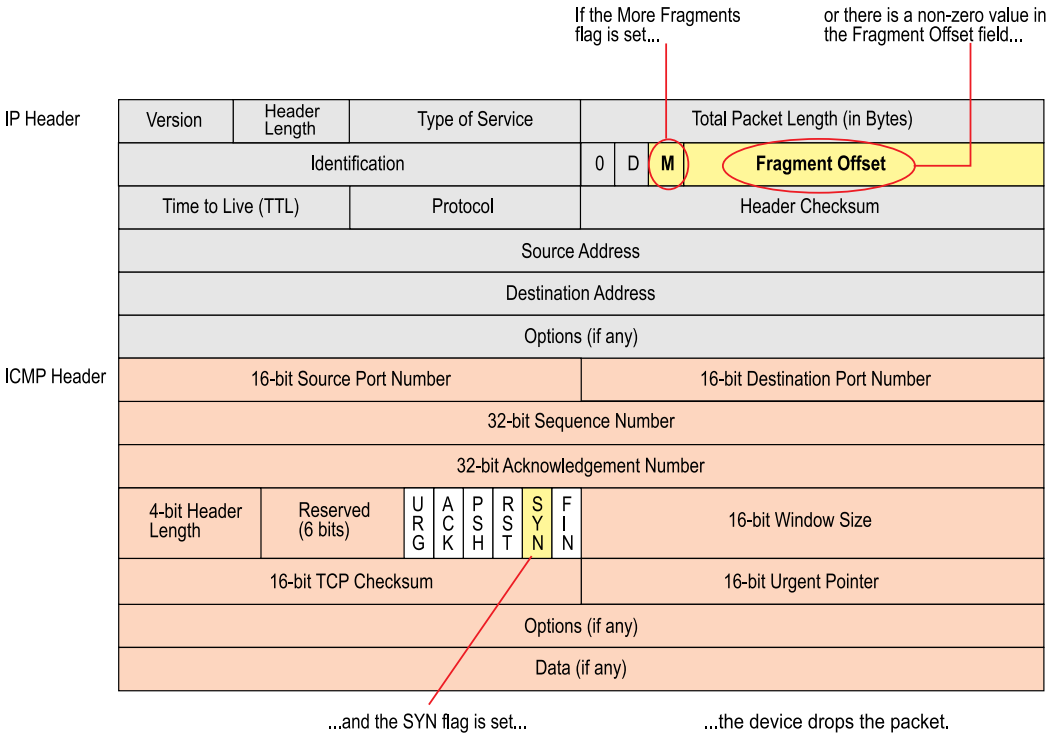
WebUI

Screening > Screen (Zone: select a zone name): Select **SYN Fragment Protection**, then click **Apply**.

CLI

set zone *zone* screen syn-frag

Figure 192: SYN Fragments



Part 5
Virtual Private Networks

Virtual Private Networks describes virtual private network (VPN) concepts and ScreenOS VPN-specific features.

This guide contains the following chapters:

- "Internet Protocol Security" on page 707 provides background information about IPsec, presents a flow sequence for Phase 1 in IKE negotiations in aggressive and main modes, and concludes with information about IKE and IPsec packet encapsulation.
- "Public Key Cryptography" on page 741 provides an introduction to public key cryptography, certificate use, and certificate revocation list (CRL) use within the context of Public Key Infrastructure (PKI).
- "Virtual Private Network Guidelines" on page 769 offers some useful information to help in the selection of the available VPN options. It also presents a packet flow chart to demystify VPN packet processing.
- "Site-to-Site Virtual Private Networks" on page 801 provides extensive examples of VPN configurations connecting two private networks.
- "Dialup Virtual Private Networks" on page 887 provides extensive examples of client-to-LAN communication using AutoKey IKE. It also details group IKE ID and shared IKE ID configurations.
- "Layer 2 Tunneling Protocol" on page 933 explains Layer 2 Tunneling Protocol (L2TP) and provides configuration examples for L2TP and L2TP-over-IPsec.
- "Advanced Virtual Private Network Features" on page 961 contains information and examples for the more advanced VPN configurations, such as NAT-Traversal, VPN monitoring, binding multiple tunnels to a single tunnel interface, and hub-and-spoke and back-to-back tunnel designs.
- "AutoConnect-Virtual Private Networks" on page 1059 describes how ScreenOS uses Next Hop Resolution Protocol (NHRP) messages to enable security devices to set up AutoConnect VPNs as needed. The chapter provides an example of a typical scenario in which AC-VPN might be used.

Chapter 19

Internet Protocol Security

This chapter introduces elements of Internet Protocol security (IPsec) and describes how they relate to virtual private network (VPN) tunneling. This chapter contains the following sections:

Introduction to Virtual Private Networks

A virtual private network (VPN) provides a means for securely communicating between remote computers across a public wide area network (WAN), such as the Internet.

A VPN connection can link two local area networks (LANs) or a remote dialup user and a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. To secure VPN communication while passing through the WAN, the two participants create an IP security (IPsec) tunnel.

NOTE: The term *tunnel* does not denote either transport or tunnel mode (see "Modes" on page 709). It refers to the IPsec connection.

An IPsec tunnel consists of a pair of unidirectional Security Associations (SAs)—one at each end of the tunnel—that specify the security parameter index (SPI), destination IP address, and security protocol (Authentication Header or Encapsulating Security Payload) employed.

For more information about SPIs, see "Security Associations" on page 714. For more information about IPsec security protocols, see "Protocols" on page 711.

Through the SA, an IPsec tunnel can provide the following security functions:

- Privacy (through encryption)
- Content integrity (through data authentication)
- Sender authentication and—if using certificates—nonrepudiation (through data origin authentication)

The security functions you employ depend on your needs. If you only need to authenticate the IP packet source and content integrity, you can authenticate the packet without applying any encryption. On the other hand, if you are only concerned with preserving privacy, you can encrypt the packet without applying any authentication mechanisms. Optionally, you can both encrypt and authenticate the packet. Most network security designers choose to encrypt, authenticate, and replay-protect their VPN traffic.

ScreenOS supports IPsec technology for creating VPN tunnels with two kinds of key creation mechanisms:

- Manual Key
- AutoKey IKE with a preshared key or a certificate

IPsec Concepts

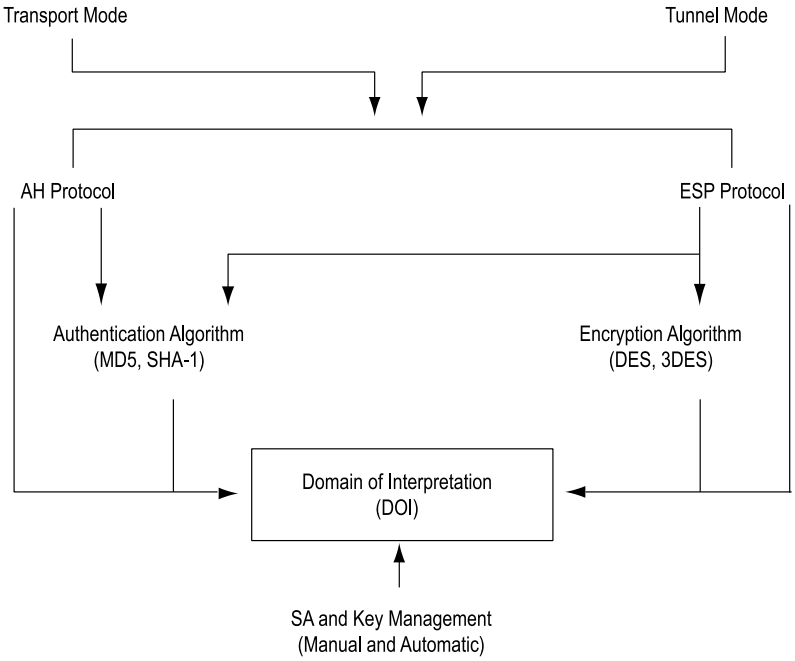
Internet Protocol security (IPsec) is a suite of related protocols for cryptographically securing communications at the IP Packet Layer. IPsec consists of two modes and two main protocols:

- Transport and tunnel modes
- The Authentication Header (AH) protocol for authentication and the Encapsulating Security Payload (ESP) protocol for encryption (and authentication)

IPsec also provides methods for the manual and automatic negotiation of security associations (SAs) and key distribution, all the attributes of which are gathered in a Domain of Interpretation (DOI). Refer to RFC 2407 and RFC 2408.

Figure 193: IPsec Architecture

Note: ScreenOS does not support transport mode with AH for IPv6 packets.



NOTE: The IPsec Domain of Interpretation (DOI) is a document containing definitions for all the security parameters required for the successful negotiation of a VPN tunnel—essentially, all the attributes required for SA and IKE negotiations.

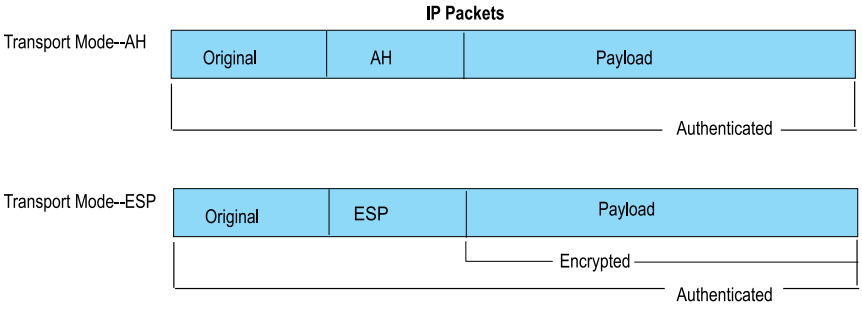
Modes

IPsec operates in one of two modes—transport or tunnel. When both ends of the tunnel are hosts, you can use either mode. When at least one of the endpoints of a tunnel is a security gateway, such as a router or firewall, you must use tunnel mode. Juniper Networks security devices always operate in tunnel mode for IPsec tunnels and transport mode for L2TP-over-IPsec tunnels.

Transport Mode

In transport mode, the original IP packet is not encapsulated within another IP packet, as shown in Figure 194 on page 709. The entire packet can be authenticated (with AH), the payload can be encrypted (with ESP), and the original header remains in plaintext as it is sent across the WAN.

Figure 194: Transport Modes

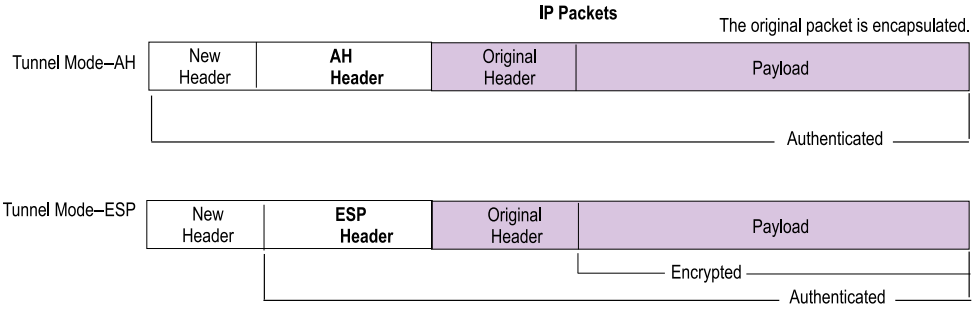


NOTE: In the current release, ScreenOS supports transport mode with AH on the high-end systems (ISG and NS series of products) for IPv4 packets only. This feature does not work if IPv6 is enabled on the device.

Tunnel Mode

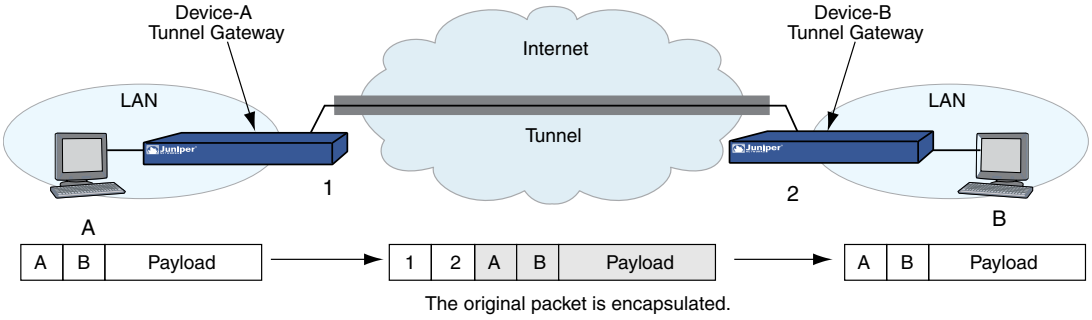
In tunnel mode, the entire original IP packet—payload and header—is encapsulated within another IP payload and a new header is prepended to it, as shown in Figure 195 on page 710. The entire original packet can be encrypted, authenticated, or both. With AH, the AH and new headers are also authenticated. With ESP, the ESP header can also be authenticated.

Figure 195: Tunnel Modes



In a site-to-site VPN, the source and destination addresses used in the new header are the IP addresses of the outgoing interface (in NAT or route mode) or the VLAN1 IP address (in transparent mode); the source and destination addresses of the encapsulated packets are the addresses of the ultimate endpoints of the connection.

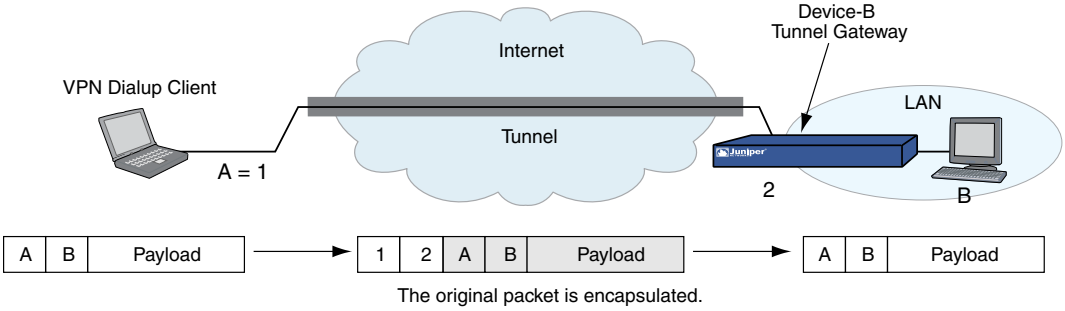
Figure 196: Site-to-Site VPN in Tunnel Mode



In a dialup VPN, there is no tunnel gateway on the VPN dialup client end of the tunnel; the tunnel extends directly to the client itself. In this case, on packets sent from the dialup client, both the new header and the encapsulated original header have the same IP address: that of the client's computer.

NOTE: Some VPN clients such as the NetScreen-Remote allow you to define a virtual inner IP address. In such cases, the virtual inner IP address is the source IP address in the original packet header of traffic originating from the client, and the IP address that the ISP dynamically assigns the dialup client is the source IP address in the outer header.

Figure 197: Dialup VPN in Tunnel Mode



Protocols

IPsec uses two protocols to secure communications at the IP Layer:

- Authentication Header (AH)—A security protocol for authenticating the source of an IP packet and verifying the integrity of its content
- Encapsulating Security Payload (ESP)—A security protocol for encrypting the entire IP packet (and authenticating its content)

Authentication Header

The Authentication Header (AH) protocol is used to verify the authenticity and integrity of the content and origin of a packet. You can authenticate the packet by the checksum calculated through a Hash Message Authentication Code (HMAC) using a secret key and the MD5, SHA-1 or SHA-2 hash functions.

- **Message Digest version 5 (MD5)**—Algorithm that produces a 128-bit hash (also called a *digital signature* **or** *message digest*) from a message of arbitrary length and a 16-byte key. The resulting hash is used, like a fingerprint of the input, to verify content and source authenticity and integrity.
- **Secure Hash Algorithm-1 (SHA-1)**—Algorithm that produces a 160-bit hash from a message of arbitrary length and a 20-byte key. It is generally regarded as more secure than MD5 because of the larger hashes it produces. Because the computational processing is done in the ASIC, the performance cost is negligible.
- **Secure Hash Algorithm-2 (SHA-2)**—Set of four algorithms named after their message digest length (in bits)—SHA2-224, SHA2-256, SHA2-384, and SHA2-512. These algorithms are generally regarded as more secure than SHA-1 because of the larger hashes they produce. This release of ScreenOS supports the SHA2-256 hash algorithm. The SHA2-256 algorithm produces a 256-bit hash from a message of arbitrary length and a 32-byte key.

NOTE: For more information about the MD5, SHA-1, and SHA2-256 hashing algorithms, refer to the following RFCs: (MD5) 1321, 2403; (SHA-1) 2404; (SHA2-256) 4753, 4868. For information about HMAC, refer to RFC 2104.

In the current release, ScreenOS supports transport mode with AH on the high-end systems for IPv4 packets only. This feature does not work if IPv6 is enabled on the device.

Encapsulating Security Payload

The Encapsulating Security Payload (ESP) protocol provides a means for ensuring privacy (encryption) and source authentication and content integrity (authentication). ESP in tunnel mode encapsulates the entire IP packet (header and payload) and then appends a new IP header to the now-encrypted packet. This new IP header contains the destination address needed to route the protected data through the network.

With ESP, you can both encrypt and authenticate, encrypt only, or authenticate only. For encryption, you can choose one of the following encryption algorithms:

- **Data Encryption Standard (DES)**—A cryptographic block algorithm with a 56-bit key.
- **Triple DES (3DES)**—A more powerful version of DES in which the original DES algorithm is applied in three rounds, using a 168-bit key. DES provides a significant performance savings but is considered unacceptable for many classified or sensitive material transfers.
- **Advanced Encryption Standard (AES)**—An emerging encryption standard which, when adopted by Internet infrastructures worldwide, will offer greater interoperability with other network security devices. ScreenOS supports AES with 128-bit, 192-bit, and 256-bit keys.

For authentication, you can use the MD5, SHA-1 or SHA2-256 algorithms.

NOTE: Even though it is possible to select **NULL** for authentication, it has been demonstrated that IPsec might be vulnerable to attack under such circumstances. Therefore, it is inadvisable to select **NULL** for authentication.

Key Management

Key distribution and management are critical to using VPNs successfully. IPsec supports both manual and automatic key-distribution methods.

Manual Key

With manual key encryption, administrators at both ends of a tunnel configure all the security parameters. This is a viable technique for small, static networks where the distribution, maintenance, and tracking of keys are not difficult. However, safely

distributing manual-key configurations across great distances poses security issues. Aside from passing a key face-to-face, you cannot be completely sure that the key has not been compromised while in transit. Also, whenever you want to change the key, you are faced with the same security issues as when you initially distributed it.

AutoKey IKE

When you need to create and manage numerous tunnels, you need a method that does not require you to manually configure every element. IPsec supports the automated generation and negotiation of keys and security associations using the Internet Key Exchange (IKE) protocol. ScreenOS refers to such automated tunnel negotiation as *AutoKey IKE* and supports AutoKey IKE with preshared keys and AutoKey IKE with certificates.

AutoKey IKE with Preshared Keys

With AutoKey IKE which uses preshared keys to authenticate the participants in an IKE session, each side must configure and securely exchange the preshared key in advance. In this regard, the issue of secure key distribution is the same as that with manual keys. However, once distributed, an autokey, unlike a manual key, can automatically change its keys at predetermined intervals using the IKE protocol. Frequently changing keys greatly improves security, and automatically doing so greatly reduces key-management responsibilities. However, changing keys increases traffic overhead; therefore, doing so too often can reduce data transmission efficiency.

NOTE: A preshared key is a key for both encryption and decryption, which both participants must possess before initiating communication.

AutoKey IKE with Certificates

When using certificates to authenticate the participants during an AutoKey IKE negotiation, each side generates a public/private key pair (see “Public Key Cryptography” on page 741) and acquires a certificate (see “Certificates and CRLs” on page 746). As long as the issuing certificate authority (CA) is trusted by both sides, the participants can retrieve the peer’s public key and verify the peer’s signature. There is no need to keep track of the keys and SAs; IKE does so automatically.

NOTE: For examples of both Manual Key and AutoKey IKE tunnels, see “Site-to-Site Virtual Private Networks” on page 801.

Key Protection

Juniper Networks security devices protect VPN-persistent private keys against unauthorized access and modification. By enabling the key protection feature, the security device encrypts VPN persistent private keys, checks integrity of the key whenever the key is used, and destroys the key memory with different key patterns in the system.

The following types of VPN private keys are encrypted:

- PKI private keys (DSA/RSA/ECDSA)
- IKE preshared keys and preshared key seeds
- VPN manual keys (keys generated from passwords)

All VPN manual keys and keys generated from passwords are encrypted from plaintext to encrypted text using a master key (a hard-coded key). The same master key is used to decrypt the encrypted key back to plaintext. You cannot access the master key if you are accessing the system through any management interface. The AES (128-bit) encryption algorithm is used to encrypt the keys. The security device uses the single-parity-bit Error Detection Code (EDC) algorithm to detect key errors.

Enabling Key Protection

You can enable key protection through the WebUI or the CLI. The key protection feature is disabled by default.

WebUI

Configuration > Admin > Management: Select **Enable Key Protection,** then click **Apply.**

CLI

```
set key protection enable
save
```

Security Associations

A security association (SA) is a unidirectional agreement between the VPN participants regarding the methods and parameters to use in securing a communication channel. Full bidirectional communication requires at least two SAs, one for each direction.

An SA groups together the following components for securing communications:

- Security algorithms and keys
- Protocol mode (transport or tunnel)
- Key-management method (manual key or AutoKey IKE)
- SA lifetime

For outbound VPN traffic, the policy invokes the SA associated with the VPN tunnel. For inbound traffic, the security device looks up the SA by using the following triplet:

- Destination IP
- Security protocol (AH or ESP)
- Security parameter index (SPI) value

Tunnel Negotiation

For a manual key IPsec tunnel, because all of the security association (SA) parameters have been previously defined, there is no need to negotiate which SAs to use. In essence, the tunnel has already been established. When traffic matches a policy using that manual key tunnel or when a route involves the tunnel, the security device simply encrypts and authenticates the data, as you determined, and forwards it to the destination gateway.

To establish an AutoKey IKE IPsec tunnel, two phases of negotiations are required:

- In Phase 1, the participants establish a secure channel in which to negotiate the IPsec SAs.
- In Phase 2, the participants negotiate the IPsec SAs for encrypting and authenticating the ensuing exchanges of user data.

NOTE: Juniper Networks security devices support the newer version of the IKE protocol known as IKEv2. For more information about IKEv2 and how security devices establish security associations (SAs) using the IKEv2 protocol, see "IKE Version 2" on page 724.

Phase 1

Phase 1 of an AutoKey IKE tunnel negotiation consists of the exchange of proposals for how to authenticate and secure the channel. The exchange can be in one of two modes: aggressive or main. Using either mode, the participants exchange proposals for acceptable security services such as:

- Encryption algorithms (DES and 3DES) and authentication algorithms (MD5, SHA-1 or SHA2-256). For more information about these algorithms, see "Protocols" on page 711.
- A Diffie-Hellman group (see "Diffie-Hellman Exchange" on page 716).
- Preshared key or RSA/DSA certificates (see "AutoKey IKE" on page 713).

A successful Phase 1 negotiation concludes when both ends of the tunnel agree to accept at least one set of the Phase 1 security parameters proposed and then process them. Juniper Networks security devices support up to four proposals for Phase 1 negotiations, allowing you to define how restrictive a range of security parameters for key negotiation you will accept.

The predefined Phase 1 proposals that ScreenOS provides are as follows:

- **Standard:** pre-g2-aes128-sha and pre-g2-3des-sha
- **Compatible:** pre-g2-3des-sha, pre-g2-3des-md5, pre-g2-des-sha, and pre-g2-des-md5
- **Basic:** pre-g1-des-sha and pre-g1-des-md5

You can also define custom Phase 1 proposals.

Main and Aggressive Modes

Phase 1 can take place in either main or aggressive mode. The two modes are described below.

Main mode: The initiator and recipient send three two-way exchanges (six messages total) to accomplish the following services:

- First exchange (messages 1 and 2): Propose and accept the encryption and authentication algorithms.
- Second exchange (messages 3 and 4): Execute a DH exchange, and the initiator and recipient each provide a pseudo-random number.
- Third exchange (messages 5 and 6): Send and verify their identities.

The information transmitted in the third exchange of messages is protected by the encryption algorithm established in the first two exchanges. Thus, the participants' identities are not transmitted in the clear.

Aggressive mode: The initiator and recipient accomplish the same objectives, but only in two exchanges, with a total of three messages:

- First message: The initiator proposes the SA, initiates a DH exchange, and sends a pseudo-random number and its IKE identity.
- Second message: The recipient accepts the SA; authenticates the initiator; and sends a pseudo-random number, its IKE identity, and, if using certificates, the recipient's certificate.
- Third message: The initiator authenticates the recipient, confirms the exchange, and, if using certificates, sends the initiator's certificate.

Because the participants' identities are exchanged in the clear (in the first two messages), aggressive mode does not provide identity protection.

NOTE: When a dialup VPN user negotiates an AutoKey IKE tunnel with a preshared key, aggressive mode must be used. Note also that a dialup VPN user can use an email address, a fully qualified domain name (FQDN), or an IP address as its IKE ID. A dynamic peer can use either an email address or FQDN, but not an IP address.

Diffie-Hellman Exchange

A Diffie-Hellman (DH) exchange allows the participants to produce a shared secret value. The strength of the technique is that it allows the participants to create the secret value over an unsecured medium without passing the secret value through the wire. ScreenOS supports DH groups 1, 2, 5, and 14 for Internet Key Exchange version 1 (IKEv1) and IKE version 2 (IKEv2). The size of the prime modulus used in each group's calculation differs as follows:

- **DH group 1:** 768 bit
- **DH group 2:** 1024 bit
- **DH group 5:** 1536 bit
- **DH group 14:** 2048 bit

NOTE: The strength of DH group 1 security has depreciated, and we do not recommend its use.

The larger the modulus, the more secure the generated key is considered to be; however, the larger the modulus, the longer the key-generation process takes. Because the modulus for each DH group is a different size, the participants must agree to use the same group.

NOTE: If you configure multiple (up to four) proposals for Phase 1 negotiations, you can use different DH groups in all proposals. The same guideline applies to multiple proposals for Phase 2 negotiations.

Elliptical Curve Diffie-Hellman

An Elliptical Curve Diffie-Hellman (ECDH) exchange is a variant of the Diffie-Hellman (DH) protocol. ECDH uses elliptical curve cryptography to generate a public-private key pair. ECDH is an integral part of the Suite B cryptography standards proposed by the National Security Agency (NSA) for protecting both classified and unclassified information. Suite B cryptography complements NSA's existing policy for using AES. In addition to AES, Suite B includes cryptographic algorithms for hashing, digital signatures, and key exchanges. Previous releases of ScreenOS already support most Suite B requirements with the exception of ECDH. In the current release, ScreenOS supports ECDH groups 19 and 20 for Phase 1 and Phase 2 IKEv1 negotiations only.

The advantage of ECDH over classic DH is that ECDH significantly reduces the size of the public-private key pair. The size of the prime modulus used in each group's calculation differs as follows:

- **DH group 19:** 256 bits ECDH prime curve
- **DH group 20:** 384 bits ECDH prime curve

NOTE: You cannot configure ECDH groups on IKEv2 gateways. If you attempt to configure a Phase 1 or 2 proposal that uses an ECDH group, the security device generates the following error: **ECDH groups are not supported by IKEv2. Use Oakley group 1, 2, 5, or 14 for IKEv2 negotiations.**

Phase 2

After the participants have established a secure and authenticated channel, they proceed through Phase 2, in which they negotiate the SAs to secure the data to be transmitted through the IPsec tunnel.

Like the process for Phase 1, the participants exchange proposals to determine which security parameters to employ in the SA. A Phase 2 proposal also includes a security protocol—either Encapsulating Security Payload (ESP) or Authentication Header (AH)—and selected encryption and authentication algorithms. The proposal can also specify a DH group, if Perfect Forward Secrecy (PFS) is desired.

Regardless of the mode used in Phase 1, Phase 2 always operates in quick mode and involves the exchange of three messages.

Juniper Networks security devices support up to four proposals for Phase 2 negotiations, allowing you to define how restrictive a range of tunnel parameters you will accept. ScreenOS also provides a replay protection feature. Use of this feature does not require negotiation because packets are always sent with sequence numbers. You simply have the option of checking or not checking the sequence numbers. (For more information about replay protection, see “Replay Protection” on page 719.)

The predefined Phase 2 proposals that ScreenOS provides are as follows:

- **Standard:** g2-esp-3des-sha and g2-esp-aes128-sha
- **Compatible:** nopfs-esp-3des-sha, nopfs-esp-3des-md5, nopfs-esp-des-sha, and nopfs-esp-des-md5
- **Basic:** nopfs-esp-des-sha and nopfs-esp-des-md5

You can also define custom Phase 2 proposals.

In Phase 2, the peers also exchange proxy IDs. A proxy ID is a three-part tuple consisting of local IP address–remote IP address–service. The proxy ID for both peers must match, which means that the service specified in the proxy ID for both peers must be the same, and the local IP address specified for one peer must be the same as the remote IP address specified for the other peer.

NOTE: Phase 2 negotiations for IPv6 support Netscreen Redundancy Protocol (NSRP).

The CREATE_CHILD_SA exchange in an IKEv2 exchange corresponds to the Phase 2 negotiations in IKEv1. For more information, see “Initial Exchanges” on page 724.

Perfect Forward Secrecy

Perfect Forward Secrecy (PFS) is a method for deriving Phase 2 keys independent from and unrelated to the preceding keys. Alternatively, the Phase 1 proposal creates the key (the SKEYID_d key) from which all Phase 2 keys are derived. The SKEYID_d key can generate Phase 2 keys with a minimum of CPU processing. Unfortunately,

if an unauthorized party gains access to the SKEYID_d key, all your encryption keys are compromised.

PFS addresses this security risk by forcing a new Diffie-Hellman key exchange to occur for each Phase 2 tunnel. Using PFS is thus more secure, although the rekeying procedure in Phase 2 might take slightly longer with PFS enabled.

Replay Protection

A replay attack occurs when somebody intercepts a series of packets and uses them later either to flood the system, causing a denial of service (DoS), or to gain entry to the trusted network. The replay-protection feature enables security devices to check every IPsec packet to see if it has been received previously. If packets arrive outside a specified sequence range, the security device rejects them.

IKE and IPsec Packets

An IPsec VPN tunnel consists of two major elements:

- **Tunnel Setup:** The peers first establish security associations (SAs), which define the parameters for securing traffic between themselves. The admins at each end can define the SAs manually, or the SAs can be defined dynamically through IKE Phase 1 and Phase 2 negotiations. Phase 1 can occur in either main or aggressive mode. Phase 2 always occurs in quick mode.
- **Applied Security:** IPsec protects traffic sent between the two tunnel endpoints by using the security parameters defined in the SAs that the peers agreed to during the tunnel setup. IPsec can be applied in one of two modes—transport or tunnel. Both modes support the two IPsec protocols—Encapsulating Security Payload (ESP) and Authentication Header (AH).

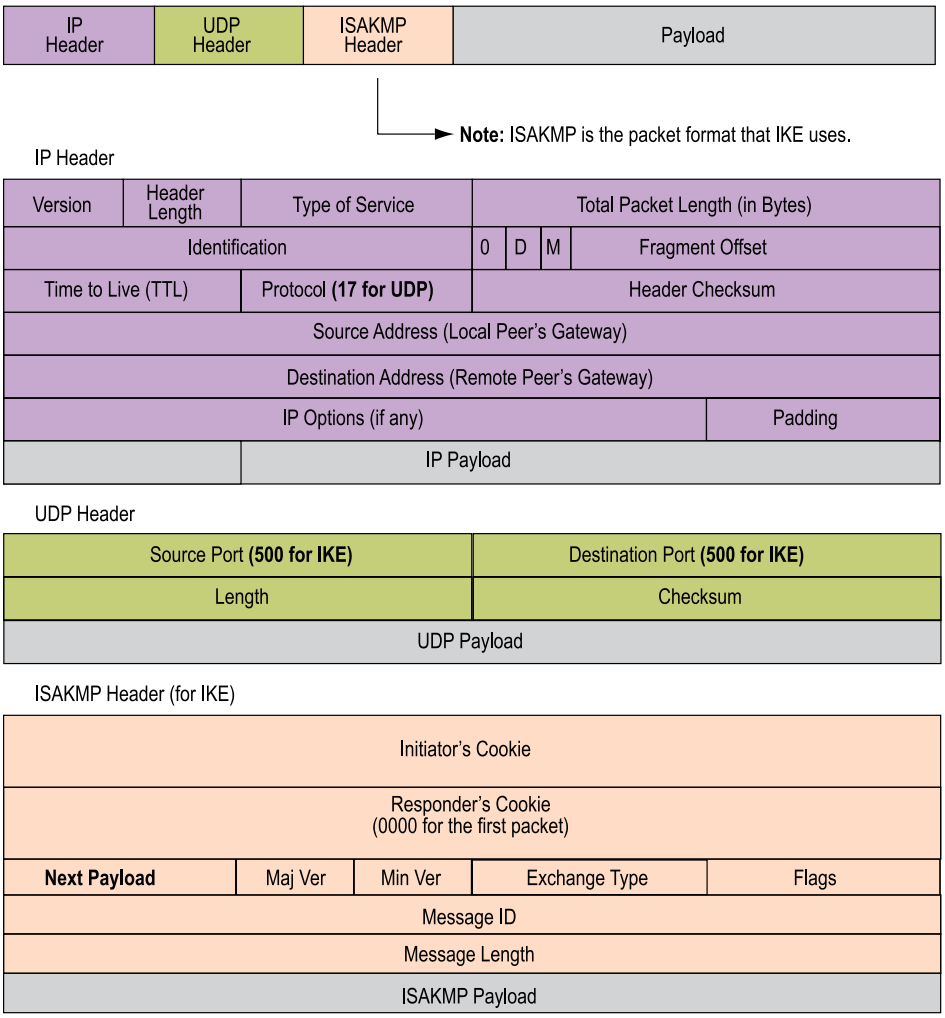
For an explanation of the packet processing that occurs during the IKE and IPsec stages of a VPN tunnel, see “IKE Packets” on page 719 and “IPsec Packets” on page 722. These sections show the packet headers for IKE and IPsec, respectively.

IKE Packets

When a clear-text packet arrives at the security device that requires tunneling and no active Phase 2 SA exists for that tunnel, the security device begins IKE negotiations (and drops the packet). The source and destination addresses in the IP packet header are those of the local and remote IKE gateways, respectively. In the IP packet payload, there is a UDP segment encapsulating an Internet Security Association and Key Management Protocol (ISAKMP), or IKE, packet. The format for IKE packets is the same for Phase 1 and Phase 2.

NOTE: When the initial IP packet is dropped, the source host resends it. Typically, by the time the second packet reaches the security device, IKE negotiations are complete and the security device protects it—and all subsequent packets in the session—with IPsec before forwarding it.

Figure 198: IKE Packet for Phases 1 and 2



The Next Payload field contains a number indicating one of the following payload types:

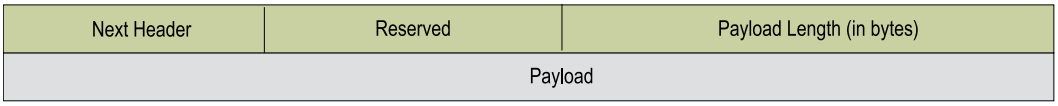
- 0002—SA Negotiation Payload: contains a definition for a Phase 1 or Phase 2 SA.
- 0004—Proposal Payload: can be a Phase 1 or Phase 2 proposal.
- 0008—Transform Payload: the transform payload gets encapsulated in a proposal payload which gets encapsulated in an SA payload.
- 0010—Key Exchange (KE) Payload: contains information necessary to perform a key exchange, such as a Diffie-Hellman public value.
- 0020—Identification (IDx) Payload.
  - In Phase 1, IDii indicates the initiator ID, and IDir indicates the responder ID.
  - In Phase 2, IDui indicates the user initiator, and IDur indicates the user responder.

The IDs are IKE ID types such as FQDN, U-FQDN, IP address, and ASN.1_DN.

- 0040—Certificate (CERT) Payload.
- 0080—Certificate Request (CERT_REQ) Payload.
- 0100—Hash (HASH) Payload: contains the digest output of a particular hash function.
- 0200—Signature (SIG) Payload: contains a digital signature.
- 0400—Nonce (Nx) Payload: contains some pseudo-random information necessary for the exchange).
- 0800—Notify Payload.
- 1000—ISAKMP Delete Payload.
- 2000—Vendor ID (VID) Payload: can be included anywhere in Phase 1 negotiations. ScreenOS uses it to mark support for Network Address Translation-Traversal (NAT-T).

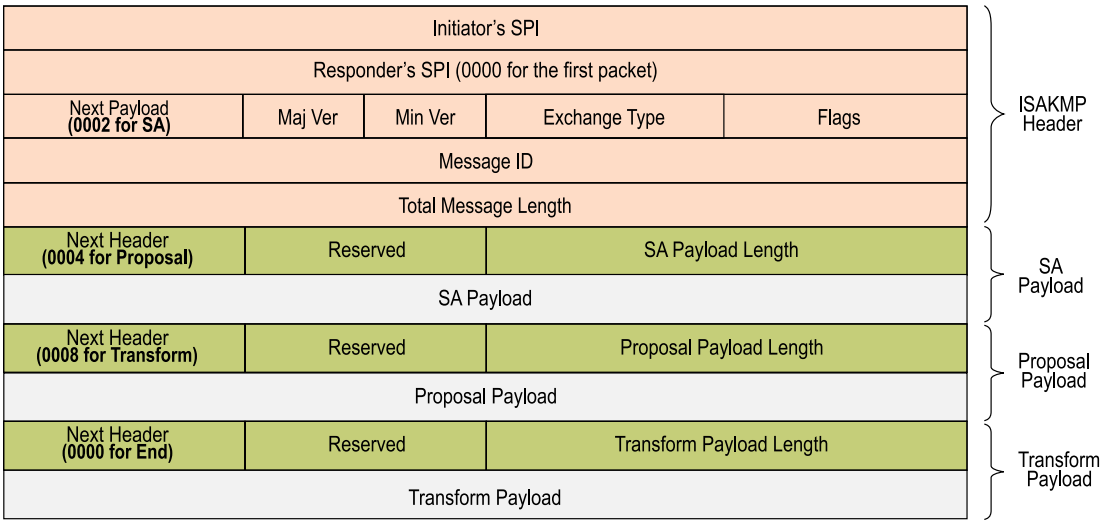
Each ISAKMP payload begins with the same generic header, as shown in Figure 199 on page 721.

Figure 199: Generic ISAKMP Payload Header



There can be multiple ISAKMP payloads chained together, with each subsequent payload type indicated by the value in the Next Header field. A value of **0000** indicates the last ISAKMP payload. See Figure 200 on page 721 for an example.

Figure 200: ISAKMP Header with Generic ISAKMP Payloads

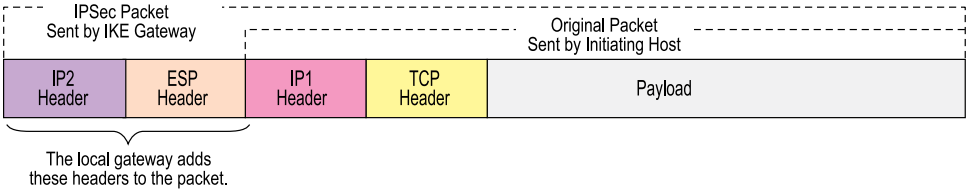


IPsec Packets

After IKE negotiations complete and the two IKE gateways have established Phase 1 and Phase 2 security associations (SAs), the security device applies IPsec protection to subsequent clear-text IP packets that hosts behind one IKE gateway send to hosts behind the other gateway (assuming that policies permit the traffic). If the Phase 2 SA specifies the Encapsulating Security Protocol (ESP) in tunnel mode, the packet looks like the one shown below. The security device adds two additional headers to the original packet that the initiating host sends.

NOTE: For information about ESP, see "Encapsulating Security Payload" on page 712. For information about tunnel mode, see "Tunnel Mode" on page 709.

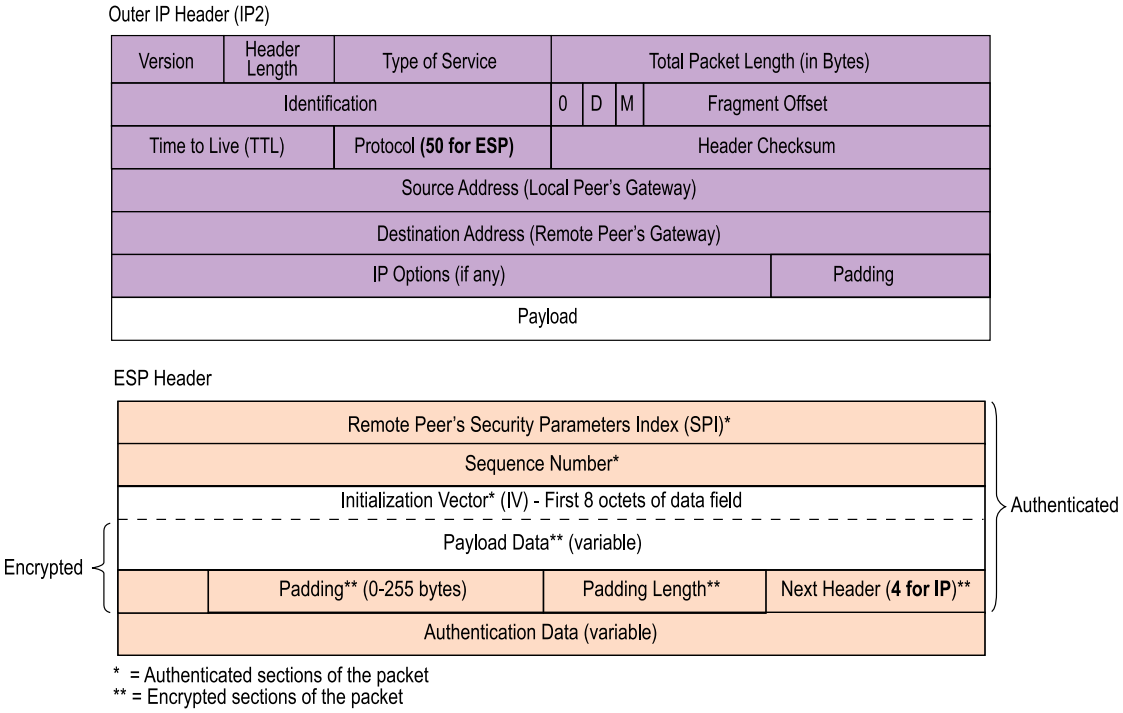
Figure 201: IPsec Packet—Encapsulating Security Payload in Tunnel Mode



As shown in Figure 201 on page 722, the packet that the initiating host constructs includes the payload, the TCP header, and the inner IP header (IP1).

The outer IP header (IP2), which the security device adds, contains the IP address of the remote gateway as the destination IP address and the IP address of the local security device as the source IP address. The security device also adds an ESP header between the outer and inner IP headers. The ESP header contains information that allows the remote peer to properly process the packet when it receives it. This is illustrated in Figure 202 on page 723.

Figure 202: Outer IP Header (IP2) and ESP Header



The Next Header field indicates the type of data in the payload field. In tunnel mode, this value is 4, indicating IP-in-IP. If ESP is applied in transport mode, this value indicates a Transport Layer protocol such as 6 for TCP or 17 for UDP.

Figure 203: Inner IP Header (IP1) and TCP Header

Inner IP Header (IP1)

Version	Header Length	Type of Service	Total Packet Length (in Bytes)			
Identification			0	D	M	Fragment Offset
Time to Live (TTL)		Protocol (6 for TCP)	Header Checksum			
Source Address (Initiating Host)						
Destination Address (Receiving Host)						
IP Options (if any)						Padding
Payload						

TCP Header

Source Port								Destination Port
Sequence Number								
Acknowledgement Number								
Header Length	Reserved	URG	ACK	PSH	RST	SYN	FIN	Window Size
Checksum								Urgent Pointer
Options (if any)								Padding
Data								

IKE Version 2

Juniper Networks security devices support a newer version of the Internet Key Exchange protocol (IKE), known as IKE version 2 (IKEv2). IKEv2 brings together various aspects of exchanging keys between IPsec endpoints, such as NAT-T, extended authentication (xauth), and ISAKMP configuration, into a single protocol and preserves most of the features of the earlier version, including identity hiding, PFS, two phases of establishing SAs, and cryptographic negotiation.

IKEv2 performs mutual authentication between two IPsec endpoints and establishes an IKE SA known as *IKE_SA,* in which the IPsec endpoints share secret information to establish SAs for Encapsulating Security Payload (ESP) protocol, Authentication Header (AH), and a set of cryptographic algorithms to be used to protect IKE traffic. The SAs for ESP or AH that get set up through the IKE_SA are called *CHILD_SAs.*

IKEv2 supports three types of exchanges: initial, CREATE_CHILD_SA, and informational. Conceptually, IKEv2 IKE_SA and CHILD_SA are equivalent to IKEv1 Phase 1 SA and Phase 2 SA, respectively.

Initial Exchanges

The IPsec endpoints start an IKEv2 SA through an initial exchange. This consists of two exchanges: IKE_SA_INIT and IKE_AUTH.

IKE_SA_INIT Exchange

An IKE_SA_INIT exchange negotiates security suites, establishes the IKE_SA, and generates the SKEYSEED from which all keys are derived for the IKE_SA. Separate keys are computed for each direction. The initiator sends the following:

- HDR—Initiator's IKE header. The header contains the security parameter indexes (SPIs), version, and flags.
- SAi1—Cryptographic algorithms the initiator supports for the IKE_SA.
- KEi—Initiator's Diffie-Hellman value
- Ni—Initiator's nonce

The responder sends the response to the initiator request with the following:

- HDR—Responder's header
- SAr1—Cryptographic algorithms the responder supports for the IKE_SA.
- KEr—Responder's Diffie-Hellman value
- Nr—Responder's nonce
- [CERTREQ]—[Optional] Certificate request

IKE_AUTH Exchange

The IKE_AUTH exchange authenticates IKE endpoints and establishes the CHILD_SA. This exchange consists of a single request/response pair. The initiator starts using the new CHILD_SA immediately after receiving the responder's response; similarly, the responder starts using the new CHILD_SA immediately after sending the response to the initiator.

In the endpoint-to-security gateway scenario where the endpoint is an Internet remote access client (IRAC) and the security gateway is an Internet remote access server (IRAS), the IRAC needs an IP address associated with the security gateway to establish a connection with the protected subnet through an IPSec tunnel. In support of this, IKEv2 enables the IRAC to request an IP address owned by the IRAS for use in the secure connection. The IRAC requests an IP address by sending a configuration payload (CP) request in the IKE_AUTH exchange. The IRAS selects an IP address from the address pool it maintains or from the external RADIUS server and sends it to the IRAC. The assigned IP address is freed when its associated IKE_SA lifetime ends. CP also supports to assign DNS and WINS servers addresses. The IRAC gets the DNS and WINS servers addresses from the IRAS itself or from an external RADIUS server.

NOTE: IKEv2 CP is not supported for IPv6 addresses.

All messages following the initial exchange are cryptographically protected using the cryptographic algorithms and keys negotiated in the first two messages of the key

exchange. These subsequent messages use the syntax of the encrypted payload. During the IKE_AUTH exchange, the endpoints exchange the following:

- HDR—Initiator's header
- IDi—Initiator's ID
- [CERT]—[Optional] Certificate
- [CERTREQ]—[Optional] Certificate Request
- IDr—Responder's ID
- AUTH—Authenticates the previous message and the initiator's identity
- CP (CFG_REQUEST)—[Optional] Exchanges the configuration information between the IKE peers
- SAi2—Initiator's SA
- TSi—Initiator's traffic selector
- TSr—Responder's traffic selector

The responder sends the following response:

- HDR—Responder's header
- IDr—Initiator's ID
- [CERT]—[Optional] Certificate
- AUTH—Authenticates the previous message and the initiator's identity
- CP (CFG_REPLY)—[Optional] Exchanges the configuration information between the IKE peers
- SAr2—Responder's SA
- TSi—Initiator's traffic selector
- TSr—Responder's traffic selector

Of these messages, except the Header, all other payload are encrypted with the secret key generated by the endpoints.

Example: Configuring IRAC and IRAS to Get an IP Address from a Local and External Databases

WebUI

1. **Configuring IRAS to Get an IP Address from a Local IP Pool:**

   VPN > AutoKey Advanced > MODECFG Profile: Enter the following, then click **New**:

   Profile Name: test
   IP Pool: ippool_test
   DNS IP1: 10.0.0.1
   IP2: 10.0.0.2
   WINS IP1: 10.0.0.3

IP2: 10.0.0.4

VPNs > AutoKey Advanced > Gateway > EAP > MODECFG Enable: Perform the following actions, then click **OK**.

Server: select
Action: select Add Route
Information Origin: select Use Local DNS
Profile: test

2. **Configuring IRAC to Get an IP address from a Local IP Pool:**

VPNs > AutoKey Advanced > Gateway > EAP > MODECFG Enable: select **Client**, then click **OK**.

CLI

1. **Configuring IRAS to Get an IP Address from a Local IP Pool:**

```
set ippool ippool_test 10.0.0.5 10.0.0.10
set ike modecfg profile name test
set ike modecfg profile test ippool ippool_test
set ike modecfg profile test dns1 10.0.0.1
set ike modecfg profile test dns2 10.0.0.2
set ike modecfg profile test wins1 10.0.0.3
set ike modecfg profile test wins2 10.0.0.4
set ike gateway ikev2 gate_test modecfg server profile test
```

2. **Configuring IRAC to Get an IP Address from a Local IP Pool:**

```
set ike gateway ikev2 gate_test modecfg client
```

WebUI

1. **Configuring IRAS to an Get an IP Address from an EAP and External RADIUS Server:**

Configuation > Auth > Auth Servers > New: Enter the following, then click **OK**.

Name: 202.0.0.1
Account Type: IKEv2EAP
Radius: select
Radius Port: 1812
Shared Secret: 1111

VPN > AutoKey Advanced > Gateway > EAP and perform the following actions:

IKEv2 EAP Authentication: (select)
Authenticator: (select)
Send ID: (select)
Query Config:

Select VPN > AutoKey Advanced > Gateway > EAP > New > IKEv2 Advanced and perform the following:

IKEv2 Auth Method: (select)
Self: select rsa-sig
Peer: select eap

VPNs > AutoKey Advanced > Gateway > EAP > MODECFG Enable: Perform the following actions, then click **OK**.

Server:
Action: select Add Route
Information Origin: select Use Local DNS
Profile: test

2. **Configuring IRAC to Get an IP Address from an EAP and External RADIUS Server:**

Select VPN > AutoKey Advanced > Gateway > EAP and perform the following actions:

IKEv2 EAP Authentication: (select)
Supplicant: (select)
User Name: temp
Password: temp

Select VPN > AutoKey Advanced > Gateway > EAP > New > IKEv2 > Advanced and perform the following:

IKEv2 Auth Method: (select)
Self: rsa-sig
Peer:eap

VPNs > AutoKey Advanced > Gateway > EAP > MODECFG Enable: Select **Client**, then click **OK**.

CLI

1. **Configuring IRAS to Get an IP Address from an EAP and External RADIUS Server:**

```
set auth-server test server-name 202.0.0.1
set auth-server test account-type eap-ikev2
set auth-server test radius port 1812
set auth-server test radius secret 1111
set ike gateway ikev2 gate_test eap authenticator passthrough test send-id-req
  query-config
set ike gateway ikev2 "v2-gw3" auth-method self rsa-sig peer eap
set ike gateway ikev2 gate_test modecfg server
```

2. **Configuring IRAC to Get an IP Address from an EAP and External Radius Server:**

```
set ike gateway ikev2 gate_test eap supplicant md5 username temp password
  temp
set ike gateway ikev2 gate_test eap supplicant md5 username temp password
  temp
set ike gateway ikev2 gate_test modecfg client
```

Example: Verifying if IRAS and IRAC receives the IP address

1. To verify if IRAS receives the IP address, use the get interface tunnel *inter_tun* CLI for route-based VPN as given below.

```
ssg5-serial-> get int t.1
...
tun.1 88.1.1.5/32 Untrust N/A - R -
## 2008-11-12 10:43:37 : IKE<20.1.1.2> ****** Recv packet if <ethernet0/0>
   of vsys <Root> ******
## 2008-11-12 10:43:37 : IKE<20.1.1.2> Catcher: get 252 bytes. src port 500
## 2008-11-12 10:43:37 : IKE<0.0.0.0 > found existing ike sa node 29319a8
## 2008-11-12 10:43:37 : IKE<20.1.1.2 > Search IKE_SA table, found 29319a8
## 2008-11-12 10:43:37 : Duplicated pkt checking ...
## 2008-11-12 10:43:37 : len in wind 288, hash in wind 1606591330, len 252,
   hash 0
## 2008-11-12 10:43:37 : hash in SA is a1dd69b2, len 252
## 2008-11-12 10:43:37 : IKE<0.0.0.0 > payload len 12, next payload
   type<39><AUTH>
## 2008-11-12 10:43:37 : IKE<0.0.0.0 > payload len 28, next payload
   type<47><CFG>
## 2008-11-12 10:43:37 : IKE<0.0.0.0 > payload len 12, next payload
   type<33><SA>
## 2008-11-12 10:43:37 : IKE<0.0.0.0 > payload len 80, next payload
   type<41><NOTIF>
## 2008-11-12 10:43:37 : IKE<0.0.0.0 > payload len 8, next payload
   type<41><NOTIF>
## 2008-11-12 10:43:37 : IKE<0.0.0.0 > payload len 8, next payload
   type<44><TSi>
## 2008-11-12 10:43:37 : IKE<0.0.0.0 > payload len 24, next payload
   type<45><TSr>
## 2008-11-12 10:43:37 : IKE<0.0.0.0 > payload len 24, next payload
   type<0><unknown>
## 2008-11-12 10:43:37 : start seq no is 1, avil seq no is 0, win size is 1
## 2008-11-12 10:43:37 : return seq no 1 as Responder
## 2008-11-12 10:43:37 : start seq no is 2, avil seq no is 0, win size is 1
## 2008-11-12 10:43:37 : IKE<0.0.0.0 > ISAKMP msg: ver 20, len 224, nxp 35
   exch 35[IKE SA AUTH], flag 08(I(1) R(0) V(0)),
...
## 2008-11-12 10:43:37 : Processing CP
## 2008-11-12 10:43:37 : Receive CP req: ip 0.0.0.0
...
## 2008-11-12 10:43:37 : Construct CP reply
## 2008-11-12 10:43:37 : Construct CP reply: ip addr 88.1.1.5
```

2. To verify if IRAC receives the IP address, use debug ike detail for policy-based VPN as given below.

```
debug ike detail
## 2008-11-12 11:10:27 : Enter IKEv2 init IKE_AUTH post processing, state 0
## 2008-11-12 11:10:27 : Constructing IKE_AUTH request
## 2008-11-12 11:10:27 : construct_ike_auth, send-auth 1 send_cp 1
## 2008-11-12 11:10:27 : Construct IKEv2 header.
## 2008-11-12 11:10:27 : Msg header built (next payload #0)
```

```
2008-11-12 11:10:27 : ID type 1, len 4.
## 2008-11-12 11:10:27 : initiator auth data len = 14.
## 2008-11-12 11:10:27 : Construct [CP] request: ip 0.0.0.0
## 2008-11-12 11:10:27 : Construct [SA] (CHILD_SA) request for IKEv2.
   conn->new_spi_r = 29504c0, &conn->new_spi_r = 29504c0.
## 2008-11-12 11:10:27 : Construct [SA] (CHILD_SA) request for IKEv2. spi =
   43242975
...
## 2008-11-12 11:10:27 : IKE<20.1.1.1> ****** Recv packet if <ethernet0/0>
   of vsys <Root> ******
## 2008-11-12 11:10:27 : IKE<20.1.1.1> Catcher: get 220 bytes. src port 500
## 2008-11-12 11:10:27 : IKE<0.0.0.0 > found existing ike sa node 28dbbe4
## 2008-11-12 11:10:27 : IKE<20.1.1.1 > Search IKE_SA table, found 28dbbe4.
## 2008-11-12 11:10:27 : IKE<0.0.0.0 > payload len 12, next payload
   type<39><AUTH>
## 2008-11-12 11:10:27 : IKE<0.0.0.0 > payload len 28, next payload
   type<47><CFG>
## 2008-11-12 11:10:27 : IKE<0.0.0.0 > payload len 16, next payload
   type<33><SA>
## 2008-11-12 11:10:27 : IKE<0.0.0.0 > payload len 40, next payload
   type<41><NOTIF>
## 2008-11-12 11:10:27 : IKE<0.0.0.0 > payload len 8, next payload
   type<41><NOTIF>
## 2008-11-12 11:10:27 : IKE<0.0.0.0 > payload len 8, next payload
   type<44><TSi>
## 2008-11-12 11:10:27 : IKE<0.0.0.0 > payload len 24, next payload
   type<45><TSr>
## 2008-11-12 11:10:27 : IKE<0.0.0.0 > payload len 24, next payload
   type<0><unknown<
## 2008-11-12 11:10:27 : start seq no is 1, avil seq no is 2, win size is 1
## 2008-11-12 11:10:27 : return seq no 1 as Initiator
## 2008-11-12 11:10:27 : start seq no is 2, avil seq no is 2, win size is 1
## 2008-11-12 11:10:27 : IKE<0.0.0.0 > ISAKMP msg: ver 20, len 188, nxp 36
   exch 35[IKE SA AUTH], flag 20(I(0) R(1) V(0)), msgid 1
...
## 2008-11-12 11:10:27 : Processing CP
## 2008-11-12 11:10:27 : Receive CP payload: ip 88.1.1.5.
## 2008-11-12 11:10:27 : ikmpd.c ike_if_ip_post_change_callback 6383, interface
   tunnel.1 pre change callabck for ike
## 2008-11-12 11:10:27 : update ike local addr sent: 3368
## 2008-11-12 11:10:27 : postprocess_ike_auth_resp: assign IP 88.1.1.5 from
   CP
```

CREATE_CHILD_SA Exchange

After the IPsec endpoints complete the initial exchanges, either endpoint can initiate the CREATE_CHILD_SA. This exchange rekeys a CHILD_SA or IKE_SA. This exchange consists of a single request/response pair and was referred to as a Phase 2 exchange in IKEv1.

All messages following the initial exchange are cryptographically protected using the cryptographic algorithms and keys negotiated in the first two messages of the IKE.

Informational Exchanges

IKEv2 uses informational exchanges to send and receive control messages, including dead peer detection (DPD).

Enabling IKEv2 on a Security Device

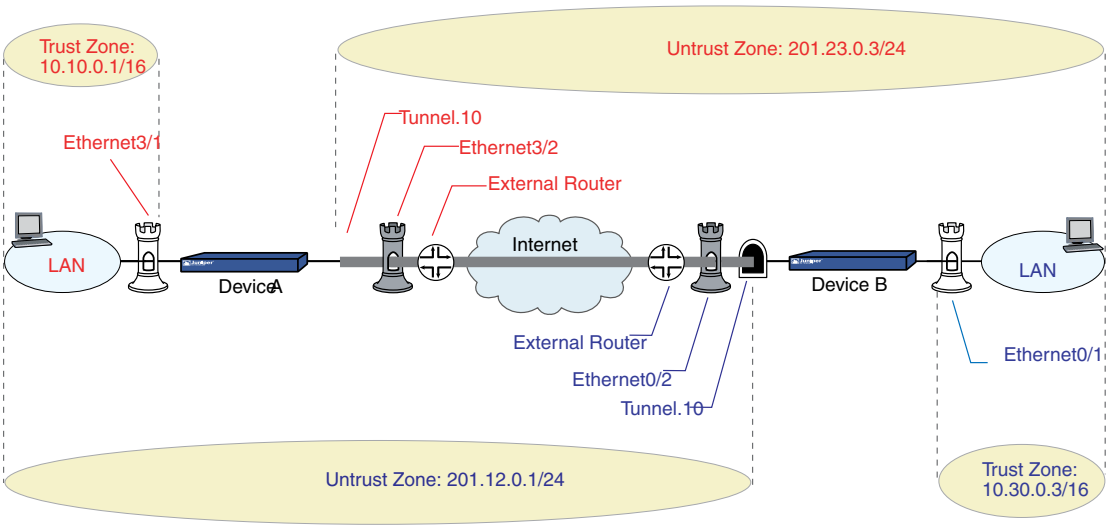
You can configure an existing IKEv1 gateway to support IKEv2. Such a converted gateway configuration functions only with IKEv2 peers, not IKEv1. When you configure your security device to support IKEv2, you should note the following differences between IKEv1 and IKEv2:

- Unlike IKEv1, where the IPsec endpoints negotiate the Diffie-Hellman (DH) group before agreeing on the DH group number, the IKEv2 initiator sends the DH group number in the first message of the IKE_INIT_SA exchange. If the initiator has multiple DH group proposals in its SA payload, the DH group that the initiator sends may not match the DH group the responder expects. In such cases, the responder notifies the initiator with the expected DH group number. The initiator responds to this message with the correct DH group number and restarts the IKE_INIT_SA exchange.
- The two endpoints in an IKEv2 SA do not negotiate the IKE_SA and CHILD_SA lifetimes; each endpoint can have its own lifetime. The endpoint with the shorter lifetime will rekey before the current IKE_SA or CHILD_SA expires (by default, 10 seconds earlier for IKE_SA and 60 seconds earlier for CHILD_SA), as long as the connection between the endpoints still needs this IKE_SA or CHILD_SA. A CHILD_SA is considered no longer needed when there has been no traffic since the last rekey or the SA has timed out. An IKE_SA is no longer needed when all its CHILD_SAs are no longer needed.
- Authentication methods between the two negotiating IKE peers can be different; the endpoints do not negotiate the authentication methods.
- All CHILD_SAs close if their parent IKE_SA is closed.
- The two endpoints maintain only one IKE_SA; all other exchanges are carried out through CHILD_SAs.

Example: Configuring an IKEv2 Gateway

In the example shown in Figure 204 on page 732, you create two VPN tunnels that use IKEv2 for automatic generation and negotiation of keys and security associations (SAs). These tunnels provide a secure connection between the two devices - Device B and Device A. A policy-based VPN is configured on Device A, while a route-based VPN is configured on Device B. For the Phase 1 and Phase 2 security levels, you specify standard and basic predefined proposals, respectively, on both the devices.

Figure 204: IKEv2 Gateway Connecting Two Security Devices



WebUI (Device A)

1. **Configuring the IKEv2 Gateway**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click *OK:*

   Gateway Name: Device B
   Version:
   IKEv2: (select)
   Remote Gateway:
   Static IP Address: (select), IPv4 Address/Hostname: 201.23.0.3

   > **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   IKEv2 Auth Method: Enable
   Self: rsa-sig
   Peer: preshare
   Preshared Key: GsbBPO0MNXYgXGsOetCXf8qaR8n5AUVILQ==
   Outgoing interface: ethernet3/2
   Security Level:
   Predefined: (select, Standard)
   Preferred Certificate (optional)
   Local cert: CN=but CN=nsisg2000.netscreen.com.CN=rsa-key.CN
   Peer CA: OU=Secure Server Certification Authority.O=RSA
   Peer Type: X509-SIG

2. **Configuring the VPN**

   VPNs > Autokey IKE > New: Enter the following, then click **Advanced:**

   VPN Name: Device B
   Remote Gateway: (select)
   Predefined: (select), Device B

> **Advanced:** Enter the following advanced settings, then click **Return** to set the advanced options and return to the basic configuration page:

Security Level
Predefined: (select, Basic)

3. **Configuring the Route**

Network > Routing > Routing Entries > Configuration: Enter the following, then click **OK:**

Virtual Router name: untrust-vr
IP Address / Netmask: 10.30.0.3/16
Next Hop: Gateway (select)
Interface: ethernet3/2

4. **Creating Policies**

Policy > Policies (From: Untrust, To: Trust) New: Enter the following, then click **OK:**

Source Address:
New Address: 10.30.0.3/16
Destination Address:
New address: 10.10.0.1/16
Service: (select), Any
Action: (select), Tunnel
Tunnel: (select), Device B

Policy > Policies (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address:
New Address: 10.10.0.1/16
Destination Address:
New address: 10.30.0.3/16
Service: (select), Any
Action: (select), Tunnel
Tunnel: (select), Device B

WebUI (Device B)

1. **Configuring the IKEv2 Gateway**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK:**

Gateway Name: Device A
Version:
IKEv2: (select)
Remote Gateway:
Static IP Address: (select), IPv4 Address/Hostname: 201.12.0.1

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

```
IKEv2 Auth Method: Enable
   Self: preshare
   Peer: rsa-sig
   Preshared Key: 3to5BAFpNn3thBsncQCmBYF5ThnQVfMIEQ==
   Outgoing interface: ethernet0/2
Security Level:
   Predefined: (select, Standard)
Preferred Certificate (optional)
   Local cert: None
   Peer CA: CN=netscreen.OU=qa
   Peer Type: X509-SIG
```

2. **Configuring the Tunnel Interface**

   Network > Interfaces > New: Enter the following, then click **OK:**

```
Tunnel Interface Name: tunnel.10
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
   Interface: ethernet0/2 (trust-vr)
```

3. **Configuring the VPN and Proxy-ID**

   VPNs > Autokey IKE > New: Enter the following, then click **Advanced:**

```
VPN Name: Device A
Remote Gateway: (select)
   Predefined: (select), Device A
```

> **Advanced:** Enter the following advanced settings, then click **Return** to set the advanced options and return to the basic configuration page:

```
Security Level
   Predefined: (select, Basic)
Bind to: Tunnel Interface (select): Select tunnel.10, Untrust-Tun from the
drop-down list
Proxy-ID: (select)
   Local IP / Netmask: 10.30.0.0/16
   Remote IP / Netmask: 10.10.0.0/16
   Service: ANY
```

4. **Configuring the Route**

   Network > Routing > Routing Entries > Configuration: Enter the following, then click **OK:**

```
Virtual Router name: trust-vr
IP Address / Netmask: 10.10.0.1/16
Next Hop: Gateway (select)
   Interface: tunnel.10
```

CLI (Device A)

1. **Configuring Addresses**

```
set address trust 10.10.0.1 10.10.0.1/16
set address untrust 10.30.0.3 10.30.0.3/16
```

2. **Configuring the IKEv2 Gateway**

```
set ike gateway ikev2 "Device B" address 201.23.0.3 outgoing-interface
"ethernet3/2" preshare "GsbBPO0MNXYgXGsOetCXf8qaR8n5AUVILQ==" proposal
 "standard"
set ike gateway ikev2 "Device B" auth-method self rsa-sig peer preshare
set ike gateway ikev2 "Device B" cert my-cert-hash
361A26F4CDE8696D10FF1C767D00AD8CCC3BF4CE
set ike gateway ikev2 "Device B" cert peer-ca-hash
0E9290B27AA8BAF65D3C9229AFE8F31DB953B2DA
```

NOTE: The local and peer certificates are generated by the device. The certificates will not work if you copy this part to the device.

3. **Setting the IKE_SA Soft Lifetime**

```
set ike ikev2 ike-sa-soft-lifetime 60
```

4. **Configuring the VPN**

```
set vpn "Device B" gateway "Device B" no-replay tunnel idletime 0 proposal
"basic"
```

5. **Configuring the Route**

```
set route 10.30.0.3/16 interface ethernet3/2
```

6. **Configuring Policies**

```
set policy id 4 from untrust to trust 10.30.0.3 10.10.0.1 any tunnel vpn "Device
 B" id 0x1 pair-policy 3
set policy id 3 from trust to untrust 10.10.0.1 10.30.0.3 any tunnel vpn "Device
 B" id 0x1 pair-policy 4
```

CLI (Device B)

1. **Configuring the IKEv2 Gateway**

```
set ike gateway ikev2 "Device A" address 201.12.0.1 outgoing-interface
"ethernet0/2" preshare "3to5BAFpNn3thBsncQCmBYF5ThnQVfMIEQ==" proposal
 "standard"
set ike gateway ikev2 "Device A" auth-method self preshare peer rsa-sig
set ike gateway ikev2 "Device A" cert peer-ca-hash
5BA819E4775F1DBAB039C48A0DAE21583DC5A916
```

2. **Configuring the VPN**

   ```
   set vpn "Device A" gateway "Device A" no-replay tunnel idletime 0 proposal
   "basic"
   ```

3. **Binding the VPN to a Tunnel**

   ```
   set vpn "Device A" id 0x2 bind interface tunnel.10
   ```

4. **Creating a VPN Proxy Configuration**

   ```
   set vpn "Device A" proxy-id local-ip 10.30.0.0/16 remote-ip 10.10.0.0/16 any
   ```

5. **Configuring the Route**

   ```
   set route 10.10.0.1/16 interface tunnel.10
   ```

6. **Setting Policy Permit**

   ```
   set policy id 4 from untrust to trust 10.30.0.3 10.10.0.1 any permit
   set policy id 3 from trust to untrust 10.10.0.1 10.30.0.3 any permit
   ```

Authentication Using Extensible Authentication Protocol

In addition to supporting authentication using public key signatures and shared secrets, IKEv2 supports authentication using Extensible Authentication Protocol (EAP). By using EAP, IKEv2 can leverage existing authentication infrastructure and credential databases, because EAP allows users to choose a method suitable for existing credentials and provides an easy means of separation of the IKEv2 responder (VPN gateway) from the RADIUS server that acts as the EAP authentication endpoint.

Juniper Networks security devices support authentication using EAP in the following ways:

- **Security device as the VPN gateway—**When the security device acts as the VPN gateway, it provides only EAP passthrough and supports a RADIUS server as the authentication server. In this implementation, the security device supports EAP-Message Digest 5 (EAP-MD5), EAP-Transport Layer Security (EAP-TLS), EAP-Tunneled TLS (EAP-TTLS), and EAP-Protected EAP (EAP-PEAP) passthrough. The security device neither times out for the connections nor provides accounting support.
- **Security device as the VPN client—**When the security device acts as the VPN client, it supports only the EAP-MD5 supplicant (client) functionality for IKEv2.

IKEv2 EAP Passthrough

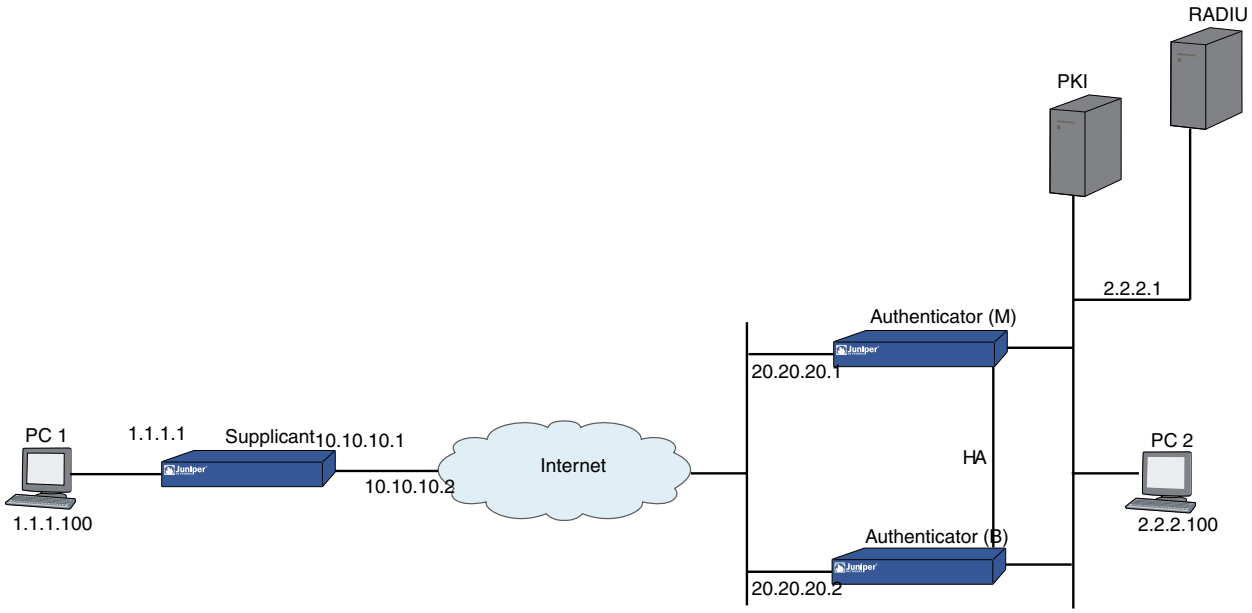
When you enable a Juniper Networks security device to use EAP to authenticate a client with a RADIUS authentication server, the security device acts as a proxy (authenticator) and passes the EAP messages between the client (supplicant) and the RADIUS (authentication) server. During EAP exchanges, the security device decapsulates the EAP messages in IKEv2 messages from the peer, encapsulates them into RADIUS messages, and sends them to the RADIUS server.

When the RADIUS server responds to the authentication requests, the security device decapsulates the EAP messages, encapsulates them into IKEv2 messages, and sends them to the peer. After the RADIUS server has authenticated the client, if there is a shared secret generated during the exchange, the security device extracts the shared secret from the RADIUS Access-Accept message and uses it to generate the AUTH payload. In this way, the security device passes the EAP messages between a client and an authentication server.

Example

The following example explains the steps involved in setting up IKEv2 EAP authentication for an authenticator and a supplicant.

Figure 205: Setting Up IKEv2 EAP Authentication



You can set up the IKEv2 EAP using the WebUI or the CLI.

WebUI (Authenticator)

1. **Setting Up Auth-Server**

   Select Configuration > Auth > Auth Servers > **New:** Enter the following, then click **OK:**

   Name: rad1
   Auth-server IP address: 10.155.43.201
   RADIUS secret: netscreen
   Account Type: IKEv2EAP (check)

2. **Setting Up IKE**

Select VPN > AutoKey Advanced > Gateway > New: Enter the following, then click OK:

Gateway Name: v2-gw3
Version: IKEv2 (select)
IP address of the remote gateway: 10.10.10.1

> Click **Advanced.** Configure the following advanced setting, then click **Return** to return to the basic Gateway configuration page:

Phase1 Proposal (select): rsa-g2-3des-sha

Select VPN > AutoKey Advanced > Gateway > **EAP:** Perform the following actions:

IKEv2 EAP authentication: (check)
Authenticator: (select)
Auth-server name: rad1
Send-id-Req: (check)

Select VPN > AutoKey Advanced > Gateway > **Edit:** Perform the following actions, then click **OK:**

edit on the gateway (select)

> Click **Advanced.** Configure the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

IKEv2 Auth: (check)
self: rsa-sig
peer:eap
Outgoing interface: loopback.3
Local cert: my-cert 1

CLI (Authenticator)

1. **Auth-Server Configuration**

```
set auth-server "rad1" server-name "10.155.43.201"
set auth-server "rad1" account-type eap-ikev2
set auth-server "rad1" radius secret netscreen
```

2. **IKE Configuration**

```
set ike gateway ikev2 "v2-gw3" dialup "Peer2" outgoing-interface "loopback.3"
 preshare abcd1234 proposal "rsa-g2-3des-sha"
set ike gateway ikev2 "v2-gw3" cert my-cert 1
set ike gateway ikev2 "v2-gw3" cert peer-ca all
set ike gateway ikev2 v2-gw3 eap authenticator passthrough rad1 send-id-req
set ike gateway ikev2 "v2-gw3" auth-method self rsa-sig peer eap
set vpn "v2-vpn3" gateway "v2-gw3" no-replay tunnel idletime 0 proposal
"g2-esp-3des-sha"
```

WebUI (Supplicant)

1. **Setting Up IKE**

   Select VPN > AutoKey Advanced > Gateway > **New:** Enter the following, then click **OK:**

   Gateway Name: v2-gw3
   Version: IKEv2 (select)
   IP address of the remote gateway: 20.20.20.1

   > Click **Advanced.** Configure the following advanced setting, then click **Return** to return to the basic Gateway configuration page:

   Phase1 Proposal (select): rsa-g2-3des-sha

   Select VPN > AutoKey Advanced > Gateway > **Edit:** Perform the following actions, then click **OK:**

   Gateway: (select)

   > Click **Advanced.** Configure the following advanced setting, then click **Return** to return to the basic Gateway configuration page:

   IKEv2 Auth: (check)
   self: eap
   peer: rsa-sig
   Outgoing interface: loopback.3
   Local cert: my-cert 1

CLI (Supplicant)

```
set ike gateway ikev2 "v2-gw3" address 203.203.203.1 local-id
"Peer2@spg.juniper.net" outgoing-interface "loopback.3" preshare abcd1234 proposal
 "rsa-g2-3des-sha"
set ike gateway ikev2 "v2-gw3" cert my-cert 1
set ike gateway ikev2 "v2-gw3" cert peer-ca all
set ike gateway ikev2 v2-gw3 eap supplicant md5 username test1 password abcd1
set ike gateway ikev2 "v2-gw3" auth-method self eap peer rsa-sig
set vpn "v2-vpn3" gateway "v2-gw3" no-replay tunnel idleitem 0 proposal
"g2-esp-3des-sha"
```

Chapter 20

Public Key Cryptography

This chapter provides an introduction to public key cryptography and the use of certificates and certificate revocation lists (CRLs) within the context of Public Key Infrastructure (PKI). This chapter includes the following topics:

Introduction to Public Key Cryptography

In public key cryptography, a public/private key pair is used to encrypt and decrypt data. Data encrypted with a public key, which the owner makes available to the public, can only be decrypted with the corresponding private key, which the owner keeps secret and protected. For example, if Alice wants to send Bob an encrypted message, Alice can encrypt it with Bob's public key and send it to him. Bob then decrypts the message with his private key.

The reverse is also useful; that is, encrypting data with a private key and decrypting it with the corresponding public key. This is known as creating a digital signature. For example, if Alice wants to present her identity as the sender of a message, she can encrypt the message with her private key and send the message to Bob. Bob then decrypts the message with Alice's public key, thus verifying that Alice is indeed the sender.

Public/private key pairs also play an important role in the use of digital certificates. The procedure for signing a certificate (by a CA) and then verifying the signature (by the recipient) works as shown in the following subsections.

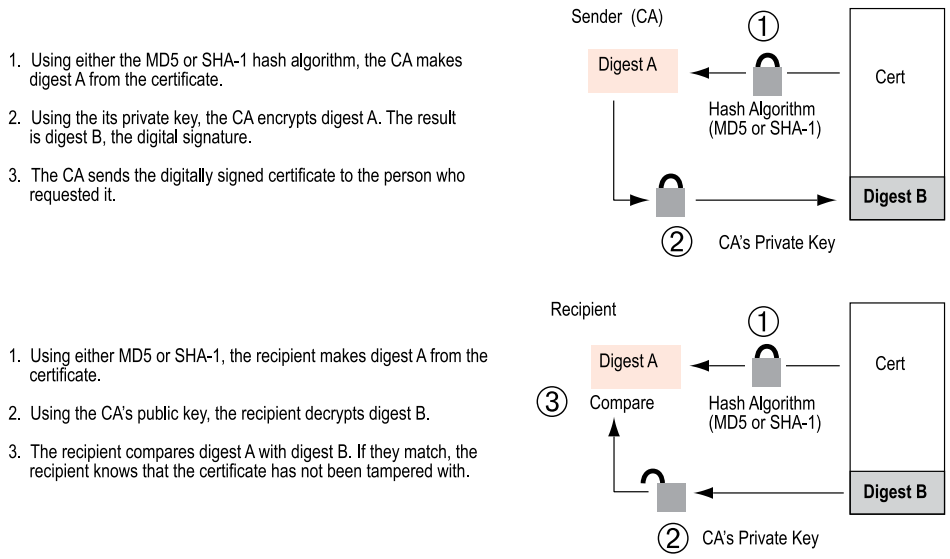
Signing a Certificate

1. The certificate authority (CA) that issues a certificate hashes the certificate by using a hash algorithm (MD5, SHA-1 or SHA2-256) to generate a digest.
2. The CA then "signs" the certificate by encrypting the digest with its private key. The result is a digital signature.
3. The CA then sends the digitally signed certificate to the person who requested it.

Verifying a Digital Signature

1. When the recipient gets the certificate, the recipient also generates another digest by applying the same hash algorithm (MD5, SHA-1 or SHA2-256) on the certificate file.
2. The recipient uses the CA's public key to decrypt the digital signature.
3. The recipient compares the decrypted digest with the digest just generated. If the two match, the recipient can confirm the integrity of the CA's signature and, by extension, the integrity of the accompanying certificate.

Figure 206: Digital Signature Verification



The procedure for digitally signing messages sent between two participants in an IKE session works very similarly, with the following differences:

- Instead of making a digest from the CA certificate, the sender makes it from the data in the IP packet payload.
- Instead of using the CA's public/private key pair, the participants use the sender's public/private key pair.

Elliptic Curve Digital Signature Algorithm

Juniper Networks security devices use Elliptic Curve Cryptography (ECC) to generate the Elliptic Curve Digital Signature Algorithm (ECDSA) key pair.

In addition to RSA and DSA, you can also generate an ECDSA public/private key pair using ECDSA. The public key size of an ECDSA key is smaller than a DSA public key. The performance speed of an ECDSA key, at higher security levels, is faster than DSA or RSA. For information about ECDSA, see RFCs 3279 and 4754.

Like DSA and RSA certificates, you can use IKEv1 with ECDSA-based certificates (see RFC 2409). You can use three different ECDSA signatures with IKEv1. Each of these signatures uses a particular elliptic curve group and hash function (see RFC 4753).

Digital Signature Algorithm	Elliptic Curve Group	Hash Function
ECDSA-256	256-bit	SHA-256
ECDSA-384	384-bit	SHA-384
ECDSA-512	512-bit	SHA-512

The current version of ScreenOS uses the SHA2-256 hashing algorithm and supports the secp256r1 parameter type of elliptic curve only.

To generate an ECDSA public/private key pair:

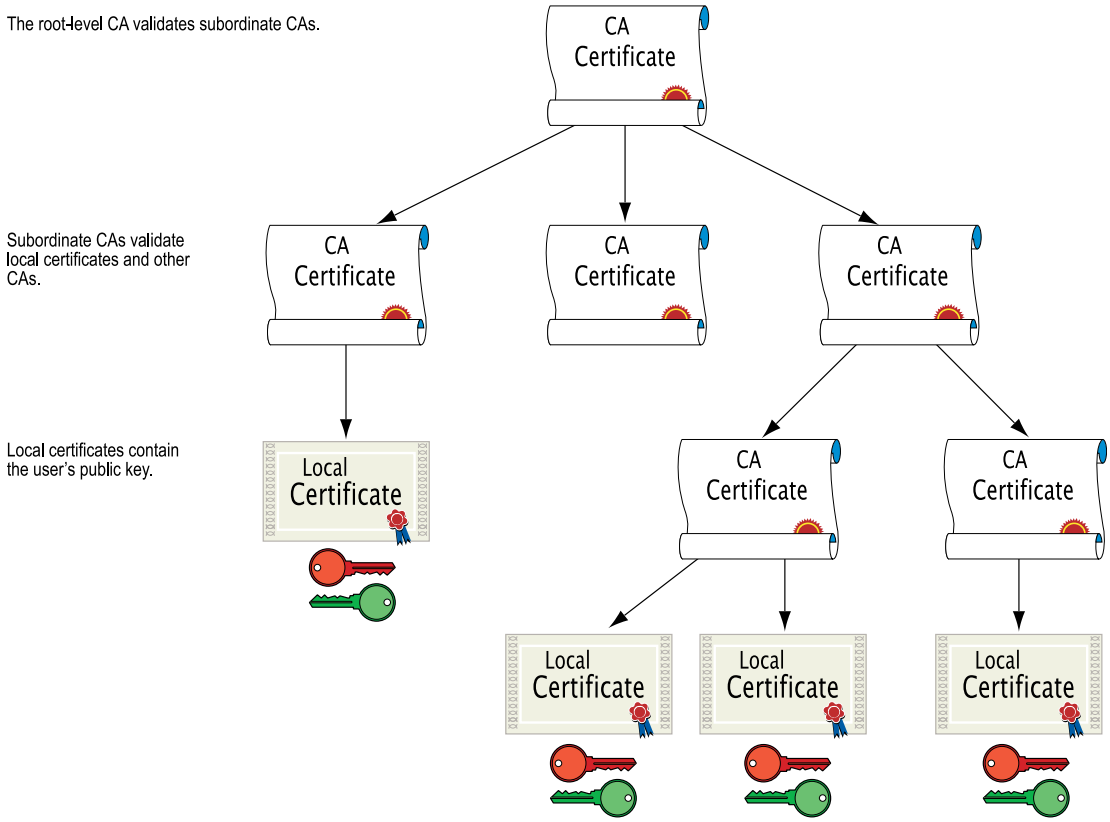
```
exec pki ecdsa new-key secp256r1
```

The ECDSA key length is defined in the elliptic curve domain parameter string **secp256r1.** The curve domain parameters conform to ANSI X9.62-1998 specifications.

Public Key Infrastructure

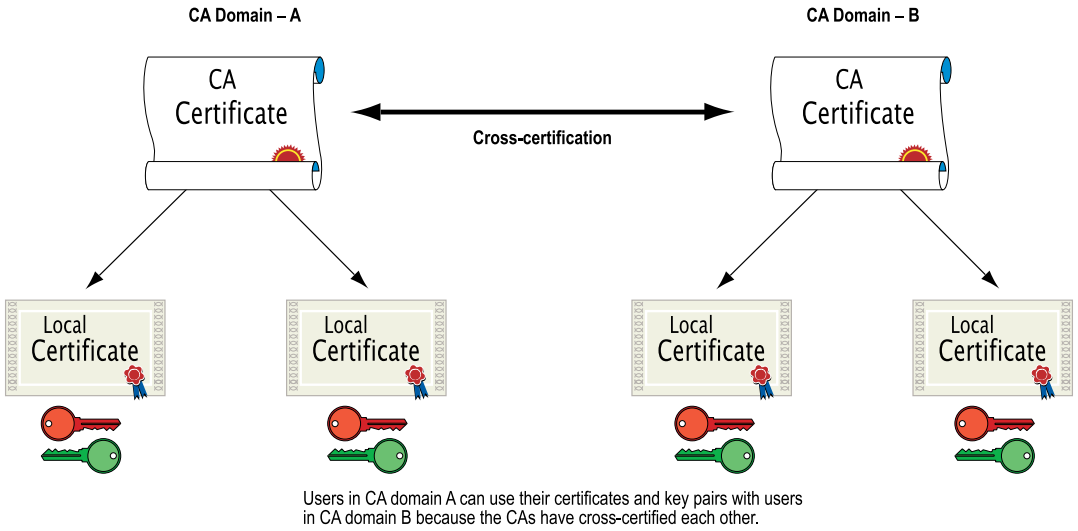
Public Key Infrastructure (PKI) refers to the hierarchical structure of trust required for the successful implementation of public key cryptography. To verify the trustworthiness of a certificate, you must be able to track a path of certified CAs from the one issuing your local certificate back to a root authority of a CA domain.

Figure 207: PKI Hierarchy of Trust—CA Domain



If certificates are used solely within an organization, that organization can have its own CA domain within which a company CA issues and validates certificates among its employees. If that organization later wants its employees to be able to exchange their certificates with those from another CA domain (for example, with employees at another organization that also has its own CA domain), the two CAs can develop cross-certification; that is, they can agree to trust the authority of each other. In this case, the PKI structure does not extend vertically but does extend horizontally.

Figure 208: Cross-Certification



For convenience and practicality, the PKI must be transparently managed and implemented. Toward this goal, ScreenOS does the following:

1. Generates a public/private key pair when you create a certificate request.
2. Supplies that public key as part of the certificate request in the form of a text file for transmission to a Certificate Authority (CA) for certificate enrollment (PKCS10 file).
3. Supports loading the local certificate, the CA certificate, and the certificate revocation list (SubinterfaceCRL) into the unit.

NOTE: The Certificate Authority usually provides a CRL. Although you can load a CRL into the security device, you cannot view it once loaded.

You can also specify an interval for refreshing the CRL online. For more information about CRLs, see “Certificates and CRLs” on page 746.

4. Provides certificate delivery when establishing an IPsec tunnel.
5. Supports certificate path validation upward through eight levels of CA authorities in the PKI hierarchy.
6. Supports the PKCS #7 cryptographic standard, which means the security device can accept X.509 certificates and CRLs packaged within a PKCS #7 envelope. PKCS #7 support allows you to submit multiple X.509 certificates within a single PKI request. You can now configure PKI to validate all the submitted certificates from the issuing CA at one time.

NOTE: ScreenOS supports a PKCS #7 file size of up to 7 Kilobytes.

7. Supports online CRL retrieval through LDAP or HTTP.

Certificates and CRLs

A digital certificate is an electronic means for verifying your identity through the word of a trusted third party, known as a Certificate Authority (CA). The CA server you use can be owned and operated by an independent CA or by your own organization, in which case you become your own CA. If you use an independent CA, you must contact them for the addresses of their CA and CRL servers (for obtaining certificates and certificate revocation lists) and for the information they require when submitting personal certificate requests. When you are your own CA, you determine this information yourself.

NOTE: ScreenOS supports the following CAs: Baltimore, Entrust, Microsoft, Netscape, RSA Keon, and Verisign.

ScreenOS contains a CA certificate for authenticating downloads from the antivirus (AV) pattern file server and the Deep Inspection (DI) attack object database server. For more information about the AV pattern file server, see "Antivirus Scanning" on page 211. For more information about the DI attack object database server, see "Attack Object Database Server" on page 566.

To use a digital certificate to authenticate your identity when establishing a secure VPN connection, you must first do the following:

- Generate a key in the security device, send it to a CA to obtain a personal certificate (also known as a *local* certificate), and load the certificate in the security device.
- Obtain a CA certificate for the CA that issued the personal certificate (basically verifying the identity of the CA verifying you), and load the CA certificate in the security device. You can perform this task manually, or you can perform this task automatically using Simple Certificate Enrollment Protocol (SCEP).
- If the certificate does not contain a certificate distribution point (CDP) extension, and you cannot automatically retrieve the CRL through LDAP or HTTP, you can retrieve a CRL manually and load that in the security device.

During the course of business, there are several events that make it necessary to revoke a certificate. You might wish to revoke a certificate if you suspect that it has been compromised or when a certificate holder leaves a company. Managing certificate revocations and validation can be accomplished locally (which is a limited solution) or by referencing a CA's CRL, which you can automatically access online at daily, weekly, or monthly intervals or at the default interval set by the CA.

To obtain a signed digital certificate using the manual method, you must complete several tasks in the following order:

1. Generate a public/private key pair.
2. Fill out the certificate request.

3. Submit your request to your CA of choice.
4. After you receive your signed certificate, you must load it into the security device along with the CA certificate.

You now have the following items for the following uses:

- A local certificate for the security device, to authenticate your identity with each tunnel connection
- A CA Certificate (their public key), to be used to verify the peer's certificate
- If the Certificate Revocation List (CRL) was included with the CA certificate, a CRL to identify invalid certificates

NOTE: A CRL might accompany a CA certificate and be stored in the ScreenOS database. Alternatively, the CA certificate might contain the CRL URL (either LDAP or HTTP) for a CRL that is stored in the CA's database. If the CRL is unobtainable by either method, you can manually enter a CRL URL in the security device, as explained in "Configuring CRL Settings" on page 751.

When you receive these files (the certificate files typically have the extension .cer, and the CRL typically has the extension .crl), load them into your security device using the procedure described in "Requesting a Certificate Manually" on page 747.

NOTE: If you are planning to use email to submit a PKCS10 file to obtain your certificates, you must properly configure your ScreenOS settings so that you can send email to your system administrator. You have to set your primary and secondary DNS servers and specify the SMTP server and email address settings.

Requesting a Certificate Manually

When you request a certificate, the security device generates a key pair. The public key becomes incorporated in the request itself and, eventually, in the digitally signed local certificate you receive from the CA.

In the following example, the security administrator is making a certificate request for Michael Zhang in the Development department at Juniper Networks in Sunnyvale, California. The certificate is going to be used for a security device at IP address 10.10.5.44. The administrator instructs the security device to send the request through email to the security administrator at *admin@juniper.net*. The security administrator then copies and pastes the request in the certificate request text field at the CA's certificate enrollment site. After the enrollment process is complete, the CA usually sends the certificates through email back to the security administrator.

NOTE: A special certificate identity string, called *domain-component*, is available only through the CLI. Devices can use this value in certificates for IPsec logon to VPN gateways. For example, the device could use this as a Group IKE ID, accepting ASN1_DN type IKE identities containing "DC = Engineering, DC = NewYork".

Before generating a certificate request, make sure that you have set the system clock and assigned a hostname and domain name to the security device. (If the security device is in an NSRP cluster, replace the hostname with a cluster name. For more information, see "Creating an NSRP Cluster" on page 1796.)

WebUI

1. **Certificate Generation**

   Objects > Certificates > New: Enter the following, then click **Generate**:

   Name: Michael Zhang
   Phone: 408-730-6000
   Unit/Department: Development
   Organization: Juniper Networks
   County/Locality: Sunnyvale
   State: CA
   Country: US
   E-mail: mzhang@juniper.net
   IP Address: 10.10.5.44
   Write to file: (select)
   RSA: (select)
   Create new key pair of *1024* length: (select)

   The device generates a PKCS #10 file and prompts you to send the file through email, save the file to disk, or automatically enroll through the Simple Certificate Enrollment Protocol (SCEP).

   Select the **E-mail to** option, type **admin@juniper.net**, then click **OK**.

NOTE: Some CAs do not support an email address in a certificate. If you do not include an email address in the local certificate request, you cannot use an email address as the local IKE ID when configuring the security device as a dynamic peer. Instead, you can use a fully qualified domain name (if it is in the local certificate), or you can leave the local ID field empty. By default the security device sends its *hostname.domainname*. If you do not specify a local ID for a dynamic peer, enter the *hostname.domainname* of that peer on the device at the other end of the IPsec tunnel in the peer ID field.

The value 1024 indicates the bit length of the key pair. If you are using the certificate for SSL (see "Secure Sockets Layer" on page 315), be sure to use a bit length that your browser also supports.

Using the email address assumes that you have already configured the IP address for your SMTP server: **set admin mail server-name** { *ip_addr* | *dom_name* }.

2. **Certificate Request**

   The security administrator opens the file and copies its contents, taking care to copy the entire text but not any blank spaces before or after the text. (Start at "*-----*BEGIN CERTIFICATE REQUEST*-----*", and end at "*-----*END CERTIFICATE REQUEST*-----*".)

   The security administrator then follows the certificate request directions at the CA's website, pasting the PKCS #10 file in the appropriate field when required.

3. **Certificate Retrieval**

   When the security administrator receives the certificate from the CA through email, the administrator forwards it to you. Copy it to a text file, and save it to your workstation (to be loaded to the security device later through the WebUI) or to a TFTP server (to be loaded later through the CLI).

CLI

1. **Certificate Generation**

```
set pki x509 dn country-name US
set pki x509 dn email mzhang@juniper.net
set pki x509 dn ip 10.10.5.44
set pki x509 dn local-name "Santa Clara"
set pki x509 dn name " Michael Zhang"
set pki x509 dn org-name " Juniper Networks"
set pki x509 dn org-unit-name Development
set pki x509 phone 408-730-6000
set pki x509 dn state-name CA
set pki x509 default send-to admin@juniper.net
exec pki rsa new-key 1024
```

NOTE: Using the email address assumes that you have already configured the IP address for your SMTP server: **set admin mail server-name** { *ip_addr* | *dom_name* }.

The certificate request is sent through email to admin@juniper.net.

2. **Certificate Request**

   The security administrator opens the file and copies its contents, taking care to copy the entire text but not any blank spaces before or after the text. (Start at "-----BEGIN CERTIFICATE REQUEST-----", and end at " -----END CERTIFICATE REQUEST-----".)

   The security administrator then follows the certificate request directions at the CA's website, pasting the PKCS #10 file in the appropriate field when required.

3. **Certificate Retrieval**

   When the security administrator receives the certificate from the CA through email, the administrator forwards it to you. Copy it to a text file, and save it to your workstation (to be loaded to the security device later through the WebUI) or to a TFTP server (to be loaded later through the CLI).

Loading Certificates and Certificate Revocation Lists

The CA returns the following three files to you for loading onto the security device:

- A CA certificate, which contains the CA's public key
- A local certificate that identifies your local machine (your public key)
- A CRL, which lists any certificates revoked by the CA

For the WebUI example, you have downloaded the files to a directory named C:\certs\ns on the administrator's workstation. For the CLI example, you have downloaded the TFTP root directory on a TFTP server with IP address 198.168.1.5.

NOTE: Juniper Networks security devices (including virtual systems) configured with ScreenOS 2.5 or later support loading multiple local certificates from different CAs.

This example illustrates how to load two certificate files named auth.cer (CA certificate) and local.cer (your public key), along with the CRL file named distrust.crl.

WebUI

1. Objects > Certificates: Select **Load Cert**, then click **Browse**.
2. Navigate to the C:\certs directory, select **auth.cer**, then click **Open**.

   The directory path and filename (C:\certs\ns\auth.cer) appear in the File Browse field.

3. Click **Load**.

   The auth.cer certificate file loads.

4. Objects > Certificates: Select **Load Cert**, then click **Browse**.
5. Navigate to the C:\certs directory, select **local.cer**, then click **Open**.

   The directory path and filename (C:\certs\ns\local.cer) appear in the File Browse field.

6. Click **Load**.

   The auth.cer certificate file loads.

7. Objects > Certificates: Select **Load CRL**, then click **Browse**.
8. Navigate to the C:\certs directory, select **distrust.crl**, then click **Open**.
9. Click **Load**.

   The distrust.crl CRL file loads.

CLI

```
exec pki x509 tftp 198.168.1.5 cert-name auth.cer
exec pki x509 tftp 198.168.1.5 cert-name local.cer
exec pki x509 tftp 198.168.1.5 crl-name distrust.crl
```

Configuring CRL Settings

In Phase 1 negotiations, participants check the CRL list to see if certificates received during an IKE exchange are still valid. If a CRL is not loaded in the ScreenOS database, the security device tries to retrieve the CRL through the LDAP or HTTP CRL location defined within the certificate itself. If there is no URL defined in the certificate, the security device uses the URL of the server that you define for that CA certificate. If you do not have the CA certificate loaded in the device (for example, an intermediate CA of the certificate chain received during IKE exchange), you cannot resolve the CRL server URL for that CA. In this case, you can specify the CRL server URL in the Default Cert Validation Settings section of the WebUI (see the next page). A CRL server URL entered here is used only when the CA certificate is not present in the device. There is no pre-defined default URL.

NOTE: The CRL distribution point extension (.cdp) in an X509 certificate can be either an HTTP URL or an LDAP URL.

With ScreenOS 2.5 and later, you can disable the checking of a CRL's digital signature when you load the CRL. However, disabling CRL certificate checking compromises the security of your device.

In this example, you first configure the Entrust CA server to check the CRL daily by connecting to the LDAP server at 2.2.2.121 and locating the CRL file. You then

configure default certificate-validation settings to use the company's LDAP server at 10.1.1.200, also checking the CRL every day.

NOTE: The index (IDX) number for the Entrust CA certificate is 1. To view a list of the IDX numbers for all the CA certificates loaded on a security device, use the following CLI command: **get pki x509 list ca-cert**.

WebUI

Objects > Certificates (Show: CA) > Server Settings (for NetScreen): Enter the following, then click **OK**:

```
X509 Cert_Path Validation Level: Full
CRL Settings:
   URL Address: ldap:///CN=Entrust,CN=en2001,CN=PublicKeyServices,
CN=Services,CN=Configuration,DC=EN2001,DC=com?CertificateRevocation
List?base?objectclass=CRLDistributionPoint
   LDAP Server: 2.2.2.121
   Refresh Frequency: Daily
```

Objects > Certificates > Default Cert Validation Settings: Enter the following, then click **OK**:

```
X509 Certificate Path Validation Level: Full
Certificate Revocation Settings:
   Check Method: CRL
   URL Address: ldap:///CN=NetScreen,CN=safecert,CN=PublicKeyServices,
CN=Services,CN=Configuration,DC=SAFECERT,DC=com?CertificateRevocation
List?base?objectclass=CRLDistributionPoint
   LDAP Server: 10.1.1.200
```

CLI

```
set pki authority 1 cert-path full
set pki authority 1 cert-status crl url
"ldap:///CN=Entrust,CN=en2001,CN=PublicKeyServices,CN=Services,CN=Configuration,DC=EN2000,DC=com?
 CertificateRevocationList?base?objectclass=CRLDistributionPoint"
set pki authority 1 cert-status crl server-name 2.2.2.121
set pki authority 1 cert-status crl refresh daily
set pki authority default cert-path full
set pki authority default cert-status crl url
"ldap:///CN=NetScreen,CN=safecert,CN=PublicKeyServices,
CN=Services,CN=Configuration,DC=SAFECERT,DC=com?
CertificateRevocationList?base?objectclass=CRLDistributionPoint"
set pki authority default cert-status crl server-name 10.1.1.200
set pki authority default cert-status crl refresh daily
save
```

Obtaining a Local Certificate Automatically

To use a digital certificate to authenticate your identity when establishing a secure VPN connection, you must first do the following:

- Obtain a certificate authority (CA) certificate from which you intend to obtain a personal certificate, and then load the CA certificate in the security device.
- Obtain a local certificate (also known as a personal certificate) from the CA whose CA certificate you have previously loaded, and then load the local certificate in the security device. You can perform this task manually, or automatically using Simple Certificate Enrollment Protocol (SCEP).

Because the manual method of requesting local certificates has steps requiring you to copy information from one certificate to another, it can be a somewhat lengthy process. To bypass these steps, you can use the automatic method.

Note that, before using SCEP, you must perform the following tasks:

- Configure and enable DNS. (See "Domain Name System Support" on page 263.)
- Set the system clock. (See "System Clock" on page 301.)
- Assign a hostname and domain name to the security device. (If the security device is in an NSRP cluster, replace the hostname with a cluster name. For more information, see"Creating an NSRP Cluster" on page 1796.)

In this example, you use the automatic method to request a local certificate. You use SCEP with the Verisign CA. You set the following CA settings:

- Full certificate path validation
- RA CGI: http://ipsec.verisign.com/cgi-bin/pkiclient.exe

NOTE: The Common Gateway Interface (CGI) is a standard way for a Web server to pass a user request to an application program and to receive data back. CGI is part of the HyperText Transfer Protocol (HTTP). You must specify an RA CGI path even if the RA does not exist. If the RA does not exist, use the value specified for the CA CGI.

- CA CGI: http://ipsec.verisign.com/cgi-bin/pkiclient.exe
- Automatic integrity confirmation of CA certificates
- CA ID, which identifies a SCEP server, where Verisign SCEP server uses a domain name, such as juniper.net or a domain set up by Verisign for your company
- Challenge password
- Automatic certificate polling every 30 minutes (the default is no polling)

You then generate an RSA key pair, specifying a key length of 1024 bits, and initiate the SCEP operation to request a local certificate from the Verisign CA with the above CA settings.

When using the WebUI, you refer to CA certificates by name. When using the CLI, you refer to CA certificates by index (IDX) number. In this example, the IDX number for the Verisign CA is "1." To see the IDX numbers for CA certificates, use the following command: **get pki x509 list ca-cert**. The output displays an IDX number

and an ID number for each certificate. Note the IDX number and use that when referencing the CA certificate in commands.

WebUI

1. **CA Server Settings**

   Objects > Certificates > Show CA > Server Settings (for Verisign): Enter the following, then click **OK**:

   X509 certificate path validation level: Full
   SCEP Settings:
   RA CGI: http://ipsec.verisign.com/cgi-bin/pkiclient.exe
   CA CGI: http://ipsec.verisign.com/cgi-bin/pkiclient.exe

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic CA Server Settings configuration page:

   Polling Interval: 30
   Certificate Authentication: Auto
   Certificate Renew: 14

2. **Local Certificate Request**

   Objects > Certificates > New: Enter the following, then click **Generate**:

   Name: Michael Zhang
   Phone: 408-730-6000
   Unit/Department: Development
   Organization: Juniper Networks
   County/Locality: Sunnyvale
   State: CA
   Country: US
   Email: mzhang@juniper.net
   IP Address: 10.10.5.44
   Key Pair Information
   RSA: (select)
   Create new key pair of **1024** length.

NOTE: The value 1024 indicates the bit length of the key pair. If you are using the certificate for SSL, be sure to use a bit length that your browser also supports.

   Issue the **get pki x509 pkcs** CLI command to have the security device generate a PKCS #10 file and then, do one of the following:

   - Send the PKCS #10 certificate request file to an email address
   - Save it to disk
   - Automatically enroll by sending the file to a CA that supports the Simple Certificate Enrollment Protocol (SCEP)

3. **Automatic Enrollment**

Select the **Automatically enroll to** option, select the **Existing CA server settings** option, then select **Verisign** from the drop-down list.

Contact Verisign to inform them of your certificate request. They must authorize the certificate request before you can download the certificate.

CLI

1. **CA Server Settings**

```
set pki authority 1 cert-path full
set pki authority 1 scep ca-cgi "http://ipsec.verisign.com/cgi-bin/pkiclient.exe"

set pki authority 1 scep ra-cgi "http://ipsec.verisign.com/cgi-bin/pkiclient.exe"

set pki authority 1 scep polling-int 30
set pki authority 1 scep renew-start 14
```

NOTE: The Common Gateway Interface (CGI) is a standard way for a Web server to pass a user request to an application program and to receive data back. CGI is part of the HyperText Transfer Protocol (HTTP).

You must specify an RA CGI path even if the RA does not exist. If the RA does not exist, use the value specified for the CA CGI.

2. **Local Certificate Request**

```
set pki x509 dn country-name US
set pki x509 dn email mzhang@juniper.net
set pki x509 dn ip 10.10.5.44
set pki x509 dn local-name "Santa Clara"
set pki x509 dn name "Michael Zhang"
set pki x509 dn org-name "Juniper Networks"
set pki x509 dn org-unit-name Development
set pki x509 phone 408-730-6000
set pki x509 dn state-name CA
exec pki rsa new 1024
```

3. **Automatic Enrollment**

```
exec pki x509 scep 1
```

If this is the first certificate request from this CA, a prompt appears presenting a fingerprint value for the CA certificate. You must contact the CA to confirm that this is the correct CA certificate.

Contact Verisign to inform them of your certificate request. They must authorize the certificate request before you can download the certificate.

Automatic Certificate Renewal

You can enable the security device to automatically renew certificates it acquired through Simple Certificate Enrollment Protocol (SCEP). This feature saves you from having to remember to renew certificates on the security device before they expire, and, by the same token, helps maintain valid certificates at all times.

This feature is disabled by default. You can configure the security device to automatically send out a request to renew a certificate before it expires. You can set the time when you want the security device to send out the certificate renewal request in number of days and minutes before the expiration date. By setting different times for each certificate, you prevent the security device from having to renew all certificates at the same time.

For this feature to work, the security device must be able to reach the SCEP server, and the certificate must be present on the security device during the renewal process. Furthermore, for this feature to work, you must also ensure that the CA issuing the certificate can do the following:

- Support automatic approval of certificate requests.
- Return the same DN (Domain Name). In other words, the CA cannot modify the subject name and SubjectAltName extension in the new certificate.

You can enable and disable the automatic SCEP certificate-renewal feature for all SCEP certificates or for individual certificates.

Key-Pair Generation

A security device holds pregenerated keys in memory. The number of pregenerated keys depends on the device model. During normal operation, the security device can manage to have enough keys available to renew certificates by generating a new key every time it uses one. The process of generating a key usually goes unnoticed as the device has time to generate a new key before one is needed. In the event that the security device renews a great number of certificates at once, thus using up keys rapidly, it might run out of pregenerated keys and have to generate them promptly for each new request. In this case, the generation of keys might affect the performance of the security device, especially in a high-availability (HA) environment where the performance of the security device might slow down for a number of minutes.

The number of pregenerated key pairs on a security device depends on the model. For more information, refer to the datasheet for your Juniper Networks security product.

Online Certificate Status Protocol

When a security device performs an operation that uses a certificate, it is usually important to verify the validity of that certificate. Certificates might have become invalid through expiration or revocation. The default way to check the status of certificates is to use certificate revocation lists (CRLs). The Online Certificate Status Protocol (OCSP) is an alternative way to check the status of certificates. OCSP can

provide additional information about certificates and provide status checks in a more timely manner.

When a security device uses OCSP, it is referred to as the *OCSP client* (or *requester*). This client sends a verification request to a server device called the *OCSP responder.* ScreenOS supports RSA Keon and Verisign as OCSP responders. The client's request contains the identity of the certificate to check. Before the security device can perform any OCSP operation, you must configure it to recognize the location of the OCSP responder.

NOTE: We have also successfully tested with the Valicert OCSP responder during an extensive evaluation in the past.

After receiving the request, the OCSP responder confirms that the status information for the certificate is available, then returns the current status to the security device. The response of the OCSP responder contains the certificate's revocation status, the name of the responder, and the validity interval of the response. Unless the response is an error message, the responder signs the response using the responder's private key. The security device verifies the validity of the responder's signature by using the certificate of the responder. The certificate of the responder may either be embedded in the OCSP response, or stored locally and specified in the OCSP configuration. If the certificate is stored locally, use the following command to specify the locally stored certificate:

set pki authority *id_num1* **cert-status ocsp cert-verify id** *id_num2*

id_num1 identifies the CA certificate that issued the certificate being verified, and *id_num2* identifies the locally stored certificate the device uses to verify the signature on the OCSP response.

If the certificate of the responder is not embedded in the OCSP response or stored locally, then the security device verifies the signature by using the CA certificate that issued the certificate in question.

You can use CLI commands to configure a security device for OCSP. Most of these commands use an identification number to associate the revocation reference URL with the CA certificate. You can obtain this ID number using the following CLI command:

get pki x509 list ca-cert

NOTE: The security device dynamically assigns the ID number to the CA certificate when you list the CA certificates. This number might change after you modify the certificate store.

Specifying a Certificate Revocation Check Method

To specify the revocation check method (CRL, OCSP, or none) for a certificate of a particular CA, use the following CLI syntax:

set pki authority *id_num* **cert-status revocation-check { CRL | OCSP | none }**

where **id_num** is the identification number for the certificate.

The following example specifies OCSP revocation checking:

```
set pki authority 3 cert-status revocation-check ocsp
```

The ID number 3 identifies the certificate of the CA.

Viewing Status Check Attributes

To display the status check attributes for a particular CA, use the following CLI syntax:

get pki authority *id_num* **cert-status**

where *id_num* is the identification number for the certificate issued by the CA.

To display the status check attributes for the CA that issued certificate 7:

```
get pki authority 7 cert-status
```

Specifying an Online Certificate Status Protocol Responder URL

To specify the URL string of an OCSP responder for a particular certificate, use the following CLI syntax:

set pki authority *id_num* **cert-status ocsp url** *url_str*

To specify the URL string of an OCSP responder (http:\\192.168.10.10) for the CA with certificate at index 5, use the following CLI syntax:

```
set pki authority 5 cert-status ocsp url http:\\192.168.10.10
```

To remove the URL (http:\\192.168.2.1) of a CRL server for a certificate 5:

```
unset pki authority 5 cert-status ocsp url http:\\192.168.2.1
```

Removing Status Check Attributes

To remove all certificate status check attributes for a CA that issued a particular certificate, use the following syntax:

unset pki authority *id_num* **cert-status**

To remove all revocation attributes related to certificate 1:

```
unset pki authority 1 cert-status
```

Self-Signed Certificates

A self-signed certificate is a certificate that is signed by and issued to the same entity; that is, the issuer and the subject of the certificate are the same. For example, the CA certificates of all root certificate authorities (CAs) are self-signed.

A security device automatically generates a self-signed certificate when powering up—if there is no certificate already configured for Secure Sockets Layer (SSL), which is the case when you first power it up. The security device that creates an auto-generated self-signed certificate is the only device that uses it. The device never exports this certificate outside itself. Even if the security device is in a NetScreen Redundancy Protocol (NSRP) cluster, it does not include the auto-generated self-signed certificate with other types of certificates when synchronizing PKI objects among other members in the cluster. (NSRP members do exchange manually generated self-signed certificates. For information about manually generating self-signed certificates, see “Manually Creating Self-Signed Certificates” on page 761.)

Although you cannot export an auto-generated self-signed certificate, you can copy its subject name and fingerprint. You can then deliver this to a remote admin who can later use the subject name and fingerprint to verify the self-signed certificate received during SSL negotiations. Checking the subject name and fingerprint is an important precaution against man-in-the-middle attacks in which someone intercepts an SSL connection attempt and pretends to be the targeted security device by responding with his own self-signed certificate. (For more information about verifying a self-signed certificate, see “Certificate Validation” on page 760.)

You can use a self-signed certificate when making a Secure Sockets Layer (SSL) connection to the security device. When you manage the device through the WebUI, SSL can provide authentication and encryption to secure your administrative traffic. You can even configure a security device to redirect an administrative connection attempt using HTTP (default port 80) to SSL (default port 443).

NOTE: For more information about SSL, including the HTTP-to-SSL redirect mechanism, see“Secure Sockets Layer” on page 315.

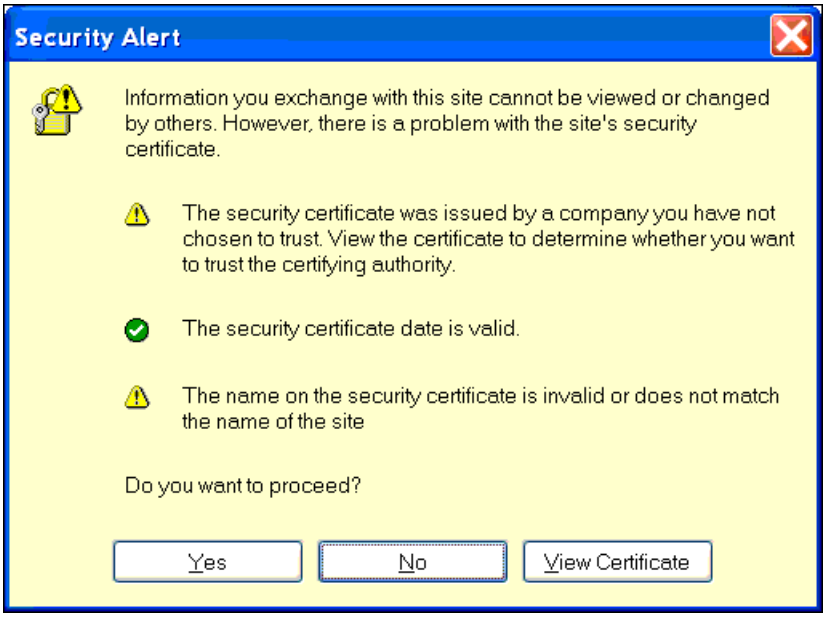
By default, the security device makes the auto-generated self-signed certificate available for use with SSL negotiations. It is the default SSL certificate. If you later install a CA-signed certificate or you configure the security device to generate another self-signed certificate, you can use one of these other certificates for SSL. If you delete the auto-generated self-signed certificate and do not assign another certificate for SSL use, the security device automatically generates another self-signed certificate the next time the device reboots.

NOTE: To learn how to create another self-signed certificate, see “Manually Creating Self-Signed Certificates” on page 761. To learn how to delete an auto-generated self-signed certificate, see “Deleting Self-Signed Certificates” on page 767.

Certificate Validation

During an SSL handshake, the security device authenticates itself by sending a certificate to the SSL client. When the security device sends a self-signed certificate, the SSL client cannot validate it by checking the issuing CA's signature because no CA issued it. When the security device presents a self-signed certificate for use in establishing an SSL session, the browser on the admin's computer tries to validate it with a CA certificate in its CA store. When it fails to find such an authority, the browser displays a message such as that shown in Figure 209 on page 760, prompting the admin to accept or refuse the certificate.

Figure 209: Security Alerts for Self-Signed Certificates



If this is the first time connecting to the security device after its initial bootup, the system clock might be inaccurate. Consequently, the validity period on the certificate might also be inaccurate. Even if you set the system clock and then regenerate a self-signed certificate, the browser can never find an authenticating CA, so the administrator must be prepared to see one of the above messages each time the security device uses a self-signed certificate for an SSL connection.

Without recourse to the certificate validation of an impartial third-party CA, the administrator logging in through SSL might wonder if the received self-signed certificate is indeed from the security device to which he is attempting to connect. (After all, the certificate might be from an interloper using a man-in-the-middle attack in an attempt to masquerade as the security device.) The admin can validate the certificate by using the subject name and fingerprint of the self-signed certificate. You can deliver the subject name and fingerprint to the admin so that the admin can validate the self-signed certificate when the security device later provides it to authenticate itself.

To see the subject name and fingerprint of an auto-generated self-signed certificate, use the following command:

```
device-> get pki x509 cert system
. . .
   CN=0043022002000186,CN=system generated,CN=self-signed,
. . .
finger print (md5) <e801eae4 56699fbc 324e38f2 4cfa5d47>
finger print (sha) <0113f5ec 6bd6d32b 4ef6ead9 f809eead 3a71435b>
```

NOTE: You cannot view the details of an auto-generated self-signed certificate through the WebUI.

After viewing the subject name and fingerprint, you can copy and deliver them (using a secure out-of-band method of your choice) to the admin that is later going to connect to the security device through SSL. When the admin's SSL client receives the certificate from the security device during the SSL handshake, the admin can compare the subject name and fingerprint in the received certificate with those that received earlier out-of-band. If they match, the admin knows that the certificate is authentic. Because there is no trusted third-party CA authority to authenticate the certificate, wIthout the subject name and fingerprint to compare, the remote admin cannot know for sure if the certificate is genuine.

Manually Creating Self-Signed Certificates

The security device automatically generates a self-signed certificate when you first power up the device so that it can support SSL for the initial connection. However, you might want to generate a different self-signed certificate from the one that the security device automatically generates. The following are some possible reasons for replacing the auto-generated self-signed certificate with an admin-defined self-signed certificate:

- The auto-generated self-signed certificate uses a fixed key size of 1024 bits. Your needs might require a larger or smaller key size, which you can control when generating your own self-signed key.
- You might want to use a certificate with a different subject name from the one that is automatically created.
- You might have a need for multiple self-signed certificates. On security devices that support virtual systems, the root system can share the auto-generated self-signed certificate with all the virtual systems. However, vsys administrators might prefer to generate their own self-signed certificates and then require their administrators to check the subject name and fingerprint of these specific certificates instead of the attributes of a shared certificate.

NOTE: Unlike an auto-generated self-signed certificate, which never passes outside the device in which it is created, a manually generated self-signed certificate is included with other certificates when a security device in an NSRP cluster synchronizes PKI objects with other members in the cluster.

Although you can configure various components of a self-signed certificate—such as the distinguished name (DN) fields, the subject alternative name, and the key size—the following common name (CN) elements always appear at the end of the DN:

"CN = *dev_serial_num,* CN = NetScreen self-signed"

Although the primary intended use of a self-signed certificate is to provide immediate out-of-the-box support for making a Secure Sockets Layer (SSL) connection to a security device, you can potentially use this certificate as you would any other CA-signed certificate. The uses for a self-signed certificate can include the following:

- Making a Secure Sockets Layer (SSL) connection to protect administrative traffic to a security device
- Securing traffic between Network and Security Manager (NSM) and a security device
- Authenticating IKE peers when establishing VPN tunnels

NOTE: For the current ScreenOS release, we support self-signed certificates only for use with SSL.

Setting an Admin-Defined Self-Signed Certificate

In this example, you define the following components of a self-signed certificate for use with SSL:

- Distinguished Name/Subject Name:
  - Name: 4ssl
  - Organization: jnpr
  - FQDN: www.juniper.net
- Key type and length: RSA, 1024 bits

After defining it, you generate the certificate and view it. You can then copy the subject name and fingerprint (also referred to as a "thumbprint") for distribution to other admins logging in through SSL to manage the security device.

When an admin attempts to log in using SSL, the security device sends him this certificate. The admin can open the certificate and compare the subject name and fingerprint in the certificate with the information received previously. If they match, the admin knows that the certificate is authentic.

WebUI

1. **Define the Certificate Attributes**

   Objects > Certificates > New: Enter the following, then click **Generate**:

Certificate Subject Information:
Name: 4ssl
Organization: jnpr
FQDN: www.juniper.net
Key Pair Information:
RSA: (select)
Create new key pair of **1024** length.

2. **Generate the Self-Signed Certificate**

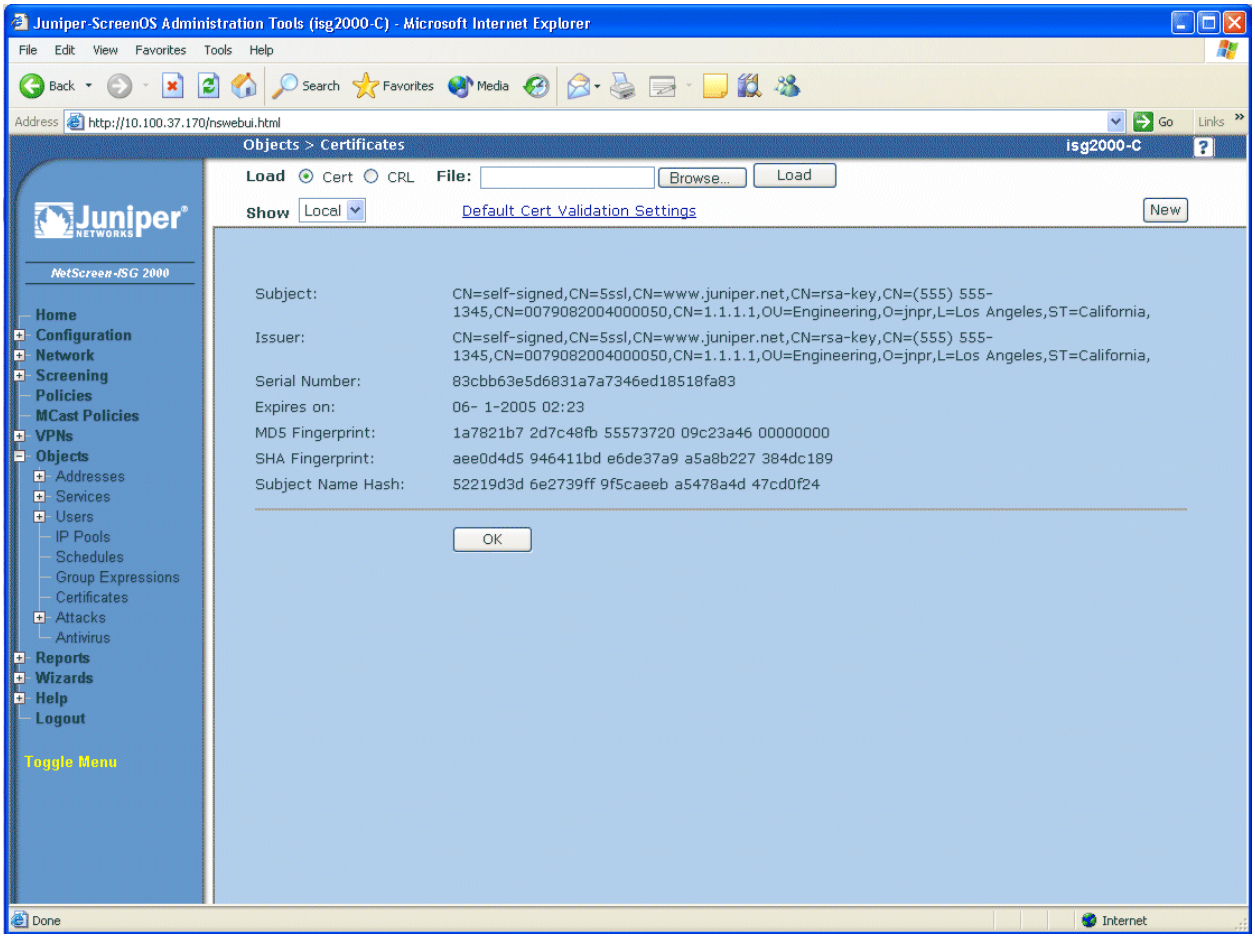
   After the security device completes the key generation it composes a certificate request. Click **Generate Self-Signed Cert**.

3. **View the Self-Signed Certificate**

   Objects > Certificates > Show Local: Click **Detail** for the certificate that you just created.

   The Certificate Details page appears, as shown in Figure 210 on page 763.

Figure 210: Certificate Details



You can copy the subject name and fingerprint from this page and communicate it to other administrators who intend to use SSL when managing the security device. When they initiate an SSL connection, they can then use this information to ensure that the certificate they receive is indeed from the security device.

CLI

1. **Define the Certificate Attributes**

```
set pki x509 dn name 4ssl
set pki x509 dn org-name jnpr
set pki x509 cert-fqdn www.juniper.net
save
```

2. **Generate the Self-Signed Certificate**

To generate a public/private key pair, which the Juniper Networks security device uses in its certificate request, enter the following command:

```
exec pki rsa new-key 1024
```

After the security device generates a key pair, it composes the following certificate request:

```
—BEGIN CERTIFICATE REQUEST—
MIIB0jCCATsCAQAwZTENMAsGA1UEChMESk5QUjEZMBcGA1UEAxMQMDA0MzAyMjAw

MjAwMDE4NjEQMA4GA1UEAxMHcnNhLWtleTEYMBYGA1UEAxMPd3d3Lmp1bmlwZXIu

bmV0MQ0wCwYDVQQDEwQ1c3NsMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDP

aAteIkL4HxQmO1w1jv9NMmrWnzdVYnGrKrXnw2MaB3xEgouWrlymEkZetA2ouKeA

D24SL0h1YvJ7Sd9PvkhwHOnvP1zkOCWA84TgvxBzcAyeBnS1UpSwcC0admX0Da6T

80EUuGrmUWodddRFUc8o5d2VGTUOM7WgcFDZRSGQGwIDAQABoC0wKwYJKoZIhvcN

AQkOMR4wHDAaBgNVHREEEzARgg93d3cuanVuaXBlci5uZXQwDQYJKoZIhvcNAQEF

BQADgYEAgvDXI4H905y/2+k4omo9Y4XQrgq44Rj3jqXAYYMgQBd0Q8HoyL5NE3+i

QUkiYjMTWO2wIWzEr4u/tdAISEVTu03achZa3zIkUtn8sD/VYKhFlyPCBVvMiaHd

FzIHUgBuMrr+awowJDG6wARhR75w7pORXy7+aAmvIjew8YRre9s=
—END CERTIFICATE REQUEST—
```

To learn the ID number for the key pair, use the following command:

```
get pki x509 list key-pair
Getting OTHER PKI OBJECT ...
IDX  ID num     X509 Certificate Subject Distinguish Name
=======================================================
0000 176095259 CN=4ssl,CN=www.juniper.net,CN=rsa-key,
CN=0043022002000186,O=jnpr,
=======================================================
```

To generate the self-signed certificate, enter the following command, referencing the key-pair ID number that you learned from the output of the previous command:

```
exec pki x509 self-signed-cert key-pair 176095259
```

3. **View the Self-Signed Certificate**

To view the self-signed certificate that you have just created, enter the following command:

```
get pki x509 list local-cert

  Getting LOCAL CERT ...
  IDX  ID num     X509 Certificate Subject Distinguish Name
 ========================================================
  0000 176095261 LOCAL CERT friendly name <29>
  LOCAL CERT friendly name <29>
 CN=self-signed,CN=4ssl,CN=www.juniper.net,CN=rsa-key,CN=0043022002000186,
    O=jnpr,
  Expire on 10-19-2009 17:20, Issued By:
 CN=self-signed,CN=4ssl,CN=www.juniper.net,CN=rsa-key,CN=0043022002000186,
    O=jnpr,
 ========================================================
```

To view the certificate in more detail, enter the following command using the ID number of the certificate:

```
get pki x509 cert 176095261

  -0001 176095261 LOCAL CERT friendly name <29>
 CN=self-signed,CN=4ssl,CN=www.juniper.net,CN=rsa-key,CN=0043022002000186,
    O=jnpr,
  Expire on 10-19-2009 17:20, Issued By:
 CN=self-signed,CN=4ssl,CN=www.juniper.net,CN=rsa-key,CN=0043022002000186,
    O=jnpr,
  Serial Number: <9d1c03365a5caa172ace4f82bb5ec9da>
  subject alt name extension:
  email(1): (empty)
  fqdn(2): (www.juniper.net)
  ipaddr(7): (empty)
  no renew
  finger print (md5) <be9e0280 02bdd9d1 175caf23 6345198e>
  finger print (sha) <87e0eee0 c06f9bac 9098bd02 0e631c1b 26e37e0e>
  subject name hash: <d82be8ae 4e71a576 2e3f06fc a98319a3 5c8c6c27>
  use count: <1>
  flag <00000000>
```

You can copy the **subject name** and **fingerprint** from this output and communicate it to other administrators who intend to use SSL when managing the security device. When they initiate an SSL connection, they can then use this information to ensure that the certificate they receive is indeed from the security device.

Certificate Auto-Generation

The first time the security device powers up, it automatically generates a self-signed certificate. The primary purpose of this certificate is to support SSL immediately after the initial bootup of a security device. To see this certificate, use the following CLI command:

```
get pki x509 cert system
    CN=0010062001000021,CN=system generated,CN=self-signed,
    Expire on 08- 3-2014 16:19, Issued By:
    CN=0010062001000021,CN=system generated,CN=self-signed,
    Serial Number: <c927f2044ee0cf8dc931cdb1fc363119>
    finger print (md5) <fd591375 83798574 88b3e698 62890b5d>
    finger print (sha) <40a1bda8 dcd628fe e9deaee1 92a2783c 817e26d9>
    subject name hash: <0324d38d 52f814fe 647aba3a 86eda7d4 a7834581>
```

By default, the security device automatically generates a self-signed certificate during the bootup process if the following conditions are met:

- No automatically generated self-signed certificate exists on the device.
- No certificate has been assigned for use with SSL.

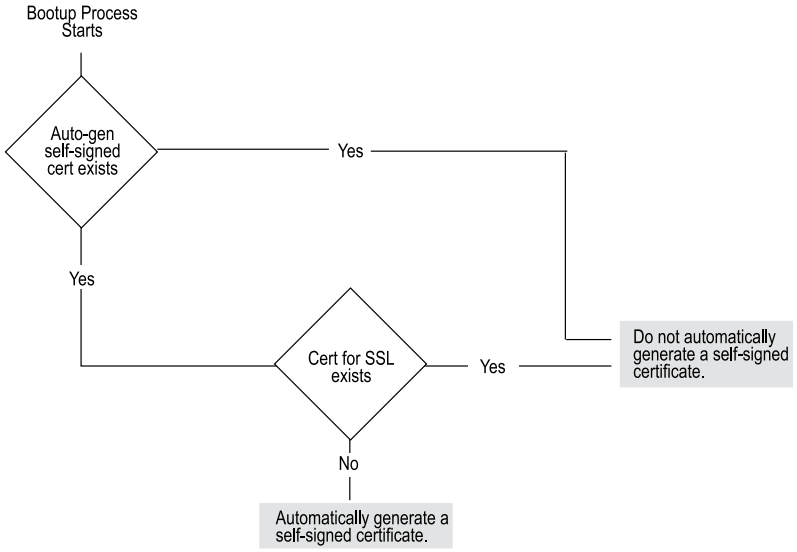
You can use the following command to see if a certificate is already configured for SSL:

```
get ssl
web SSL enable.
web SSL port number(443).
web SSL cert: Default - System Self-Signed Cert.
web SSL cipher(RC4_MD5).
```

In the above output, you can see that SSL is using the automatically generated (“System”) self-signed certificate.

Figure 211 on page 767 shows the decision path for certificate generation that the security device takes when booting up.

Figure 211: Decision Path for Certificate Auto-Generation



If you delete the automatically generated self-signed certificate, assign another certificate for use with SSL, and then reset the device, the security device does not regenerate another self-signed certificate during the bootup process. If you next change the SSL configuration so that no certificate is assigned to it and then reset the device, the security device does automatically regenerate a new self-signed certificate during the next bootup process.

Deleting Self-Signed Certificates

You can delete a self-signed certificate that is automatically or manually generated as you can with any type of certificate. Perhaps you obtain a CA-signed certificate that you prefer to use for SSL instead of a self-signed certificate. For whatever reason, to delete the auto-generated self-signed certificate, use the following CLI command:

```
delete pki object-id system
```

To delete an admin-configured self-signed certificate, use the following command, where *id_num* is the ID number of the certificate that you want to delete:

```
delete pki object-id id_num
```

If you delete the auto-generated self-signed certificate and then later want the security device to generate another one, do the following:

- Assign no other certificate for SSL (You can use the following command: **unset ssl cert**).
- Reset the security device.

The security device can redirect HTTP traffic (default port 80) sent to the device itself to SSL (default port 443). Therefore, to ensure that a certificate is available for SSL, during the bootup process, the security device always checks if an auto-generated self-signed certificate exists or if another certificate has been assigned for SSL to use.

If there is no auto-generated self-signed certificate and no other certificate is assigned for SSL use, the security device automatically generates a self-signed certificate.

NOTE: You can only delete an automatically generated self-signed certificate through the CLI.

To learn the ID number for a certificate, use the following command: **get pki x509 list local-cert**.

For information about the redirection of HTTP traffic to SSL, see "Redirecting HTTP to SSL" on page 318.

Chapter 21

Virtual Private Network Guidelines

ScreenOS offers a variety of cryptographic options for configuring a virtual private network (VPN) tunnel. Even when you are configuring a simple tunnel, you must make choices. The goals of the first half of this chapter are to first summarize all the choices for a basic site-to-site VPN and a basic dialup VPN and to then present one or more reasons for choosing one option or another.

The second half of the chapter explores the difference between policy-based and route-based VPN tunnels. It also examines the packet flow for a route-based and policy-based site-to-site AutoKey IKE VPN tunnel to see the outbound and inbound processing stages that a packet undergoes. The chapter concludes with some VPN configuration tips to keep in mind when configuring a tunnel.

This chapter contains the following sections:

Cryptographic Options

When configuring a virtual private network (VPN), you must make many decisions about the cryptography you want to use. Questions arise about which Diffie-Hellman (DH) group is the right one to choose, which encryption algorithm provides the best balance between security and performance, and so on. This section presents all the cryptographic options required to configure a basic site-to-site VPN tunnel and a basic dialup VPN tunnel and explains one or more benefits about each one to help you make your decisions.

The first decision that you must make is whether the tunnel is for a site-to-site VPN tunnel (between two security devices) or whether it is for a dialup VPN (from the NetScreen-Remote VPN client to a security device). Although this is a networking decision, the distinction between the two types of tunnels affects some cryptographic options. Therefore, the options are presented in two different figures:

- “Site-to-Site Cryptographic Options” on page 770 explains Figure 212 on page 770.
- “Dialup VPN Options” on page 777 explains Figure 213 on page 778.

After you decide whether you are going to configure a site-to-site tunnel or a dialup tunnel, you can refer to either Figure 212 on page 770 or Figure 213 on page 778 for guidance. Each figure presents the cryptographic choices that you must make while configuring the tunnel. Following each figure are reasons for choosing each option that appears in the figure.

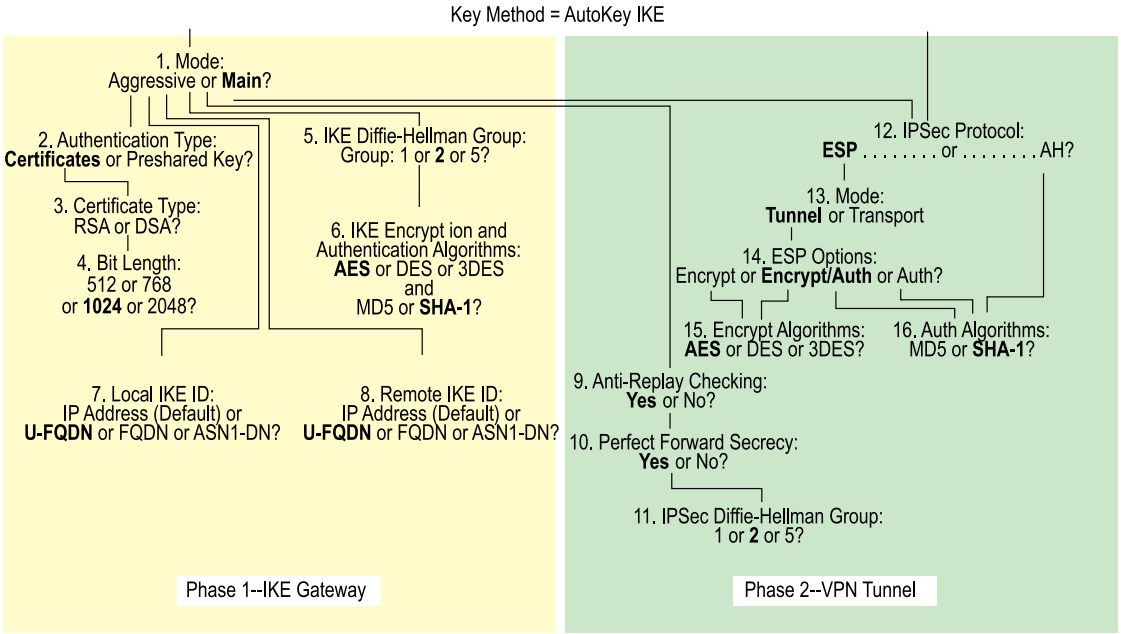
NOTE: Examples for configuring both kinds of tunnels are in Chapter 4, "Site-to-Site Virtual Private Networks" on page 801 and Chapter 5, "Dialup Virtual Private Networks" on page 887

Site-to-Site Cryptographic Options

When configuring a basic site-to-site VPN tunnel, you must choose among the cryptographic options shown in Figure 212 on page 770. Advantages for each option follow the figure.

NOTE: Figure 212 on page 770 shows recommended options in **boldface**. For background information about the different IPsec options, see "Internet Protocol Security" on page 707.

Figure 212: Cryptographic Options for a Site-to-Site VPN Tunnel



1. **Key Method: Manual Key or AutoKey IKE?**

   AutoKey IKE

   - Recommended

- Provides automatic key renewal and key freshness, thereby increasing security

Manual Key

- Useful for debugging IKE problems
- Eliminates IKE negotiation delays when establishing a tunnel

2. **Mode: Aggressive or Main?**

Aggressive

Required when the IP address of one of the IPsec peers is dynamically assigned and a preshared key is used

Main

- Recommended
- Provides identity protection
- Can be used when the dialup user has a static IP address or if certificates are used for authentication

3. **Authentication Type: Preshared Key or Certificates?**

Certificates

- Recommended
- Greater security than provided by preshared keys because you can validate certificates with a certificate authority (CA) (For more information, see "Public Key Cryptography" on page 741.)

Preshared Key

Easier to use and faster to set up because it does not require a Public Key Infrastructure (PKI)

4. **Certificate Type: RSA or DSA?**

This depends on the CA from whom you get your certificates. There is no advantage of one certificate type over the other.

5. **Bit Length: 512 or 768 or 1024 or 2048?**

512

Incurs the least processing overhead

768

- Provides more security than 512 bits
- Incurs less processing overhead than 1024 and 2048 bits

1024

- Recommended
- Provides more security than 512 and 768 bits
- Incurs less processing overhead than 2048 bits

2048

Provides the most security

6. **IKE Diffie-Hellman Group: 1 or 2 or 5 or 14?**

Diffie-Hellman Group 1

- Incurs less processing overhead than DH groups 2, 5 and 14
- Processing acceleration provided in Juniper Networks security hardware

Diffie-Hellman Group 2

- Recommended
- Incurs less processing overhead than DH groups 5 and 14
- Provides more security than DH group 1
- Processing acceleration provided in Juniper Networks security hardware

Diffie-Hellman Group 5

- Provides more security that DH groups 1 and 2
- Incurs less processing overhead than DH group 14

Diffie-Hellman Group 14

- Provides the most security

7. **IKE Encrypt and Auth Algorithms: AES or DES or 3DES and MD5 or SHA-1 or SHA2-256?**

AES

- Recommended
- Cryptographically stronger than DES and 3DES if key lengths are all equal
- Processing acceleration provided in Juniper Networks security hardware
- Approved encryption algorithm for Federal Information Processing Standards (FIPS) and Common Criteria EAL4 standards

DES

- Incurs less processing overhead than 3DES and AES
- Useful when the remote peer does not support AES

3DES

- Provides more cryptographic security than DES
- Processing acceleration provided in Juniper Networks security hardware

MD5

- Incurs less processing overhead than SHA-1 and SHA2-256

SHA-1

- Recommended
- Provides more cryptographic security than MD5
- Incurs less processing overhead than SHA2-256
- The only authentication algorithm that FIPS accepts

SHA2-256

- Provides more cryptographic security than SHA-1

8. **Local IKE ID: IP Address (Default) or U-FQDN or FQDN or ASN1-DN?**

IP Address

- Recommended
- Can only be used if the local security device has a static IP address
- Default IKE ID when using a preshared key for authentication
- Can be used with a certificate if the IP address appears in the SubjectAltName field

U-FQDN

User-Fully Qualified Domain Name (U-FQDN—an email address): Can be used with a preshared key or a certificate if the U-FQDN appears in the SubjectAltName field

FQDN

- Fully Qualified Domain Name (FQDN): Can be used with a preshared key or a certificate if the FQDN appears in the SubjectAltName field
- Useful for VPN gateways that have dynamic IP addresses
- Default IKE ID when using RSA or DSA certificates for authentication

ASN1-DN

- Can be used only with certificates
- Useful if the CA does not support the SubjectAltName field in the certificates it issues

9. **Remote IKE ID: IP Address (Default) or U-FQDN or FQDN or ASN1-DN?**

   IP Address

   - Recommended
   - Does not require you to enter a remote IKE ID for a peer at a static IP address when using preshared keys for authentication and the peer is a security device
   - Can be used for a device with a static IP address
   - Can be used with a preshared key or a certificate if the IP address appears in the SubjectAltName field

   U-FQDN

   User-Fully Qualified Domain Name (U-FQDN—an email address): Can be used with a preshared key or a certificate if the U-FQDN appears in the SubjectAltName field

   FQDN

   - Fully Qualified Domain Name (FQDN): Can be used with a preshared key or a certificate if the FQDN appears in the SubjectAltName field
   - Useful for VPN gateways that have dynamic IP addresses
   - Does not require you to enter a remote IKE ID when using certificates for authentication and the peer is a security device

   ASN1-DN

   - Can be used only with certificates
   - Useful if the CA does not support the SubjectAltName field in the certificates it issues

10. **Anti-Replay Checking: No or Yes?**

    Yes

    - Recommended
    - Enables the recipient to check sequence numbers in packet headers to prevent denial of service (DoS) attacks caused when a malefactor resends intercepted IPsec packets

    No

    Disabling this might resolve compatibility issues with third-party peers

11. **Perfect Forward Secrecy: No or Yes?**

    Yes

    - Recommended

- Perfect Forward Secrecy (PFS): Provides increased security because the peers perform a second DH exchange to produce the key used for IPsec encryption/decryption

No

- Provides faster tunnel setup
- Incurs less processing during Phase 2 IPsec negotiations

12. **IPsec Diffie-Hellman Group: 1 or 2 or 5 or 14?**

Diffie-Hellman Group 1

- Incurs less processing overhead than DH groups 2, 5 and 14
- Processing acceleration provided in Juniper Networks security hardware

Diffie-Hellman Group 2

- Recommended
- Incurs less processing overhead than DH groups 5 and 14
- Provides more security than DH group 1
- Processing acceleration provided in Juniper Networks security hardware

Diffie-Hellman Group 5

- Provides more security that DH groups 1 and 2
- Incurs less processing overhead than DH group 14

Diffie-Hellman Group 14

- Provides the most security

13. **IPsec Protocol: ESP or AH?**

ESP

- Recommended
- Encapsulating Security Payload (ESP): Can provide both confidentiality through encryption and encapsulation of the original IP packet and integrity through authentication
- Can provide either encryption alone or authentication alone

AH

Authentication Header (AH): Provides authentication of the entire IP packet, including the IPsec header and outer IP header

14. **Mode: Tunnel or Transport?**

Tunnel

- Recommended
- Conceals the original IP header, thereby increasing privacy

Transport

Required for L2TP-over-IPsec tunnel support

15. **ESP Options: Encrypt or Encrypt/Auth or Auth?**

Encrypt

- Provides faster performance and incurs less processing overhead than using encrypt/auth
- Useful when you require confidentiality but do not require authentication

Encrypt/Auth

- Recommended
- Useful when you want confidentiality and authentication

Auth

Useful when you want authentication but do not require confidentiality. Perhaps when the information is not secret, but it is important to establish that the information truly comes from the person who claims to send it and that nobody tampered with the content while in transit.

16. **Encrypt Algorithms: AES or DES or 3DES?**

AES

- Recommended
- Cryptographically stronger than DES and 3DES if key lengths are all equal
- Processing acceleration provided in Juniper Networks security hardware
- Approved encryption algorithm for FIPS and Common Criteria EAL4 standards

DES

- Incurs less processing overhead than 3DES and AES
- Useful when the remote peer does not support AES

3DES

- Provides more cryptographic security than DES
- Processing acceleration provided in Juniper Networks security hardware

17. **Auth Algorithms: MD5 or SHA-1 or SHA2-256?**

MD5

Incurs less processing overhead than SHA-1

SHA-1

- Recommended
- Provides more cryptographic security than MD5

SHA2-256

- Provides more cryptographic security than SHA-1

Using the recommended options from the previous list, a generic site-to-site VPN configuration between two security devices with static IP addresses would consist of the following components:

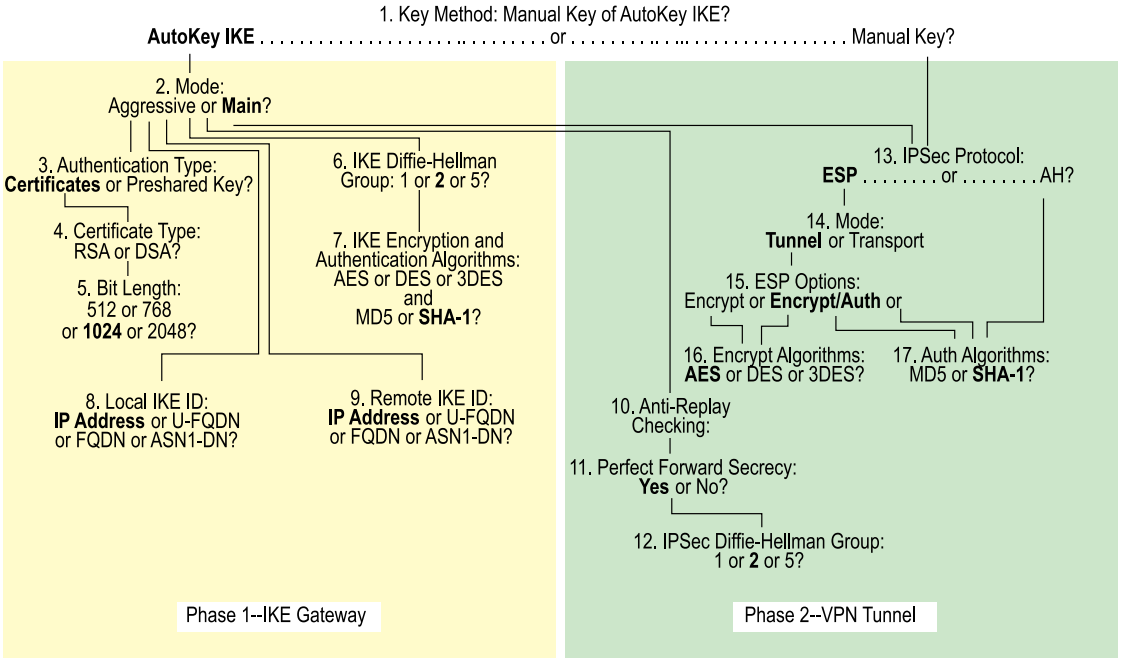
■ AutoKey IKE	■ Perfect Forward Secrecy (PFS) = yes
■ Main mode	■ Phase 2 DH group 2
■ 1024-bit certificates (RSA or DSA)	■ Encapsulating Security Payload (ESP)
■ Phase 1 DH group 2	■ Tunnel mode
■ Encryption = AES	■ Encryption/Authentication
■ Authentication = SHA-1	■ Encryption = AES
■ IKE ID = IP address (default)	■ Authentication = SHA-1
■ Anti-replay protection = yes	

Dialup VPN Options

When configuring a basic dialup VPN tunnel, you must choose among the cryptographic options shown in Figure 213 on page 778. Advantages for each option follow the figure.

NOTE: Figure 213 on page 778 shows recommended options in **boldface**. For background information about the different IPsec options, see “Internet Protocol Security” on page 707.

Figure 213: Cryptographic Options for a Dialup VPN Tunnel



1. **Mode: Aggressive or Main?**

   Aggressive

   - Recommended
   - Required when the IP address of one of the IPsec peers is dynamically assigned and a preshared key is used
   - Can be used with either certificates or preshared keys for authentication

   Main

   Provides identity protection

2. **Authentication Type: Preshared Key or Certificates?**

   Certificates

   - Recommended
   - Greater security than provided by preshared keys because you can validate certificates with a certificate authority (CA) (For more information, see “Public Key Cryptography” on page 741.)

   Preshared Key

   Easier to use and faster to set up because it does not require a Public Key Infrastructure (PKI)

3. **Certificate Type: RSA or DSA?**

This depends on the CA from whom you get your certificates. There is no advantage of one certificate type over the other.

4. **Bit Length: 512 or 768 or 1024 or 2048?**

   512

   Incurs the least processing overhead

   768

   - Provides more security than 512 bits
   - Incurs less processing overhead than 1024 and 2048 bits

   1024

   - Recommended
   - Provides more security than 512 and 768 bits
   - Incurs less processing overhead than 2048 bits

   2048

   Provides the most security

5. **IKE Diffie-Hellman Group: 1 or 2 or 5 or 14?**

   Diffie-Hellman Group 1

   - Incurs less processing overhead than DH groups 2, 5 and 14
   - Processing acceleration provided in Juniper Networks security hardware

   Diffie-Hellman Group 2

   - Recommended
   - Incurs less processing overhead than DH groups 5 and 14
   - Provides more security than DH group 1
   - Processing acceleration provided in Juniper Networks security hardware

   Diffie-Hellman Group 5

   - Incurs less processing overhead than DH group 14
   - Provides more security than DH groups 1 and 2

   Diffie-Hellman Group 14

   - Provides the most security

6. **IKE Encrypt and Auth Algorithms: AES or DES or 3DES and MD5 or SHA-1 or SHA2-256?**

AES

- Recommended
- Cryptographically stronger than DES and 3DES if key lengths are all equal
- Processing acceleration provided in Juniper Networks security hardware
- Approved encryption algorithm for FIPS and Common Criteria EAL4 standards

DES

- Incurs less processing overhead than 3DES and AES
- Useful when the remote peer does not support AES

3DES

- Provides more cryptographic security than DES
- Processing acceleration provided in Juniper Networks security hardware

MD5

- Incurs less processing overhead than SHA-1 and SHA2-256

SHA-1

- Recommended
- Provides more cryptographic security than MD5
- Incurs less processing overhead than SHA2-256

SHA2-256

- Provides more cryptographic security than SHA-1

7. **Local IKE ID: IP Address (Default) or U-FQDN or FQDN or ASN1-DN?**

IP Address (Default)

- Does not require you to enter an IKE ID for a device with a static IP address
- Can be used for a device with a static IP address
- Can be used with a preshared key or a certificate if the IP address appears in the SubjectAltName field

U-FQDN

- Recommended
- User-Fully Qualified Domain Name (U-FQDN—an email address): Can be used with a preshared key or a certificate if the U-FQDN appears in the SubjectAltName field

FQDN

- Fully Qualified Domain Name (FQDN): Can be used with a preshared key or a certificate if the FQDN appears in the SubjectAltName field
- Useful for VPN gateways that have dynamic IP addresses

ASN1-DN

- Can be used only with certificates
- Useful if the CA does not support the SubjectAltName field in the certificates it issues

8. **Anti-Replay Checking: No or Yes?**

Yes

- Recommended
- Enables the recipient to check sequence numbers in packet headers to prevent denial of service (DoS) attacks caused when a malefactor resends intercepted IPsec packets

No

Disabling this might resolve compatibility issues with third-party peers

9. **Perfect Forward Secrecy: No or Yes?**

Yes

- Recommended
- Perfect Forward Secrecy (PFS): Provides increased security because the peers perform a second DH exchange to produce the key used for IPsec encryption/decryption

No

- Provides faster tunnel setup
- Incurs less processing during Phase 2 IPsec negotiations

10. **IPsec Diffie-Hellman Group: 1 or 2 or 5 or 14?**

Diffie-Hellman Group 1

- Incurs less processing overhead than DH groups 2, 5 and 14
- Processing acceleration provided in Juniper Networks security hardware

Diffie-Hellman Group 2

- Recommended
- Incurs less processing overhead than DH groups 5 and 14

- Provides more security than DH group 1
- Processing acceleration provided in Juniper Networks security hardware

Diffie-Hellman Group 5

- Provides more security than DH groups 1 and 2
- Incurs less processing overhead than DH group 14

Diffie-Hellman Group 14

- Provides the most security

11. **IPsec Protocol: ESP or AH?**

ESP

- Recommended
- Encapsulating Security Payload (ESP): Can provide both confidentiality through encryption and encapsulation of the original IP packet and integrity through authentication
- Can provide either encryption alone or authentication alone

AH

Authentication Header (AH): Provides authentication of the entire IP packet, including the IPsec header and outer IP header.

12. **Mode: Tunnel or Transport?**

Tunnel

- Recommended
- Conceals the original IP header, thereby increasing privacy

Transport

Required for L2TP-over-IPsec tunnel support

13. **ESP Options: Encrypt or Encrypt/Auth or Auth?**

Encrypt

- Provides faster performance and incurs less processing overhead than using encrypt/auth
- Useful when you require confidentiality but do not require authentication

Encrypt/Auth

- Recommended
- Useful when you want confidentiality and authentication

Auth

Useful when you want authentication but do not require confidentiality. Perhaps when the information is not secret, but it is important to establish that the information truly comes from the person who claims to send it and that nobody tampered with the content while in transit.

14. **Encrypt Algorithms: AES or DES or 3DES?**

AES

- Recommended
- Cryptographically stronger than DES and 3DES if key lengths are all equal
- Processing acceleration provided in Juniper Networks security hardware
- Approved encryption algorithm for FIPS and Common Criteria EAL4 standards

DES

- Incurs less processing overhead than 3DES and AES
- Useful when the remote peer does not support AES

DES

- Provides more cryptographic security than DES
- Processing acceleration provided in Juniper Networks security hardware

15. **Auth Algorithms: MD5 or SHA-1 or SHA2-256?**

MD5

- Incurs less processing overhead than SHA-1 and SHA2-256

SHA-1

- Recommended
- Incurs less processing overhead than SHA2-256
- Provides more cryptographic security than MD5

SHA2-256

- Provides more cryptographic security than the MD5 and SHA-1

Using the recommended options from the above list, a generic dialup VPN configuration between two security devices with static IP addresses would consist of the following components:

■ Aggressive mode	■ Perfect Forward Secrecy (PFS) = yes
■ 1024-bit certificates (RSA or DSA)	■ Phase 2 DH group 2

■ Phase 1 DH group 2	■ Encapsulating Security Payload (ESP)
■ Encryption = AES	■ Tunnel mode
■ Authentication = SHA-1	■ Encryption/Authentication
■ IKE ID = U-FQDN (email address)	■ Encryption = AES
■ Anti-replay protection = yes	■ Authentication = SHA-1

Cryptographic Policy

A root admin user or a read-write admin user with a cryptographic role can configure a cryptographic policy on the security device. For information on role attributes, see “Role Attributes” on page 346.

To create a cryptographic policy:

```
set crypto-policy
```

You then configure all cryptographic attributes for the policy, such as encryption and authentication algorithms, authentication method, mode of operation, Diffie-Hellman (DH) Group, and Phase 1 and Phase 2 security associations (SA) lifetime values.

To set the attributes for a new cryptographic policy:

```
set crypto-policy
set encrypt-alg aes256
set auth-alg sha2-256
set dh group2
set auth-method rsa-sig
set mode aggressive
set p1-sa-lifetime upper-threshold days 1
set p2-sa-lifetime upper-threshold days 1
save
```

NOTE: You must use the CLI to configure a cryptographic policy.

To make all cryptographic-related configurations conform to the new cryptographic policy, you must restart the security device.

Route-Based and Policy-Based Tunnels

The configuration of a security device for VPN support is particularly flexible. You can create route-based and policy-based VPN tunnels. Additionally, each type of tunnel can use Manual Key or AutoKey IKE to manage the keys used for encryption and authentication.

With policy-based VPN tunnels, a tunnel is treated as an object (or a building block) that together with source, destination, service, and action, comprises a policy that

permits VPN traffic. (Actually, the VPN policy action is *tunnel,* but the action *permit* is implied, if unstated). In a policy-based VPN configuration, a policy specifically references a VPN tunnel by name.

With route-based VPNs, the policy does not specifically reference a VPN tunnel. Instead, the policy references a destination address. When the security device does a route lookup to find the interface through which it must send traffic to reach that address, it finds a route through a tunnel interface, which is bound to a specific VPN tunnel.

NOTE: Typically, a tunnel interface is bound to a single tunnel. You can also bind a tunnel interface to multiple tunnels. For more information, see "Multiple Tunnels per Tunnel Interface" on page 983.

Thus, with a policy-based VPN tunnel, you can consider a tunnel as an element in the construction of a policy. With a route-based VPN tunnel, you can consider a tunnel as a means for delivering traffic, and the policy as a method for either permitting or denying the delivery of that traffic.

The number of policy-based VPN tunnels that you can create is limited by the number of policies that the device supports. The number of route-based VPN tunnels that you create is limited by the number of route entries or the number of tunnel interfaces that the device supports—whichever number is lower.

A route-based VPN tunnel configuration is a good choice when you want to conserve tunnel resources while setting granular restrictions on VPN traffic. Although you can create numerous policies referencing the same VPN tunnel, each policy creates an individual IPsec security association (SA) with the remote peer, each of which counts as an individual VPN tunnel. With a route-based approach to VPNs, the regulation of traffic is not coupled to the means of its delivery. You can configure dozens of policies to regulate traffic flowing through a single VPN tunnel between two sites, and there is just one IPsec SA at work. Also, a route-based VPN configuration allows you to create policies referencing a destination reached through a VPN tunnel in which the action is *deny,* unlike a policy-based VPN configuration, in which—as stated earlier—the action must be *tunnel,* implying *permit.*

Another advantage that route-based VPNs offer is the exchange of dynamic routing information through VPN tunnels. You can enable an instance of a dynamic routing protocol, such as Border Gateway Protocol (BGP), on a tunnel interface that is bound to a VPN tunnel. The local routing instance exchanges routing information through the tunnel with a neighbor enabled on a tunnel interface bound to the other end.

When a tunnel does not connect large networks running dynamic routing protocols and you do not need to conserve tunnels or define various policies to filter traffic through the tunnel, a policy-based tunnel makes sense. Also, because there is no network beyond a dialup VPN client, policy-based VPN tunnels can be a good choice for dialup VPN configurations.

That said, when the dialup client supports a virtual internal IP address—which the NetScreen-Remote does—there are also compelling reasons for using a route-based VPN configuration. A route-based dialup VPN tunnel offers the following benefits:

- You can bind its tunnel interface to any zone to require or not require policy enforcement.
- You can define routes to force traffic through the tunnel, unlike a policy-based VPN configuration.
- A route-based VPN tunnel simplifies the addition of a spoke to a hub-and-spoke configuration (see "Creating Hub-and-Spoke VPNs" on page 1047).
- You can adjust the proxy ID to accept any IP address from the dialup VPN client by configuring the remote client's address as 255.255.255.255/32.
- You can define one or more Mapped IP (MIP) addresses on the tunnel interface.

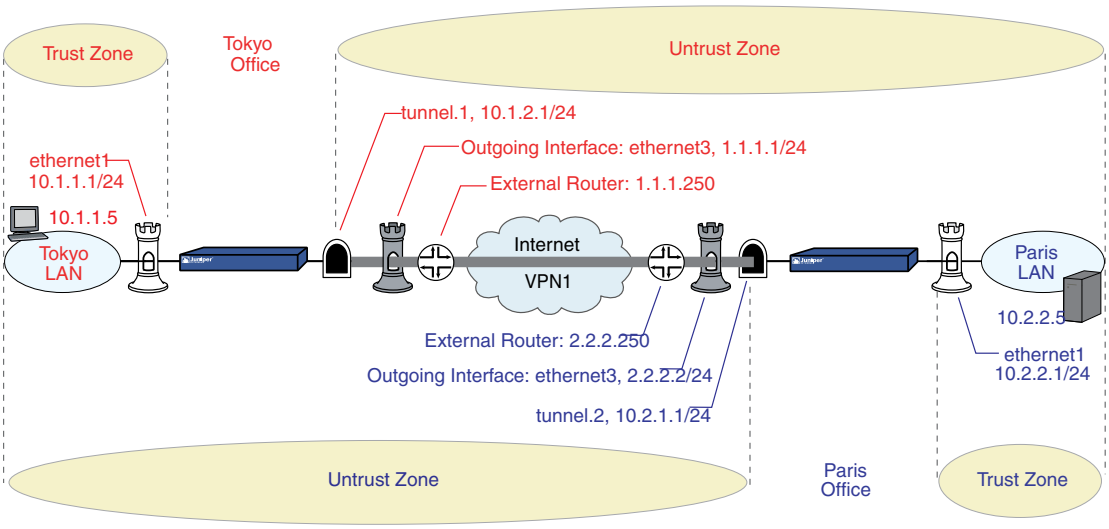
NOTE: For an example of a route-based VPN configuration for a dialup client, see "Route-Based Dialup VPN, Dynamic Peer" on page 894.

Packet Flow: Site-to-Site VPN

To better understand how the various components comprising the creation of an IPsec tunnel work in relation to each other, this section looks at the processing of a packet flow through a tunnel—both when a security device sends outbound VPN traffic and when it receives inbound VPN traffic. The processing for a route-based VPN is presented, followed by an addendum noting the two places in the flow that differ for a policy-based VPN.

A company based in Tokyo has just opened a branch office in Paris and needs to connect the two sites through an IPsec tunnel. The tunnel uses AutoKey IKE, the ESP protocol, AES for encryption, SHA-1 for authentication using a preshared key, and has anti-replay checking enabled. The security devices protecting each site are in NAT mode, and all the zones are in the trust-vr routing domain. The addresses are as shown in Figure 214 on page 787.

Figure 214: Site-to-Site VPN Tunnel



The path of a packet coming from 10.1.1.5/32 in the Tokyo LAN and going to 10.2.2.5/32 in the Paris LAN through an IPsec tunnel proceeds as described in the following subsections.

Tokyo (Initiator)

1. The host at 10.1.1.5 sends a packet destined for 10.2.2.5 to 10.1.1.1, which is the IP address ethernet1 and is the default gateway configured in the TCP/IP settings of host.
2. The packet arrives at ethernet1, which is bound to the Trust zone.
3. If you have enabled SCREEN options such as IP spoof detection for the Trust zone, the security device activates the SCREEN module at this point. SCREEN checking can produce one of the following three results:
   - If a SCREEN mechanism detects anomalous behavior for which it is configured to block the packet, the security device drops the packet and makes an entry in the event log.
   - If a SCREEN mechanism detects anomalous behavior for which it is configured to record the event but not block the packet, the security device records the event in the SCREEN counters list for ethernet1 and proceeds to the next step.
   - If the SCREEN mechanisms detect no anomalous behavior, the security device proceeds to the next step.

     If you have not enabled any SCREEN options for the Trust zone, the security device immediately proceeds to the next step.
4. The session module performs a session lookup, attempting to match the packet with an existing session.

   If the packet does not match an existing session, the security device performs First Packet Processing, a procedure involving the remaining steps.

If the packet matches an existing session, the security device performs Fast Processing, using the information available from the existing session entry to process the packet. Fast Processing bypasses the route and policy lookups because the information generated by the bypassed steps has already been obtained during the processing of the first packet in the session.

5. The address-mapping module checks if a Mapped IP (MIP) configuration uses the destination IP address 10.2.2.5. Because 10.2.2.5 is not used in a MIP configuration, the security device proceeds to the next step. (For information about packet processing when MIPs, VIPs, or destination address translation [NAT-dst] is involved, see "Packet Flow for NAT-Dst" on page 1501.)
6. To determine the destination zone, the route module does a route lookup for 10.2.2.5. (The route module uses the ingress interface to determine which virtual router to use for the route lookup.) It finds a route entry directing traffic to 10.2.2.5 through the tunnel.1 interface bound to a VPN tunnel named "vpn1". The tunnel interface is in the Untrust zone. By determining the ingress and egress interfaces, the security device has thereby determined the source and destination zones and can now do a policy lookup.
7. The policy engine does a policy lookup between the Trust and Untrust zones (as determined by the corresponding ingress and egress interfaces). The action specified in the policy matching the source address and zone, destination address and zone, and service is permit.
8. The IPsec module checks if an active Phase 2 security association (SA) exists with the remote peer. The Phase 2 SA check can produce either of the following results:
   - If the IPsec module discovers an active Phase 2 SA with that peer, it proceeds to step 10.
   - If the IPsec module does not discover an active Phase 2 SA with that peer, it drops the packet and triggers the IKE module.
9. The IKE module checks if an active Phase 1 SA exists with the remote peer. The Phase 1 SA check can produce either of the following results:
   - If the IKE module discovers an active Phase 1 SA with the peer, it uses this SA to negotiate a Phase 2 SA.
   - If the IKE module does not discover an active Phase 1 SA with that peer, it begins Phase 1 negotiations in main mode, and then Phase 2 negotiations.
10. The IPsec module puts an ESP header and then an outer IP header on the packet. Using the address specified as the outgoing interface, it puts 1.1.1.1 as the source IP address in the outer header. Using the address specified for remote gateway, it puts 2.2.2.2 as the destination IP address in the outer header. Next, it encrypts the packet from the payload to the next header field in the original IP header. Then, it authenticates the packet from the ESP trailer to the ESP header.
11. The security device sends the encrypted and authenticated packet destined for 2.2.2.2 through the outgoing interface (ethernet3) to the external router at 1.1.1.250.

Paris (Recipient)

1. The packet arrives at 2.2.2.2, which is the IP address of ethernet3, an interface bound to the Untrust zone.
2. Using the SPI, destination IP address, and IPsec protocol contained in the outer packet header, the IPsec module attempts to locate an active Phase 2 SA with the initiating peer along with the keys to authenticate and decrypt the packet. The Phase 2 SA check can produce one of the following three results:
   - If the IPsec module discovers an active Phase 2 SA with the peer, it proceeds to step 4.
   - If the IPsec module does not discover an active Phase 2 SA with the peer but it can match an inactive Phase 2 SA using the source IP address but not the SPI, it drops the packet, makes an event log entry, and sends a notification that it received a bad SPI to the initiating peer.
   - If the IPsec module does not discover an active Phase 2 SA with that peer, it drops the packet and triggers the IKE module.
3. The IKE module checks if an active Phase 1 SA exists with the remote peer. The Phase 1 SA check can produce either of the following results:
   - If the IKE module discovers an active Phase 1 SA with the peer, it uses this SA to negotiate a Phase 2 SA.
   - If the IKE module does not discover an active Phase 1 SA with that peer, it begins Phase 1 negotiations in main mode, and then Phase 2 negotiations.
4. The IPsec module performs an anti-replay check. This check can produce one of two results:
   - If the packet fails the anti-replay check, because it detects a sequence number that the security device has already received, the security device drops the packet.
   - If the packet passes the anti-replay check, the security device proceeds to the next step.
5. The IPsec module attempts to authenticate the packet. The authentication check can produce one of two results:
   - If the packet fails the authentication check, the security device drops the packet.
   - If the packet passes the authentication check, the security device proceeds to the next step.
6. Using the Phase 2 SA and keys, the IPsec module decrypts the packet, uncovering its original source address (10.1.1.5) and its ultimate destination (10.2.2.5). It learns that the packet came through vpn1, which is bound to tunnel.1. From this point forward, the security device treats the packet as if its ingress interface is tunnel.1 instead of ethernet3. It also adjusts the anti-replay sliding window at this point.
7. If you have enabled SCREEN options for the Untrust zone, the security device activates the SCREEN module at this point. SCREEN checking can produce one of the following three results:

- If a SCREEN mechanism detects anomalous behavior for which it is configured to block the packet, the security device drops the packet and makes an entry in the event log.
- If a SCREEN mechanism detects anomalous behavior for which it is configured to record the event but not block the packet, the security device records the event in the SCREEN counters list for ethernet3 and proceeds to the next step.
- If the SCREEN mechanisms detect no anomalous behavior, the security device proceeds to the next step.

8. The session module performs a session lookup, attempting to match the packet with an existing session. It then either performs First Packet Processing or Fast Processing.

   If the packet matches an existing session, the security device performs Fast Processing, using the information available from the existing session entry to process the packet. Fast Processing bypasses all but the last two steps (encrypting the packet and forwarding it) because the information generated by the bypassed steps has already been obtained during the processing of the first packet in the session.
9. The address-mapping module checks if a Mapped IP (MIP) or Virtual IP (VIP) configuration uses the destination IP address 10.2.2.5. Because 10.2.2.5 is not used in a MIP or VIP configuration, the security device proceeds to the next step.
10. The route module first uses the ingress interface to determine the virtual router to use for the route lookup; in this case, the trust-vr. It then performs a route lookup for 10.2.2.5 in the trust-vr and discovers that it is accessed through ethernet1. By determining the ingress interface (tunnel.1) and the egress interface (ethernet1), the security device can thereby determine the source and destination zones. The tunnel.1 interface is bound to the Untrust zone, and ethernet1 is bound to the Trust zone. The security device can now do a policy lookup.
11. The policy engine checks its policy list from the Untrust zone to the Trust zone and finds a policy that grants access.
12. The security device forwards the packet through ethernet1 to its destination at 10.2.2.5.

Addendum: Policy-Based VPN

The packet flow for a policy-based VPN configuration differs from that of a route-based VPN configuration at two points: the route lookup and the policy lookup.

Tokyo (Initiator)

The first stages of the outbound packet flow are the same for both route-based and policy-based VPN configurations until the route lookup and subsequent policy lookup occur:

- **Route Lookup:** To determine the destination zone, the route module does a route lookup for 10.2.2.5. Not finding an entry for that specific address, the route module resolves it to a route through ethernet3, which is bound to the Untrust zone. By determining the ingress and egress interfaces, the security device has

thereby determined the source and destination zones, and can now perform a policy lookup.

- **Policy Lookup:** The policy engine does a policy lookup between the Trust and Untrust zones. The lookup matches the source address and zone, destination address and zone, and service and finds a policy that references a VPN tunnel named vpn1.

The security device then forwards the packet through ethernet1 to its destination at 10.2.2.5.

Paris (Recipient)

Most stages of the inbound packet flow on the recipient's end are the same for both route-based and policy-based VPN configurations except that the tunnel is not bound to a tunnel interface, but to a tunnel zone. The security device learns that the packet came through vpn1, which is bound to the Untrust-Tun tunnel zone, whose carrier zone is the Untrust zone. Unlike route-based VPNs, the security device considers ethernet3 to be the ingress interface of the decrypted packet—not tunnel.1.

The flow changes after packet decryption is complete. At this point, the route and policy lookups differ:

- **Route Lookup:** The route module performs a route lookup for 10.2.2.5 and discovers that it is accessed through ethernet1, which is bound to the Trust zone. By learning that the Untrust zone is the source zone (because vpn1 is bound to the Untrust-Tun tunnel zone, whose carrier zone is the Untrust zone) and by determining the destination zone based on the egress interface (ethernet1 is bound to the Trust zone), the security device can now check for a policy from the Untrust to the Trust zones that references vpn1.
- **Policy Lookup:** The policy engine checks its policy list from the Untrust zone to the Trust zone and finds a policy that references a VPN tunnel named vpn1 and that grants access to 10.2.2.5.

The security device then forwards the packet to its destination.

Tunnel Configuration Guidelines

This section offers some guidelines for configuring VPN tunnels. When configuring an IPsec VPN tunnel, you might want to consider the following:

- ScreenOS supports up to four proposals for Phase 1 negotiations and up to four proposals for Phase 2 negotiations. A peer must be configured to accept at least one Phase 1 proposal and one Phase 2 proposal proffered by the other peer. For information about Phase 1 and Phase 2 IKE negotiations, see "Tunnel Negotiation" on page 715.
- If you want to use certificates for authentication and there is more than one local certificate loaded on the security device, you must specify which certificate you want each VPN tunnel configuration to use. For more information about certificates, see "Public Key Cryptography" on page 741.
- For a basic policy-based VPN:

- Use user-defined addresses in the policy, not the pre-defined address “Any”.
- The addresses and service specified in policies configured at both ends of the VPN must match.
- Use symmetric policies for bidirectional VPN traffic.

- The proxy ID for both peers must match, which means that the service specified in the proxy ID for both peers is the same, and the local IP address specified for one peer is the same as the remote IP address specified for the other peer.

NOTE: The proxy ID is a three-part tuple consisting of local IP address–remote IP address–service.

- For a route-based VPN configuration, the proxy ID is user-configurable.
- For a policy-based VPN configuration, the security device—by default—derives the proxy ID from the source address, destination address, and service specified in the policy that references that VPN tunnel in the policy list. You can also define a proxy ID for a policy-based VPN that supersedes the derived proxy ID.

The simplest way to ensure that the proxy IDs match is to use 0.0.0.0/0 for the local address, 0.0.0.0/0 for the remote address, and “any” for the service. Instead of using the proxy ID for access control, you use policies to control the traffic to and from the VPN. For examples of VPN configurations with user-configurable proxy IDs, see the route-based VPN examples in “Site-to-Site Virtual Private Networks” on page 801.

NOTE: When the remote address is the virtual internal address of a dialup VPN client, use 255.255.255.255/32 for the remote IP address /netmask in the proxy ID.

- As long as the peers' proxy ID settings match, it does not matter if one peer defines a route-based VPN and the other defines a policy-based VPN. If peer-1 uses a policy-based VPN configuration and peer-2 uses a route-based VPN configuration, then peer-2 must define a proxy ID that matches the proxy ID derived from peer-1's policy. If peer-1 performs Source Network Address Translation (NAT-src) using a DIP pool, use the address and netmask for the DIP pool as the remote address in peer-2's proxy ID. For example:

When the DIP Pool Is:	Use This in the Proxy ID:
1.1.1.8 – 1.1.1.8	1.1.1.8/32
1.1.1.20 – 1.1.1.50	1.1.1.20/26
1.1.1.100 – 1.1.1.200	1.1.1.100/25
1.1.1.0 – 1.1.1.255	1.1.1.0/24

For more information about proxy IDs when used with NAT-src and NAT-dst, see "VPN Sites with Overlapping Addresses" on page 863.

NOTE: Peer-1 can also define a proxy ID that matches peer-2's proxy ID. Peer-1's user-defined proxy ID supersedes the proxy ID that the security device derives from the policy components.

- Because proxy IDs support either a single service or all services, the service in a proxy ID derived from a policy-based VPN referencing a service group is considered as "any".
- When both peers have static IP addresses, they can each use the default IKE ID, which is their IP addresses. When a peer or dialup user has a dynamically assigned IP address, that peer or user must use another type of IKE ID. An FQDN is a good choice for a dynamic peer and a U-FQDN (email address) is a good choice for a dialup user. You can use both FQDN and U-FQDN IKE ID types with preshared keys and certificates (if the FQDN or U-FQDN appears in the SubjectAltName field in the certificate). If you use certificates, the dynamic peer or dialup user can also use all or part of the ASN1-DN as the IKE ID.

Route-Based Virtual Private Network Security Considerations

Although route changes do not affect policy-based VPNs, route-based VPNs are a different matter. The security device can route packets through a route-based VPN tunnel with a combination of static routes and dynamic routing protocols. As long as no route change occurs, the security device consistently encrypts and forwards packets destined for tunnel interfaces bound to route-based VPN tunnels.

However, when using VPN monitoring with a route-based VPN tunnel configuration, the state of a tunnel might change from up to down. When this occurs, all route table

entries referencing the tunnel interface bound to that tunnel change to inactive. Then, when the security device does a route lookup for traffic originally intended to be encrypted and sent through a tunnel bound to that tunnel interface, it bypasses the route referencing the tunnel interface and searches for a route with the next longest match. The route that it finds might be the default route. Using this route, the security device would then send the traffic unencrypted (that is, in *clear* or *plain text*) out through a non-tunnel interface to the public WAN.

To avoid rerouting traffic originally intended for a VPN tunnel to the public WAN in clear text, you can configure the security device to reroute such traffic to another tunnel, reroute it to a leased line, or just drop it, by using one of the following work-arounds:

- "Null Route" on page 794 (drops traffic when the route to the tunnel interface becomes inactive)
- "Dialup or Leased Line" on page 796 (reroutes traffic to an alternate secure path when the route to the tunnel interface becomes inactive)
- "Decoy Tunnel Interface" on page 799 (drops traffic when the route to the tunnel interface becomes inactive)
- "Virtual Router for Tunnel Interfaces" on page 799 (drops traffic when the route to the tunnel interface becomes inactive)
- "Reroute to Another Tunnel" on page 800 (reroutes traffic to an alternate VPN tunnel when the route to the tunnel interface becomes inactive)

Null Route

If the state of a VPN tunnel changes to "down," the security device changes any route referencing that tunnel interface to " inactive." If the route to the tunnel interface becomes unavailable and the next choice is the default route (for example), then the security device uses the default route to forward the traffic originally intended for the VPN tunnel. To avoid sending traffic in plain text out to the public WAN when a route change occurs, you can make use of a null route. A null route targets the same destination address as the route through the tunnel interface, but it instead points the traffic to the Null interface. The Null interface is a logical interface that drops traffic sent to it. You give the null route a higher metric (farther from zero) than the route using the tunnel interface so that the null route is less preferred.

NOTE: Releases prior to ScreenOS 5.1.0 do not support a null interface. However, you can use a decoy tunnel interface to accomplish the same objective. For information, see "Decoy Tunnel Interface" on page 799.

For example, if you create a static route through tunnel.1 to a remote LAN with the IP address 10.2.2.0/24, it automatically receives the default value of 1 for its metric:

```
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.1
get route
```

```
...
Dest-Routes for <trust-vr> (4 entries)
```

	ID	IP-Prefix	Interface	Gateway	P	Pref	Mtr	Vsys
*	3	0.0.0.0/0	eth3	1.1.1.250	S	20	1	Root
*	2	1.1.1.0/24	eth3	0.0.0.0	C	0	0	Root
*	1	10.1.1.0/24	eth1	0.0.0.0	C	0	0	Root
*	4	*10.2.2.0/24*	**tun.1**	0.0.0.0	S	20	**1**	Root

In the above routing table, an asterisk (*) indicates that a route is active, S indicates a static route, and "C" indicates a connected route.

In the routing table above, the security device has two routes to reach any address in the 10.2.2.0/24 subnet. The first choice is route #4 because it has the longest match with that address. The second choice is the default route (0.0.0.0/0).

If you then add another route to 10.2.2.0/24 through the Null interface and give it a value greater than 1, that route becomes the second routing choice to any address in the 10.2.2.0/24 subnet. If the route to 10.2.2.0/24 through tunnel.1 becomes inactive, then the security device uses the route to the Null interface. The security device forwards traffic for 10.2.2.0/24 to that interface, and then drops it.

```
set vrouter trust-vr route 10.2.2.0/24 interface null metric 10
get route
...
Dest-Routes for <trust-vr> (5 entries)
```

	ID	IP-Prefix	Interface	Gateway	P	Pref	Mtr	Vsys
*	3	0.0.0.0/0	eth3	1.1.1.250	S	20	1	Root
*	2	1.1.1.0/24	eth3	0.0.0.0	C	0	0	Root
*	1	10.1.1.0/24	eth1	0.0.0.0	C	0	0	Root
	4	**10.2.2.0/24**	**tun.1**	0.0.0.0	S	20	1	Root
*	5	**10.2.2.0/24**	**null**	0.0.0.0	S	20	**10**	Root

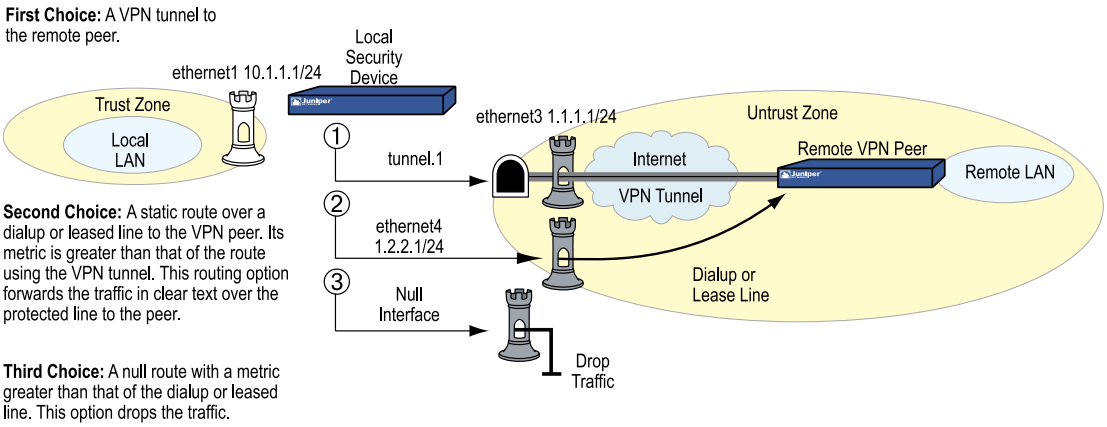
In the routing table above, the route to 10.2.2.0/24 through tunnel.1 is inactive (indicated by the absence of an asterisk in the far left column). Therefore, the security device searches for the next route that has the longest match to the destination address, and it finds route #5. (The next choice after route #5 is the default route with ID #3.) The security device then forwards traffic for 10.2.2.0/24 to the null interface, which drops the traffic. As a result, if the route using tunnel.1 becomes inactive, the security device drops traffic for 10.2.2.0/24 rather than using route #3 to forward it out ethernet3 as clear text to the router at 1.1.1.250.

Dialup or Leased Line

If you do not want to drop traffic to a remote peer when the tunnel to that peer becomes inactive, you can add an alternate route to that peer that flows over a dialup or leased line. This alternate route uses the same destination IP address as that in the route through the VPN tunnel, but it has a different egress interface and a less-preferred metric. If the route through the VPN tunnel becomes inactive, then the security device reroutes traffic to the remote peer through the dialup or leased line.

When using a dialup or leased line as the next-choice route, there is still the possibility that both the first- and second-choice routes can become inactive at the same time. Then the security device resorts to the third choice, which might be the default route. In anticipation of such a situation, you can make the dialup or leased line route the second choice and the null route the third choice (see “Null Route” on page 794). Figure 215 on page 796 shows how these options for handling a routing failover can work together.

Figure 215: Routing Failover Alternatives for VPN Traffic



VPN Failover to Leased Line or Null Route

In this example, you want traffic from the branch office behind Device A to reach the corporate network behind Device B over a secure VPN connection. If the tunnel fails, you then want traffic to flow over a leased line to the corporate office. If both the VPN tunnel and the leased line fail, you want Device A to drop the traffic rather than send it out onto the Internet in cleartext.

You create three routes on Device A to reach 10.2.2.0/24 and assign each a different metric:

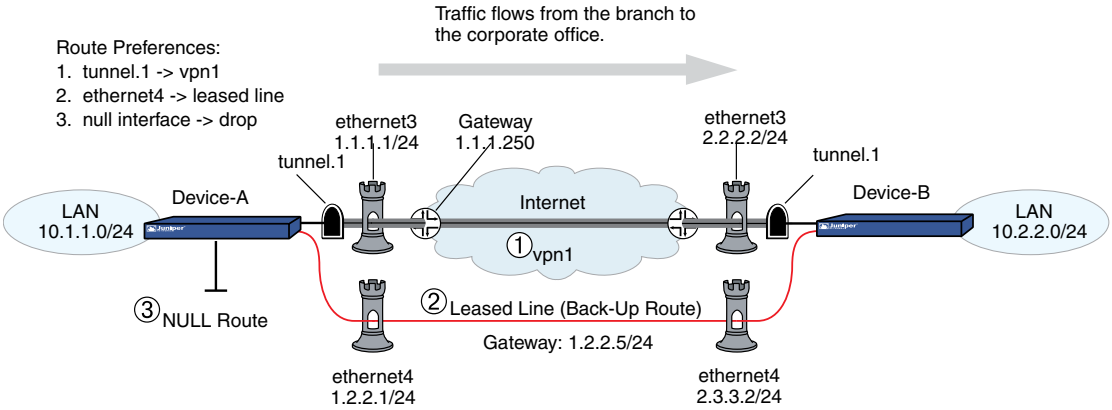
- **Preferred Route**—use tunnel.1, which is bound to vpn1 (metric = 1)
- **Secondary Route**—use ethernet4 and the gateway at 1.2.2.5 to use the leased line (metric = 2)
- **Tertiary Route**—use the null interface to drop traffic (metric = 10)

When you create the preferred route, you use the default metric for a static route, which is 1. You assign a metric of 2 to the secondary route; that is, the backup route over the leased line (shown in Figure 216 on page 797). The metric is less than that of the preferred route through the VPN tunnel. The security device does not use the secondary route unless the preferred route through the VPN tunnel fails.

Finally, you add a NULL route with a metric of 10. If the preferred route fails and then the secondary route fails, the security device drops all packets. All the security zones are in the trust-vr routing domain.

NOTE: This example shows only the configuration for four routes—three for the failovers plus the default route—on Device A.

Figure 216: Routing Failover to a Leased Line and Then to a Null Route



WebUI (Device A)

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250
Metric: 1

Network > Routing > Routing Entries > trust-vr New: Enter the following and then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0
Metric: 1

Network > Routing > Routing Entries > trust-vr New: Enter the following and then click **OK**:

```
Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
   Interface: ethernet4
   Gateway IP Address: 1.2.2.5
Metric: 2
```

Network > Routing > Routing Entries > trust-vr New: Enter the following and then click **OK**:

```
Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
   Interface: Null
   Gateway IP Address: 0.0.0.0
Metric: 10
```

CLI (Device A)

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.1
set vrouter trust-vr route 10.2.2.0/24 interface ethernet4 gateway 1.2.2.5 metric 2
set vrouter trust-vr route 10.2.2.0/24 interface null metric 10
save
```

You can verify that the new routes are present by executing the **get route** command.

```
device-C-> get route
IPv4 Dest-Routes for <untrust-vr> (0 entries)
---------------------------------------------

H: Host C: Connected S: Static A: Auto-Exported
I: Imported R: RIP P: Permanent D: Auto-Discovered
iB: IBGP eB: EBGP O: OSPF E1: OSPF external type 1
E2: OSPF external type 2

IPv4 Dest-Routes for <trust-vr> (7 entries)
---------------------------------------------
  ID        IP-Prefix      Interface       Gateway   P Pref    Mtr     Vsys
---------------------------------------------
*  8        0.0.0.0/0        eth1/1    10.100.37.1   S   20     1     Root
*  7        1.1.1.1/32       eth1/2        0.0.0.0   H    0     0     Root
*  3     192.168.1.1/32         mgt        0.0.0.0   H    0     0     Root
*  2     192.168.1.0/24         mgt        0.0.0.0   C    0     0     Root
*  4     10.100.37.0/24       eth1/1        0.0.0.0   C    0     0     Root
*  5   10.100.37.170/32       eth1/1        0.0.0.0   H    0     0     Root
*  6        1.1.1.0/24       eth1/2        0.0.0.0   C    0     0     Root
```

The route table entry with ID 5 directs traffic for 10.2.2.0/24 to tunnel.1 and then through the VPN tunnel. It is the preferred route for traffic to reach the 10.2.2.0 network. If that tunnel fails, the next best route is route entry 6 over a leased line through a gateway at 1.2.2.5. If the connection for route entry 6 fails, route entry 7

becomes the next best route, and the security device directs traffic for 10.2.2.0/24 to the null interface, which then drops it.

Decoy Tunnel Interface

Instead of failing over traffic from a VPN tunnel to a null interface (and then dropping it), you can use a nonfunctioning tunnel interface to accomplish the same objective.

NOTE: Releases prior to ScreenOS 5.1.0 do not support a null interface (see "Null Route" on page 794). However, you can use a decoy tunnel interface to accomplish the same objective.

To set up a decoy tunnel interface, do the following:

1. Create a second tunnel interface, but do not bind it to a VPN tunnel. Instead, bind it to a tunnel zone that is in the same virtual routing domain as the first tunnel interface.

NOTE: If a tunnel interface is bound to a tunnel zone, its status is always up.

2. Define a second route to the same destination using this second tunnel interface, and assign it a higher metric (farther from zero) than the preferred route.

   If the state of the functioning tunnel interface changes from up to down and the route table entry referencing that interface becomes inactive, all subsequent route lookups find this second route to the nonfunctioning tunnel interface. The security device forwards traffic to the second tunnel interface and because it is not bound to a VPN tunnel, the device drops the traffic.

Virtual Router for Tunnel Interfaces

To avoid the case where the route through a VPN tunnel becomes deactivated and then fails over traffic originally intended to pass through the tunnel to the default route, you can create a special virtual routing domain exclusively for VPN traffic. To set this up, take the following steps:

1. Create a separate virtual router to use for all routes pointing to tunnel interfaces and name it, for example, "VR-VPN."
2. Create a security zone—named, for example, "VPN zone" —and bind it to VR-VPN.
3. Bind all tunnel interfaces to the VPN zone, and also put all addresses for remote sites that you want to reach through VPN tunnels in this zone.
4. Configure static routes in all other virtual routers to VR-VPN for traffic that you want encrypted and sent through the tunnels. If necessary, define static routes for decrypted traffic from VR-VPN to the other virtual routers. Such routes are necessary to allow inbound VPN traffic through the tunnel if it is initiated from the remote site.

If the state of a tunnel interface changes from up to down, the security device still forwards traffic to VR-VPN, where—because the state of the route to that interface is now inactive and there are no other matching routes—the security device drops the traffic.

Reroute to Another Tunnel

You can configure two or more VPN tunnels to the same remote peer. If one tunnel goes down, the security device can then reroute traffic through another VPN tunnel. For information and examples about configuring redundant VPN tunnels, see the following:

- Active-to-Backup Tunnel Failover on page 1825
- Configuring Dual Active Tunnels on page 1847
- Configuring Tunnel Failover Weights on page 1854

Chapter 22

Site-to-Site Virtual Private Networks

This chapter explains how to configure a site-to-site virtual private network (VPN) tunnel between two Juniper Networks security devices. It examines route-based and policy-based VPN tunnels, presents the various elements that you must consider when setting up a tunnel, and offers several examples.

This chapter contains the following sections:

Site-to-Site VPN Configurations

An IPsec VPN tunnel exists between two gateways, and each gateway needs an IP address. When both gateways have static IP addresses, you can configure the following kinds of tunnels:

- Site-to-Site VPN, AutoKey IKE tunnel (with a preshared key or certificates)
- Site-to-Site VPN, Manual Key tunnel

When one gateway has a static address and the other has a dynamically assigned address, you can configure the following kind of tunnel:

- Dynamic Peer Site-to-Site VPN, AutoKey IKE tunnel (with a preshared key or certificates)

As used here, a static site-to-site VPN involves an IPsec tunnel connecting two sites, each with a security device operating as a secure gateway. The physical interface or subinterface used as the outgoing interface on both devices has a fixed IP address, and the internal hosts also have static IP addresses. If the security device is in transparent mode, it uses the VLAN1 address as the IP address for the outgoing interface. With a static site-to-site VPN, hosts at either end of the tunnel can initiate the VPN tunnel setup because the IP address of the remote gateway remains constant and thus reachable.

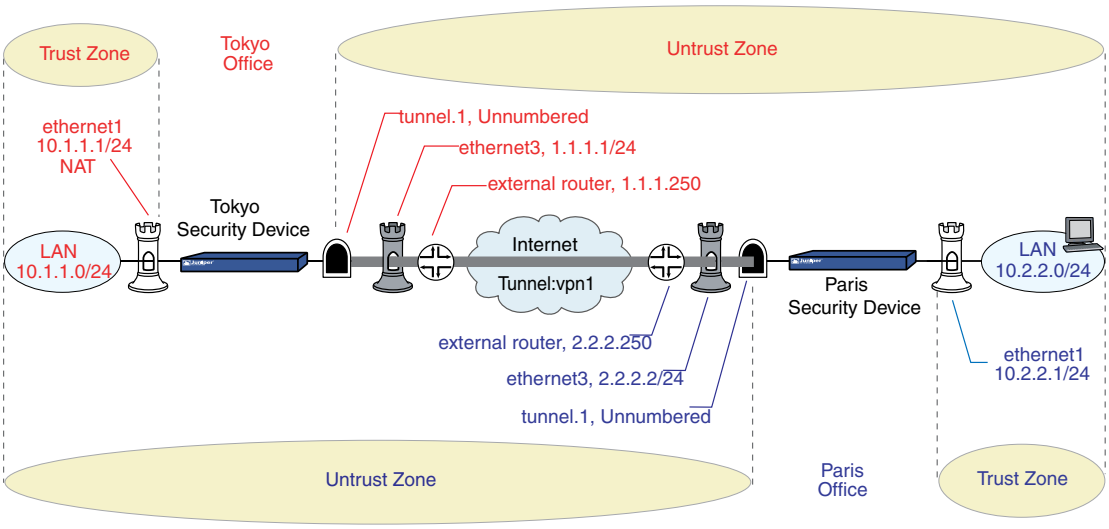
If the outgoing interface of one of the security devices has a dynamically assigned IP address, that device is termed a dynamic peer and the VPN is configured differently.

With a dynamic peer site-to-site VPN, only hosts behind the dynamic peer can initiate the VPN tunnel setup because only their remote gateway has a fixed IP address and is thus reachable from their local gateway. However, after a tunnel is established between a dynamic peer and a static peer, hosts behind either gateway can initiate VPN traffic if the destination hosts have fixed IP addresses.

NOTE: For background information about the available VPN options, see "Internet Protocol Security" on page 707. For guidance when choosing among the various options, see "Virtual Private Network Guidelines" on page 769.

The configuration of a site-to-site VPN tunnel requires the coordination of the tunnel configuration with that of other settings—interfaces, addresses, routes, and policies. The three example VPN configurations in this section are set in the following context: an office in Tokyo wants to communicate securely with an office in Paris through an IPsec VPN tunnel.

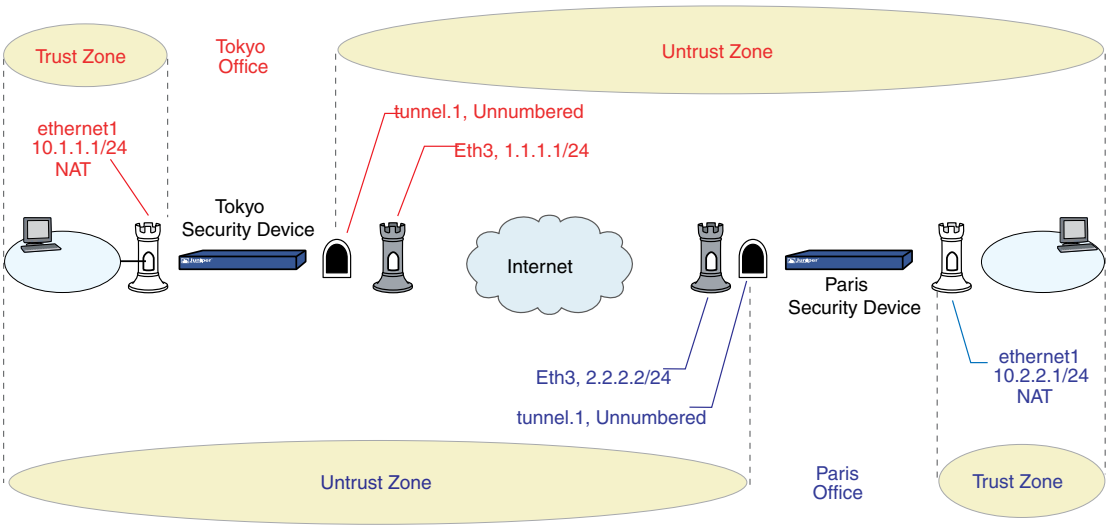
Figure 217: Site-to-Site VPN Tunnel Configuration



The administrators in both offices configure the following settings:

- Interfaces – Security Zones and Tunnel
- Addresses
- VPN (one of the following)
  - AutoKey IKE
  - Dynamic Peer
  - Manual Key
- Routes
- Policies

Figure 218: Site-to-Site Tunnel Configuration—Interfaces



1. **Interfaces – Security Zones and Tunnel**

   The admin at the Tokyo office configures the security zone and tunnel interfaces with the settings that appear in the upper half of Figure 218 on page 803. The admin at the Paris office does likewise with the settings that appear in the lower half of the figure.

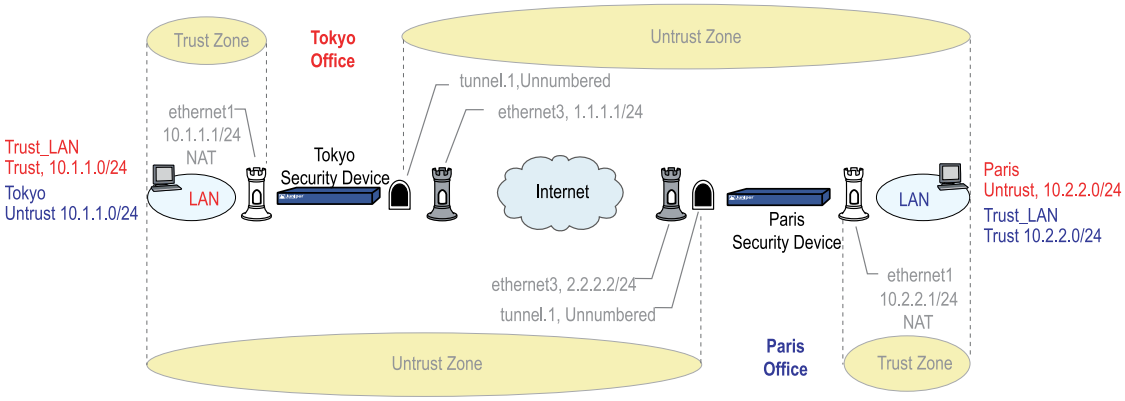
   Ethernet3 is going to be the outgoing interface for VPN traffic and the remote gateway for VPN traffic sent from the other end of the tunnel.

   Ethernet1 is in NAT mode so each admin can assign IP addresses to all the internal hosts, yet when traffic passes from the Trust zone to the Untrust zone, the security device translates the source IP address in the packet headers to the address of the Untrust zone interface, ethernet3—1.1.1.1 for Tokyo, and 2.2.2.2 for Paris.

   For a route-based VPN, each admin binds the tunnel interface tunnel.1 to the VPN tunnel vpn1. By defining a route to the address space of the remote office LAN, the security device can direct all traffic bound for that LAN to the tunnel.1 interface and thus through the tunnel to which tunnel.1 is bound.

   Because policy-based NAT services are not needed, a route-based VPN configuration does not require tunnel.1 to have an IP address/netmask, and a policy-based VPN configuration does not even require a tunnel interface.

Figure 219: Site-to-Site Tunnel Configuration—Addresses



2. **Addresses**

   The admins define addresses for later use in inbound and outbound policies. The admin at the Tokyo office defines the addresses that appear in the upper half of Figure 219 on page 804. The admin at the Paris office does likewise with the addresses that appear in the lower half of the figure.

   For policy-based VPNs, the security device derives proxy IDs from policies. Because the proxy IDs used by the security devices at both ends of the VPN tunnel must match perfectly, you cannot use the predefined address “ANY,” whose IP address is 0.0.0.0/0, at one end of the tunnel if you use a more specific address at the other end. For example:

If the proxy ID in Tokyo is as follows:

From: 0.0.0.0/0
To: 10.2.2.0/24
Service: ANY

And if the proxy ID in Paris is as follows:

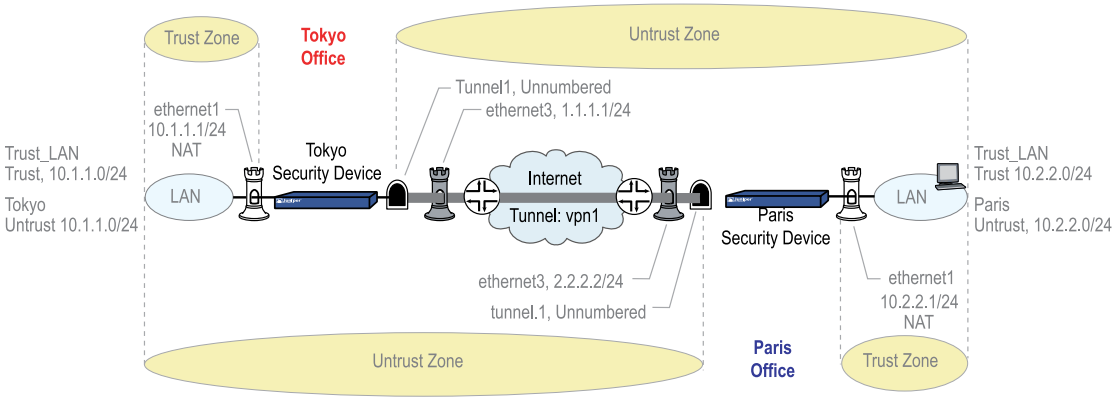
To: 10.1.1.0/24
From: 10.2.2.0/24
Service: ANY

Then the proxy IDs do not match, and IKE negotiations will fail.

NOTE: Beginning with ScreenOS 5.0.0, you can also define proxy IDs for VPN tunnels referenced in policy-based VPN configurations.

For route-based VPNs, you can use "0.0.0.0/0–0.0.0.0/0–any" to define the local and remote IP addresses and service type for a proxy ID. You can then use more restrictive policies to filter the inbound and outbound VPN traffic by source address, destination address, and service type.

Figure 220: Site-to-Site Tunnel Configuration—VPN Tunnel



3. **VPN**

You can configure one of the following three VPNs:

- AutoKey IKE

  The AutoKey IKE method uses a preshared key or a certificate to refresh—that is, change—the encryption and authentication keys automatically at user-defined intervals (known as key lifetimes). Essentially, frequently updating these keys strengthens security, although excessively short lifetimes might reduce overall performance.

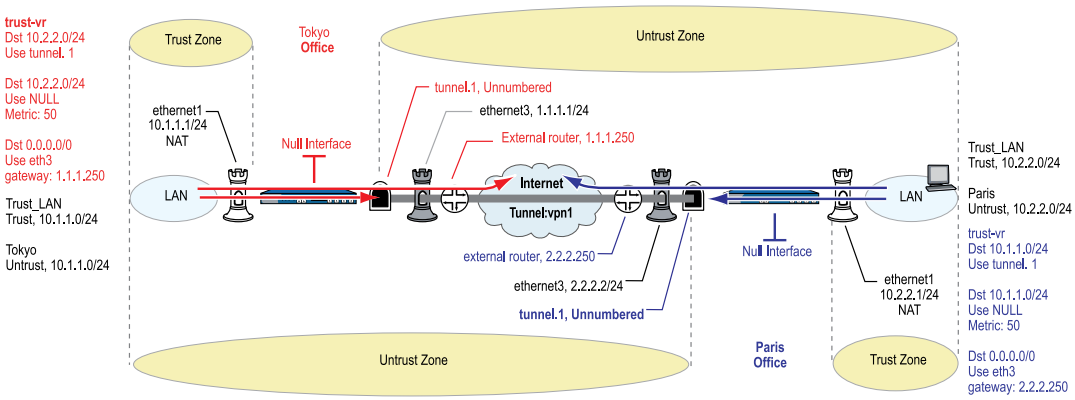
- Dynamic Peer

A dynamic peer is a remote gateway that has a dynamically assigned IP address. Because the IP address of the remote peer might be different each time IKE negotiations begin, hosts behind the peer must initiate VPN traffic. Also—if using a preshared key for authentication—the peer must send an IKE ID during the first message of Phase 1 negotiations in aggressive mode to identify itself.

- Manual Key

  The Manual Key method requires you to set and update the encryption and authentication keys manually. This method is a viable option for a small set of VPN tunnels.

Figure 221: Site-to-Site Tunnel Configuration—Routes



4. **Routes**

   The admins at each site must configure at least the following routes:

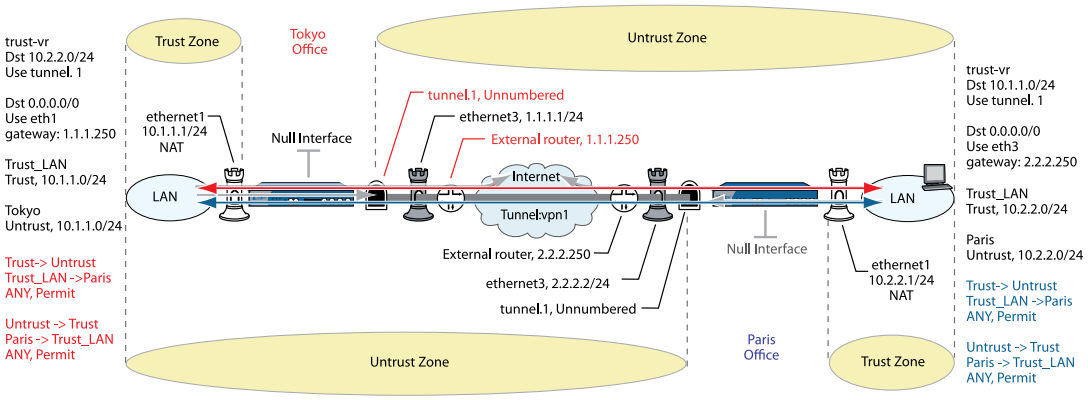
   - A route for traffic to reach an address on the remote LAN to use tunnel.1
   - A default route for all other traffic, including the outer VPN tunnel traffic, to reach the internet through ethernet3 and then the external router beyond it—1.1.1.250 for the Tokyo office and 2.2.2.250 for Paris. The external router is the default gateway to which the security device forwards any traffic for which it does not have a specific route in its routing table.

NOTE: If the security device at the Tokyo office receives its external IP address dynamically from its ISP (that is, from the point of view of the Paris office, the security device at the Tokyo office is its dynamic peer), then the ISP automatically provides the Tokyo device with its default gateway IP address.

   - A null route so that if the state of tunnel.1 ever changes to “down” and any route referencing tunnel.1 becomes deactivated, the security device does not use the default route to forward traffic destined to the remote LAN unencrypted out ethernet3. A null route uses the remote LAN as the destination address, but it points traffic to the Null interface, a logical interface that drops traffic sent to it. You give the null route a higher metric (farther

from zero) than the route to the remote LAN using tunnel.1, making the null route less preferred than the route referencing the tunnel.1 interface.

Figure 222: Site-to-Site Tunnel Configuration—Policies



5. **Policies**

   The admins at each site define policies to permit traffic between the two offices:

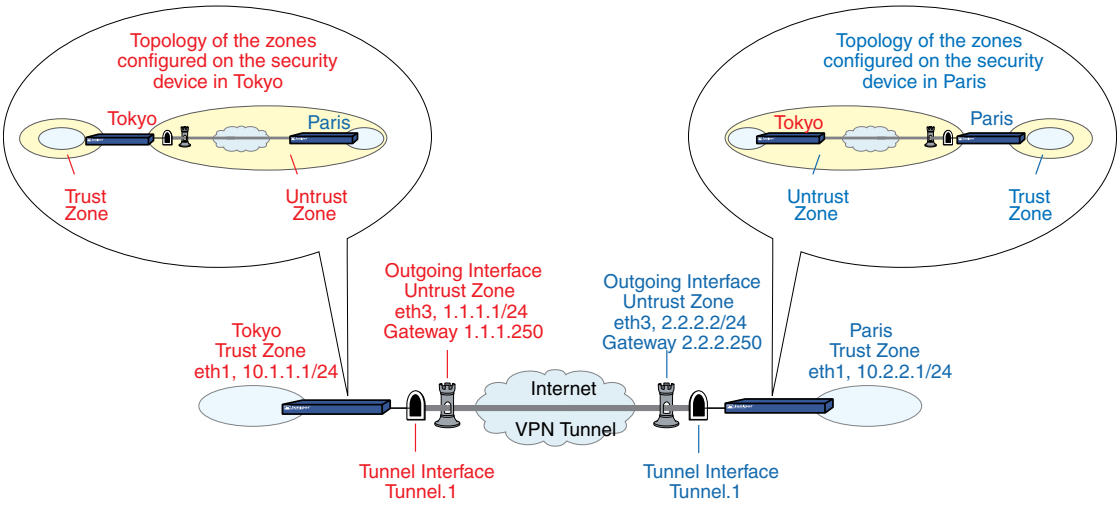
   - A policy permitting any kind of traffic from “Trust_LAN” in the Trust zone to “Paris” or “Tokyo” in the Untrust zone
   - A policy permitting any kind of traffic from “Paris” or “Tokyo” in the Untrust zone to “Trust_LAN” in the Trust zone

     Because the preferred route to the remote site specifies tunnel.1, which is bound to the VPN tunnel vpn1, the policy does not need to reference the VPN tunnel.

Route-Based Site-to-Site VPN, AutoKey IKE

In this example, an AutoKey IKE tunnel using either a preshared secret or a pair of certificates (one at each end of the tunnel) provides the secure connection between the Tokyo and Paris offices. For the Phase 1 and Phase 2 security levels, you specify one Phase 1 proposal—either pre-g2-3des-sha for the preshared key method or rsa-g2-3des-sha for certificates—and select the predefined “Compatible” set of proposals for Phase 2. All zones are in the trust-vr.

Figure 223: Route-Based Site-to-Site VPN, AutoKey IKE



Setting up a route-based AutoKey IKE tunnel using either a preshared secret or certificates, involves the following steps:

1. Assign IP addresses to the physical interfaces bound to the security zones and to the tunnel interface.
2. Configure the VPN tunnel, designate its outgoing interface in the Untrust zone, bind it to the tunnel interface, and configure its proxy-ID.
3. Enter the IP addresses for the local and remote endpoints in the address books for the Trust and Untrust zones.
4. Enter a default route to the external router in the trust-vr, a route to the destination through the tunnel interface, and a null route to the destination. You assign a higher metric (farther from zero) to the null route so that it becomes the next-choice route to the destination. Then, if the state of the tunnel interface changes to “down” and the route referencing that interface becomes inactive, the security device uses the null route, which essentially drops any traffic sent to it, rather than the default route, which forwards unencrypted traffic.
5. Set up policies for VPN traffic to pass between each site.

In the following examples, the preshared key is h1p8A24nG5. It is assumed that both participants already have RSA certificates and are using Entrust as the certificate authority (CA). (For information about obtaining and loading certificates, see “Certificates and CRLs” on page 746.)

WebUI (Tokyo)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24

Select the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet3 (trust-vr)

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Trust_LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

Policy > Policy Elements > > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Paris_Office
IP Address/Domain Name:
IP/Netmask: (select), 10.2.2.0/24
Zone: Untrust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_Paris
Security Level: Custom
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 2.2.2.2

Preshared Key

Preshared Key: h1p8A24nG5
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (for Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)

(or)

Certificates

Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (for Custom Security Level): rsa-g2-3des-sha
Preferred certificate (optional)
Peer CA: Entrust
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: Tokyo_Paris
Security Level: Compatible
Remote Gateway:
  Predefined: (select), To_Paris

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Security Level: Compatible
Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 10.1.1.0/24
Remote IP / Netmask: 10.2.2.0/24
Service: ANY

4. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
  Interface: ethernet3
  Gateway IP Address: 1.1.1.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
  Interface: Tunnel.1
  Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
Interface: Null
Gateway IP Address: 0.0.0.0
Metric: 10

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: To Paris
Source Address: Trust_LAN
Destination Address: Paris_Office
Service: ANY
Action: Permit
Position at Top: (select)

Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK**:

Name: From Paris
Source Address: Paris_Office
Destination Address: Trust_LAN
Service: ANY
Action: Permit
Position at Top: (select)

WebUI (Paris)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.2.2.1/24

Select the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 2.2.2.2/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet3 (trust-vr)

2. **Addresses**

Policy > Policy Elements > > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Trust_LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.2.2.0/24
Zone: Trust

Policy > Policy Elements > > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Tokyo_Office
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Untrust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_Tokyo
Security Level: Custom
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 1.1.1.1

Preshared Key

Preshared Key: h1p8A24nG5
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)

(or)

Certificates

Outgoing Interface: ethernet3

Security Level: Custom
Phase 1 Proposal (for Custom Security Level): rsa-g2-3des-sha
Preferred certificate (optional)
Peer CA: Entrust
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

Name: Paris_Tokyo
Security Level: Compatible
Remote Gateway:
Predefined: (select), To_Tokyo

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Compatible
Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 10.2.2.0/24
Remote IP / Netmask: 10.1.1.0/24
Service: ANY

4. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
  Interface: ethernet3
  Gateway IP Address: 2.2.2.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.1.0/24
Gateway: (select)
  Interface: Tunnel.1
  Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.1.0/24
Gateway: (select)
  Interface: Null
  Gateway IP Address: 0.0.0.0
  Metric: 10

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: To_Tokyo
Source Address:
  Address Book Entry: (select), Trust_LAN
Destination Address:
  Address Book Entry: (select), Tokyo_Office
Service: ANY
Action: Permit
Position at Top: (select)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Name: From_Tokyo
Source Address:
  Address Book Entry: (select), Tokyo_Office
Destination Address:

Address Book Entry: (select), Trust_LAN
Service: ANY
Action: Permit
Position at Top: (select)

CLI (Tokyo)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

2. **Addresses**

```
set address trust Trust_LAN 10.1.1.0/24
set address untrust Paris_Office 10.2.2.0/24
```

3. **VPN**

Preshared Key

```
set ike gateway To_Paris address 2.2.2.2 main outgoing-interface ethernet3
preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn Tokyo_Paris gateway To_Paris sec-level compatible
set vpn Tokyo_Paris bind interface tunnel.1
set vpn Tokyo_Paris proxy-id local-ip 10.1.1.0/24 remote-ip 10.2.2.0/24 any
```

(or)

```
set ike gateway To_Paris address 2.2.2.2 main outgoing-interface ethernet3
proposal rsa-g2-3des-sha
set ike gateway To_Paris cert peer-ca 1
set ike gateway To_Paris cert peer-cert-type x509-sig
set vpn Tokyo_Paris gateway To_Paris sec-level compatible
set vpn Tokyo_Paris bind interface tunnel.1
set vpn Tokyo_Paris proxy-id local-ip 10.1.1.0/24 remote-ip 10.2.2.0/24 any
```

NOTE: The number 1 is the CA ID number. To discover the CA's ID number, use the following command: **get ike ca**.

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.1
set vrouter trust-vr route 10.2.2.0/24 interface null metric 10
```

5. **Policies**

```
set policy top name “To Paris” from trust to untrust Trust_LAN Paris_Office any
 permit
set policy top name “From Paris” from untrust to trust Paris_Office Trust_LAN
any permit
save
```

CLI (Paris)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

2. **Addresses**

```
set address trust Trust_LAN 10.2.2.0/24
set address untrust Tokyo_Office 10.1.1.0/24
```

3. **VPN**

Preshared Key

```
set ike gateway To_Tokyo address 1.1.1.1 main outgoing-interface ethernet3
preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn Paris_Tokyo gateway To_Tokyo sec-level compatible
set vpn Paris_Tokyo bind interface tunnel.1
set vpn Paris_Tokyo proxy-id local-ip 10.2.2.0/24 remote-ip 10.1.1.0/24 any
```

(or)

Certificate

```
set ike gateway To_Tokyo address 1.1.1.1 main outgoing-interface ethernet3
proposal rsa-g2-3des-sha
set ike gateway To_Tokyo cert peer-ca 1
set ike gateway To_Tokyo cert peer-cert-type x509-sig
set vpn Paris_Tokyo gateway To_Tokyo sec-level compatible
set vpn Paris_Tokyo bind interface tunnel.1
set vpn Paris_Tokyo proxy-id local-ip 10.2.2.0/24 remote-ip 10.1.1.0/24 any
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
set vrouter trust-vr route 10.1.1.0/24 interface tunnel.1
set vrouter trust-vr route 10.1.1.0/24 interface null metric 10
```

5. **Policies**

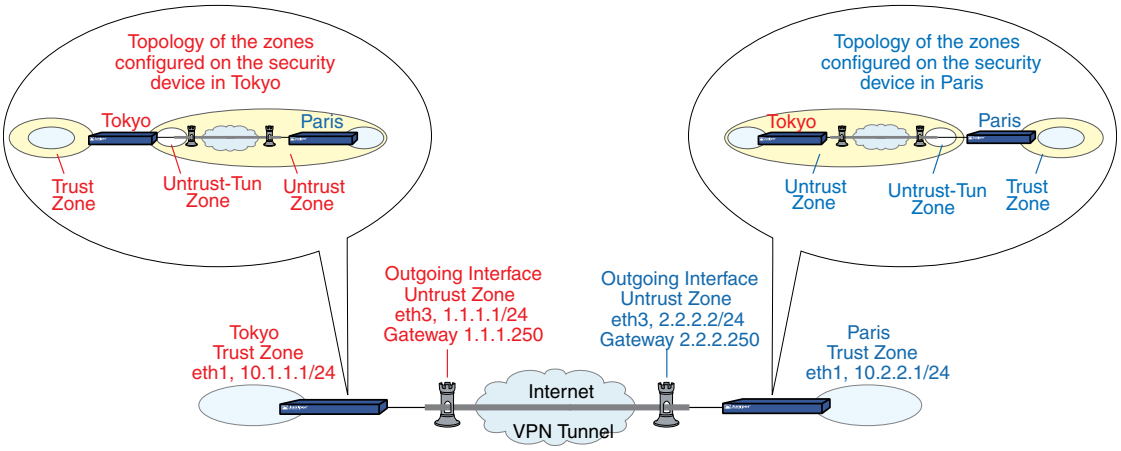
```
set policy top name “To Tokyo” from trust to untrust Trust_LAN Tokyo_Office any
 permit
```

```
set policy top name “From Tokyo” from untrust to trust Tokyo_Office Trust_LAN
 any permit
save
```

Policy-Based Site-to-Site VPN, AutoKey IKE

In this example, an AutoKey IKE tunnel using either a preshared secret or a pair of certificates (one at each end of the tunnel) provides the secure connection between the Tokyo and Paris offices. For the Phase 1 and Phase 2 security levels, you specify one Phase 1 proposal—either pre-g2-3des-sha for the preshared key method or rsa-g2-3des-sha for certificates—and select the predefined “Compatible” set of proposals for Phase 2. All zones are in the trust-vr.

Figure 224: Policy-Based Site-to-Site VPN, AutoKey IKE



Setting up the AutoKey IKE tunnel using AutoKey IKE, with either a preshared secret or certificates, involves the following steps:

1. Define security zone interface IP addresses.
2. Make address book entries for the local and remote end entities.
3. Define the remote gateway and key exchange mode, and specify either a preshared secret or a certificate.
4. Create the Autokey IKE VPN.
5. Set a default route to the external router.
6. Configure policies.

In the following examples, the preshared key is h1p8A24nG5. It is assumed that both participants already have RSA certificates and are using Entrust as the certificate authority (CA). (For information about obtaining and loading certificates, see “Certificates and CRLs” on page 746.)

WebUI (Tokyo)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Trust_LAN
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Paris_Office
   IP Address/Domain Name:
   IP/Netmask: (select), 10.2.2.0/24
   Zone: Untrust

3. **VPN**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: To_Paris
   Security Level: Custom
   Remote Gateway Type:
   Static IP Address: (select), IP Address/Hostname: 2.2.2.2

   Preshared Key

   Preshared Key: h1p8A24nG5
   Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **OK** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)

(or)

Certificates

Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **OK** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): rsa-g2-3des-sha
Mode (Initiator): Main (ID Protection)
Preferred certificate (optional)
Peer CA: Entrust
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: Tokyo_Paris
Security Level: Compatible
Remote Gateway: Predefined: (select), To_Paris

4. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
  Interface: ethernet3
  Gateway IP Address: 1.1.1.250

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: To/From Paris
Source Address:
  Address Book Entry: (select), Trust_LAN
Destination Address:
  Address Book Entry: (select), Paris_Office
Service: ANY
Action: Tunnel
Tunnel VPN: Tokyo_Paris
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

WebUI (Paris)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.2.2.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK:**

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 2.2.2.2/24

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Trust_LAN
   IP Address/Domain Name:
   IP/Netmask: (select), 10.2.2.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Tokyo_Office
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.0/24
   Zone: Untrust

3. **VPN**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: To_Tokyo
   Security Level: Custom
   Remote Gateway Type:
   Static IP Address: (select), IP Address/Hostname: 1.1.1.1

   Preshared Key

   Preshared Key: h1p8A24nG5
   Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)

(or)

Certificates

Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): rsa-g2-3des-sha
Mode (Initiator): Main (ID Protection)
Preferred certificate (optional)
Peer CA: Entrust
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK:**

VPN Name: Paris_Tokyo
Security Level: Compatible
Remote Gateway: Predefined: (select), To_Tokyo

4. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
  Interface: ethernet3
  Gateway IP Address: 2.2.2.250

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: To/From Tokyo
Source Address:
  Address Book Entry: (select), Trust_LAN
Destination Address:
  Address Book Entry: (select), Tokyo_Office
Service: ANY
Action: Tunnel
Tunnel VPN: Paris_Tokyo
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

CLI (Tokyo)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address trust Trust_LAN 10.1.1.0/24
set address untrust paris_office 10.2.2.0/24
```

3. **VPN**

Preshared Key

```
set ike gateway to_paris address 2.2.2.2 main outgoing-interface ethernet3
preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn tokyo_paris gateway to_paris sec-level compatible
```

(or)

Certificates

```
set ike gateway to_paris address 2.2.2.2 main outgoing-interface ethernet3
proposal rsa-g2-3des-sha
set ike gateway to_paris cert peer-ca 1
set ike gateway to_paris cert peer-cert-type x509-sig
set vpn tokyo_paris gateway to_paris sec-level compatible
```

NOTE: The number 1 is the CA ID number. To discover the CA's ID number, use the following command: **get ike ca**.

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

5. **Policies**

```
set policy top name "To/From Paris" from trust to untrust Trust_LAN paris_office
 any
   tunnel vpn tokyo_paris
set policy top name "To/From Paris" from untrust to trust paris_office Trust_LAN
 any
   tunnel vpn tokyo_paris
save
```

CLI (Paris)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
```

2. **Addresses**

```
set address trust Trust_LAN 10.2.2.0/24
set address untrust tokyo_office 10.1.1.0/24
```

3. **VPN**

Preshared Key

```
set ike gateway to_tokyo address 1.1.1.1 main outgoing-interface ethernet3
preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn paris_tokyo gateway to_tokyo sec-level compatible
```

(or)

Certificates

```
set ike gateway to_tokyo address 1.1.1.1 main outgoing-interface ethernet3
proposal rsa-g2-3des-sha
set ike gateway to_tokyo cert peer-ca 1
set ike gateway to_tokyo cert peer-cert-type x509-sig
set vpn paris_tokyo gateway to_tokyo tunnel proposal nopfs-esp-3des-sha
```

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
```

5. **Policies**

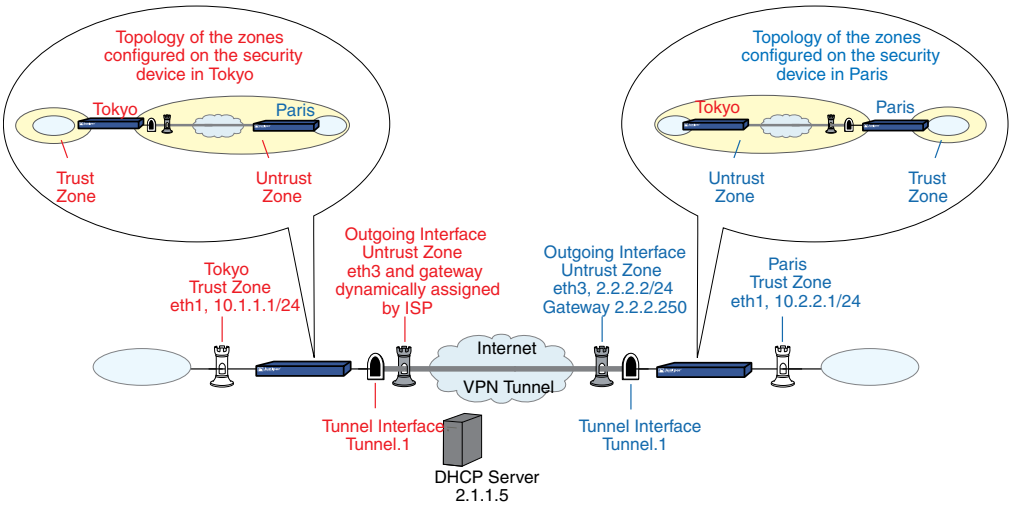
```
set policy top name “To/From Tokyo” from trust to untrust Trust_LAN tokyo_office
 any tunnel vpn paris_tokyo
set policy top name “To/From Tokyo” from untrust to trust tokyo_office Trust_LAN
 any tunnel vpn paris_tokyo
save
```

Route-Based Site-to-Site VPN, Dynamic Peer

In this example, an AutoKey IKE VPN tunnel using either a preshared key or a pair of certificates (one at each end of the tunnel) provides a secure connection between security devices protecting the Tokyo and Paris offices. The Untrust zone interface for the Paris security device has a static IP address. The ISP serving the Tokyo office assigns the IP address for the Untrust zone interface dynamically through DHCP. Because only the Paris security device has a fixed address for its Untrust zone, VPN

traffic must originate from hosts in the Tokyo office. After a tunnel has been established, traffic through the tunnel can originate from either end. All security and tunnel zones are in the trust-vr.

Figure 225: Route-Based Site-to-Site VPN, Dynamic Peer



The preshared key is h1p8A24nG5. It is assumed that both participants already have RSA certificates from the certificate authority (CA) Verisign and that the email address *pmason@abc.com* appears in the local certificate on Device A. (For information about obtaining and loading certificates, see "Certificates and CRLs" on page 746.) For the Phase 1 and Phase 2 security levels, you specify one Phase 1 proposal—either pre-g2-3des-sha for the preshared key method or rsa-g2-3des-sha for certificates—and select the "Compatible" set of proposals for Phase 2.

You enter three routes on the security devices at each end of the VPN tunnel:

- A default route to the external router in the trust-vr
- A route to the destination through the tunnel interface
- A null route to the destination. You assign a higher metric (farther from zero) to the null route so that it becomes the next-choice route to the destination. Then, if the state of the tunnel interface changes to "down" and the route referencing that interface becomes inactive, the security device uses the null route, which essentially drops any traffic sent to it, rather than the default route, which forwards unencrypted traffic.

Finally, you configure policies to permit bidirectional traffic between the two sites.

WebUI (Tokyo)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24

Select the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **Apply**:

Zone Name: Untrust

Enter the following, then click **OK**:

Obtain IP using DHCP: (select)

NOTE: You cannot specify the IP address of the DHCP server through the WebUI; however, you can do so through the CLI.

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet3 (trust-vr)

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Trust_LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Paris_Office
IP Address/Domain Name:
IP/Netmask: (select), 10.2.2.0/24
Zone: Untrust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_Paris
Security Level: Custom

Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 2.2.2.2

Preshared Key

Preshared Key: h1p8A24nG5
Local ID: pmason@abc.com
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Aggressive

(or)

Certificates

Local ID: pmason@abc.com
Outgoing Interface: ethernet3

NOTE: The U-FQDN "pmason@abc.com" must appear in the SubjectAltName field in the certificate.

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): rsa-g2-3des-sha
Mode (Initiator): Aggressive
Preferred Certificate (optional):
Peer CA: Verisign
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: Tokyo_Paris
Security Level: Compatible
Remote Gateway:
Predefined: (select), To_Paris

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface: (select), tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 10.1.1.0/24
Remote IP / Netmask: 10.2.2.0/24
Service: ANY

4. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 0.0.0.0

NOTE: The ISP provides the gateway IP address dynamically through DHCP.

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
Interface: Tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
Interface: Null
Gateway IP Address: 0.0.0.0
Metric: 10

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Trust_LAN
Destination Address:
Address Book Entry: (select), Paris_Office
Service: Any
Action: Permit
Position at Top: (select)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Paris_Office
Destination Address:
Address Book Entry: (select), Trust_LAN
Service: Any
Action: Permit
Position at Top: (select)

WebUI (Paris)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.2.2.1/24

Select the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address: 2.2.2.2/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet3 (trust-vr)

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Trust_LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.2.2.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Tokyo_Office
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Untrust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_Tokyo
Security Level: Custom
Remote Gateway Type:
Dynamic IP Address: (select), Peer ID: pmason@abc.com

Preshared Key

Preshared Key

Preshared Key: h1p8A24nG5
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Aggressive

(or)

Certificates

Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): rsa-g2-3des-sha
Mode (Initiator): Aggressive
Preferred Certificate (optional):
Peer CA: Verisign
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: Paris_Tokyo
Security Level: Compatible
Remote Gateway:
  Predefined: (select), To_Tokyo

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface: (select), tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 10.2.2.0/24
Remote IP / Netmask: 10.1.1.0/24
Service: ANY

4. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
  Interface: ethernet3
  Gateway IP Address: (select), 2.2.2.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.1.0/24
Gateway: (select)
Interface: Tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.1.0/24
Gateway: (select)
Interface: Null
Gateway IP Address: 0.0.0.0
Metric: 10

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Trust_LAN
Destination Address:
Address Book Entry: (select), Tokyo_Office
Service: Any
Action: Permit
Position at Top: (select)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Tokyo_Office
Destination Address:
Address Book Entry: (select), Trust_LAN
Service: Any
Action: Permit
Position at Top: (select)

CLI (Tokyo)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 dhcp client
set interface ethernet3 dhcp client settings server 1.1.1.5
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

2. **Addresses**

```
set address trust Trust_LAN 10.1.1.0/24
set address untrust Paris_Office 10.2.2.0/24
```

3. **VPN**

Preshared Key

```
set ike gateway To_Paris address 2.2.2.2 aggressive local-id pmason@abc.com
 outgoing-interface ethernet3 preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn Tokyo_Paris gateway To_Paris tunnel sec-level compatible
set vpn Tokyo_Paris bind interface tunnel.1
set vpn Tokyo_Paris proxy-id local-ip 10.1.1.0/24 remote-ip 10.2.2.0/24 any
```

(or)

Certificates

```
set ike gateway To_Paris address 2.2.2.2 aggressive local-id pmason@abc.com
 outgoing-interface ethernet3 proposal rsa-g2-3des-sha
set ike gateway To_Paris cert peer-ca 1
set ike gateway To_Paris cert peer-cert-type x509-sig
set vpn Tokyo_Paris gateway To_Paris tunnel sec-level compatible
set vpn Tokyo_Paris bind interface tunnel.1
set vpn Tokyo_Paris proxy-id local-ip 10.1.1.0/24 remote-ip 10.2.2.0/24 any
```

NOTE: The U-FQDN "pmason@abc.com" must appear in the SubjectAltName field in the certificate.

The number 1 is the CA ID number. To discover the CA's ID number, use the following command: **get ike ca**.

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.1
set vrouter trust-vr route 10.2.2.0/24 interface null metric 10
```

NOTE: The ISP provides the gateway IP address dynamically through DHCP, so you cannot specify it here.

5. **Policies**

```
set policy top from trust to untrust Trust_LAN Paris_Office any permit
set policy top from untrust to trust Paris_Office Trust_LAN any permit
save
```

CLI (Paris)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
```

```
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

2. **Addresses**

```
set address trust Trust_LAN 10.2.2.0/24
set address untrust Tokyo_Office 10.1.1.0/24
```

3. **VPN**

Preshared Key

```
set ike gateway To_Tokyo dynamic pmason@abc.com aggressive
outgoing-interface ethernet3 preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn Paris_Tokyo gateway To_Tokyo tunnel sec-level compatible
set vpn Paris_Tokyo bind interface tunnel.1
set vpn Paris_Tokyo proxy-id local-ip 10.2.2.0/24 remote-ip 10.1.1.0/24 any
```

(or)

Certificates

```
set ike gateway To_Tokyo dynamic pmason@abc.com aggressive
outgoing-interface ethernet3 proposal rsa-g2-3des-sha
set ike gateway To_Tokyo cert peer-ca 1
set ike gateway To_Tokyo cert peer-cert-type x509-sig
set vpn Paris_Tokyo gateway To_Tokyo tunnel sec-level compatible
set vpn Paris_Tokyo bind interface tunnel.1
set vpn Paris_Tokyo proxy-id local-ip 10.2.2.0/24 remote-ip 10.1.1.0/24 any
```

NOTE: The number 1 is the CA ID number. To discover the CA's ID number, use the following command: **get ike ca**.

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
set vrouter trust-vr route 10.1.1.0/24 interface tunnel.1
set vrouter trust-vr route 10.1.1.0/24 interface null metric 10
```

5. **Policies**

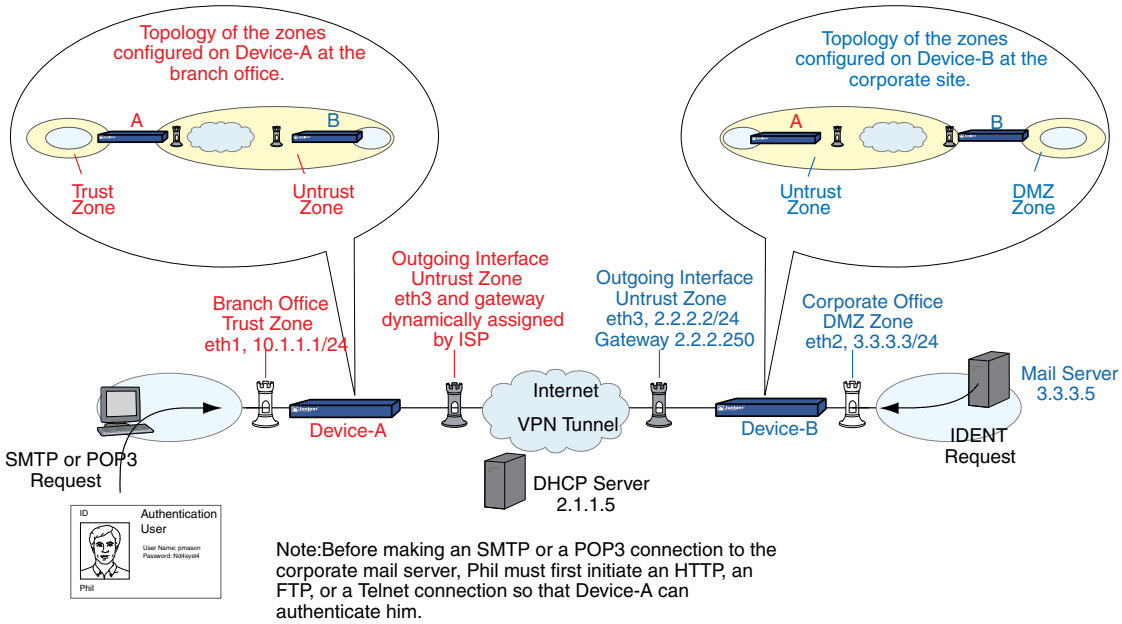
```
set policy top from trust to untrust Trust_LAN Tokyo_Office any permit
set policy top from untrust to trust Tokyo_Office Trust_LAN any permit
save
```

Policy-Based Site-to-Site VPN, Dynamic Peer

In this example, a VPN tunnel securely connects the users in the Trust zone behind Device A to the mail server in the corporate DMZ zone, protected by Device B. The Untrust zone interface for Device B has a static IP address. The ISP serving Device A assigns the IP address for its Untrust zone interface dynamically through DHCP.

Because only Device B has a fixed address for its Untrust zone, VPN traffic must originate from hosts behind Device A. After Device A has established the tunnel, traffic through the tunnel can originate from either end. All zones are in the trust-vr routing domain.

Figure 226: Policy-Based Site-to-Site VPN, Dynamic Peer



Note:Before making an SMTP or a POP3 connection to the corporate mail server, Phil must first initiate an HTTP, an FTP, or a Telnet connection so that Device-A can authenticate him.

In this example, the local auth user Phil (login name: pmason; password: Nd4syst4) wants to get his email from the mail server at the corporate site. When he attempts to do so, he is authenticated twice: first, Device A authenticates him locally before allowing traffic from him through the tunnel; second, the mail server program authenticates him, sending the IDENT request through the tunnel.

NOTE: Because Phil is an authentication user, before he can make an SMTP of POP3 request, he must first initiate an HTTP, FTP, or Telnet connection so that Device A can respond with a firewall user/login prompt to authenticate him. After Device A authenticates him, he has permission to contact the corporate mail server through the VPN tunnel.

The mail server can send the IDENT request through the tunnel only if the Device A and B administrators add a custom service for it (TCP, port 113) and set up policies allowing that traffic through the tunnel to the 10.10.10.0/24 subnet.

The preshared key is h1p8A24nG5. It is assumed that both participants already have RSA certificates from the certificate authority (CA) Verisign and that the email address *pmason@abc.com* appears in the local certificate on Device A. (For information about obtaining and loading certificates, see “Certificates and CRLs” on page 746.) For the Phase 1 and Phase 2 security levels, you specify one Phase 1 proposal—either

pre-g2-3des-sha for the preshared key method or rsa-g2-3des-sha for certificates—and select the predefined "Compatible" set of proposals for Phase 2.

WebUI (Device A)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Obtain IP using DHCP: (select)

NOTE: You cannot specify the IP address of the DHCP server through the WebUI; however, you can do so through the CLI.

2. **User**

   Objects > Users > Local > New: Enter the following, then click **OK**:

   User Name: pmason
   Status: Enable
   Authentication User: (select)
   User Password: Nd4syst4
   Confirm Password: Nd4syst4

3. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Trusted_network
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Mail_Server
   IP Address/Domain Name:
   IP/Netmask: (select), 3.3.3.5/32
   Zone: Untrust

4. **Services**

   Policy > Policy Elements > Services > Custom > New: Enter the following, then click **OK**:

   Service Name: Ident
   Service Timeout:
   Use protocol default: (select)
   Transport Protocol: TCP (select)
   Source Port: Low 0, High 65535
   Destination Port: Low 113, High 113

   Policy > Policy Elements > Services > Group > New: Enter the following, move the following services, then click **OK**:

   Group Name: Remote_Mail
   Group Members << Available Members:
   HTTP
   FTP
   Telnet
   Ident
   MAIL
   POP3

5. **VPN**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: To_Mail
   Security Level: Custom
   Remote Gateway Type:
   Static IP Address: (select), IP Address/Hostname: 2.2.2.2

   Preshared Key

   Preshared Key: h1p8A24nG5
   Local ID: pmason@abc.com
   Outgoing Interface: ethernet3

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

   Security Level: Custom
   Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
   Mode (Initiator): Aggressive

   (or)

   Certificates

   Local ID: pmason@abc.com
   Outgoing Interface: ethernet3

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): rsa-g2-3des-sha
Mode (Initiator): Aggressive
Preferred Certificate (optional):
Peer CA: Verisign
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

Name: branch_corp
Security Level: Compatible
Remote Gateway Tunnel: To_Mail

6. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 0.0.0.0

NOTE: The ISP provides the gateway IP address dynamically through DHCP.

7. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Trusted_network
Destination Address:
Address Book Entry: (select), Mail_Server
Service: Remote_Mail
Action: Tunnel
VPN Tunnel: branch_corp
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

Authentication: (select)
Auth Server: Local
User: (select), Local Auth User - pmason

WebUI (Device B)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: DMZ
Static IP: (select this option when present)
IP Address/Netmask: 3.3.3.3/24

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 2.2.2.2/24

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Mail Server
IP Address/Domain Name:
IP/Netmask: (select), 3.3.3.5/32
Zone: DMZ

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK:**

Address Name: branch office
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Untrust

3. **Services**

Policy > Policy Elements > Services > Custom > New: Enter the following, then click **OK**:

Service Name: Ident
Service Timeout:
Use protocol default: (select)
Transport Protocol: TCP (select)
Source Port: Low 0, High 65535
Destination Port: Low 113, High 113

Policy > Policy Elements > Services > Groups > New: Enter the following, move the following services, then click **OK**:

Group Name: Remote_Mail
Group Members << Available Members:
Ident
MAIL
POP3

4. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_branch
Security Level: Custom

Remote Gateway Type:
Dynamic IP Address: (select), Peer ID: pmason@abc.com

Preshared Key

Preshared Key: h1p8A24nG5
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Aggressive

(or)

Certificates

Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): rsa-g2-3des-sha
Mode (Initiator): Aggressive
Preferred Certificate (optional):
Peer CA: Verisign
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: corp_branch
Security Level: Compatible
Remote Gateway:
Predefined: (select), To_branch

5. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 2.2.2.250

6. **Policies**

Policies > (From: DMZ, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Mail_Server
Destination Address:
Address Book Entry: (select), branch_office

Service: Remote_Mail
Action: Tunnel
VPN Tunnel: corp_branch
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

CLI (Device A)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 dhcp client
set interface ethernet3 dhcp client settings server 1.1.1.5
```

2. **User**

```
set user pmason password Nd4syst4
```

3. **Addresses**

```
set address trust “trusted network” 10.1.1.0/24
set address untrust “ mail server” 3.3.3.5/32
```

4. **Services**

```
set service ident protocol tcp src-port 0-65535 dst-port 113-113
set group service remote_mail
set group service remote_mail add http
set group service remote_mail add ftp
set group service remote_mail add telnet
set group service remote_mail add ident
set group service remote_mail add mail
set group service remote_mail add pop3
```

5. **VPN**

Preshared Key

```
set ike gateway to_mail address 2.2.2.2 aggressive local-id pmason@abc.com
 outgoing-interface ethernet3 preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn branch_corp gateway to_mail sec-level compatible
```

(or)

Certificates

```
set ike gateway to_mail address 2.2.2.2 aggressive local-id pmason@abc.com

   outgoing-interface ethernet3 proposal rsa-g2-3des-sha
set ike gateway to_mail cert peer-ca 1
set ike gateway to_mail cert peer-cert-type x509-sig
set vpn branch_corp gateway to_mail sec-level compatible
```

NOTE: The U-FQDN "pmason@abc.com" must appear in the SubjectAltName field in the certificate.

The number 1 is the CA ID number. To discover the CA's ID number, use the following command: **get ike ca**.

6. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3
```

NOTE: The ISP provides the gateway IP address dynamically through DHCP.

7. **Policies**

```
set policy top from trust to untrust "trusted network" "mail server" remote_mail
 tunnel vpn branch_corp auth server Local user pmason
set policy top from untrust to trust "mail server" "trusted network" remote_mail
 tunnel vpn branch_corp
save
```

CLI (Device B)

1. **Interfaces**

```
set interface ethernet2 zone dmz
set interface ethernet2 ip 3.3.3.3/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
```

2. **Addresses**

```
set address dmz "mail server" 3.3.3.5/32
set address untrust " branch office" 10.1.1.0/24
```

3. **Services**

```
set service ident protocol tcp src-port 0-65535 dst-port 113-113
set group service remote_mail
set group service remote_mail add ident
set group service remote_mail add mail
set group service remote_mail add pop3
```

4. **VPN**

Preshared Key

```
set ike gateway to_branch dynamic pmason@abc.com aggressive
outgoing-interface ethernet3 preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn corp_branch gateway to_branch tunnel sec-level compatible
```

(or)

Certificates

```
set ike gateway to_branch dynamic pmason@abc.com aggressive
outgoing-interface ethernet3 proposal rsa-g2-3des-sha
set ike gateway to_branch cert peer-ca 1
set ike gateway to_branch cert peer-cert-type x509-sig
set vpn corp_branch gateway to_branch sec-level compatible
```

NOTE: The number 1 is the CA ID number. To discover the CA's ID number, use the following command: **get ike ca**.

5. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
```

6. **Policies**

```
set policy top from dmz to untrust "mail server" "branch office" remote_mail
tunnel vpn corp_branch
set policy top from untrust to dmz "branch office" "mail server" remote_mail
tunnel vpn corp_branch
save
```

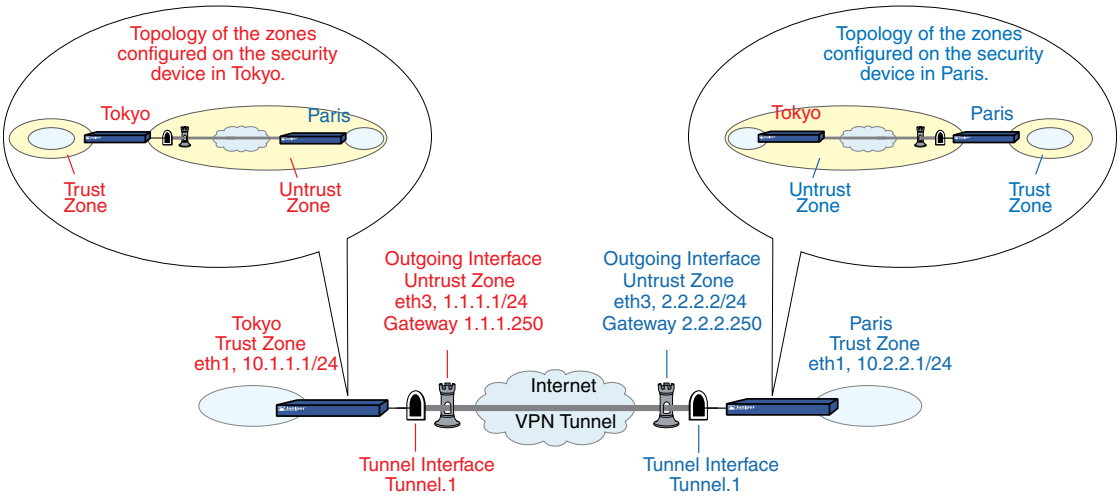
Route-Based Site-to-Site VPN, Manual Key

In this example, a Manual Key tunnel provides a secure communication channel between offices in Tokyo and Paris. The Trust zones at each site are in NAT mode. The addresses are as follows:

- Tokyo:
  - Trust zone interface (ethernet1): 10.1.1.1/24
  - Untrust zone interface (ethernet3): 1.1.1.1/24
- Paris:
  - Trust zone interface (ethernet1): 10.2.2.1/24
  - Untrust zone interface (ethernet3): 2.2.2.2/24

The Trust and Untrust security zones are all in the trust-vr routing domain. The Untrust zone interface (ethernet3) serves as the outgoing interface for the VPN tunnel.

Figure 227: Route-Based Site-to-Site VPN, Manual Key



To set up the tunnel, perform the following steps on the security devices at both ends of the tunnel:

1. Assign IP addresses to the physical interfaces bound to the security zones and to the tunnel interface.
2. Configure the VPN tunnel, designate its outgoing interface in the Untrust zone, and bind it to the tunnel interface.
3. Enter the IP addresses for the local and remote endpoints in the address books for the Trust and Untrust zones.
4. Enter a default route to the external router in the trust-vr, a route to the destination through the tunnel interface, and a null route to the destination. You assign a higher metric (farther from zero) to the null route so that it becomes the next-choice route to the destination. Then, if the state of the tunnel interface changes to "down" and the route referencing that interface becomes inactive, the security device uses the null route, which essentially drops any traffic sent to it, rather than the default route, which forwards unencrypted traffic.
5. Set up policies for VPN traffic to pass between each site.

WebUI (Tokyo)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet3 (trust-vr)

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Trust_LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Paris_Office
IP Address/Domain Name:
IP/Netmask: (select), 10.2.2.0/24
Zone: Untrust

3. **VPN**

VPNs > Manual Key > New: Enter the following, then click **OK**:

VPN Tunnel Name: Tokyo_Paris
Gateway IP: 2.2.2.2
Security Index: 3020 (Local), 3030 (Remote)
Outgoing Interface: ethernet3
ESP-CBC: (select)
Encryption Algorithm: 3DES-CBC
Generate Key by Password: asdlk24234
Authentication Algorithm: SHA-1
Generate Key by Password: PNas134a

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Manual Key tunnel configuration page:

Bind to: Tunnel Interface, tunnel.1

4. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
Interface: Null
Gateway IP Address: 0.0.0.0
Metric: 10

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: To Paris
Source Address:
Address Book Entry: (select), Trust_LAN
Destination Address:
Address Book Entry: (select), Paris_Office
Service: ANY
Action: Permit
Position at Top: (select)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Name: From Paris
Source Address:
Address Book Entry: (select), Paris_Office
Destination Address:
Address Book Entry: (select), Trust_LAN
Service: ANY
Action: Permit
Position at Top: (select)

WebUI (Paris)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.2.2.1/24

Select the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 2.2.2.2/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet3 (trust-vr)

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Trust_LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.2.2.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Tokyo_Office
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Untrust

3. **VPN**

VPNs > Manual Key > New: Enter the following, then click **OK**:

VPN Tunnel Name: Paris_Tokyo
Gateway IP: 1.1.1.1
Security Index: 3030 (Local), 3020 (Remote)
Outgoing Interface: ethernet3
ESP-CBC: (select)
Encryption Algorithm: 3DES-CBC
Generate Key by Password: asdlk24234
Authentication Algorithm: SHA-1
Generate Key by Password: PNas134a

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Manual Key tunnel configuration page:

Bind to: Tunnel Interface, tunnel.1

4. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 2.2.2.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.1.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.1.0/24
Gateway: (select)
Interface: Null
Gateway IP Address: 0.0.0.0
Metric: 10

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: To_Tokyo
Source Address:
Address Book Entry: (select), Trust_LAN
Destination Address:
Address Book Entry: (select), Tokyo_Office
Service: ANY
Action: Permit
Position at Top: (select)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Name: From_Tokyo
Source Address:
Address Book Entry: (select), Tokyo_Office
Destination Address:
Address Book Entry: (select), Trust_LAN
Service: ANY
Action: Permit
Position at Top: (select)

CLI (Tokyo)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

2. **Addresses**

```
set address trust Trust_LAN 10.1.1.0/24
set address untrust Paris_Office 10.2.2.0/24
```

3. **VPN**

```
set vpn Tokyo_Paris manual 3020 3030 gateway 2.2.2.2 outgoing-interface
ethernet3 esp 3des password asdlk24234 auth sha-1 password PNas134a
set vpn Tokyo_Paris bind interface tunnel.1
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.1
set vrouter trust-vr route 10.2.2.0/24 interface null metric 10
```

5. **Policies**

```
set policy top name “To Paris” from trust to untrust Trust_LAN Paris_Office any
 permit
set policy top name “From Paris” from untrust to trust Paris_Office Trust_LAN
any permit
save
```

CLI (Paris)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

2. **Addresses**

```
set address trust Trust_LAN 10.2.2.0/24
set address untrust Tokyo_Office 10.1.1.0/24
```

3. **VPN**

```
set vpn Paris_Tokyo manual 3030 3020 gateway 1.1.1.1 outgoing-interface
ethernet3 esp 3des password asdlk24234 auth sha-1 password PNas134a
set vpn Paris_Tokyo bind interface tunnel.1
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
set vrouter trust-vr route 10.1.1.0/24 interface tunnel.1
set vrouter trust-vr route 10.1.1.0/24 interface null metric 10
```

5. **Policies**

```
set policy top name “To Tokyo” from trust to untrust Trust_LAN Tokyo_Office any
 permit
set policy top name “ From Tokyo” from untrust to trust Tokyo_Office Trust_LAN
 any permit
save
```

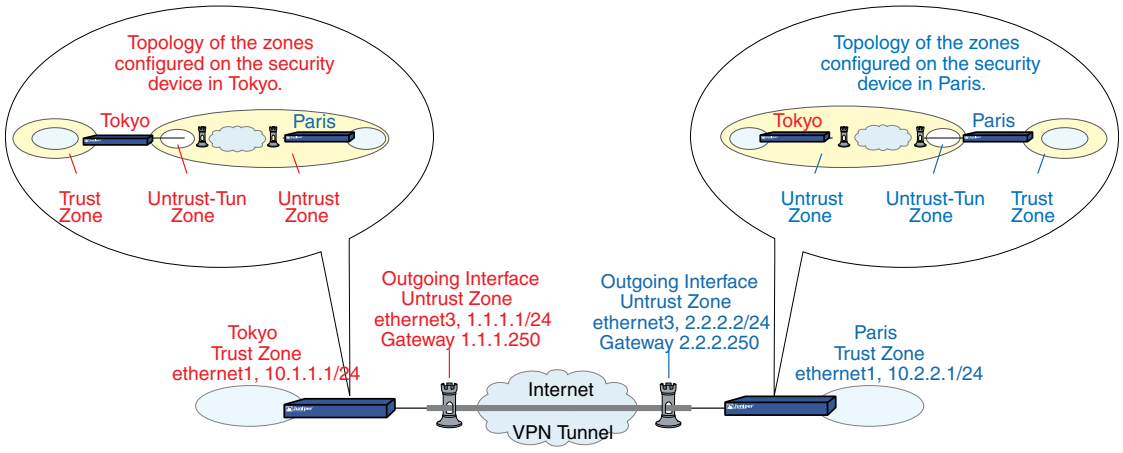
Policy-Based Site-to-Site VPN, Manual Key

In this example, a Manual Key tunnel provides a secure communication channel between offices in Tokyo and Paris, using ESP with 3DES encryption and SHA-1 authentication. The Trust zones at each site are in NAT mode. The addresses are as follows:

- Tokyo:
  - Trust interface (ethernet1): 10.1.1.1/24
  - Untrust interface (ethernet3): 1.1.1.1/24
- Paris:
  - Trust interface (ethernet1): 10.2.2.1/24
  - Untrust interface (ethernet3): 2.2.2.2/24

The Trust and Untrust security zones and the Untrust-Tun tunnel zone are in the trust-vr routing domain. The Untrust zone interface (ethernet3) serves as the outgoing interface for the VPN tunnel.

Figure 228: Policy-Based Site-to-Site VPN, Manual Key



To set up the tunnel, perform the following five steps on the security devices at both ends of the tunnel:

1. Assign IP addresses to the physical interfaces bound to the security zones.
2. Configure the VPN tunnel, and designate its outgoing interface in the Untrust zone.
3. Enter the IP addresses for the local and remote endpoints in the Trust and Untrust address books.
4. Enter a default route to the external router.
5. Set up policies for VPN traffic to pass bidirectionally through the tunnel.

WebUI (Tokyo)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Trust_LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Paris_Office
IP Address/Domain Name:
IP/Netmask: (select), 10.2.2.0/24
Zone: Untrust

3. **VPN**

VPNs > Manual Key > New: Enter the following, then click **OK**:

VPN Tunnel Name: Tokyo_Paris
Gateway IP: 2.2.2.2
Security Index: 3020 (Local), 3030 (Remote)
Outgoing Interface: ethernet3
ESP-CBC: (select)
Encryption Algorithm: 3DES-CBC
Generate Key by Password: asdlk24234
Authentication Algorithm: SHA-1
Generate Key by Password: PNas134a

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Manual Key tunnel configuration page:

Bind to: Tunnel Zone, Untrust-Tun

4. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: To/From Paris
Source Address:
Address Book Entry: (select), Trust_LAN
Destination Address:
Address Book Entry: (select), Paris_Office
Service: ANY
Action: Tunnel

Tunnel VPN: Tokyo_Paris
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

WebUI (Paris)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.2.2.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 2.2.2.2/24

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Trust_LAN
   IP Address/Domain Name:
   IP/Netmask: (select), 10.2.2.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Tokyo_Office
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.0/24
   Zone: Untrust

3. **VPN**

   VPNs > Manual Key > New: Enter the following, then click **OK**:

   VPN Tunnel Name: Paris_Tokyo
   Gateway IP: 1.1.1.1
   Security Index (HEX Number): 3030 (Local), 3020 (Remote)
   Outgoing Interface: ethernet3
   ESP-CBC: (select)
   Encryption Algorithm: 3DES-CBC
   Generate Key by Password: asdlk24234

Authentication Algorithm: SHA-1
Generate Key by Password: PNas134a

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Manual Key tunnel configuration page:

Bind to: Tunnel Zone, Untrust-Tun

4. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
  Interface: ethernet3
  Gateway IP Address: 2.2.2.250

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: To/From Tokyo
Source Address:
  Address Book Entry: (select), Trust_LAN
Destination Address:
  Address Book Entry: (select), Tokyo_Office
Service: ANY
Action: Tunnel
Tunnel VPN: Paris_Tokyo
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

CLI (Tokyo)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address trust Trust_LAN 10.1.1.0/24
set address untrust paris_office 10.2.2.0/24
```

3. **VPN**

```
set vpn tokyo_paris manual 3020 3030 gateway 2.2.2.2 outgoing-interface
ethernet3 esp 3des password asdlk24234 auth sha-1 password PNas134a
set vpn tokyo_paris bind zone untrust-tun
```

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

5. **Policies**

```
set policy top name “To/From Paris” from trust to untrust Trust_LAN paris_office
 any tunnel vpn tokyo_paris
set policy top name “ To/From Paris” from untrust to trust paris_office Trust_LAN
 any tunnel vpn tokyo_paris
save
```

CLI (Paris)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
```

2. **Addresses**

```
set address trust Trust_LAN 10.2.2.0/24
set address untrust tokyo_office 10.1.1.0/24
```

3. **VPN**

```
set vpn paris_tokyo manual 3030 3020 gateway 1.1.1.1 outgoing-interface
ethernet3 esp 3des password asdlk24234 auth sha-1 password PNas134a
set vpn paris_tokyo bind zone untrust-tun
```

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
```

5. **Policies**

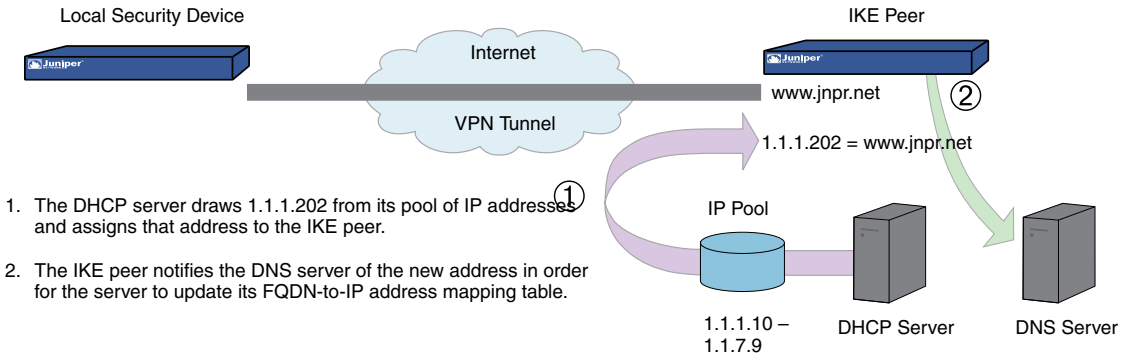
```
set policy top name “To/From Tokyo” from trust to untrust Trust_LAN tokyo_office
 any tunnel vpn paris_tokyo
set policy top name “ To/From Tokyo” from untrust to trust tokyo_office Trust_LAN
 any tunnel vpn paris_tokyo
save
```

Dynamic IKE Gateways Using FQDN

For an IKE peer that obtains its IP address dynamically, you can specify its fully qualified domain name (FQDN) in the local configuration for the remote gateway. For example, an Internet service provider (ISP) might assign IP addresses through DHCP to its customers. The ISP draws addresses from a large pool of addresses and assigns them when its customers come online. Although the IKE peer has an unchanging FQDN, it has an unpredictably changing IP address. The IKE peer has three methods available for maintaining a Domain Name System (DNS) mapping of its FQDN to its dynamically assigned IP address (a process known as dynamic DNS).

- If the remote IKE peer is a security device, the admin can manually notify the DNS server to update its FQDN-to-IP address mapping each time the security device receives a new IP address from its ISP.
- If the remote IKE peer is another kind of VPN termination device that has dynamic DNS software running on it, that software can automatically notify the DNS server of its address changes so the server can update its FQDN-to-IP address mapping table.
- If the remote IKE peer is a security device or any other kind of VPN termination device, a host behind it can run an FQDN-to-IP address automatic update program that alerts the DNS server of address changes.

Figure 229: IKE Peer with a Dynamic IP Address

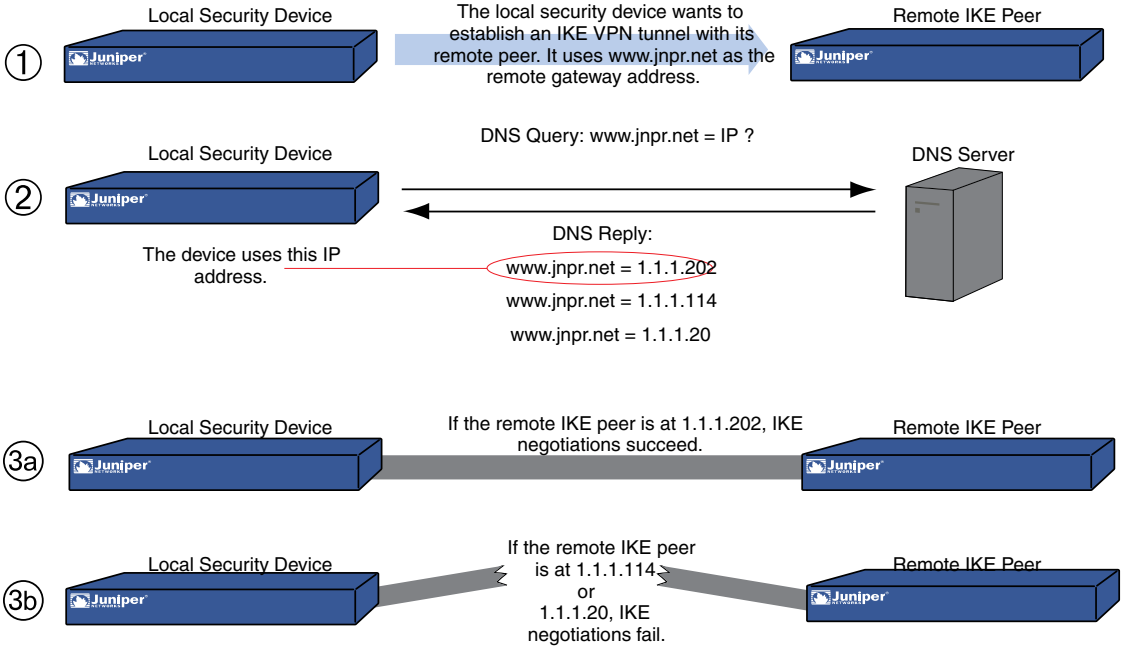


Without needing to know the current IP address of a remote IKE peer, you can now configure an AutoKey IKE VPN tunnel to that peer using its FQDN instead of an IP address.

Aliases

You can also use an alias for the FQDN of the remote IKE peer if the DNS server that the local security device queries returns only one IP address. If the DNS server returns several IP addresses, the local device uses the first one it receives. Because there is no guarantee for the order of the addresses in the response from the DNS server, the local security device might use the wrong IP address, and IKE negotiations might fail.

Figure 230: Multiple DNS Replies Leading to IKE Negotiation Success or Failure

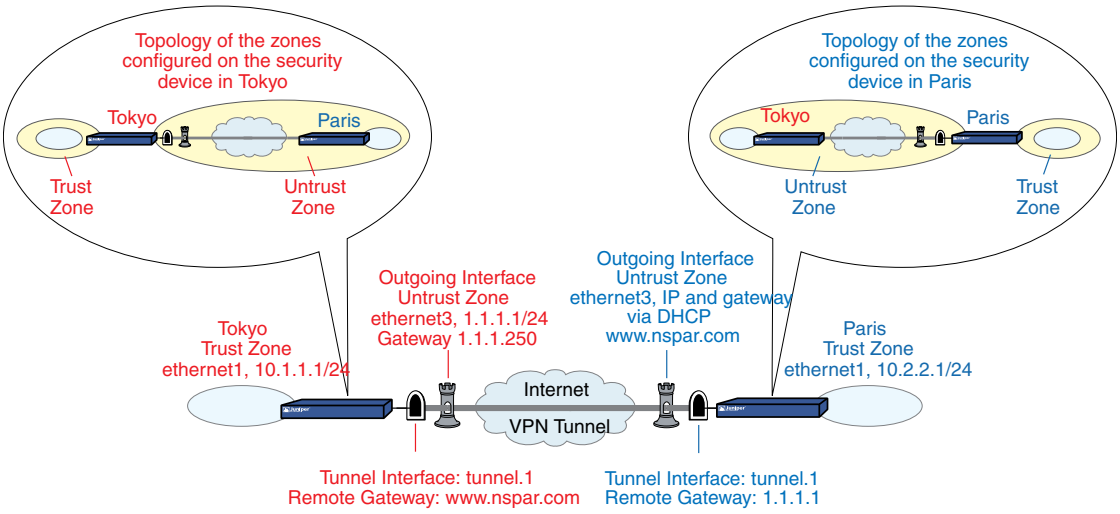


Setting AutoKey IKE Peer with FQDN

In this example, an AutoKey IKE VPN tunnel using either a preshared secret or a pair of certificates (one at each end of the tunnel) provides a secure connection between two offices in Tokyo and Paris. The Paris office has a dynamically assigned IP address, so the Tokyo office uses the remote peer's FQDN (www.nspar.com) as the address of the remote gateway in its VPN tunnel configuration.

The configuration shown in Figure 231 on page 855 is for a route-based VPN tunnel. For the Phase 1 and Phase 2 security levels, you specify one Phase 1 proposal—either pre-g2-3des-sha for the preshared key method or rsa-g2-3des-sha for certificates—and select the predefined "Compatible" set of proposals for Phase 2. All zones are in the trust-vr.

Figure 231: AutoKey IKE Peer with FQDN



Setting up a route-based AutoKey IKE tunnel using either a preshared secret or certificates involves the following steps:

1. Assign IP addresses to the physical interfaces bound to the security zones and to the tunnel interface.
2. Define the remote gateway and key exchange mode, and specify either a preshared secret or a certificate.
3. Configure the VPN tunnel, designate its outgoing interface in the Untrust zone, bind it to the tunnel interface, and configure its proxy-ID.
4. Enter the IP addresses for the local and remote endpoints in the Trust and Untrust address books.
5. Enter a default route to the external router in the trust-vr, a route to the destination through the tunnel interface, and a null route to the destination. You assign a higher metric (farther from zero) to the null route so that it becomes the next-choice route to the destination. Then, if the state of the tunnel interface changes to “down” and the route referencing that interface becomes inactive, the security device uses the null route, which essentially drops any traffic sent to it, rather than the default route, which forwards unencrypted traffic.
6. Set up policies for traffic to pass between each site.

In the following examples, the preshared key is h1p8A24nG5. It is assumed that both participants already have RSA certificates and are using Entrust as the certificate authority (CA). (For information about obtaining and loading certificates, see “Public Key Cryptography” on page 741.)

WebUI (Tokyo)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24

Select the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet3 (trust-vr)

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Trust_LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Paris_Office
IP Address/Domain Name:
IP/Netmask: (select), 10.2.2.0/24
Zone: Untrust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_Paris
Security Level: Custom
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: www.nspar.com

Preshared Key

Preshared Key: h1p8A24nG5
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (for Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)

(or)

Certificates

Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (for Custom Security Level): rsa-g2-3des-sha
Preferred certificate (optional)
Peer CA: Entrust
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: Tokyo_Paris
Security Level: Compatible
Remote Gateway:
  Predefined: (select), To_Paris

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Security Level: Compatible
Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
  Local IP / Netmask: 10.1.1.0/24
  Remote IP / Netmask: 10.2.2.0/24
  Service: ANY

4. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
  Interface: ethernet3
  Gateway IP Address: 0.0.0.0

NOTE: The ISP provides the gateway IP address dynamically through DHCP.

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK:**

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
Interface: Null
Gateway IP Address: 0.0.0.0
Metric: 10

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: To Paris
Source Address: Trust_LAN
Destination Address: Paris_Office
Service: ANY
Action: Permit
Position at Top: (select)

Policies > Policy (From: Untrust, To: Trust) > New Policy: Enter the following, then click **OK**:

Name: From Paris
Source Address: Paris_Office
Destination Address: Trust_LAN
Service: ANY
Action: Permit
Position at Top: (select)

WebUI (Paris)

1. **Host Name and Domain Name**

Network > DNS: Enter the following, then click **Apply**:

Host Name: www
Domain Name: nspar.com

2. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.2.2.1/24

Select the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Obtain IP using DHCP: (select)

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet3 (trust-vr)

3. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Trust_LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.2.2.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Tokyo_Office
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Untrust

4. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_Tokyo
Security Level: Custom
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 1.1.1.1

Preshared Key

Preshared Key: h1p8A24nG5
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)

(or)

Certificates

Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (for Custom Security Level): rsa-g2-3des-sha
Preferred certificate (optional)
Peer CA: Entrust
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

Name: Paris_Tokyo
Security Level: Custom
Remote Gateway:
Predefined: (select), To_Tokyo

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Security Level: Compatible
Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 10.2.2.0/24
Remote IP / Netmask: 10.1.1.0/24
Service: ANY

5. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 2.2.2.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.1.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.1.0/24
Gateway: (select)
Interface: Null
Gateway IP Address: 0.0.0.0
Metric: 10

6. **Policies**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Name: To Tokyo
   Source Address: Trust_LAN
   Destination Address: Tokyo_Office
   Service: ANY
   Action: Permit
   Position at Top: (select)

   Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

   Name: From Tokyo
   Source Address: Tokyo_Office
   Destination Address: Trust_LAN
   Service: ANY
   Action: Permit
   Position at Top: (select)

CLI (Tokyo)

1. **Interfaces**

   ```
   set interface ethernet1 zone trust
   set interface ethernet1 ip 10.1.1.1/24
   set interface ethernet1 nat
   set interface ethernet3 zone untrust
   set interface ethernet3 ip 1.1.1.1/24
   set interface tunnel.1 zone untrust
   set interface tunnel.1 ip unnumbered interface ethernet3
   ```

2. **Addresses**

   ```
   set address trust Trust_LAN 10.1.1.0/24
   set address untrust paris_office 10.2.2.0/24
   ```

3. **VPN**

   Preshared Key

   ```
   set ike gateway to_paris address www.nspar.com main outgoing-interface
   ethernet3 preshare h1p8A24nG5 proposal pre-g2-3des-sha
   set vpn tokyo_paris gateway to_paris sec-level compatible
   set vpn tokyo_paris bind interface tunnel.1
   set vpn tokyo_paris proxy-id local-ip 10.1.1.0/24 remote-ip 10.2.2.0/24 any
   ```

   (or)

   Certificate

   Certificate
   ```
   set ike gateway to_paris address www.nspar.com main outgoing-interface
   ethernet3 proposal rsa-g2-3des-sha
   set ike gateway to_paris cert peer-ca 1
   ```

```
set ike gateway to_paris cert peer-cert-type x509-sig
set vpn tokyo_paris gateway to_paris sec-level compatible
set vpn tokyo_paris bind interface tunnel.1
set vpn tokyo_paris proxy-id local-ip 10.1.1.0/24 remote-ip 10.2.2.0/24 any
```

NOTE: The number 1 is the CA ID number. To discover the CA's ID number, use the following command: **get ike ca**.

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.1
set vrouter trust-vr route 10.2.2.0/24 interface null metric 10
```

5. **Policies**

```
set policy top name "To Paris" from trust to untrust Trust_LAN paris_office any
 permit
set policy top name "From Paris" from untrust to trust paris_office Trust_LAN
any permit
save
```

CLI (Paris)

1. **Host Name and Domain Name**

```
set hostname www
set domain nspar.com
```

2. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip dhcp-client enable
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

3. **Addresses**

```
set address trust Trust_LAN 10.2.2.0/24
set address untrust tokyo_office 10.1.1.0/24
```

4. **VPN**

Preshared Key

```
set ike gateway to_tokyo address 1.1.1.1 main outgoing-interface ethernet3
preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn paris_tokyo gateway to_tokyo sec-level compatible
set vpn paris_tokyo bind interface tunnel.1
set vpn paris_tokyo proxy-id local-ip 10.2.2.0/24 remote-ip 10.1.1.0/24 any
```

(or)

Certificate

```
set ike gateway to_tokyo address 1.1.1.1 main outgoing-interface ethernet3
proposal rsa-g2-3des-sha
set ike gateway to_tokyo cert peer-ca 1
set ike gateway to_tokyo cert peer-cert-type x509-sig
set vpn paris_tokyo gateway to_tokyo sec-level compatible
set vpn paris_tokyo bind interface tunnel.1
set vpn paris_tokyo proxy-id local-ip 10.2.2.0/24 remote-ip 10.1.1.0/24 any
```

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
set vrouter trust-vr route 10.1.1.0/24 interface tunnel.1
set vrouter trust-vr route 10.1.1.0/24 interface null metric 10
```

6. **Policies**

```
set policy top name "To Tokyo" from trust to untrust Trust_LAN tokyo_office any
 permit
set policy top name "From Tokyo" from untrust to trust tokyo_office Trust_LAN
any permit
save
```

VPN Sites with Overlapping Addresses

Because the range of private IP addresses is relatively small, there is a good chance that the addresses of protected networks of two VPN peers overlap. For bidirectional VPN traffic between two end entities with overlapping addresses, the security devices at both ends of the tunnel must apply Source Network Address Translation (NAT-src) and Destination Network Address Translation (NAT-dst) to the VPN traffic passing between them.

NOTE: An overlapping address space is when the IP address range in two networks are partially or completely the same.

For NAT-src, the interfaces at both ends of the tunnel must have IP addresses in mutually unique subnets, with a dynamic IP (DIP) pool in each of those subnets. The policies regulating outbound VPN traffic can then apply NAT-src using DIP pool addresses to translate original source addresses to those in a neutral address space.

NOTE: The range of addresses in a DIP pool must be in the same subnet as the tunnel interface, but the pool must not include the interface IP address or any MIP or VIP addresses that might also be in that subnet. For security zone interfaces, you can also define an extended IP address and an accompanying DIP pool in a different subnet from that of the interface IP address. For more information, see "Using DIP in a Different Subnet" on page 181.

To provide NAT-dst on inbound VPN traffic, there are two options:

- Policy-based NAT-dst: A policy can apply NAT-dst to translate inbound VPN traffic to an address that is either in the same subnet as the tunnel interface—but not in the same range as the local DIP pool used for outbound VPN traffic—or to an address in another subnet to which the security device has an entry in its route table. (For information about routing considerations when configuring NAT-dst, see "Routing for NAT-Dst" on page 1503.)
- Mapped IP (MIP): A policy can reference a MIP as the destination address. The MIP uses an address in the same subnet as the tunnel interface—but not in the same range as the local DIP pool used for outbound VPN traffic. (For information about MIPs, see "Mapped IP Addresses" on page 1535.)

VPN traffic between sites with overlapping addresses requires address translation in both directions. Because the source address on outbound traffic cannot be the same as the destination address on inbound traffic—the NAT-dst address or MIP cannot be in the DIP pool—the addresses referenced in the inbound and outbound policies cannot be symmetrical.

When you want the security device to perform source and destination address translation on bidirectional VPN traffic through the same tunnel, you have two choices:

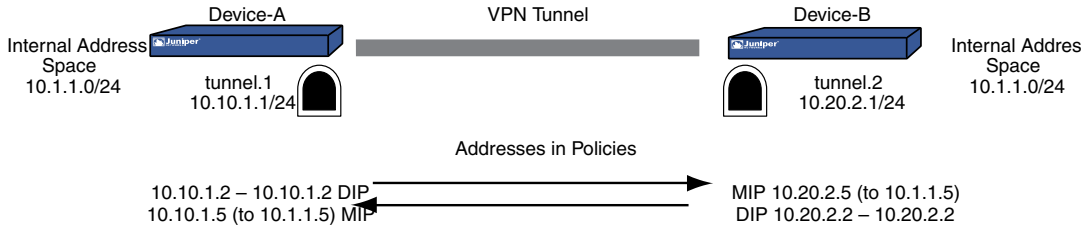
- You can define a proxy ID for a policy-based VPN configuration. When you specifically reference a VPN tunnel in a policy, the security device derives a proxy ID from the components in the policy that references that tunnel. The security device derives the proxy ID when you first create the policy, and each time the device reboots thereafter. However, if you manually define a proxy ID for a VPN tunnel that is referenced in a policy, the security device applies the user-defined proxy ID, not the proxy ID derived from the policy.

NOTE: A proxy ID is a kind of agreement between IKE peers to permit traffic through a tunnel if the traffic matches a specified tuple of local address, remote address, and service.

- You can use a route-based VPN tunnel configuration, which must have a user-defined proxy ID. With a route-based VPN tunnel configuration, you do not specifically reference a VPN tunnel in a policy. Instead, the policy controls access (permit or deny) to a particular destination. The route to that destination points to a tunnel interface that in turn is bound to a VPN tunnel. Because the VPN tunnel is not directly associated with a policy from which it can derive a proxy ID from the source address, destination address, and service, you must manually define a proxy ID for it. (Note that a route-based VPN configuration also allows you to create multiple policies that make use of a single VPN tunnel; that is, a single Phase 2 SA.)

Consider the addresses in Figure 232 on page 865, which illustrates a VPN tunnel between two sites with overlapping address spaces.

Figure 232: Overlapping Addresses at Peer Sites



If the security devices in Figure 232 on page 865 derive proxy IDs from the policies, as they do in policy-based VPN configurations, then the inbound and outbound policies produce the following proxy IDs:

Device A				**Device B**			
	Local	Remote	Service		Local	Remote	Service
Outbound	10.10.1.2/32	10.20.2.5/32	Any	Inbound	10.20.2.5/32	10.10.1.2/32	Any
Inbound	10.10.1.5/32	10.20.2.2/32	Any	Outbound	10.20.2.2/32	10.10.1.5/32	Any

As shown in the table, there are two proxy IDs: one for outbound VPN traffic and another for inbound. When Device A first sends traffic from 10.10.1.2/32 to 10.20.2.5/32, the two peers perform IKE negotiations and produce Phase 1 and Phase 2 security associations (SAs). The Phase 2 SA results in the above outbound proxy ID for Device A, and the inbound proxy ID for Device B.

If Device B then sends traffic to Device A, the policy lookup for traffic from 10.20.2.2/32 to 10.10.1.5/32 indicates that there is no active Phase 2 SA for such a proxy ID. Therefore, the two peers use the existing Phase 1 SA (assuming that its lifetime has not yet expired) to negotiate a different Phase 2 SA. The resulting proxy IDs are shown above as the inbound proxy ID for Device A and the outbound proxy ID for Device B. There are two Phase 2 SAs—two VPN tunnels—because the addresses are asymmetrical and require different proxy IDs.

To create just one tunnel for bidirectional VPN traffic, you can define the following proxy IDs with addresses whose scope includes both the translated source and destination addresses at each end of the tunnel:

Device A			**Device B**		
Local	Remote	Service	Local	Remote	Service
10.10.1.0/24	10.20.2.0/24	Any	10.20.2.0/24	10.10.1.0/24	Any
		or			
0.0.0.0/0	0.0.0.0/0	Any	0.0.0.0/0	0.0.0.0/0	Any

The above proxy IDs encompass addresses appearing in both inbound and outbound VPN traffic between the two sites. The address 10.10.1.0/24 includes both the DIP pool 10.10.1.2 – 10.10.1.2 and the MIP 10.10.1.5. Likewise, the address 10.20.2.0/24 includes both the DIP pool 10.20.2.2 – 10.20.2.2 and the MIP 10.20.2.5. The above proxy IDs are symmetrical; that is, the local address for Device A is the remote address for Device B, and vice versa. If Device A sends traffic to Device B, the Phase 2 SA and proxy ID also apply to traffic sent from Device B to Device A. Thus, a single Phase 2 SA—that is, a single VPN tunnel—is all that is required for bidirectional traffic between the two sites.

NOTE: The address 0.0.0.0/0 includes all IP addresses, and thus the addresses of the DIP pool and MIP.

To create one VPN tunnel for bidirectional traffic between sites with overlapping address spaces when the addresses for NAT-src and NAT-dst configured on the same device are in different subnets from each other, the proxy ID for the tunnel must be (local IP) 0.0.0.0/0 – (remote IP) 0.0.0.0/0 – *service type*. If you want to use more restrictive addresses in the proxy ID, then the addresses for NAT-src and NAT-dst must be in the same subnet.

In this example, you configure a VPN tunnel between Device A at a corporate site and Device B at a branch office. The address space for the VPN end entities overlaps; they both use addresses in the 10.1.1.0/24 subnet. To overcome this conflict, you use NAT-src to translate the source address on outbound VPN traffic and NAT-dst to translate the destination address on inbound VPN traffic. The policies permit all addresses in the corporate LAN to reach an FTP server at the branch site, and for all addresses at the branch office site to reach an FTP server at the corporate site.

NOTE: For more information about Source Network Address Translation (NAT-src) and Destination Network Address Translation (NAT-dst), see *"Address Translation" on page 1467*.

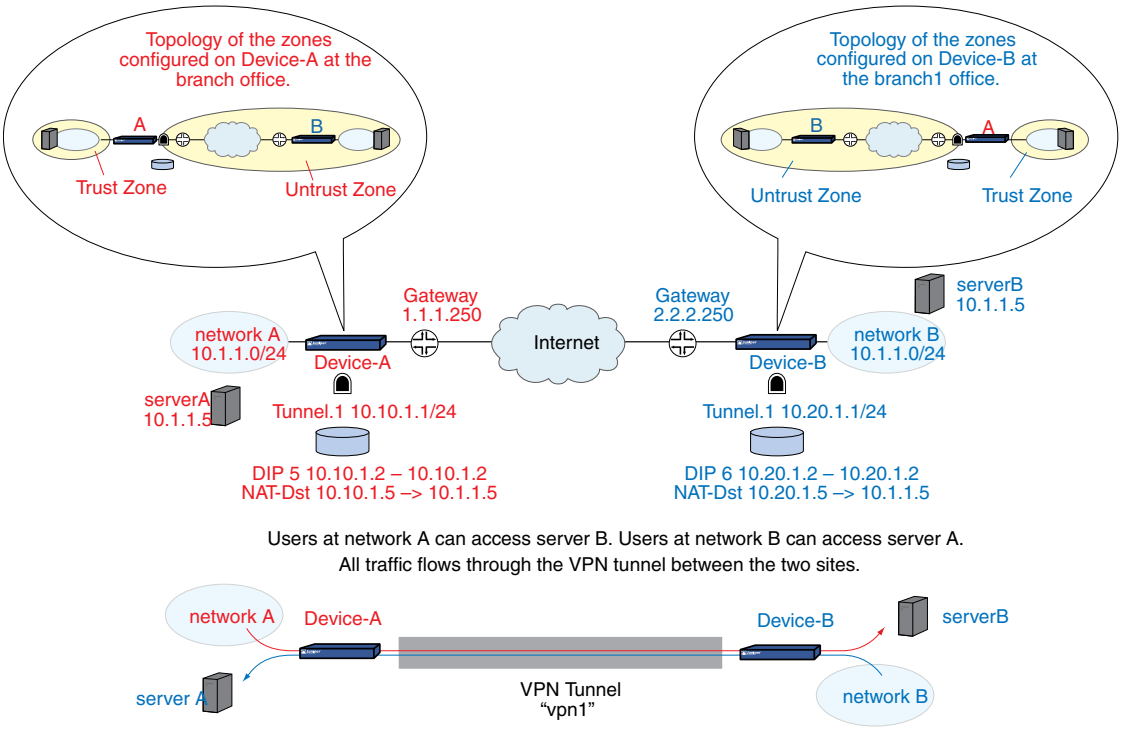
The tunnel configurations at both ends of the tunnel use the following parameters: AutoKey IKE, preshared key ("netscreen1"), and the security level predefined as "Compatible" for both Phase 1 and Phase 2 proposals. (For details about these proposals, see "Tunnel Negotiation" on page 715.)

The outgoing interface on Device A at the corporate site is ethernet3, which has IP address 1.1.1.1/24 and is bound to the Untrust zone. Device B at the branch office uses this address as its remote IKE gateway.

The outgoing interface on Device B at the branch office is ethernet3, which has IP address 2.2.2.2/24 and is bound to the Untrust zone. Device A at the corporate site uses this address as its remote IKE gateway.

The Trust zone interface on both security devices is ethernet1 and has IP address 10.1.1.1/24. All zones on both security devices are in the trust-vr routing domain.

Figure 233: Tunnel Interface with NAT-Src and NAT-Dst



WebUI (Device A)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Fixed IP: (select)
   IP Address / Netmask: 10.10.1.1/24

2. **DIP**

   Network > Interfaces > Edit (for tunnel.1) > DIP > New: Enter the following, then click **OK**:

   ID: 5
   IP Address Range: (select), 10.10.1.2 ~ 10.10.1.2
   Port Translation: (select)
   In the same subnet as the interface IP or its secondary IPs: (select)

3. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: corp
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: virtualA
   IP Address/Domain Name:
   IP/Netmask: (select), 10.10.1.5/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: branch1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.20.1.2/32
   Zone: Untrust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: serverB
   IP Address/Domain Name:
   IP/Netmask: (select), 10.20.1.5/32
   Zone: Untrust

4. **VPN**

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: vpn1
   Security Level: Compatible
   Remote Gateway: Create a Simple Gateway: (select)
   Gateway Name: branch1
   Type: Static IP: (select), Address/Hostname: 2.2.2.2
   Preshared Key: netscreen1
   Security Level: Compatible
   Outgoing Interface: ethernet3

NOTE: The outgoing interface does not have to be in the same zone to which the tunnel interface is bound, although in this case they are in the same zone.

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 10.10.1.0/24
Remote IP / Netmask: 10.20.1.0/24
Service: ANY

5. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.20.1.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.20.1.0/24
Gateway: (select)
Interface: Null
Gateway IP Address: 0.0.0.0
Metric: 10

6. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), corp
Destination Address:
Address Book Entry: (select), serverB
Service: FTP
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Source Translation: (select)
DIP On: 5 (10.10.1.2–10.10.1.2)/X-late

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), branch1
Destination Address:
Address Book Entry: (select), virtualA
Service: FTP
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Destination Translation: (select)
Translate to IP: (select), 10.1.1.5
Map to Port: (clear)

WebUI (Device B)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24

Select the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 2.2.2.2/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK:**

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Fixed IP: (select)
IP Address / Netmask: 10.20.1.1/24

2. **DIP**

Network > Interfaces > Edit (for tunnel.1) > DIP > New: Enter the following, then click **OK**:

ID: 6
IP Address Range: (select), 10.20.1.2 ~ 10.20.1.2
Port Translation: (select)
In the same subnet as the interface IP or its secondary IPs: (select)

3. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: branch1
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: virtualB
IP Address/Domain Name:
IP/Netmask: (select), 10.20.1.5/32
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: corp
IP Address/Domain Name:
IP/Netmask: (select), 10.10.1.2/32
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: serverA
IP Address/Domain Name:
IP/Netmask: (select), 10.10.1.5/32
Zone: Untrust

4. **VPN**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: corp
Type: Static IP: (select), Address/Hostname: 1.1.1.1
Preshared Key: netscreen1
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 10.20.1.0/24
Remote IP / Netmask: 10.10.1.0/24
Service: ANY

NOTE: The outgoing interface does not have to be in the same zone to which the tunnel interface is bound, although in this case they are in the same zone.

5. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 2.2.2.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.10.1.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.10.1.0/24
Gateway: (select)
Interface: Null
Gateway IP Address: 0.0.0.0
Metric: 10

6. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), corp
Destination Address:
Address Book Entry: (select), serverA
Service: FTP
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Source Translation: (select)
DIP on: 6 (10.20.1.2–10.20.1.2)/X-late

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select), corp
Destination Address:
Address Book Entry: (select), virtualB
Service: FTP
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Destination Translation: (select)
Translate to IP: 10.1.1.5
Map to Port: (clear)

CLI (Device A)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.10.1.1/24
```

2. **DIP**

```
set interface tunnel.1 dip 5 10.10.1.2 10.10.1.2
```

3. **Addresses**

```
set address trust corp 10.1.1.0/24
set address trust virtualA 10.10.1.5/32
set address untrust branch1 10.20.1.2/32
set address untrust serverB 10.20.1.5/32
```

4. **VPN**

```
set ike gateway branch1 address 2.2.2.2 outgoing-interface ethernet3 preshare
 netscreen1 sec-level compatible
set vpn vpn1 gateway branch1 sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 10.10.1.0/24 remote-ip 10.20.1.0/24 any
```

NOTE: The outgoing interface does not have to be in the same zone to which the tunnel interface is bound, although in this case they are in the same zone.

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
set vrouter trust-vr route 10.20.1.0/24 interface tunnel.1
set vrouter trust-vr route 10.20.1.0/24 interface null metric 10
```

6. **Policies**

```
set policy top from trust to untrust corp serverB ftp nat src dip-id 5 permit
set policy top from untrust to trust branch1 virtualA ftp nat dst ip 10.1.1.5 permit
save
```

CLI (Device B)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.20.1.1/24
```

2. **DIP**

```
set interface tunnel.1 dip 6 10.20.1.2 10.20.1.2
```

3. **Addresses**

```
set address trust branch1 10.1.1.0/24
set address trust virtualB 10.20.1.5/32
set address untrust corp 10.10.1.2/32
set address untrust serverA 10.10.1.5/32
```

4. **VPN**

```
set ike gateway corp address 1.1.1.1 outgoing-interface ethernet3 preshare
netscreen1 sec-level compatible
set vpn vpn1 gateway corp sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 10.20.1.0/24 remote-ip 10.10.1.0/24 any
```

NOTE: The outgoing interface does not have to be in the same zone to which the tunnel interface is bound, although in this case they are in the same zone.

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
set vrouter trust-vr route 10.10.1.0/24 interface tunnel.1
set vrouter trust-vr route 10.10.1.0/24 interface null metric 10
```

6. **Policies**

```
set policy top from trust to untrust branch1 serverA ftp nat src dip-id 6 permit
set policy top from untrust to trust corp virtualB ftp nat dst ip 10.1.1.5 permit
save
```

Transparent Mode VPN

When the security device interfaces are in transparent mode (that is, they have no IP addresses and are operating at Layer 2 in the OSI Model), you can use the VLAN1 IP address as a VPN termination point. In place of an outgoing interface, as used when the interfaces are in route or NAT mode (that is, they have IP addresses and are operating at Layer 3), a VPN tunnel references an outgoing zone. By default, a tunnel uses the V1-Untrust zone as its outgoing zone. If you have multiple interfaces bound to the same outgoing zone, the VPN tunnel can use any one of them.

NOTE: The OSI Model is a networking industry standard model of network protocol architecture. The OSI Model consists of seven layers, in which Layer 2 is the Data-Link Layer and Layer 3 is the Network Layer.

At the time of this release, a security device whose interfaces are in transparent mode supports only policy-based VPNs. For more information about transparent mode, see “Transparent Mode” on page 99.

It is not necessary that the interfaces of both security devices be in transparent mode. The interfaces of the device at one end of the tunnel can be in transparent mode and those of the other device can be in route or NAT mode.

In this example, you set up a policy-based AutoKey IKE VPN tunnel between two security devices with interfaces operating in transparent mode.

NOTE: It is not necessary that the interfaces of both security devices be in transparent mode. The interfaces of the device at one end of the tunnel can be in transparent mode and those of the other device can be in route or NAT mode.

The key elements of the configuration for the security devices at both ends of the tunnel are as follows:

Configuration Elements	Device A	Device B
V1-Trust Zone	Interface: ethernet1, 0.0.0.0/0 (enable management for the local admin)	Interface: ethernet1, 0.0.0.0/0 (enable management for the local admin)

Configuration Elements	Device A	Device B
V1-Untrust Zone	Interface: ethernet3, 0.0.0.0/0	Interface: ethernet3, 0.0.0.0/0
VLAN1 Interface	IP Address: 1.1.1.1/24 Manage IP: 1.1.1.2 *Note:* You can separate administrative from VPN traffic by using the Manage IP address to receive administrative traffic and the VLAN1 address to terminate VPN traffic.	IP Address: 2.2.2.2/24 Manage IP: 2.2.2.3
Addresses	local_lan: 1.1.1.0/24 in V1-Trust peer_lan: 2.2.2.0/24 in V1-Untrust	local_lan: 2.2.2.0/24 in V1-Trust peer_lan: 1.1.1.0/24 in V1-Untrust
IKE Gateway	gw1, 2.2.2.2, preshared key h1p8A24nG5, security: compatible	gw1, 1.1.1.1, preshared key h1p8A24nG5, security: compatible
VPN tunnel	security: compatible	security: compatible
Policies	local_lan -> peer_lan, any service, vpn1 peer_lan -> local_lan, any service, vpn1	local_lan -> peer_lan, any service, vpn1 peer_lan -> local_lan, any service, vpn1
External Router	IP Address: 1.1.1.250	IP Address: 2.2.2.250
Route	0.0.0.0/0, use VLAN1 interface to gateway 1.1.1.250	0.0.0.0/0, use VLAN1 interface to gateway 2.2.2.250

Configuring a policy-based AutoKey IKE tunnel for a security device whose interfaces are in transparent mode involves the following steps:

1. Remove any IP addresses from the physical interfaces, and bind them to the Layer 2 security zones.
2. Assign an IP address and Manage IP address to the VLAN1 interface.
3. Enter the IP addresses for the local and remote endpoints in the address books for the V1-Trust and V1-Untrust zones.
4. Configure the VPN tunnel and designate its outgoing zone as the V1-Untrust zone.
5. Enter a default route to the external router in the trust-vr.
6. Set up policies for VPN traffic to pass between each site.

WebUI (Device A)

1. **Interfaces**

NOTE: Moving the VLAN1 IP address to a different subnet causes the security device to delete any routes involving the previous VLAN1 interface. When configuring a security device through the WebUI, your workstation must reach the first VLAN1 address and then be in the same subnet as the new address. After changing the VLAN1 address, you must then change the IP address of your workstation so that it is in the same subnet as the new VLAN1 address. You might also have to relocate your workstation to a subnet physically adjacent to the security device.

Network > Interfaces > Edit (for the VLAN1 interface): Enter the following, then click **OK**:

IP Address/Netmask: 1.1.1.1/24
Manage IP: 1.1.1.2
Management Services: WebUI, Telnet, Ping

NOTE: You enable the management options for WebUI, Telnet, and Ping on both the V1-Trust zone and the VLAN1 interface so that a local admin in the V1-Trust zone can reach the VLAN1 Manage IP address. If management through the WebUI is not already enabled on VLAN1 and the V1-Trust zone interfaces, you cannot reach the security device through the WebUI to make these settings. Instead, you must first set WebUI manageability on these interfaces through a console connection.

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Management Services: WebUI, Telnet
Other Services: Ping

Select the following, then click **OK**:

Zone Name: V1-Trust
IP Address/Netmask: 0.0.0.0/0

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: V1-Untrust
IP Address/Netmask: 0.0.0.0/0

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: local_lan
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.0/24
Zone: V1-Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: peer_lan
IP Address/Domain Name:
IP/Netmask: (select), 2.2.2.0/24
Zone: V1-Untrust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: gw1
Security Level: Compatible
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 2.2.2.2
Preshared Key: h1p8A24nG5
Outgoing Zone: V1-Untrust

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway:
Predefined: (select), gw1

4. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: VLAN1 (VLAN)
Gateway IP Address: 1.1.1.250

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), local_lan
Destination Address:
Address Book Entry: (select), peer_lan
Service: ANY
Action: Tunnel
Tunnel VPN: vpn1
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

WebUI (Device B)

1. **Interfaces**

NOTE: Moving the VLAN1 IP address to a different subnet causes the security device to delete any routes involving the previous VLAN1 interface. When configuring a security device through the WebUI, your workstation must reach the first VLAN1 address and then be in the same subnet as the new address. After changing the VLAN1 address, you must then change the IP address of your workstation so that it is in the same subnet as the new VLAN1 address. You might also have to relocate your workstation to a subnet physically adjacent to the security device.

Network > Interfaces > Edit (for the VLAN1 interface): Enter the following, then click **OK**:

IP Address/Netmask: 2.2.2.2/24
Manage IP: 2.2.2.3
Management Services: WebUI, Telnet, Ping

NOTE: If management through the WebUI is not already enabled on VLAN1 and the V1-Trust zone interfaces, you cannot reach the security device through the WebUI to make these settings. Instead, you must first set WebUI manageability on these interfaces through a console connection.

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Management Services: WebUI, Telnet
Other Services: Ping

Select the following, then click **OK**:

Zone Name: V1-Trust
IP Address/Netmask: 0.0.0.0/0

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: V1-Untrust
IP Address/Netmask: 0.0.0.0/0

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: local_lan
IP Address/Domain Name:
IP/Netmask: (select), 2.2.2.0/24
Zone: V1-Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: peer_lan
IP Address/Domain Name:

IP/Netmask: (select), 1.1.1.0/24
Zone: V1-Untrust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: gw1
Security Level: Compatible
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 1.1.1.1
Preshared Key: h1p8A24nG5
Outgoing Zone: V1-Untrust

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway:
Predefined: (select), gw1

4. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: VLAN1 (VLAN)
Gateway IP Address: 2.2.2.250

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), local_lan
Destination Address:
Address Book Entry: (select), peer_lan
Service: ANY
Action: Tunnel
Tunnel VPN: vpn1
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

CLI (Device A)

1. **Interfaces and Zones**

```
unset interface ethernet1 ip
unset interface ethernet1 zone
set interface ethernet1 zone v1-trust
set zone v1-trust manage web
set zone v1-trust manage telnet
```

```
set zone v1-trust manage ping
unset interface ethernet3 ip
unset interface ethernet3 zone
set interface ethernet3 zone v1-untrust
set interface vlan1 ip 1.1.1.1/24
set interface vlan1 manage-ip 1.1.1.2
set interface vlan1 manage web
set interface vlan1 manage telnet
set interface vlan1 manage ping
```

NOTE: You enable the management options for WebUI, Telnet, and Ping on both the V1-Trust zone and the VLAN1 interface so that a local admin in the V1-Trust zone can reach the VLAN1 Manage IP address.

2. **Addresses**

```
set address v1-trust local_lan 1.1.1.0/24
set address v1-untrust peer_lan 2.2.2.0/24
```

3. **VPN**

```
set ike gateway gw1 address 2.2.2.2 main outgoing-interface v1-untrust preshare
 h1p8A24nG5 sec-level compatible
set vpn vpn1 gateway gw1 sec-level compatible
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface vlan1 gateway 1.1.1.250
```

5. **Policies**

```
set policy top from v1-trust to v1-untrust local_lan peer_lan any tunnel vpn vpn1
set policy top from v1-untrust to v1-trust peer_lan local_lan any tunnel vpn vpn1
save
```

CLI (Device B)

1. **Interfaces and Zones**

```
unset interface ethernet1 ip
unset interface ethernet1 zone
set interface ethernet1 zone v1-trust
set zone v1-trust manage
unset interface ethernet3 ip
unset interface ethernet3 zone
set interface ethernet3 zone v1-untrust
set interface vlan1 ip 2.2.2.2/24
set interface vlan1 manage-ip 2.2.2.3
set interface vlan1 manage
```

2. **Addresses**

```
set address v1-trust local_lan 2.2.2.0/24
set address v1-untrust peer_lan 1.1.1.0/24
```

3. **VPN**

```
set ike gateway gw1 address 1.1.1.1 main outgoing-interface v1-untrust preshare
 h1p8A24nG5 sec-level compatible
set vpn vpn1 gateway gw1 sec-level compatible
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface vlan1 gateway 2.2.2.250
```

5. **Policies**

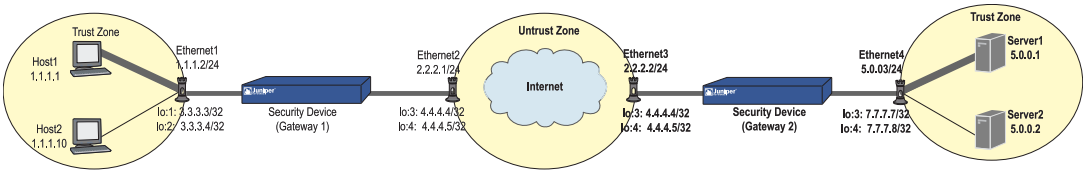
```
set policy top from v1-trust to v1-untrust local_lan peer_lan any tunnel vpn vpn1
set policy top from v1-untrust to v1-trust peer_lan local_lan any tunnel vpn vpn1
save
```

Transport mode IPsec VPN

Juniper Networks security devices support transport mode IPSsec VPN for traffic between the security gateways. In order to support transport mode IPsec for traffic between the gateways, the security gateway meets the RFC standard that the source address for outgoing packets and the destination address for incoming packets is an address belonging to the security gateway.

Consider the scenario explained in Figure 1, in which the two hosts (h-1 and h-2) are in one trust zone and the two servers (s-1 and s-2) are in another trust zone. GW -1 and GW -2 are in an untrust zone.

Figure 234: Transport Mode IPsec VPN



To configure NAT transport mode in the above scenario, perform the following steps:

1. Build transport mode IPSec VPN between host-1 and GW-1
2. Build transport or tunnel mode IPSec VPN between GW-1 and GW-2
3. Build transport mode IPSec VPN between GW -2 and server-1
4. Do source-NAT and MIP on GW-1
5. Do MIP (MIP and reversed MIP) on GW-2.

The following section explains the steps involved in configuring a transport mode IPsec VPN. GW-1 uses src-NAT when h-1 is accessing s-1, and GW-2 uses MIP when h-2 is accessing s-2.

Gateway - 1 Configuration

1. **IKE Configuration on host-1 and host-2**

   ```
   set ike gateway gateway1 address 1.1.1.1 aggressive outgoing-interface
   loopback.1 preshare test1 sec-level standard
   set ike gateway gateway2 address 1.1.1.10 aggressive outgoing-interface
   loopback.2 preshare test1 sec-level standard
   ```

2. **VPN Configuration on host-1 and host-2**

   ```
   set vpn v1 gateway gateway1 transport sec-level standard
   set vpn v2 gateway gateway2 transport sec-level standard
   ```

3. **MIP Configuration**

   ```
   set interface loopback.1 mip 3.3.3.3 host 6.6.6.6
   set interface loopback.2 mip 3.3.3.4 host 6.6.6.7
   ```

4. **IKE Configuration for GW-2**

   ```
   set ike gateway s1 address 6.6.6.6 aggressive outgoing-interface loopback.3
   preshare test1 sec-level standard
   set ike gateway s2 address 6.6.6.7 aggressive outgoing-interface loopback.4
   preshare test1 sec-level standard
   ```

5. **VPN Configuration for s1 and s2**

   ```
   set vpn v3 gateway s1 transport sec-level standard
   set vpn v4 gateway s2 transport sec-level standard
   ```

6. **DIP Configuration**

   ```
   set interface eth2 ext ip 4.4.4.4 255.255.255.255 dip 10 4.4.4.4 4.4.4.4
   set interface eth2 ext ip 4.4.4.5 255.255.255.255 dip 11 4.4.4.5 4.4.4.5
   ```

7. **Policy Setup**

   Outgoing policy

   ```
   set policy id 3 from trust to untrust "1.1.1.1" "3.3.3.3" any nat src dip-id 10
   tunnel vpn v3
   set policy id 4 from trust to untrust "1.1.1.10" "3.3.3.4" any nat src dip-id 11
   tunnel vpn v4
   ```

   Incoming policy

   ```
   set policy id 1 from trust to untrust "1.1.1.1" "(MIP)3.3.3.3" any tunnel vpn v1
   set policy id 2 from trust to untrust "1.1.1.10" "(MIP)3.3.3.4" any tunnel vpn
   v2
   ```

NOTE: Users need to configure the outgoing policy before configuirng the incoming policy. This is because we do policy search twice, the first one is to check the incoming packet, and the second one is to find another VPN (the outgoing VPN) through which we send the packet.

GW-2 Configuration

8. **IKE and VPN Configuration to Server-PC**

```
set ike gateway gateway1 address  5.0.0.1 aggressive outgoing-interface lo.3
preshare test sec-level standard
set ike gateway gateway2 address  5.0.0.2 aggressive outgoing-interface lo.4
preshare test sec-level standard
```

9. **VPN Configuration on server-1 and server-2**

```
set vpn v3 gateway gateway1 transport sec-level standard
set vpn v4 gateway gateway2 transport sec-level standard
```

10. **Reversed MIP (Traffic Is from Untrust to Trust)**

```
set interface lo.3 mip 7.7.7.7 host 4.4.4.4
set interface lo.4 mip 7.7.7.8 host 4.4.4.5
```

11. **IKE and VPN configuration to GW-1 (Client-PC)**

```
set ike gateway h1 address 4.4.4.4 aggressive outgoing-interface lo.1 preshare
 test sec-level standard
set ike gateway h2 address 4.4.4.5 aggressive outgoing-interface lo.2 preshare
 test sec-level standard
```

12. **VPN Configuration on host-1 and host-2**

```
set vpn v1 gateway h1 transport sec-level standard
set vpn v2 gateway h2 transport sec-level standard
```

13. **MIP**

```
set interface lo.1 mip 6.6.6.6 host 5.0.0.1
set interface lo.2 mip 6.6.6.7 host 5.0.0.2
```

14. **Policy Setup**

Outgoing policy

```
set policy id 7 from untrust to trust "4.4.4.4" "6.6.6.6" any tunnel vpn v3
set policy id 8 from untrust to trust "4.4.4.5" "6.6.6.7" any tunnel vpn v4
```

Incoming policy

```
set policy id 5 from untrust to trust "4.4.4.4" "(MIP)6.6.6.6" any tunnel vpn v1
set policy id 6 from untrust to trust "4.4.4.5" "(MIP)6.6.6.7" any tunnel vpn v2
```

1. When a packet from h-1 arrives at GW-1, the GW-1 decrypts the packet and finds the destination MIP for the packet.
2. GW-1 matches the packet against the policy (policy 1) that defines the host VPN. It does a policy search again and finds the policy (policy 3) that defines the server VPN and src-nat.
3. GW-1 then does a destination MIP, which changes the destination IP address from 3.3.3.3 to 6.6.6.6 and the source NAT, which changes the source IP address from 1.1.1.1 to 4.4.4.4.
4. GW-2 decrypts the packet and finds the destination MIP for the packet.

   It matches the decrypted packet with the policy (policy 5) that defines the host VPN. It does a policy search again and finds the policy (policy 7) that defines the server VPN.
5. Before sending the packet out, GW-2 finds the reversed-MIP on lo.3 for packet's src-IP 4.4.4.4, so the src-ip is changed from 4.4.4.4 to 7.7.7.7
6. GW-2 forwards the packet to s-1 through interface 7.7.7.7
7. S-1 (5.0.0.1) processes the packet and sends it to GW-2 (6.6.6.6) through interface 7.7.7.7.
8. GW-2 identifies the reversed MIP (7.7.7.7 -> 4.4.4.4) and sends the packet to GW-1 (4.4.4.4). From GW-1, the packet is sent to h-1.

Chapter 23

Dialup Virtual Private Networks

Juniper Networks security devices can support dialup virtual private network (VPN) connections. You can configure a security device that has a static IP address to secure an IPsec tunnel with a NetScreen-Remote client or with another security device with a dynamic IP address.

This chapter contains the following sections:

Dialup

You can configure tunnels for VPN dialup users individually, or you can form users into a VPN dialup group for which you need only configure one tunnel. You can also create a Group IKE ID user that allows you to define one user whose IKE ID is used as part of the IKE IDs of dialup IKE users. This approach is particularly timesaving when there are large groups of dialup users because you do not have to configure each IKE user individually.

NOTE: For more information about creating IKE user groups, see "IKE Users and User Groups" on page 1637. For more information about the Group IKE ID feature, see "Group IKE ID" on page 911.

If the dialup client can support a virtual internal IP address, which the NetScreen-Remote does, you can also create a dynamic peer dialup VPN, AutoKey IKE tunnel (with a preshared key or certificates). You can configure a Juniper Networks security gateway with a static IP address to secure an IPsec tunnel with a NetScreen-Remote client or with another security device with a dynamic IP address.

NOTE: For background information about the available VPN options, see "Internet Protocol Security" on page 707. For guidance when choosing among the various options, see "Virtual Private Network Guidelines" on page 769.

You can configure policy-based VPN tunnels for VPN dialup users. For a dialup dynamic peer client, you can configure either a policy-based or route-based VPN.

Because a dialup dynamic peer client can support a virtual internal IP address, which the NetScreen-Remote does, you can configure a routing table entry to that virtual internal address through a designated tunnel interface. Doing so allows you to configure a route-based VPN tunnel between the security device and that peer.

NOTE: A dialup dynamic peer client is a dialup client that supports a virtual internal IP address.

The dialup dynamic peer is nearly identical to the Site-to-Site dynamic peer except that the internal IP address for the dialup client is a virtual address.

Policy-Based Dialup VPN, AutoKey IKE

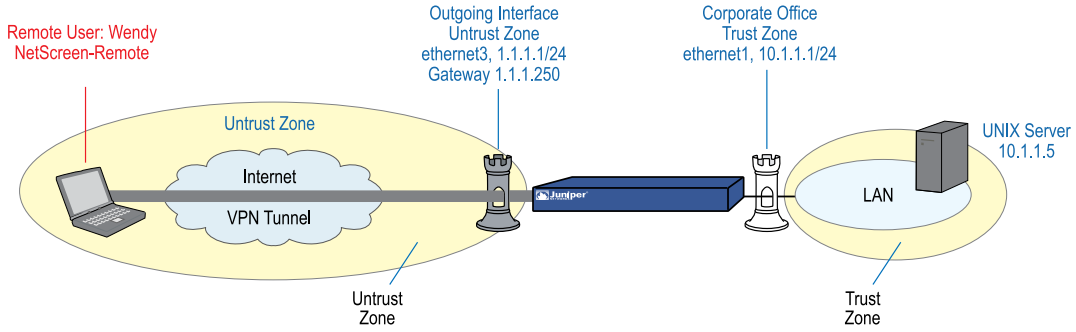
In this example, an AutoKey IKE tunnel using either a preshared key or a pair of certificates (one at each end of the tunnel) provides the secure communication channel between the IKE user Wendy and the UNIX server. The tunnel again uses ESP with 3DES encryption and SHA-1 authentication.

NOTE: The preshared key is h1p8A24nG5. It is assumed that both participants already have certificates. For more information about certificates, see "Certificates and CRLs" on page 746.

Setting up the AutoKey IKE tunnel using AutoKey IKE with either a preshared key or certificates requires the following configuration at the corporate site:

1. Configure interfaces for the Trust and Untrust zones, both of which are in the trust-vr routing domain.
2. Enter the address of the UNIX server in the Trust zone address book.
3. Define Wendy as an IKE user.
4. Configure the remote gateway and AutoKey IKE VPN.
5. Set up a default route.
6. Create a policy from the Untrust zone to the Trust zone permitting access to the UNIX from the dialup user.

Figure 235: Policy-Based Dialup VPN, AutoKey IKE



The preshared key is h1p8A24nG5. This example assumes that both participants already have RSA certificates issued by Verisign and that the local certificate on the NetScreen-Remote contains the U-FQDN wparker@email.com. (For information about obtaining and loading certificates, see "Certificates and CRLs" on page 746.) For the Phase 1 and 2 security levels, you specify one Phase 1 proposal—either pre-g2-3des-sha for the preshared key method or rsa-g2-3des-sha for certificates—and select the predefined "Compatible" set of proposals for Phase 2.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK:**

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: UNIX
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.5/32
   Zone: Trust

3. **User**

   Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Wendy
Status: Enable (select)
IKE User: (select)
Simple Identity: (select)
IKE Identity: wparker@email.com

4. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: Wendy_NSR
Security Level: Custom
Remote Gateway Type:
Dialup User: (select), User: Wendy

Certificates

Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): rsa-g2-3des-sha
Mode (Initiator): Aggressive
Preferred Certificate (optional):
Peer CA: Verisign
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK:**

VPN Name: Wendy_UNIX
Security Level: Compatible
Remote Gateway:
Predefined: (select), Wendy_NSR

(or)

Preshared Key

CAUTION: Aggressive mode is insecure. Because of protocol limitations, main mode IKE in combination with preshared key (PSK) is not possible for dialup VPN users. In addition, it is never advisable to use aggressive mode because this mode has inherent security problems. Consequently, it is strongly advisable to configure dialup VPN users with PKI certificates and main mode.

Preshared Key: h1p8A24nG5
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Aggressive

5. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

6. **Policy**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Dial-Up VPN
Destination Address:
Address Book Entry: (select), UNIX
Service: ANY
Action: Tunnel
Tunnel VPN: Wendy_UNIX
Modify matching bidirectional VPN policy: (clear)
Position at Top: (select)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Address**

```
set address trust unix 10.1.1.5/32
```

3. **User**

```
set user wendy ike-id u-fqdn wparker@email.com
```

4. **VPN**

Certificates

```
set ike gateway wendy_nsr dialup wendy aggressive outgoing-interface ethernet3
 proposal rsa-g2-3des-sha
set ike gateway wendy_nsr cert peer-ca 1
set ike gateway wendy_nsr cert peer-cert-type x509-sig
set vpn wendy_unix gateway wendy_nsr sec-level compatible
```

NOTE: The number 1 is the CA ID number. To discover the CA's ID number, use the following command: **get pki x509 list ca-cert**.

(or)

Preshared Key

CAUTION: Aggressive mode is insecure. Due to protocol limitations, main mode IKE in combination with preshared key (PSK) is not possible for dialup VPN users. In addition, it is never advisable to use aggressive mode because this mode has inherent insecurity problems. Consequently, it is strongly advisable to configure dialup VPN users with PKI certificates and main mode.

```
set ike gateway wendy_nsr dialup wendy aggressive outgoing-interface ethernet3
 preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn wendy_unix gateway wendy_nsr sec-level compatible
```

5. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

6. **Policy**

```
set policy top from untrust to trust "Dial-Up VPN" unix any tunnel vpn wendy_unix
save
```

NetScreen-Remote Security Policy Editor

1. Click **Options > Secure > Specified Connections**.
2. Click **Add a new connection**, and type **UNIX** next to the new connection icon that appears.
3. Configure the connection options:

```
Connection Security: Secure
Remote Party Identity and Addressing:
   ID Type: IP Address, 10.1.1.5
   Protocol: All
   Connect using Secure Gateway Tunnel: (select)
   ID Type: IP Address, 1.1.1.1
```

4. Click the **PLUS** symbol, located to the left of the UNIX icon, to expand the connection policy.
5. Click **My Identity**: Do either of the following:

   Click **Pre-shared Key > Enter Key**: Type **h1p8A24nG5**, then click **OK**.

   ID Type: (select **E-mail Address**), and type **wparker@email.com**.

(or)

Select a certificate from the Select Certificate drop-down list.

ID Type: (select **E-mail Address**)

NOTE: The email address from the certificate automatically appears in the identifier field.

6. Click the **Security Policy** icon, then select **Aggressive Mode** and clear **Enable Perfect Forward Secrecy (PFS)**.
7. Click the **PLUS** symbol, located to the left of the Security Policy icon, and then the **PLUS** symbol to the left of Authentication (Phase 1) and Key Exchange (Phase 2) to expand the policy further.
8. Click **Authentication (Phase 1) > Proposal 1**: Select the following authentication method and algorithms:

   Authentication Method: Pre-Shared Key

   (or)

   Authentication Method: RSA Signatures
   Encrypt Alg: Triple DES
   Hash Alg: SHA-1
   Key Group: Diffie-Hellman Group 2

9. Click **Key Exchange (Phase 2) > Proposal 1**: Select the following IPsec protocols:

   Encapsulation Protocol (ESP): (select)
   Encrypt Alg: Triple DES
   Hash Alg: SHA-1
   Encapsulation: Tunnel

10. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: MD5
    Encapsulation: Tunnel

11. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: SHA-1
    Encapsulation: Tunnel

12. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

Encapsulation Protocol (ESP): (select)
Encrypt Alg: DES
Hash Alg: MD5
Encapsulation: Tunnel

13. Click **File > Save Changes**.

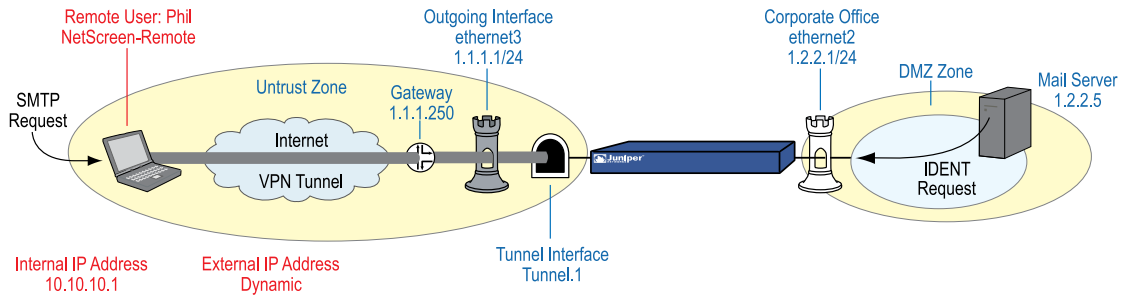
Route-Based Dialup VPN, Dynamic Peer

In this example, a VPN tunnel securely connects the user behind NetScreen-Remote to the Untrust zone interface of the security device protecting the mail server in the DMZ zone. The Untrust zone interface has a static IP address. The NetScreen-Remote client has a dynamically assigned external IP address and a static (virtual) internal IP address. The administrator of the security device must know the peer's internal IP address for the following two purposes:

- The admin can use it in policies.
- The admin can create a route linking the address with a tunnel interface bound to an appropriate tunnel.

After the NetScreen-Remote client establishes the tunnel, traffic through the tunnel can then originate from either end. All zones on the security device are in the trust-vr routing domain.

Figure 236: Route-Based Dialup VPN, Dynamic Peer



In this example, Phil wants to get his email from the mail server at the company site. When he attempts to do so, he is authenticated by the mail server program, which sends him an IDENT request through the tunnel.

NOTE: The mail server can send the IDENT request through the tunnel only if the security administrator adds a custom service for it (TCP, port 113) and sets up an outgoing policy allowing that traffic through the tunnel to 10.10.10.1.

The preshared key is h1p8A24nG5. It is assumed that both participants already have RSA certificates issued by Verisign and that the local certificate on the NetScreen-Remote contains the U-FQDN *pm@juniper.net*. (For information about obtaining and loading certificates, see "Certificates and CRLs" on page 746.) For the Phase 1 and Phase 2 security levels, you specify one Phase 1 proposal—either

pre-g2-3des-sha for the preshared key method or rsa-g2-3des-sha for certificates—and select the predefined "Compatible" set of proposals for Phase 2.

You enter the following three routes on the security device:

- A default route to the external router in the trust-vr
- A route to the destination through the tunnel interface
- A null route to the destination. You assign a higher metric (farther from zero) to the null route so that it becomes the next-choice route to the destination. Then, if the state of the tunnel interface changes to "down" and the route referencing that interface becomes inactive, the security device uses the null route, which essentially drops any traffic sent to it, rather than the default route, which forwards unencrypted traffic.

Finally, you create policies allowing traffic to flow in both directions between Phil and the mail server.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select this option when present)
   IP Address/Netmask: 1.2.2.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Unnumbered: (select)
   Interface: ethernet3 (trust-vr)

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Mail Server
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.5/32
   Zone: DMZ

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Phil
IP Address/Domain Name:
IP/Netmask: (select), 10.10.10.1/32
Zone: Untrust

3. **Services**

Policy > Policy Elements > Services > Custom > New: Enter the following, then click **OK**:

Service Name: Ident
Service Timeout:
Use protocol default: (select)
Transport Protocol: TCP (select)
Source Port: Low 1, High 65535
Destination Port: Low 113, High 113

Policy > Policy Elements > Services > Group > New: Enter the following, move the following services, then click **OK**:

Group Name: Remote_Mail
Group Members << Available Members:
Ident
MAIL
POP3

4. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_Phil
Security Level: Custom
Remote Gateway Type:
Dynamic IP Address: (select), Peer ID: pm@juniper.net

Preshared Key

Preshared Key: h1p8A24nG5
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Aggressive

(or)

Certificates

Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): rsa-g2-3des-sha
Mode (Initiator): Aggressive
Preferred Certificate (optional):
Peer CA: Verisign
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: corp_Phil
Security Level: Compatible
Remote Gateway:
  Predefined: (select), To_Phil

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface: (select), tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 1.2.2.5/32
Remote IP / Netmask: 10.10.10.1/32
Service: Any

5. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
  Interface: ethernet3
  Gateway IP Address: 1.1.1.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.10.10.1/32
Gateway: (select)
  Interface: tunnel.1
  Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.10.10.1/32
Gateway: (select)
  Interface: Null
  Gateway IP Address: 0.0.0.0
  Metric: 10

6. **Policies**

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), Phil

Destination Address:
Address Book Entry: (select), Mail Server
Service: Remote_Mail
Action: Permit
Position at Top: (select)

Policies > (From: DMZ, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Mail Server
Destination Address:
Address Book Entry: (select), Phil
Service: Remote_Mail
Action: Permit
Position at Top: (select)

CLI

1. **Interfaces**

```
set interface ethernet2 zone dmz
set interface ethernet2 ip 1.2.2.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

2. **Addresses**

```
set address dmz “Mail Server” 1.2.2.5/32
set address untrust phil 10.10.10.1/32
```

3. **Services**

```
set service ident protocol tcp src-port 1-65535 dst-port 113-113
set group service remote_mail
set group service remote_mail add ident
set group service remote_mail add mail
set group service remote_mail add pop3
```

4. **VPN**

Preshared Key

```
set ike gateway to_phil dynamic pm@juniper.net aggressive outgoing-interface
ethernet3 preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn corp_phil gateway to_phil sec-level compatible
set vpn corp_phil bind interface tunnel.1
set vpn corp_phil proxy-id local-ip 1.2.2.5/32 remote-ip 10.10.10.1/32 any
```

(or)

Certificates

```
set ike gateway to_phil dynamic pm@juniper.net aggressive outgoing-interface
ethernet3 proposal rsa-g2-3des-sha
set ike gateway to_phil cert peer-ca 1
set ike gateway to_phil cert peer-cert-type x509-sig
set vpn corp_phil gateway to_phil sec-level compatible
set vpn corp_phil bind interface tunnel.1
set vpn corp_phil proxy-id local-ip 1.2.2.5/32 remote-ip 10.10.10.1/32 any
```

NOTE: The number 1 is the CA ID number. To discover the CA's ID number, use the following command: **get pki x509 list ca-cert**.

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
set vrouter trust-vr route 10.10.10.1/32 interface tunnel.1
set vrouter trust-vr route 10.10.10.1/32 interface null metric 10
```

6. **Policies**

```
set policy top from dmz to untrust "Mail Server" phil remote_mail permit
set policy top from untrust to dmz phil "Mail Server" remote_mail permit
save
```

NetScreen-Remote

1. Click **Options > Global Policy Settings**, and select the Allow to Specify Internal Network Address check box.
2. **Options > Secure > Specified Connections**.
3. Click the **Add a new connection** button, and type **Mail** next to the new connection icon that appears.
4. Configure the connection options:

```
Connection Security: Secure
Remote Party Identity and Addressing:
   ID Type: IP Address, 1.2.2.5
   Protocol: All
   Connect using Secure Gateway Tunnel: (select)
   ID Type: IP Address, 1.1.1.1
```

5. Click the **PLUS** symbol, located to the left of the UNIX icon, to expand the connection policy.
6. Click the **Security Policy** icon, then select **Aggressive Mode** and clear **Enable Perfect Forward Secrecy (PFS)**.
7. Click **My Identity** and do either of the following:

   Click **Pre-shared Key > Enter Key**: Type **h1p8A24nG5**, then click **OK**.

```
ID Type: E-mail Address; pm@juniper.net
Internal Network IP Address: 10.10.10.1
```

(or)

Select the certificate that contains the email address “pm@juniper.net” from the Select Certificate drop-down list.

ID Type: E-mail Address; pm@juniper.net
Internal Network IP Address: 10.10.10.1

8. Click the **PLUS** symbol, located to the left of the Security Policy icon, and then the **PLUS** symbol to the left of Authentication (Phase 1) and Key Exchange (Phase 2) to expand the policy further.
9. Click **Authentication (Phase 1) > Proposal 1**: Select the following Authentication Method and Algorithms:

   Authentication Method: Pre-Shared Key

   (or)

   Authentication Method: RSA Signatures
   Encrypt Alg: Triple DES
   Hash Alg: SHA-1
   Key Group: Diffie-Hellman Group 2

10. Click **Key Exchange (Phase 2) > Proposal 1**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: SHA-1
    Encapsulation: Tunnel

11. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: MD5
    Encapsulation: Tunnel

12. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: SHA-1
    Encapsulation: Tunnel

13. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

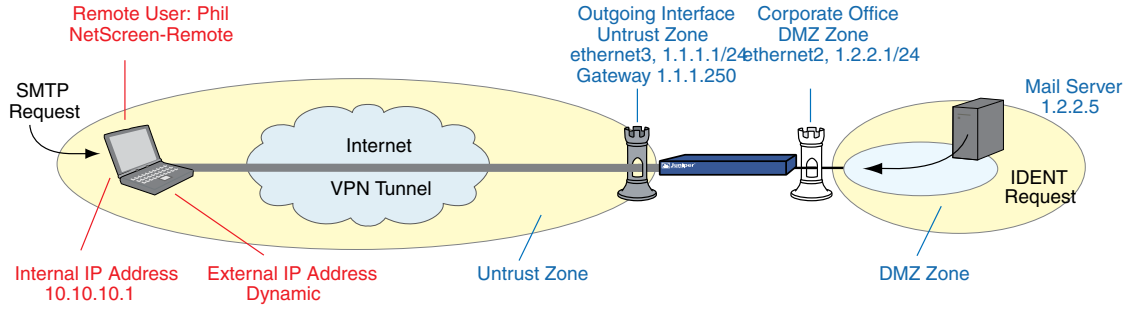
    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: MD5
    Encapsulation: Tunnel

14. Click **File > Save Changes**.

Policy-Based Dialup VPN, Dynamic Peer

In this example, a VPN tunnel securely connects the user behind the NetScreen-Remote to the Untrust zone interface of the security device protecting the mail server in the DMZ zone. The Untrust zone interface has a static IP address. The NetScreen-Remote client has a dynamically assigned external IP address and a static (virtual) internal IP address. The administrator of the security device must know the client's internal IP address so that he can add it to the Untrust address book for use in policies to tunnel traffic from that source. After the NetScreen-Remote client establishes the tunnel, traffic through the tunnel can originate from either end.

Figure 237: Policy-Based Dialup VPN, Dynamic Peer



In this example, Phil wants to get his email from the mail server at the company site. When he attempts to do so, he is authenticated by the mail server program, which sends him an IDENT request through the tunnel.

NOTE: The mail server can send the IDENT request through the tunnel only if the security administrator adds a custom service for it (TCP, port 113) and sets up an outgoing policy allowing that traffic through the tunnel to 10.10.10.1.

The preshared key is h1p8A24nG5. This example assumes that both participants have RSA certificates issued by Verisign and that the local certificate on the NetScreen-Remote contains the U-FQDN *pm@juniper.net*. (For more information about obtaining and loading certificates, see "Certificates and CRLs" on page 746.) For the Phase 1 and Phase 2 security levels, you specify one Phase 1 proposal—either pre-g2-3des-sha for the preshared key method or rsa-g2-3des-sha for certificates—and select the predefined "Compatible" set of proposals for Phase 2.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select this option when present)
   IP Address/Netmask: 1.2.2.1/24

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Mail Server
IP Address/Domain Name:
IP/Netmask: (select), 1.2.2.5/32
Zone: DMZ

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Phil
IP Address/Domain Name:
IP/Netmask: (select), 10.10.10.1/32
Zone: Untrust

3. **Services**

Policy > Policy Elements > Services > Custom > New: Enter the following, then click **OK**:

Service Name: Ident
Service Timeout:
Use protocol default: (select)
Transport Protocol: TCP (select)
Source Port: Low 1, High 65535
Destination Port: Low 113, High 113

Policy > Policy Elements > Services > Group > New: Enter the following, move the following services, then click **OK**:

Group Name: Remote_Mail
Group Members << Available Members:
Ident
MAIL
POP3

4. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_Phil
Security Level: Custom
Remote Gateway Type:
Dynamic IP Address: (select), Peer ID: pm@juniper.net

Preshared Key

Preshared Key: h1p8A24nG5
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Aggressive

(or)

Certificates

Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): rsa-g2-3des-sha
Mode (Initiator): Aggressive
Preferred Certificate (optional):
Peer CA: Verisign
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: corp_Phil
Security Level: Compatible
Remote Gateway:
  Predefined: (select), To_Phil

5. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
  Interface: ethernet3
  Gateway IP Address: 1.1.1.250

6. **Policies**

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), Phil
Destination Address:
  Address Book Entry: (select), Mail Server
Service: Remote_Mail
Action: Tunnel
VPN Tunnel: corp_Phil
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

CLI

1. **Interfaces**

```
set interface ethernet2 zone dmz
set interface ethernet2 ip 1.2.2.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address dmz “mail server” 1.2.2.5/32
set address untrust phil 10.10.10.1/32
```

3. **Services**

```
set service ident protocol tcp src-port 1-65535 dst-port 113-113
set group service remote_mail
set group service remote_mail add ident
set group service remote_mail add mail
set group service remote_mail add pop3
```

4. **VPN**

Preshared Key

```
set ike gateway to_phil dynamic pm@juniper.net aggressive outgoing-interface
ethernet3 preshare h1p8A24nG5 proposal pre-g2-3des-sha
set vpn corp_phil gateway to_phil sec-level compatible
```

(or)

Certificates

```
set ike gateway to_phil dynamic pm@juniper.net aggressive outgoing-interface
ethernet3 proposal rsa-g2-3des-sha
set ike gateway to_phil cert peer-ca 1
set ike gateway to_phil cert peer-cert-type x509-sig
set vpn corp_phil gateway to_phil sec-level compatible
```

NOTE: The number 1 is the CA ID number. To discover the CA’s ID number, use the following command: **get pki x509 list ca-cert**.

5. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

6. **Policies**

```
set policy top from untrust to dmz phil “mail server” remote_mail tunnel vpn
corp_phil
set policy top from dmz to untrust “mail server” phil remote_mail tunnel vpn
```

```
corp_phil
save
```

NetScreen-Remote

1. Click **Options > Global Policy Settings**, and select **Allow to Specify Internal Network Address**.
2. **Options > Secure > Specified Connections**.
3. Click **Add a new connection**, and type **Mail** next to the new connection icon that appears.
4. Configure the connection options:

   ```
   Connection Security: Secure
   Remote Party Identity and Addressing:
     ID Type: IP Address, 1.2.2.5
     Protocol: All
     Connect using Secure Gateway Tunnel: (select)
     ID Type: IP Address, 1.1.1.1
   ```

5. Click the **PLUS** symbol, located to the left of the UNIX icon, to expand the connection policy.
6. Click the **Security Policy** icon, then select **Aggressive Mode** and clear **Enable Perfect Forward Secrecy (PFS)**.
7. Click **My Identity** and do either of the following:

   Click **Pre-shared Key > Enter Key**: Type **h1p8A24nG5**, then click **OK**.

   ```
   Internal Network IP Address: 10.10.10.1
   ID Type: E-mail Address; pm@juniper.net
   ```

   (or)

   Select the certificate that contains the email address "pmason@email.com" from the Select Certificate drop-down list.

   ```
   Internal Network IP Address: 10.10.10.1
   ID Type: E-mail Address; pm@juniper.net
   ```

8. Click the **PLUS** symbol, located to the left of the Security Policy icon, and then the **PLUS** symbol to the left of Authentication (Phase 1) and Key Exchange (Phase 2) to expand the policy further.
9. Click **Authentication (Phase 1) > Proposal 1**: Select the following Authentication Method and Algorithms:

   ```
   Authentication Method: Pre-Shared Key
   ```

   (or)

   ```
   Authentication Method: RSA Signatures
   Encrypt Alg: Triple DES
   ```

Hash Alg: SHA-1
Key Group: Diffie-Hellman Group 2

10. Click **Key Exchange (Phase 2) > Proposal 1**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: SHA-1
    Encapsulation: Tunnel

11. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: MD5
    Encapsulation: Tunnel

12. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: SHA-1
    Encapsulation: Tunnel

13. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: MD5
    Encapsulation: Tunnel

14. Click **File > Save Changes**.

Bidirectional Policies for Dialup VPN Users

You can create bidirectional policies for dialup VPNs. This configuration provides similar functionality as a dynamic peer VPN configuration. However, with a dynamic peer VPN configuration, the security device admin must know the internal IP address space of the dialup user, so that the admin can use it as the destination address when configuring an outgoing policy (see “Policy-Based Dialup VPN, Dynamic Peer” on page 901). With a dialup VPN user configuration, the admin at the LAN site does not need to know the internal address space of the dialup user. The security device protecting the LAN uses the predefined address “Dial-Up VPN” as the source address in the incoming policy and the destination in the outgoing policy.

The ability to create bidirectional policies for a dialup VPN tunnel allows traffic to originate from the LAN end of the VPN connection after the connection has been established. (The remote end must first initiate the tunnel creation.) Note that unlike a dialup dynamic peer VPN tunnel, this feature requires that the services on the incoming and outgoing policies be identical.

NOTE: ScreenOS does not support service groups and address groups in bidirectional policies that reference a dialup VPN configuration.

The internal address space of two or more concurrently connected dialup VPN users might overlap. For example, dialup users A and B might both have an internal IP address space of 10.2.2.0/24. If that happens, the security device sends all outbound VPN traffic to both user A and user B through the VPN referenced in the first policy it finds in the policy list. For example, if the outbound policy referencing the VPN to user A appears first in the policy list, then the security device sends all outbound VPN traffic intended for users A and B to user A.

Similarly, the internal address of a dialup user might happen to overlap an address in any other policy—whether or not that other policy references a VPN tunnel. If that occurs, the security device applies the first policy that matches the basic traffic attributes of source address, destination address, source port number, destination port number, service. To avoid a bidirectional dialup VPN policy with a dynamically derived address superseding another policy with a static address, Juniper Networks recommends positioning the bidirectional dialup VPN policy lower in the policy list.

In this example, you configure bidirectional policies for a dialup AutoKey IKE VPN tunnel named VPN_*dial* for IKE user *dialup-j* with IKE ID *jf@ns.com*. For Phase 1 negotiations, you use the proposal *pre-g2-3des-sha,* with the preshared key *Jf11d7uU*. You select the predefined "Compatible" set of proposals for Phase 2 negotiations.

The IKE user initiates a VPN connection to the security device from the Untrust zone to reach corporate servers in the Trust zone. After the IKE user establishes the VPN connection, traffic can initiate from either end of the tunnel.

The Trust zone interface is ethernet1, has IP address 10.1.1.1/24, and is in NAT mode. The Untrust zone interface is ethernet3 and has IP address 1.1.1.1/24. The default route points to the external router at 1.1.1.250.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Objects**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: trust_net
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.0/24
   Zone: Trust

   Policy > Policy Elements > Users > Local > New: Enter the following, then click **OK**:

   User Name: dialup-j
   Status: Enable
   IKE User: (select)
   Simple Identity: (select); jf@ns.com

3. **VPN**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: dialup1
   Security Level: Custom
   Remote Gateway Type:
   Dialup User: (select); dialup-j
   Preshared Key: Jf11d7uU

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

   Security Level: Custom
   Phase 1 Proposal (For Custom Security Level): pre-g2-3des-sha
   Mode (Initiator): Aggressive

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: VPN_dial
   Security Level: Compatible
   Remote Gateway:
   Create a Simple Gateway: (select)
   Gateway Name: dialup1
   Type:
   Dialup User: (select); dialup-j
   Preshared Key: Jf11d7uU
   Security Level: Compatible
   Outgoing Interface: ethernet3

4. **Route**

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 0.0.0.0/0
   Gateway: (select)

Interface: ethernet1
Gateway IP Address: 1.1.1.250

5. **Policies**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Dial-Up VPN
Destination Address:
Address Book Entry: (select), trust_net
Service: ANY
Action: Tunnel
VPN Tunnel: VPN_dial
Modify matching bidirectional VPN policy: (select)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Objects**

```
set address trust trust_net 10.1.1.0/24
set user dialup-j ike-id u-fqdn jf@ns.com
```

3. **VPN**

```
set ike gateway dialup1 dialup dialup-j aggressive outgoing-interface ethernet3
preshare Jf11d7uU proposal pre-g2-3des-sha
set vpn VPN_dial gateway dialup1 sec-level compatible
```

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

5. **Policies**

```
set policy from untrust to trust “Dial-Up VPN” trust_net any tunnel vpn VPN_dial
set policy from trust to untrust trust_net “Dial-Up VPN” any tunnel vpn VPN_dial
save
```

NetScreen-Remote Security Policy Editor

1. Click **Options > Secure > Specified Connections**.
2. Click **Add a new connection**, and type **Corp** next to the new connection icon that appears.

3. Configure the connection options:

   Connection Security: Secure
   Remote Party Identity and Addressing
   ID Type: IP Subnet
   Subnet: 10.1.1.0
   Mask: 255.255.255.0
   Protocol: All
   Connect using Secure Gateway Tunnel: (select)
   ID Type: IP Address, 1.1.1.1

4. Click the **PLUS** symbol, located to the left of the UNIX icon, to expand the connection policy.

5. Click **My Identity**: Do either of the following:

   Click **Pre-shared Key > Enter Key**: Type **Jf11d7uU**, then click **OK**.

   ID Type: (select **E-mail Address**), and type **jf@ns.com**.

6. Click the **Security Policy** icon, then select **Aggressive Mode** and clear **Enable Perfect Forward Secrecy (PFS)**.

7. Click the **PLUS** symbol, located to the left of the Security Policy icon, and then the **PLUS** symbol to the left of Authentication (Phase 1) and Key Exchange (Phase 2) to expand the policy further.

8. Click **Authentication (Phase 1) > Proposal 1**: Select the following Authentication Method and Algorithms:

   Authentication Method: Pre-Shared Key

   (or)

   Authentication Method: RSA Signatures
   Encrypt Alg: Triple DES
   Hash Alg: SHA-1
   Key Group: Diffie-Hellman Group 2

9. Click **Key Exchange (Phase 2) > Proposal 1**: Select the following IPsec protocols:

   Encapsulation Protocol (ESP): (select)
   Encrypt Alg: Triple DES
   Hash Alg: SHA-1
   Encapsulation: Tunnel

10. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: MD5
    Encapsulation: Tunnel

11. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

Encapsulation Protocol (ESP): (select)
Encrypt Alg: DES
Hash Alg: SHA-1
Encapsulation: Tunnel

12. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: MD5
    Encapsulation: Tunnel

13. Click **File > Save Changes**.

Group IKE ID

Some organizations have many dialup VPN users. For example, a sales department might have hundreds of users, many of whom require secure dialup communication when offsite. With so many users, it is impractical to create a separate user definition, dialup VPN configuration, and policy for each one.

To avoid this difficulty, the Group IKE ID method makes one user definition available for multiple users. The Group IKE ID user definition applies to all users having certificates with specified values in the distinguished name (DN) field or to all users whose full IKE ID and preshared key on their VPN client match a partial IKE ID and preshared key on the security device. In this release of ScreenOS, Group IKE ID (both with certificates and with preshared keys) is also supported for IKEv2.

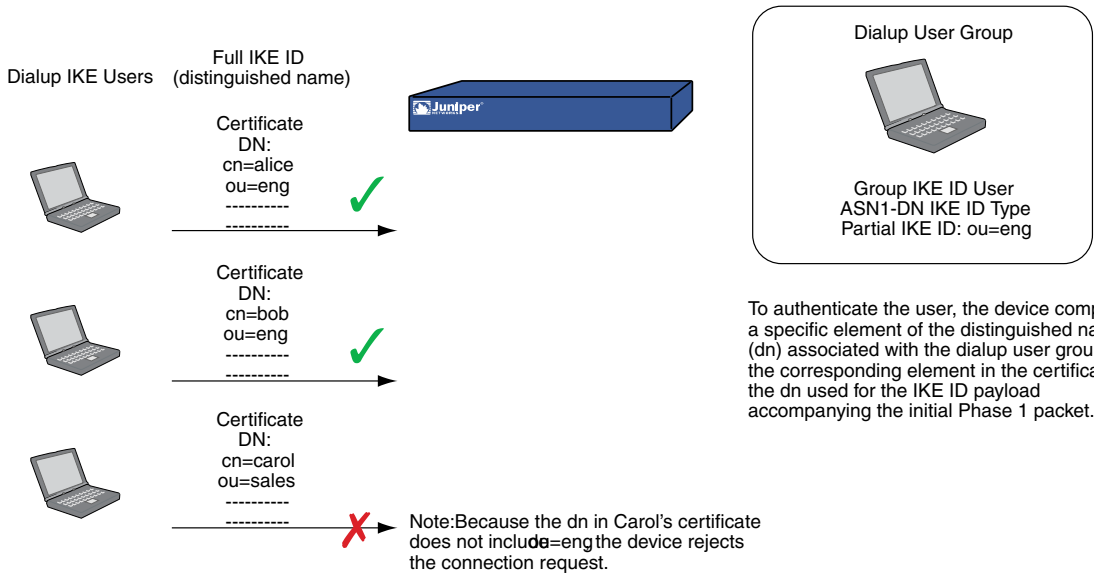
NOTE: When a dialup IKE user connects to the security device, the security device first extracts and uses the full IKE ID to search its peer gateway records in case the user does not belong to a Group IKE ID user group. If the full IKE ID search produces no matching entry, the security device then checks for a partial IKE ID match between the incoming embedded IKE ID and a configured Group IKE ID user.

You add a single Group IKE ID user to an IKE dialup VPN user group and specify the maximum number of concurrent connections that the group supports. The maximum number of concurrent sessions cannot exceed the maximum number of allowed Phase 1 SAs or the maximum number of VPN tunnels allowed on the platform.

Group IKE ID with Certificates

Group IKE ID with certificates is a technique for performing IKE authentication for a group of dialup IKE users without configuring a separate user profile for each one. Instead, the security device uses a single Group IKE ID user profile that contains a partial IKE ID. A dialup IKE user can successfully build a VPN tunnel to a security device if the VPN configuration on his VPN client specifies a certificate that contains distinguished name elements that match those configured as the partial IKE ID definition in the Group IKE ID user profile on the security device.

Figure 238: Group IKE ID with Certificates



You can set up Group IKE ID with certificates as follows:

On the Security Device:

1. Create a new Group IKE ID user with a partial IKE identity (such as *ou = sales,o = netscreen*), and specify how many dialup users can use the Group IKE ID profile to log on.
2. Assign the new Group IKE ID user to a dialup user group, and name the group.

NOTE: You can put only one Group IKE ID user in an IKE user group.

---

3. In the dialup AutoKey IKE VPN configuration, specify the name of the dialup user group, that the Phase 1 negotiations be in aggressive mode and that certificates (RSA or DSA, depending on the type of certificate loaded on the dialup VPN clients) be used for authentication.
4. Create a policy permitting inbound traffic through the specified dialup VPN.

On the VPN Client:

1. Obtain and load a certificate whose distinguished name contains the same information as defined in the partial IKE ID on the security device.
2. Configure a VPN tunnel to the security device using aggressive mode for Phase 1 negotiations, specify the certificate that you have previously loaded, and select *Distinguished Name* for the local IKE ID type.

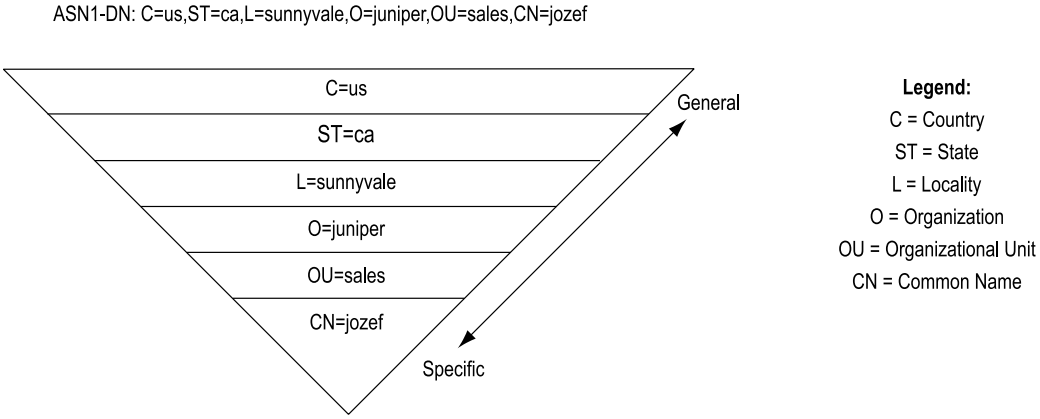
Thereafter, each individual dialup IKE user with a certificate with distinguished name elements that match the partial IKE ID defined in the Group IKE ID user profile can successfully build a VPN tunnel to the security device. For example, if the Group IKE

ID user has IKE ID *OU = sales,O = netscreen,* the security device accepts Phase 1 negotiations from any user with a certificate containing those elements in its distinguished name. The maximum number of such dialup IKE users that can connect to the security device depends upon the maximum number of concurrent sessions that you specify in the Group IKE ID user profile.

Wildcard and Container ASN1-DN IKE ID Types

When you define the IKE ID for a Group IKE user, you must use the Abstract Syntax Notation, version 1, distinguished name (ASN1-DN) as the IKE ID type of identity configuration. This notation is a string of values, which is frequently although not always ordered from general to specific. See Figure 239 on page 913 for an example.

Figure 239: ASN1 Distinguished Name



When configuring the Group IKE ID user, you must specify the peer's ASN1-DN ID as one of two types:

- **Wildcard:** ScreenOS authenticates a dialup IKE user's ID if the values in the dialup IKE user's ASN1-DN identity fields match those in the Group IKE user's ASN1-DN identity fields. The wildcard ID type supports only one value per identity field (for example, "ou = eng" or "ou = sw" but not "ou = eng,ou = sw"). The ordering of the identity fields in the two ASN1-DN strings is inconsequential.
- **Container:** ScreenOS authenticates a dialup IKE user's ID if the values in the dialup IKE user's ASN1-DN identity fields exactly match the values in the Group IKE user's ASN1-DN identity fields. The container ID type supports multiple entries for each identity field (for example, "ou = eng,ou = sw,ou = screenos"). The ordering of the values in the identity fields of the two ASN1-DN strings must be identical.

When configuring an ASN1-DN ID for a remote IKE user, specify the type as either "wildcard" or "container" and define the ASN1-DN ID that you expect to receive in the peer's certificate (for example, "c = us,st = ca,cn = kgreen"). When configuring an ASN1-DN ID for a local IKE ID, use the following keyword: [DistinguishedName]. Include the brackets and spell it exactly as shown.

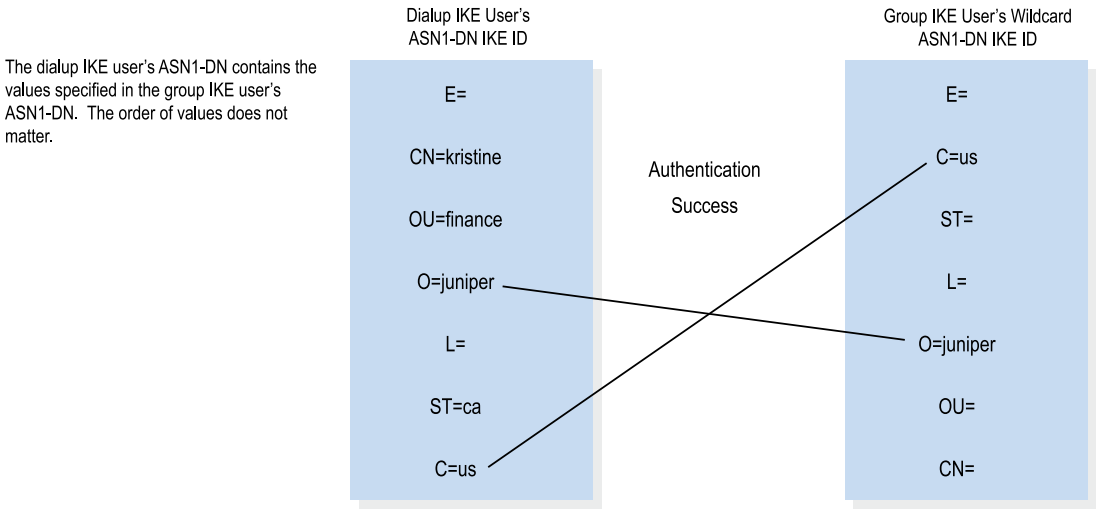
Wildcard ASN1-DN IKE ID

A wildcard ASN1-DN requires values in the remote peer's distinguished name IKE ID to match values in the Group IKE user's partial ASN1-DN IKE ID. The sequencing of these values in the ASN1-DN string is inconsequential. For example, if the dialup IKE user's ID and the Group IKE user's ID are as follows:

- Dialup IKE user's full ASN1-DN IKE ID:
  CN = kristine,OU = finance,**O = juniper**,ST = ca,**C = us**
- Group IKE user's partial ASN1-DN IKE ID: **C = us**,**O = juniper**

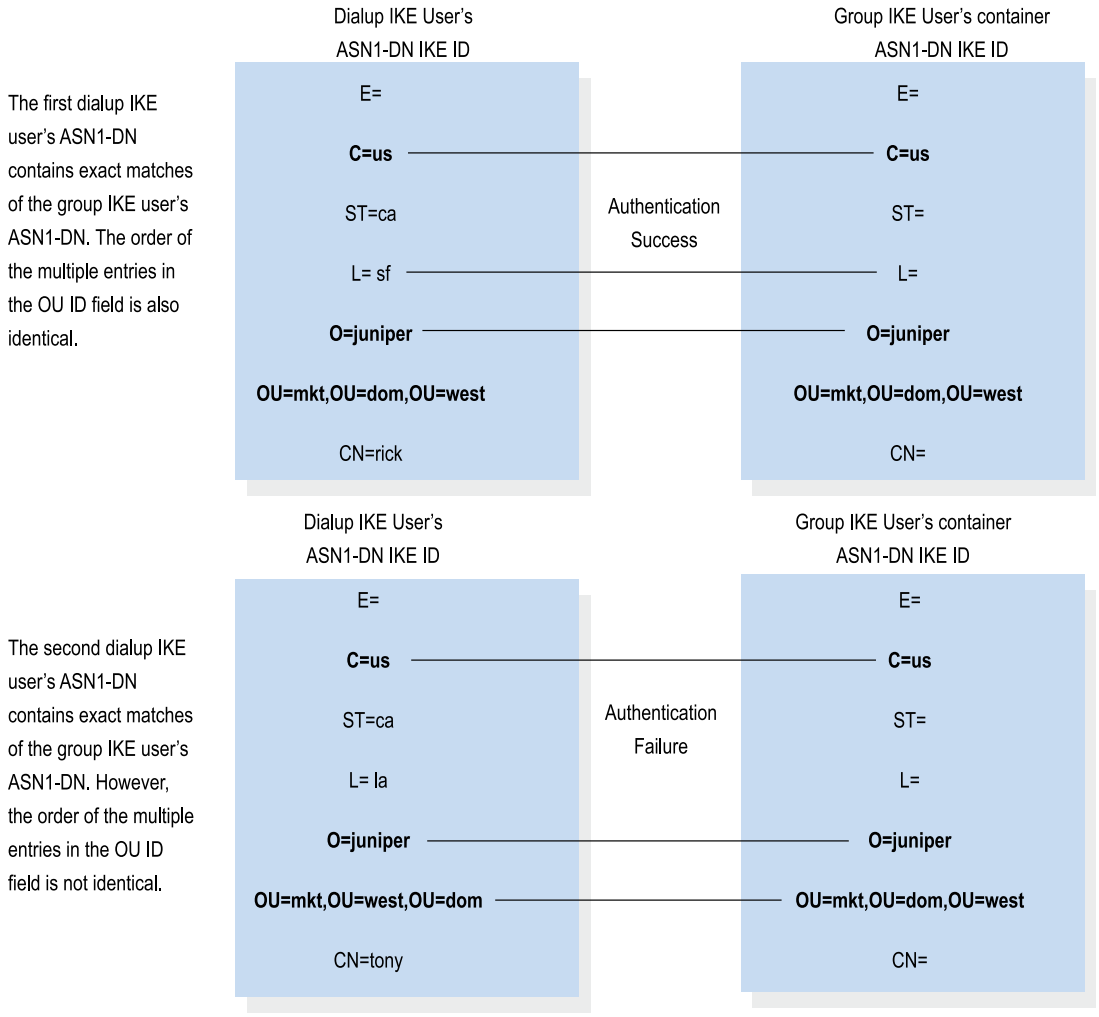
then a wildcard ASN1-DN IKE ID successfully matches the two IKE IDs, even though the order of values in the two IDs is different.

Figure 240: Successful Wildcard ASN1-DN Authentication



A container ASN1-DN ID allows the Group IKE user's ID to have multiple entries in each identity field. ScreenOS authenticates a dialup IKE user if the dialup user's ID contains values that exactly match the values in the Group IKE user's ID. Unlike the wildcard type, the order of the ASN1-DN fields must be identical in both the dialup IKE user's and Group IKE user's IDs and the order of multiple values in those fields must be identical.

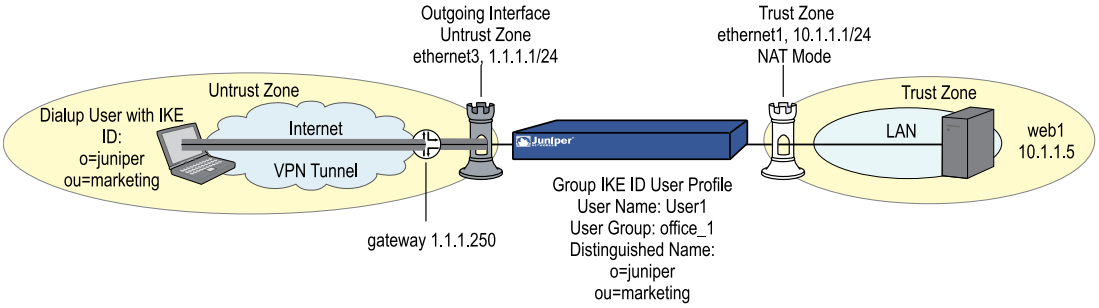
Figure 241: Authentication Success and Failure Using Container ASN1-DN IDs



Creating a Group IKE ID (Certificates)

In this example, you create a new Group IKE ID user definition named User1. You configure it to accept up to 10 Phase 1 negotiations concurrently from VPN clients with RSA certificates containing *O = netscreen* and *OU = marketing*. The certificate authority (CA) is Verisign. You name the dialup IKE user group *office_1*.

Figure 242: Group IKE ID



The dialup IKE users send a distinguished name as their IKE ID. The distinguished name (dn) in a certificate for a dialup IKE user in this group might appear as the following concatenated string:

C=us,ST=ca,L=sunnyvale,**O=netscreen**,**OU=marketing**,CN=carrie nowocin,CN=a2010002,CN=ns500,CN=4085557800,CN=rsa-key,CN=10.10.5.44

Because the values *O = netscreen* and *OU = marketing* appear in the peer's certificate and the user uses the distinguished name as its IKE ID type, the security device authenticates the user.

For the Phase 1 and Phase 2 security levels, you specify one Phase 1 proposal — rsa-g2-3des-sha for certificates—and select the predefined "Compatible" set of proposals for Phase 2.

You configure a dialup VPN and a policy permitting HTTP traffic through the VPN tunnel to reach the Web server *Web1*. The configuration of the remote VPN client (using NetScreen-Remote) is also included.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK:**

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: web1
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.5/32
Zone: Trust

3. **Users**

Policy > Policy Elements > Users > Local > New: Enter the following, then click **OK**:

User Name: User1
Status Enable: (select)
IKE User: (select)
Number of Multiple Logins with same ID: 10
Use Distinguished Name For ID: (select)
OU: marketing
Organization: juniper

Objects > User Groups > Local > New: Type **office_1** in the Group Name field, do the following, then click **OK**:

Select **User1** and use the **<<** button to move her from the Available Members column to the Group Members column.

4. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: Corp_GW
Security Level: Custom
Remote Gateway Type: Dialup User Group: (select), Group: office_1
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (For Custom Security Level): rsa-g2-3des-sha
Mode (Initiator): Aggressive
Preferred Certificate (optional):
Peer CA: Verisign
Peer Type: X509-SIG

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: Corp_VPN
Security Level: Compatible
Remote Gateway: Predefined: (select), Corp_GW

5. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

6. **Policy**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Dial-Up VPN
Destination Address:
Address Book Entry: (select), web1
Service: HTTP
Action: Tunnel
Tunnel VPN: Corp_VPN
Modify matching bidirectional VPN policy: (clear)
Position at Top: (select)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Address**

```
set address trust web1 10.1.1.5/32
```

3. **Users**

```
set user User1 ike-id asn1-dn wildcard o=juniper,ou=marketing share-limit 10
set user-group office_1 user User1
```

4. **VPN**

```
set ike gateway Corp_GW dialup office_1 aggressive outgoing-interface ethernet3
 proposal rsa-g2-3des-sha
set ike gateway Corp_GW cert peer-ca 1
set ike gateway Corp_GW cert peer-cert-type x509-sig
set vpn Corp_VPN gateway Corp_GW sec-level compatible
```

NOTE: The number 1 is the CA ID number. To discover the CA's ID number, use the following command: **get pki x509 list ca-cert**.

5. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

6. **Policy**

   set policy top from untrust to trust "Dial-Up VPN" web1 http tunnel vpn Corp_VPN save

NetScreen-Remote Security Policy Editor

1. Click **Options > Secure > Specified Connections**.
2. Click **Add a new connection**, and type **web1** next to the new connection icon that appears.
3. Configure the connection options:

   Connection Security: Secure
   Remote Party Identity and Addressing
   ID Type: IP Address, 10.1.1.5
   Protocol: Highlight **All**, type **HTTP**, press the **Tab** key, and type **80**.
   Connect using Secure Gateway Tunnel: (select)
   ID Type: IP Address, 1.1.1.1

4. Click the **PLUS** symbol, located to the left of the web1 icon, to expand the connection policy.
5. Click **My Identity**: Select the certificate that has o = netscreen,ou = marketing as elements in its distinguished name from the Select Certificate drop-down list.

   ID Type: Select **Distinguished Name** from the drop-down list.

NOTE: This example assumes that you have already loaded a suitable certificate on the NetScreen-Remote client. For information about loading certificates on the NetScreen-Remote, refer to the NetScreen-Remote documentation.

6. Click the **Security Policy** icon, then select **Aggressive Mode** and clear **Enable Perfect Forward Secrecy (PFS)**.
7. Click the **PLUS** symbol, located to the left of the Security Policy icon, and then the **PLUS** symbol to the left of Authentication (Phase 1) and Key Exchange (Phase 2) to expand the policy further.
8. Click **Authentication (Phase 1) > Proposal 1**: Select the following Encryption and Data Integrity Algorithms:

   Authentication Method: RSA Signatures
   Encrypt Alg: Triple DES
   Hash Alg: SHA-1
   Key Group: Diffie-Hellman Group 2

9. Click **Key Exchange (Phase 2) > Proposal 1**: Select the following IPsec protocols:

   Encapsulation Protocol (ESP): (select)
   Encrypt Alg: Triple DES
   Hash Alg: SHA-1
   Encapsulation: Tunnel

10. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: MD5
    Encapsulation: Tunnel

11. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: SHA-1
    Encapsulation: Tunnel

12. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

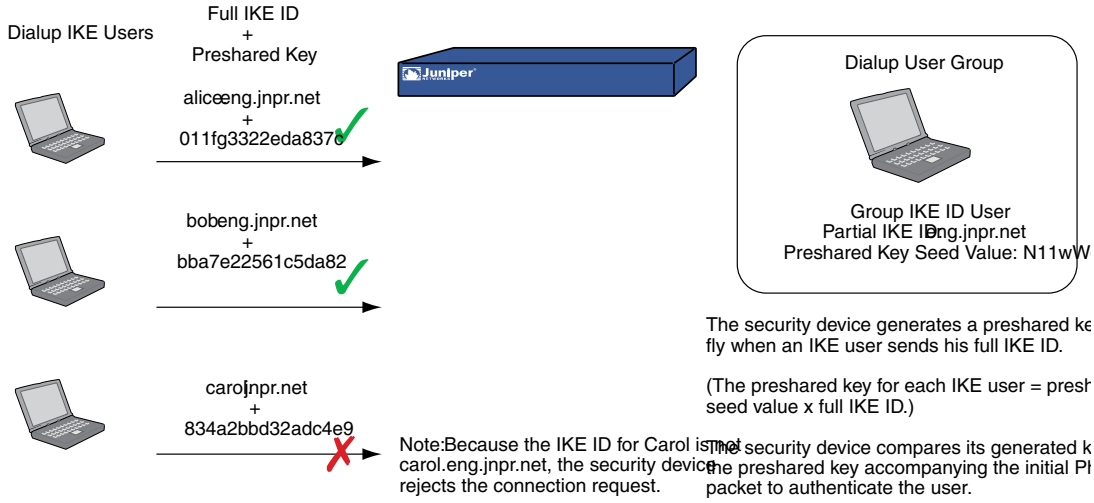
    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: MD5
    Encapsulation: Tunnel

13. Click **File > Save Changes**.

Setting a Group IKE ID with Preshared Keys

Group IKE ID with preshared keys is a technique for performing IKE authentication for a group of dialup IKE users without configuring a separate user profile for each one. Instead, the security device uses a single Group IKE ID user profile, which contains a partial IKE ID. A dialup IKE user can successfully build a VPN tunnel to a security device if the VPN configuration on his VPN client has the correct preshared key and if the rightmost part of the user's full IKE ID matches the Group IKE ID user profile's partial IKE ID.

Figure 243: Group IKE ID with Preshared Keys



The IKE ID type that you can use for the Group IKE ID with Preshared Key feature can be either an email address or a fully qualified domain name (FQDN).

You can set up Group IKE ID with preshared keys as follows:

On the Security Device:

1. Create a new Group IKE ID user with a partial IKE identity (such as **juniper.net**), and specify the number of dialup users that can use the Group IKE ID profile to log on.
2. Assign the new Group IKE ID user to a dialup user group.
3. In the dialup AutoKey IKE VPN configuration, assign a name for the remote gateway (such as **road1**), specify the dialup user group, and enter a preshared key seed value.
4. Use the following CLI command to generate an individual dialup user's preshared key using the preshared key seed value and the full user IKE ID (such as **lisa@juniper.net**)

   exec ike preshare-gen name_str usr_name_str (for example) **exec ike preshare-gen road1 lisa@juniper.net**

5. Record the preshared key for use when configuring the remote VPN client.

   On the VPN Client:

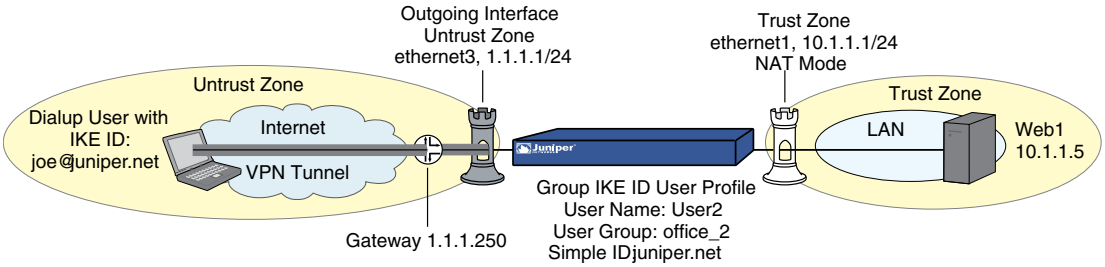
   Configure a VPN tunnel to the security device using aggressive mode for Phase 1 negotiations, and enter the preshared key that you previously generated on the security device.

Thereafter, the security device can successfully authenticate each individual user whose full IKE ID contains a section that matches the partial Group IKE ID user profile. For example, if the Group IKE ID user has IKE identity **juniper.net**, any user with that domain name in his IKE ID can initiate Phase 1 IKE negotiations in aggressive

mode with the security device. For example: **alice@juniper.net**, **bob@juniper.net,** and **carol@juniper.net**. How many such users can log on depends upon a maximum number of concurrent sessions specified in the Group IKE ID user profile.

In this example, you create a new Group IKE ID user named User2. You configure it to accept up to 10 Phase 1 negotiations concurrently from VPN clients with preshared keys containing an IKE ID ending with the string **juniper.net**. The seed value for the preshared key is **jk930k**. You name the dialup IKE user group **office_2**.

Figure 244: Group IKE ID (Preshared Keys)



For both the Phase 1 and Phase 2 negotiations, you select the security level predefined as "Compatible." All the security zones are in the trust-vr routing domain.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Address**

   Policy > Policy Elements > Addresses > List > New : Enter the following, then click **OK**:

   Address Name: web1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.5/32
   Zone: Trust

3. **Users**

Policy > Policy Elements > Users > Local > New: Enter the following, then click **OK**:

User Name: User2
Status: Enable
IKE User: (select)
Number of Multiple Logins with same ID: 10
Simple Identity: (select)
IKE Identity: juniper.net

Policy > Policy Elements > User Groups > Local > New: Type **office_2** in the Group Name field, do the following, then click **OK:**

Select **User2** and use the **<<** button to move him from the Available Members column to the Group Members column.

4. **VPN**

NOTE: The WebUI allows you to enter only a value for a preshared key, not a seed value from which the security device derives a preshared key. To enter a preshared key seed value when configuring an IKE gateway, you must use the CLI.

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: Corp_VPN
Security Level: Compatible
Remote Gateway: Predefined: (select), Corp_GW

5. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

6. **Policy**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Dial-Up VPN
Destination Address:
Address Book Entry: (select), web1
Service: HTTP
Action: Tunnel
Tunnel VPN: Corp_VPN
Modify matching bidirectional VPN policy: (clear)
Position at Top: (select)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Address**

```
set address trust web1 10.1.1.5/32
```

3. **Users**

```
set user User2 ike-id u-fqdn juniper.net share-limit 10
set user-group office_2 user User2
```

4. **VPN**

```
set ike gateway Corp_GW dialup office_2 aggressive seed-preshare jk930k
sec-level compatible
set vpn Corp_VPN gateway Corp_GW sec-level compatible
```

5. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

6. **Policy**

```
set policy top from untrust to trust “Dial-Up VPN” web1 http tunnel vpn Corp_VPN
save
```

Obtaining the Preshared Key

You can only obtain the preshared key by using the following CLI command:

exec ike preshare-gen *name_str usr_name_str*

The preshared key, based on the preshared key seed value jk930k (as specified in the configuration for the remote gateway named Corp_GW), and the full identity of individual *user heidi@juniper.net is 11ccce1d396f8f29ffa93d11257f691af96916f2*.

NetScreen-Remote Security Policy Editor

1. Click **Options > Secure > Specified Connections**.
2. Click **Add a new connection**, and type **web1** next to the new connection icon that appears.
3. Configure the connection options:

Connection Security: Secure
Remote Party Identity and Addressing
ID Type: IP Address, 10.1.1.5
Protocol: Highlight **All**, type **HTTP**, press the **Tab** key, and type **80**.
Connect using Secure Gateway Tunnel: (select)
ID Type: IP Address, 1.1.1.1

4. Click the **PLUS** symbol, located to the left of the web1 icon, to expand the connection policy.
5. Click the **Security Policy** icon, then select **Aggressive Mode** and clear **Enable Perfect Forward Secrecy (PFS)**.
6. Click **My Identity**: Click **Pre-shared Key > Enter Key**: Type **11ccce1d396f8f29ffa93d11257f691af96916f2**, then click **OK**.

   ID Type: (select **E-mail Address**), and type **heidi@juniper.net**.
7. Click the **PLUS** symbol, located to the left of the Security Policy icon, then click the **PLUS** symbol to the left of Authentication (Phase 1) and Key Exchange (Phase 2) to expand the policy further.
8. Click **Authentication (Phase 1) > Proposal 1**: Select the following Encryption and Data Integrity Algorithms:

   Authentication Method: Pre-Shared Key
   Encrypt Alg: Triple DES
   Hash Alg: SHA-1
   Key Group: Diffie-Hellman Group 2
9. Click **Authentication (Phase 1) > Create New Proposal**: Select the following IPsec protocols:

   Authentication Method: Pre-Shared Key
   Encrypt Alg: Triple DES
   Hash Alg: MD5
   Key Group: Diffie-Hellman Group 2
10. Click **Authentication (Phase 1) > Create New Proposal**: Select the following IPsec protocols:

    Authentication Method: Pre-Shared Key
    Encrypt Alg: DES
    Hash Alg: SHA-1
    Key Group: Diffie-Hellman Group 2
11. Click **Authentication (Phase 1) > Create New Proposal**: Select the following IPsec protocols:

    Authentication Method: Pre-Shared Key
    Encrypt Alg: DES
    Hash Alg: MD5
    Key Group: Diffie-Hellman Group 2
12. Click **Key Exchange (Phase 2) > Proposal 1**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES

Hash Alg: SHA-1
Encapsulation: Tunnel

13. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: MD5
    Encapsulation: Tunnel

14. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: SHA-1
    Encapsulation: Tunnel

15. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: MD5
    Encapsulation: Tunnel

16. Click **File > Save Changes**.

Shared IKE ID

The Shared IKE ID feature facilitates the deployment of a large number of dialup users. With this feature, the security device authenticates multiple dialup VPN users using a single Group IKE ID and preshared key. Thus, it provides IPsec protection for large remote user groups through a common VPN configuration. In this release of ScreenOS, Shared IKE ID is also supported for IKEv2.

This feature is similar to the Group IKE ID with preshared keys feature, with the following differences:

- With the Group IKE ID feature, the IKE ID can be an email address or a fully qualified domain name (FQDN). For this feature, the IKE ID must be an email address.
- Instead of using the preshared key seed value and the full user IKE ID to generate a preshared key for each user, you specify a single preshared key for all users in the group.
- You must use XAuth (for IKEv1) or EAP (for IKEv2) to authenticate the individual users.

To set up a Shared IKE ID and preshared key on the security device:

1. Create a new Group IKE ID user, and specify how many dialup users can use the Group IKE ID to log on. For this feature, use an email address as the IKE ID.
2. Assign the new Group IKE ID to a dialup user group.
3. In the dialup-to-LAN AutoKey IKE VPN configuration, create a Shared IKE ID gateway.
4. Define the XAuth users and enable XAuth on the remote IKE gateway.

On the VPN Client:

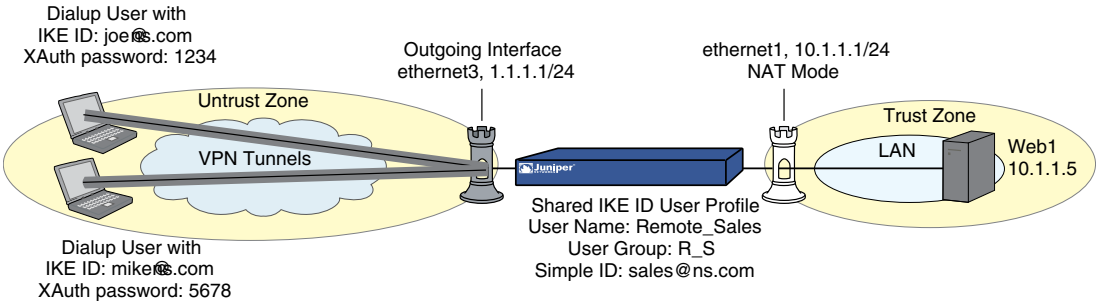
Configure a VPN tunnel to the security device using aggressive mode for Phase 1 negotiations, and enter the preshared key that you previously defined on the security device.Thereafter, the security device authenticates each remote user as follows:

During Phase 1 negotiations, the security device first authenticates the VPN client by matching the IKE ID and preshared key that the client sends with the IKE ID and preshared key on the security device. If there is a match, then the security device uses XAuth to authenticate the individual user. It sends a login prompt to the user at the remote site between Phase 1 and Phase 2 IKE negotiations. If the remote user successfully logs on with the correct username and password, Phase 2 negotiations begin.

In this example, you create a new Group IKE ID user named Remote_Sales. It accepts up to 250 Phase 1 negotiations concurrently from VPN clients with the same preshared key (abcd1234). You name the dialup IKE user group **R_S**. In addition, you configure two XAuth users, Joe and Mike.

For both the Phase 1 and Phase 2 negotiations, you select the security level predefined as Compatible. All the security zones are in the trust-vr routing domain.

Figure 245: Shared IKE ID (Preshared Keys)



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)

IP Address/Netmask: 10.1.1.1/24
Select the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: web1
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.5/32
Zone: Trust

3. **Users**

Policy > Policy Elements > Users > Local > New: Enter the following, then click **OK**:

User Name: Remote_Sales
Status: Enable
IKE User: (select)
Number of Multiple Logins with same ID: 250
Simple Identity: (select)
IKE Identity: sales@ns.com

Policy > Policy Elements > User Groups > Local > New: Type **R_S** in the Group Name field, do the following, then click **OK**:

Select **Remote_sales** and use the **<<** button to move him from the Available Members column to the Group Members column.

Policy > Policy Elements > Users > Local > New: Enter the following, then click **OK**:

User Name: Joe
Status: Enable
XAuth User: (select)
Password: 1234
Confirm Password: 1234

Policy > Policy Elements > Users > Local > New: Enter the following, then click **OK**:

User Name: Mike
Status: Enable
XAuth User: (select)
Password: 5678
Confirm Password: 5678

4. **VPN**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: sales_gateway
   Security Level: Compatible (select)
   Remote Gateway Type: Dialup Group (select), R_S
   Preshared Key: abcd1234
   Outgoing Interface: ethernet3

   > Advanced: Enter the following, then click **Return** to return to the base Gateway configuration page:

   Enable XAuth: (select)
   Local Authentication: (select)
   Allow Any: (select)

   VPNs > AutoKey IKE > New: Enter the following, then click **OK:**

   VPN Name: Sales_VPN
   Security Level: Compatible
   Remote Gateway: Predefined: (select) sales_gateway

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

   Bind to: Tunnel Zone, Untrust-Tun

5. **Route**

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   Network Address / Netmask: 0.0.0.0/0
   Gateway: (select)
      Interface: ethernet3
      Gateway IP Address: 1.1.1.250

6. **Policy**

   Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

   Source Address:
      Address Book Entry: (select), Dial-Up VPN
   Destination Address:
      Address Book Entry: (select), web1
   Service: HTTP
   Action: Tunnel
   Tunnel VPN: Sales_VPN
   Modify matching bidirectional VPN policy: (clear)
   Position at Top: (select)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Address**

```
set address trust web1 10.1.1.5/32
```

3. **Users**

```
set user Remote_Sales ike-id sales@ns.com share-limit 250
set user-group R_S user Remote_Sales
set user Joe password 1234
set user Joe type xauth
set user Mike password 5678
set user Mike type xauth
```

4. **VPN**

```
set ike gateway sales_gateway dialup R_S aggressive outgoing-interface ethernet3
 preshare abcd1234 sec-level compatible
set ike gateway sales_gateway xauth
set vpn sales_vpn gateway sales_gateway sec-level compatible
set vpn sales_vpn bind zone untrust-tun
```

5. **Route**

```
set route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

6. **Policy**

```
set policy top from untrust to trust “Dial-Up VPN” web1 http tunnel vpn sales_vpn
save
```

NetScreen-Remote Security Policy Editor

This example shows the configuration for the user named Joe.

1. Click **Options > Secure > Specified Connections**.
2. Click **Add a new connection**, and type **web1** next to the new connection icon that appears.
3. Configure the connection options:

```
Connection Security: Secure
Remote Party ID Type: IP Address
IP Address: 10.1.1.5
```

Connect using Secure Gateway Tunnel: (select)
ID Type: IP Address; 1.1.1.1

4. Click the **PLUS** symbol, located to the left of the web1 icon, to expand the connection policy.
5. Click the **Security Policy** icon, then select **Aggressive Mode** and clear **Enable Perfect Forward Secrecy (PFS)**.
6. Click **My Identity**: Click **Pre-shared Key > Enter Key**: Type **abcd1234**, then click **OK**.

   ID Type: (select **E-mail Address**), and type **sales@ns.com**.
7. Click the **PLUS** symbol, located to the left of the Security Policy icon, then click the **PLUS** symbol to the left of Authentication (Phase 1) and Key Exchange (Phase 2) to expand the policy further.
8. Click **Authentication (Phase 1) > Proposal 1**: Select the following Encryption and Data Integrity Algorithms:

   Authentication Method: Pre-Shared Key; Extended Authentication
   Encrypt Alg: Triple DES
   Hash Alg: SHA-1
   Key Group: Diffie-Hellman Group 2
9. Click **Authentication (Phase 1) > Create New Proposal**: Select the following IPsec protocols:

   Authentication Method: Pre-Shared Key; Extended Authentication
   Encrypt Alg: Triple DES
   Hash Alg: MD5
   Key Group: Diffie-Hellman Group 2
10. Click **Authentication (Phase 1) > Create New Proposal**: Select the following IPsec protocols:

    Authentication Method: Pre-Shared Key; Extended Authentication
    Encrypt Alg: DES
    Hash Alg: SHA-1
    Key Group: Diffie-Hellman Group 2
11. Click **Authentication (Phase 1) > Create New Proposal**: Select the following IPsec protocols:

    Authentication Method: Pre-Shared Key; Extended Authentication
    Encrypt Alg: DES
    Hash Alg: MD5
    Key Group: Diffie-Hellman Group 2
12. Click **Key Exchange (Phase 2) > Proposal 1**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: SHA-1
    Encapsulation: Tunnel

13. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: MD5
    Encapsulation: Tunnel

14. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: SHA-1
    Encapsulation: Tunnel

15. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: MD5
    Encapsulation: Tunnel

16. Click **File > Save Changes**.

Chapter 24

Layer 2 Tunneling Protocol

This chapter provides an introduction to Layer 2 Tunneling Protocol (L2TP), its use alone and with IPsec support, and some configuration examples for L2TP and L2TP-over-IPsec. It contains the following sections:

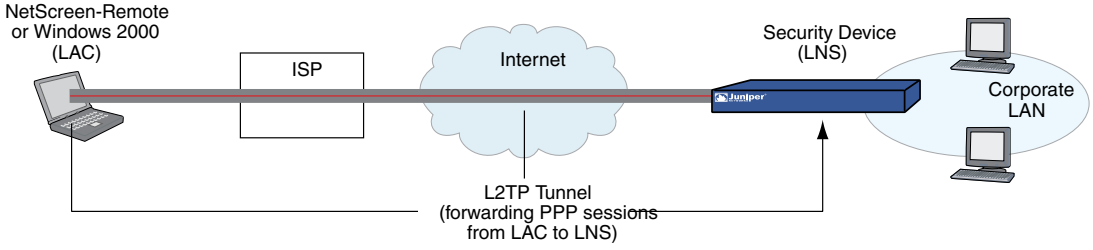
Introduction to L2TP

Layer 2 Tunneling Protocol (L2TP) provides a way for a dialup user to make a virtual Point-to-Point Protocol (PPP) connection to an L2TP network server (LNS), which can be a security device. L2TP sends PPP frames through a tunnel between an L2TP access concentrator (LAC) and the LNS.

Originally, L2TP was designed so that a LAC residing at an ISP site tunneled to an LNS at either another ISP or corporate site. The L2TP tunnel did not extend completely to the dialup user's computer, but only to the LAC at the dialup user's local ISP. (This is sometimes referred to as a compulsory L2TP configuration.)

A NetScreen-Remote client on Windows 2000 or Windows NT, or a Windows 2000 client by itself, can act as a LAC. The L2TP tunnel can extend directly to the dialup user's computer, thus providing end-to-end tunneling. (This approach is sometimes referred to as a voluntary L2TP configuration.)

Figure 246: L2TP Tunnel Between VPN Client (LAC) and Security Device (LNS)

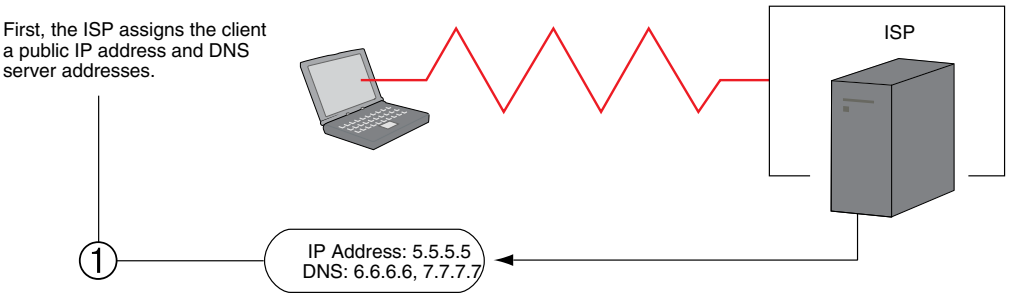


Because the PPP link extends from the dialup user across the Internet to the security device (LNS), it is the security device, not the ISP, that assigns the client its IP address,

DNS and WINS servers addresses, and authenticates the user, either from the local database or from an external auth server (RADIUS, SecurID, or LDAP).

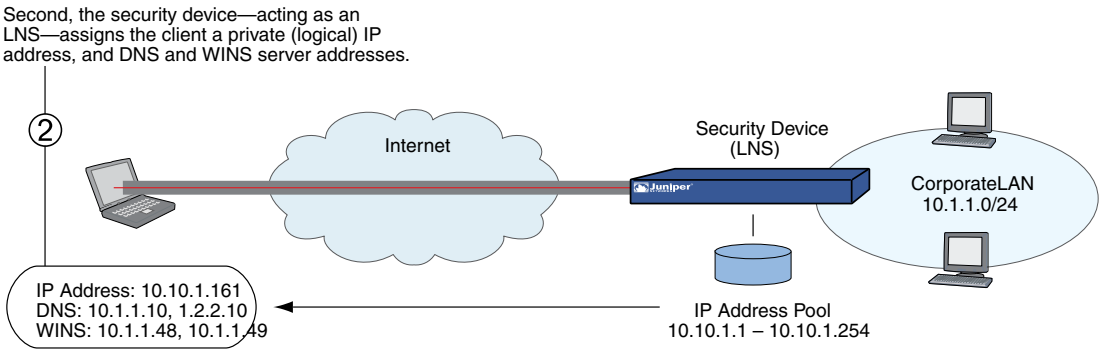
In fact, the client receives two IP addresses—one for its physical connection to the ISP, and a logical one from the LNS. When the client contacts its ISP, perhaps using PPP, the ISP makes IP and DNS assignments, and authenticates the client. This allows users to connect to the Internet with a public IP address, which becomes the outer IP address of the L2TP tunnel.

Figure 247: IP and DNS Assignments from ISP



Then, when the L2TP tunnel forwards the encapsulated PPP frames to the security device, the security device assigns the client an IP address, and DNS and WINS settings. The IP address can be from the set of private addresses not used on the Internet. This address becomes the inner IP address of the L2TP tunnel.

Figure 248: IP and DNS Assignments from LNS



NOTE: The IP addresses assigned to the L2TP client must be in a different subnet from the IP addresses in the corporate LAN.

The current version of ScreenOS provides the following L2TP support:

- L2TP tunnels originating from a host running Windows 2000

NOTE: By default, Windows 2000 performs L2TP-over-IPsec. To force it to use L2TP only, you must navigate to the ProhibitIPSec key in the registry and change **0** (L2TP-over-IPsec) to **1** (L2TP only). (Before performing this, Juniper Networks recommends that you back up your registry.) Click **Start > Run**: Type **regedit**. Double-click **HKEY_LOCAL_MACHINE > System > CurrentControlSet > Services > RasMan > Parameters**. Double-click **ProhibitIPSec**: Type **1** in the Value data field, select **Hexadecimal** as the base value, then click **OK**. Reboot. (If you do not find such an entry in the registry, see Microsoft Windows documentation for information about how to create one.)

- Combination of L2TP and IPsec in transport mode (L2TP-over-IPsec)
  - For NetScreen-Remote: L2TP-over-IPsec with main mode negotiations using certificates, and aggressive mode using either a preshared key or certificates
  - For Windows 2000: L2TP-over-IPsec with main mode negotiations using certificates
- Outgoing dialup policy for L2TP and L2TP-over-IPsec tunnels (An outgoing dialup policy can be paired with an incoming policy to provide a bidirectional tunnel.)
- User authentication using either the Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP) from the local database or an external auth server (RADIUS, SecurID, or LDAP)

NOTE: The local database and RADIUS servers support both PAP and CHAP. SecurID and LDAP servers support PAP only.

- The assignment of dialup users' IP address, Domain Name System (DNS) servers, and Windows Internet Naming Service (WINS) servers from either the local database or a RADIUS server
- L2TP tunnels and L2TP-over-IPsec tunnels for the root system and virtual systems

NOTE: To use L2TP, the security device must be operating at Layer 3, with security zone interfaces in NAT or route mode. When the security device is operating at Layer 2, with security zone interfaces in transparent mode, no L2TP-related material appears in the WebUI, and L2TP-related CLI commands elicit error messages.

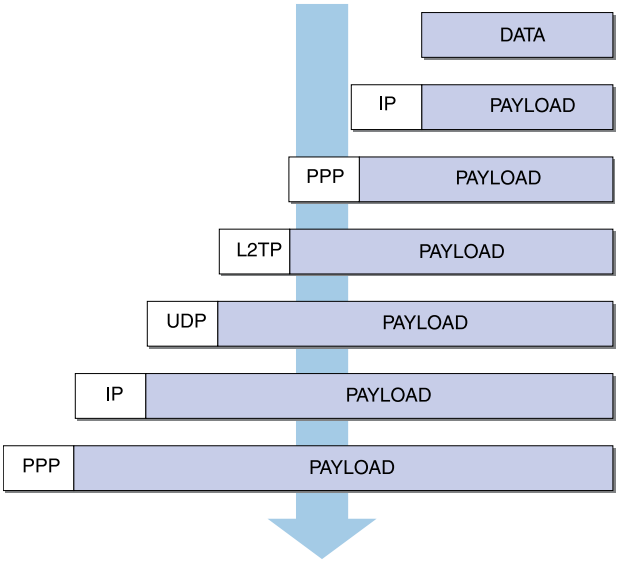
Packet Encapsulation and Decapsulation

L2TP employs encapsulation of packets as the means for transporting PPP frames from the LAC to the LNS. Before looking at specific examples for setting up L2TP and L2TP-over-IPsec, an overview of the encapsulation and decapsulation involved in the L2TP process is presented.

Encapsulation

When a dialup user on an IP network sends data over an L2TP tunnel, the LAC encapsulates the IP packet within a series of Layer 2 frames, Layer 3 packets, and Layer 4 segments. Assuming that the dialup user connects to the local ISP over a PPP link, the encapsulation proceeds as shown in Figure 249 on page 936.

Figure 249: L2TP Packet Encapsulation

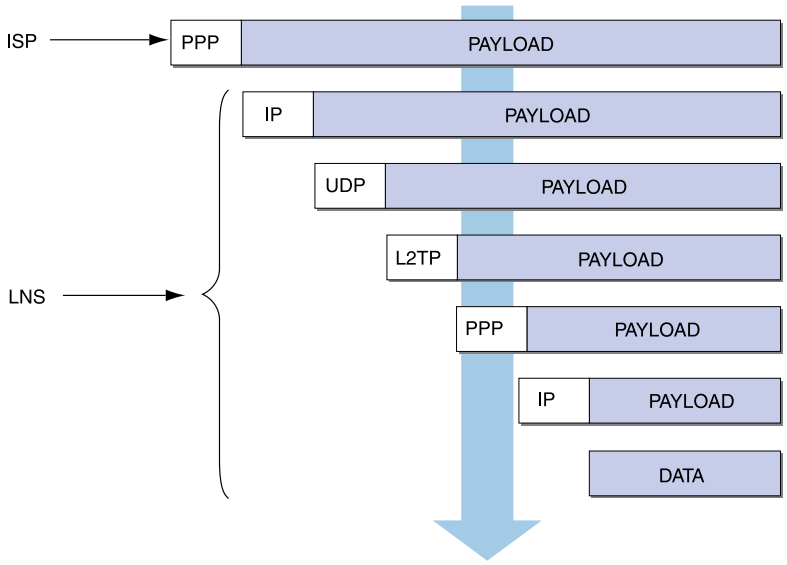


1. The data is placed in an IP payload.
2. The IP packet is encapsulated in a PPP frame.
3. The PPP frame is encapsulated in an L2TP frame.
4. The L2TP frame is encapsulated in a UDP segment.
5. The UDP segment is encapsulated in an IP packet.
6. The IP packet is encapsulated in a PPP frame to make the physical connection between the dialup user and the ISP.

Decapsulation

When the LAC initiates the PPP link to the ISP, the decapsulation and forwarding of the nested contents proceed as shown in Figure 250 on page 937.

Figure 250: L2TP Packet Decapsulation



1. The ISP completes the PPP link and assigns the user's computer an IP address.

   Inside the PPP payload is an IP packet.

2. The ISP removes the PPP header and forwards the IP packet to the LNS.
3. The LNS removes the IP header.

   Inside the IP payload is a UDP segment specifying port 1701, the port number reserved for L2TP.

4. The LNS removes the UDP header.

   Inside the UDP payload is an L2TP frame.

5. The LNS processes the L2TP frame, using the tunnel ID and call ID in the L2TP header to identify the specific L2TP tunnel. The LNS then removes the L2TP header.

   Inside the L2TP payload is a PPP frame.

6. The LNS processes the PPP frame, assigning the user's computer a logical IP address.

   Inside the PPP payload is an IP packet.

7. The LNS routes the IP packet to its ultimate destination, where the IP header is removed and the data in the IP packet is extracted.

Setting L2TP Parameters

The LNS uses L2TP to provide the PPP settings for a dialup user, which typically come from an ISP. These settings are as follows:

- IP address – The security device selects an address from a pool of IP addresses and assigns it to the dialup user's computer. The selection process operates cyclically through the IP address pool; that is, in a pool from 10.10.1.1 to 10.10.1.3, the addresses are selected in the following cycle: 10.10.1.1 – 10.10.1.2 – 10.10.1.3 – 10.10.1.1 – 10.10.1.2 …
- DNS primary and secondary server IP addresses – The security device provides these addresses to the dialup user's computer.
- WINS primary and secondary server IP addresses – The security device also provides these addresses to the dialup user's computer.

The LNS also authenticates the user through a username and password. You can enter the user in the local database or in an external auth server (RADIUS, SecurID, or LDAP).

NOTE: The RADIUS or SecurID server that you use for authenticating L2TP users can be the same server you use for network users, or it can be a different server.

In addition, you can specify one of the following schemes for the PPP authentication:

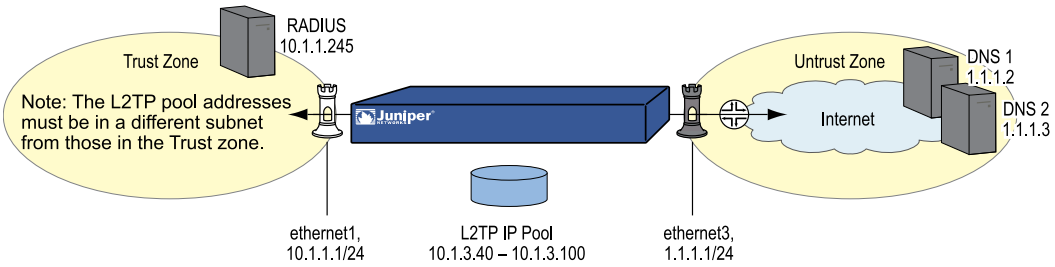
- Challenge Handshake Authentication Protocol (CHAP), in which the security device sends a challenge (encryption key) to the dialup user after he or she makes a PPP link request, and the user encrypts his or her login name and password with the key. The local database and RADIUS servers support CHAP.
- Password Authentication Protocol (PAP), which sends the dialup user's password in the clear along with the PPP link request. The local database and RADIUS, SecurID, and LDAP servers support PAP.
- "ANY", meaning that the security device negotiates CHAP, and then if that fails, PAP.

You can apply to dialup users and dialup user groups the default L2TP parameters that you configure on the L2TP Default Configuration page (VPNs > L2TP > Default Settings) or with the **set l2tp default** command. You can also apply L2TP parameters that you configure specifically for L2TP users on the User Configuration page (Users > Users > Local > New) or with the **set user** *name_str* **remote-settings** command. The user-specific L2TP settings supersede the default L2TP settings.

As shown in Figure 251 on page 939, you define an IP address pool with addresses ranging from 10.1.3.40 to 10.1.3.100. You specify DNS server IP addresses 1.1.1.2 (primary) and 1.1.1.3 (secondary). The security device performs PPP authentication using CHAP.

NOTE: You specify the auth server on a per-L2TP tunnel basis.

Figure 251: IP Pool and L2TP Default Settings



WebUI

1. **IP Pool**

   Objects > IP Pools > New: Enter the following, then click **OK**:

   IP Pool Name: Sutro
   Start IP: 10.1.3.40
   End IP: 10.1.3.100

2. **Default L2TP Settings**

   VPNs > L2TP > Default Settings: Enter the following, then click **Apply**:

   IP Pool Name: Sutro
   PPP Authentication: CHAP
   DNS Primary Server IP: 1.1.1.2
   DNS Secondary Server IP: 1.1.1.3
   WINS Primary Server IP: 0.0.0.0
   WINS Secondary Server IP: 0.0.0.0

CLI

1. **IP Pool**

   ```
   set ippool sutro 10.1.3.40 10.1.3.100
   ```

2. **Default L2TP Settings**

   ```
   set l2tp default ippool sutro
   set l2tp default ppp-auth chap
   set l2tp default dns1 1.1.1.2
   set l2tp default dns2 1.1.1.3
   save
   ```

L2TP and L2TP-over-IPsec

Although the dialup user can be authenticated using CHAP or PAP, an L2TP tunnel is not encrypted, and therefore is not a true VPN tunnel. The purpose of L2TP is

simply to permit the administrator of the local security device to assign IP addresses to remote dialup users. These addresses can then be referenced in policies.

To encrypt an L2TP tunnel, you need to apply an encryption scheme to the L2TP tunnel. Because L2TP assumes that the network between the LAC and the LNS is IP, you can employ IPsec to provide encryption. This combination is called L2TP-over-IPsec. L2TP-over-IPsec requires setting up both an L2TP tunnel and an IPsec tunnel with the same endpoints, and then linking them together in a policy. L2TP-over-IPsec requires that the IPsec tunnel be in transport mode so that the tunnel endpoint addresses remain in the clear. (For information about transport and tunnel mode, see “Modes” on page 709.)

You can create an L2TP tunnel between a security device and a host running Windows 2000 if you change the registry settings. (For instructions on how to change the registry, see the note in “Introduction to L2TP” on page 933 .)

You can create an L2TP-over-IPsec tunnel between a security device and either of the following VPN clients:

- A host running NetScreen-Remote on a Windows 2000 or Windows NT operating system
- A host running Windows 2000 (without NetScreen-Remote)

NOTE: To provide authentication when using WIndows 2000 without NetScreen-Remote, you must use certificates.

Configuring L2TP

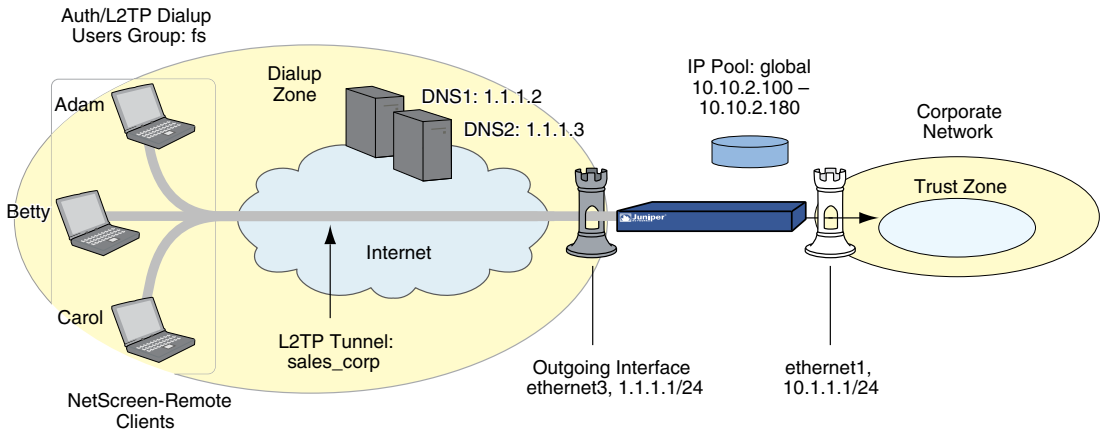
In this example, as illustrated in Figure 252 on page 941, you create a dialup user group called “fs” (for “field-sales”) and configure an L2TP tunnel called “sales_corp,” using ethernet3 (Untrust zone) as the outgoing interface for the L2TP tunnel. The security device applies the following default L2TP tunnel settings to the dialup user group:

- The L2TP users are authenticated through the local database.
- PPP authentication uses CHAP.
- The range of addresses in the IP pool (named “global”) is from 10.10.2.100 to 10.10.2.180.
- The DNS servers are 1.1.1.2 (primary) and 1.1.1.3 (secondary).

NOTE: An L2TP-only configuration is not secure. It is recommended only for debugging purposes.

The addresses in the L2TP IP pool must be in a different subnet than the addresses in the corporate network.

Figure 252: Configuring L2TP



The remote L2TP clients are on Windows 2000 operating systems. For information about how to configure L2TP on the remote clients, refer to your Windows 2000 documentation. Only the configuration for the security device end of the L2TP tunnel is provided below.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **L2TP Users**

   Objects > Users > Local > New: Enter the following, then click **OK**:

   User Name: Adam
   Status: Enable
   L2TP User: (select)
   User Password: AJbioJ15
   Confirm Password: AJbioJ15

   Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Betty
Status: Enable
L2TP User: (select)
User Password: BviPsoJ1
Confirm Password: BviPsoJ1

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Carol
Status: Enable
L2TP User: (select)
User Password: Cs10kdD3
Confirm Password: Cs10kdD3

3. **L2TP User Group**

Objects > User > Local Groups > New: Type **fs** in the Group Name field, do the following, then click **OK**:

Select **Adam** and use the < < button to move him from the Available Members column to the Group Members column.

Select **Betty** and use the < < button to move her from the Available Members column to the Group Members column.

Select **Carol** and use the < < button to move her from the Available Members column to the Group Members column.

4. **Default L2TP Settings**

Objects > IP Pools > New: Enter the following, then click **OK**:

IP Pool Name: global
Start IP: 10.10.2.100
End IP: 10.10.2.180

VPNs > L2TP > Default Settings: Enter the following, then click **OK**:

IP Pool Name: global
PPP Authentication: CHAP
DNS Primary Server IP: 1.1.1.2
DNS Secondary Server IP: 1.1.1.3
WINS Primary Server IP: 0.0.0.0
WINS Secondary Server IP: 0.0.0.0

5. **L2TP Tunnel**

VPNs > L2TP > Tunnel > New: Enter the following, then click **OK**:

Name: sales_corp
Use Custom Settings: (select)
Authentication Server: Local
Dialup Group: Local Dialup Group - fs
Outgoing Interface: ethernet3
Peer IP: 0.0.0.0
Host Name (optional): Enter the name of the computer acting as the LAC.

Secret (optional): Enter a secret shared between the LAC and the LNS.
Keep Alive: 60

Peer IP: Because the peer's ISP dynamically assigns it an IP address, you would enter **0.0.0.0** in the above example.

LAC: To find the name of a computer running Windows 2000, do the following: Click **Start > Settings > Control Panel > System**. The System Properties dialog box appears. Click the **Network Identification** tab, and see entry following **Full computer name**.

To add a secret to the LAC for authenticating the L2TP tunnel, you must modify the Windows 2000 registry as follows:

1. Click **Start > Run**, and then type **regedit**. The Registry Editor opens.
2. Click **HKEY_LOCAL_MACHINE**.
3. Right-click **SYSTEM**, and then select **Find** from the pop-up menu that appears.
4. Type **ms_l2tpminiport**, then click **Find Next**.
5. In the Edit menu, highlight **New**, and then select **String Value**.
6. Type **Password**.
7. Double-click **Password**. The Edit String dialog box appears.
8. Type the password in the Value data field. This must be the same as the word in the L2TP Tunnel Configuration Secret field on the security device.
9. Reboot the computer running Windows 2000.

   When using L2TP-over-IPsec, which is the Windows 2000 default, tunnel authentication is unnecessary; all L2TP messages are encrypted and authenticated inside IPsec.

Keep-Alive: The Keep Alive value is the number of seconds of inactivity before the security device sends an L2TP hello signal to the LAC.

6. **Route**

   Network > Routing > Routing Entries > New: Enter the following, then click OK:

   Network Address/Netmask: 0.0.0.0/0
   Gateway: (select)
   Interface: ethernet3
   Gateway IP Address: 1.1.1.250

7. **Policy**

   Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Dial-Up VPN
   Destination Address:

Address Book Entry: (select), Any
NAT: Off
Service: ANY
Action: Tunnel
Tunnel L2TP: sales_corp
Position at Top: (select)

CLI

1. **Dialup Users**

```
set user adam type l2tp
set user adam password AJbioJ15
unset user adam type auth
set user betty type l2tp
set user betty password BviPsoJ1
unset user betty type auth
set user carol type l2tp
set user carol password Cs10kdD3
unset user carol type auth
```

NOTE: Defining a password for a user automatically classifies the user as an auth user. Therefore, to define the user type strictly as L2TP, you must unset the auth user type.

2. **L2TP User Group**

```
set user-group fs location local
set user-group fs user adam
set user-group fs user betty
set user-group fs user carol
```

3. **Default L2TP Settings**

```
set ippool global 10.10.2.100 10.10.2.180
set l2tp default ippool global
set l2tp default auth server Local
set l2tp default ppp-auth chap
set l2tp default dns1 1.1.1.2
set l2tp default dns2 1.1.1.3
```

4. **L2TP Tunnel**

```
set l2tp sales_corp outgoing-interface ethernet3
set l2tp sales_corp auth server Local user-group fs
```

5. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

6. **Policy**

```
set policy top from untrust to trust “Dial-Up VPN” any tunnel l2tp sales_corp
save
```

Configuring L2TP-over-IPsec

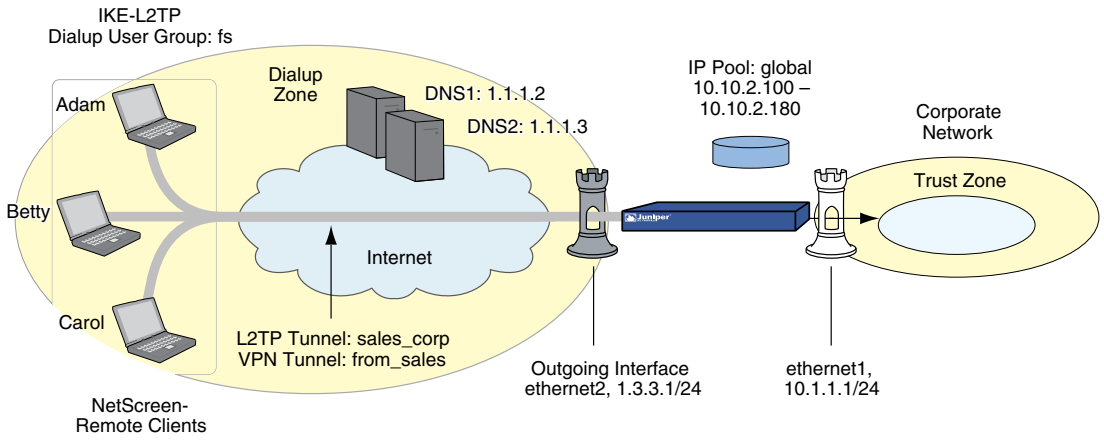
This example uses the same L2TP tunnel created in the previous example (“Configuring L2TP” on page 940). Additionally, you overlay an IPsec tunnel onto the L2TP tunnel to provide encryption. The IPsec tunnel negotiates Phase 1 in Aggressive Mode using a previously loaded RSA certificate, 3DES encryption and SHA-1 authentication. The certificate authority (CA) is Verisign. (For information about obtaining and loading certificates, see “Public Key Cryptography” on page 741.) The Phase 2 negotiation uses the security level predefined as “Compatible” for Phase 2 proposals. The IPsec tunnel is in transport mode.

The predefined Trust zone and the user-defined Dialup zone are in the trust-vr routing domain. The interfaces for the Dialup and Trust zones are ethernet2 (1.3.3.1/24) and ethernet1 (10.1.1.1/24), respectively. The Trust zone is in NAT mode.

The dialup users Adam, Betty, and Carol use NetScreen-Remote clients on a Windows 2000 operating system. The NetScreen-Remote configuration for dialup user Adam is also included below. (The NetScreen-Remote configuration for the other two dialup users is the same as that for Adam.)

NOTE: To configure an L2TP-over-IPsec tunnel for Windows 2000 (without NetScreen-Remote), the Phase 1 negotiations must be in main mode and the IKE ID type must be ASN1-DN.

Figure 253: Configuring L2TP-over-IPsec



WebUI

1. **User-Defined Zone**

Network > Zones > New: Enter the following, then click **OK**:

Zone Name: Dialup
Virtual Router Name: trust-vr
Zone Type: Layer 3 (select)
Block Intra-Zone Traffic: (select)
TCP/IP Reassembly for ALG: (clear)

NOTE: The Trust zone is preconfigured. You do not need to create it.

2. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24
Select the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: Dialup
Static IP: (select this option when present)
IP Address/Netmask: 1.3.3.1/24

3. **IKE/L2TP Users**

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Adam
Status: Enable
IKE User: (select)
  Simple Identity: (select)
    IKE Identity: ajackson@abc.com
L2TP User: (select)
  User Password: AJbioJ15
  Confirm Password: AJbioJ15

NOTE: The IKE ID that you enter must be the same as the one that the client sends, which is the email address that appears in the certificate that the client uses for authentication.

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Betty
Status: Enable
IKE User: (select)
  Simple Identity: (select)
    IKE Identity: bdavis@abc.com
L2TP User: (select)

User Password: BviPsoJ1
Confirm Password: BviPsoJ1

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Carol
Status: Enable
IKE User: (select)
Simple Identity: (select)
IKE Identity: cburnet@abc.com
L2TP User: (select)
User Password: Cs10kdD3
Confirm Password: Cs10kdD3

4. **IKE/L2TP User Group**

Objects > Users > Local Groups > New: Type **fs** in the Group Name field, do the following, then click **OK**:

Select **Adam** and use the < < button to move him from the Available Members column to the Group Members column.

Select **Betty** and use the < < button to move her from the Available Members column to the Group Members column.

Select **Carol** and use the < < button to move her from the Available Members column to the Group Members column.

5. **IP Pool**

Objects > IP Pools > New: Enter the following, then click **OK**:

IP Pool Name: global
Start IP: 10.10.2.100
End IP: 10.10.2.180

6. **Default L2TP Settings**

VPNs > L2TP > Default Settings: Enter the following, then click **Apply**:

IP Pool Name: global
PPP Authentication: CHAP
DNS Primary Server IP: 1.1.1.2
DNS Secondary Server IP: 1.1.1.3
WINS Primary Server IP: 0.0.0.0
WINS Secondary Server IP: 0.0.0.0

7. **L2TP Tunnel**

VPNs > L2TP > Tunnel > New: Enter the following, then click **OK**:

Name: sales_corp
Dialup Group: (select), Local Dialup Group - fs
Authentication Server: Local
Outgoing Interface: ethernet2
Peer IP: 0.0.0.0

Host Name (optional): If you want to restrict the L2TP tunnel to a specific host, enter the name of the computer acting as the LAC.

Secret (optional): Enter a secret shared between the LAC and the LNS.

Keep Alive: 60

LAC: To find the name of a computer running Windows 2000, do the following: Click **Start > Settings > Control Panel > System**. The System Properties dialog box appears. Click the **Network Identification** tab, and see entry following **Full computer name**.

Secret: To add a secret to the LAC for authenticating the L2TP tunnel, you must modify the Windows 2000 registry as follows:

1. Click **Start > Run**, and then type **regedit**. The Registry Editor opens.
2. Click **HKEY_LOCAL_MACHINE**.
3. Right-click **SYSTEM**, and then select **Find** from the pop-up menu that appears.
4. Type **ms_l2tpminiport**, then click **Find Next**.
5. In the Edit menu, highlight **New**, and then select **String Value**.
6. Type **Password**.
7. Double-click **Password**. The Edit String dialog box appears.
8. Type the password in the Value data field. This must be the same as the word in the L2TP Tunnel Configuration Secret field on the security device.
9. Reboot the computer running Windows 2000.

   When using L2TP-over-IPsec, which is the Windows 2000 default, tunnel authentication is unnecessary; all L2TP messages are encrypted and authenticated inside IPsec.

Keep-Alive: The Keep Alive value is the number of seconds of inactivity before the security device sends an L2TP hello signal to the LAC.

NOTE: The hostname and secret settings can usually be ignored. Only advanced users are recommended to use these settings.

8. **VPN Tunnel**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: field
   Security Level: Custom
   Remote Gateway Type:
   Dialup User Group: (select), Group: fs
   Outgoing Interface: ethernet2

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: User Defined: Custom
Phase 1 Proposal: rsa-g2-3des-sha
Mode (Initiator): Aggressive
Preferred Certificate (Optional):
Peer CA: Verisign
Peer Type: X509-SIG

NOTE: Windows 2000 (without NetScreen-Remote) supports main mode negotiations only.

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

Name: from_sales
Security Level: Compatible
Remote Gateway: Predefined: field

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Security Level: Compatible
Transport Mode: (select)

9. **Policy**

Policies > (From: Dialup, To: Trust) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), Dial-Up VPN
Destination Address:
  Address Book Entry: (select), Any
Service: ANY
Action: Tunnel
Tunnel VPN: from_sales
  Modify matching bidirectional VPN policy: (clear)
L2TP: sales_corp
Position at Top: (select)

CLI

1. **User-Defined Zone**

```
set zone name dialup
set zone dialup vrouter trust-vr
set zone dialup block
```

2. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
```

```
set interface ethernet1 nat
set interface ethernet2 zone dialup
set interface ethernet2 ip 1.3.3.1/24
```

3. **L2TP/IKE Users**

```
set user adam type ike l2tp
set user adam password AJbioJ15
unset user adam type auth
set user adam ike-id u-fqdn ajackson@abc.com
set user betty type ike l2tp
set user betty password BviPsoJ1
unset user betty type auth
set user betty ike-id u-fqdn bdavis@abc.com
set user carol type ike l2tp
set user carol password Cs10kdD3
unset user carol type auth
set user carol ike-id u-fqdn cburnet@abc.com
```

4. **IKE/L2TP User Group**

```
set user-group fs location Local
set user-group fs user adam
set user-group fs user betty
set user-group fs user carol
```

5. **IP Pool**

```
set ippool global 10.10.2.100 10.10.2.180
```

6. **Default L2TP Settings**

```
set l2tp default ippool global
set l2tp default ppp-auth chap
set l2tp default dns1 1.1.1.2
set l2tp default dns2 1.1.1.3
```

7. **L2TP Tunnel**

```
set l2tp sales_corp outgoing-interface ethernet2
set l2tp sales_corp auth server Local user-group fs
```

8. **VPN Tunnel**

```
set ike gateway field dialup fs aggressive outgoing-interface ethernet2 proposal
 rsa-g2-3des-sha
set ike gateway field cert peer-ca1
set ike gateway field cert peer-cert-type x509-sig
set vpn from_sales gateway field transport sec-level compatible
```

NOTE: Windows 2000 (without NetScreen-Remote) supports main mode negotiations only.

The number 1 is the CA ID number. To discover the CA's ID number, use the following command: **get pki x509 list ca-cert**.

9. **Policy**

```
set policy top from dialup to trust "Dial-Up VPN" any any tunnel vpn from_sales
 l2tp sales_corp
save
```

NetScreen-Remote Security Policy Editor (Adam)

To configure L2TP-over-IPsec tunnels for Betty and Carol's NetScreen-Remote clients, follow the same procedure as that provided here for Adam.

1. Click **Options > Secure > Specified Connections**.
2. Click **Add a new connection**, and type **AJ** next to the new connection icon that appears.
3. Configure the connection options:

   Connection Security: Secure
   Remote Party ID Type: IP Address
   IP Address: 1.3.3.1
   Protocol: UDP
   Port: L2TP
   Connect using Secure Gateway Tunnel: (clear)

4. Click the **PLUS** symbol, located to the left of the AJ icon, to expand the connection policy.
5. Click **My Identity**, and configure the following:

   Select the certificate with the email address specified as the user's IKE ID on the security device from the Select Certificate drop-down list:

   ID Type: **E-mail Address**
   Port: L2TP

NOTE: The email address from the certificate appears in the identifier field automatically.

6. Click the **Security Policy** icon, and select **Aggressive Mode**.
7. Click the **PLUS** symbol, located to the left of the Security Policy icon, and then the **PLUS** symbol to the left of Authentication (Phase 1) and Key Exchange (Phase 2) to expand the policy further.

8. Click **Authentication (Phase 1) > Proposal 1**: Select the following Authentication Method and Algorithms:

   Authentication Method: Pre-Shared Key
   (or)
   Authentication Method: RSA Signatures
   Hash Alg: SHA-1
   Key Group: Diffie-Hellman Group 2

9. Click **Key Exchange (Phase 2) > Proposal 1**: Select the following IPsec protocols:

   Encapsulation Protocol (ESP): (select)
   Encrypt Alg: Triple DES
   Hash Alg: SHA-1
   Encapsulation: Transport

10. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: MD5
    Encapsulation: Transport

11. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: SHA-1
    Encapsulation: Transport

12. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: MD5
    Encapsulation: Transport

13. Click **File > Save Changes**.

14. You also need to set up the network connection for your Windows 2000 operating system using the Network Connection Wizard.

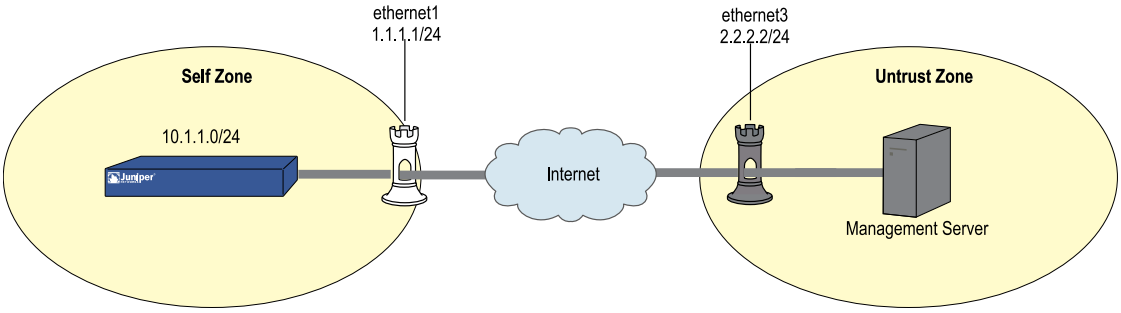
NOTE: When configuring the Network Connection Wizard, you must enter a destination hostname or IP address. Enter **1.3.3.1**. Later, when initiating a connection and are prompted for a username and password, enter adam, **AJbioJ15**. For more information, consult Microsoft Windows 2000 documentation.

Configuring an IPsec Tunnel to Secure Management Traffic

To establish secure communications for management traffic such as Web, SNMP, and Telnet, the current ScreenOS release allows the management traffic to pass through an IPsec tunnel that is not bound to L2TP or Generic Routing Encapsulation (GRE). You can create and configure a policy for an IPsec tunnel to function in transport mode and thereby enable it to carry management traffic between the security gateway and the management server.

In this example, as illustrated in Figure x on page y, you configure a VPN tunnel named **"management-vpn"** in transport mode.The outgoing interface ethernet0/1(1.1.1.1/24) is in Untrust zone, and the remote peer's IP address is 2.2.2.2/24. In this configuration, the telnet traffic matches the policy configured between the Untrust zone(1.1.1.1/24) and the management server (2.2.2.2) and successfully passes through the VPN tunnel created between the security gateway and the management server.

Figure 254: Configuring IPsec Tunnel for Management Traffic



WebUI

1. **VPN Tunnel**

   VPN > AutoKey Advanced > Gateway > New: Enter the following, then click OK:

   Gateway name: management-gw
   IPv4/v6 Address/Host name : 2.2.2.2

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

   Mode(Initiator): Aggressive
   Outgoing Interface: ethernet0/1
   Preshared Key: test
   Security Level: basic

VPN > AutoKey > New: Enter the following, then click OK:

VPN name: management–vpn
Remote gateway: Predefined: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Replay Protection: (clear)
Security Level: basic
Transport Mode: (select)

2. **Policy**

Policies > (From: Self, To:Untrust) > New: Enter the following, then click OK:

Source Address:
Address book Entry: (select), Any-IPV4
Destination Address:
New Address: 2.2.2.2/32
Service : Telnet
Action : tunnel
Tunnel VPN : management-vpn

CLI

1. **Policy**

```
set policy from self to untrust “Any” “2.2.2.2/32” telent tunnel vpn
management-vpn
save
```

2. **VPN**

```
set ike gateway management-gw  address 2.2.2.2 aggressive outgoing-interface

   ethernet0/1 preshare test  sec-level basic
set vpn management-vpn gateway management-gw no-replay transport idletime
 0
   sec-level basic
save
```

NOTE: ScreenOS allows management traffic to pass-through only a policy-based VPN established between the secured endpoints. You must specify the following when you configure the policy for management traffic:

- The source zone should be a **self** zone and the source IP address is **“Any”**.
- The destination zone should be a **MGT** zone and the destination IP address should be **“Any”**, if you configure the VPN in the management zone

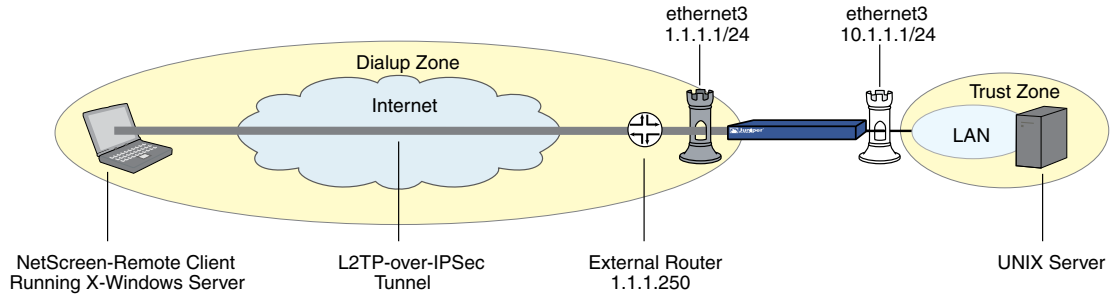
Bidirectional L2TP-over-IPsec

In this example, ethernet1 (10.1.1.1/24) is the Trust zone interface and is in NAT mode, and ethernet3 (1.1.1.1/24) is the Untrust zone interface. You create L2TP-over-IPsec tunnels between a dialup user and a corporate LAN. The remote user is running an X-Windows application, which requires bidirectional policies.

You configure incoming and outgoing policies for the dialup AutoKey IKE VPN tunnel named VPN_*dial* for IKE user *dialup-j* with IKE ID *jf@ns.com.*, and the L2TP tunnel named tun1. The IKE user initiates a IPsec connection to the security device from the Untrust zone to reach corporate servers in the Trust zone. At this point, only L2TP communication is allowed. After L2TP/PPP negotiation, the L2TP tunnel is established. With bidirectional policies configured, traffic can initiate from either end of the tunnel.

The dialup user *dialup-j* uses a NetScreen-Remote client on a Windows 2000 operating system. The NetScreen-Remote configuration for dialup user *dialup-j* is included after Figure 255 on page 955.

Figure 255: Bidirectional L2TP-over-IPsec



NOTE: To configure an L2TP-over-IPsec tunnel for Windows 2000 (without NetScreen-Remote), the Phase 1 negotiations must be in main mode and the IKE ID type must be ASN1-DN.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: trust_net
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

3. **L2TP/IKE User**

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: dialup-j
Status: Enable
IKE User: (select)
Simple Identity: (select)
IKE Identity: jf@ns.com
Authentication User: (select)
L2TP User: (select)
User Password: abc123
Confirm Password: abc123

NOTE: The IKE ID that you enter must be the same as the one that the NetScreen-Remote client sends, which is the email address that appears in the certificate that the client uses for authentication.

4. **L2TP**

VPNs > L2TP > Tunnel > New: Enter the following, then click **OK**:

Name: tun1
Use Default Settings: (select)
Secret: netscreen
Keepalive: 60

5. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click OK:

Gateway Name: dialup1
Security Level: (select), Standard
Remote Gateway Type: Dialup User; (select), dialup-j
Preshared Key: n3TsCr33N
Outgoing Interface: (select), ethernet3

> Advanced: Enter the following, and then click **Return** to return to the basic AutoKey IKE Gateway configuration page:

Mode (Initiator): Aggressive
Enable NAT-Traversal: (select)
UDP Checksum: (select)
Keepalive Frequency: 5

VPNs > AutoKey IKE > New: Enter the following, then click OK:

VPN Name: VPN_dial
Remote Gateway: Predefined: (select), dialup1

> Advanced: Enter the following, and then click **Return** to return to the basic AutoKey IKE configuration page:

Security Level: Standard (select)
Transport Mode (For L2TP-over-IPsec only): (select)

6. **Route**

Network > Routing > Routing Entries > New: Enter the following, then click OK:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
  Interface: ethernet3
  Gateway IP Address: 1.1.1.250

7. **Policies**

Policies > (From: Untrust, To: Trust) > New: Enter the following, then click OK:

Source Address:
  Address Book Entry: (select), Dial-Up VPN
Destination Address:
  Address Book Entry: (select), trust_net
Service: ANY
Action: Tunnel
Tunnel VPN: VPN_dial
  Modify matching bidirectional VPN policy: (select)
L2tp: (select) tun1

Policies > (From: Trust, To: Untrust) > New: Enter the following, then click OK:

Source Address:
  Address Book Entry: (select), trust_net
Destination Address:
  Address Book Entry: (select), Dial-Up VPN
Service: ANY
Action: Tunnel
Tunnel VPN: VPN_dial
  Modify matching bidirectional VPN policy: (select)
L2TP: tun1

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Address**

```
set address trust trust_net 10.1.1.0/24
```

3. **L2TP/IKE User**

```
set user dialup-j ike-id u-fqdn jf@ns.com
set user dialup-j type auth ike l2tp
set user dialup-j password abc123
```

4. **L2TP**

```
set L2TP tun1 outgoing-interface ethernet3 secret "netscreen" keepalive 60
```

5. **VPN**

```
set ike gateway dialup1 dialup "dialup-j" aggressive outgoing-interface ethernet3
 preshare n3TsCr33N sec-level standard
set ike gateway dialup1 nat-traversal udp-checksum
set ike gateway dialup1 nat-traversal keepalive-frequency 5
set vpn VPN_dial gateway dialup1 no-replay transport idletime 0 sec-level standard
```

6. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

7. **Policies**

```
set policy from untrust to trust “Dial-Up VPN” “trust_net” any tunnel vpn VPN_dial
 tun1
set policy from trust to untrust trust_net “Dial-Up VPN” any tunnel vpn VPN_dial
 l2tp tun1
save
```

NetScreen-Remote Security Policy Editor (for User “dialup-j”)

1. Click **Options > Secure > Specified Connections**.
2. Click **Add a new connection**, and type **dialup-j** next to the new connection icon that appears.
3. Configure the connection options:

Connection Security: Secure
Remote Party ID Type: IP Address
IP Address: 1.1.1.1
Protocol: UDP
Port: L2TP
Connect using Secure Gateway Tunnel: (clear)

4. Click the **PLUS** symbol, located to the left of the dialup-j icon, to expand the connection policy.
5. Click **My Identity**, and configure the following:

   Select the certificate with the email address specified as the user's IKE ID on the security device from the Select Certificate drop-down list

   ID Type: **E-mail Address**
   Port: L2TP

NOTE: The email address from the certificate appears in the identifier field automatically.

6. Click the **Security Policy** icon, and select **Aggressive Mode**.
7. Click the **PLUS** symbol, located to the left of the Security Policy icon, and then the **PLUS** symbol to the left of Authentication (Phase 1) and Key Exchange (Phase 2) to expand the policy further.
8. Click **Authentication (Phase 1) > Proposal 1**: Select the following Authentication method and algorithms:

   Authentication Method: Pre-Shared Key
   (or)
   Authentication Method: RSA Signatures
   Hash Alg: SHA-1
   Key Group: Diffie-Hellman Group 2

NOTE: When Perfect Forwarding Secrecy (PFS) is enabled on the security device (DF group 1,2, or 5), it must also be enabled for the VPN client in NetScreen-Remote.

9. Click **Key Exchange (Phase 2) > Proposal 1**: Select the following IPsec protocols:

   Encapsulation Protocol (ESP): (select)
   Encrypt Alg: Triple DES
   Hash Alg: SHA-1
   Encapsulation: Transport

10. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: Triple DES
    Hash Alg: MD5
    Encapsulation: Transport

11. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: SHA-1
    Encapsulation: Transport

12. Click **Key Exchange (Phase 2) > Create New Proposal**: Select the following IPsec protocols:

    Encapsulation Protocol (ESP): (select)
    Encrypt Alg: DES
    Hash Alg: MD5
    Encapsulation: Transport

13. Click **File > Save Changes**.

You also need to set up the network connection for your Windows 2000 operating system using the Network Connection Wizard.

NOTE: When you configure the Network Connection Wizard, you must enter a destination hostname or IP address. Enter **1.1.1.1**. Later, when you initiate a connection and are prompted for a username and password, enter **dialup-j, abc123**. For more information, consult your Microsoft Windows 2000 documentation.

Chapter 25

Advanced Virtual Private Network Features

This chapter covers the following uses of virtual private network (VPN) technology:

NAT-Traversal

Network Address Translation (NAT) and Network Address Port Translation (NAPT) are Internet standards that allow a local-area network (LAN) to use one set of IP addresses for internal traffic and a second set of addresses for external traffic. NAT devices generate these external addresses from predetermined pools of IP addresses.

When setting up an IPsec tunnel, the presence of a NAT device along the data path has no effect on Phase 1 and Phase 2 IKE negotiations, which always encapsulate IKE packets within User Datagram Protocol (UDP) segments. However, after the Phase 2 negotiations complete, performing NAT on the IPsec packets causes the tunnel to fail. Of the many reasons why NAT causes disruption to IPsec, one reason is that, for the Encapsulating Security Protocol (ESP), NAT devices cannot discern the location of the Layer 4 header for port translation (because it is encrypted). For the Authentication Header (AH) protocol, NAT devices can modify the port number, but the authentication check, which includes the entire IPsec packet, fails.

NOTE: For a list of IPsec/NAT incompatibilities, refer to *draft-ietf-ipsec-nat-regts-00.txt* by Bernard Aboba.

To solve these problems, security devices and the NetScreen-Remote client (version 6.0 or later) can apply a NAT-Traversal (NAT-T) feature. NAT-T adds a layer of UDP encapsulation to IPsec packets after detecting one or more NAT devices along the

data path during Phase 1 exchanges, as prescribed in the IETF drafts *draft-ietf-ipsec-nat-t-ike-00.txt* and *draft-ietf-ipsec-udp-encaps-00.txt*, as well as in later versions of these drafts.

NOTE: NetScreen-Remote 6 and NetScreen-Remote 7 support NAT-T as described in *draft-ietf-ipsec-nat-t-ike-00.txt* and *draft-ietf-ipsec-udp-encaps-00.txt*. NetScreen-Remote 8.2 supports *draft 02*.

NAT devices can create another problem if they are also IKE/IPsec-aware and attempt to process packets with the IKE port number of 500 or the IPsec protocol numbers 50 (for ESP) and 51 (for AH). To avoid such intermediary processing of IKE packets, version 2 of the previously mentioned IETF drafts proposes the shifting (or "floating") of UDP port numbers for IKE from 500 to 4500. To avoid intermediary processing of IPsec packets, both drafts 0 and 2 insert a UDP header between the outer IP header and the ESP or AH header, thereby changing the value in the Protocol field from 50 or 51 (for ESP or AH, respectively) to 17 (for UDP). In addition, the inserted UDP header also uses port 4500. The current version of ScreenOS supports NAT-T based on *draft-ietf-ipsec-nat-t-ike-02.txt* and *draft-ietf-ipsec-udp-encaps-02.txt*, as well as version 0 of these drafts.

NOTE: ScreenOS does not support NAT-T for Manual Key tunnels nor for IPsec traffic using AH. ScreenOS only supports NAT-T for AutoKey IKE tunnels using ESP.

Probing for NAT

To check that both ends of the VPN tunnel support NAT-T, ScreenOS sends two MD-5 hashes in the vendor ID payload in the first two exchanges of Phase 1 negotiations—one hash for the title of draft 0 and one of the title for draft 2:

- "4485152d 18b6bbcd 0be8a846 9579ddcc" —which is an MD-5 hash of "draft-ietf-ipsec-nat-t-ike-00"
- "90cb8091 3ebb696e 086381b5 ec427b1f" —which is an MD-5 hash of "draft-ietf-ipsec-nat-t-ike-02"

Both peers must send and receive at least one of these values in the vendor payload ID for the NAT-T probe to continue. If they send hashes for both drafts, ScreenOS uses the NAT-T implementation for draft 2.

If the devices at each endpoint support NAT-T, they send each other NAT discovery (NAT-D) payloads in the third and fourth Phase 1 exchanges (main mode) or in the second and third exchanges (aggressive mode). The NAT discovery (NAT-D) payload is a IKE payload type for NAT-T. The NAT-D payload type number is 0130. For a list of other IKE payload types, see "IKE Packets" on page 719.

NOTE: ScreenOS can handle multiple NAT-Discovery (NAT-D) payloads in an IKE negotiation.

The NAT-D payloads contain a negotiated hash of the following information:

- Destination NAT-D hash:
  - Initiator's cookie (CKY-I)
  - Responder's cookie (CKY-R)
  - Remote (destination) IKE peer's IP address
  - Destination port number
- Source NAT-D hash (one or more):
  - Initiator's cookie (CKY-I)
  - Responder's cookie (CKY-R)
  - Local (source) IKE peer's IP address
  - Source port number

NOTE: NAT-T supports multiple source NAT-D hashes for devices with multiple interfaces and implementations that do not specify an outgoing interface.

When each peer compares the hashes it receives with the ones it sends, it can tell if address translation has occurred between them. Distinguishing which packet has been modified also indicates the location of the NAT device:

If	Matches	Then
the local peer's destination hash	at least one of the remote peer's source hashes	no address translation has occurred.
at least one of the local peer's source hashes	the remote peer's destination hash	no address translation has occurred.

If	Does not match	Then
the local peer's destination hash	at least one of the remote peer's source hashes	no address translation has occurred.
at least one of the local peer's source hashes	the remote peer's destination hash	no address translation has occurred.

Knowing the location of the NAT device is important because IKE keepalives must initiate from the peer behind the NAT device. See "Keepalive Packets" on page 966.

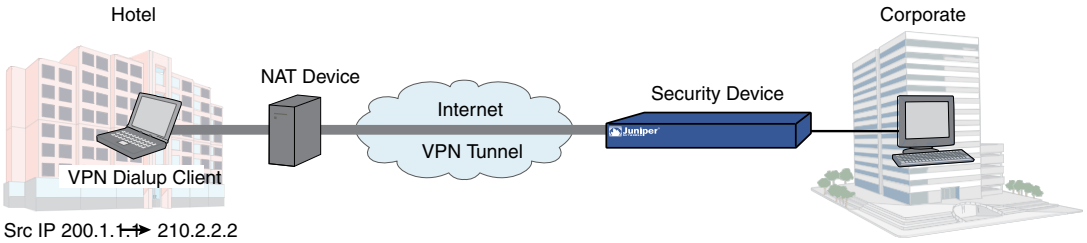
If both peers support IETF draft 2, then they also float the IKE port number from 500 to 4500 as soon as they detect a NAT device between themselves during Phase 1 negotiations. In main mode, the port numbers float to 4500 in the fifth and sixth

exchanges of Phase 1, and then for all Phase 2 exchanges. In aggressive mode, the port number floats to 4500 in the third—and final—exchange of Phase 1, and then for all Phase 2 exchanges. The peers also use 4500 for the UDP port number for all subsequent IPsec traffic.

Traversing a NAT Device

In Figure 256 on page 964, a NAT device at the perimeter of a hotel LAN receives a packet from a VPN dialup client with IP address 2.1.1.5, assigned by the hotel. For all outbound traffic, the NAT device replaces the original source IP address in the outer header with a new address 2.2.2.2. During Phase 1 negotiations, the VPN client and the security device detect that both VPN participants support NAT-T, that a NAT device is present along the data path, and that it is located in front of the VPN client.

Figure 256: NAT-Traversal



Encapsulating the IPsec packets within UDP packets—which both the VPN client and the security device do—solves the problem of the authentication check failure. The NAT device processes them as UDP packets, changing the source port in the UDP header and leaving the SPI in the AH or ESP header unmodified. The VPN participants strip off the UDP layer and process the IPsec packets, which pass the authentication check because none of the authenticated content has been changed.

Another problem can arise if the NAT device is IKE/IPsec-aware. An IKE/IPsec-aware NAT device might attempt to process IKE/IPsec traffic instead of forwarding it. To prevent such intermediary processing, NAT-T (v2) changes the source and destination UDP port numbers for IKE from 500 to 4500. NAT-T also inserts a non-ESP marker in the UDP header just before the payload. For IPsec traffic, NAT-T (v0 and v2) inserts a UDP header between the outer IP header and the ESP header. The UDP packet also uses 4500 for both the source and destination port numbers.

As mentioned, NAT-T (v2) adds a non-ESP marker between the header and payload of the UDP segment encapsulating the ISAKMP packet. The non-ESP marker is 4 bytes of zero (0000), and is added to the UDP segment to distinguish an encapsulated ISAKMP packet from an encapsulated ESP packet, which does not have such a marker. Without the non-ESP marker, the recipient would be unsure if the encapsulated packet was an ISAKMP packet or an ESP packet because the UDP header uses 4500 for both types. Using this marker indicates the correct type of packet that is encapsulated so that the recipient can correctly demultiplex it.

As shown in Figure 257 on page 965, after detecting a NAT device in the data path, the source and destination port numbers in the UDP header of an IKE packet change from 500 to 4500. Also, the VPN tunnel endpoints insert a non-ESP marker between the UDP header and payload to distinguish the encapsulated ISAKMP packet from

an ESP packet. The recipient can use this marker to distinguish the encapsulated ISAKMP packet from an ESP packet and demultiplex it correctly.

Figure 257: IKE Packet (for Phases 1 and 2)

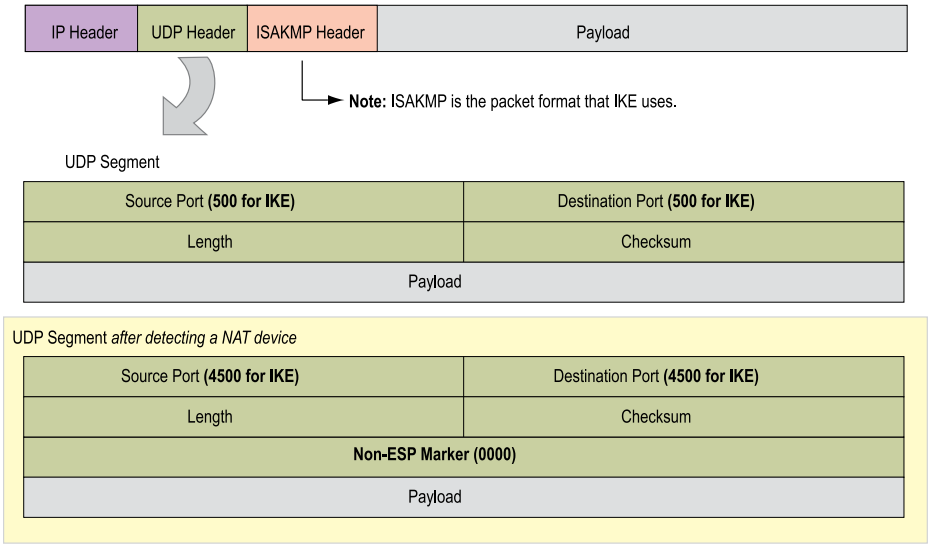
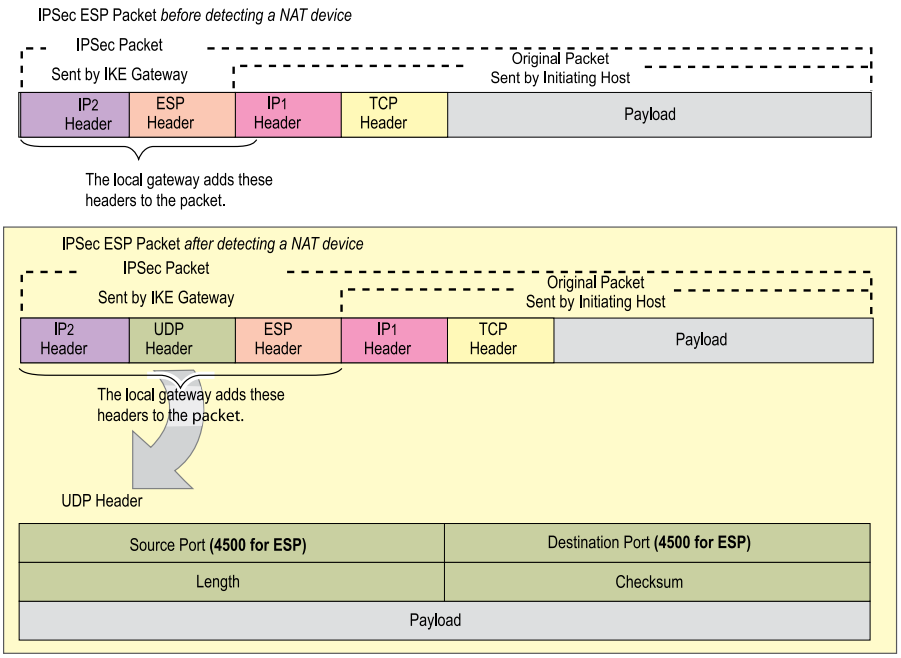


Figure 258 on page 965 shows how, after detecting a NAT device in the data path, the VPN tunnel endpoints insert an additional UDP header between the outer IP header and the ESP header of an IPsec packet. Because there is no non-ESP marker, the recipient can distinguish the encapsulated ESP packet from an ISAKMP packet and demultiplex the ESP packet correctly.

Figure 258: IPsec ESP Packet Before and After NAT Detection



UDP Checksum

All UDP packets contain a UDP checksum, a calculated value that ensures UDP packets are free of transmission errors. A security device does not require use of the UDP checksum for NAT-T, so the WebUI and CLI present the checksum as an optional setting. Even so, some NAT devices require a checksum, so you might have to enable or disable this setting. By default, the UDP checksum is included when you enable NAT-T.

WebUI

VPNs > AutoKey Advanced > Gateway > New:

Enter the necessary parameters for the new tunnel gateway as described in "Site-to-Site Virtual Private Networks" on page 801 or "Dialup Virtual Private Networks" on page 887; enter the following, then click **OK**:

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Enable NAT-Traversal: (select)
UDP Checksum: Enable

CLI

```
set ike gateway name nat-traversal udp-checksum
unset ike gateway name nat-traversal udp-checksum
```

Keepalive Packets

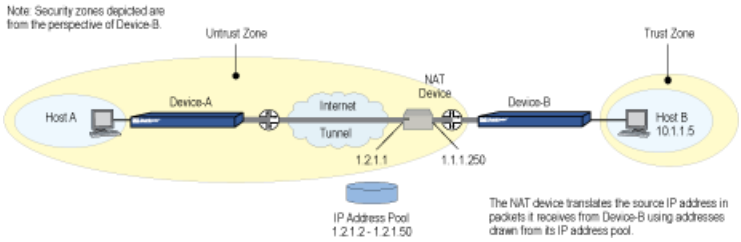
When a NAT device assigns an IP address to a host, the NAT device determines how long the new address remains valid when no traffic occurs. For example, a NAT device might invalidate any generated IP address that remains unused for 20 seconds. Therefore, it is usually necessary for the IPsec participants to send periodic keepalive packets—empty UDP packets—through the NAT device, so that the NAT mapping does not change until the Phase 1 and Phase 2 SAs expire.

NOTE: NAT devices have different session timeout intervals, depending on the manufacturer and model. It is important to determine what the interval is for the NAT device and to set the keepalive frequency value below that.

Initiator/Responder Symmetry

When two security devices establish a tunnel in the absence of a NAT device, either device can serve as initiator or responder. However, if either host resides behind a NAT device, such initiator/responder symmetry might be impossible. This happens whenever the NAT device generates IP addresses dynamically.

Figure 259: Security Device with a Dynamically Assigned IP Address Behind a NAT Device



In Figure 259 on page 967, Device B resides in a subnet located behind a NAT device. If the NAT device generates new source IP addresses for packets it receives from Device B—drawing them dynamically from a pool of IP addresses—Device A cannot unambiguously identify Device B. Therefore, Device A cannot successfully initiate a tunnel with Device B. Device A must be the responder, Device B the initiator, and they must perform Phase 1 negotiations in aggressive mode.

However, if the NAT device generates the new IP address using a mapped IP (MIP) address, or some other one-to-one addressing method, Device A can unambiguously identify Device B. Consequently, either Device A or Device B can be the initiator, and both can use main or aggressive mode for Phase 1. Device B, which is not behind the NAT device, configures this new IP address as the IKE gateway address. At this time, the local ID or ID (peer ID) needs to be set.

NOTE: If you enable NAT-T on a security device acting as the responder and configure it to perform IKE negotiations in main mode, that device and all its peers of the following types that are configured on the same outgoing interface must use the same Phase 1 proposals presented in the same order as each other:

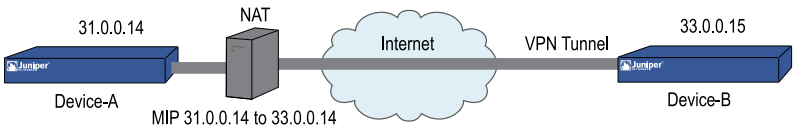
Dynamic peer (peers with dynamically assigned IP addresses)
Dialup VPN users
Peers with static IP addresses behind a NAT device

Because it is not possible to know the identity of a peer when negotiating Phase 1 in main mode until the last two messages, the Phase 1 proposals must all be the same so that IKE negotiations can proceed.

The security device automatically checks that all Phase 1 proposals are the same and in the same order when you configure IKE in main mode to one of the above peer types on the same outgoing interface. If the proposals are different, the security device generates an error message.

In the example shown in Figure 260 on page 968, two devices, Device A and Device B, are connected by a VPN tunnel. Device A is behind a NAT and has a private IP 31.0.0.14. The NAT generates a new public IP using MIP for Device A. You use the MIP address as the gateway address while configuring IKEv2 gateway on Device B. For more information about MIPs, see “Mapped IP Addresses” on page 1535.

Figure 260: Security Device with a Mapped IP Address Behind a NAT Device



Device A Configuration

```
set ike gateway ikev2 "dev-b" address 33.0.0.15 id "dev-b.net" local-id "dev-a.net"
 outgoing-interface "ethernet0/1" preshare
"KghBa3TbNruG2Es6e2C5zkr83SnLzly1MQ==" proposal "pre-g2-3des-md5"
set ike gateway ikev2 "dev-b" nat-traversal
set ike gateway ikev2 "dev-b" nat-traversal udp-checksum
set ike gateway ikev2 "dev-b" nat-traversal keepalive-frequency 5
set vpn "dev-b" gateway "dev-b" no-replay tunnel idletime 0 proposal
"g2-esp-3des-md5"
set vpn "dev-b" id 0x1 bind interface tunnel.1
```

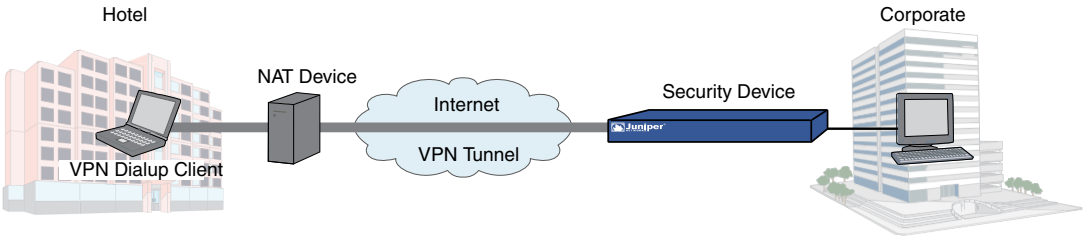
Device B Configuration

```
set ike gateway ikev2 "dev-a" address 33.0.0.14 id "dev-a.net" local-id "dev-b.net"
 outgoing-interface "ethernet2/3" preshare
"5LXhnzFYNz8E06srN9CgzDdrpKnEep28Uw==" proposal "pre-g2-3des-md5"
set ike gateway ikev2 "dev-a" nat-traversal
set ike gateway ikev2 "dev-a" nat-traversal udp-checksum
set ike gateway ikev2 "dev-a" nat-traversal keepalive-frequency 5
set vpn "dev-a" gateway "dev-a" no-replay tunnel idletime 0 proposal
"g2-esp-3des-md5"
set vpn "dev-a" id 0x1 bind interface tunnel.1
```

Enabling NAT-Traversal

In Figure 261 on page 968, a NAT device at the perimeter of a hotel LAN assigns an address to the VPN dialup client used by Jozef, a salesman attending a convention. For Jozef to reach the corporate LAN through a dialup VPN tunnel, you must enable NAT-T for the remote gateway "jozef," configured on the security device, and for the remote gateway configured on the VPN dialup client. You also enable the security device to include a UDP checksum in its transmissions, and you set the keepalive frequency to 8 seconds.

Figure 261: Enabling NAT-Traversal



WebUI

VPNs > AutoKey Advanced > Gateway > New: Enter the necessary parameters for the new tunnel gateway (described in "Site-to-Site Virtual Private Networks" on page 801 or "Dialup Virtual Private Networks" on page 887), enter the following, then click **OK**:

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Enable NAT-Traversal: (select)
UDP Checksum: Enable
Keepalive Frequency: 8 Seconds (0~300 Sec)

NOTE: When you configure a dialup VPN through the CLI, the security device automatically enables NAT-Traversal.

CLI

```
set ike gateway jozef nat-traversal
set ike gateway jozef nat-traversal udp-checksum
set ike gateway jozef nat-traversal keepalive-frequency 8
save
```

Using IKE IDs with NAT-Traversal

When two VPN gateways negotiate in main mode, they exchange IP addresses in order to identify each other and activate the tunnel. When devices at either or both ends of the tunnel have dynamically assigned IP addresses, however, you must configure IKE IDs (local ID and peer ID) on the devices at both ends of the tunnel. An IKE ID is a unique identifier that remains static. The IKE ID is set up during the phase when you configure the IKE gateway.You configure IKE IDs instead of remote IP address.

Without NAT-T, a VPN tunnel can be activated using only the local ID on the local side and only the peer ID on the remote side. But when using NAT-Traversal with dynamic VPN in main mode using certificates, you must set both the local ID and peer ID on both sides of the VPN tunnel. The following example shows how you can configure local IDs and peer IDs on firewall1 and firewall2 so they can identify each other and activate a tunnel between them.

WebUI

On firewall1, enter the following:

VPNs > AutoKey IKE Advanced > Gateway > New: Enter the following, then click **Advanced**:

Gateway Name: test_gw
Address/Hostname: 0.0.0.0
Peer-ID: firewall2

Click **Advanced**, then enter the following:

Security Level: Standard
Local-ID: firewall1
Outgoing Interface: ethernet0/0
Mode: Main
Security Level: Standard

On firewall2, enter the following:

VPNs > AutoKey IKE Advanced > Gateway > New: Enter the following, then click **Advanced**:

Gateway Name: gw_bap15_p1
Address/Hostname: 1.1.1.1
Peer-ID: firewall1

Click **Advanced**, then enter the following:

Security Level: Standard
Local-ID: firewall2
Outgoing Interface: ethernet0/0
Mode: Main
Security Level: Standard

CLI

On firewall1, enter the following:

```
set ike gateway test-gw address 0.0.0.0 id firewall2 main local-id firewall1
outgoing-interface ethernet0/0 proposal standard
```

On firewall2, enter the following

```
set ike gateway gw_bap15_p1 address 1.1.1.1 id firewall1 main local-id firewall2
outgoing-interface ethernet0/0 proposal standard
```

The following table shows the CLI command for each of the IPsec NAT-T tasks:

To	Use This CLI Command
Enable IPsec NAT-T per gateway	**set ike nat-t gateway name**
Disable IPsec NAT-T per gateway	**unset ike nat-t gateway name**
Set the IPsec NAT-T keepalive per gateway	**set ike nat-t keep-alive period seconds**
Set the IPsec NAT-T keepalive count per gateway	**set ike nat-t keep-alive count count**
Get the IPsec NAT-T status	**get ike nat-t gateway name**

VPN Monitoring

When you enable VPN monitoring for a specific tunnel, the security device sends ICMP echo requests (or "pings") through the tunnel at specified intervals (configured in seconds) to monitor network connectivity through the tunnel.

NOTE: To change the ping interval, you can use the following CLI command: **set vpnmonitor interval** number. The default is 10 seconds.

If the ping activity indicates that the VPN monitoring status has changed, the security device triggers one of the following Simple Network Management Protocol (SNMP) traps:

- **Up to Down:** This trap occurs when the state of VPN monitoring for the tunnel is up, but a specified consecutive number of ICMP echo requests does not elicit a reply and there is no other incoming VPN traffic. Then the state changes to down.
- **Down to Up:** When the state of VPN monitoring for the tunnel is down, but the ICMP echo request elicits a single response, then the state changes to up. The down-to-up trap occurs only if you have disabled the rekey option and the Phase 2 SA is still active when an ICMP echo request elicits a reply through the tunnel.

NOTE: To change the threshold for the number of consecutive unsuccessful ICMP echo requests, you can use the following CLI command: **set vpnmonitor threshold** number. The default is 10 consecutive requests.

For more information about the SNMP data that VPN monitoring provides, see "SNMP VPN Monitoring Objects and Traps" on page 982.

You apply VPN monitoring per VPN object, not necessarily per VPN tunnel. A VPN object is what you define with the **set vpn** command or with its WebUI counterpart. After you define one VPN object, you can then reference it in one or more policies (creating policy-based VPNs). Because ScreenOS derives a policy-based VPN tunnel from a VPN object plus the other policy parameters, a single VPN object can be an element in numerous VPN tunnels. This distinction between VPN object and VPN tunnel is important because Juniper Networks recommends that you apply VPN monitoring to no more than 100 IPsec VPN tunnels—if you do not enable optimization. If you do enable optimization, then there is no limitation to the number of VPN tunnels to which you can apply VPN monitoring. To learn about the optimization option, see "Rekey and Optimization Options" on page 972.

NOTE: VPN monitoring optimization operates on a per-object basis. You can enable it on all VPN objects, on none, or only on some.

Rekey and Optimization Options

If you enable the rekey option, the security device starts sending ICMP echo requests immediately upon completion of the tunnel configuration and continues to send them indefinitely. The echo requests trigger an attempt to initiate IKE negotiations to establish a VPN tunnel until the state of VPN monitoring for the tunnel is up. The security device then uses the pings for VPN monitoring purposes. If the state of VPN monitoring for the tunnel changes from up to down, the security device deactivates its Phase 2 security association (SA) for that peer. The security device continues to send echo requests to its peer at defined intervals, triggering attempts to reinitiate IKE Phase 2 negotiations—and Phase 1 negotiations, if necessary—until it succeeds. At that point, the security device reactivates the Phase 2 SA, generates a new key, and reestablishes the tunnel. A message appears in the event log stating that a successful rekey operation has occurred.

NOTE: If a security device is a DHCP client, a DHCP update to a different address causes IKE to rekey. However, a DHCP update to the same address does not provoke the IKE rekey operation.

You can use the rekey option to ensure that an AutoKey IKE tunnel is always up, perhaps to monitor devices at the remote site or to allow dynamic routing protocols to learn routes at a remote site and transmit messages through the tunnel. Another use to which you can apply VPN monitoring with the rekey option is for automatic population of the next-hop tunnel binding table (NHTB table) and the route table when multiple VPN tunnels are bound to a single tunnel interface. For an example of this last use, see “Multiple Tunnels per Tunnel Interface” on page 983.

If you disable the rekey option, the security device performs VPN monitoring only when the tunnel is active with user-generated traffic.

By default, VPN monitoring optimization is disabled. If you enable it (**set vpn** name **monitor optimized**), the VPN monitoring behavior changes as follows:

- The security device considers incoming traffic through the VPN tunnel to be the equivalent of ICMP echo replies. Accepting incoming traffic as a substitute for ICMP echo replies can reduce false alarms that might occur when traffic through the tunnel is heavy and the echo replies do not get through.
- If there is both incoming and outgoing traffic through the VPN tunnel, the security device suppresses VPN monitoring pings altogether. Doing so can help reduce network traffic.

Although VPN monitoring optimization offers some benefits, be aware that VPN monitoring can no longer provide accurate SNMP statistics, such as VPN network delay time, when the optimization option is active. Also, if you are using VPN monitoring to track the availability of a particular destination IP address at the remote end of a tunnel, the optimization feature can produce misleading results.

Source Interface and Destination Address

By default, the VPN monitoring feature uses the IP address of the local outgoing interface as the source address and the IP address of the remote gateway as the destination address. If the remote peer is a VPN dialup client—such as the NetScreen-Remote—that has an internal IP address, the security device automatically detects its internal address and uses that as the destination. The VPN client can be an XAuth user with an assigned internal IP address, or a dialup VPN user or a member of a dialup VPN group with an internal IP address. You can also specify the use of other source and destination IP addresses for VPN monitoring—mainly to provide support for VPN monitoring when the other end of a VPN tunnel is not a security device.

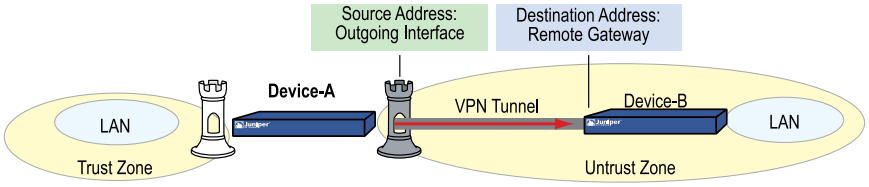
Because VPN monitoring operates independently at the local and remote sites, the source address configured on the device at one end of a tunnel does not have to be the destination address configured on the device at the other end. In fact, you can enable VPN monitoring at both ends of a tunnel or at only one end.

Figure 262: Source and Destination Addresses for VPN Monitoring

Device-A –> Device-B

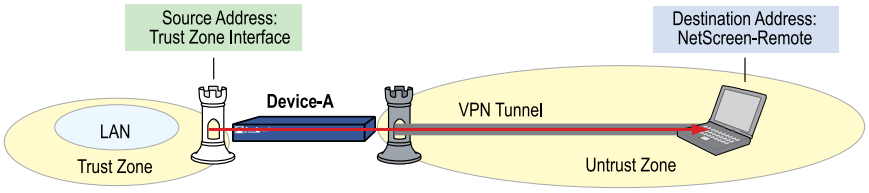
Device-A pings from its outgoing interface to the remote gateway; that is, from the Untrust zone interface on Device-A to the Untrust zone interface on Device-B.

(Default Behavior)



Device-A –> NetScreen-Remote

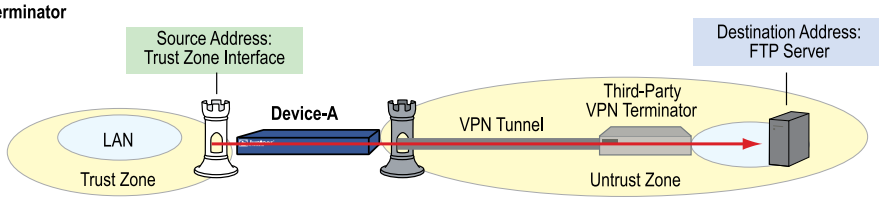
Device-A pings from its Trust zone interface to the NetScreen-Remote. The NetScreen-Remote requires a policy permitting inbound ICMP traffic from an address beyond the remote gateway; that is, from beyond the Untrust zone interface of Device-A.



Note: Device-A requires a policy permitting ping traffic from the Trust to Untrust zones.

Device-A –> Third-Party VPN Terminator

Device-A pings from its Trust zone interface to a device beyond the remote gateway. This might be necessary if the remote peer does not respond to pings but can support policies permitting inbound ping traffic.



Note: Device-A requires a policy permitting ping traffic from the Trust to Untrust zones.

NOTE: If the other end of a tunnel is the NetScreen-Remote VPN client that receives its address through XAuth, then the security device, by default, uses the XAuth-assigned IP address as the destination for VPN monitoring. For information about XAuth, see “XAuth Users and User Groups” on page 1640.

Policy Considerations

You must create a policy on the sending device to permit pings from the zone containing the source interface to pass through the VPN tunnel to the zone containing the destination address if:

- The source interface is in a different zone from the destination address
- The source interface is in the same zone as the destination address, and intrazone blocking is enabled

Likewise, you must create a policy on the receiving device to permit pings from the zone containing the source address to pass through the VPN tunnel to the zone containing the destination address if:

- The destination address is in a different zone from the source address
- The destination address is in the same zone as the source address, and intrazone blocking is enabled

NOTE: If the receiving device is a third-party product that does not respond to the ICMP echo requests, change the destination to an internal host in the remote peer's LAN that does respond. The remote peer's firewall must have a policy permitting the ICMP echo requests to pass through it.

Configuring the VPN Monitoring Feature

To enable VPN monitoring, do the following:

WebUI

VPNs > AutoKey IKE > New: Configure the VPN, click **Advanced**, enter the following, click **Return** to go back to the basic VPN configuration page, then click **OK**:

VPN Monitor: Select to enable VPN monitoring of this VPN tunnel.

Source Interface: Choose an interface from the drop-down list. If you choose **default**, the security device uses the outgoing interface.

Destination IP: Enter a destination IP address. If you do not enter anything, the security device uses the remote gateway IP address.

Rekey: Select this option if you want the security device to attempt IKE Phase 2 negotiations—and IKE Phase 1 negotiations if necessary—if the tunnel status changes from up to down. When you select this option, the security device attempts IKE negotiations to set up the tunnel and begin VPN monitoring immediately after you finish configuring the tunnel.

Clear this option if you do not want the security device to attempt IKE negotiations if the tunnel status changes from up to down. When the rekey option is disabled,

VPN monitoring begins after user-generated traffic has triggered IKE negotiations and stops when the tunnel status changes from up to down.

(Or)

VPNs > Manual Key > New: Configure the VPN, click **Advanced**, enter the following, click **Return** to go back to the basic VPN configuration page, then click **OK**:

VPN Monitor: Select to enable VPN monitoring of this VPN tunnel.

Source Interface: Choose an interface from the drop-down list. If you choose **default**, the security device uses the outgoing interface.

Destination IP: Enter a destination IP address. If you do not enter anything, the security device uses the remote gateway IP address.

CLI

```
set vpnmonitor frequency number
set vpnmonitor threshold number
set vpn name_str monitor [ source-interface interface [ destination-ip ip_addr ]
[optimized] [ rekey ]
save
```

NOTE: The VPN monitoring frequency is in seconds. The default setting is 10-second intervals.

The VPN monitoring threshold number is the consecutive number of successful or unsuccessful ICMP echo requests that determines whether the remote gateway is reachable through the VPN tunnel or not. The default threshold is 10 consecutive successful or 10 consecutive unsuccessful ICMP echo requests.

If you do not choose a source interface, the security device uses the outgoing interface as the default.

If you do not choose a destination IP address, the security device uses the IP address for the remote gateway.

The rekey option is not available for Manual Key VPN tunnels.

In this example, you configure an AutoKey IKE VPN tunnel between two security devices (Device A and Device B). On Device A, you set up VPN monitoring from its Trust zone interface (ethernet1) to the Trust zone interface (10.2.1.1/24) on Device B. On the Device B, you set up VPN monitoring from its Trust zone interface (ethernet1) to a corporate intranet server (10.1.1.5) behind Device A.

NOTE: A Phase 1 security level of Compatible includes these proposals: pre-g2-3des-sha, pre-g2-3des-md5, pre-g2-des-sha, and pre-g2-des-md5.

A Phase 2 security level of Compatible includes these proposals: nopfs-esp-3des-sha, nopfs-esp-3des-md5, nopfs-esp-des-sha, and nopfs-esp-des-md5.

Device A	Device B
Zones and Interfaces	
■ ethernet1 ■ Zone: Trust ■ IP address: 10.1.1.1/24 ■ Interface mode: NAT ■ ethernet3 ■ Zone: Untrust ■ IP address: 1.1.1.1/24	■ ethernet1 ■ Zone: Trust ■ IP address: 10.2.1.1/24 ■ Interface mode: NAT ■ ethernet3 ■ Zone: Untrust ■ IP address: 2.2.2.2/24
Route-Based AutoKey IKE Tunnel Parameters	
■ Phase 1 ■ Gateway name: gw1 ■ Gateway static IP address: 2.2.2.2 ■ Security level: Compatible ■ Preshared Key: Ti82g4aX ■ Outgoing interface: ethernet3 ■ Mode: Main ■ Phase 2 ■ VPN tunnel name: vpn1 ■ Security level: Compatible ■ VPN Monitoring: src = ethernet1; dst = 10.2.1.1 ■ Bound to interface: tunnel.1	■ Phase 1 ■ Gateway name: gw1 ■ Gateway static IP address: 1.1.1.1 ■ Proposals: Compatible ■ Preshared Key: Ti82g4aX ■ Outgoing interface: ethernet3 ■ Mode: Main ■ Phase 2 ■ VPN tunnel name: vpn1 ■ Security level: Compatible ■ VPN Monitoring: src = ethernet1; dst = 10.1.1.5 ■ Bound to interface: tunnel.1

Device A	Device B
Routes	
To 0.0.0.0/0, use ethernet3, gateway 1.1.1.250 To 10.2.1.0/24, use tunnel.1, no gateway (Null route–to drop traffic to 10.2.1.0/24 if tunnel.1 goes down) To 10.2.1.0/24, use null interface, metric: 10	To 0.0.0.0/0, use ethernet3, gateway 2.2.2.250 To 10.1.1.0/24, use tunnel.1, no gateway (Null route–to drop traffic to 10.1.1.0/24 if tunnel.1 goes down) To 10.1.1.0/24, use null interface, metric: 10

Because both devices ping from an interface in their Trust zone to an address in their Untrust zone, the admins at both ends of the VPN tunnel must define policies permitting pings to pass from zone to zone.

NOTE: Because both VPN terminators are security devices in this example, you can use the default source and destination addresses for VPN monitoring. The use of other options is included purely to illustrate how you can configure a security device to use them.

WebUI (Device A)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > Tunnel IF New: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Trust (trust-vr)
   Unnumbered: (select)
   Interface: ethernet1(trust-vr)

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Trust_LAN
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Remote_LAN
   IP Address/Domain Name:
   IP/Netmask: (select), 10.2.1.0/24
   Zone: Untrust

3. **VPN**

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   ```
   VPN Name: vpn1
   Security Level: Compatible
   Remote Gateway:
      Create a Simple Gateway: (select)
        Gateway Name: gw1
        Type:
          Static IP: (select), Address/Hostname: 2.2.2.2
        Preshared Key: Ti82g4aX
        Security Level: Compatible
        Outgoing Interface: ethernet3
   ```

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

   ```
   Bind to: Tunnel Interface, tunnel.1
   Proxy-ID: (select)
   Local IP / Netmask: 10.1.1.0/24
   Remote IP / Netmask: 10.2.1.0/24
   Service: ANY
   VPN Monitor: (select)
   Source Interface: ethernet1
   Destination IP: 10.2.1.1
   Rekey: (clear)
   ```

4. **Routes**

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   ```
   Network Address / Netmask: 0.0.0.0/0
   Gateway: (select)
      Interface: ethernet3
      Gateway IP Address: 1.1.1.250
   ```

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   ```
   Network Address / Netmask: 10.2.1.0/24
   Gateway: (select)
      Interface: Tunnel.1
      Gateway IP Address: 0.0.0.0
   ```

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   ```
   Network Address / Netmask: 10.2.1.0/24
   Gateway: (select)
      Interface: Null
      Gateway IP Address: 0.0.0.0
      Metric: 10
   ```

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), Trust_LAN
Destination Address:
  Address Book Entry: (select), Remote_LAN
Service: ANY
Action: Permit
Position at Top: (select)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK:**

Source Address:
  Address Book Entry: (select), Remote_LAN
Destination Address:
  Address Book Entry: (select), Trust_LAN
Service: Any
Action: Permit
Position at Top: (select)

WebUI (Device B)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.2.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 2.2.2.2/24

   Network > Interfaces > Tunnel IF New: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Trust (trust-vr)
   Unnumbered: (select)
     Interface: ethernet1(trust-vr)

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Trust_LAN
   IP Address/Domain Name:
     IP/Netmask: (select), 10.2.1.0/24
   Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Remote_LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Untrust

3. **VPN**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway:
Create a Simple Gateway: (select)
Gateway Name: gw1
Type:
Static IP: (select), Address/Hostname: 1.1.1.1
Preshared Key: Ti82g4aX
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 10.2.1.0/24
Remote IP / Netmask: 10.1.1.0/24
Service: ANY
VPN Monitor: (select)
Source Interface: ethernet1
Destination IP: 10.1.1.5
Rekey: (clear)

4. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 2.2.2.250

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.1.1.0/24
Gateway: (select)
Interface: Tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.1.1.0/24
Gateway: (select)
Interface: Null
Gateway IP Address: 0.0.0.0
Metric: 10

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Trust_LAN
Destination Address:
Address Book Entry: (select), Remote_LAN
Service: ANY
Action: Permit
Position at Top: (select)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Remote_LAN
Destination Address:
Address Book Entry: (select), Trust_LAN
Service: Any
Action: Permit
Position at Top: (select)

CLI (Device A)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone trust
set interface tunnel.1 ip unnumbered interface ethernet1
```

2. **Addresses**

```
set address trust Trust_LAN 10.1.1.0/24
set address untrust Remote_LAN 10.2.1.0/24
```

3. **VPN**

```
set ike gateway gw1 address 2.2.2.2 main outgoing-interface ethernet3 preshare
 Ti82g4aX sec-level compatible
set vpn vpn1 gateway gw1 sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 10.1.1.0/24 remote-ip 10.2.1.0/24 any
set vpn vpn1 monitor source-interface ethernet1 destination-ip 10.2.1.1
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
set vrouter trust-vr route 10.2.1.0/24 interface tunnel.1
set vrouter trust-vr route 10.2.1.0/24 interface null metric 10
```

5. **Policies**

```
set policy top from trust to untrust Trust_LAN Remote_LAN any permit
set policy top from untrust to trust Remote_LAN Trust_LAN any permit
save
```

CLI (Device B)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.1 zone trust
set interface tunnel.1 ip unnumbered interface ethernet1
```

2. **Addresses**

```
set address trust Trust_LAN 10.2.1.0/24
set address untrust Remote_LAN 10.1.1.0/24
```

3. **VPN**

```
set ike gateway gw1 address 1.1.1.1 main outgoing-interface ethernet3 preshare
 Ti82g4aX sec-level compatible
set vpn vpn1 gateway gw1 sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 10.2.1.0/24 remote-ip 10.1.1.0/24 any
set vpn vpn1 monitor source-interface ethernet1 destination-ip 10.1.1.5
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
set vrouter trust-vr route 10.1.1.0/24 interface tunnel.1
set vrouter trust-vr route 10.1.1.0/24 interface null metric 10
```

5. **Policies**

```
set policy top from trust to untrust Trust_LAN Remote_LAN any permit
set policy top from untrust to trust Remote_LAN Trust_LAN any permit
save
```

SNMP VPN Monitoring Objects and Traps

ScreenOS provides the ability to determine the status and condition of active VPNs through the use of Simple Network Management Protocol (SNMP) VPN monitoring objects and traps. The VPN monitoring MIB notes whether each ICMP echo request

elicits a reply, a running average of successful replies, the latency of the reply, and the average latency over the last 30 attempts.

NOTE: To enable your SNMP manager application to recognize the VPN monitoring MIBs, you must import the ScreenOS-specific MIB extension files into the application. You can find the MIB extension files on the documentation CD that shipped with your security device.

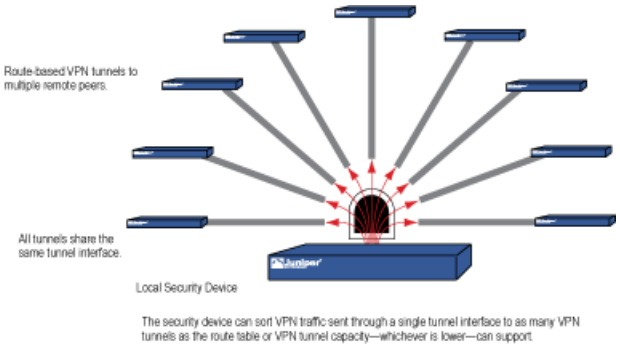
By enabling the VPN monitoring feature on an AutoKey IKE or Manual Key VPN tunnel, the security device activates its SNMP VPN monitoring objects, which include data on the following:

- The total number of active VPN sessions
- The time each session started
- The Security Association (SA) elements for each session:
  - ESP encryption (DES or 3DES) and authentication algorithm (MD5, SHA-1 or SHA2-256) types
  - AH algorithm type (MD5, SHA-1 or SHA2-256)
  - Key exchange protocol (AutoKey IKE or Manual Key)
  - Phase 1 authentication method (preshared key or certificates)
  - VPN type (dialup or peer-to-peer)
  - Peer and local gateway IP addresses
  - Peer and local gateway IDs
  - Security Parameter Index (SPI) numbers
- Session status parameters
  - VPN monitoring status (up or down)
  - Tunnel status (up or down)
  - Phase 1 and 2 status (inactive or active)
  - Phase 1 and 2 lifetime (time in seconds before rekeying; Phase 2 lifetime is also reported in remaining bytes before rekeying)

Multiple Tunnels per Tunnel Interface

You can bind multiple IPsec VPN tunnels to a single tunnel interface. To link a specific destination to one of a number of VPN tunnels bound to the same tunnel interface, the security device uses two tables: the route table and the next-hop tunnel binding (NHTB) table. The security device maps the next-hop gateway IP address specified in the route table entry to a particular VPN tunnel specified in the NHTB table. With this technique, a single tunnel interface can support many VPN tunnels. (See “Route-to-Tunnel Mapping” on page 984.)

Figure 263: One Tunnel Interface Bound to Multiple Tunnels



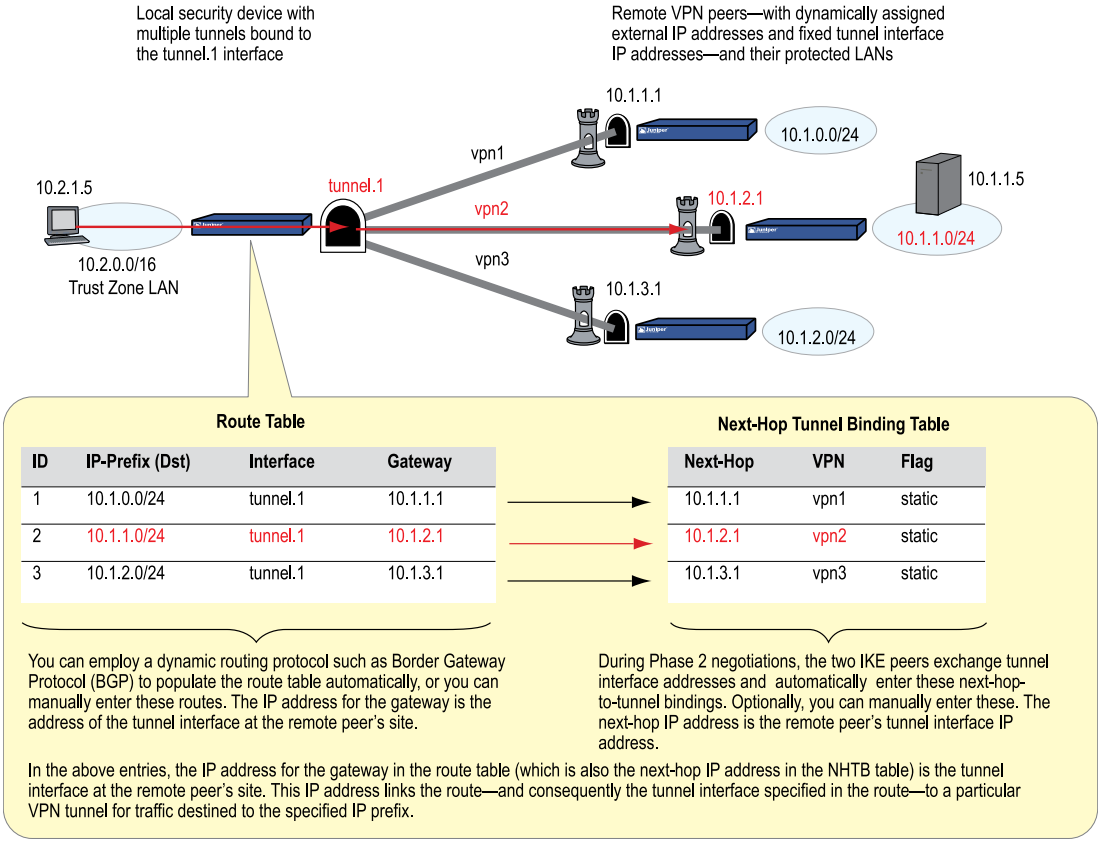
The maximum number of VPN tunnels is not limited by the number of tunnel interfaces that you can create, but by either route table capacity or the maximum number of dedicated VPN tunnels allowed—whichever is lower. For instance, if your security device supports 4000 routes and 1000 dedicated VPN tunnels, you can create 1000 VPN tunnels and bind them to a single tunnel interface. If your security device supports 8192 routes and 10,000 dedicated VPN tunnels, then you can create over 8000 VPN tunnels and bind them to a single tunnel interface. To see the maximum route and tunnel capacities for your security device, refer to the relevant product datasheet.

NOTE: If route-table capacity is the limiting factor, you must subtract the routes automatically generated by security zone interfaces and any other static routes—such as the route to the default gateway—that you might need to define from the total available for route-based VPN tunnels.

Route-to-Tunnel Mapping

To sort traffic among multiple VPN tunnels bound to the same tunnel interface, the security device maps the next-hop gateway IP address specified in the route to a particular VPN tunnel name. The mapping of entries in the route table to entries in the NHTB table is shown below. In Figure 264 on page 985, the local security device routes traffic sent from 10.2.1.5 to 10.1.1.5 through the tunnel.1 interface and then through vpn2.

Figure 264: Route Table and Next-Hop Tunnel Binding (NHTB) Table



The security device uses the IP address of the remote peer's tunnel interface as the gateway and next-hop IP address. You can enter the route manually, or you can allow a dynamic routing protocol to enter a route referencing the peer's tunnel interface IP address as the gateway in the route table automatically. The same IP address must also be entered as the next hop, along with the appropriate VPN tunnel name, in the NHTB table. Again, there are two options: you can either enter it manually, or you can allow the security device to obtain it from the remote peer during Phase 2 negotiations and enter it automatically.

The security device uses the gateway IP address in the route table entry and the next-hop IP address in the NHTB table entry as the common element to link the tunnel interface with the corresponding VPN tunnel. The security device can then direct traffic destined for the IP-prefix specified in the route with the correct VPN tunnel specified in the NHTB table.

Remote Peers' Addresses

The internal addressing scheme for all remote peers reached through route-based VPNs must be unique among each other. One way to accomplish this is for each remote peer to perform Network Address Translation (NAT) for the source and destination addresses. In addition, the tunnel interface IP addresses must also be unique among all remote peers. If you intend to connect to large numbers of remote

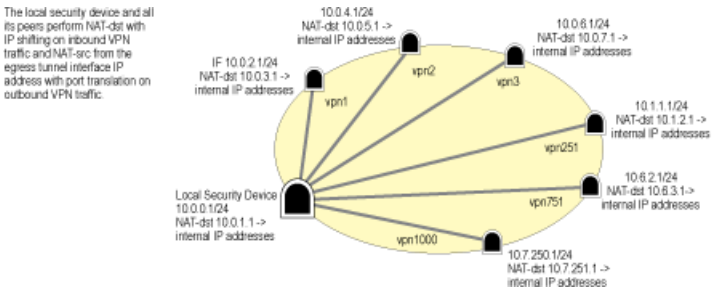
sites, an address plan becomes imperative. The following is a possible addressing plan for up to 1000 VPN tunnels:

Dst in Local Route Table	Local Tunnel Interface	Gateway/Next-Hop (Peer's Tunnel Interface)	VPN Tunnel
10.0.3.0/24	tunnel.1	10.0.2.1/24	vpn1
10.0.5.0/24	tunnel.1	10.0.4.1/24	vpn2
10.0.7.0/24	tunnel.1	10.0.6.1/24	vpn3
…	…	…	…
10.0.251.0/24	tunnel.1	10.0.250.1/24	vpn125
10.1.3.0/24	tunnel.1	10.1.2.1/24	vpn126
10.1.5.0/24	tunnel.1	10.1.4.1/24	vpn127
10.1.7.0/24	tunnel.1	10.1.6.1/24	vpn128
…	…	…	…
10.1.251.0/24	tunnel.1	10.1.250.1/24	vpn250
10.2.3.0/24	tunnel.1	10.2.2.1/24	vpn251
…	…	…	…
10.2.251.0/24	tunnel.1	10.2.250.1/24	vpn375
…	…	…	…
10.7.3.0/24	tunnel.1	10.7.2.1/24	vpn876
…	…	…	…
10.7.251.0/24	tunnel.1	10.7.250.1/24	vpn1000

The tunnel interface on the local security device: is 10.0.0.1/24. On all remote hosts, there is a tunnel interface with an IP address, which appears as the gateway/next-hop IP address in the local route table and NHTB table.

For an example illustrating multiple tunnels bound to a single tunnel interface with address translation, see "Setting VPNs on a Tunnel Interface to Overlapping Subnets" on page 989.

Figure 265: Multiple Tunnels Bound to a Single Tunnel Interface with Address Translation



Manual and Automatic Table Entries

You can make entries in the NHTB and route tables manually. You can also automate the populating of the NHTB and route tables. For a small number of tunnels bound to a single tunnel interface, the manual method works well. For a large number of tunnels, the automatic method reduces administrative setup and maintenance as the routes dynamically self-adjust if tunnels or interfaces become unavailable on the tunnel interface at the hub site.

Manual Table Entries

You can manually map a VPN tunnel to the IP address of a remote peer's tunnel interface in the next-hop tunnel binding (NHTB) table. First, you must contact the remote admin and learn the IP address used for the tunnel interface at that end of a tunnel. Then, you can associate that address with the VPN tunnel name in the NHTB table with the following command:

set interface tunnel.1 nhtb *peer's_tunnel_interface_addr* **vpn***name_str*

After that, you can enter a static route in the route table that uses that tunnel interface IP address as the gateway. You can enter the route either through the WebUI or through the following CLI command:

set vrouter name_str **route***dst_addr* **interface tunnel.1 gateway** *peer's_tunnel_interface_addr*

Automatic Table Entries

To make the population of both the NHTB and route tables automatic, the following conditions must be met:

- The remote peers for all VPN tunnels bound to a single local tunnel interface must be security devices running ScreenOS 5.0.0 or later.
- Each remote peer must bind its tunnel to a tunnel interface, and that interface must have an IP address unique among all peer tunnel interface addresses.
- At both ends of each VPN tunnel, enable VPN monitoring with the rekey option, or enable the IKE heartbeat reconnect option for each remote gateway.
- The local and remote peers must have an instance of a dynamic routing protocol enabled on their connecting tunnel interfaces.

The use of VPN monitoring with the rekey option allows the security devices at both ends of a tunnel to set up the tunnel without having to wait for user-originated VPN traffic. After you enable VPN monitoring with the rekey option at both ends of a VPN tunnel, the two security devices perform Phase 1 and Phase 2 IKE negotiations to establish the tunnel. (For more information, see "VPN Monitoring" on page 971.)

NOTE: If you are running a dynamic routing protocol on the tunnel interfaces, traffic generated by the protocol can trigger IKE negotiations even without enabling VPN monitoring with the rekey option or enabling the IKE heartbeat reconnect option. Still, we recommend that you not rely on dynamic routing traffic to trigger IKE negotiations. Instead use VPN monitoring with the rekey option or the IKE heartbeat reconnect option.

For Open Shortest Path First (OSPF), you must configure the tunnel interface on the local peer as a point-to-multipoint interface before you enable the routing protocol on the interface.

For remote peers with a dynamically assigned external IP address or with a fully qualified domain name (FQDN) mapped to a dynamic IP address, the remote peer must first initiate IKE negotiations. However, because the Phase 2 SA on the local security device caches the remote peer's dynamically assigned IP address, either peer can reinitiate IKE negotiations to reestablish a tunnel whose VPN monitoring state has changed from up to down.

During Phase 2 negotiations, the security devices exchange tunnel interface IP addresses with each other. Each IKE module can then automatically enter the tunnel interface IP address and its corresponding VPN tunnel name in the NHTB table.

To enable the local security device to enter routes to remote destinations automatically in its route table, you must enable an instance of BGP on the local and remote tunnel interfaces. The basic steps are as follows:

1. Create a BGP routing instance on the virtual router that contains the tunnel interface to which you have bound multiple VPN tunnels.
2. Enable the routing instance on the virtual router.
3. Enable the routing instance on the tunnel interface leading to the BPG peers.

The remote peers also perform these steps.

On the local (or hub) device, you must also define a default route and a static route to each peer's tunnel interface IP address. Static routes to the peers' tunnel interfaces are necessary for the hub device to reach its BGP neighbors initially through the correct VPN tunnel.

After establishing communications, the BGP neighbors exchange routing information so that they can each automatically populate their route tables. After the two peers establish a VPN tunnel between themselves, the remote peers can send and receive routing information to and from the local device. When the dynamic routing instance on the local security device learns a route to a peer through a local tunnel interface,

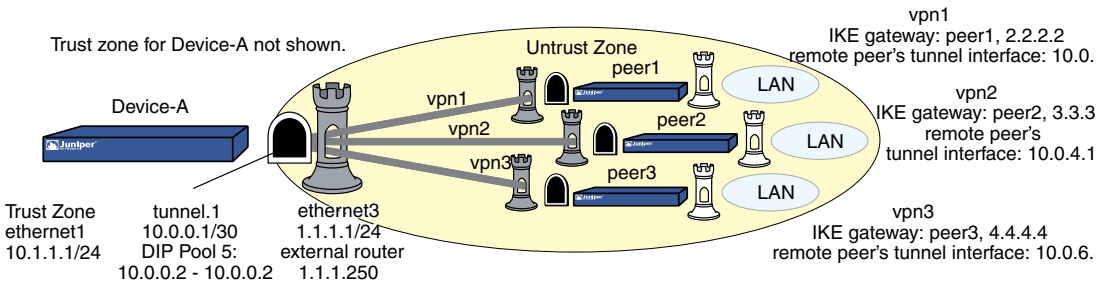
it includes the IP address of the remote peer's tunnel interface as the gateway in the route.

For an example illustrating the configuration of multiple tunnels bound to a single tunnel interface where the "hub" device populates its NHTB and route tables automatically, see "Binding Automatic Route and NHTB Table Entries" on page 1008.

Setting VPNs on a Tunnel Interface to Overlapping Subnets

In this example, you bind three route-based AutoKey IKE VPN tunnels—vpn1, vpn2, and vpn3—to a single tunnel interface—tunnel.1. The tunnels lead from Device A to three remote peers—peer1, peer2, and peer3. You manually add both the route table and NHTB table entries on Device A for all three peers. To see a configuration that provides an automatic means of populating the route and NHTB tables, see "Binding Automatic Route and NHTB Table Entries" on page 1008.

Figure 266: Tunnel.1 interface Bound to Three VPN Tunnels



The VPN tunnel configurations at both ends of each tunnel use the following parameters:

- AutoKey IKE
- Preshared key for each peer:
  - peer1 uses "netscreen1"
  - peer2 uses "netscreen2"
  - peer3 uses "netscreen3"
- Security level predefined as "Compatible" for both Phase 1 and Phase 2 proposals. (For details about these proposals, see "Tunnel Negotiation" on page 715.)

All security zones and interfaces on each device are in the trust-vr virtual routing domain for that device.

This example uses the same address space—10.1.1.0/24—for every LAN to show how you can use Source Network Address Translation (NAT-src) and Destination Network Address Translation (NAT-dst) to overcome addressing conflicts among IPsec peers. For more information about NAT-src and NAT-dst, see *"Address Translation" on page 1467*.

WebUI (Device A)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Fixed IP: (select)
      IP Address / Netmask: 10.0.0.1/30

   Network > Interfaces > Edit (for tunnel.1) > DIP > New: Enter the following, then click **OK**:

   ID: 5
   IP Address Range: (select), 10.0.0.2 ~ 10.0.0.2
      Port Translation: (select)
   In the same subnet as the interface IP or its secondary IPs: (select)

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: corp
   IP Address/Domain Name:
      IP/Netmask: (select), 10.1.1.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: oda1
   IP Address/Domain Name:
      IP/Netmask: (select), 10.0.1.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK:**

Address Name: peers
IP Address/Domain Name:
IP/Netmask: (select), 10.0.0.0/16
Zone: Untrust

3. **VPNs**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: peer1
Type: Static IP: (select), Address/Hostname: 2.2.2.2
Preshared Key: netscreen1
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn2
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: peer2
Type: Static IP: (select), Address/Hostname: 3.3.3.3
Preshared Key: netscreen2
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn3
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: peer3
Type: Static IP: (select), Address/Hostname: 4.4.4.4
Preshared Key: netscreen3

Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

4. **Routes**

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.1.0/24
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.3.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 10.0.2.1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.2.2/32
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 10.0.2.1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.5.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 10.0.4.1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.4.2/32
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 10.0.4.1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.7.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 10.0.6.1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.6.2/32
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 10.0.6.1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.0.0/16
Gateway: (select)
Interface: null
Gateway IP Address: 0.0.0.0
Metric: 10

Network > Interfaces > Edit (for tunnel.1) > NHTB > New: Enter the following, then click **Add**:

New Next Hop Entry:
IP Address: 10.0.2.1
VPN: vpn1

Network > Interfaces > Edit (for tunnel.1) > NHTB: Enter the following, then click **Add**:

New Next Hop Entry:
IP Address: 10.0.4.1
VPN: vpn2

Network > Interfaces > Edit (for tunnel.1) > NHTB: Enter the following, then click **Add**:

New Next Hop Entry:
IP Address: 10.0.6.1
VPN: vpn3

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book: (select), corp
Destination Address:
Address Book: (select), peers
Service: Any
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Source Translation: (select)
DIP On: 5 (10.0.0.2–10.0.0.2)/X-late

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), peers
Destination Address:
Address Book Entry: (select), oda1
Service: Any
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Destination Translation: (select)
Translate to IP Range: (select), 10.1.1.0 - 10.1.1.254

CLI (Device A)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.0.0.1/30
set interface tunnel.1 dip 5 10.0.0.2 10.0.0.2
```

2. **Addresses**

```
set address trust corp 10.1.1.0/24
set address trust oda1 10.0.1.0/24
set address untrust peers 10.0.0.0/16
```

3. **VPNs**

```
set ike gateway peer1 address 2.2.2.2 outgoing-interface ethernet3 preshare
netscreen1 sec-level compatible
set vpn vpn1 gateway peer1 sec-level compatible
```

```
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
set ike gateway peer2 address 3.3.3.3 outgoing-interface ethernet3 preshare
netscreen2 sec-level compatible
set vpn vpn2 gateway peer2 sec-level compatible
set vpn vpn2 bind interface tunnel.1
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
set ike gateway peer3 address 4.4.4.4 outgoing-interface ethernet3 preshare
netscreen3 sec-level compatible
set vpn vpn3 gateway peer3 sec-level compatible
set vpn vpn3 bind interface tunnel.1
set vpn vpn3 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
set vrouter trust-vr route 10.0.1.0/24 interface ethernet1
set vrouter trust-vr route 10.0.3.0/24 interface tunnel.1 gateway 10.0.2.1
set vrouter trust-vr route 10.0.2.2/32 interface tunnel.1 gateway 10.0.2.1
set vrouter trust-vr route 10.0.5.0/24 interface tunnel.1 gateway 10.0.4.1
set vrouter trust-vr route 10.0.4.2/32 interface tunnel.1 gateway 10.0.4.1
set vrouter trust-vr route 10.0.7.0/24 interface tunnel.1 gateway 10.0.6.1
set vrouter trust-vr route 10.0.6.2/32 interface tunnel.1 gateway 10.0.6.1
set vrouter trust-vr route 10.0.0.0/16 interface null metric 10
set interface tunnel.1 nhtb 10.0.2.1 vpn vpn1
set interface tunnel.1 nhtb 10.0.4.1 vpn vpn2
set interface tunnel.1 nhtb 10.0.6.1 vpn vpn3
```

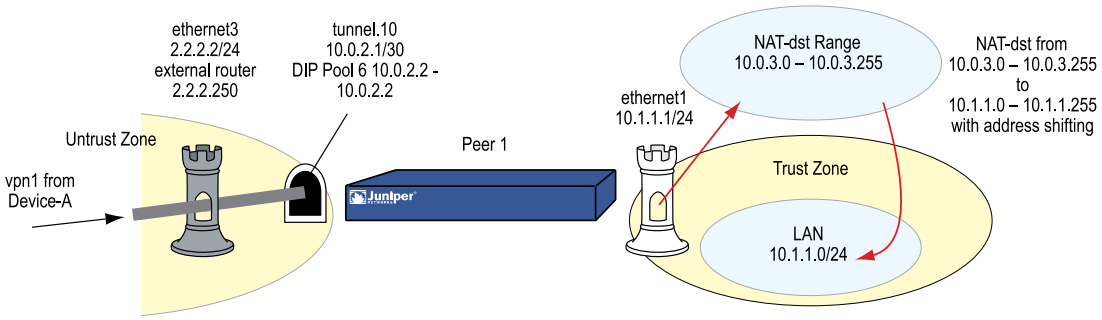
5. **Policies**

```
set policy from trust to untrust corp peers any nat src dip-id 5 permit
set policy from untrust to trust peers oda1 any nat dst ip 10.1.1.0 10.1.1.254
 permit
save
```

Peer1

The following configuration, as illustrated in Figure 267 on page 996, is what the remote admin for the security device at the peer1 site must enter to create a VPN tunnel to Device A at the corporate site. The remote admin configures the security device to perform source and destination NAT (NAT-src and NAT-dst) because the internal addresses are in the same address space as those in the corporate LAN: 10.1.1.0/24. Peer1 performs NAT-src using DIP pool 6 to translate all internal source addresses to 10.0.2.2 when sending traffic through VPN1 to Device A. Peer1 performs NAT-dst on VPN traffic sent from Device A, translating addresses from 10.0.3.0/24 to 10.1.1.0/24 with address shifting in effect.

Figure 267: Peer1 Performing NAT-Dst



NOTE: For more information about NAT-src and NAT-dst, see "Address Translation" on page 1467.

WebUI (Peer1)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 2.2.2.2/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.10
   Zone (VR): Untrust (trust-vr)
   Fixed IP: (select)
   IP Address / Netmask: 10.0.2.1/30

   Network > Interfaces > Edit (for tunnel.10) > DIP > New: Enter the following, then click **OK**:

   ID: 6
   IP Address Range: (select), 10.0.2.2 ~ 10.0.2.2
   Port Translation: (select)
   In the same subnet as the interface IP or its secondary IPs: (select)

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: lan
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: oda2
IP Address/Domain Name:
IP/Netmask: (select), 10.0.3.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: to_corp
IP Address/Domain Name:
IP/Netmask: (select), 10.0.1.0/24
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: fr_corp
IP Address/Domain Name:
IP/Netmask: (select), 10.0.0.2/32
Zone: Untrust

3. **VPN**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: corp
Type: Static IP: (select), Address/Hostname: 1.1.1.1
Preshared Key: netscreen1
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.10
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

4. **Routes**

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 2.2.2.250
Metric: 1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.3.0/24
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 0.0.0.0
Metric: 1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.0.0/8
Gateway: (select)
Interface: tunnel.10
Gateway IP Address: 0.0.0.0
Metric: 10

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK:**

Network Address / Netmask: 10.0.0.0/8
Gateway: (select)
Interface: null
Gateway IP Address: 0.0.0.0
Metric: 12

5. **Policies**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), fr_corp
Destination Address:
Address Book Entry: (select), oda2
Service: Any
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Destination Translation: (select)
Translate to IP Range: (select), 10.1.1.0 - 10.1.1.254

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), lan
Destination Address:
Address Book Entry: (select), to_corp
Service: Any
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Source Translation: (select)
DIP On: 6 (10.0.2.2–10.0.2.2)/X-late

CLI (Peer1)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.10 zone untrust
set interface tunnel.10 ip 10.0.2.1/30
set interface tunnel.10 dip 6 10.0.2.2 10.0.2.2
```

2. **Addresses**

```
set address trust lan 10.1.1.0/24
set address trust oda2 10.0.3.0/24
set address untrust to_corp 10.0.1.0/24
set address untrust fr_corp 10.0.0.2/32
```

3. **VPN**

```
set ike gateway corp address 1.1.1.1 outgoing-interface ethernet3 preshare
netscreen1 sec-level compatible
set vpn vpn1 gateway corp sec-level compatible
set vpn vpn1 bind interface tunnel.10
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
metric 1
set vrouter trust-vr route 10.0.3.0/24 interface ethernet1 metric 1
set vrouter trust-vr route 10.0.0.0/8 interface tunnel.10 metric 10
set vrouter trust-vr route 10.0.0.0/8 interface null metric 12
```

5. **Policies**

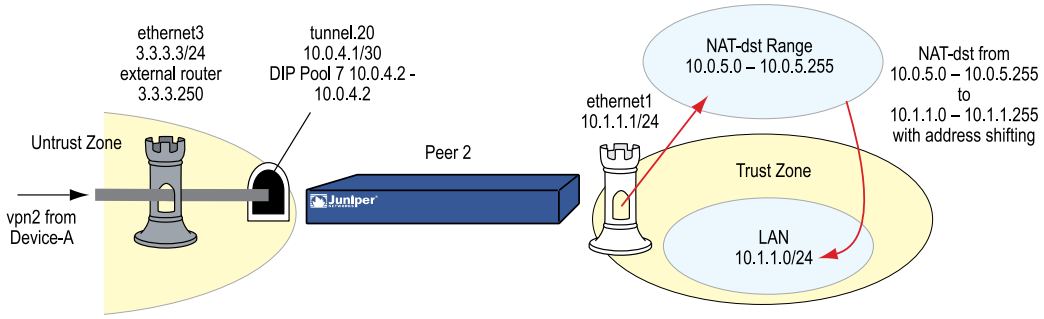
```
set policy from trust to untrust lan to_corp any nat src dip-id 6 permit
set policy from untrust to trust fr_corp oda2 any nat dst ip 10.1.1.0 10.1.1.254
```

```
    permit
save
```

Peer2

The following configuration, as illustrated in Figure 268 on page 1000, is what the remote admin for the security device at the peer2 site must enter to create a VPN tunnel to Device A at the corporate site. The remote admin configures the security device to perform source and destination NAT (NAT-src and NAT-dst) because the internal addresses are in the same address space as those in the corporate LAN: 10.1.1.0/24. Peer2 performs NAT-src using DIP pool 7 to translate all internal source addresses to 10.0.4.2 when sending traffic through VPN2 to Device A. Peer2 performs NAT-dst on VPN traffic sent from Device A, translating addresses from 10.0.5.0/24 to 10.1.1.0/24 with address shifting in effect.

Figure 268: Peer2



NOTE: For more information about NAT-src and NAT-dst, see "Address Translation" on page 1467.

WebUI (Peer2)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 3.3.3.3/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.20
Zone (VR): Untrust (trust-vr)
Fixed IP: (select)
IP Address / Netmask: 10.0.4.1/30

Network > Interfaces > Edit (for tunnel.20) > DIP > New: Enter the following, then click **OK**:

ID: 7
IP Address Range: (select), 10.0.4.2 ~ 10.0.4.2
Port Translation: (select)
In the same subnet as the interface IP or its secondary IPs: (select)

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: lan
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: oda3
IP Address/Domain Name:
IP/Netmask: (select), 10.0.5.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: to_corp
IP Address/Domain Name:
IP/Netmask: (select), 10.0.1.0/24
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK:**

Address Name: fr_corp
IP Address/Domain Name:
IP/Netmask: (select), 10.0.0.2/32
Zone: Untrust

3. **VPN**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn2
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)

Gateway Name: corp
Type: Static IP: (select), Address/Hostname: 1.1.1.1
Preshared Key: netscreen2
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.20
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

4. **Routes**

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 3.3.3.250
Metric: 1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.5.0/24
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 0.0.0.0
Metric: 1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.1.0/24
Gateway: (select)
Interface: tunnel.20
Gateway IP Address: 0.0.0.0
Metric: 10

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK:**

Network Address / Netmask: 10.0.1.0/24
Gateway: (select)
Interface: null
Gateway IP Address: 0.0.0.0
Metric: 12

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), lan
Destination Address:
Address Book Entry: (select), to_corp
Service: Any
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Source Translation: (select)
DIP On: 7 (10.0.4.2–10.0.4.2)/X-late

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), fr_corp
Destination Address:
Address Book Entry: (select), oda3
Service: Any
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Destination Translation: (select)
Translate to IP Range: (select), 10.1.1.0 - 10.1.1.254

CLI (Peer2)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 3.3.3.3/24
set interface tunnel.20 zone untrust
set interface tunnel.20 ip 10.0.4.1/30
set interface tunnel.20 dip 7 10.0.4.2 10.0.4.2
```

2. **Addresses**

```
set address trust lan 10.1.1.0/24
set address trust oda3 10.0.5.0/24
set address untrust to_corp 10.0.1.0/24
set address untrust fr_corp 10.0.0.2/32
```

3. **VPN**

```
set ike gateway corp address 1.1.1.1 outgoing-interface ethernet3 preshare
netscreen2 sec-level compatible
```

```
set vpn vpn2 gateway corp sec-level compatible
set vpn vpn2 bind interface tunnel.20
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 3.3.3.250
metric 1
set vrouter trust-vr route 10.0.5.0/24 interface ethernet1 metric 1
set vrouter trust-vr route 10.0.0.0/8 interface tunnel.20 metric 10
set vrouter trust-vr route 10.0.0.0/8 interface null metric 12
```

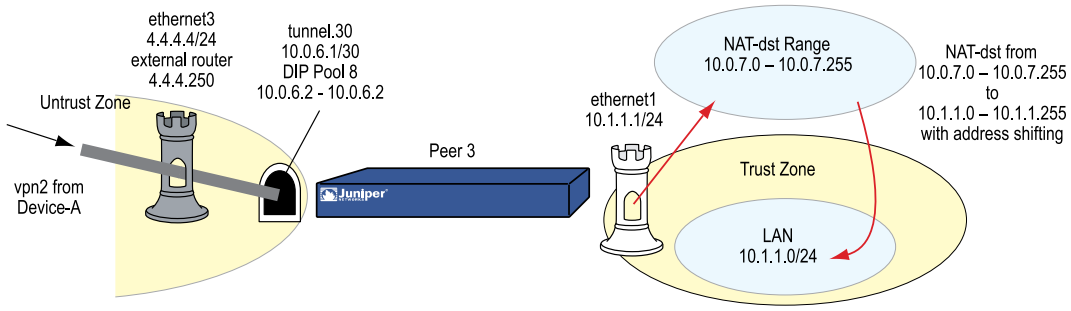
5. **Policies**

```
set policy from trust to untrust lan to_corp any nat src dip-id 7 permit
set policy from untrust to trust fr_corp oda3 any nat dst ip 10.1.1.0 10.1.1.254
 permit
save
```

Peer3

The following configuration, as illustrated in Figure 269 on page 1004, is what the remote admin for the security device at the peer3 site must enter to create a VPN tunnel to Device A at the corporate site. The remote admin configures the security device to perform source and destination NAT (NAT-src and NAT-dst) because the internal addresses are in the same address space as those in the corporate LAN: 10.1.1.0/24. Peer3 performs NAT-src using DIP pool 8 to translate all internal source addresses to 10.0.6.2 when sending traffic through VPN3 to Device A. Peer3 performs NAT-dst on VPN traffic sent from Device A, translating addresses from 10.0.7.0/24 to 10.1.1.0/24 with address shifting in effect.

Figure 269: Peer3



NOTE: For more information about NAT-dst, see *"Address Translation" on page 1467*.

WebUI (Peer3)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24
Enter the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 4.4.4.4/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.30
Zone (VR): Untrust (trust-vr)
Fixed IP: (select)
IP Address / Netmask: 10.0.6.1/30

Network > Interfaces > Edit (for tunnel.320) > DIP > New: Enter the following, then click **OK**:

ID: 7
IP Address Range: (select), 10.0.6.2 ~ 10.0.6.2
Port Translation: (select)
In the same subnet as the interface IP or its secondary IPs: (select)

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: lan
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: oda4
IP Address/Domain Name:
IP/Netmask: (select), 10.0.7.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: to_corp
IP Address/Domain Name:

IP/Netmask: (select), 10.0.1.0/24
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK:**

Address Name: fr_corp
IP Address/Domain Name:
IP/Netmask: (select), 10.0.0.2/32
Zone: Untrust

3. **VPN**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn3
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: corp
Type: Static IP: (select), Address/Hostname: 1.1.1.1
Preshared Key: netscreen3
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.30
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

4. **Routes**

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 4.4.4.250
Metric: 1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.7.0/24
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 0.0.0.0
Metric: 1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.0.0.0/8
Gateway: (select)
Interface: tunnel.20
Gateway IP Address: 10.0.0.1
Metric: 10

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK:**

Network Address / Netmask: 10.0.0.0/8
Gateway: (select)
Interface: null
Gateway IP Address: 10.0.0.1
Metric: 12

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), lan
Destination Address:
Address Book Entry: (select), to_corp
Service: Any
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Source Translation: (select)
DIP On: 8 (10.0.6.2–10.0.6.2)/X-late

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), fr_corp
Destination Address:
Address Book Entry: (select), oda4
Service: Any
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Destination Translation: (select)
Translate to IP Range: (select), 10.1.1.0 - 10.1.1.254

CLI (Peer3)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 4.4.4.4/24
set interface tunnel.30 zone untrust
set interface tunnel.30 ip 10.0.6.1/30
set interface tunnel.30 dip 8 10.0.6.2 10.0.6.2
```

2. **Addresses**

```
set address trust lan 10.1.1.0/24
set address trust oda4 10.0.7.0/24
set address untrust to_corp 10.0.1.0/24
set address untrust fr_corp 10.0.0.2/32
```

3. **VPN**

```
set ike gateway corp address 1.1.1.1 outgoing-interface ethernet3 preshare
netscreen3 sec-level compatible
set vpn vpn3 gateway corp sec-level compatible
set vpn vpn3 bind interface tunnel.30
set vpn vpn3 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 3.3.3.250 metric
 1
set vrouter trust-vr route 10.0.7.0/24 interface ethernet1 metric 1
set vrouter trust-vr route 10.0.0.0/8 interface tunnel.30 metric 10
set vrouter trust-vr route 10.0.0.0/8 interface null metric 12
```

5. **Policies**

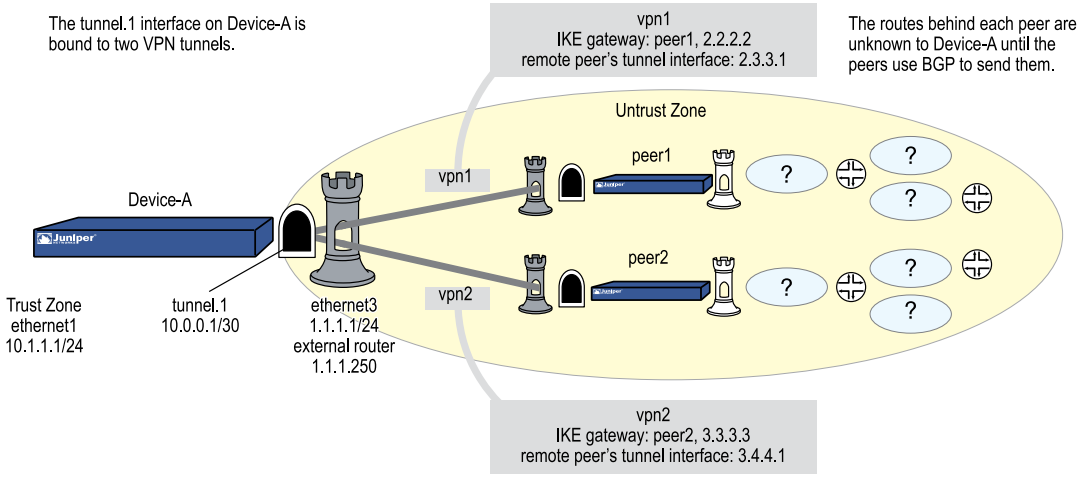
```
set policy from trust to untrust lan to_corp any nat src dip-id 8 permit
set policy from untrust to trust fr_corp oda4 any nat dst ip 10.1.1.0 10.1.1.254
 permit
save
```

Binding Automatic Route and NHTB Table Entries

In Figure 270 on page 1009, you bind two route-based AutoKey IKE VPN tunnels—vpn1, vpn2—to a single tunnel interface—tunnel.1 on Device A at the corporate site. The network that each remote peer protects has multiple routes behind the connected route. Using Border Gateway Protocol (BGP), the peers communicate their routes to Device A. This example permits VPN traffic from the corporate site behind Device A to the peer sites.

NOTE: You can also use Open Shortest Path First (OSPF) instead of BGP as the routing protocol in this example. See “Using OSPF for Automatic Route Table Entries” on page 1020 for the OSPF configurations.

Figure 270: Automatic Route and NHTB Table Entries (Device A)



The VPN tunnel configurations at both ends of each tunnel use the following parameters: AutoKey IKE, preshared key (peer1: "netscreen1", peer2: "netscreen2"), and the security level predefined as "Compatible" for both Phase 1 and Phase 2 proposals. (For details about these proposals, see "Tunnel Negotiation" on page 715.)

By configuring the following two features, you can enable Device A to populate its NHTB and route tables automatically:

- VPN monitoring with the rekey option (or the IKE heartbeats reconnect option)
- BGP dynamic routing on tunnel.1

NOTE: If you are running a dynamic routing protocol on the tunnel interfaces, traffic generated by the protocol can trigger IKE negotiations even without enabling VPN monitoring with the rekey option or enabling the IKE heartbeat reconnect option. Still, Juniper Networks recommends that you not rely on dynamic routing traffic to trigger IKE negotiations. Instead use VPN monitoring with the rekey option or the IKE heartbeat reconnect option.

If you are running BGP on the tunnel interfaces, the BGP-generated traffic can trigger IKE negotiations even without enabling VPN monitoring with the rekey option or enabling the IKE heartbeat reconnect option. Still, Juniper Networks recommends that you not rely on BGP traffic to trigger IKE negotiations. Instead, use VPN monitoring with the rekey option or the IKE heartbeat reconnect option.

When you enable VPN monitoring with the rekey option for an AutoKey IKE VPN tunnel, Device A establishes a VPN connection with its remote peer as soon as you and the admin at the remote site finish configuring the tunnel. The devices do not wait for user-generated VPN traffic to perform IKE negotiations. During Phase 2 negotiations, the security devices exchange their tunnel interface IP address, so that Device A can automatically make a VPN-to-next-hop mapping in its NHTB table.

The rekey option ensures that when the Phase 1 and Phase 2 key lifetimes expire, the devices automatically negotiate the generation of new keys without the need for human intervention. VPN monitoring with the rekey option enabled essentially provides a means for keeping a VPN tunnel up continually, even when there is no user-generated traffic. This is necessary so that the BGP dynamic routing instances that you and the remote admins create and enable on the tunnel interfaces at both ends of the tunnels can send routing information to Device A and automatically populate its route table with the routes it needs to direct traffic through the VPN tunnel before those routes are required for user-generated traffic. (The admins at the peer sites still need to enter a single static route to the rest of the virtual private network through the tunnel interface at each respective site.)

You enter a default route and static routes on Device A to reach its BGP neighbors through the correct VPN tunnels. All security zones and interfaces on each device are in the trust-vr virtual routing domain for that device.

WebUI (Device A)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Fixed IP: (select)
   IP Address / Netmask: 10.0.0.1/30

2. **VPNs**

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: vpn1
   Security Level: Compatible
   Remote Gateway: Create a Simple Gateway: (select)
   Gateway Name: peer1
   Type: Static IP: (select), Address/Hostname: 2.2.2.2
   Preshared Key: netscreen1
   Security Level: Compatible
   Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY
VPN Monitor: (select)
Rekey: (select)

NOTE: Leave the Source Interface and Destination IP options at their default settings. For information about these options, see "VPN Monitoring" on page 971.

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn2
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: peer2
Type: Static IP: (select), Address/Hostname: 3.3.3.3
Preshared Key: netscreen2
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY
VPN Monitor: (select)
Rekey: (select)

NOTE: Leave the Source Interface and Destination IP options at their default settings. For information about these options, see "VPN Monitoring" on page 971.

3. **Static Route**

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 2.3.3.1/32
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 2.3.3.1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 3.4.4.1/32
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 3.4.4.1

4. **Dynamic Routing**

Network > Routing > Virtual Routers > Edit (for trust-vr) > Create BGP Instance: Enter the following, then click **OK**:

AS Number (required): 99
BGP Enabled: (select)

Network > Interfaces > Edit (for tunnel.1) > BGP: Select the Protocol BGP check box, then click **OK**.

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors: Enter the following, then click **Add**:

AS Number: 99
Remote IP: 2.3.3.1
Outgoing Interface: tunnel.1

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors: Enter the following, then click **Add**:

AS Number: 99
Remote IP: 3.4.4.1
Outgoing Interface: tunnel.1

5. **Policy**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book: (select), Any
Destination Address:
Address Book: (select), Any
Service: ANY
Action: Permit

CLI (Device A)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
```

```
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.0.0.1/30
```

2. **VPNs**

```
set ike gateway peer1 address 2.2.2.2 outgoing-interface ethernet3 preshare
netscreen1 sec-level compatible
set vpn vpn1 gateway peer1 sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
set vpn vpn1 monitor rekey
set ike gateway peer2 address 3.3.3.3 outgoing-interface ethernet3 preshare
netscreen2 sec-level compatible
set vpn vpn2 gateway peer2 sec-level compatible
set vpn vpn2 bind interface tunnel.1
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
set vpn vpn2 monitor rekey
```

3. **Static Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
set vrouter trust-vr route 2.3.3.1/32 interface tunnel.1 gateway 2.3.3.1
set vrouter trust-vr route 2.4.4.1/32 interface tunnel.1 gateway 2.4.4.1
```

4. **Dynamic Routing**

```
device-> set vrouter trust-vr protocol bgp 99
device-> set vrouter trust-vr protocol bgp enable
device-> set interface tunnel.1 protocol bgp
device-> set vrouter trust-vr
device(trust-vr)-> set protocol bgp
device(trust-vr/bgp)-> set neighbor 2.3.3.1 remote-as 99 outgoing interface
tunnel.1
device(trust-vr/bgp)-> set neighbor 2.3.3.1 enable
device(trust-vr/bgp)-> set neighbor 3.4.4.1 remote-as 99 outgoing interface
tunnel.1
device(trust-vr/bgp)-> set neighbor 3.4.4.1 enable
device(trust-vr/bgp)-> exit
device(trust-vr)-> exit
```

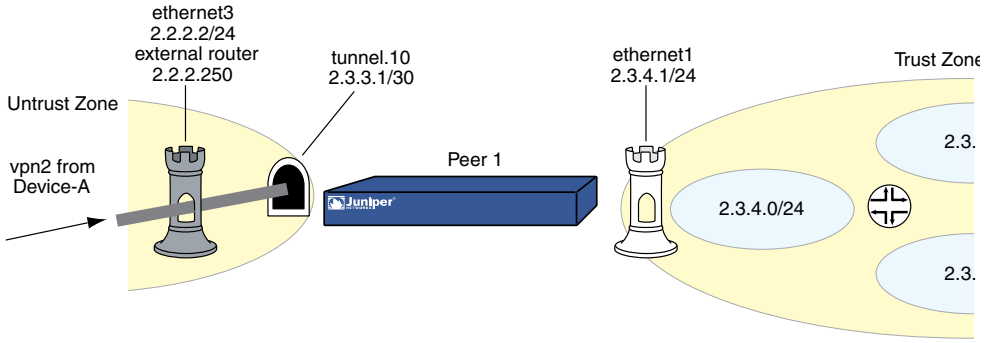
5. **Policy**

```
set policy from trust to untrust any any any permit
save
```

Peer1

The following configuration, as illustrated in Figure 271 on page 1014, is what the remote admin for the security device at the peer1 site must enter to create a VPN tunnel to Device A at the corporate site. The remote admin configures the security device to permit inbound traffic from the corporate site and to communicate internal routes to its BGP neighbor through vpn1.

Figure 271: Peer1



WebUI (Peer1)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 2.3.4.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 2.2.2.2/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.10
   Zone (VR): Untrust (trust-vr)
   Fixed IP: (select)
   IP Address / Netmask: 2.3.3.1/30

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: corp
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.0/24
   Zone: Untrust

3. **VPN**

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: vpn1
   Security Level: Compatible
   Remote Gateway: Create a Simple Gateway: (select)
   Gateway Name: corp

Type: Static IP: (select), Address/Hostname: 1.1.1.1
Preshared Key: netscreen1
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to: Tunnel Interface, tunnel.10
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

4. **Static Routes**

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 2.2.2.250
Metric: 1

Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.1.1.0/24
Gateway: (select)
Interface: tunnel.10
Gateway IP Address: 0.0.0.0
Metric: 1

5. **Dynamic Routing**

Network > Routing > Virtual Routers > Edit (for trust-vr) > Create BGP Instance: Enter the following, then click **OK**:

AS Number (required): 99
BGP Enabled: (select)

Network > Interfaces > Edit (for tunnel.10) > BGP: Select the Protocol BGP check box, then click **OK**.

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors: Enter the following, then click **Add**:

AS Number: 99
Remote IP: 10.0.0.1
Outgoing Interface: tunnel.10

6. **Policy**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), corp
Destination Address:
  Address Book Entry: (select), Any
Service: ANY
Action: Permit

CLI (Peer1)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 2.3.4.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.10 zone untrust
set interface tunnel.10 ip 2.3.3.1/30
```

2. **Address**

```
set address untrust corp 10.1.1.0/24
```

3. **VPN**

```
set ike gateway corp address 1.1.1.1 outgoing-interface ethernet3 preshare
netscreen1 sec-level compatible
set vpn vpn1 gateway corp sec-level compatible
set vpn vpn1 bind interface tunnel.10
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

4. **Static Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
metric 1
set vrouter trust-vr route 10.1.1.0/24 interface tunnel.10 metric 1
```

5. **Dynamic Routing**

```
device-> set vrouter trust-vr protocol bgp 99
device-> set vrouter trust-vr protocol bgp enable
device-> set interface tunnel.10 protocol bgp
device-> set vrouter trust-vr
device(trust-vr)-> set protocol bgp
device(trust-vr/bgp)-> set neighbor 10.0.0.1 remote-as 99 outgoing interface
tunnel.10
device(trust-vr/bgp)-> set neighbor 10.0.0.1 enable
device(trust-vr/bgp)-> exit
device(trust-vr)-> exit
```

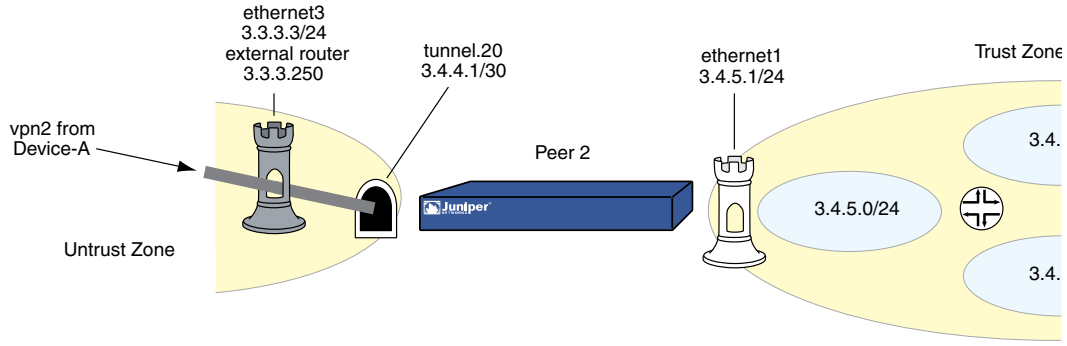
6. **Policy**

```
set policy from untrust to trust corp any any permit
save
```

Peer2

The following configuration, as illustrated in Figure 272 on page 1017, is what the remote admin for the security device at the peer2 site must enter to create a VPN tunnel to Device A at the corporate site. The remote admin configures the security device to permit inbound traffic from the corporate site and communicate internal routes to its BGP neighbor through vpn2.

Figure 272: Peer2



WebUI (Peer2)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 2.3.4.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 3.3.3.3/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.20
   Zone (VR): Untrust (trust-vr)
   Fixed IP: (select)
   IP Address / Netmask: 3.4.4.1/30

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: corp
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.0/24
   Zone: Untrust

3. **VPN**

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: vpn2
   Security Level: Compatible
   Remote Gateway: Create a Simple Gateway: (select)
   Gateway Name: corp
   Type: Static IP: (select), Address/Hostname: 1.1.1.1
   Preshared Key: netscreen2
   Security Level: Compatible
   Outgoing Interface: ethernet3

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

   Bind to: Tunnel Interface, tunnel.20
   Proxy-ID: (select)
   Local IP / Netmask: 0.0.0.0/0
   Remote IP / Netmask: 0.0.0.0/0
   Service: ANY

4. **Static Routes**

   Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

   Network Address / Netmask: 0.0.0.0/0
   Gateway: (select)
   Interface: ethernet3
   Gateway IP Address: 3.3.3.250
   Metric: 1

   Network > Routing > Routing Entries > (trust-vr) New: Enter the following, then click **OK**:

   Network Address / Netmask: 10.1.1.0/24
   Gateway: (select)
   Interface: tunnel.20
   Gateway IP Address: 0.0.0.0
   Metric: 1

5. **Dynamic Routing**

   Network > Routing > Virtual Routers > Edit (for trust-vr) > Create BGP Instance: Enter the following, then click **OK**:

   AS Number (required): 99
   BGP Enabled: (select)

   Network > Interfaces > Edit (for tunnel.20) > BGP: Select the Protocol BGP check box, then click **OK**.

   Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors: Enter the following, then click **Add**:

AS Number: 99
Remote IP: 10.0.0.1
Outgoing Interface: tunnel.20

6. **Policy**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), corp
Destination Address:
  Address Book Entry: (select), Any
Service: ANY
Action: Permit

CLI (Peer2)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 3.4.5.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 3.3.3.3/24
set interface tunnel.20 zone untrust
set interface tunnel.20 ip 3.4.4.1/30
```

2. **Address**

```
set address untrust corp 10.1.1.0/24
```

3. **VPN**

```
set ike gateway corp address 1.1.1.1 outgoing-interface ethernet3 preshare
netscreen2 sec-level compatible
set vpn vpn1 gateway corp sec-level compatible
set vpn vpn1 bind interface tunnel.20
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

4. **Static Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 3.3.3.250
metric 1
set vrouter trust-vr route 10.1.1.0/24 interface tunnel.20 metric 1
```

5. **Dynamic Routing**

```
device-> set vrouter trust-vr protocol bgp 99
device-> set vrouter trust-vr protocol bgp enable
device-> set interface tunnel.20 protocol bgp
device-> set vrouter trust-vr
device(trust-vr)-> set protocol bgp
device(trust-vr/bgp)-> set neighbor 10.0.0.1 remote-as 99 outgoing interface
tunnel.20
device(trust-vr/bgp)-> set neighbor 10.0.0.1 enable
```

```
device(trust-vr/bgp)-> exit
device(trust-vr)-> exit
```

6. **Policy**

```
set policy from untrust to trust corp any any permit
save
```

Using OSPF for Automatic Route Table Entries

You can also configure OSPF instead of BGP dynamic routing for the peers to communicate routes to Device A. To allow tunnel.1 on Device A to form OSPF adjacencies with its peers, you must configure the tunnel interface as a point-to-multipoint interface. The OSPF dynamic routing configuration for each device is shown below.

WebUI (Device A)

Dynamic Routing (OSPF)

Network > Routing > Virtual Routers > Edit (for trust-vr) > Create OSPF Instance: Select **OSPF Enabled**, then click **Apply**.

Area > Configure (for area 0.0.0.0): Click **< < Add** to move the tunnel.1 interface from the Available Interface(s) list to the Selected Interface(s) list, then click **OK**.

Network > Interfaces > Edit (for tunnel.1) > OSPF: Enter the following, then click **Apply**:

```
Bind to Area: (select), Select 0.0.0.0 from the drop down list
Protocol OSPF: Enable
Link Type: Point-to-Multipoint (select)
```

CLI (Device A)

Dynamic Routing (OSPF)

```
device-> set vrouter trust-vr protocol ospf
device-> set vrouter trust-vr protocol ospf enable
device-> set interface tunnel.1 protocol ospf area 0
device-> set interface tunnel.1 protocol ospf link-type p2mp
device-> set interface tunnel.1 protocol ospf enable
device-> save
```

WebUI (Peer1)

Dynamic Routing (OSPF)

Network > Routing > Virtual Routers > Edit (for trust-vr) > Create OSPF Instance: Select **OSPF Enabled**, then click **Apply**.

Area > Configure (for area 0.0.0.0): Click **< < Add** to move the tunnel.1 interface from the Available Interface(s) list to the Selected Interface(s) list, then click **OK**.

Network > Interfaces > Edit (for tunnel.1) > OSPF: Enter the following, then click **Apply**:

Bind to Area: (select), Select 0.0.0.0 from the drop down list
Protocol OSPF: Enable

CLI (Peer1)

Dynamic Routing (OSPF)

```
device-> set vrouter trust-vr protocol ospf
device-> set vrouter trust-vr protocol ospf enable
device-> set interface tunnel.1 protocol ospf area 0
device-> set interface tunnel.1 protocol ospf enable
device-> save
```

WebUI (Peer2)

Dynamic Routing (OSPF)

Network > Routing > Virtual Routers > Edit (for trust-vr) > Create OSPF Instance: Select **OSPF Enabled**, then click **Apply**.

Area > Configure (for area 0.0.0.0): Click **< < Add** to move the tunnel.1 interface from the Available Interface(s) list to the Selected Interface(s) list, then click **OK**.

Network > Interfaces > Edit (for tunnel.1) > OSPF: Enter the following, then click **Apply**:

Bind to Area: (select), Select 0.0.0.0 from the drop down list
Protocol OSPF: Enable

CLI (Peer2)

Dynamic Routing (OSPF)

```
device-> set vrouter trust-vr protocol ospf
device-> set vrouter trust-vr protocol ospf enable
device-> set interface tunnel.1 protocol ospf area 0
device-> set interface tunnel.1 protocol ospf enable
device-> save
```

Multiple Proxy IDs on a Route-Based VPN

When multiple tunnels exist between peers, the security device cannot use the route and NHTB tables to direct the traffic through a particular tunnel. In such cases, the security device uses proxy IDs to direct the traffic, as defined in RFC 2409. A proxy ID is a kind of agreement between IKE peers to permit traffic through a tunnel if the traffic matches a specified tuple of local address, remote address, and service. You can define multiple proxy IDs for a route-based VPN. To make the security device

using proxy IDs for routing the traffic, you use the **set vpn string proxy-id check** command. If only one proxy ID is defined for a route-based VPN, you can enable or disable the proxy-id check. If you enable the check, traffic that does not match the proxy ID is discarded.

NOTE: If more than one proxy ID is defined for a route-based VPN, a proxy-ID check is always performed, even if it is disabled.

For each proxy ID, a specific tunnel and Phase 2 SA are negotiated. When traffic matching a proxy ID arrives, the security device uses the tunnel and the Phase 2 SA associated with that proxy ID to route the traffic. Multiple proxy IDs are not supported on the following:

- Policy-based VPN
- Dialup VPN
- Manual VPN
- VPN group
- GRE-over-IPSec VPN
- L2TP-over-IPSec VPN
- Transport mode VPN

In the case of a hub-and-spoke network topology, you should define both hub-to-spoke and spoke-to-spoke proxy-ID configurations to enable communication between the spokes.

You can define a proxy ID through the WebUI or the CLI. You can use either an IP address or an address name of the local and remote device to define a proxy ID. In the following example, you define multiple proxy IDs and enable a proxy-ID check on vpn1 and vpn2. To simplify the example, it does not show the steps involved in configuring a route-based VPN.

WebUI (Device A)

1. **Defining Multiple Proxy IDs Using an IP Address**

   VPNs > AutoKey IKE > Proxy IDs (for vpn1): Enter the following proxy-ID entries, then click **New**:

   Local IP: (select)
   IP/Netmask: 10.1.1.0/24
   Remote IP: (select)
   IP/Netmask: 20.1.1.0/24
   Service: ANY

   VPNs > AutoKey IKE > Proxy IDs (for vpn1): Enter the following proxy-ID entries, then click **New**:

Local IP: (select)
IP/Netmask: 10.1.1.0/24
Remote IP: (select)
IP/Netmask: 20.1.2.0/24
Service: ANY

VPNs > AutoKey IKE > Proxy IDs (for vpn1): Enter the following proxy-ID entries, then click **New**:

Local IP: (select)
IP/Netmask: 10.1.2.0/24
Remote IP: (select)
IP/Netmask: 20.1.1.0/24
Service: ANY

VPNs > AutoKey IKE > Proxy IDs (for vpn1): Enter the following proxy-ID entries, then click **New**:

Local IP: (select)
IP/Netmask: 10.1.2.0/24
Remote IP: (select)
IP/Netmask: 20.1.2.0/24
Service: ANY

2. **Defining Multiple Proxy IDs Using the Address Book**

VPNs > AutoKey IKE > Proxy IDs (for vpn1): Enter the following proxy-ID entries, then click **New**:

Local Address: (select)
Zone: trust
Address: vpn_local
Remote Address: (select)
Zone: untrust
Address: vpn_remote
Service: ANY

NOTE: Before you can use this option, you should have the IP addresses defined in the address book. The address book defined by a DNS domain name is not supported.

3. **Enabling a Proxy-ID Check on a Route-Based VPN**

VPNs > AutoKey IKE > Advanced (for vpn1): Enter the following, then click **Return**:

Proxy-ID Check: (select)

CLI (Device A)

1. **Defining Multiple Proxy IDs Using an IP Address**

```
set vpn vpn1 proxy-id local-ip 10.1.1.0/24 remote-ip 20.1.1.0/24 any
set vpn vpn1 proxy-id local-ip 10.1.1.0/24 remote-ip 20.1.2.0/24 any
set vpn vpn1 proxy-id local-ip 10.1.2.0/24 remote-ip 20.1.1.0/24 any
set vpn vpn1 proxy-id local-ip 10.1.2.0/24 remote-ip 20.1.2.0/24 any
```

2. **Defining Multiple Proxy IDs Using the Address Book**

```
set vpn vpn1 proxy-id local-addr trust vpn_local remote-addr untrust vpn_remote
 any
```

NOTE: Before you can use this command, you should have the IP addresses defined in the address book. The address book defined by a DNS domain name is not supported.

3. **Enabling a Proxy-ID Check on a Route-Based VPN**

```
set vpn vpn1 proxy-id check
```

WebUI (Device B)

1. **Defining Multiple Proxy IDs Using an IP Address**

VPNs > AutoKey IKE > Proxy IDs (for vpn2): Enter the following proxy-ID entries, then click **New**:

Local IP: (select)
IP/Netmask: 20.1.1.0/24
Remote IP: (select)
IP/Netmask: 10.1.1.0/24
Service: ANY

VPNs > AutoKey IKE > Proxy IDs (for vpn2): Enter the following proxy-ID entries, then click **New**:

Local IP: (select)
IP/Netmask: 20.1.1.0/24
Remote IP: (select)
IP/Netmask: 10.1.2.0/24
Service: ANY

VPNs > AutoKey IKE > Proxy IDs (for vpn2): Enter the following proxy-ID entries, then click **New**:

Local IP: (select)
IP/Netmask: 20.1.2.0/24
Remote IP: (select)
IP/Netmask: 10.1.1.0/24
Service: ANY

VPNs > AutoKey IKE > Proxy IDs (for vpn2): Enter the following proxy-ID entries, then click **New**:

Local IP: (select)
IP/Netmask: 20.1.2.0/24
Remote IP: (select)
IP/Netmask: 10.1.2.0/24
Service: ANY

2. **Defining Multiple Proxy IDs Using the Address Book**

   VPNs > AutoKey IKE > Proxy IDs (for vpn2): Enter the following proxy-ID entries, then click **New**:

   Local Address: (select)
   Zone: trust
   Address: vpn_local
   Remote Address: (select)
   Zone: untrust
   Address: vpn_remote
   Service: ANY

NOTE: Before you can use this option, you should have the IP addresses defined in the address book. The address book defined by a DNS domain name is not supported.

3. **Enabling a Proxy-ID Check on a Route-Based VPN**

   VPNs > AutoKey IKE > Advanced (for vpn2): Enter the following, then click **Return**:

   Proxy-ID Check: (select)

CLI (Device B)

1. **Defining Multiple Proxy IDs Using an IP Address**

```
set vpn vpn2 proxy-id local-ip 20.1.1.0/24 remote-ip 10.1.1.0/24 any
set vpn vpn2 proxy-id local-ip 20.1.1.0/24 remote-ip 10.1.2.0/24 any
set vpn vpn2 proxy-id local-ip 20.1.2.0/24 remote-ip 10.1.1.0/24 any
set vpn vpn2 proxy-id local-ip 20.1.2.0/24 remote-ip 10.1.2.0/24 any
```

2. **Defining Multiple Proxy IDs Using the Address Book**

   set vpn vpn2 proxy-id local-addr *trust vpn_local* remote-addr *untrust vpn_remote any*

NOTE: Before you can use this command, you should have the IP addresses defined in the address book. The address book defined by a DNS domain name is not supported.

3. **Enabling a Proxy-ID Check on a Route-Based VPN**

```
set vpn vpn2 proxy-id check
```

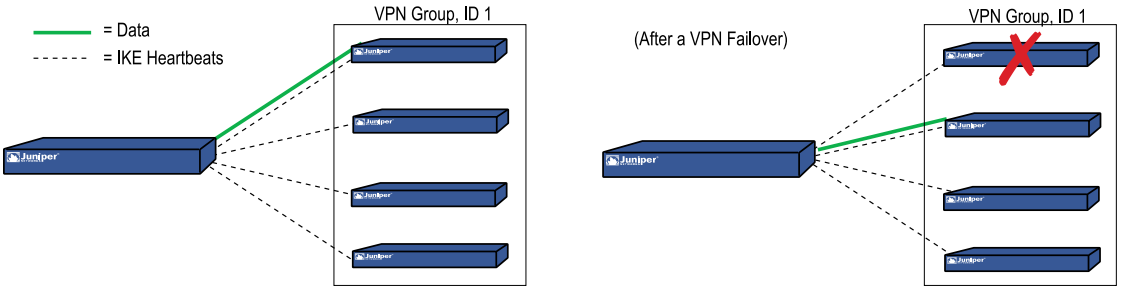
Redundant VPN Gateways

The redundant gateway feature provides a solution for continuous VPN connectivity during and after a site-to-site failover. You can create a VPN group to provide a set of up to four redundant gateways to which policy-based site-to-site or site-to-site dynamic peer AutoKey IKE IPsec VPN tunnels can connect. When the security device first receives traffic matching a policy referencing a VPN group, it performs Phase 1 and Phase 2 IKE negotiations with all members in that group. The security device sends data through the VPN tunnel to the gateway with the highest priority, or weight, in the group. For all other gateways in the group, the security device maintains the Phase 1 and 2 SAs and keeps the tunnels active by sending IKE keepalive packets through them. If the active VPN tunnel fails, the tunnel can fail over to the tunnel and gateway with the second highest priority in the group.

NOTE: VPN groups do not support L2TP, L2TP-over-IPsec, dialup, Manual Key, or route-based VPN tunnel types. In a Site-to-Site Dynamic Peer arrangement, the security device monitoring the VPN group must be the one whose untrust IP address is dynamically assigned, while the untrust IP addresses of the VPN group members must be static.

This scheme assumes that the sites behind the redundant gateways are connected so that data is mirrored among hosts at all sites. Furthermore, each site—being dedicated to high availability (HA)—has a redundant cluster of security devices operating in HA mode. Therefore, the VPN failover threshold must be set higher than the device failover threshold or VPN failovers might occur unnecessarily.

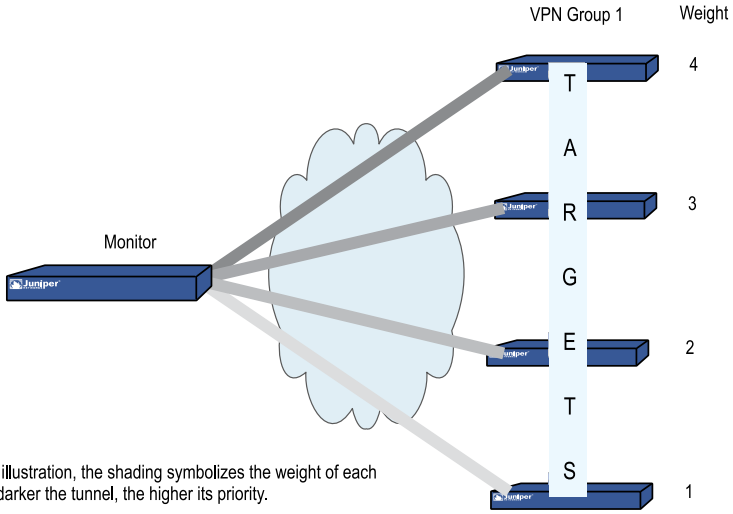
Figure 273: Redundant VPN Gateways for VPN Tunnel Failover



VPN Groups

A VPN group is a set of VPN tunnel configurations for up to four targeted remote gateways. The Phase 1 and Phase 2 security association (SA) parameters for each tunnel in a group can be different or identical (except for the IP address of the remote gateway, which obviously must be different). The VPN group, shown in Figure 274 on page 1027, has a unique ID number, and each member in the group is assigned a unique weight to indicate its place in rank of preference to be the active tunnel. A value of 1 indicates the lowest, or least-preferred, ranking.

Figure 274: Targeted Remote Gateways



The security device communicating with VPN group members and the members themselves have a monitor-to-target relationship. The monitoring device continually monitors the connectivity and wellbeing of each targeted device. The tools that the monitor uses to do this are as follows:

- IKE heartbeats
- IKE recovery attempts

Both tools are presented in the next section, “Monitoring Mechanisms” on page 1027.

NOTE: The monitor-to-target relationship need not be one way. The monitoring device might also be a member of a VPN group and thus be the target of another monitoring device.

Monitoring Mechanisms

Two mechanisms monitor members of a VPN group to determine their ability to terminate VPN traffic:

- IKE heartbeats
- IKE recovery attempts

Using these two tools, plus the TCP application failover option (see “TCP SYN-Flag Checking” on page 1031), security devices can detect when a VPN failover is required and shift traffic to the new tunnel without disrupting VPN service.

IKE Heartbeats

IKE heartbeats are hello messages that IKE peers send to each other under the protection of an established Phase 1 security association (SA) to confirm the

connectivity and wellbeing of the other. If, for example, device_m (the “monitor”) does not receive a specified number of heartbeats (the default is 5) from device_t (the “target”), device_m concludes that device_t is down. Device_m clears the corresponding Phase 1 and Phase 2 security associations (SAs) from its SA cache and begins the IKE recovery procedure. (See “IKE Recovery Procedure” on page 1030.) Device_t also clears its SAs.

NOTE: The IKE heartbeats feature must be enabled on the devices at both ends of a VPN tunnel in a VPN group. If it is enabled on device_m but not on device_t, device_m suppresses IKE heartbeat transmission and generates the following message in the event log: “Heartbeats have been disabled because the peer is not sending them.”

Figure 275: IKE Heartbeats Flow in Both Directions

To define the IKE heartbeat interval and threshold for a specified VPN tunnel (the default is 5), do the following:

WebUI

VPNs > AutoKey Advanced > Gateway > Edit (for the gateway whose IKE heartbeat threshold you want to modify) > Advanced: Enter the new values in the Heartbeat Hello and Heartbeat Threshold fields, then click **OK**.

CLI

```
set ike gateway name_str heartbeat hello number
set ike gateway name_str heartbeat threshold number
```

Dead Peer Detection

DPD is a protocol that network devices use to verify the current existence and availability of other peer devices.

You can use DPD as an alternative to the IKE heartbeat feature (described above). However, you cannot use both features simultaneously. In addition, IKE heartbeat can be a global setting, which affects all IKE gateways configured on the device. The IKE heartbeat setting can also apply to an individual IKE gateway context, which affects an individual gateway only. By contrast, you can configure DPD only in an individual IKE gateway context, not as a global parameter.

A device performs DPD verification by sending encrypted IKE Phase 1 notification payloads (R-U-THERE) to peers and waiting for DPD acknowledgements (R-U-THERE-ACK) from the peers. The device sends an R-U-THERE request only if it has not received any traffic from the peer during a specified DPD interval. If a DPD-enabled device receives traffic on a tunnel, it resets its R-U-THERE counter for

that tunnel, thus starting a new interval. If the device receives an R-U-THERE-ACK from the peer during this interval, it considers the peer alive. If the device does not receive an R-U-THERE-ACK response during the interval, it considers the peer dead.

When the device changes the status of a peer device to be dead, the device removes the Phase 1 SA and all Phase 2 SAs for that peer. In previous ScreenOS releases, the failed VPN tunnel could not fail over to another tunnel and gateway in the group. To maintain continuous connectivity even if a peer goes dead, in the current ScreenOS release the dead peer fails over the tunnel to another group member with the second highest weight. Meanwhile, the device attempts to renegotiate the tunnel with the peer identified as dead. Once the tunnel is successfully negotiated, the tunnel automatically fails back to the first member. The weighting system always causes the best ranked gateway in the group to handle the VPN data whenever possible.

You can configure the following DPD parameters, either through the CLI or the WebUI:

- The **interval** parameter specifies the DPD interval. This interval is the amount of time (expressed in seconds) the device allows to pass before considering a peer to be dead.
- The **always-send** parameter instructs the device to send DPD requests regardless of whether there is IPsec traffic with the peer.
- The **retry** parameter specifies the maximum number of times to send the R-U-THERE request before considering the peer to be dead. As with an IKE heartbeat configuration, the default number of transmissions is 5 times, with a permissible range of 1-128 retries. A setting of zero disables DPD.
- The **reconnect** parameter instructs the device to renegotiate the tunnel at the specified interval with the peer considered to be dead. As with an IKE recovery configuration, the minimum setting is 60 seconds. You can set the interval at any value between 60 and 9999 seconds.

In the following example you create a gateway that uses a DPD interval of five seconds.

WebUI

VPNs > AutoKey Advanced > Gateway > Edit: Create a gateway by entering the following values, then clicking **OK**.

```
Gateway Name: our_gateway
Security Level: Standard
Remote Gateway Type: Static IP Address
   IP Address/Hostname: 1.1.1.1
Preshared Key: jun9345
```

VPNs > AutoKey Advanced > Gateway > Edit (our_gateway): Enter the following values, then click **OK**.

```
Predefined: Standard (select)
DPD:
   Interval: 5
```

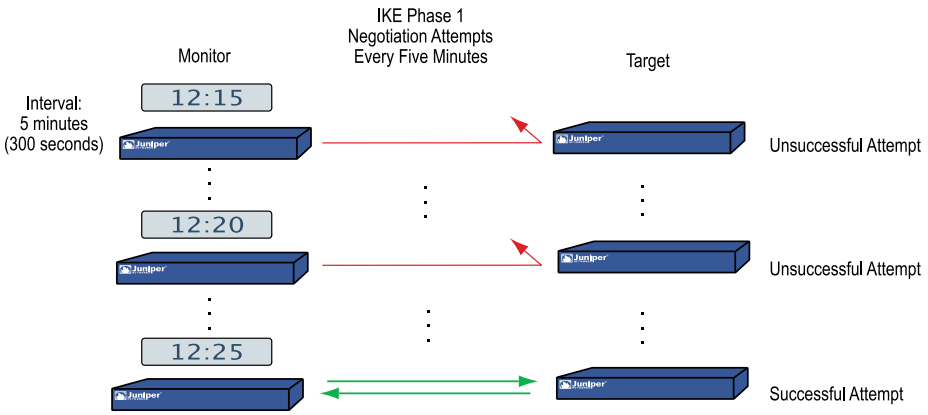
CLI

```
set ike gateway "our_gateway" address 1.1.1.1 main outgoing-interface "untrust"
preshare "jun9345" sec-level standard
set ike gateway "our_gateway" dpd interval 5
```

IKE Recovery Procedure

After the monitoring security device determines that a targeted device is down, the monitor stops sending IKE heartbeats and clears the SAs for that peer from its SA cache. After a defined interval, the monitor attempts to initiate Phase 1 negotiations with the failed peer. If the first attempt is unsuccessful, the monitor continues to attempt Phase 1 negotiations at regular intervals until negotiations are successful.

Figure 276: Repeated IKE Phase 1 Negotiation Attempts



To define the IKE recovery interval for a specified VPN tunnel (the minimum setting is 60 seconds), do either of the following:

WebUI

VPNs > AutoKey Advanced > Gateway > Edit (for the gateway whose IKE reconnect interval you want to modify) > Advanced: Enter the value in seconds in the Heartbeat Reconnect field, then click **OK**.

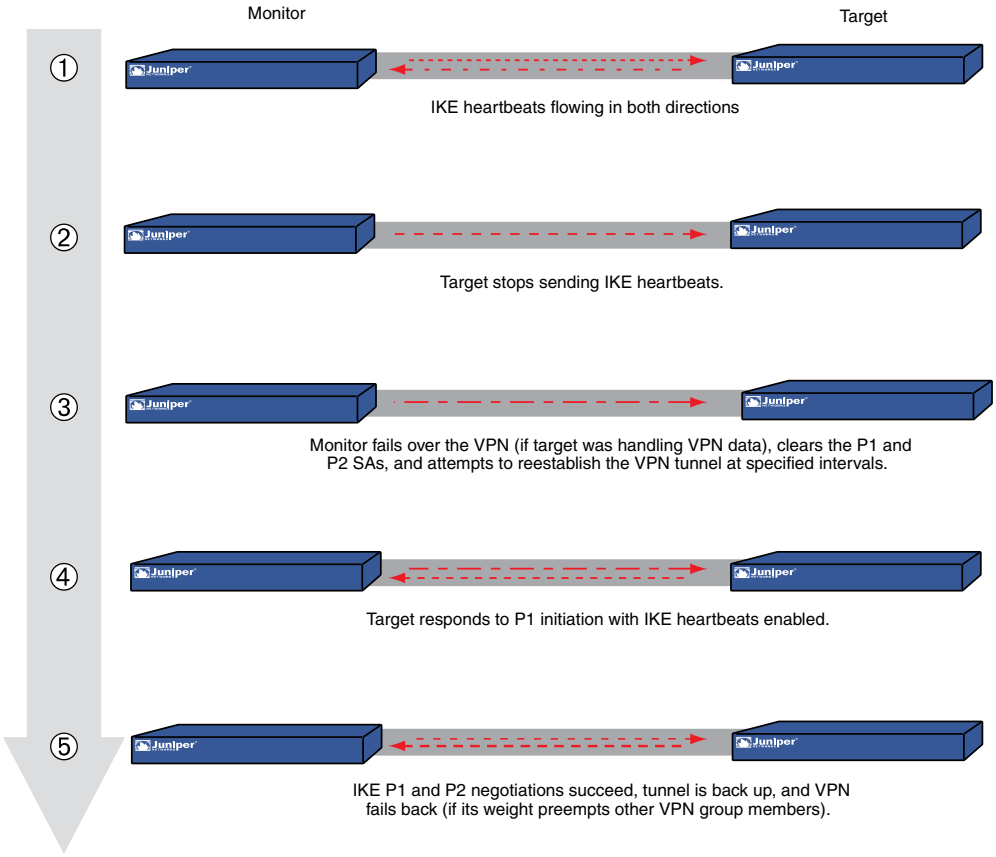
CLI

```
set ike gateway name_str heartbeat reconnect number
```

When a VPN group member with the highest weight fails over the tunnel to another group member and then reconnects with the monitoring device, the tunnel automatically fails back to the first member. The weighting system always causes the best ranking gateway in the group to handle the VPN data whenever it can do so.

Figure 277 on page 1031 presents the process that a member of a VPN group undergoes when the missing heartbeats from a targeted gateway surpass the failure threshold.

Figure 277: Failover and Then Recovery



TCP SYN-Flag Checking

For a seamless VPN failover to occur, the handling of TCP sessions must be addressed. If, after a failover, the new active gateway receives a packet in an existing TCP session, the new gateway treats it as the first packet in a new TCP session and checks if the SYN flag is set in the packet header. Because this packet is really part of an existing session, it does not have the SYN flag set. Consequently, the new gateway rejects the packet. With TCP SYN flag checking enabled, all TCP applications have to reconnect after the failover occurs.

To resolve this, you can disable SYN-flag checking for TCP sessions in VPN tunnels, as follows:

WebUI

You cannot disable SYN-flag checking through the WebUI.

CLI

```
unset flow tcp-syn-check-in-tunnel
```

NOTE: By default, SYN-flag checking is enabled.

Creating Redundant VPN Gateways

In this example, a corporate site has one VPN tunnel to a data center and a second tunnel to a backup data center. All the data is mirrored through a leased line connection between the two data center sites. The data centers are physically separate to provide continuous service even in the event of a catastrophic failure such as an all-day power outage or a natural disaster.

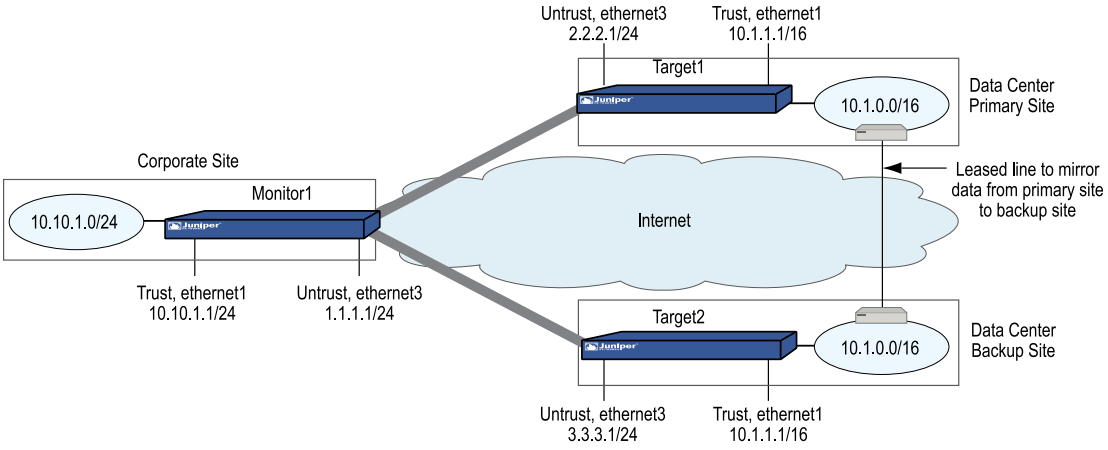
The device location and name, the physical interfaces and their IP addresses for the Trust and Untrust zones, and the VPN group ID and weight for each security device are as follows:

Device Location	Device Name	Physical Interface and IP Address (Trust Zone)	Physical Interface, IP Address, Default Gateway (Untrust Zone)	VPN Group ID and Weight
Corporate	Monitor1	ethernet1, 10.10.1.1/24	ethernet3, 1.1.1.1/24, (GW) 1.1.1.2	– –
Data Center (Primary)	Target1	ethernet1, 10.1.1.1/16	ethernet3, 2.2.2.1/24, (GW) 2.2.2.2	ID = 1, Weight = 2
Data Center (Backup)	Target2	ethernet1, 10.1.1.1/16	ethernet3, 3.3.3.1/24, (GW) 3.3.3.2	ID = 1, Weight = 1

NOTE: The internal address space at both data center sites must be identical.

All security zones are in the trust-vr routing domain. All the Site-to-Site AutoKey IKE tunnels use the security level predefined as “Compatible” for both Phase 1 and Phase 2 proposals. Preshared keys authenticate the participants.

Figure 278: Redundant VPN Gateways



WebUI (Monitor1)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.10.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: in_trust
   IP Address/Domain Name:
   IP/Netmask: (select), 10.10.1.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: data_ctr
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.0.0/16
   Zone: Untrust

3. **VPNs**

   VPNs > AutoKey Advanced > VPN Group: Enter 1 in the VPN Group ID field, then click **Add**.

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: target1
   Security Level: Compatible
   Remote Gateway Type: Static IP Address: (select), IP Address: 2.2.2.1
   Preshared Key: SLi1yoo129
   Outgoing Interface: ethernet3

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Compatible
Mode (Initiator): Main (ID Protection)
Heartbeat:
Hello: 3 Seconds
Reconnect: 60 seconds
Threshold: 5

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: to_target1
Security Level: Compatible
Remote Gateway: Predefined: (select), target1

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

VPN Group: VPN Group-1
Weight: 2

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: target2
Security Level: Compatible
Remote Gateway Type: Static IP Address: (select), IP Address: 3.3.3.1
Preshared Key: CMFwb7oN23
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Compatible
Mode (Initiator): Main (ID Protection)
Heartbeat:
Hello: 3 Seconds
Reconnect: 60 seconds
Threshold: 5

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: to_target2
Security Level: Compatible
Remote Gateway: Predefined: (select), target2

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

VPN Group: VPN Group-1
Weight: 1

4. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.2(untrust)

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), in_trust
Destination Address:
Address Book Entry: (select), data_ctr
Service: ANY
Action: Tunnel
VPN: VPN Group-1
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

WebUI (Target1)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/16
Enter the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 2.2.2.1/24

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: in_trust
IP Address/Domain Name:
IP/Netmask: (select), 10.1.0.0/16
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: corp
IP Address/Domain Name:

IP/Netmask: (select), 10.10.1.0/24
Zone: Untrust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: monitor1
Security Level: Compatible
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 1.1.1.1
Preshared Key: SLi1yoo129
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Compatible
Mode (Initiator): Main (ID Protection)
Heartbeat:
Hello: 3 Seconds
Reconnect: 0 seconds

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

Name: to_monitor1
Security Level: Compatible
Remote Gateway: Predefined: (select), monitor1

4. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 2.2.2.2

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), in_trust
Destination Address:
Address Book Entry: (select), corp
Service: ANY
Action: Tunnel
Tunnel VPN: monitor1
Modify matching bidirectional VPN policy: (select)
Position at Top: (select)

WebUI (Target2)

NOTE: Follow the Target1 configuration steps to configure Target2, but define the Untrust zone interface IP address as 3.3.3.1/24, the default gateway IP address as 3.3.3.2, and use CMFwb7oN23 to generate the preshared key.

CLI (Monitor1)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.10.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address trust in_trust 10.10.1.0/24
set address untrust data_ctr 10.1.0.0/16
```

3. **VPNs**

```
set ike gateway target1 address 2.2.2.1 main outgoing-interface ethernet3
preshare SLi1yoo129 sec-level compatible
set ike gateway target1 heartbeat hello 3
set ike gateway target1 heartbeat reconnect 60
set ike gateway target1 heartbeat threshold 5
set vpn to_target1 gateway target1 sec-level compatible
set ike gateway target2 address 3.3.3.1 main outgoing-interface ethernet3
preshare CMFwb7oN23 sec-level compatible
set ike gateway target2 heartbeat hello 3
set ike gateway target2 heartbeat reconnect 60
set ike gateway target2 heartbeat threshold 5
set vpn to_target2 gateway target2 sec-level compatible
set vpn-group id 1 vpn to_target1 weight 2
set vpn-group id 1 vpn to_target2 weight 1
unset flow tcp-syn-check-in-tunnel
```

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.2
```

5. **Policies**

```
set policy top from trust to untrust in_trust data_ctr any tunnel “vpn-group 1”
set policy top from untrust to trust data_ctr in_trust any tunnel “vpn-group 1”
save
```

CLI (Target1)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/16
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.1/24
```

2. **Addresses**

```
set address trust in_trust 10.1.0.0/16
set address untrust corp 10.10.1.0/24
```

3. **VPN**

```
set ike gateway monitor1 address 1.1.1.1 main outgoing-interface ethernet3
preshare SLi1yoo129 sec-level compatible
set ike gateway monitor1 heartbeat hello 3
set ike gateway monitor1 heartbeat threshold 5
set vpn to_monitor1 gateway monitor1 sec-level compatible
```

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.2
```

5. **Policies**

```
set policy top from trust to untrust in_trust corp any tunnel vpn to_monitor
set policy top from untrust to trust corp in_trust any tunnel vpn to_monitor
save
```

CLI (Target2)

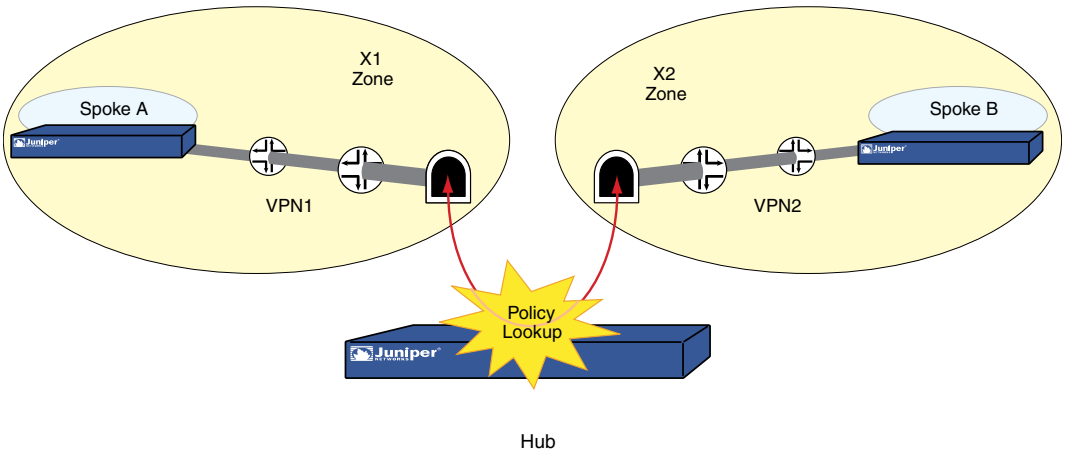
NOTE: Follow the Target1 configuration steps to configure Target2, but define the Untrust zone interface IP address as 3.3.3.1/24, the default gateway IP address as 3.3.3.2, and use CMFwb7oN23 to generate the preshared key.

Creating Back-to-Back VPNs

You can enforce interzone policies at the hub site for traffic passing from one VPN tunnel to another by putting the spoke sites in different zones. Because they are in different zones, the security device at the hub must do a policy lookup before routing the traffic from one tunnel to another. You can control the traffic flowing through the VPN tunnels between the spoke sites. This arrangement is called *back-to-back VPNs*.

NOTE: Optionally, you can enable intrazone blocking and define an intrazone policy to control traffic between the two tunnel interfaces within the same zone.

Figure 279: Back-to-Back VPNs



Following are a few benefits of back-to-back VPNs:

- You can conserve the number of VPNs you need to create. For example, perimeter site A can link to the hub and to perimeter sites B, C, D…, but A only has to set up one VPN tunnel. Especially for NetScreen-5XP users, who can use a maximum of ten VPN tunnels concurrently, applying the hub-and-spoke method dramatically increases their VPN options and capabilities.
- The administrator (admin) at the hub device can completely control VPN traffic between perimeter sites. For example,
  - The admin might permit only HTTP traffic to flow from sites A to B, but allow any kind of traffic to flow from B to A.
  - The admin can allow traffic originating from A to reach C, but deny traffic originating from C to reach A.
  - The admin can allow a specific host at A to reach the entire D network, while allowing only a specific host at D to reach a different host at A.
- The administrator at the hub device can completely control outbound traffic from all perimeter networks. At each perimeter site, there must first be a policy that tunnels all outbound traffic through the spoke VPNs to the hub; for example: **set policy top from trust to untrust any any any tunnel vpn** name_str (where name_str defines the specific VPN tunnel from each perimeter site to the hub). At the hub, the administrator can control Internet access, allowing certain kinds of traffic (such as HTTP only), performing URL blocking on undesirable websites, and so on.
- Regional hubs can be used and interconnected through spoke tunnels, allowing spoke sites in one region to reach spoke sites in another.

The following example is similar to “Creating Hub-and-Spoke VPNs” on page 1047 except that the security device at the hub site in New York performs policy checking on the traffic it routes between the two tunnels to the branch offices in Tokyo and Paris. By putting each remote site in a different zone, you control the VPN traffic at the hub.

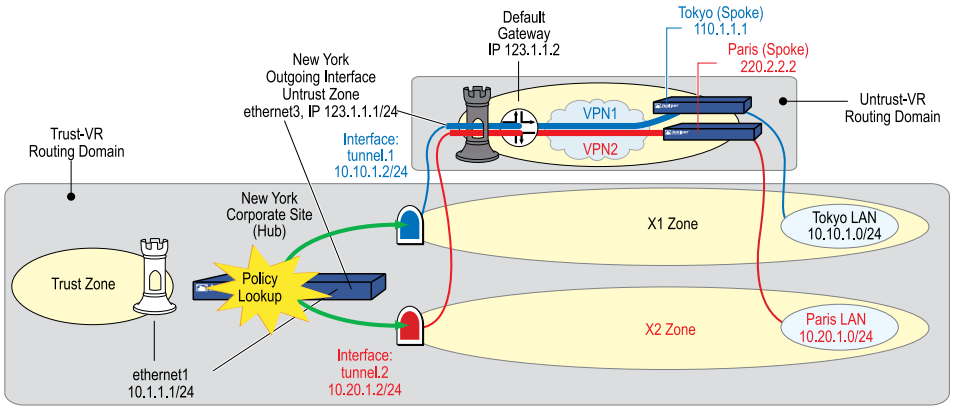
The Tokyo LAN address is in the user-defined X1 zone, and the Paris LAN address is in the user-defined X2 zone. Both zones are in the Trust-VR routing domain.

NOTE: To create user-defined zones, you might first need to obtain and load a zone software key on the security device.

You bind the VPN1 tunnel to the tunnel.1 interface and the VPN2 tunnel to the tunnel.2 interface. Although you do not assign IP addresses to the X1 and X2 zone interfaces, you do give addresses to both tunnel interfaces. Routes for these interfaces automatically appear in the Trust-VR routing table. By putting the IP address for a tunnel interface in the same subnet as that of the destination, traffic destined for that subnet is routed to the tunnel interface.

The outgoing interface is ethernet3, which is bound to the Untrust zone. As you can see in Figure 280 on page 1041, both tunnels terminate in the Untrust zone; however, the endpoints for the traffic that makes use of the tunnels are in the X1 and X2 zones. The tunnels use AutoKey IKE, with preshared keys. You select the security level predefined as “Compatible” for both Phase 1 and Phase 2 proposals. You bind the Untrust zone to the untrust-vr. Because the tunnels are route-based (that is, the correct tunnel is determined by routing, not by a tunnel name specified in a policy), proxy IDs are included in the configuration of each tunnel.

Figure 280: Back-to-Back VPNs with Two Routing Domains and Multiple Security Zones


Default
Gateway
IP 123.1.1.2
Tokyo (Spoke)
110.1.1.1
Paris (Spoke)
220.2.2.2
New York
Outgoing Interface
Untrust Zone
ethernet3, IP 123.1.1.1/24
VPN1
VPN2
Untrust-VR
Routing Domain
Trust-VR
Routing Domain
Interface:
tunnel.1
10.10.1.2/24
New York
Corporate Site
(Hub)
X1 Zone
Tokyo LAN
10.10.1.0/24
Trust Zone
Policy
Lookup
X2 Zone
Paris LAN
10.20.1.0/24
ethernet1
10.1.1.1/24
Interface:
tunnel.2
10.20.1.2/24

WebUI

1. **Security Zones and Virtual Routers**

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   IP Address/Netmask: 0.0.0.0/0
   Manage IP: 0.0.0.0

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Null

   Network > Zones > Edit (for Untrust): Enter the following, then click **OK**:

   Virtual Router Name: untrust-vr
   Block Intra-Zone Traffic: (select)

   Network > Zones > New: Enter the following, then click **OK**:

   Zone Name: X1
   Virtual Router Name: trust-vr
   Block Intra-Zone Traffic: (select)

   Network > Zones > New: Enter the following, then click **OK**:

   Name: X2
   Virtual Router Name: trust-vr
   Block Intra-Zone Traffic: (select)

2. **Interfaces**

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 123.1.1.1/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): X1 (trust-vr)
   Fixed IP: (select)
      IP Address / Netmask: 10.10.1.2/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.2
   Zone (VR): X2 (trust-vr)
   Fixed IP: (select)
      IP Address / Netmask: 10.20.1.2/24

3. **VPN for Tokyo Office**

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: VPN1
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: Tokyo
Type: Static IP: (select), Address/Hostname: 110.1.1.1
Preshared Key: netscreen1
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Proxy-ID: (select)
Local IP / Netmask: 10.20.1.0/24
Remote IP / Netmask: 10.10.1.0/24
Service: ANY

NOTE: When configuring the VPN tunnel on the security device protecting the Tokyo and Paris offices, do either of the following:

(Route-based VPN) Select the Enable Proxy-ID check box, and enter **10.10.1.0/24** (Tokyo) and **10.20.1.0/24** (Paris) for the Local IP and Netmask and **10.20.1.0/24** (Tokyo) and **10.10.1.0/24** (Paris) for the Remote IP and Netmask.

(Policy-based VPN) Make an entry in the Trust zone address book for 10.10.1.0/24 (Tokyo) and 10.20.1.0/24 (Paris) and another in the Untrust zone address book for 10.20.1.0/24 (Tokyo) and 10.10.1.0/24 (Paris). Use those as the source and destination addresses in the policy referencing the VPN tunnel to the hub site.

4. **VPN for Paris Office**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: VPN2
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: Paris
Type: Static IP: (select), Address/Hostname: 220.2.2.2
Preshared Key: netscreen2
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Proxy-ID: (select)
Local IP / Netmask: 10.10.1.0/24
Remote IP / Netmask: 10.20.1.0/24
Service: ANY

5. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Next Hop Virtual Router Name: (select), untrust-vr

Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 123.1.1.2

6. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Tokyo LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.10.1.0/24
Zone: X1

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Paris LAN
IP Address/Domain Name:
IP/Netmask: (select), 10.20.1.0/24
Zone: X2

7. **Policies**

Policy > (From: X1, To: X2) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Tokyo LAN
Destination Address:
Address Book Entry: (select), Paris LAN
Service: ANY
Action: Permit
Position at Top: (select)

Policy > (From: X2, To: X1) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Paris LAN
Destination Address:
Address Book Entry: (select), Tokyo LAN
Service: ANY
Action: Permit
Position at Top: (select)

CLI

1. **Security Zones and Virtual Routers**

```
unset interface ethernet3 ip
unset interface ethernet3 zone
set zone untrust vrouter untrust-vr
set zone untrust block
set zone name X1
set zone x1 vrouter trust-vr
set zone x1 block
set zone name x2
set zone x2 vrouter trust-vr
set zone x2 block
```

2. **Interfaces**

```
set interface ethernet3 zone untrust
set interface ethernet3 ip 123.1.1.1/24
set interface tunnel.1 zone x1
set interface tunnel.1 ip 10.10.1.2/24
set interface tunnel.2 zone x2
set interface tunnel.2 ip 10.20.1.2/24
```

3. **VPN for Tokyo Office**

```
set ike gateway Tokyo address 110.1.1.1 outgoing-interface ethernet3 preshare
 netscreen1 sec-level compatible
set vpn VPN1 gateway Tokyo sec-level compatible
set vpn VPN1 bind interface tunnel.1
set vpn VPN1 proxy-id local-ip 10.20.1.0/24 remote-ip 10.10.1.0/24 any
```

NOTE: When configuring the VPN tunnel on the security device protecting the Tokyo and Paris offices, do either of the following:

(Route-based VPN) Enter the following commands: **set vpn VPN1 proxy-id local-ip 10.20.1.0/24 remote-ip 10.10.1.0/24** (Tokyo) and **set vpn VPN1 proxy-id local-ip 10.10.1.0/24 remote-ip 10.20.1.0/24** (Paris).

(Policy-based VPN) Make an entry in the Trust zone address book for 10.10.1.0/24 (Tokyo) and 10.20.1.0/24 (Paris) and another in the Untrust zone address book for 10.20.1.0/24 (Tokyo) and 10.10.1.0/24 (Paris). Use those as the source and destination addresses in the policies referencing the VPN tunnel to the hub site.

4. **VPN for Paris Office**

```
set ike gateway Paris address 220.2.2.2 outgoing-interface ethernet3 preshare
 netscreen2 sec-level compatible
set vpn VPN2 gateway Paris sec-level compatible
set vpn VPN2 bind interface tunnel.2
set vpn VPN2 proxy-id local-ip 10.10.1.0/24 remote-ip 10.20.1.0/24 any
```

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 vrouter untrust-vr
set vrouter untrust-vr route 0.0.0.0/0 interface ethernet3 gateway 123.1.1.2
```

6. **Addresses**

```
set address x1 “Tokyo LAN” 10.10.1.0/24
set address x2 “Paris LAN” 10.20.1.0/24
```

7. **Policies**

```
set policy top from x1 to x2 “Tokyo LAN” “Paris LAN” any permit
set policy top from x2 to x1 “Paris LAN” “Tokyo LAN” any permit
save
```

NOTE: You can ignore the following message, which appears because tunnel interfaces are in NAT mode:

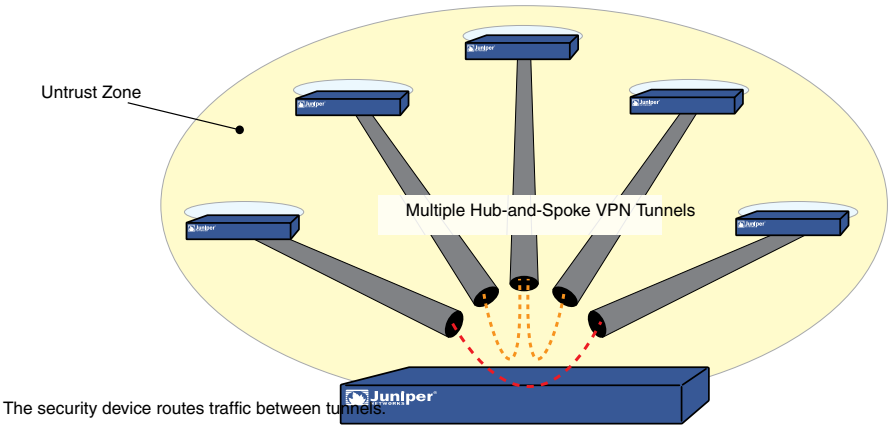
Warning: Some interfaces in the *zone_name* zone are in NAT mode. Traffic might not pass through them!

Creating Hub-and-Spoke VPNs

If you create two VPN tunnels that terminate at a security device, you can set up a pair of routes so that the security device directs traffic exiting one tunnel to the other tunnel. If both tunnels are contained within a single zone, you do not need to create a policy to permit the traffic to pass from one tunnel to the other. You only need to define the routes. Such an arrangement is known as a *hub-and-spoke VPN*.

You can also configure multiple VPNs in one zone and route traffic between any two tunnels.

Figure 281: Multiple Tunnels in a Hub-and-Spoke VPN Configuration



In this example, two branch offices in Tokyo and Paris communicate with each other through a pair of VPN tunnels—VPN1 and VPN2. Each tunnel originates at the remote site and terminates at the corporate site in New York. The security device at the corporate site routes traffic exiting one tunnel into the other tunnel.

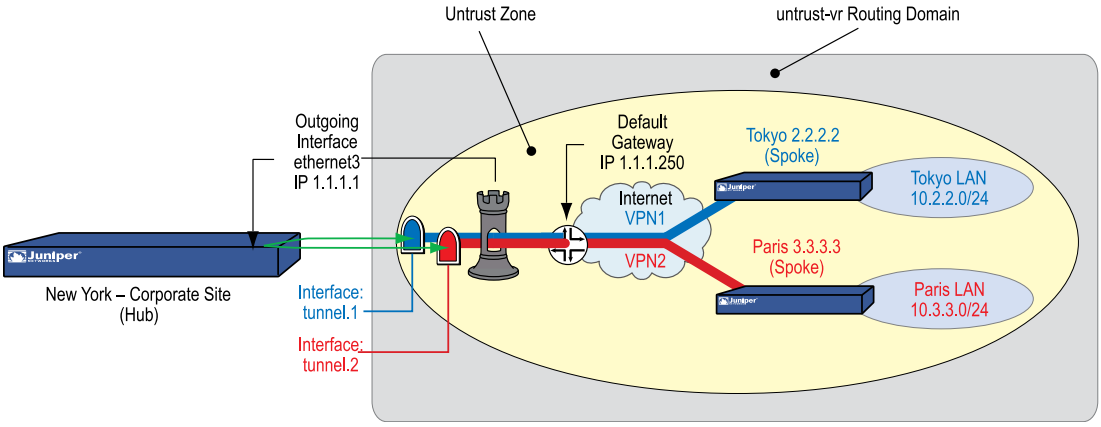
By disabling intrazone blocking, the security device at the corporate site only needs to do a route lookup—not a policy lookup—when conducting traffic from tunnel to tunnel because both remote endpoints are in the same zone (the Untrust Zone).

NOTE: Optionally, you can leave intrazone blocking enabled and define an intrazone policy permitting traffic between the two tunnel interfaces.

You bind the tunnels to the tunnel interfaces—tunnel.1 and tunnel.2—which are both unnumbered. The tunnels use AutoKey IKE, with the preshared keys. You select the security level predefined as "Compatible" for both Phase 1 and Phase 2 proposals. You bind the Untrust zone to the untrust-vr. The Untrust zone interface is ethernet3.

NOTE: The following configuration is for route-based VPNs. If you configure policy-based hub-and-spoke VPNs, you must use the Trust and Untrust zones in the policies; you cannot use user-defined security zones.

Figure 282: Hub-and-Spoke VPNs



WebUI (New York)

1. **Security Zones and Virtual Routers**

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   IP Address/Netmask: 0.0.0.0/0
   Manage IP: 0.0.0.0

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Null

   Network > Zones > Edit (for Untrust): Enter the following, then click **OK**:

   Virtual Router Name: untrust-vr
   Block Intra-Zone Traffic: (clear)

2. **Interfaces**

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (untrust-vr)
   Unnumbered: (select)
      Interface: ethernet3 (untrust-vr)

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.2
   Zone (VR): Untrust (untrust-vr)
   Unnumbered: (select)
      Interface: ethernet3 (untrust-vr)

3. **VPN for Tokyo Office**

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: VPN1
   Security Level: Compatible
   Remote Gateway: Create a Simple Gateway: (select)
      Gateway Name: Tokyo
      Type: Static IP: (select), Address/Hostname: 2.2.2.2
      Preshared Key: netscreen1
      Security Level: Compatible
      Outgoing Interface: ethernet3

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

   Proxy-ID: (select)
      Local IP / Netmask: 0.0.0.0/0
      Remote IP / Netmask: 0.0.0.0/0
      Service: ANY

4. **VPN for Paris Office**

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: VPN2
   Security Level: Compatible
   Remote Gateway: Create a Simple Gateway: (select)
      Gateway Name: Paris
      Type: Static IP: (select), Address/Hostname: 3.3.3.3
      Preshared Key: netscreen2
      Security Level: Compatible
      Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

5. **Routes**

Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.2.2.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK:**

Network Address / Netmask: 10.3.3.0/24
Gateway: (select)
Interface: tunnel.2
Gateway IP Address: 0.0.0.0

Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

WebUI (Tokyo)

1. **Security Zones and Virtual Routers**

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

IP Address/Netmask: 0.0.0.0/0
Manage IP: 0.0.0.0

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Null

Network > Zones > Edit (for Untrust): Enter the following, then click **OK**:

Virtual Router Name: untrust-vr
Block Intra-Zone Traffic: (select)

2. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.2.2.1/24
Select the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 2.2.2.2/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (untrust-vr)
Unnumbered: (select)
  Interface: ethernet3 (untrust-vr)

3. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Paris
IP Address/Domain Name:
  IP/Netmask: (select), 10.3.3.0/24
Zone: Untrust

4. **VPN**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: VPN1
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
  Gateway Name: New York
  Type: Static IP: (select), Address/Hostname: 1.1.1.1
  Preshared Key: netscreen1
  Security Level: Compatible
  Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Proxy-ID: (select)
  Local IP / Netmask: 0.0.0.0/0
  Remote IP / Netmask: 0.0.0.0/0
  Service: ANY

5. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Next Hop Virtual Router Name: (select); untrust-vr

Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 2.2.2.250

Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.3.3.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

6. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Paris
Service: ANY
Action: Permit

WebUI (Paris)

1. **Security Zones and Virtual Routers**

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

IP Address/Netmask: 0.0.0.0/0
Manage IP: 0.0.0.0

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Null

Network > Zones > Edit (for Untrust): Enter the following, then click **OK**:

Virtual Router Name: untrust-vr
Block Intra-Zone Traffic: (select)

2. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.3.3.1/24
Select the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 3.3.3.3/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (untrust-vr)
Unnumbered: (select)
  Interface: ethernet3 (untrust-vr)

3. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Tokyo
IP Address/Domain Name:
  IP/Netmask: (select), 10.2.2.0/24
Zone: Untrust

4. **VPN**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: VPN2
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
  Gateway Name: New York
  Type: Static IP: (select), Address/Hostname: 1.1.1.1
  Preshared Key: netscreen2
  Security Level: Compatible
  Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Proxy-ID: (select)
  Local IP / Netmask: 0.0.0.0/0
  Remote IP / Netmask: 0.0.0.0/0
  Service: ANY

5. **Routes**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Next Hop Virtual Router Name: (select); untrust-vr

Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 3.3.3.250

Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.2.2.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

6. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Tokyo
Service: ANY
Action: Permit

CLI (New York)

1. **Security Zones and Virtual Routers**

```
unset interface ethernet3 ip
unset interface ethernet3 zone
set zone untrust vrouter untrust-vr
unset zone untrust block
```

2. **Interfaces**

```
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.2 zone untrust
set interface tunnel.2 ip unnumbered interface ethernet3
```

3. **VPN for Tokyo Office**

```
set ike gateway Tokyo address 2.2.2.2 outgoing-interface ethernet3 preshare
netscreen1 sec-level compatible
```

```
set vpn VPN1 gateway Tokyo sec-level compatible
set vpn VPN1 bind interface tunnel.1
set vpn VPN1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

4. **VPN for Paris Office**

```
set ike gateway Paris address 3.3.3.3 outgoing-interface ethernet3 preshare
netscreen2 sec-level compatible
set vpn VPN2 gateway Paris sec-level compatible
set vpn VPN2 bind interface tunnel.2
set vpn VPN2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

5. **Routes**

```
set vrouter untrust-vr route 10.2.2.0/24 interface tunnel.1
set vrouter untrust-vr route 10.3.3.0/24 interface tunnel.2
set vrouter untrust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
save
```

CLI (Tokyo)

1. **Security Zones and Virtual Routers**

```
unset interface ethernet3 ip
unset interface ethernet3 zone
set zone untrust vrouter untrust-vr
```

2. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

3. **Address**

```
set address untrust Paris 10.3.3.0/24
```

4. **VPN**

```
set ike gateway "New York" address 1.1.1.1 outgoing-interface ethernet3
preshare netscreen1 sec-level compatible
set vpn VPN1 gateway "New York" sec-level compatible
set vpn VPN1 bind interface tunnel.1
set vpn VPN1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 vrouter untrust-vr
set vrouter untrust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
set vrouter untrust-vr route 10.3.3.0/24 interface tunnel.1
```

6. **Policies**

```
set policy from trust to untrust any Paris any permit
set policy from untrust to trust Paris any any permit
save
```

CLI (Paris)

1. **Security Zones and Virtual Routers**

```
unset interface ethernet3 ip
unset interface ethernet3 zone
set zone untrust vrouter untrust-vr
```

2. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.3.3.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 3.3.3.3/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

3. **Address**

```
set address untrust Tokyo 10.2.2.0/24
```

4. **VPN**

```
set ike gateway “New York” address 1.1.1.1 outgoing-interface ethernet3
preshare netscreen2 sec-level compatible
set vpn VPN2 gateway “New York” sec-level compatible
set vpn VPN2 bind interface tunnel.1
set vpn VPN2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 an
```

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 vrouter untrust-vr
set vrouter untrust-vr route 0.0.0.0/0 interface ethernet3 gateway 3.3.3.250
set vrouter untrust-vr route 10.2.2.0/24 interface tunnel.1
```

6. **Policies**

```
set policy from trust to untrust any Tokyo any permit
set policy from untrust to trust Tokyo any any permit
save
```

IKE and IPsec Passthrough Traffic

Beginning with ScreenOS 6.3.0, Network Address Translation (NAT) supports both NAT-Traversal and Non-NAT-Traversal Internet Key Exchange (IKE) and IPsec passthrough traffic.

NAT-T IKE and IPsec Passthrough Traffic

Beginning with the ScreenOS 6.3.0 release, the security device performs sticky Dynamic Internet Protocol (DIP) for IKE packets with Network Address Translation-Traversal (NAT-T) enabled.

To ensure that the security device assigns the same source IP address as that of the IKE packet, you can enable the "sticky" DIP address feature by entering the CLI command **set dip sticky**.

For example, to create a new service for NAT-T to support IKE packets with destination port 4500, you must create policies to allow the IKE packets to pass through the security device.

```
set dip sticky
set interface ethernet1 dip-id 4 1.1.1.1  1.1.1.10
set service IKE-NAT-T protocol udp src-port 4500-4500 dst-port 4500-4500
set policy id 1  from trust to untrust  any any IKE-NAT nat src dip-id 4 permit log
set policy id 2  from trust  to untrust  any  any IKE-NAT-T  nat src dip-id 4 permit log
```

Non-NAT-T IKE and IPsec Passthrough Traffic

For IKE packets without NAT-Traversal, you create a pinhole when IKE starts phase 2 negotiations for the inbound traffic that passes through the device. The security device blocks all the subsequent IKE phase 2 packets until the pinhole is removed.

When the first IPsec packets arrive at the pinhole, a session is established. The subsequent IPsec packets that match the session pass through the security device.

To specify the IPsec and IKE packets without NAT-T in the policy, use the predefined service IKE-NAT.

Example:

```
set policy id 1 from trust  to untrust any any IKE-NAT nat src dip-id 4 permit log
```

For information on Predefined Services, see "Services" on page 134.

Chapter 26

AutoConnect-Virtual Private Networks

This chapter describes the AutoConnect-virtual private network (AC-VPN) feature in ScreenOS, explains how it works in a hub-and-spoke network topology, and provides a configuration example of a typical scenario in which it might be used.

Overview

Small enterprise organizations that secure their remote satellite sites with virtual private network (VPN) tunnels typically interconnect all sites in a full-mesh VPN, because remote sites need to communicate with each other as well as with headquarters. In this type of network, remote sites usually run low-end security devices that support a maximum of 25 VPN tunnels. When the total number of sites exceeds 25, however, the enterprise must either place security devices with greater capacity at its remote sites (at considerable cost) or switch from full-mesh to a hub-and-spoke network topology.

A hub-and-spoke configuration solves the problem of scalability, but its principle drawback is that all communication between spokes must go through the hub. This generally is not an issue when traffic is simple data, even with more than one thousand spokes. However, if traffic is video or Voice over Internet Protocol (VoIP), the processing overhead on the hub can cause latency, a critical problem for such applications.

AC-VPN provides a way for you to configure your hub-and-spoke network so that spokes can dynamically create VPN tunnels directly between each other as needed. This not only solves the problem of latency between spokes but also reduces processing overhead on the hub and thus improves overall network performance. Additionally, because AC-VPN creates dynamic tunnels that time out when traffic ceases to flow through them, network administrators are freed from the time-consuming task of maintaining a complex network of static VPN tunnels.

How It Works

AC-VPN is designed to be implemented in a hub-and-spoke network in which all spokes are connected to the hub by VPN tunnels. After you set up a static VPN tunnel between the hub and each of the spokes, you configure AC-VPN on the hub and the spokes and then enable the Next Hop Resolution Protocol (NHRP). The hub uses NHRP to obtain a range of information about each spoke, including its public-to-private

address mappings, subnetmask length, and routing and hop count information, which the hub caches. Then, when any spoke begins communicating with another spoke (through the hub), the hub uses this information, in combination with information obtained from the AC-VPN configuration on the spokes, to enable the spokes to set up an AC-VPN tunnel between themselves. While the tunnel is being negotiated, communication continues to flow between the two spokes through the hub. When the dynamic tunnel becomes active, the hub drops out of the link and traffic flows directly between the two spokes. When traffic ceases to flow through the dynamic tunnel, the tunnel times out.

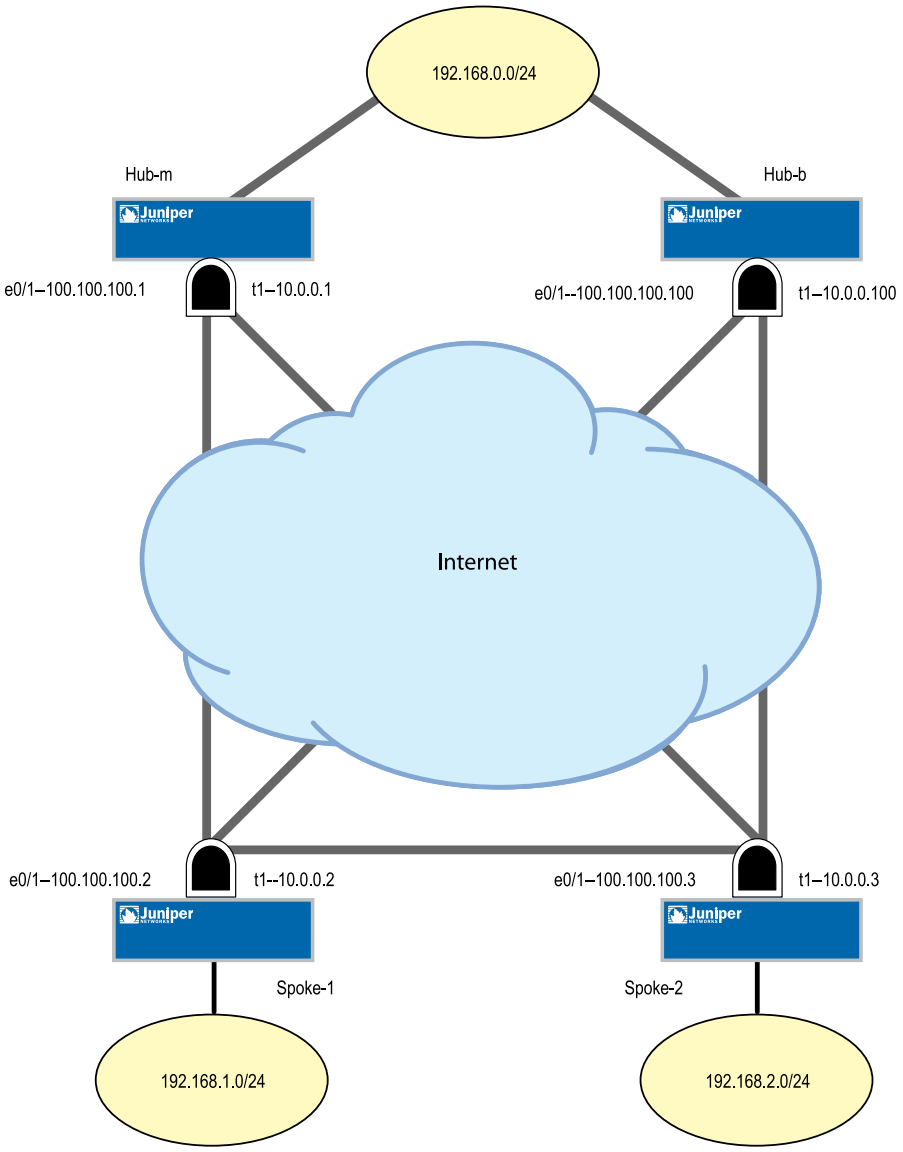
Dual-Hub AC-VPN

In cases when the hub fails and the dynamic tunnel expires, the spokes cannot reestablish the connection. To avoid this, in the current release ScreenOS allows you to configure two hubs on the same virtual router (VR) so that connectivity is not lost even if one hub fails. In Figure 283 on page 1061, hub-m and hub-b act as Next Hop Servers (NHSs) and are configured on the same VR. Spoke1 and spoke2 are Next Hop Clients (NHCs) and are connected to each hub by a static VPN tunnel. Based on the priority you set to a routing instance on the hubs, either of the hubs may be active. If the priority at hub-m is higher, then all traffic from spoke1 to spoke2 will pass through hub-m until a dynamic VPN is established. We also recommend that you enable VPN monitoring for the tunnels at the spoke end so that the spokes monitor the status of the hubs. For more information, see “VPN Monitoring” on page 971. When hub-m fails, the dynamic tunnel and its associated NHRP routing instance will be removed at both the spokes. Traffic now flows through hub-b, which creates a new dynamic tunnel between the spokes. If hub-m comes back, the spokes choose routing through hub-m as the best route because of the priority setting. However, the traffic continues to flow through the new dynamic tunnel until hub-b fails.

NOTE: If you run an OSPF routing instance on the hubs and want hub-m to be the primary, you need to set the OSPF cost value at hub-m to be less than hub-b.

As AC-VPN supports dynamic routing protocols, traffic from other subnets behind the spoke that needs to be routed through a hub may pass through the dynamic tunnel already created by the first cached subnet. To avoid this, in the current release ScreenOS allows you to disable the dynamic routing operation on the AC-VPN tunnel. Additionally in this release, you can redistribute routes learned from NHRP into dynamic routing protocols such as BGP, OSPF, and RIP. In the same way, routes learned by the dynamic routing protocols can be redistributed automatically into the NHRP routing instance.

Figure 283: Dual-Hub AC-VPN



NHRP Messages

In the context of NHRP, the hub in a hub-and-spoke network is called the Next Hop Server (NHS), and the spoke is called the Next Hop Client (NHC). NHRP communication between NHS and NHC takes place though a formal exchange of NHRP messages. The nonbroadcast multi access (NBMA) Next Hop Resolution Protocol (RFC 2332) defines seven NHRP messages. To these seven messages, ScreenOS adds two more. These nine messages and their operation in an AC-VPN hub-and-spoke network are defined as follows:

- **Registration Request—**After a static VPN tunnel becomes active between an NHC and its NHS, the NHC sends an NHRP Registration Request message to the NHS. The message contains a number of Client Information Entries (CIEs), which include such things as the NHC's public-to-private address mappings, subnetmask length, and routing and hop-count information.

NOTE: In the current ScreenOS implementation, NHRP does not redistribute any routes to its peers, and BGP and OSPF do not redistribute NHRP routes to their peers.

- **Registration Reply—**The NHS can ACK or NAK a registration request. If the NHS does not recognize the packet, the packet is not acknowledged (NAK) and is dropped. Upon successful registration, the NHS caches the CIEs contained in the registration request and sends a Registration Reply ACK.
- **Resolution Request, Resolution Reply—**With the introduction of the ScreenOS proprietary Resolution-set and Resolution-ack messages, the function of the NHRP Resolution Request and Resolution Reply message pair is relegated to keeping cached CIEs on the NHS current. The NHCs do this in conjunction with the holding time configured on the NHS, which specifies the lifetime of cached entries for each NHC. To ensure that the NHS has current information about their subnetworks, NHCs periodically send Resolution Request messages to the NHS. If any devices have been added to or removed from their subnetworks, that information is contained in the Resolution Request message, and the NHS updates its cache and sends a Resolution Reply.
- **Purge Request, Purge Reply—**When an administrator shuts down an NHC, the device sends a Purge Request message to the NHS. Upon receipt of the Purge Request, the NHS removes all cached entries for that NHC and sends a Purge Reply. If the NHC experiences system failure and goes off line, the NHS removes cached entries for the device after the configured lifetime for the cache expires.
- **Error Indication—**This message logs NHRP error conditions.

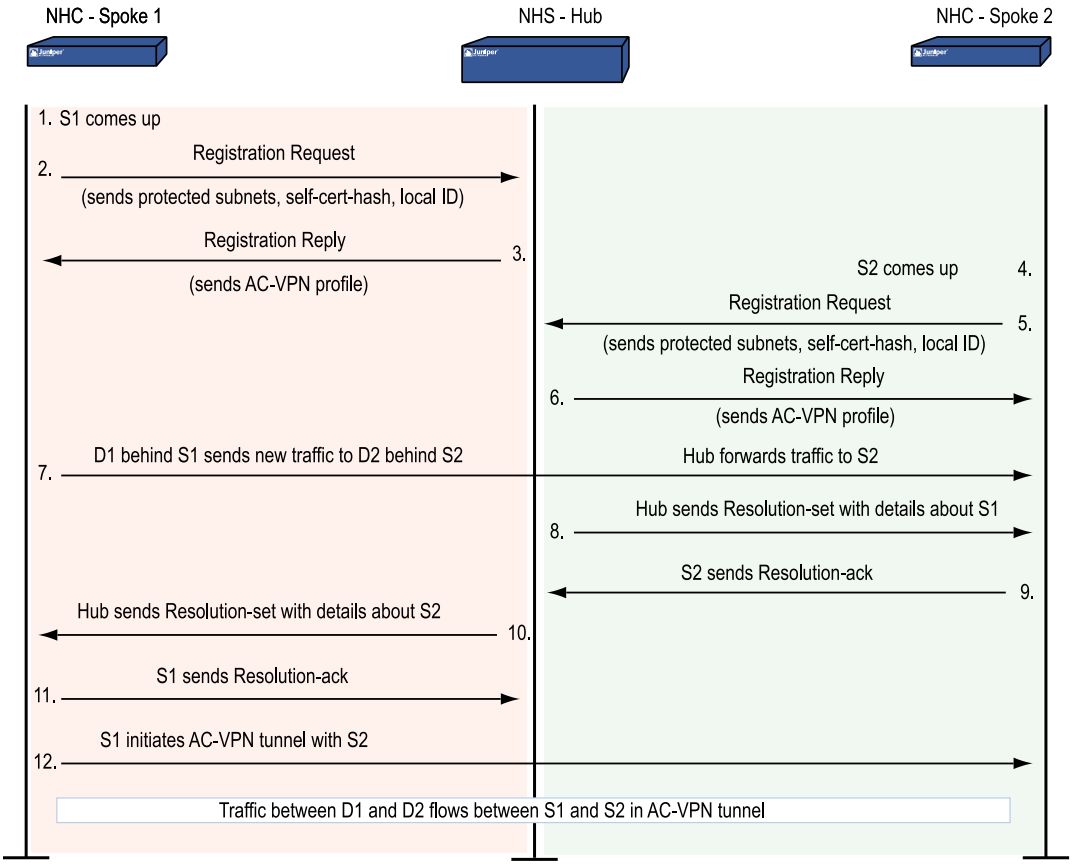
To support AC-VPN, ScreenOS adds the following message pair:

- **Resolution-set, Resolution-ack—**When the NHS detects traffic from one static VPN tunnel to another, it sends Resolution-set messages to the NHCs at the end of each static tunnel. These messages contain all the information each NHC needs about the other to set up an AC-VPN tunnel. When the NHCs reply with Resolution-ack messages, the NHS directs one of the NHCs to initiate AC-VPN tunnel negotiation.

AC-VPN Tunnel Initiation

Figure 284 on page 1063 illustrates how ScreenOS triggers the setup of an AC-VPN tunnel using the NHRP Registration Request and Registration Reply messages, and the custom ScreenOS Resolution-set and Resolution-ack messages. For simplification, the figure does not show the exchange of Resolution Request and Resolution Reply messages, nor the Purge and Error messages. (The abbreviations S1 and S2 refer to Spoke 1 and Spoke 2, respectively; D1 and D2 refer to destinations behind those spokes.)

Figure 284: AC-VPN Set Up Via NHRP



Configuring AC-VPN

The following general restrictions apply:

- All VPN tunnels configured toward the hub must be route based.
- Automatic key management in phase 1 must be in aggressive mode.
- The authentication method must be self-signed certificate and generic PKI.
- All spokes must be connected to a single zone on the hub.
- Configuring NHRP in multiple instances of virtual routers is supported only on the NHS.

Network Address Translation

The following restrictions apply with NAT:

- Nat-Traversal—AC-VPN can create a dynamic tunnel between two spokes if one of the spokes is behind a NAT device in the path toward the hub; if both spokes are behind NAT devices, however, a dynamic tunnel will not be created and communication between the spokes will proceed through the hub. See "NAT-Traversal" on page 961 for a discussion of NAT-Traversal.
- NAT is supported only with Mapped Internet Protocol (MIP) addressing.
- Port Address Translation (PAT) is supported only between one spoke and the hub. For example, you can have a NAT device between one spoke and the hub and a dynamic tunnel can be created between that spoke and another spoke as long as there is no NAT device between that other spoke and the hub. In this scenario, the hub will force the spoke behind the NAT device to initiate the tunnel to the other spoke.
- PAT directly in front of the hub is supported.
- PAT between spokes is not supported.

Configuration on the Hub

The spoke configuration includes the following:

1. Create a static gateway and VPN.
2. Create static tunnels to the spokes and bind the VPNs to the tunnels.
3. Create an AC-VPN gateway profile.
4. Create an AC-VPN VPN profile.
5. Enable NHRP on the virtual router.
6. Select the ACVPN-Profile for NHRP.
7. Enable NHRP on the tunnel interface.
8. Configure routing.

Configuration on Each Spoke

The hub includes the following:

1. Create a static tunnel to the hub.
2. Create a gateway.
3. Create a VPN gateway.
4. Create an ACVPN-Dynamic gateway.
5. Create ACVPN-Dynamic VPN
6. Enable NHRP on the virtual router
7. Configure the NHS IP address

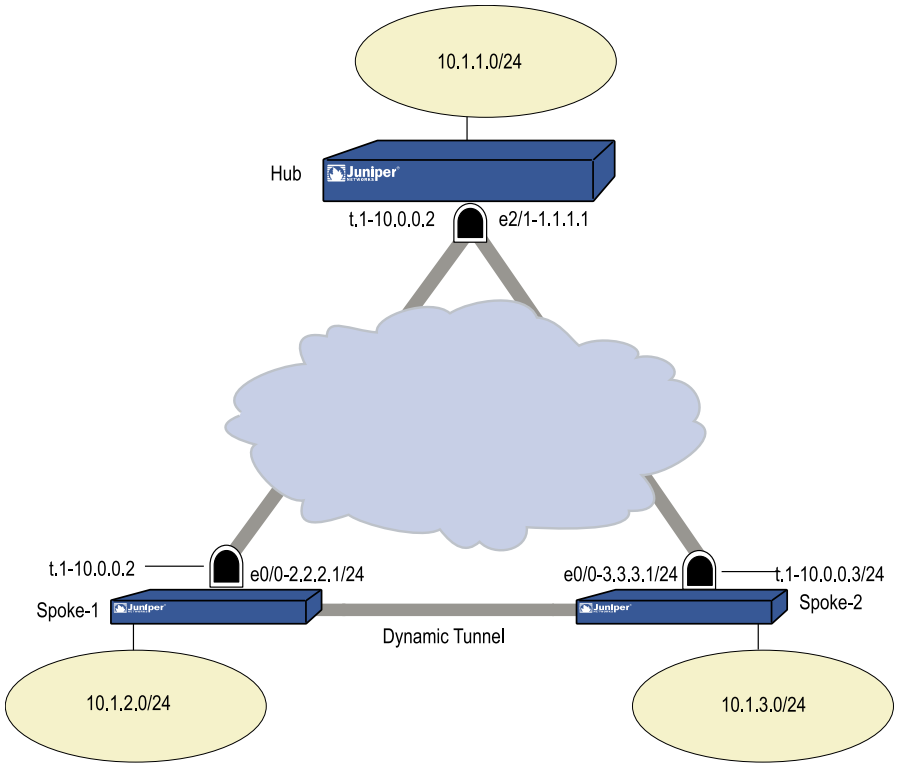
8. Enable NHRP on the tunnel interface.
9. Configure routing.

Example

NOTE: AC-VPN also supports Dynamic Host Control Protocol (DHCP).

In this example, a high-end security device acting as the hub in a hub and spoke network is configured to act as the Next Hop Server (NHS) in an AC-VPN configuration in which Spoke1 and Spoke2 (low-end security devices) are Next Hop Clients (NHCs). After configuring interfaces on the devices, you configure static VPN tunnels between the hub and each of the spokes, then configure AC-VPN and enable NHRP on the connecting interfaces. Although this example uses the Open Shortest Path First (OSPF) routing protocol, ScreenOS supports all dynamic routing protocols with AC-VPN.

Figure 285: Next Hop Server (NHS) in a AC-VPN Configuration



WebUI (Hub)

NOTE: After you configure static gateways and static VPNs from the hub to the spokes and from the spokes to the hub, you can use the AC-VPN wizard to complete the AC-VPN configuration.

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2/1): Enter the following, then click **Apply**:

   Zone Name: Untrust
   IP Address/Netmask: 1.1.1.1/24
   Interface Mode: Route

   Network > Interfaces > Edit (for ethernet2/2): Enter the following, then click **Apply**:

   Zone Name: Trust
   IP Address/Netmask: 10.1.1.1/24
   Interface Mode: NAT

   Network > Interfaces > New (Tunnel IF): Enter the following, then click **Apply**:

   Tunnel Interface Name: 1
   Zone (VR): Trust-vr
   Fixed IP
   IP Address/Netmask: 10.0.0.1/24
   Unnumbered
   Interface:
   NHRP Enable: (Select)

2. **Configure Tunnels to Spoke1 and Spoke2**

   VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

   Gateway Name: Spoke1
   Remote Gateway: (Select)
   Static IP Address: (Select) IPv4/v6 Address/Hostname: 2.2.2.1

   Preshare key: Juniper
   Outgoing interface: (select) ethernet2/1
   Security Level: Standard

3. **Configure VPN Spoke to Gateway**

   VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

Gateway Name: Spoke2
Remote Gateway: (Select)
Static IP Address: (Select) IPv4/v6 Address/Hostname: 3.3.3.1

Preshare key: Juniper
Outgoing interface: (select) ethernet2/1
Security Level: Standard

4. **Configure VPN Spoke to Gateway**

VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

VPN Name: spoke1
Remote Gateway: (select) Predefined
(select appropriate gateway name from predefined list in drop-down list)
Security Level (select) Standard
Bind To: (select) Tunnel Interface
(select Tunnel.1 from the drop-down list)

VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

VPN Name: spoke2
Remote Gateway: (select) Predefined
(select appropriate gateway name from predefined list from drop-down list)

Security Level (select) Standard)
Bind To: (select) Tunnel Interface
(select Tunnel.1 from the drop-down list)

5. **Configure the ACVPN-Profile**

VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

Gateway Name: ac-spoke-gw
ACVPN-Profile: (select)

Security Level: (select) Standard

VPNs > AutoKey IKE > New: Configure the ACVPN profile, click Advanced and set the security level and Replay Protection, then click **Return** to go back to the VPN configuration page and click **OK**

VPN Name: ac-vpn
ACVPN-Profile: (select)
Binding to tunnel: (select) ac-spoke-gw

Security Level: (select) Standard
Replay Protection: (select)

6. **Configure Vrouter**

Network > Routing > Virtual Router > (for trust-vr) Edit: Enter the following, then click **Apply:**

Next Hop Resolution Protocol(NHRP) Support
NHRP: (select) NHRP Setting
NHS Setting: (select)
Profile: (select) ACVPN-Profile name

7. **Enable NHRP on the Tunnel Interface**

Network > Interfaces > New (for TunnelIF): Enter the following, then click **Apply:**

Tunnel Interface Name: 1
NHRP Enable: (select)

8. **Configure Routing**

Network > Routing > Destination > New: Enter the following, then click **Apply:**

IP Address/netmask: 0.0.0.0
Gateway (select)
Gateway IP Address: 1.1.1.2

CLI (Hub)

```
set interface ethernet2/1 zone Untrust
set interface ethernet2/2 zone Trust

set interface tunnel.1 zone Trust
set interface ethernet2/1 ip 1.1.1.1/24
set interface ethernet2/1 route
set interface ethernet2/2 ip 10.1.1.1/24
set interface ethernet2/2 nat
set interface tunnel.1 ip 10.0.0.1/24
set ike gateway spoke2 address 3.3.3.1 Main outgoing-interface ethernet2/1 preshare
 juniper== sec-level standard

set ike gateway spoke1 address 2.2.2.1 Main outgoing-interface ethernet2/1 preshare
 juniper== sec-level standard
set vpn spoke2 gateway spoke2 no-replay tunnel idletime 0 sec-level standard

set vpn spoke2 id 1 bind interface tunnel.1
set vpn spoke1 gateway spoke1 no-replay tunnel idletime 0 sec-level standard
set vpn spoke1 id 2 bind interface tunnel.1

set ike gateway ac-spoke-gw acvpn-profile sec-level standard
set vpn ac-vpn acvpn-profile ac-spoke-gw no-replay tunnel idletime 0 sec-level standard

set vrouter trust-vr
set protocol nhrp
set protocol nhrp acvpn-profile ac-vpn
exit
set interface tunnel.1 protocol nhrp enable
```

```
set vr trust protocol ospf
set vr trust protocol ospf enable
set vr trust protocol ospf area 10
set interface ethernet2/2 protocol ospf area 0.0.0.10
set interface ethernet2/2 protocol ospf enable
set interface tunnel.1 protocol ospf area 0.0.0.0
set interface tunnel.1 protocol ospf link-type p2mp
set interface tunnel.1 protocol ospf enable

set route 0.0.0.0/0 gateway 1.1.1.2
```

WebUI (Spoke1)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet0/0): Enter the following, then click **Apply**:

   Zone Name: Untrust
   IP Address/Netmask: 2.2.2.1/24
   Interface Mode: Route

   Network > Interfaces > Edit (for bgroup0): Enter the following, then click **Apply**:

   Zone Name: Trust

   Network > Interfaces > Edit (for ethernet2/2): Enter the following, then click **Apply**:

   Zone Name: Trust
   IP Address/Netmask: 10.1.2.1/24
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet2/2), Select Bind Port, enter the following, then click **Apply:**

   ethernet0/2 bgroup0: (select) Bind to Current Bgroup

   Network > Interfaces > New (Tunnel IF): Enter the following, then click **Apply**:

   Tunnel Interface Name: 1
   Zone (VR): Trust-vr
   Fixed IP
   IP Address/Netmask: 10.0.0.2/24
   NHRP Enable: (Select)

2. **Configure Tunnel to the Hub**

   VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

   Gateway Name: hub-gw
   Remote Gateway: (Select)

Static IP Address: (Select) IPv4/v6 Address/Hostname: 1.1.1.1

Preshare key: Juniper
Outgoing interface: (select) ethernet2/1
Security Level: Standard

3. **Configure VPN Spoke to Gateway**

   VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

   VPN Name: vpn-hub
   Remote Gateway: (select) Predefined
   (select appropriate gateway name from predefined list from drop-down list)

   Security Level (select) Standard
   Bind To: (select) Tunnel Interface
   (select Tunnel.1 from the drop-down list)

4. **Configure ACVPN-Dynamic**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: ac-hub-gw
   ACVPN-Dynamic: (select)

   VPNs > AutoKey > New: Enter the following, then click **OK**:

   VPN Name: ac-hub-vpn
   ACVPN-Dynamic: (select)
   Gateway (select): ac-hub-gw
   Tunnel Towards Hub: (select) vpn-hub

5. **Configure the Virtual Router**

   Network > Routing > Virtual Router > (for trust-vr) Edit: Enter the following, then click **Apply:**

   Next Hop Resolution Protocol(NHRP) Support
   NHRP Enable: (select)
   NHC Setting: (select)
   NHS IP Address: 10.1.1.1

6. **Enable NHRP on the Tunnel Interface**

   Network > Interfaces > New (for TunnelIF): Enter the following, then click **Apply:**

   Tunnel Interface Name: 1
   NHRP Enable: (select)

7. **Configure Routing**

   Network > Routing > Destination > New: Enter the following, then click **Apply:**

IP Address/netmask: 0.0.0.0
Gateway (select)
Gateway IP Address: 2.2.2.2

CLI (Spoke1)

```
set interface ethernet0/0 zone Untrust
set interface bgroup0 zone Trust
set interface bgroup0 port ethernet0/2
set interface tunnel.1 zone Trust

set interface ethernet0/0 ip 2.2.2.1/24
set interface ethernet0/0 route
set interface bgroup0 ip 10.1.2.1/24
set interface bgroup0 nat
set interface tunnel.1 ip 10.0.0.2/24

set ike gateway hub-gw address 1.1.1.1 Main outgoing-interface ethernet0/0 preshare
 juniper== sec-level standard
set vpn vpn-hub id 1 bind interface tunnel.1

set ike gateway ac-hub-gw acvpn-dynamic
set vpn ac-hub-vpn acvpn-dynamic ac-hub-gw vpn-hub
set vrouter trust-vr
set protocol nhrp
set protocol nhrp nhs 10.0.0.1
exit
set interface tunnel.1 protocol nhrp enable

set vr trust protocol ospf
set vr trust protocol ospf enable
set vr trust protocol ospf area 20

set interface bgroup0 protocol ospf area 0.0.0.20
set interface bgroup0 protocol ospf enable
set interface tunnel.1 protocol ospf area 0.0.0.0
set interface tunnel.1 protocol ospf enable
set route 0.0.0.0/0 gateway 2.2.2.2
```

WebUI (Spoke2)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet0/0): Enter the following, then click **Apply**:

   Zone Name: Untrust
   IP Address/Netmask: 3.3.3.1/24
   Interface Mode: Route

   Network > Interfaces > Edit (for bgroup0): Enter the following, then click **Apply**:

   Zone Name: Trust

Network > Interfaces > Edit (for ethernet2/2): Enter the following, then click **Apply**:

Zone Name: Trust
IP Address/Netmask: 10.1.3.1/24
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet2/2), Select Bind Port, enter the following, then click **Apply:**

ethernet0/2 bgroup0: (select) Bind to Current Bgroup

Network > Interfaces > New (Tunnel IF): Enter the following, then click **Apply**:

Tunnel Interface Name: 1
Zone (VR): Trust-vr
Fixed IP
IP Address/Netmask: 10.0.0.3/24
NHRP Enable: (Select)

2. **Configure Tunnel to the Hub**

VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

Gateway Name: hub-gw
Remote Gateway: (Select)
Static IP Address: (Select) IPv4/v6 Address/Hostname: 1.1.1.1

Preshare key: Juniper
Outgoing interface: (select) ethernet2/1
Security Level: Standard

3. **Configure VPN Spoke to Gateway**

VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

VPN Name: vpn-hub
Remote Gateway: (select) Predefined
(select appropriate gateway name from predefined list from drop-down list)

Security Level (select) Standard
Bind To: (select) Tunnel Interface
(select Tunnel.1 from the drop-down list)

4. **Configure ACVPN-Dynamic**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: ac-hub-gw
ACVPN-Dynamic: (select)

VPNs > AutoKey > New: Enter the following, then click **OK**:

VPN Name: ac-hub-vpn
ACVPN-Dynamic: (select)
Gateway (select): ac-hub-gw
Tunnel Towards Hub: (select) vpn-hub

5. **Configure Vrouter**

Network > Routing > Virtual Router > (for trust-vr) Edit: Enter the following, then click **Apply:**

Next Hop Resolution Protocol(NHRP) Support
NHRP Enable: (select)
NHC Setting: (select)
NHS IP Address: 1.1.1.1

6. **Enable NHRP on the Tunnel Interface**

Network > Interfaces > New (for TunnelIF): Enter the following, then click **Apply:**

Tunnel Interface Name: 1
NHRP Enable: (select)

7. **Configure Routing**

Network > Routing > Destination > New: Enter the following, then click **Apply:**

IP Address/netmask: 0.0.0.0
Gateway (select)
Gateway IP Address: 3.3.3.3

CLI (Spoke2)

```
set interface ethernet0/0 zone Untrust
set interface bgroup0 zone Trust
set interface bgroup0 port ethernet0/2
set interface tunnel.1 zone Trust

set interface ethernet0/0 ip 3.3.3.1/24
set interface ethernet0/0 route
set interface bgroup0 ip 10.1.3.1/24
set interface bgroup0 nat
set interface tunnel.1 ip 10.0.0.3/24

set ike gateway hub-gw address 1.1.1.1 Main outgoing-interface ethernet0/0 preshare
 juniper== sec-level standard
set vpn vpn-hub id 1 bind interface tunnel.1

set ike gateway ac-hub-gw acvpn-dynamic
set vpn ac-hub-vpn acvpn-dynamic ac-hub-gw vpn-hub

set vrouter trust-vr
set protocol nhrp
```

```
set protocol nhrp nhs 10.0.0.1
exit

set interface tunnel.1 protocol nhrp enable

set vr trust protocol ospf
set vr trust protocol ospf enable
set vr trust protocol ospf area 30
set interface bgroup0 protocol ospf area 0.0.0.30
set interface bgroup0 protocol ospf enable
set interface tunnel.1 protocol ospf area 0.0.0.0
set interface tunnel.1 protocol ospf enable
set route 0.0.0.0/0 gateway 3.3.3.3
```

Configuring Dual-Hub AC-VPN

In this example, two hubs are configured on the same virtual router to act as Next Hop Servers (NHSs). Each spoke is connected to the two hubs by a static VPN tunnel. In this scenario, connectivity is not lost even if one hub fails. After configuring the interfaces, you configure static VPN tunnels between the hubs and each of the spokes, enable VPN monitoring at the spoke sides, disable dynamic routing operation on the AC-VPN tunnel at the spoke sides, and configure AC-VPN and enable NHRP on the connecting interfaces. Although this example uses the Open Shortest Path First (OSPF) routing protocol, ScreenOS supports all dynamic routing protocols with AC-VPN. This example is based on the topology shown in Figure 283 on page 1061.

WebUI (Hub-m)

1. **Configure Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

   Zone Name: Untrust
   IP Address/Netmask: 100.100.100.1/24

   Network > Interfaces > New (Tunnel IF): Enter the following, then click **Apply**:

   Tunnel Interface Name: 1
   Zone (VR): Trust-vr
   Fixed IP
   IP Address/Netmask: 10.0.0.1/24
   NHRP Enable: (Select)

2. **Enable Dymanic Routing (OSPF)**

   Network > Routing > Virtual Routers > Edit (for trust-vr) > Create OSPF Instance: Select **OSPF Enabled**, then click **Apply**:

   Area > Configure (for area 0.0.0.0): Click **< < Add** to move the tunnel.1 interface from the Available Interface(s) list to the Selected Interface(s) list, then click **OK**.

   Network > Interfaces > Edit (for tunnel.1) > OSPF: Enter the following, then click **Apply**:

Bind to Area: (select), Select **0.0.0.0** from the drop-down list.
Protocol OSPF: Enable
Link Type: Point-to-Multipoint (select)
Cost: 5

3. **Configure Tunnels to Spoke1 and Spoke2**

VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

Gateway Name: Spoke1
Remote Gateway: (select)
Static IP Address: (select) IPv4/v6 Address/Hostname: 100.100.100.2
Preshare key: Juniper
Outgoing interface: (select) ethernet0/1
Security Level: Standard

VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

Gateway Name: Spoke2
Remote Gateway: (select)
Static IP Address: (Select) IPv4/v6 Address/Hostname: 100.100.100.3
Preshare key: Juniper
Outgoing interface: (select) ethernet0/1
Security Level: Standard

4. **Configure VPN Spoke to Gateway**

VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

VPN Name: spoke1
Remote Gateway: (select) spoke1
Security Level (select) Standard
Bind To: (select) Tunnel.1

VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

VPN Name: spoke2
Remote Gateway: (select) spoke2
Security Level (select) Standard
Bind To: (select) Tunnel.1

5. **Configure ACVPN-Profile**

VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

Gateway Name: ac-gw
ACVPN-Profile: (select)
Security Level: (select) Standard

VPNs > AutoKey IKE > New: Configure the ACVPN profile, click **Advanced** and set the security level, then click **Return** to go back to the VPN configuration page and click **OK**:

VPN Name: ac-vpn
ACVPN-Profile: (select)
Binding to tunnel: (select) ac-gw
Security Level: (select) Standard

6. **Configuration on Vrouter**

Network > Routing > Virtual Router > (for trust-vr) Edit: Enter the following, then click **Apply:**

Next Hop Resolution Protocol (NHRP) Support
NHRP: (select) NHRP Setting
Enable: (select)
NHS Setting: (select)
Profile: (select) ac-vpn

CLI (Hub-m)

```
set vrouter trust-vr protocol ospf

set interface tunnel.1 zone trust
set interface tunnel.1 ip 10.0.0.1/24

set ike gateway spoke1 address 100.100.100.2 main outgoing-interface ethernet0/1
 preshare Juniper sec-level standard
set ike gateway spoke2 address 100.100.100.3 main outgoing-interface ethernet0/1
 preshare Juniper sec-level standard

set ike gateway ac-gw acvpn-profile sec-level standard

set vpn spoke1 gateway spoke1 sec-level standard
set vpn spoke1 bind interface tunnel.1
set vpn spoke2 gateway spoke2 sec-level standard
set vpn spoke2 bind interface tunnel.1

set vpn ac-vpn acvpn-profile ac-gw sec-level standard

set vrouter trust-vr protocol nhrp
set vrouter trust-vr protocol nhrp acvpn-profile ac-vpn

set interface tunnel.1 protocol nhrp enable
set interface tunnel.1 protocol ospf area 0
set interface tunnel.1 protocol ospf link-type p2mp
```

```
set interface tunnel.1 protocol ospf enable
set interface tunnel.1 protocol ospf cost 5
```

WebUI (Hub-b)

1. **Configure Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

   Zone Name: Untrust
   IP Address/Netmask: 100.100.100.100/24

   Network > Interfaces > New (Tunnel IF): Enter the following, then click **Apply**:

   Tunnel Interface Name: 1
   Zone (VR): Trust-vr
   Fixed IP
   IP Address/Netmask: 10.0.0.100/24
   NHRP Enable: (Select)

2. **Enable Dymanic Routing (OSPF)**

   Network > Routing > Virtual Routers > Edit (for trust-vr) > Create OSPF Instance: Select **OSPF Enabled**, then click **Apply**:

   Area > Configure (for area 0.0.0.0): Click **< < Add** to move the tunnel.1 interface from the Available Interface(s) list to the Selected Interface(s) list, then click **OK**.

   Network > Interfaces > Edit (for tunnel.1) > OSPF: Enter the following, then click **Apply**:

   Bind to Area: (select), Select **0.0.0.0** from the drop-down list.
   Protocol OSPF: Enable
   Link Type: Point-to-Multipoint (select)

3. **Configure Tunnels to Spoke1 and Spoke2**

   VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

   Gateway Name: Spoke1
   Remote Gateway: (select)
   Static IP Address: (Select) IPv4/v6 Address/Hostname: 100.100.100.2
   Preshare key: Juniper
   Outgoing interface: (select) ethernet0/1
   Security Level: Standard

   VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

   Gateway Name: Spoke2
   Remote Gateway: (select)

Static IP Address: (Select) IPv4/v6 Address/Hostname: 100.100.100.3
Preshare key: Juniper
Outgoing interface: (select) ethernet0/1
Security Level: Standard

4. **Configure VPN Spoke to Gateway**

VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

VPN Name: spoke1
Remote Gateway: (select) spoke1
Security Level (select) Standard
Bind To: (select) Tunnel.1

VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

VPN Name: spoke2
Remote Gateway: (select) spoke2
Security Level (select) Standard
Bind To: (select) Tunnel.1

5. **Configure ACVPN-Profile**

VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

Gateway Name: ac-gw
ACVPN-Profile: (select)
Security Level: (select) Standard

VPNs > AutoKey IKE > New: Configure the ACVPN profile, click **Advanced** and set the security level, then click **Return** to go back to the VPN configuration page and click **OK**:

VPN Name: ac-vpn
ACVPN-Profile: (select)
Binding to tunnel: (select) ac-gw
Security Level: (select) Standard

6. **Configuration on Vrouter**

Network > Routing > Virtual Router > (for trust-vr) Edit: Enter the following, then click **Apply:**

Next Hop Resolution Protocol (NHRP) Support
NHRP: (select) NHRP Setting
Enable: (select)
NHS Setting: (select)
Profile: (select) ac-vpn

CLI (Hub-b)

```
set vrouter trust-vr protocol ospf

set interface tunnel.1 zone trust
set interface tunnel.1 ip 10.0.0.100/24

set ike gateway spoke1 address 100.100.100.2 main outgoing-interface ethernet0/1
 preshare Juniper sec-level standard
set ike gateway spoke2 address 100.100.100.3 main outgoing-interface ethernet0/1
 preshare Juniper sec-level standard

set ike gateway ac-gw acvpn-profile sec-level standard

set vpn spoke1 gateway spoke1 sec-level standard
set vpn spoke1 bind interface tunnel.1
set vpn spoke2 gateway spoke2 sec-level standard
set vpn spoke2 bind interface tunnel.1

set vpn ac-vpn acvpn-profile ac-gw sec-level standard

set vrouter trust-vr protocol nhrp
set vrouter trust-vr protocol nhrp acvpn-profile ac-vpn

set interface tunnel.1 protocol nhrp enable
set interface tunnel.1 protocol ospf area 0
set interface tunnel.1 protocol ospf link-type p2mp
set interface tunnel.1 protocol ospf enable
```

WebUI (Spoke1)

NOTE: This example assumes that you run RIP instance on the network behind spoke1.

1. **Configure Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

   Zone Name: Untrust
   IP Address/Netmask: 100.100.100.2/24

   Network > Interfaces > New (Tunnel IF): Enter the following, then click **Apply**:

   Tunnel Interface Name: 1
   Zone (VR): Trust-vr
   Fixed IP
   IP Address/Netmask: 10.0.0.2/24
   NHRP Enable: (Select)
   ACVPN Dynamic Routing: (confirm this not selected)

2. **Enable Dymanic Routing (OSPF)**

Network > Routing > Virtual Routers > Edit (for trust-vr) > Create OSPF Instance: Select **OSPF Enabled**, then click **Apply**:

Area > Configure (for area 0.0.0.0): Click **< < Add** to move the tunnel.1 interface from the Available Interface(s) list to the Selected Interface(s) list, then click **OK**.

Network > Interfaces > Edit (for tunnel.1) > OSPF: Enter the following, then click **Apply**:

Bind to Area: (select), Select **0.0.0.0** from the drop-down list.
Protocol OSPF: Enable
Link Type: Point-to-Multipoint (select)

Network > Interfaces > Edit (for ethernet0/1) > OSPF: Enter the following, then click **Apply**:

Protocol OSPF: Enable (select)

3. **Configure Tunnels to hub-m and hub-b**

VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

Gateway Name: hub-m
Remote Gateway: (select)
Static IP Address: (Select) IPv4/v6 Address/Hostname: 100.100.100.1
Preshare key: Juniper
Outgoing interface: (select) ethernet0/1
Security Level: Standard

VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

Gateway Name: hub-b
Remote Gateway: (select)
Static IP Address: (Select) IPv4/v6 Address/Hostname: 100.100.100.100
Preshare key: Juniper
Outgoing interface: (select) ethernet0/1
Security Level: Standard

4. **Configure VPN Spoke to Gateway**

VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

VPN Name: hub-m
Remote Gateway: (select) hub-m
Security Level (select) Standard
Bind To: (select) Tunnel.1
VPN Monitor: (select)
Rekey: (select)

Network > Interfaces > Edit (for tunnel.1) > NHTB: Enter the following, then click **Add**:

New Next Hop Entry:
IP Address: 10.0.0.1
VPN: hub-m

VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

VPN Name: hub-b
Remote Gateway: (select) hub-b
Security Level (select) Standard
Bind To: (select) Tunnel.1
VPN Monitor: (select)
Rekey: (select)

Network > Interfaces > Edit (for tunnel.1) > NHTB: Enter the following, then click **Add**:

New Next Hop Entry:
IP Address: 10.0.0.100
VPN: hub-b

5. **Configure ACVPN-Dynamic**

VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

Gateway Name: ac-gw
ACVPN-Dynamic: (select)

VPNs > AutoKey IKE > New: Configure the ACVPN profile, click **Advanced** and set the security level, then click **Return** to go back to the VPN configuration page and click **OK**:

VPN Name: ac-vpn
ACVPN-Dynamic: (select)
Gateway: (select) ac-gw

6. **Configuration on Vrouter**

Network > Routing > Virtual Router > (for trust-vr) Edit: Do the following:

> NHS Setting > Multiple NHS Setting: Enter the IP address of the NHS server and click **Add**, then click **Back to NHRP Setting** to go back to the NHRP Setting page:

IP: 10.0.0.1
IP: 10.0.0.100

> NHS Setting > Cache Setting: Enter the following and click **Add**, then click **Back to NHRP Setting** to go back to the NHRP Setting page:

NHS: 192.168.1.0/24

> NHS Setting: Enter the following, then click **Apply**:

NHC Setting: (select)
ACVPN Name: (select) ac-vpn
Enable: (select)

7. **Create Route Map**

Network > Routing > Virtual Router > Route Map > New (for trust-vr): Enter the following, then click **OK**:

Map Name: rtmap1
Sequence No.: 1
Action: permit (select)
Match Properties:
Interface: (select), tunnel.1

8. **Redistribute RIP Routes**

Network > Routing > Virtual Router > Edit (for trust-vr) > Edit RIP Instance > Redistributable Rules: Select the following, then click **Add**:

Route Map: rtmap1
Protocol: OSPF

Network > Routing > Virtual Router > Edit (for trust-vr) > Edit RIP Instance > Redistributable Rules: Select the following, then click **Add**:

Route Map: rtmap1
Protocol: NHRP

CLI (Spoke1)

```
set vrouter trust-vr protocol ospf

set interface ethernet0/1 zone untrust
set interface ethernet0/1 ip 100.100.100.2/24
set interface tunnel.1 zone trust
set interface tunnel.1 ip 10.0.0.2/24

set ike gateway hub-m address 100.100.100.1 outgoing-interface ethernet0/1
preshare Juniper sec-level standard
set ike gateway hub-b address 100.100.100.100 outgoing-interface ethernet0/1
preshare Juniper sec-level standard

set ike gateway ac-gw acvpn-dynamic

set vpn hub-m gateway hub-m sec-level standard
set vpn hub-m monitor rekey
set vpn hub-m bind interface tunnel.1
Set interface tunnel.1 nhtb 10.0.0.1 vpn hub-m
set vpn hub-b gateway hub-b sec-level standard
```

```
set vpn hub-b monitor rekey
set vpn hub-b bind interface tunnel.1
set interface tunnel.1 nhtb 10.0.0.100 vpn hub-b

set vpn ac-vpn acvpn-dynamic ac-gw

set vrouter trust-vr protocol nhrp
set vrouter trust-vr protocol nhrp acvpn-dynamic ac-vpn
set vrouter trust-vr protocol nhrp nhs 10.0.0.1
set vrouter trust-vr protocol nhrp nhs 10.0.0.100
set vrouter trust-vr protocol nhrp cache 192.168.1.0/24

set interface tunnel.1 protocol nhrp enable
unset interface tunnel.1 acvpn-dynamic-routing

set vrouter trust-vr
set route-map name rtmap1 permit 1
set match interface tunnel.1
exit

set protocol rip
set redistribute route-map rtmap1 protocol ospf
set redistribute route-map rtmap1 protocol nhrp
exit
exit

set interface tunnel.1 protocol ospf area 0
set interface tunnel.1 protocol ospf link-type p2mp
set interface tunnel.1 protocol ospf enable
```

WebUI (Spoke2)

NOTE: This example assumes that NHRP cache entries are not set manually.

1. **Configure Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **Apply**:

   Zone Name: Untrust
   IP Address/Netmask: 100.100.100.3/24

   Network > Interfaces > Edit (for ethernet0/0): Enter the following, then click **Apply**:

   OSPF Enable: (select)

   Network > Interfaces > New (Tunnel IF): Enter the following, then click **Apply**:

   Tunnel Interface Name: 1
   Zone (VR): Trust-vr
   Fixed IP
   IP Address/Netmask: 10.0.0.3/24

NHRP Enable: (Select)
ACVPN Dynamic Routing: (confirm this not selected)

2. **Enable Dymanic Routing (OSPF)**

   Network > Routing > Virtual Routers > Edit (for trust-vr) > Create OSPF Instance: Select **OSPF Enabled**, then click **Apply**:

   Area > Configure (for area 0.0.0.0): Click **< < Add** to move the tunnel.1 interface from the Available Interface(s) list to the Selected Interface(s) list, then click **OK**.

   Network > Interfaces > Edit (for tunnel.1) > OSPF: Enter the following, then click **Apply**:

   Bind to Area: (select), Select **0.0.0.0** from the drop-down list.
   Protocol OSPF: Enable
   Link Type: Point-to-Multipoint (select)

   Network > Interfaces > Edit (for ethernet0/1) > OSPF: Enter the following, then click **Apply**:

   Protocol OSPF: Enable (select)

3. **Configure Tunnels to hub-m and hub-b**

   VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

   Gateway Name: hub-m
   Remote Gateway: (select)
   Static IP Address: (Select) IPv4/v6 Address/Hostname: 100.100.100.1
   Preshare key: Juniper
   Outgoing interface: (select) ethernet0/1
   Security Level: Standard

   VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

   Gateway Name: hub-b
   Remote Gateway: (select)
   Static IP Address: (Select) IPv4/v6 Address/Hostname: 100.100.100.100
   Preshare key: Juniper
   Outgoing interface: (select) ethernet0/1
   Security Level: Standard

4. **Configure VPN Spoke to Gateway**

   VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

   VPN Name: hub-m
   Remote Gateway: (select) hub-m
   Security Level (select) Standard

Bind To: (select) Tunnel.1
VPN Monitor: (select)
Rekey: (select)

Network > Interfaces > Edit (for tunnel.1) > NHTB: Enter the following, then click **Add**:

New Next Hop Entry:
IP Address: 10.0.0.1
VPN: hub-m

VPNs > AutoKey IKE > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

VPN Name: hub-b
Remote Gateway: (select) hub-b
Security Level (select) Standard
Bind To: (select) Tunnel.1
VPN Monitor: (select)
Rekey: (select)

Network > Interfaces > Edit (for tunnel.1) > NHTB: Enter the following, then click **Add**:

New Next Hop Entry:
IP Address: 10.0.0.100
VPN: hub-b

5. **Configure ACVPN-Dynamic**

VPNs > AutoKey Advanced > Gateway > New: Configure the IKE gateway, click **Advanced** and set the security level, then click **Return** to go back to the IKE gateway configuration page and click **OK**:

Gateway Name: ac-gw
ACVPN-Dynamic: (select)

VPNs > AutoKey IKE > New: Configure the ACVPN profile, click **Advanced** and set the security level, then click **Return** to go back to the VPN configuration page and click **OK**:

VPN Name: ac-vpn
ACVPN-Dynamic: (select)
Gateway: (select) ac-gw

6. **Configuration on Vrouter**

Network > Routing > Virtual Router > (for trust-vr) Edit: Do the following:

> NHS Setting > Multiple NHS Setting: Enter the IP address of the NHS server and click **Add**, then click **Back to NHRP Setting** to go back to the NHRP Setting page:

IP: 10.0.0.1
IP: 10.0.0.100

> NHS Setting: Enter the following, then click **Apply**:

NHC Setting: (select)
ACVPN Name: (select) ac-vpn
Enable: (select)

7. **Create Route Map**

Network > Routing > Virtual Router > Route Map > New (for trust-vr): Enter the following, then click **OK**:

Map Name: rtmap2
Sequence No.: 1
Action: permit (select)
Match Properties:
Interface: (select), ethernet1

8. **Redistribute NHRP Routes**

Network > Routing > Virtual Router > Edit (for trust-vr) > Edit NHRP Instance > Redistributable Rules: Select the following, then click **Add**:

Route Map: rtmap2
Protocol: connected

CLI (Spoke2)

```
set vrouter trust-vr protocol ospf

set interface tunnel.1 zone trust
set interface tunnel.1 ip 10.0.0.3/24

set ike gateway hub-m address 100.100.100.1 outgoing-interface ethernet0/1
preshare Juniper sec-level standard
set ike gateway hub-b address 100.100.100.100 outgoing-interface ethernet0/1
preshare Juniper sec-level standard

set ike gateway ac-gw acvpn-dynamic

set vpn hub-m gateway hub-m sec-level standard
set vpn hub-m monitor rekey
set vpn hub-m bind interface tunnel.1
set interface tunnel.1 nhtb 10.0.0.1 vpn hub-m

set vpn hub-b gateway hub-b sec-level standard
set vpn hub-b monitor rekey
set vpn hub-b bind interface tunnel.1
set interface tunnel.1 nhtb 10.0.0.100 vpn hub-b

set vpn ac-vpn acvpn-dynamic ac-gw

set vrouter trust-vr protocol nhrp
set vrouter trust-vr protocol nhrp acvpn-dynamic ac-vpn
set vrouter trust-vr protocol nhrp nhs 10.0.0.1
```

```
set vrouter trust-vr protocol nhrp nhs 10.0.0.100
set interface tunnel.1 protocol nhrp enable
unset interface tunnel.1 acvpn-dynamic-routing

set vrouter trust-vr
set route-map name rtmap2 permit 1
set match interface ethernet1
exit

set protocol nhrp
set redistribute route-map rtmap2 protocol connected
exit
exit

set interface tunnel.1 protocol ospf area 0
set interface tunnel.1 protocol ospf link-type p2mp
set interface tunnel.1 protocol ospf enable
set interface ethernet0/0 protocol ospf enable
```

Part 6
Voice-over-Internet Protocol

Voice-over-Internet Protocol describes the supported VoIP Application Layer Gateways (ALGs) and contains the following chapters:

- "H.323 Application Layer Gateway" on page 1091 describes the H.323 protocol and provides examples of typical scenarios.
- "Session Initiation Protocol Application Layer Gateway" on page 1105 describes the Session Initiation Protocol (SIP) and shows how the SIP ALG processes calls in Route and Network Address Translation (NAT) modes. Examples of typical scenarios follow a summary of the SIP architecture.
- "Media Gateway Control Protocol Application Layer Gateway" on page 1157 presents an overview of the Media Gateway Control Protocol (MGCP) ALG and lists the firewall security features of the implementation. Examples of typical scenarios follow a summary of the MGCP architecture.
- "Skinny Client Control Protocol Application Layer Gateway" on page 1171 presents an overview of the Skinny Client Control Protocol (SCCP) ALG and lists the firewall security features of the implementation. Examples of typical scenarios follow a summary of the SCCP architecture.
- "Apple iChat Application Layer Gateway" on page 1203 presents an overview of the AppleiChat ALG and lists the firewall security features of the implementation. Examples of typical scenarios follow a summary of the AppleiChat architecture.

Chapter 27

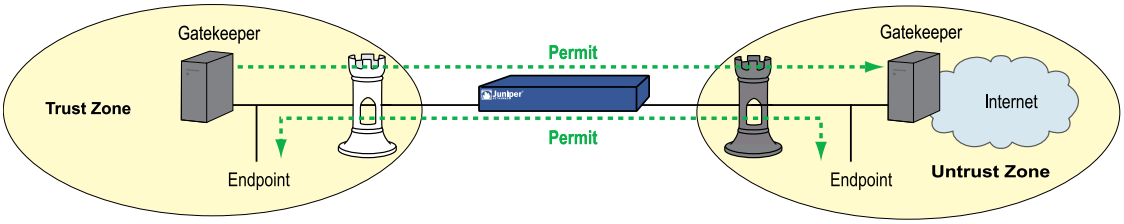
H.323 Application Layer Gateway

This chapter describes the H.323 protocol and provides examples for configuring the H.323 Application Layer Gateway (ALG) on a Juniper Networks security device. This chapter contains the following sections:

Overview

The H.323 Application Layer Gateway (ALG) allows you secure voice over IP (VoIP) communication between terminal endpoints such as IP phones and multimedia devices. In such a telephony system, gatekeeper devices manage call registration, admission, and call status for VoIP calls. Gatekeepers can reside in the two different zones or in the same zone.

Figure 286: H.323 Protocol



NOTE: Illustrations in this chapter use IP phones for illustrative purposes, although it is possible to make configurations for other hosts that use VoIP, such as NetMeeting multimedia devices.

Alternate Gatekeeper

The H.323 ALG in ScreenOS supports the use of an alternate gatekeeper. All the IP end points must register with a gatekeeper through the Registration, Admission, and Status (RAS) protocol before they can attempt calls between them. During the registration process, the primary gatekeeper sends Gatekeeper Confirm (GCF) and

Registration Confirm (RCF) messages to the endpoint. These messages contain the list of available alternate gatekeepers.

The alternate gatekeeper provides high availability, redundancy and scalability for the IP end points. If the primary gatekeeper fails, IP-enabled phones and other multimedia devices registered with that gatekeeper are registered with the alternate gatekeeper.

You can configure the primary and alternate gatekeepers in the Trust, Untrust, or DMZ zones.

NOTE: Currently, the Juniper Networks H.323 ALG supports the gatekeeper and the alternate gatekeeper in the same zone.

To use the alternate gatekeeper feature, you need to configure a security policy that allows the endpoint device to reach the alternate gatekeeper when the endpoint cannot reach the primary gatekeeper.

Examples

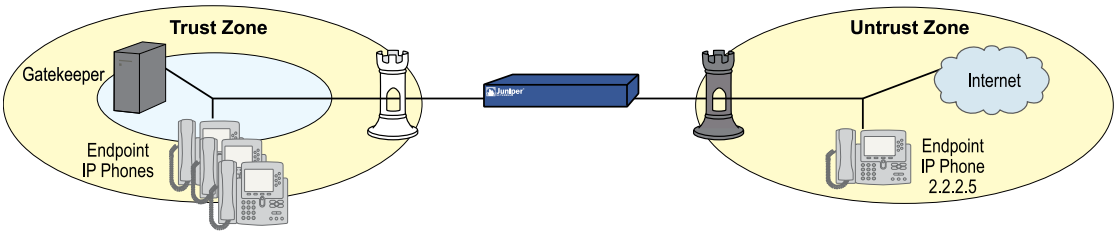
This section contains the following configuration scenarios:

- Example: Gatekeeper in the Trust Zone on page 1092
- Example: Gatekeeper in the Untrust Zone on page 1093
- Example: Outgoing Calls with NAT on page 1095
- Example: Incoming Calls with NAT on page 1098
- Example: Gatekeeper in the Untrust Zone with NAT on page 1101

Example: Gatekeeper in the Trust Zone

In the following example, you set up two policies that allow H.323 traffic to pass between IP phone hosts and a gatekeeper in the Trust zone, and an IP phone host (2.2.2.5) in the Untrust zone. In this example, the security device can be in either transparent mode or route mode. Both the Trust and Untrust security zones are in the trust-vr routing domain.

Figure 287: H.323 Gatekeeper in the Trust Zone



WebUI

1. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: IP_Phone
   IP Address/Domain Name:
   IP/Netmask: (select), 2.2.2.5/32
   Zone: Untrust

2. **Policies**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), IP_Phone
   Service: H.323
   Action: Permit

   Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), IP_Phone
   Destination Address:
   Address Book Entry: (select), Any
   Service: H.323
   Action: Permit

CLI

1. **Address**

   ```
   set address untrust IP_Phone 2.2.2.5/32
   ```

2. **Policies**

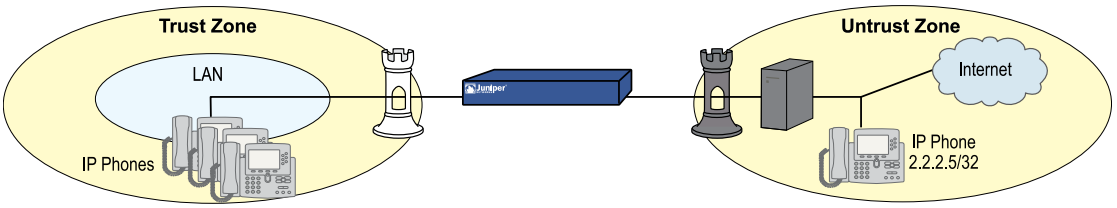
   ```
   set policy from trust to untrust any IP_Phone h.323 permit
   set policy from untrust to trust IP_Phone any h.323 permit
   save
   ```

Example: Gatekeeper in the Untrust Zone

Because transparent and route modes do not require address mapping of any kind, security device configuration for a gatekeeper in the Untrust zone is usually identical to the configuration for a gatekeeper in the Trust zone.

In the following example, you set up two policies to allow H.323 traffic to pass between IP phone hosts in the Trust zone, and the IP phone at IP address 2.2.2.5 (and the gatekeeper) in the Untrust zone. The device can be in transparent or route mode. Both the Trust and Untrust security zones are in the trust-vr routing domain.

Figure 288: H.323 Gatekeeper in the Untrust Zone



WebUI

1. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: IP_Phone
   IP Address/Domain Name:
   IP/Netmask: (select), 2.2.2.5/32
   Zone: Untrust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Gatekeeper
   IP Address/Domain Name:
   IP/Netmask: (select), 2.2.2.10/32
   Zone: Untrust

2. **Policies**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), IP_Phone
   Service: H.323
   Action: Permit

   Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), IP_Phone
   Destination Address:
   Address Book Entry: (select), Any
   Service: H.323
   Action: Permit

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), Any
Destination Address:
  Address Book Entry: (select), Gatekeeper
Service: H.323
Action: Permit

CLI

1. **Addresses**

```
set address untrust IP_Phone 2.2.2.5/32
set address untrust gatekeeper 2.2.2.10/32
```

2. **Policies**

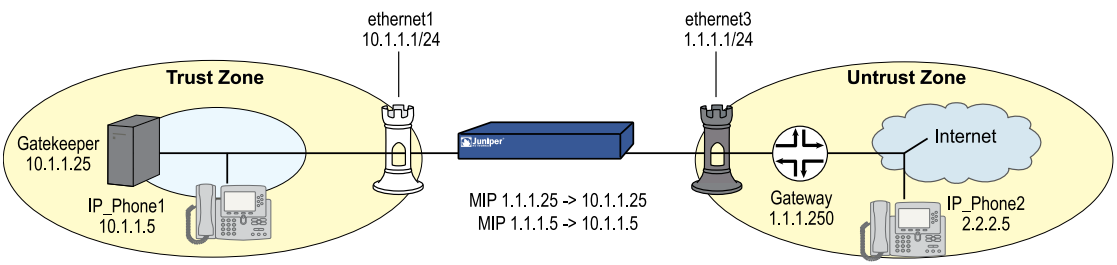
```
set policy from trust to untrust any IP_Phone h.323 permit
set policy from trust to untrust any gatekeeper h.323 permit
set policy from untrust to trust IP_Phone any h.323 permit
set policy from untrust to trust gatekeeper any h.323 permit
save
```

Example: Outgoing Calls with NAT

When the security device uses NAT (Network Address Translation), a gatekeeper or endpoint device in the Trust zone has a private address, and when it is in the Untrust zone it has a public address. When you set a security device in NAT mode, you must map a public IP address to each device that needs to receive incoming traffic with a private address.

In this example, the devices in the Trust zone include the endpoint host (10.1.1.5) and the gatekeeper device (10.1.1.25). IP_Phone2 (2.2.2.5) is in the Untrust zone. You configure the security device to allow traffic between the endpoint host IP_Phone1 and the gatekeeper in the Trust zone and the endpoint host IP_Phone2 in the Untrust zone. Both the Trust and Untrust security zones are in the trust-vr routing domain.

Figure 289: Network Address Translation—Outgoing Calls



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: IP_Phone1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.5/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Gatekeeper
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.25/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: IP_Phone2
   IP Address/Domain Name:
   IP/Netmask: (select), 2.2.2.5/32
   Zone: Untrust

3. **Mapped IP Addresses**

   Network > Interfaces > Edit (for ethernet3) > MIP > New: Enter the following, then click **OK**:

   Mapped IP: 1.1.1.5
   Netmask: 255.255.255.255

Host IP Address: 10.1.1.5
Host Virtual Router Name: trust-vr

Network > Interfaces > Edit (for ethernet3) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 1.1.1.25
Netmask: 255.255.255.255
Host IP Address: 10.1.1.25
Host Virtual Router Name: trust-vr

4. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
  Interface: ethernet3
  Gateway IP Address: 1.1.1.250

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), IP_Phone1
Destination Address:
  Address Book Entry: (select), IP_Phone2
Service: H.323
Action: Permit

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), Gatekeeper
Destination Address:
  Address Book Entry: (select), IP_Phone2
Service: H.323
Action: Permit

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), IP_Phone2
Destination Address:
  Address Book Entry: (select), MIP(1.1.1.5)
Service: H.323
Action: Permit

Policies 7gt; (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), IP_Phone2
Destination Address:
  Address Book Entry: (select), MIP(1.1.1.25)

Service: H.323
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address trust IP_Phone1 10.1.1.5/32
set address trust gatekeeper 10.1.1.25/32
set address untrust IP_Phone2 2.2.2.5/32
```

3. **Mapped IP Addresses**

```
set interface ethernet3 mip 1.1.1.5 host 10.1.1.5
set interface ethernet3 mip 1.1.1.25 host 10.1.1.25
```

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

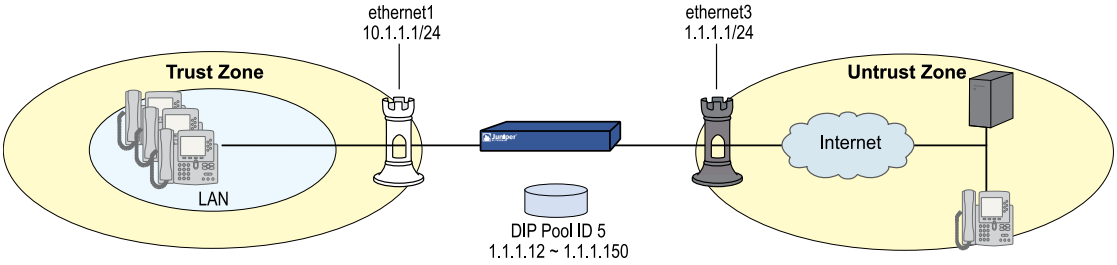
5. **Policies**

```
set policy from trust to untrust IP_Phone1 IP_Phone2 h.323 permit
set policy from trust to untrust gatekeeper IP_Phone2 h.323 permit
set policy from untrust to trust IP_Phone2 mip(1.1.1.5) h.323 permit
set policy from untrust to trust IP_Phone2 mip (1.1.1.25) h.323 permit
save
```

Example: Incoming Calls with NAT

In this example, you configure the security device to accept incoming calls over a NAT boundary. To do this, you can create a DIP address pool for dynamically allocating destination addresses. This differs from most configurations, where a DIP pool provides source addresses only.

Figure 290: Network Address Translation—Incoming Calls



The name of the DIP pool can be DIP(id_num) for a user-defined DIP, or DIP(interface) when the DIP pool uses the same address as an interface IP address. You can use such address entries as destination addresses in policies, together with the services H.323, SIP, or other VoIP (Voice-over-IP) protocols, to support incoming calls.

The following example uses DIP in an H.323 VoIP configuration. The keyword " incoming" instructs the device to add the DIP and interface addresses to the global zone.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **DIP with Incoming NAT**

   Network > Interface > Edit (for ethernet3) > DIP > New: Enter the following, then click **OK**:

   ID: 5
   IP Address Range: (select), 1.1.1.12 ~ 1.1.1.150
   Port Translation: (select)
   In the same subnet as the interface IP or its secondary IPs: (select)
   Incoming NAT: (select)

3. **Addresses**

   Policy > Policy Elements > Addresses > List > New (for Trust): Enter the following, then click **OK**:

Address Name: IP_Phones1
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.5/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New (for Untrust): Enter the following, then click **OK**:

Address Name: IP_Phone2
IP Address/Domain Name:
IP/Netmask: (select), 2.2.2.5/32
Zone: Untrust

4. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), IP_Phones1
Destination Address:
Address Book Entry: (select), Any
Service: H.323
Action: Permit

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), IP_Phone2
Destination Address:
Address Book Entry: (select), DIP(5)
Service: H.323
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **DIP with Incoming NAT**

```
set interface ethernet3 dip 5 1.1.1.12 1.1.1.150 incoming
```

3. **Addresses**

```
set address trust IP_Phones1 10.1.1.5/24
set address untrust IP_Phone2 2.2.2.5/32
```

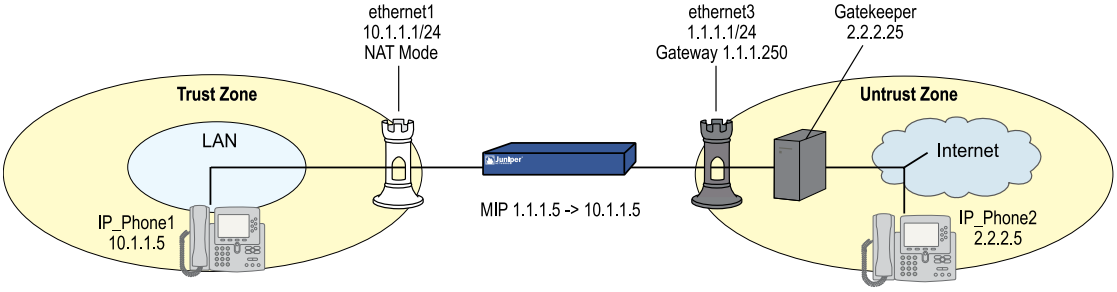
4. **Policies**

```
set policy from trust to untrust IP_Phones1 any h.323 nat src dip 5 permit
set policy from untrust to trust IP_Phone2 dip(5) h.323 permit
save
```

Example: Gatekeeper in the Untrust Zone with NAT

In this example, the gatekeeper device (2.2.2.25) and host IP_Phone2 (2.2.2.5) are in the Untrust zone and host IP_Phone1 (10.1.1.5) is in the Trust zone. You configure the security device to allow traffic between host IP_Phone1 in the Trust zone and host IP_Phone2 (and the gatekeeper) in the Untrust zone. Both the Trust and Untrust security zones are in the trust-vr routing domain.

Figure 291: Gatekeeper in the Untrust Zone



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: IP_Phone1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.5/32
   Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Gatekeeper
IP Address/Domain Name:
IP/Netmask: (select), 2.2.2.25/32
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: IP_Phone2
IP Address/Domain Name:
IP/Netmask: (select), 2.2.2.5/32
Zone: Untrust

3. **Mapped IP Address**

Network > Interfaces > Edit (for ethernet3) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 1.1.1.5
Netmask: 255.255.255.255
Host IP Address: 10.1.1.5

4. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

5. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), IP_Phone1
Destination Address:
Address Book Entry: (select), IP_Phone2
Service: H.323
Action: Permit

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), IP_Phone1
Destination Address:
Address Book Entry: (select), Gatekeeper
Service: H.323
Action: Permit

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), IP_Phone2
Destination Address:
  Address Book Entry: (select), MIP(1.1.1.5)
Service: H.323
Action: Permit

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Gatekeeper
Destination Address:
Address Book Entry: (select), MIP(1.1.1.5)
Service: H.323
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address trust IP_Phone1 10.1.1.5/32
set address untrust gatekeeper 2.2.2.25/32
set address untrust IP_Phone2 2.2.2.5/32
```

3. **Mapped IP Addresses**

```
set interface ethernet3 mip 1.1.1.5 host 10.1.1.5
```

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

5. **Policies**

```
set policy from trust to untrust IP_Phone1 IP_Phone2 h.323 permit
set policy from trust to untrust IP_Phone1 gatekeeper h.323 permit
set policy from untrust to trust IP_Phone2 mip(1.1.1.5) h.323 permit
set policy from untrust to trust gatekeeper mip(1.1.1.5) h.323 permit
save
```

Chapter 28

Session Initiation Protocol Application Layer Gateway

This chapter describes the Session Initiation Protocol (SIP) Application Layer Gateway (ALG) and contains the following sections:

Overview

Session Initiation Protocol (SIP) is an Internet Engineering Task Force (IETF)-standard protocol for initiating, modifying, and terminating multimedia sessions over the Internet. Such sessions might include conferencing, telephony, or multimedia, with features such as instant messaging and application-level mobility in network environments.

Juniper Networks security devices support SIP as a service and can screen SIP traffic, allowing and denying it based on a policy that you configure. SIP is a predefined service in ScreenOS and uses port 5060 as the destination port.

SIP's primary function is to distribute session-description information and, during the session, to negotiate and modify the parameters of the session. SIP is also used to terminate a multimedia session.

Session-description information is included in INVITE and ACK messages and indicates the multimedia type of the session, for example, voice or video. Although SIP can use different description protocols to describe the session, the Juniper Networks SIP ALG supports only Session Description Protocol (SDP).

SDP provides information that a system can use to join a multimedia session. SDP might include information such as IP addresses, port numbers, times, and dates. Note that the IP address and port number in the SDP header (the " c = " and " m = " fields, respectively) are the address and port where the client wants to receive the media streams and not the IP address and port number from which the SIP request originates (although they can be the same). See "Session Description Protocol Sessions" on page 1109 for more information.

SIP messages consist of requests from a client to a server and responses to the requests from a server to a client with the purpose of establishing a session (or a

call). A User Agent (UA) is an application that runs at the endpoints of the call and consists of two parts: the User Agent Client (UAC), which sends SIP requests on behalf of the user; and a User Agent Server (UAS), which listens to the responses and notifies the user when they arrive. Examples of UAs are SIP proxy servers and phones.

SIP Request Methods

The SIP transaction model includes a number of request and response messages, each of which contains a method field that denotes the purpose of the message. ScreenOS supports the following method types and response codes:

- INVITE—A user sends an INVITE request to invite another user to participate in a session. The body of an INVITE request may contain the description of the session. In NAT mode, the IP addresses in the Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified as shown in Table 73 on page 1119.
- ACK—The user from whom the INVITE originated sends an ACK request to confirm reception of the final response to the INVITE request. If the original INVITE request did not contain the session description, the ACK request must include it. In NAT mode, the IP addresses in the Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified as shown in Table 73 on page 1119.
- OPTIONS—Used by the User Agent (UA) to obtain information about the capabilities of the SIP proxy. A server responds with information about what methods, session description protocols, and message encoding it supports. In NAT mode, when the OPTIONS request is sent from a UA outside NAT to a proxy inside NAT, the SIP ALG translates the address in the Request-URI and the IP address in the To: field to the appropriate IP address of the internal client. When the UA is inside NAT and the proxy is outside NAT, the SIP ALG translates the From:, Via:, and Call-ID: fields as shown in Table 73 on page 1119.
- BYE—A user sends a BYE request to abandon a session. A BYE request from either user automatically terminates the session. In NAT mode, the IP addresses in the Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified as shown in Table 73 on page 1119.
- CANCEL—A user can send a CANCEL request to cancel a pending INVITE request. A CANCEL request has no effect if the SIP server processing the INVITE had sent a final response for the INVITE before it received the CANCEL. In NAT mode, the IP addresses in the Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified as shown in Table 73 on page 1119.
- REGISTER—A user sends a REGISTER request to a SIP registrar server to inform it of the current location of the user. A SIP registrar server records all the information it receives in REGISTER requests and makes this information available to any SIP server attempting to locate a user. In NAT mode, REGISTER requests are handled as follows:
  - REGISTER requests from an external client to an internal registrar—When the SIP ALG receives the incoming REGISTER request it translates the IP address, if any, in the Request-URI. Incoming REGISTER messages are allowed only to a MIP or VIP address. No translation is needed for the outgoing response.

  - REGISTER requests from an internal client to an external registrar—When the SIP ALG receives the outgoing REGISTER request it translates the IP addresses in the To:, From:, Via:, Call-ID:, and Contact: header fields. A backward translation is performed for the incoming response.

- Info—Used to communicate mid-session signaling information along the signaling path for the call. In NAT mode, the IP addresses in the Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified as shown in Table 73 on page 1119.
- Subscribe—Used to request current state and state updates from a remote node. In NAT mode, the address in the Request-URI is changed to a private IP address if the messages is coming from the external network into the internal network. The IP addresses in Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified as shown in the table in Table 73 on page 1119.
- Notify—Sent to inform subscribers of changes in state to which the subscriber has a subscription. In NAT mode, the IP address in the Request-URI: header field is changed to a private IP address if the message is coming from the external network into the internal network. The IP address in the Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified as shown in Table 73 on page 1119.
- Refer—Used to refer the recipient (identified by the Request-URI) to a third party by the contact information provided in the request. In NAT mode, the IP address in the Request-URI is changed to a private IP address if the message is coming from the external network into the internal network. The IP addresses in the Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified as shown in Table 73 on page 1119.

  For example, if user A in a private network refers user B, in a public network, to user C, who is also in the private network, the SIP ALG allocates a new IP address and port number for user C so that user C can be contacted by user B. If user C is registered with a registrar, however, its port mapping is stored in the ALG NAT table and is reused to perform the translation.
- Update—Used to open pinhole for new or updated SDP information. The Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified as shown in Table 73 on page 1119.
- 1xx, 202, 2xx, 3xx, 4xx, 5xx, 6xx Response Codes—Used to indicate the status of a transaction. Header fields are modified as shown in Table 73 on page 1119.

Classes of SIP Responses

SIP responses provide status information about SIP transactions and include a response code and a reason phrase. SIP responses are grouped into the following classes:

- Informational (100 to 199)—Request received, continuing to process the request.
- Success (200 to 299)—Action successfully received, understood, and accepted.
- Redirection (300 to 399)—Further action required to complete the request.
- Client Error (400 to 499)—Request contains bad syntax or cannot be fulfilled at this server.

- Server Error (500 to 599)—Server failed to fulfill an apparently valid request.
- Global Failure (600 to 699)—Request cannot be fulfilled at any server.

Table 72 on page 1108 provides a complete list of current SIP responses, all of which are supported on Juniper Networks security devices.

Table 72: Session Initiation Protocol Responses

Class	Response Code-Reason Phrase	Response Code-Reason Phrase	Response Code-Reason Phrase
Informational	100 Trying	180 Ringing	181 Call is being forwarded
	182 Queued	183 Session progress	
Success	200 OK	202 Accepted	
Redirection	300 Multiple choices	301 Moved permanently	302 Moved temporarily
	305 Use proxy	380 Alternative service	
Client Error	400 Bad request	401 Unauthorized	402 Payment required
	403 Forbidden	404 Not found	405 Method not allowed
	406 Not acceptable	407 Proxy authentication required	408 Request time-out
	409 Conflict	410 Gone	411 Length required
	413 Request entity too large	414 Request-URL too large	415 Unsupported media type
	420 Bad extension	480 Temporarily not available	481 Call leg/transaction does not exist
	482 Loop detected	483 Too many hops	484 Address incomplete
	485 Ambiguous	486 Busy here	487 Request canceled
	488 Not acceptable here		
Server Error	500 Server internal error	501 Not implemented	502 Bad gateway
	502 Service unavailable	504 Gateway time-out	505 SIP version not supported
Global Failure	600 Busy everywhere	603 Decline	604 Does not exist anywhere
	606 Not acceptable		

SIP Application Layer Gateway

There are two types of SIP traffic, the signaling and the media stream. SIP signaling traffic consists of request and response messages between client and server and uses transport protocols such as User Datagram Protocol (UDP) or Transmission Control

Protocol (TCP). The media stream carries the data (audio data, for example) and uses Application Layer protocols such as Real Time Protocol (RTP) over UDP.

Juniper Networks security devices support SIP signaling messages on port 5060. You can simply create a policy that permits SIP service, and the security device filters SIP signaling traffic like any other type of traffic, permitting or denying it. The media stream, however, uses dynamically assigned port numbers that can change several times during the course of a call. Without fixed ports, it is impossible to create a static policy to control media traffic. In this case, the security device invokes the SIP ALG. The SIP ALG reads SIP messages and their SDP content and extracts the port-number information it needs to dynamically open pinholes and let the media stream traverse the security device.

NOTE: We refer to a pinhole as the limited opening of a port to allow exclusive traffic.

The SIP ALG monitors SIP transactions and dynamically creates and manages pinholes based on the information it extracts from these transactions. The Juniper Networks SIP ALG supports all SIP methods and responses (see “SIP Request Methods” on page 1106 and “Classes of SIP Responses” on page 1107). You can allow SIP transactions to traverse the Juniper Networks firewall by creating a static policy that permits SIP service. This policy enables the security device to intercept SIP traffic and do one of the following actions: permit or deny the traffic or enable the SIP ALG to open pinholes to pass the media stream. The SIP ALG needs to open pinholes only for the SIP requests and responses that contain media information (SDP). For SIP messages that do not contain SDP, the security device simply lets them through.

The SIP ALG intercepts SIP messages that contain SDP and, using a parser, extracts the information it requires to create pinholes. The SIP ALG examines the SDP portion of the packet, and a parser extracts information such as IP addresses and port numbers, which the SIP ALG records in a pinhole table. The SIP ALG uses the IP addresses and port numbers recorded in the pinhole table to open pinholes and allow media streams to traverse the security device.

The SIP ALG for IPv6 supports Netscreen Redundancy Protocol (NSRP).

NOTE: Juniper Networks security devices do not support encrypted SDP. If a security device receives a SIP message in which SDP is encrypted, the SIP ALG permits it through the firewall but generates a log message informing the user that it cannot process the packet. If SDP is encrypted, the SIP ALG cannot extract the information it needs from SDP to open pinholes. As a result, the media content that SDP describes cannot traverse the security device.

Session Description Protocol Sessions

An SDP session description is text-based and consists of a set of lines. It can contain session-level and media-level information. The session-level information applies to the whole session, while the media-level information applies to a particular media stream. An SDP session description always contains session-level information, which

appears at the beginning of the description, and might contain media-level information, which comes after.

NOTE: In the SDP session description, the media-level information begins with the m = field.

Of the many fields in the SDP description, two are particularly useful to the SIP ALG because they contain Transport Layer information. The two fields are the following:

- **c =** for connection information

  This field can appear at the session or media level. It displays in this format:

- c = < network type > < address type > < connection address >

  Currently, the security device supports "IN" (for Internet) as the network type, "IP4 and IP6" as address types, and a unicast IP address or domain name as the destination (connection) IP address.

NOTE: Generally, the destination IP address can also be a multicast IP address, but ScreenOS does not currently support multicast with SIP.

  If the destination IP address is a unicast IP address, the SIP ALG creates pinholes using the IP address and port numbers specified in the media description field m = .

- **m =** for media announcement

  This field appears at the media level and contains the description of the media. It displays in this format:

  m = < media > < port > < transport > < fmt list >

  Currently, the security device supports only “ audio” as the media and “ RTP” as the Application Layer transport protocol. The port number indicates the destination (not the origin) of the media stream. The format list (fmt list) provides information about the Application Layer protocol that the media uses.

  In this release of ScreenOS, the security device opens ports only for RTP and RTCP. Every RTP session has a corresponding Real Time Control Protocol (RTCP) session. Therefore, whenever a media stream uses RTP, the SIP ALG must reserve ports (create pinholes) for both RTP and RTCP traffic. By default, the port number for RTCP is one higher than the RTP port number.

NOTE: Generally, the destination IP address can also be a multicast IP address, but ScreenOS does not currently support multicast with SIP.

Pinhole Creation

Both pinholes for the RTP and RTCP traffic share the same destination IP address. The IP address comes from the c= field in the SDP session description. Because the c= field can appear in either the session-level or media-level portion of the SDP session description, the parser determines the IP address based on the following rules (in accordance with SDP conventions):

- First, the SIP ALG parser verifies if there is a c= field containing an IP address in the media level. If there is one, the parser extracts that IP address, and the SIP ALG uses it to create a pinhole for the media.
- If there is no c= field in the media level, the SIP ALG parser extracts the IP address from the c= field in the session level, and the SIP ALG uses it to create a pinhole for the media. If the session description does not contain a c= field in either level, this indicates an error in the protocol stack, and the security device drops the packet and logs the event.

The following lists the information the SIP ALG needs to create a pinhole. This information comes from the SDP session description and parameters on the security device:

- Protocol: UDP.
- Source IP: Unknown.
- Source port: Unknown.
- Destination IP: The parser extracts the destination IP address from the c= field in the media or session level.
- Destination port: The parser extracts the destination port number for RTP from the m= field in the media level and calculates the destination port number for RTCP using the following formula:

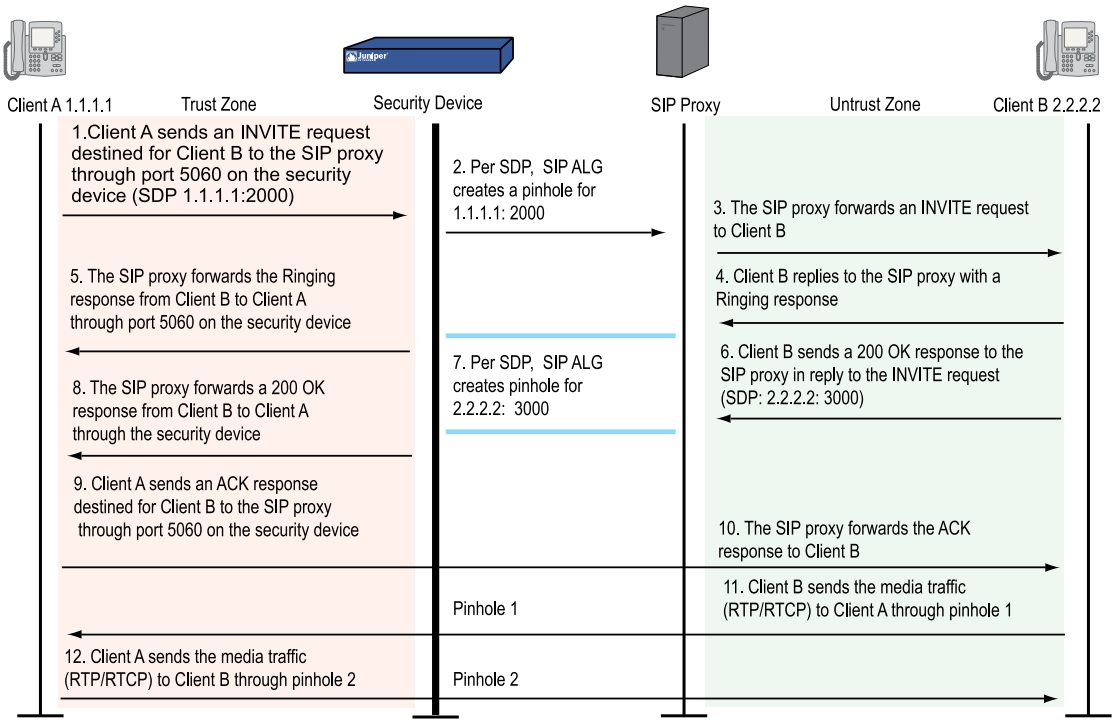
  RTP port number + one

- Lifetime: This value indicates the length of time (in seconds) during which a pinhole is open to allow a packet through. A packet must go through the pinhole before the lifetime expires. When the lifetime expires, the SIP ALG removes the pinhole.

  When a packet goes through the pinhole within the lifetime period, immediately afterwards the SIP ALG removes the pinhole for the direction from which the packet came.

  Figure 292 on page 1112 describes a call setup between two SIP clients and how the SIP ALG creates pinholes to allow RTP and RTCP traffic. The illustration assumes that the security device has a policy that permits SIP, thus opening port 5060 for SIP signaling messages.

Figure 292: SIP ALG Call Setup



NOTE: The SIP ALG does not create pinholes for RTP and RTCP traffic when the destination IP address is 0.0.0.0, which indicates that the session is on hold. To put a session on hold, during a telephone communication, for example, a user (User A) sends the other user (User B) a SIP message in which the destination IP address is 0.0.0.0. Doing so indicates to User B not to send any media until further notice. If User B sends media anyway, the security device drops the packets.

Session Inactivity Timeout

Typically a call ends when one of the clients sends a BYE or CANCEL request. The SIP ALG intercepts the BYE or CANCEL request and removes all media sessions for that call. There could be reasons or problems preventing clients in a call from sending BYE or CANCEL requests, for example, a power failure. In this case, the call might go on indefinitely, consuming resources on the security device. The inactivity-timeout feature helps the security device to monitor the liveliness of the call and terminate it if there is no activity for a specific period of time.

A call can have one or more voice channels. Each voice channel has two sessions (or two media streams), one for RTP and one for RTCP. When managing the sessions, the security device considers the sessions in each voice channel as one group. Settings such as the inactivity timeout apply to a group as opposed to each session.

There are two types of inactivity timeouts that determine the lifetime of a group:

- Signaling-inactivity timeout: This parameter indicates the maximum length of time (in seconds) a call can remain active without any SIP-signaling traffic. Each time a SIP-signaling message occurs within a call, this timeout resets. The default setting is 43200 seconds (12 hours).
- Media-inactivity timeout: This parameter indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. The default setting is 120 seconds.

If either of these timeouts expires, the security device removes all sessions for this call from its table, thus terminating the call.

SIP Attack Protection

The ability of the SIP proxy server to process calls can be affected by repeat SIP INVITE requests, whether malicious or through client or server error, that it initially denied. To prevent the SIP proxy server from being overwhelmed by such requests, you can use the **sip protect deny** command to configure the security device to monitor INVITE requests and proxy server replies to them. The **sip protect deny** command supports both IPv4 and IPv6 addresses. If a reply contains a 3xx, 4xx, or 5xx response code (see "Classes of SIP Responses" on page 1107), the ALG stores the source IP address of the request and the IP address of the proxy server in a table. Subsequently, the security device checks all INVITE requests against this table and, for a configurable number of seconds (the default is 3), discards any packets that match entries in the table. You can also configure the security device to monitor INVITE request to a specific proxy server by specifying the destination IP address. SIP attack protection is configured globally.

Example: SIP Protect Deny

In this example, you configure the security device to protect a single SIP proxy server (1.1.1.3/24) from repeat SIP requests to which it has already denied service. Packets are dropped for a period of 5 seconds, after which the security device resumes forwarding INVITE requests from those sources.

WebUI

You must use the CLI to protect SIP proxy servers from being inundated by SIP messages.

CLI

```
set alg sip app-screen protect deny dst-ip 1.1.1.3/24
set alg sip protect deny timeout 5
save
```

Example: Signaling-Inactivity and Media-Inactivity Timeouts

In this example, you configure the signaling-inactivity timeout to 30,000 seconds and the media-inactivity timeout to 90 seconds.

WebUI

NOTE: You must use the CLI to set SIP-signaling and media-inactivity timeouts.

CLI

```
set alg sip signaling-inactivity-timeout 30000
set alg sip media-inactivity-timeout 90
save
```

Example: UDP Flooding Protection

You can protect the security device against UDP flooding by zone and destination address. In this example, you set a threshold of 80,000 per second for the number of UDP packets that can be received on IP address 1.1.1.5, in the Untrust zone, before the security device generates an alarm and drops subsequent packets for the remainder of that second.

NOTE: This example uses a general ScreenOS command and is not necessarily SIP-specific. For more information about UDP flood protection and how to determine effective settings, see “UDP Flood” on page 489.

WebUI

Security > Screening > Screen: Enter the following, then click **Apply**:

```
Zone: Untrust
UDP Flood Protection (select)
```

> Destination IP: Enter the following, then click the Back arrow in your browser to return to the Screen configuration page:

```
Destination IP: 1.1.1.5
Threshold: 80000
Add: (select)
```

CLI

```
set zone untrust screen udp-flood dst-ip 1.1.1.5 threshold 80000
save
```

Example: SIP Connection Maximum

In this example, you prevent flood attacks on the SIP network from attackers in the Untrust zone by setting a maximum of 20 concurrent sessions from a single IP address. If the security device detects more than 20 connection attempts from the

same IP address, it begins dropping subsequent attempts until the number of sessions drops below the specified maximum.

NOTE: This example uses a general ScreenOS command and is not necessarily SIP-specific. For more information about source-based session limits and how to determine effective settings, see "Source-Based and Destination-Based Session Limits" on page 464

WebUI

Screening > Screen (Zone: Untrust): Enter the following, then click **OK**:

Source IP Based Session Limit: (select)
Threshold: 20 Sessions

CLI

```
set zone untrust screen limit-session source-ip-based 20
save
```

SIP with Network Address Translation

The Network Address Translation (NAT) protocol enables multiple hosts in a private subnet to share a single public IP address to access the Internet. For outgoing traffic, NAT replaces the private IP address of the host in the private subnet with the public IP address. For incoming traffic, the public IP address is converted back into the private address, and the message is routed to the appropriate host in the private subnet.

Using NAT with the SIP service is more complicated because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. The SIP headers contain information about the caller and the receiver, and the security device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media. The security device translates SDP information for allocating resources to send and receive the media.

How IP addresses and port numbers in SIP messages are replaced depends on the direction of the message. For an outgoing message, the private IP address and port number of the client are replaced with the public IP address and port number of the Juniper Networks firewall. For an incoming message, the public address of the firewall is replaced with the private address of the client.

When an INVITE message is sent out across the firewall, the SIP ALG collects information from the message header into a call table, which it uses to forward subsequent messages to the correct end point. When a new message arrives, for example an ACK or 200 OK, the ALG compares the From:, To:, and Call-ID: fields against the call table to identify the call context of the message. If a new INVITE message arrives that matches the existing call, the ALG processes it as a REINVITE.

When a message containing SDP information arrives, the ALG allocates ports and creates a NAT mapping between them and the ports in the SDP. Because the SDP requires sequential ports for the Real Time Protocol (RTP) and Real Time Control Protocol (RTCP) channels, the ALG provides consecutive even-odd ports. If it is unable to find a pair of ports it discards the SIP message.

Outgoing Calls

When a SIP call is initiated with a SIP request message from the internal to the external network, NAT replaces the IP addresses and port numbers in the SDP and creates a binding to map the IP addresses and port numbers to the Juniper Networks firewall. Via:, Contact:, Route:, and Record-Route: SIP header fields, if present, are also bound to the firewall IP address. The ALG stores these mappings for use in retransmissions and for SIP response messages.

The SIP ALG then opens pinholes in the firewall to allow media through the security device on the dynamically assigned ports negotiated based on information in the SDP and the Via:, Contact:, and Record-Route: header fields. The pinholes also allow incoming packets to reach the Contact:, Via:, and Record-Route: IP addresses and ports. When processing return traffic, the ALG inserts the original Contact:, Via:, Route:, and Record-Route: SIP fields back into the packets.

Incoming Calls

Incoming calls are initiated from the public network to public Mapped IP (MIP) addresses or to interface IP addresses on the security device. MIPs are statically configured IP addresses that point to internal hosts; interface IP addresses are dynamically recorded by the ALG as it monitors REGISTER messages sent by internal hosts to the SIP registrar. (For more information, see “Examples” on page 1122.) When the security device receives an incoming SIP packet, it sets up a session and forwards the payload of the packet to the SIP ALG.

The ALG examines the SIP request message (initially an INVITE) and, based on information in the SDP, opens gates for outgoing media. When a 200 OK response message arrives, the SIP ALG performs NAT on the IP addresses and ports and opens pinholes in the outbound direction. (The opened gates have a short time-to-live, and time out if a 200 OK response message is not received quickly.)

When a 200 OK response arrives, the SIP proxy examines the SDP information and reads the IP addresses and port numbers for each media session. The SIP ALG on the security device performs NAT on the addresses and port numbers, opens pinholes for outbound traffic, and refreshes the timeout for gates in the inbound direction.

When the ACK arrives for the 200 OK, it also passes through the SIP ALG. If the message contains SDP information, the SIP ALG ensures that the IP addresses and port numbers are not changed from the previous INVITE—if they are, the ALG deletes old pinholes and creates new pinholes to allow media to pass through. The ALG also monitors the Via:, Contact:, and Record-Route: SIP fields and opens new pinholes if it determines that these fields have changed.

Forwarded Calls

A forwarded call is when, for example, user A outside the network calls user B inside the network, and user B forwards the call to user C outside the network. The SIP ALG processes the INVITE from user A as a normal incoming call. But when the ALG examines the forwarded call from B to C outside the network and notices that B and C are reached using the same interface, it does not open pinholes in the firewall, because media will flow directly between user A and user C.

Call Termination

The BYE message is used to terminate a call. When the security device receives a BYE message, it translates the header fields just as it does for any other message, But because a BYE message must be acknowledged by the receiver with a 200 OK, the ALG delays call teardown for five seconds to allow time for transmission of the 200 OK.

Call Re-INVITE Messages

Re-INVITE messages are used to add new media sessions to a call, and to removing existing media sessions. When new media sessions are added to a call, new pinholes are opened in the firewall and new address bindings created. The process is identical to the original call setup. When one or more media sessions are removed from a call, pinholes are closed and bindings released just as with a BYE message.

Call Session Timers

The SIP ALG uses the Session-Expires value to time out a session if a Re-INVITE or UPDATE message is not received. The ALG gets the Session-Expires value, if present, from the 200 OK response to the INVITE and uses this value for signaling timeout. If the ALG receives another INVITE before the session times out, it resets all timeout values to this new INVITE or to default values, and the process is repeated.

As a precautionary measure, the SIP ALG uses hard timeout values to set the maximum amount of time a call can exist. This ensures that the security device is protected in the event of the following:

- End systems crash during a call and a BYE message is not received.
- Malicious users never send a BYE in an attempt to attack a SIP ALG.
- Poor implementations of sip proxy fail to process Record-Route and never send a BYE message.
- Network failures prevent a BYE message from being received.

Call Cancellation

Either party can cancel a call by sending a CANCEL message. Upon receiving a CANCEL message, the SIP ALG closes pinholes through the firewall—if any have been opened—and releases address bindings. Before releasing the resources, the ALG

delays the control channel age-out for approximately five seconds to allow time for the final 200 OK to pass through. The call is terminated when the five second timeout expires, regardless of whether a 487 or non-200 response arrives.

Forking

Forking enables a SIP proxy to send a single INVITE message to multiple destinations simultaneously. When the multiple 200 OK response messages arrive for the single call, the SIP ALG parses but updates call information with the first 200 OK message it receives.

SIP Messages

The SIP message format consists of a SIP header section, and the SIP body. In request messages, the first line of the header section is the request line, which includes the method type, Request-URI, and protocol version. In response messages, the first line is the status line, which contains a status code. SIP headers contain IP addresses and port numbers used for signaling. The SIP body, separated from the header section by a blank line, is reserved for session description information, which is optional. Juniper Networks security devices currently support the Session Description Protocol (SDP) only. The SIP body contains IP addresses and port numbers used to transport the media.

In NAT mode, the security device translates information in the SIP headers to hide the information from the outside network. NAT is performed on SIP body information to allocate resources, that is, port numbers where the media is to be received.

SIP Headers

In the following sample SIP request message, NAT replaces the IP addresses in the header fields—shown in bold font—to hide them from the outside network.

INVITE bob@**10.150.20.5** SIP/2.0
Via: SIP/2.0/UDP **10.150.20.3**:5434
From: alice@**10.150.20.3**
To: bob@**10.150.20.5**
Call-ID: a12abcde@**10.150.20.3**
Contact: alice@**10.150.20.3**:5434
Route: <sip:netscreen@**10.150.20.3**:5060>
Record-Route: <sip:netscreen@**10.150.20.3**:5060>

How IP address translation is performed depends on the type and direction of the message, which can be any of the following:

- Inbound request
- Outbound response
- Outbound request
- Inbound response

Table 73 on page 1119 shows how NAT is performed in each of these cases. Note that for several of the header fields the ALG must know more than just whether the

messages comes from inside or outside the network. It must also know what client initiated the call, and whether the message is a request or response.

Table 73: Requesting Messages with NAT

Message Type	Fields	Action
Inbound Request (from public to private)	To:	Replace ALG address with local address
	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	Replace ALG address with local address
	Contact:	None
	Record-Route:	None
	Route:	None
Outbound Response (from private to public)	To:	Replace ALG address with local address
	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	N/A
	Contact:	Replace local address with ALG address
	Record-Route:	Replace local address with ALG address
	Route:	None
Outbound Request (from private to public)	To:	None
	From:	Replace local address with ALG address
	Call-ID:	Replace local address with ALG address
	Via:	Replace local address with ALG address
	Request-URI:	None
	Contact:	Replace local address with ALG address
	Record-Route:	Replace local address with ALG address
	Route:	Replace ALG address with local address

Table 73: Requesting Messages with NAT *(continued)*

Message Type	Fields	Action
Outbound Response	To:	None
(from public to private)	From:	Replace ALG address with local address
	Call-ID:	Replace ALG address with local address
	Via:	Replace ALG address with local address
	Request-URI:	N/A
	Contact:	None
	Record-Route:	Replace ALG address with local address
	Route:	Replace ALG address with local address

SIP Body

The SDP information in the SIP body includes IP addresses the ALG uses to create channels for the media stream. Translation of the SDP section also allocates resources, that is, port numbers to send and receive the media.

The following except from a sample SDP section shows the fields that are translated for resource allocation.

```
o=user 2344234 55234434 IN IP4 10.150.20.3
c=IN IP4 10.150.20.3
m=audio 43249 RTP/AVP 0
```

SIP messages can contain more than one media stream. The concept is similar to attaching multiple files to an email message. For example, an INVITE message sent from a SIP client to a SIP server might have the following fields:

```
c=IN IP4 10.123.33.4
m=audio 33445 RTP/AVP 0
c=IN IP4 10.123.33.4
m=audio 33447 RTP/AVP 0
c=IN IP4 10.123.33.4
m=audio 33449 RTP/AVP 0
```

Juniper Networks security devices support up to 6 SDP channels negotiated for each direction, for a total of 12 channels per call. For more information, see “Session Description Protocol Sessions” on page 1109.

SIP NAT Scenario

In Figure 293 on page 1121, ph1 sends a SIP INVITE message to ph2. Note how the IP addresses in the header fields—shown in bold font—are translated by the security device.

The SDP section of the INVITE message indicates where the caller is willing to receive media. Note that the Media Pinhole contains two port numbers, 52002 and 52003, for RTCP and RTP. The Via/Contact Pinhole provides port number 5060 for SIP signaling.

Observe how, in the 200 OK response message, the translations performed in the INVITE message are reversed. The IP addresses in this message, being public, are not translated, but gates are opened to allow the media stream access to the private network.

Figure 293: SIP NAT Scenario 1

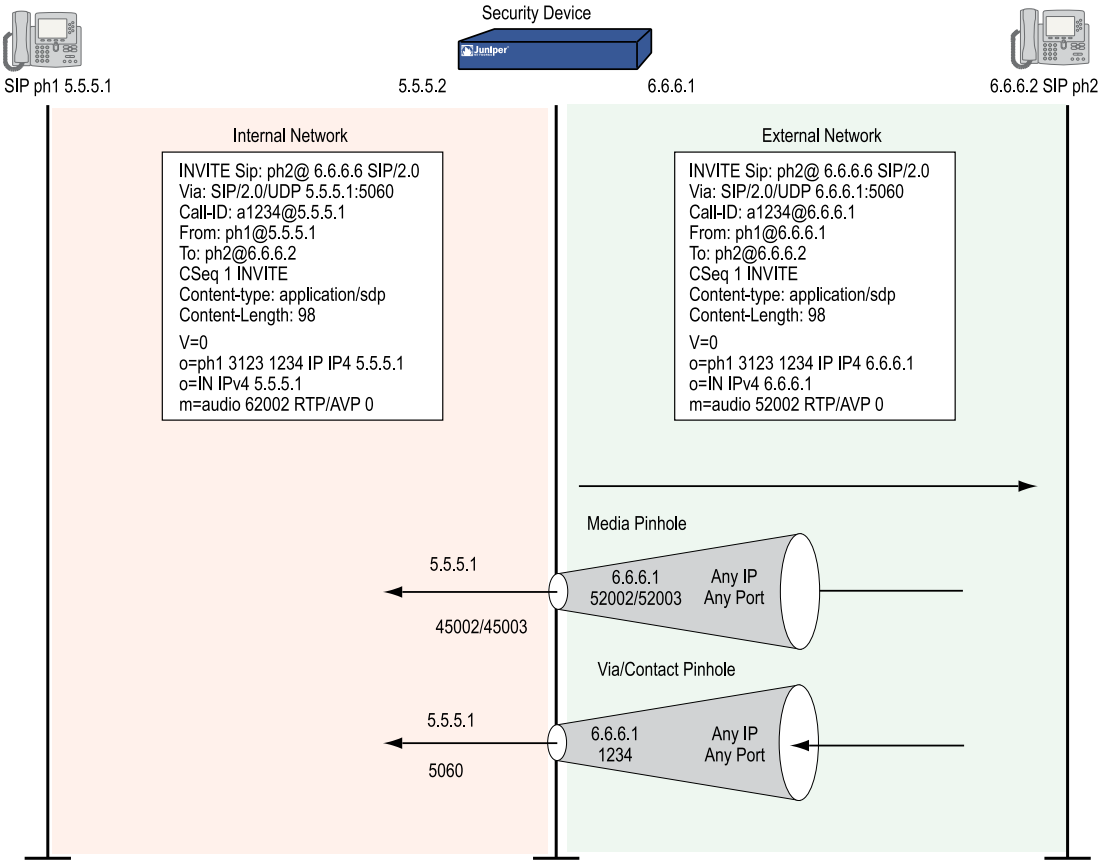
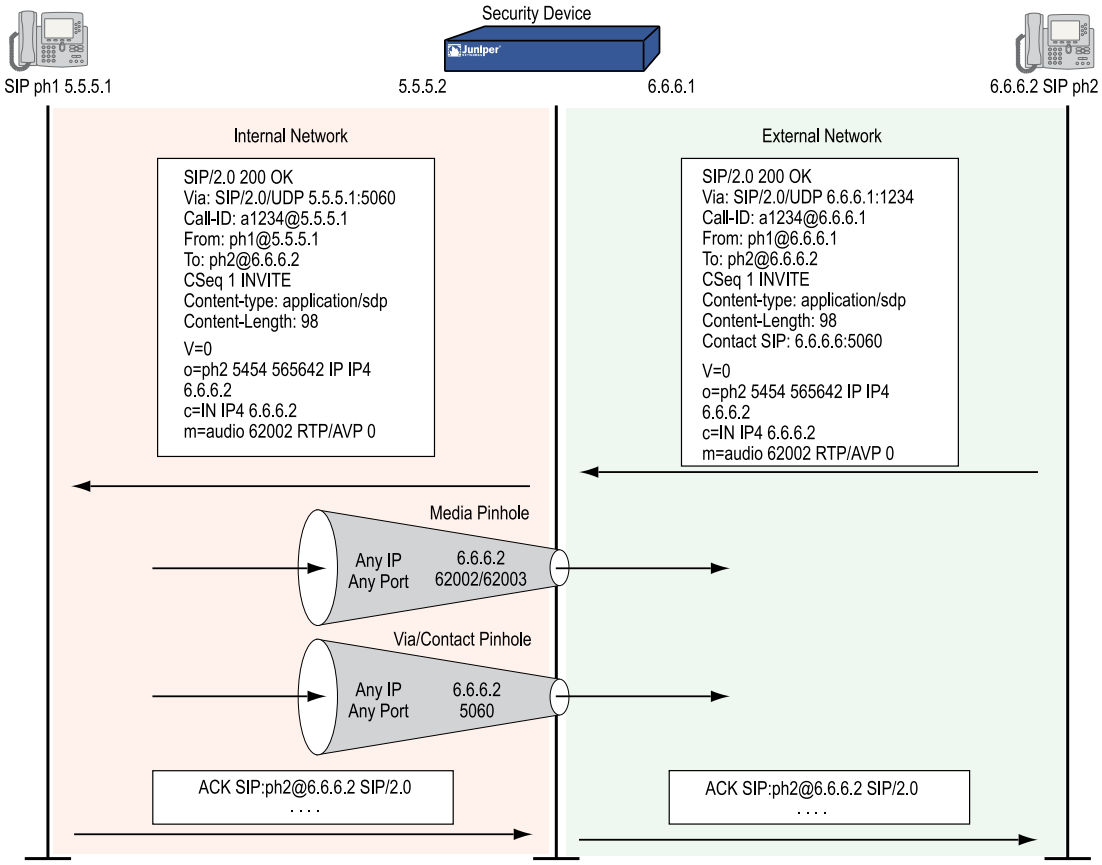


Figure 294: SIP NAT Scenario 2



Examples

This section contains the following sample scenarios:

- Incoming SIP Call Support Using the SIP Registrar on page 1122
- Example: Incoming Call with MIP on page 1128
- Example: Proxy in the Private Zone on page 1131
- Example: Proxy in the Public Zone on page 1133
- Example: Three-Zone, Proxy in the DMZ on page 1135
- Example: Untrust Intrazone on page 1139
- Example: Trust Intrazone on page 1143
- Example: Full-Mesh VPN for SIP on page 1146

Incoming SIP Call Support Using the SIP Registrar

SIP registration provides a discovery capability by which SIP proxies and location servers are able to identify the location or locations where users want to be contacted.

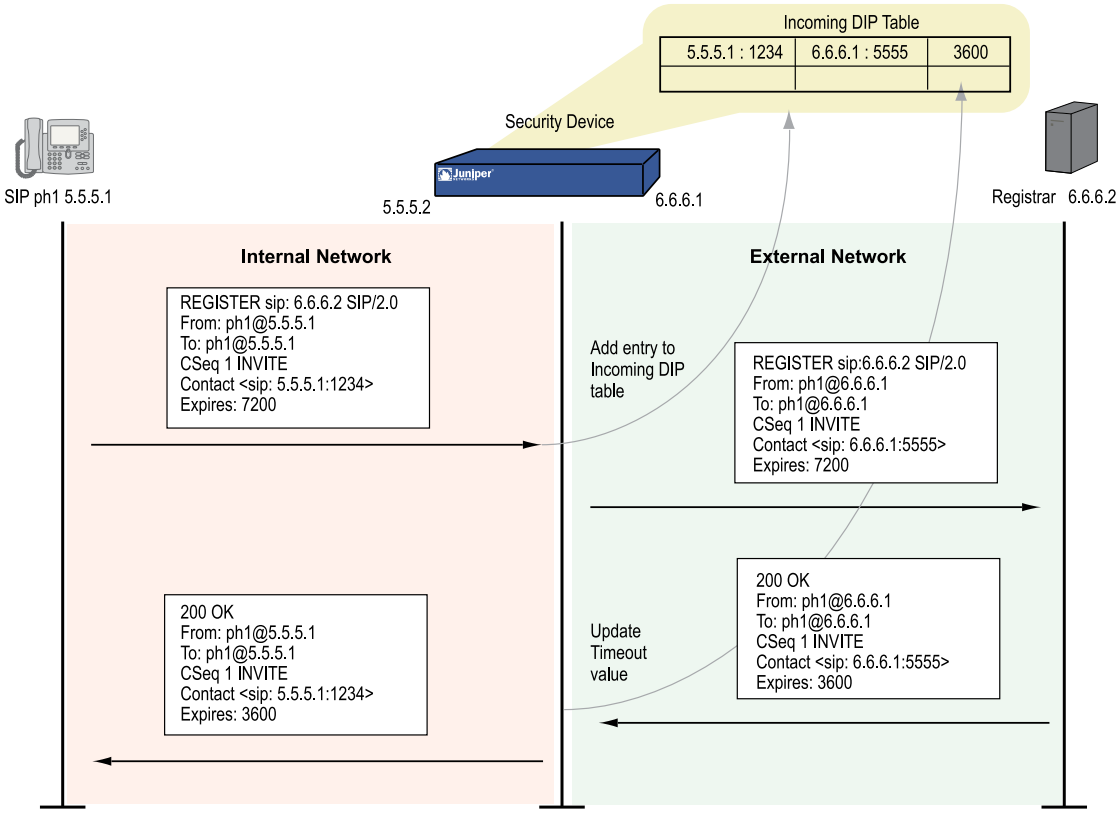
A user registers one or more contact locations by sending a REGISTER message to the registrar. The To: and Contact: fields in the REGISTER message contain the address-of-record URI and one or more contact URIs, as shown in Figure 295 on page 1123. Registration creates bindings in a location service that associates the address-of-record with the contact address or addresses.

The security device monitors outgoing REGISTER messages, performs NAT on these addresses, and stores the information in an Incoming DIP table. Then, when an INVITE message is received from outside the network, the security device uses the Incoming DIP table to identify which internal host to route the INVITE message to. You can take advantage of SIP proxy registration service to allow incoming calls by configuring Interface DIP or DIP pools on the egress interface of the security device. Interface DIP is adequate for handling incoming calls in a small office, while we recommend setting up DIP pools for larger networks or an enterprise environment.

NOTE: Incoming call support using Interface DIP or a DIP pool is supported for SIP and H.323 services only.

For incoming calls, security devices currently support UDP and TCP only. Domain name resolution is also currently not supported; therefore, URIs must contain IP addresses, as shown in Figure 295 on page 1123.

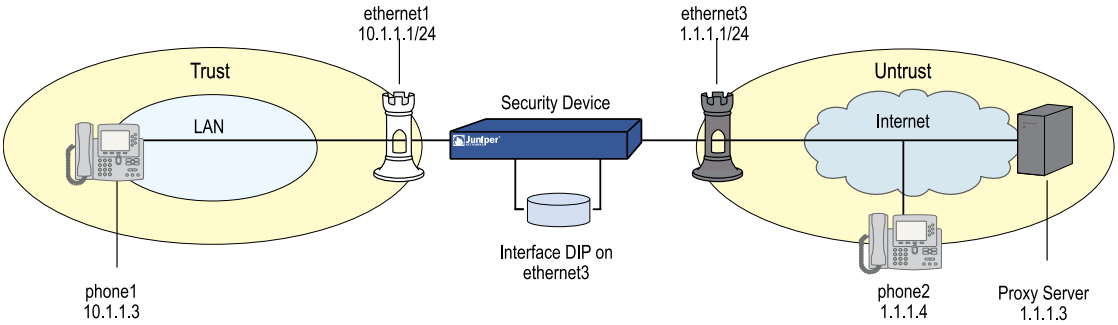
Figure 295: Incoming SIP



Example: Incoming Call (Interface DIP)

In this example, phone1 is on the ethernet1 interface in the Trust zone, and phone2 and the proxy server are on the ethernet3 interface in the Untrust zone. You set Interface DIP on the ethernet3 interface to do NAT on incoming calls, then create a policy permitting SIP traffic from the Untrust zone to the Trust zone and reference that DIP in the policy. You also create a policy that permits SIP traffic from the Trust to the Untrust zone using NAT Source. This enables phone1 in the Trust zone to register with the proxy in the Untrust zone. For an explanation of how incoming DIP works with the SIP registration service, see “Examples” on page 1122.

Figure 296: Incoming Call with Interface DIP on ethernet3 Interface



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24
   Interface Mode: Route

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: phone1
   IP Address/Domain Name:

IP/Netmask: (select), 10.1.1.3/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone2
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.4/24
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: proxy
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.3/24
Zone: Untrust

3. **DIP with Incoming NAT**

Network > Interface > Edit (for ethernet3) > DIP > New: Select the Incoming NAT option, then click **OK**.

4. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address
Address Book Entry: (select), phone1
Destination Address
Address Book Entry: (select), any
Service: SIP
Action: Permit

> Advanced: Enter the following, then click Return to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): None (Use Egress Interface IP)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address
Address Book Entry: (select), Any
Destination Address
Address Book Entry: (select), DIP(ethernet3)
Service: SIP
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 route
```

2. **Addresses**

```
set address trust phone1 10.1.1.3/24
set address untrust phone2 1.1.1.4/24
set address untrust proxy 1.1.1.3/24
```

3. **DIP with Incoming NAT**

```
set interface ethernet3 dip interface-ip incoming
set dip sticky
```

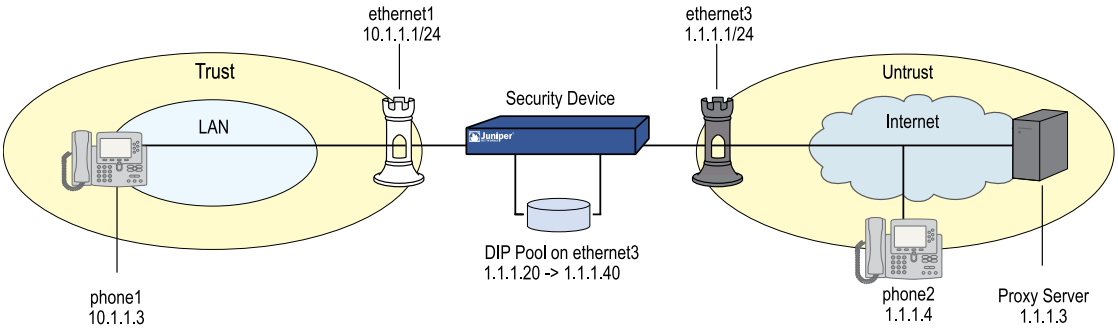
4. **Policies**

```
set policy from trust to untrust phone1 any sip nat src permit
set policy from untrust to trust any dip(ethernet3) sip permit
save
```

Example: Incoming Call (DIP Pool)

This example, phone1 is in the Trust zone, and phone2 and the proxy server are in the Untrust zone. You set a DIP pool on the ethernet3 interface to do NAT on incoming calls, then set a policy permitting SIP traffic from the Untrust zone to the Trust zone and reference that DIP pool in the policy. You also create a policy that permits SIP traffic from the Trust to the Untrust zone using NAT Source. This enables phone1 in the Trust zone to register with the proxy in the Untrust zone. For an explanation of how DIP works with the SIP registration service, see “Examples” on page 1122.

Figure 297: Incoming Call with DIP Pool



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24
Enter the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24
Interface Mode: Route

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone1
IP Address/Domain Name:
  IP/Netmask: (select), 10.1.1.3/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone2
IP Address/Domain Name:
  IP/Netmask: (select), 1.1.1.4/24
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: proxy
IP Address/Domain Name:
  IP/Netmask: (select), 1.1.1.3/24
Zone: Untrust

3. **DIP Pool with Incoming NAT**

Network > Interface > Edit (for ethernet3) > DIP > New: Enter the following, then click **OK**:

ID: 5
IP Address Range: (select), 1.1.1.20 ~ 1.1.1.40
  Port Translation: (select)
In the same subnet as the interface IP or its secondary IPs: (select)
  Incoming NAT: (select)

4. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address
  Address Book Entry: (select), phone1

Destination Address
Address Book Entry: (select), Any
Service: SIP
Action: Permit

> Advanced: Enter the following, then click Return to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): 5 (1.1.1.20-1.1.1.40)/port-xlate

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address
Address Book Entry: (select), Any
Destination Address
Address Book Entry: (select), DIP(5)
Service: SIP
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 route
```

2. **Addresses**

```
set address trust phone1 10.1.1.3/24
set address untrust phone2 1.1.1.4/24
set address untrust proxy 1.1.1.3/24
```

3. **DIP Pool with Incoming NAT**

```
set interface ethernet3 dip 5 1.1.1.20 1.1.1.40 incoming
set dip sticky
```

4. **Policies**

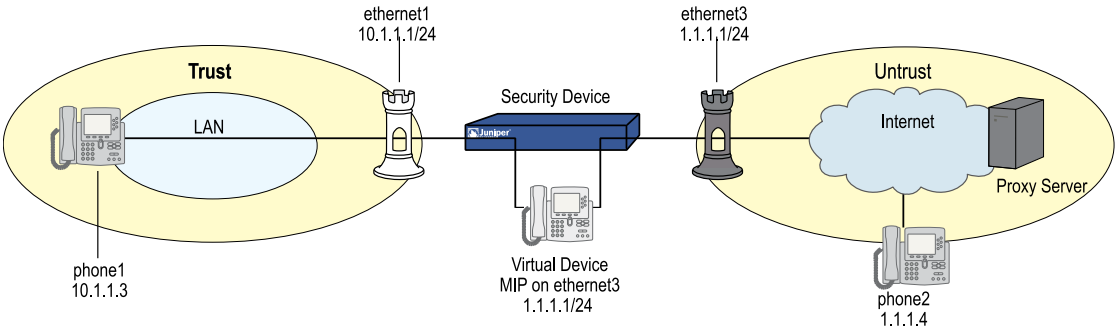
```
set policy from trust to untrust phone1 any sip nat src dip 5 permit
set policy from untrust to trust any dip(5) sip permit
save
```

Example: Incoming Call with MIP

In this example, phone1 is on the ethernet1 interface in the Trust zone, and phone2 and the proxy server are on the ethernet3 interface in the Untrust zone. You put a MIP on the ethernet3 interface to phone1, then create a policy that allows SIP traffic

from the Untrust zone to the Trust zone and reference that MIP in the policy. You also create a policy allowing phone1 to register with the proxy server in the Untrust zone. This example is similar to the previous two examples (“Example: Incoming Call (Interface DIP)” on page 1124 and “Example: Incoming Call (DIP Pool)” on page 1126), except that with a MIP you need one public address for each private address in the Trust zone, while with Interface DIP or a DIP pool a single interface address can serve multiple private addresses.

Figure 298: Incoming Call with MIP



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK:**
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone: Untrust
   IP Address/Netmask: 1.1.1.1/24
   Interface Mode: Route

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: phone1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.3/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone2
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.4/24
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: proxy
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.3/24
Zone: Untrust

3. **MIP**

Network > Interfaces > Edit (for ethernet3) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 1.1.1.3
Netmask: 255.255.255.255
Host IP Address: 10.1.1.3

4. **Policy**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), any
Destination Address:
Address Book Entry: (select), MIP(1.1.1.3)
Service: SIP
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 route
```

2. **Addresses**

```
set address trust phone1 10.1.1.3/24
set address untrust phone2 1.1.1.4/24
set address untrust proxy 1.1.1.3/24
```

3. **MIP**

```
set interface ethernet3 mip 1.1.1.3 host 10.1.1.3
```

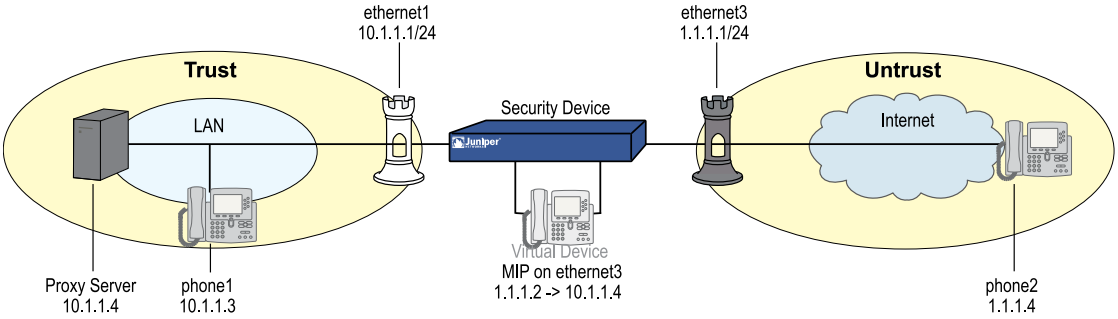
4. **Policy**

```
set policy from untrust to trust any mip(1.1.1.3) sip permit
save
```

Example: Proxy in the Private Zone

In this example, phone1 and the SIP proxy server are on the ethernet1 interface in the Trust (private) zone, and phone2 is on the ethernet3 interface in the Untrust zone. You put a MIP on the ethernet3 interface to the proxy server to allow phone2 to register with the proxy, then create a policy allowing SIP traffic from the Untrust to the Trust zone and reference that MIP in the policy. You also create a policy from the Trust to the Untrust zone to allow phone1 to call out.

Figure 299: Proxy in the Private Zone



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

   Zone: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click OK:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone: Untrust
   IP Address/Netmask: 1.1.1.1/24
   Interface Mode: Route

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: phone1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.3/24
   Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone2
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.4/24
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: proxy
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.4/24
Zone: Trust

3. **MIP**

Network > Interfaces > Edit (for loopback.3) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 1.1.1.2
Netmask: 255.255.255.255
Host IP Address: 10.1.1.4
Host Virtual Router Name: trust-vr

4. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select) any
Destination Address:
Address Book Entry: (select) phone2
Service: SIP
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): None (Use Egress Interface IP)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select), phone2
Destination Address:
Address Book Entry: (select), MIP(1.1.1.2)
Service: SIP
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 route
```

2. **Addresses**

```
set address trust phone1 10.1.1.3/24
set address untrust phone2 1.1.1.4/24
set address trust proxy 10.1.1.4/24
```

3. **MIP**

```
set interface ethernet3 mip 1.1.1.2 host 10.1.1.4
```

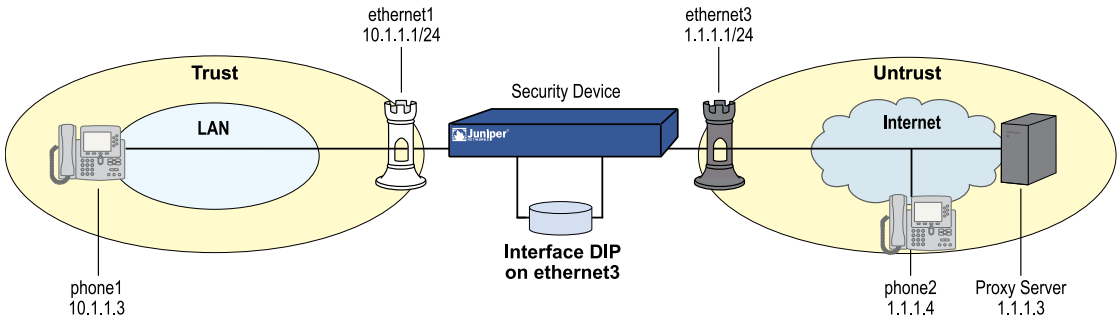
4. **Policies**

```
set policy from trust to untrust any phone2 sip nat src permit
set policy from untrust to trust phone2 mip(1.1.1.2) sip permit
save
```

Example: Proxy in the Public Zone

In this example, phone1 is on the ethernet1 interface in the Trust zone, and the proxy server and phone2 are on the ethernet3 interface in the Untrust (public) zone. You configure Interface DIP on the Untrust interface, then create a policy permitting SIP traffic from the Untrust zone to the Trust zone and reference that DIP in the policy. You also create a policy from Trust to Untrust to allow phone1 to register with the proxy server in the Untrust zone. This example is similar to the previous incoming call examples (see "Example: Incoming Call (DIP Pool)" on page 1126 and "Example: Incoming Call with MIP" on page 1128) and, as with those examples, you can use DIP or MIP on the Untrust interface.

Figure 300: Proxy in the Public Zone



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK:**
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone: Untrust
   IP Address/Netmask: 1.1.1.1/24
   Interface Mode: Route

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: phone1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.3/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: phone2
   IP Address/Domain Name:
   IP/Netmask: (select), 1.1.1.4/24
   Zone: Untrust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: proxy
   IP Address/Domain Name:
   IP/Netmask: (select), 1.1.1.3/24
   Zone: Untrust

3. **Interface DIP**

   Network > Interface > Edit (for ethernet3) > DIP: Select the Incoming NAT check box.

4. **Policies**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select) phone1
Destination Address:
Address Book Entry: (select) Any
Service: SIP
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): None (Use Egress Interface IP)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), DIP(ethernet3)
Service: SIP
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address trust phone1 10.1.1.3/24
set address untrust phone2 1.1.1.4/24
set address untrust proxy 1.1.1.3/24
```

3. **Interface DIP**

```
set interface ethernet3 dip interface-ip incoming
```

4. **Policies**

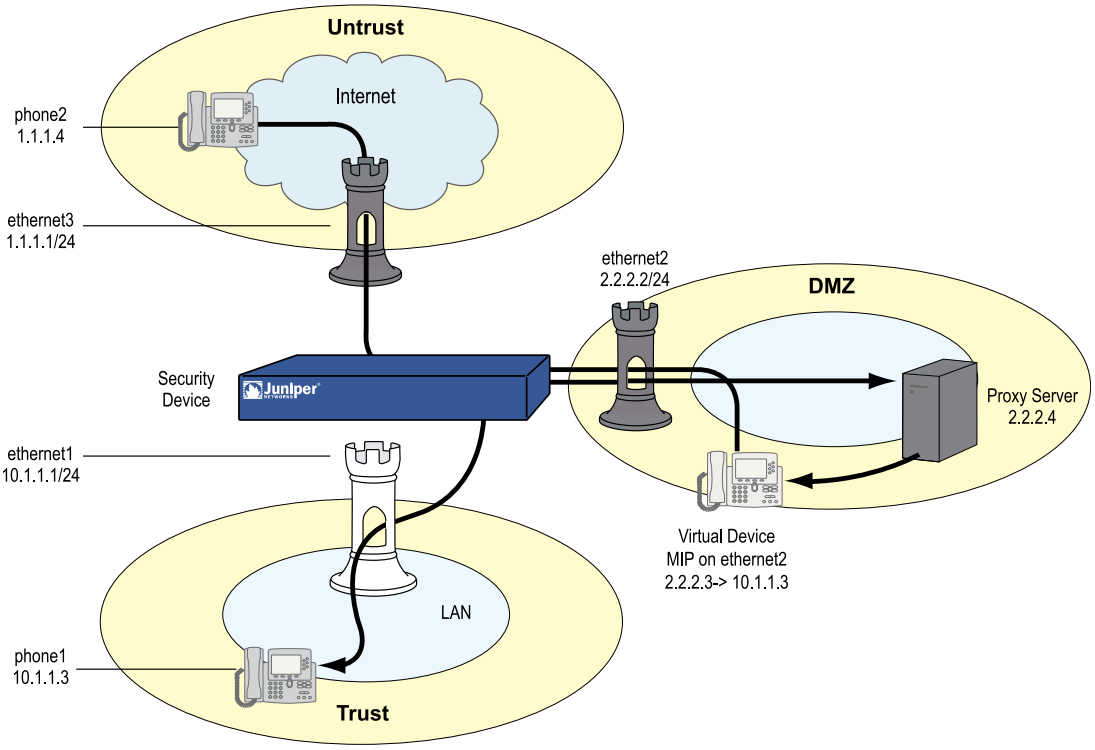
```
set policy from trust to untrust phone1 any sip nat src permit
set policy from untrust to trust any dip(ethernet3) sip permit
save
```

Example: Three-Zone, Proxy in the DMZ

In this example, phone1 is on the ethernet1 interface in the Trust zone, phone2 is on the ethernet3 interface in the Untrust zone, and the proxy server is on the ethernet2 interface in the DMZ. You put a MIP on the ethernet2 interface to phone1

in the Trust zone, and create a policy from the DMZ to the Trust zone and reference that MIP in the policy. In fact, with three zones, you need to create bidirectional policies between each of the zones. The arrows in Figure 301 on page 1136 show the flow of SIP signaling traffic when phone2 in the Untrust zone places a call to phone1 in the Trust zone. After the session is initiated, the media flows directly between phone1 and phone2.

Figure 301: Proxy in the DMZ



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone: Trust
   Static IP: (select when this option is present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK:**
   Interface Mode: NAT
   Zone Name: DMZ
   Static IP: (select when this option is present)
   IP Address/Netmask: 2.2.2.2/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 1.1.1.1/24

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone1
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.3/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone2
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.4/24
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: proxy
IP Address/Domain Name:
IP/Netmask: (select), 2.2.2.4/24
Zone: DMZ

3. **MIP**

Network > Interfaces > Edit (for ethernet2) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 2.2.2.3
Netmask: 255.255.255.255
Host IP Address: 10.1.1.3

4. **Policies**

Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select), phone1
Destination Address:
Address Book Entry: (select), proxy
Service: SIP
Action: Permit

> Advanced: Enter the following, then click Return to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: Enable
(DIP on): None (Use Egress Interface IP)

Policies > (From: DMZ, To: Untrust) New: Enter the following, then click **OK:**

Source Address:
  Address Book Entry: (select), proxy
Destination Address:
  Address Book Entry: (select), phone2
Service: SIP
Action: Permit

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK:**

Source Address:
  Address Book Entry: (select), phone2
Destination Address:
  Address Book Entry: (select), phone1
Service: SIP
Action: Permit

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK:**

Source Address:
  Address Book Entry: (select), phone2
Destination Address:
  Address Book Entry: (select), proxy
Service: SIP
Action: Permit

Policies > (From: DMZ, To: Trust) New: Enter the following, then click **OK:**

Source Address:
  Address Book Entry: (select), proxy
Destination Address:
  Address Book Entry: (select), MIP(2.2.2.3)
Service: SIP
Action: Permit

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address:
  Address Book Entry: (select), phone1
Destination Address:
  Address Book Entry: (select), phone2
Service: SIP
Action: Permit

> Advanced: Enter the following, then click Return to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: Enable
  (DIP on): None (Use Egress Interface IP)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 route
set interface ethernet2 zone dmz
set interface ethernet2 ip 2.2.2.2/24
set interface ethernet2 route
```

2. **Addresses**

```
set address trust phone1 10.1.1.3/24
set address untrust phone2 1.1.1.4/24
set address dmz proxy 2.2.2.4
```

3. **MIP**

```
set interface2 mip 2.2.2.3 host 10.1.1.3
```

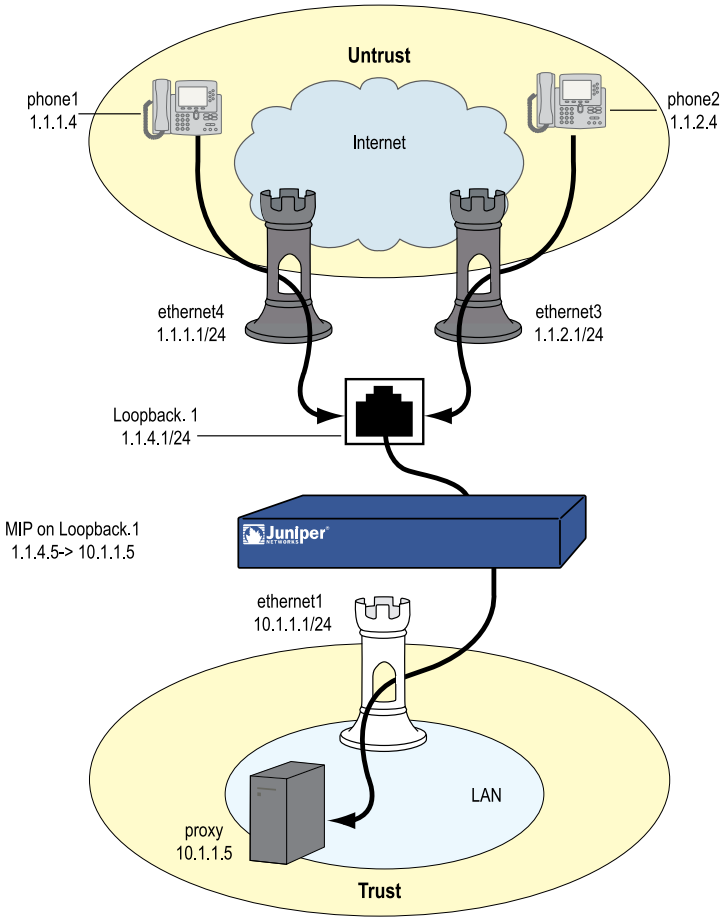
4. **Policies**

```
set policy from trust to dmz phone1 proxy sip nat src permit
set policy from dmz to untrust proxy phone2 sip permit
set policy from untrust to trust phone2 phone1 sip permit
set policy from untrust to dmz phone2 proxy sip permit
set policy from dmz to trust proxy mip(2.2.2.3) sip permit
set policy from trust to untrust phone1 phone2 sip nat src permit
save
```

Example: Untrust Intrazone

In this example, phone1 is on the ethernet4 interface in the Untrust zone, phone2 is in a subnet on the ethernet3 interface in the Untrust zone, and the proxy server is on the ethernet1 interface in the Trust zone. To allow intrazone SIP traffic between the two phones in the Untrust zone, you create a loopback interface, add ethernet3 and ethernet4 to a loopback group, then put a MIP on the loopback interface to the IP address of the proxy server. Creating a loopback interface enables you to use a single MIP for the proxy server in the Trust zone. Because blocking is on by default in the Untrust zone, you must also turn off blocking to allow intrazone communication. For more information about using loopback interfaces, see “MIP and the Loopback Interface” on page 1545.

Figure 302: Untrust Intrazone



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone: Trust
   Static IP: (select when this option is present)
   IP Address/Netmask: 10.1.1.1/24

   Enter the following, then click **OK:**

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet4): Enter the following, then click **OK**:

   Zone: Untrust
   Static IP: (select when this option is present)
   IP Address/Netmask: 1.1.1.1/24

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 1.1.2.1/24

Network > Interfaces > New Loopback IF: Enter the following, then click **OK**:

Interface Name: loopback.1
Zone: Untrust (trust-vr)
IP Address/Netmask: 1.1.4.1/24

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: proxy
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.5/32
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone1
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.4/32
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone2
IP Address/Domain Name:
IP/Netmask: (select), 1.1.2.4/32
Zone: Untrust

3. **Loopback Group**

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

As member of loopback group: (select) loopback.1
Zone Name: Untrust

Network > Interfaces > Edit (for ethernet4): Enter the following, then click **OK**:

As member of loopback group: (select) loopback.1
Zone Name: Untrust

4. **MIP**

Network > Interfaces > Edit (for loopback.1) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 1.1.4.5
Netmask: 255.255.255.255
Host IP Address: 10.1.1.5
Host Virtual Router Name: trust-vr

5. **Blocking**

   Network > Zones > Edit (for Untrust): Enter the following, then click **OK**:

   Block Intra-Zone Traffic: (clear)

6. **Policies**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

   Source Address:
   Address Book Entry: (select), proxy
   Destination Address:
   Address Book Entry: (select), Any
   Service: SIP
   Action: Permit

   > Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   NAT:
   Source Translation: Enable
   (DIP on): None (Use Egress Interface IP)

   Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), MIP(1.1.4.5)
   Service: SIP
   Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.2.1/24
set interface ethernet3 route
set interface ethernet4 zone untrust
set interface ethernet4 ip 1.1.1.1/24
set interface ethernet4 route
set interface loopback.1 zone untrust
set interface loopback.1 ip 1.1.4.1/24
set interface loopback.1 route
```

2. **Addresses**

set address trust proxy 10.1.1.5/32
set address untrust phone1 1.1.1.4/32
set address untrust phone2 1.1.2.4/32

3. **Loopback Group**

   set interface ethernet3 loopback-group loopback.1
   set interface ethernet4 loopback-group loopback.1

4. **MIP**

   set interface loopback.1 mip 1.1.4.5 host 10.1.1.5

5. **Blocking**

   unset zone untrust block

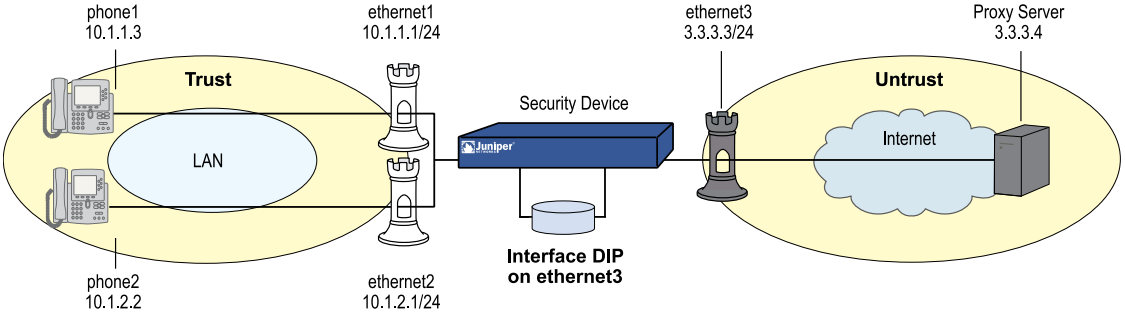
6. **Policies**

   set policy from trust to untrust proxy any sip nat src permit
   set policy from untrust to trust any mip(1.1.4.5) sip permit
   save

Example: Trust Intrazone

In this example, phone1 is on the ethernet1 interface in the Trust zone, phone 2 is on the ethernet2 interface in a subnet in the Trust zone, and the proxy server is on the ethernet3 interface in the Untrust zone. To allow both phones in the Trust zone to communicate with each other, you configure Interface DIP on the ethernet3 interface to allow them to contact the proxy server, then set policies to allow bidirectional SIP traffic between the Trust and the Untrust zones. Blocking is off by default in the Trust zone (as it is in custom zones you define).

Figure 303: Trust Intrazone



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone: Trust
Static IP: (select when this option is present)
IP Address/Netmask: 10.1.1.1/24
Enter the following, then click **OK:**
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **Apply**:

Zone: Trust
Static IP: (select when this option is present)
IP Address/Netmask: 10.1.2.1/24
Enter the following, then click **OK:**
Interface Mode: NAT
Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 3.3.3.3/24

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone1
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.3/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone2
IP Address/Domain Name:
IP/Netmask: (select), 10.1.2.2/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: proxy
IP Address/Domain Name:
IP/Netmask: (select), 3.3.3.4/24
Zone: Untrust

3. **DIP with Incoming NAT**

Network > Interface > Edit (for ethernet3) > DIP > New: Enter the following, then click **OK**:

Incoming NAT: (select)

4. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), proxy
Service: SIP
Action: Permit

> Advanced: Enter the following, then click Return to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: Enable
(DIP on): None (Use Egress Interface IP)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK:**

Source Address
Address Book Entry: (select) proxy
Destination Address
Address Book Entry: (select) Any
Service: SIP
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options:

NAT:
Source Translation: (select)
(DIP on): None (Use Egress Interface IP)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet2 zone trust
set interface ethernet2 ip 10.1.2.1/24
set interface ethernet2 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 3.3.3.3/24
set interface ethernet3 route
```

2. **Addresses**

```
set address trust phone1 10.1.1.3/24
set address trust phone2 10.1.2.2/24
set address untrust proxy 3.3.3.4/24
```

3. **Interface DIP**

```
set interface ethernet3 dip interface-ip incoming
```

4. **Policies**

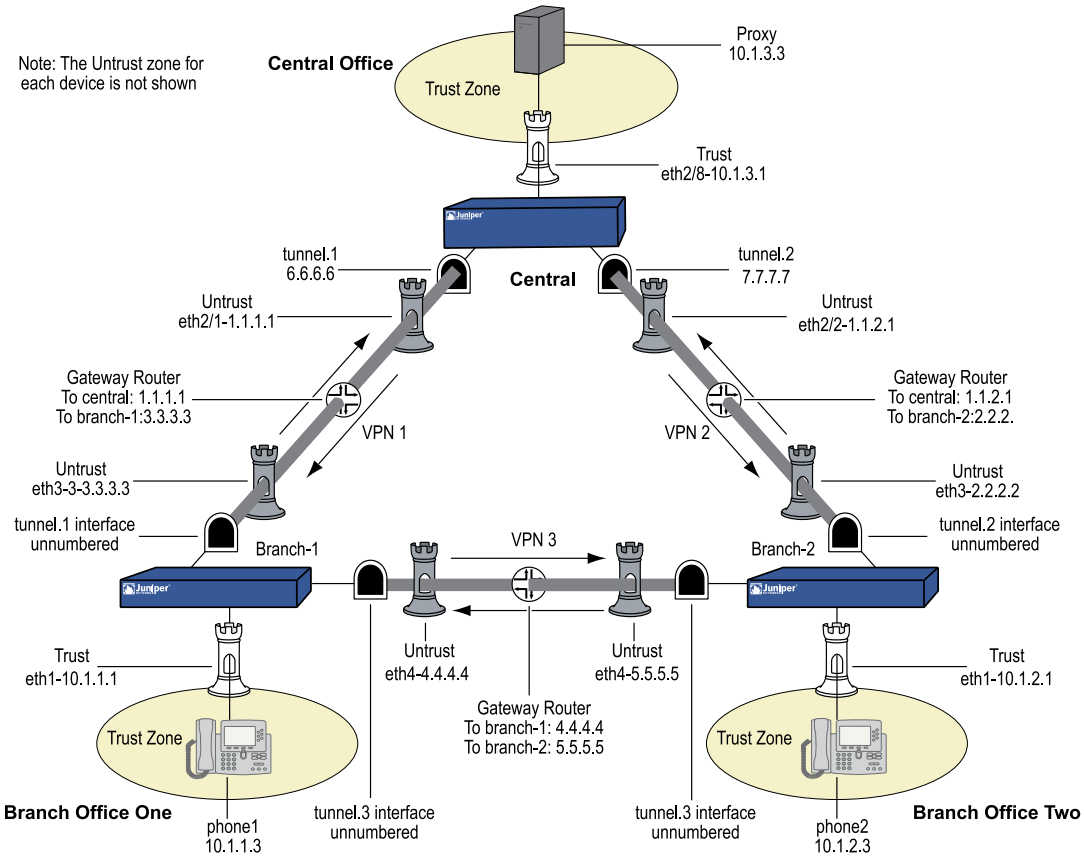
```
set policy from trust to untrust any proxy sip nat src permit
set policy from untrust to trust proxy dip(ethernet3) sip permit
save
```

Example: Full-Mesh VPN for SIP

In this example, the central office and two branch offices are linked by a full-mesh VPN. Each site has a single security device. The proxy server is in the Trust zone at the Central Office, phone1 is in the Trust zone at Branch Office One, and phone2 is in the Trust zone at Branch Office Two. All interfaces connecting the devices are in their respective Untrust zones. On each device, you configure two tunnels, one to each of the other devices, to create a fully meshed network.

NOTE: The security devices used in this example must have at least three independently configurable interfaces available.

Figure 304: Full-Mesh VPN for SIP



WebUI (for Central)

1. Interfaces

Network > Interfaces > Edit (for ethernet2/1): Enter the following, then click **Apply**:

Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 1.1.1.1/24

Network > Interfaces > Edit (for ethernet2/2): Enter the following, then click **Apply**:

Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 1.1.2.1/24

Network > Interfaces > Edit (for ethernet2/8): Enter the following, then click **Apply**:

Zone: Trust
Static IP: (select when this option is present)
IP Address/Netmask: 10.1.3.1/24
Enter the following, then click OK:
Interface mode: route

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

Tunnel Interface Name: 1
Zone (VR): Untrust
IP Address / Netmask: 6.6.6.6/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

Tunnel Interface Name: 2
Zone (VR): Untrust
IP Address / Netmask: 7.7.7.7/24

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Proxy
IPv4/Netmask: 10.1.3.3/32
Zone: Trust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: to-branch-1
Security Level: Standard
IPvc4/v6 Address/Hostname: 3.3.3.3
Preshare Key: netscreen
Outgoing Interface: ethernet2/1

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn-branch-1

Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Bind to: (select) Tunnel Interface, tunnel.1

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: to-branch-2
Security Level: Standard
IPvc4/v6 Address/Hostname: 2.2.2.2
Preshare Key: netscreen
Outgoing Interface: ethernet2/2

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn-branch-2

Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Bind to: (select) Tunnel Interface, tunnel.2

4. **Routing**

Network > Routing > Destination > New: Enter the following, then click **OK:**

Network Address / Netmask: 10.1.1.0/24
Interface (select): tunnel.1

Network > Routing > Destination > New: Enter the following, then click **OK:**

Network Address / Netmask: 10.1.2.0/24
Interface (select): tunnel.2

5. **Policies**

Policies > (From: Trust, To: Untrust) New Enter the following, then click **OK:**

Source Address (select) Address Book Entry: Proxy
Destination Address (select) Address Book Entry: Any-IPv4
Service: SIP
Action: Permit

Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK:**

Source Address (select) Address Book Entry: Any-IPv4
Destination Address (select) Address Book Entry: Proxy
Service: SIP
Action: Permit

CLI (for Central)

1. **Interfaces**

```
set interface ethernet2/1 zone untrust
set interface ethernet2/1 ip 1.1.1.1/24
set interface ethernet2/2 zone untrust
set interface ethernet2/2 ip 1.1.2.1/24
set interface ethernet2/8 zone trust
set interface ethernet2/8 ip 10.1.3.1/24
set interface ethernet2/8 route
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 6.6.6.6/24
set interface tunnel.2 zone untrust
set interface tunnel.2 ip 7.7.7.7/24
```

2. **Address**

```
set address trust proxy 10.1.3.3/32
```

3. **VPN**

```
set ike gateway to-branch-1 address 3.3.3.3 main outgoing-interface ethernet2/1
 preshare netscreen sec-level standard
set ike gateway to-branch-2 address 2.2.2.2 main outgoing-interface ethernet2/2
 preshare netscreen sec-level standard
set vpn vpn_branch-1 gateway to-branch-1 no-reply tunnel idletime 0 sec-level
standard
set vpn vpn-branch-1 id 1 bind interface tunnel.1
set vpn vpn-branch-2 gateway to-branch-2 no-reply tunnel idletime 0 sec-level
standard
set vpn vpn-branch-2 id 2 bind interface tunnel.2
```

4. **Routing**

```
set route 10.1.2.0/24 interface tunnel.2
set route 10.1.1.0/24 interface tunnel.1
```

5. **Policies**

```
set policy from untrust to trust any proxy sip permit
set policy from trust to untrust proxy any sip permit
save
```

WebUI (for Branch Office 1)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

```
Zone: Trust
Static IP: (select when this option is present)
```

IP Address/Netmask: 10.1.1.1/24
Interface mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **Apply**:

Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 3.3.3.3/24

Network > Interfaces > Edit (for ethernet4): Enter the following, then click **Apply**:

Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 4.4.4.4/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

Tunnel Interface Name: 2
Zone (VR): Untrust
Unnumbered (select) Interface: ethernet3

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

Tunnel Interface Name: 3
Zone (VR): Untrust
Unnumbered (select) Interface: ethernet4

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone1
IPv4/Netmask: 10.1.1.3/32
Zone: V1-Trust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: to-central
Security Level: Standard
IPvc4/v6 Address/Hostname: 1.1.2.1
Preshare Key: netscreen
Outgoing Interface: ethernet3

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn-central

Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Bind to (select): Tunnel Interface, tunnel.1

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: to-ns50
Security Level: Standard
IPvc4/v6 Address/Hostname: 5.5.5.5
Preshare Key: netscreen
Outgoing Interface: ethernet4

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn-ns50

Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Bind to (select): Tunnel Interface, tunnel.3

4. **Routing**

Network > Routing > Destination > New: Enter the following, then click **OK:**

Network Address / Netmask: 10.1.2.0/24
Interface (select): tunnel.3

Network > Routing > Destination > New: Enter the following, then click **OK:**

Network Address / Netmask: 10.1.3.0/24
Interface (select): tunnel.1

5. **Policies**

Policies > (From: Trust, To: Untrust) > New: Enter the following, then click **OK:**

Source Address (select) Address Book Entry: phone2
Destination Address (select) Address Book Entry: Any-IPv4
Service: SIP
Action: Permit

Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK:**

Source Address (select) Address Book Entry: Any-IPv4
Destination Address (select) Address Book Entry: phone2
Service: SIP
Action: Permit

CLI (for Branch Office 1)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 route
```

```
set interface ethernet3 zone untrust
set interface ethernet3 ip 3.3.3.3/24
set interface ethernet4 zone untrust
set interface ethernet4 ip 4.4.4.4/24
set interface tunnel.2 zone untrust
set interface tunnel.2 ip unnumbered interface ethernet3
set interface tunnel.3 zone untrust
set interface tunnel.3 ip unnumbered interface ethernet4
```

2. **Address**

```
set address trust phone1 10.1.1.3/32
```

3. **VPN**

```
set ike gateway to-central address 1.1.1.1 main outgoing-interface ethernet3
preshare netscreen sec-level standard
set ike gateway to-ns50 address 5.5.5.5 main outgoing-interface ethernet4
preshare netscreen sec-level standard
set vpn vpncentral gateway to-central no-replay tunnel idletime 0 sec-level
standard
set vpn vpncentral bind interface tunnel.1
set vpn vpn-ns50 gateway to-ns50 no-replay tunnel idletime 0 sec-level standard
set vpn vpn-ns50 bind interface tunnel.3
```

4. **Routes**

```
set route 10.1.2.0/24 interface tunnel.3
set route 10.1.3.0/24 interface tunnel.1
```

5. **Policies**

```
set policy from trust to untrust phone1 any sip permit
set policy from untrust to trust any phone1 sip permit
save
```

WebUI (for Branch Office 2)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone: Trust
Static IP: (select when this option is present)
IP Address/Netmask: 10.1.2.1/24
Enter the following, then click **OK:**
Interface mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **Apply**:

Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 2.2.2.2/24

Network > Interfaces > Edit (for ethernet4): Enter the following, then click **Apply**:

Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 4.4.4.4/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

Tunnel Interface Name: 2
Zone (VR): Untrust
Unnumbered (select) Interface: ethernet3

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

Tunnel Interface Name: 3
Zone (VR): Untrust

Unnumbered (select) Interface: ethernet4

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK:**

Address Name: phone2
IPv4/Netmask: 10.1.2.3/32
Zone: Trust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: to-central
Security Level: Standard
IPvc4/v6 Address/Hostname: 1.1.2.1
Preshare Key: netscreen
Outgoing Interface: ethernet3

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn-central

Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Bind to (select): Tunnel Interface, tunnel.2

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: to-ns50
Security Level: Standard
IPvc4/v6 Address/Hostname: 4.4.4.4

Preshare Key: netscreen
Outgoing Interface: ethernet4

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn-ns50

Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Bind to (select): Tunnel Interface, tunnel.3

4. **Routing**

Network > Routing > Destination > New: Enter the following, then click **OK:**

Network Address / Netmask: 10.1.3.0/24
Interface (select): tunnel.2

Network > Routing > Destination > New: Enter the following, then click **OK:**

Network Address / Netmask: 10.1.1.0/24
Interface (select): tunnel.3

5. **Policies**

Policies > (From: Trust, To: Untrust) New Enter the following, then click **OK:**

Source Address (select) Address Book Entry: phone2
Destination Address (select) Address Book Entry: Any-IPv4
Service: SIP
Action: Permit

Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK:**

Source Address (select) Address Book Entry: Any-IPv4
Destination Address (select) Address Book Entry: phone2
Service: SIP
Action: Permit

CLI (for Branch Office 2)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.2.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface ethernet4 zone untrust
set interface ethernet4 ip 4.4.4.4/24
set interface tunnel.2 zone untrust
set interface tunnel.2 ip unnumbered interface ethernet3
set interface tunnel.3 zone untrust
set interface tunnel.3 ip unnumbered interface ethernet4
```

2. **Address**

   ```
   set address trust phone2 10.1.2.3/32
   ```

3. **VPN**

   ```
   set ike gateway to-central address 1.1.2.1 Main outgoing-interface ethernet3
   preshare netscreen sec-level standard
   set ike gateway to-ns50 address 4.4.4.4Main outgoing-interface ethernet4
   preshare netscreen sec-level standard
   set vpn vpncentral gateway to-central no-replay tunnel idletime 0 sec-level
   standard
   set vpn vpncentral id 4 bind interface tunnel.2
   set vpn vpn-ns50 gateway to-ns50 no-replay tunnel idletime 0 sec-level standard
   set vpn vpn-ns50 id 5 bind interface tunnel.3
   ```

4. **Routes**

   ```
   set route 10.1.3.0/24 interface tunnel.2
   set route 10.1.1.0/24 interface tunnel.3
   ```

5. **Policies**

   ```
   set policy from trust to untrust phone2 any sip permit
   set policy from untrust to trust any phone2 sip permit
   save
   ```

Bandwidth Management for VoIP Services

We recommend the following ways to manage bandwidth for VoIP services, using the standard ScreenOS traffic shaping mechanisms:

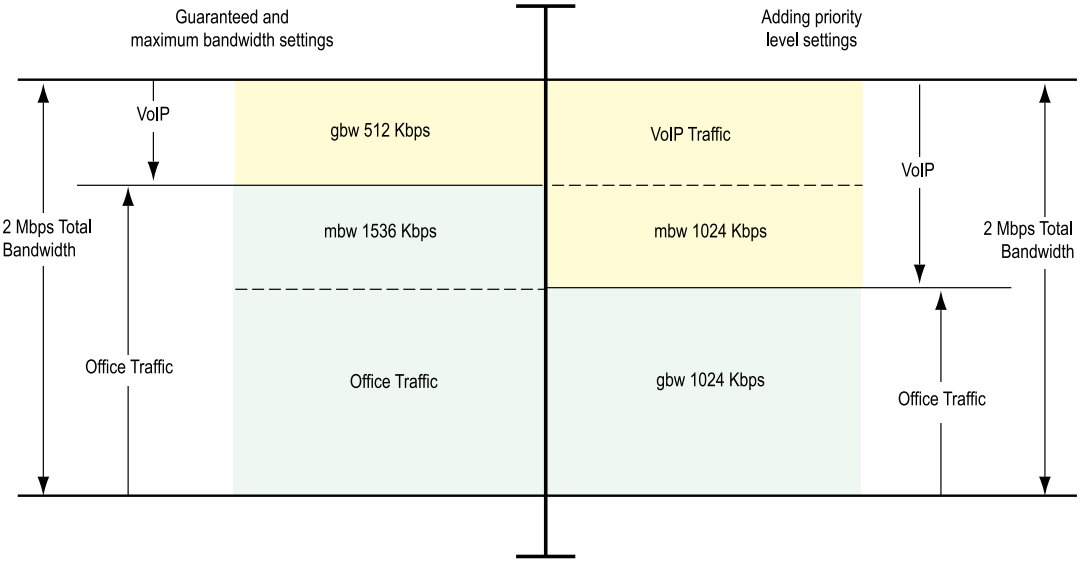
- Guarantee bandwidth for VoIP traffic—The most effective way to ensure quality VoIP service, and still allow other types of traffic on the interface, is to create a policy guaranteeing the minimum bandwidth necessary for the amount of VoIP traffic you expect on the interface and set priority queuing to the highest level. The advantage of this strategy is that VoIP traffic can use additional bandwidth when it is available, and other types of traffic can use bandwidth not guaranteed for VoIP when VoIP traffic is not using it.
- Limit bandwidth for non-VoIP traffic—By setting a maximum bandwidth for non-VoIP traffic, you make the remaining bandwidth available to VoIP traffic. You would also set priority queuing to the highest level for VoIP traffic. The disadvantage of this method is that non-VoIP traffic cannot use additional bandwidth even when VoIP traffic is not using it.
- Use priority queuing and Differentiated Services Codepoint (DSCP) marking—Guaranteeing bandwidth for VoIP traffic and limiting bandwidth for non-VoIP traffic both govern throughput on the security device. DSCP marking enables you to preserve your priority-queuing settings downstream and to keep or change the received DSCP value set by the originating networking device upstream so that the next-hop router, typically the LAN or WAN edge router, can enforce Quality of Service (QoS) in its DiffServ domain. In VPN configurations, the security device marks the outer header of the IP packet (if the policy is configured to do so), or leaves the TOS byte as 0 so that the next-hop router can

enforce the correct QoS on the encrypted traffic. For information about how DSCP works with priority levels in policies, see “Traffic Shaping” on page 212 .

Figure 305 on page 1156 shows how priority-level settings can affect guaranteed bandwidth (gbw) and maximum bandwidth (mbw) usage on an ethernet1 (2 Mbps) interface. The illustration assumes you have determined you need to support at least eight VoIP calls (8 x 64 Kbps bandwidth per call, for a total of 512 Kbps) and occasionally as many as 16 calls. You have guaranteed the remaining bandwidth to general office traffic and have set maximum bandwidth for your office traffic to include bandwidth not guaranteed to VoIP. This creates a 512 Kbps overlap of maximum bandwidth for VoIP and office-traffic services, shown by the dashed lines.

The left side of Figure 305 on page 1156 shows what bandwidth usage with these settings looks like with high office-traffic usage and low VoIP traffic usage on the interface. If VoIP traffic suddenly needs more bandwidth, it cannot get it unless it has a higher priority than the office-traffic services. The right side of Figure 305 on page 1156 shows what bandwidth usage looks like in the same circumstance when you give VoIP traffic a higher priority and set office traffic to a lower priority. For more information about configuring bandwidth and priority levels, see “Traffic Shaping” on page 233 .

Figure 305: Priority-Level Settings



Chapter 29

Media Gateway Control Protocol Application Layer Gateway

This chapter presents an overview of the Media Gateway Control Protocol (MGCP) Application Layer Gateway (ALG) and lists the firewall security features of the implementation. Examples of typical scenarios follow a summary of the MGCP architecture. This chapter includes the following sections:

Overview

The Media Gateway Control Protocol (MGCP) is supported on security devices in Route, transparent, and Network Address Translation (NAT) mode. MGCP is a text-based Application Layer protocol used for call setup and control. MGCP is based on a master-slave call control architecture in which the media gateway controller, via the call agent, maintains call control intelligence, while the media gateways carry out the instructions of the call agent.

The MGCP ALG performs the following procedures:

- Conducts VoIP signaling payload inspection. The payload of the incoming VoIP signaling packet is fully inspected based on related RFCs and proprietary standards. Any malformed packet attack is blocked by the ALG.
- Conducts MGCP signaling payload inspection. The payload of the incoming MGCP signaling packet is fully inspected in accordance with RFC 3435. Any malformed-packet attack is blocked by the ALG.
- Provides stateful processing. The corresponding VoIP-based state machines are invoked to process the parsed information. Any out-of-state or out-of-transaction packet is identified and properly handled.
- Performs Network Address Translation (NAT). Any embedded IP address and port information in the payload is properly translated based on the existing

routing information and network topology, and is replaced with the translated IP address and port number, if necessary.

- Manages pinholes for VoIP traffic. To keep the VoIP network secure, the IP address and port information used for media or signaling is identified by the ALG, and any needed pinhole is dynamically created and closed during call setup.

MGCP Security

The MGCP ALG includes the following security features:

- Denial of Service (DoS) attack protection—the ALG performs stateful inspection at the UDP packet level, the transaction level, and at the call level. MGCP packets matching the RFC 3435 message format, transaction state, and call state, are processed. All other messages are dropped.
- Firewall policy enforcement between gateway and gateway controller (signaling policy).
- Firewall policy enforcement between gateways (media policy).
- Per-gateway MGCP message flooding control. Any malfunctioning or hacked gateway will not disrupt the whole VoIP network. Combined with per-gateway flooding control, damage is contained within the impacted gateway.
- Per-gateway MGCP connection flooding control.
- Seamless switchover/failover if calls, including calls in progress, are switched to the standby firewall in case of system failure.

About MGCP

MGCP is a text-based, application layer protocol that can be used for call setup and control. The protocol is based on a master/slave call control architecture: the media gateway controller (call agent) maintains call control intelligence, and media gateways carry out the instructions from the call agent.

Entities in MGCP

There are four basic entities in MGCP:

- Endpoint on page 1158
- Connection on page 1159
- Call on page 1159
- Call Agent on page 1159

Endpoint

A media gateway (MG) is a collection of endpoints. An endpoint can be an analog line, trunk, or any other access point. An endpoint is named as below:

```
local-endpoint-name@domain-name
```

The following are some valid endpoint IDs:

group1/Trk8@mynetwork.net
group2/Trk1/*@[192.168.10.8] (wild-carding)
$@voiptel.net (any endpoint within the MG)
*@voiptel.net (all endpoints within the MG)

Connection

Connections are created on each endpoint by a MG during call setup. A typical VoIP call involves two connections. A complex call, for example a three-party call or conference call, might require more connections. The media gateway controller (MGC) can instruct media gateways to create, modify, delete and audit a connection.

A connection is identified by its connection ID which is created by the MG when it is requested to create a connection. Connection ID is presented as a hexadecimal string, and its maximum length is 32 characters.

Call

A call is identified by its call ID, which is created by the MGC when establishing a new call. Call ID is a hexadecimal string with a maximum length of 32 characters. Call ID is unique within the MGC. Two or more connections can have the same call ID if they belong to the same call.

Call Agent

One or more call agents (also called media gateway controllers) are supported in MGCP to enhance reliability in VoIP network. The following are two examples of call agent names:

CallAgent@voipCA.mynetwork.com
voipCA.mynetwork.com

Several network addresses can be associated under one domain name in the Domain Name System (DNS). By keeping track of the time to live (TTL) of DNS query/response data and implementing retransmission using other alternative network addresses, switchover and failover is achieved in MGCP.

The concept of *notified entity* is essential in MGCP. The notified entity for an endpoint is the call agent currently controlling that endpoint. An endpoint should send any MGCP command to its notified entity. However, different call agents might send MGCP commands to this endpoint.

The notified entity is set to a provisioned value upon startup, but could be changed by a call agent through the use of a *Notified Entity* parameter contained in a MGCP message. If the notified entity for an endpoint is empty or has not been set explicitly, its value defaults to the source address of the last successful non-audit MGCP command received for that endpoint.

Commands

The MGCP protocol defines nine commands for controlling endpoints and connections. All commands are composed of a command header, optionally followed by session description protocol (SDP) information. A command header has the following elements:

- A command line: command verb + transaction ID + endpointId + MGCP version.
- Zero or more parameter lines, composed of a parameter name followed by a parameter value.

Table 74 on page 1160 lists supported MGCP commands, with a description of each, the command syntax, and examples. Refer to RFC 2234 for a complete explanation of command syntax.

Table 74: MGCP Commands

Command Verb	Description	Command Syntax	Examples
EPCF	EndpointConfiguration—used by a call agent to inform a gateway of coding characteristics (a-law or mu-law) expected by the line side of the endpoint.	ReturnCode [PackageList] EndpointConfiguration (EndpointId, [BearerInformation])	EPCF 2012 wxx/T2@mynet.com MGCP 1.0 B: e:mu
CRCX	CreateConnection—used by a call agent to instruct the gateway to create a connection with, and endpoint inside, the gateway.	ReturnCode, [ConnectionId,] [SpecificEndPointId,] [LocalConnectionDescriptor,] [SecondEndPointId,] [SecondConnectionId,][Pac kageList] CreateConnection (CallId, EndpointId, [NotifiedEntity,] [LocalConnectionOption,] Mode, [{RemoteConnectionDescriptor | SecondEndpoindId},] [encapsulated RQNT,] [encapsulated EPCF])	CRCX 1205 aaln/1@gw-25.att.net MGCP 1.0 C: A3C47F21456789F0 L: p:10, a:PCMU M: sendrecv X: 0123456789AD R: L/hd S: L/rg v = 0 o = - 25678 753849 IN IP4 128.96.41.1 s = - c = IN IP4 128.96.41.1 t = 0 0 m = audio 3456 RTP/AVP 0

Table 74: MGCP Commands *(continued)*

Command Verb	Description	Command Syntax	Examples
MDCX	ModifyConnection—used by a call agent to instruct a gateway to change the parameters for an existing connection.	ReturnCode, [LocalConnectionDescriptor,] [PackageList] ModifyConnection (CallId, EndpointId, ConnectionId, [NotifiedEntity,] [LocalConnectionOption,] [Mode,] [RemoteConnectionDescriptor,] [encapsulated RQNT,] [encapsulated EPCF])	MDCX 1210 aaln/1@rgw-25.att.net MGCP 1.0 C: A3C47F21456789F0 I: FDE234C8 M: recvonly X: 0123456789AE R: L/hu S: G/rt v = 0 o = - 4723891 7428910 IN IP4 128.96.63.25 s = - c = IN IP4 128.96.63.25 t = 0 0 m = audio 3456 RTP/AVP 0
DLCX	DeleteConnection—used by a call agent to instruct a gateway to delete an existing connection. DeleteConnection can also be used by a gateway to release a connection that can no longer be sustained.	ReturnCode, ConnectionParameters, [PackageList] DeleteConnection (CallId, EndpointId, ConnectionId, [NotifiedEntity,] [encapsulated RQNT,] [encapsulated EPCF])	Example 1: MGC -> MG DLCX 9210 aaln/1@rgw-25.att.net MGCP 1.0 C: A3C47F21456789F0 I: FDE234C8 Example 2: MG -> MGC DLCX 9310 aaln/1@rgw-25.att.net MGCP 1.0 C: A3C47F21456789F0 I: FDE234C8 E: 900 - Hardware error P: PS = 1245, OS = 62345, PR = 780, OR = 45123, PL = 10, JI = 27, LA = 48
RQNT	The NotificationRequest command is used by a call agent to instruct a MG to monitor for certain event(s) or signal(s) for a specific endpoint.	ReturnCode, [PackageList] NotificationRequest[(EndpointId, [NotifiedEntity,] [RequestedEvents,] RequestIdentifier, [DigitMap,] [SignalRequests,] [QuarantineHandling,] [DetectEvents,] [encapsulated EPCF])	RQNT 1205 aaln/1@rgw-25.att.net MGCP 1.0 N: ca-new@callagent-ca.att.net X: 0123456789AA R: L/hd(A, E(S(L/dl),R(L/oc,L/hu,D/[0-9#*T](D)))) D: (0T|00T|xx|91xxxxxxxxxx|9011x.T) S: T: G/ft
NTFY	Notify—used by a gateway to inform the call agent when requested event(s) or signal(s) occur.	ReturnCode, [PackageList] Notify (EndpointID, [NotifiedEntity,] RequestIdentifier, ObservedEvents)	NTFY 2002 aaln/1@rgw-25.att.net MGCP 1.0 N: ca@ca1.att.net:5678 X: 0123456789AC O: L/hd,D/9,D/1,D/2,D/0,D/1,D/8,D/2,D/9,D/4, D/2,D/6,D/6

Table 74: MGCP Commands *(continued)*

Command Verb	Description	Command Syntax	Examples
AUEP	AuditEndpoint—used by a call agent to audit the status of the endpoint.	ReturnCode, EndPointIdList,	{ [RequestedEvents,] [QuarantineHandling,] [DigitMap,] [SignalRequests,] [RequestedIdentifier,] [NotifiedEntity,] [ConnectionIdentifier,] [DetectEvents,] [ObservedEvents,] [EventStats,] [BearerInformation,] [BearerMethod,] [RestartDelay,] [ReasonCode,] [MaxMGCPDatagram,] [Capabilities]} [PackageList] AuditEndpoint (EndpointId, [RequestedInfo])	Example 1: AUEP 1201 aaln/1@rgw-25.att.net MGCP 1.0 F: A, R,D,S,X,N,I,T,O Example 2: AUEP 1200 *@rgw-25.att.net MGCP 1.0
AUCX	AuditConnection—used by a call agent to collect the parameters applied to a connection.	ReturnCode, [CallId,] [NotifiedEntity,] [LocalConnectionOptions,] [Mode,] [RemoteConnectionDescriptor,] [LocalConnectionDescriptor,] [ConnectionParameters,] [PackageList] AuditConnection (EndpointId, ConnectionId, RequestedInfo)	AUCX 3003 aaln/1@rgw-25.att.net MGCP 1.0 I: 32F345E2 F: C,N,L,M,LC,P
RSIP	RestartInProgress—used by a gateway to notify a call agent that one or more endpoints are being taken out of service or placed back in service.	ReturnCode, [NotifiedEntity,] [PackageList] RestartInProgress (EndpointId, RestartMethod, [RestartDelay,] [ReasonCode])	RSIP 5200 aaln/1@rg2-25.att.net MGCP 1.0 RM: graceful RD: 300

Response Codes

Every command sent by the calling agent or gateway, whether successful or not, requires a response code. The response code is in the header of the response message, and optionally is followed by session description information.

The response header is composed of a response line, followed by zero or more parameter lines, each containing a parameter name letter followed by its value. The response header is composed of a 3-digit response code, transaction ID, and optionally followed by commentary. The response header in the following response message shows the response code 200 (successful completion), followed by ID 1204, and the comment: OK:

```
200 1204 OK
I: FDE234C8
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 96
a=rtpmap:96 G726-32/8000
```

The ranges of response codes are defined as follows:

- 000 – 099: indicate a response acknowledgement.
- 100 – 199: indicate a provisional response.
- 200 – 299: indicate a successful completion (final response).
- 400 – 499: indicate a transient error (final response).
- 500 – 599: indicate a permanent error (final response).

Refer to RFC 3661 for detailed information about response codes.

A response to a command is sent to the source address of the command, not to the current notified entity. A media gateway can receive MGCP commands from various network addresses simultaneously, and send back responses to corresponding network addresses. However, it sends all MGCP commands to its current notified entity.

Examples

This section includes the following configuration scenarios:

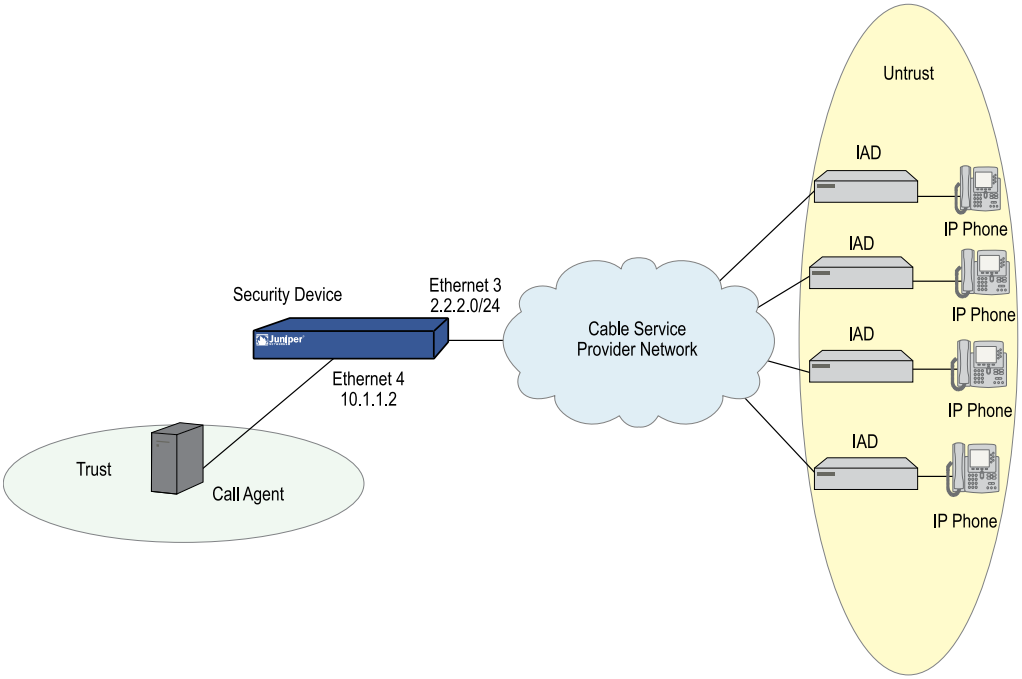
- Media Gateway in Subscribers' Homes—Call Agent at the ISP on page 1163
- ISP-Hosted Service on page 1166

Media Gateway in Subscribers' Homes—Call Agent at the ISP

In this example (see Figure 306 on page 1164) you configure a security device at a Cable Service Provider to support MGCP for their network of residential subscribers. The security device and the call agent are on the cable service provider's premises. An integrated Access Device (IAD), or set-top box, is in each subscriber's home, acting as a gateway—each IAD represents a separate residence. The call agent is in the trust_ca zone; residential customers are in the res_cust zone.

After creating zones—untrust_subscriber for the customers and trust_ca for the service provider, you configure addresses, and then policies. Although gateways frequently reside in different zones, requiring policies for media traffic, in this example both gateways are in the same subnet. RTP traffic between the gateways never passes through the firewall, therefore no policy is needed for media.

Figure 306: Media Gateway in Subscribers' Home



WebUI

1. **Zones**

   Network > Zones > New: Enter the following, then click **OK**:

   Zone Name: untrust_subscriber

   Network > Zones > New: Enter the following, then click **OK**:

   Zone Name: trust_ca

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: SubscriberSubNet
   Comment: Our subscribers' network
   IP Address/Domain Name:
   IP/Netmask: (select), 2.2.2.0/24
   Zone: untrust-subscriber

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: call_agent1
Comment: Our No. 1 call agent
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.101/32
Zone: trust_ca

3. **Interfaces**

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **Apply**:

Zone Name: untrust_subscriber
Static IP: (select this option when present)
IP Address/Netmask: 2.2.2.0/24
Enter the following, then click **OK**:
Interface Mode: route

Network > Interfaces > Edit (for ethernet4): Enter the following, then click **Apply**:

Zone Name: trust_ca
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.101/32
Enter the following, then click **OK**:
Interface Mode: route

4. **Policies**

Policies > (From: trust-ca, To: untrust_subscriber) New: Enter the following, then click **OK**:

Name: Pol-CA-To-Subscribers
Source Address
Address Book Entry: (select), call_agent1
Destination Address
Address Book Entry: (select), SubscriberSubNet
Service: MGCP-UA
Action: Permit

Policies > (From: untrust_subscriber, To: trust-ca) New: Enter the following, then click **OK**:

Name: Pol-Subscribers-To-CA
Source Address
Address Book Entry: (select), SubscriberSubNet
Destination Address
Address Book Entry: (select), call_agent1
Service: MGCP-CA

Action: Permit

CLI

1. **Zones**

```
set zone name untrust_subscriber
set zone name trust_ca
```

2. **Addresses**

```
set address untrust_subscriber SubscriberSubNet 2.2.2.0 255.255.255.0 “
Our subscribers' network”
set address trust_ca call_agent1 10.1.1.101 255.255.255.255 “ Our No. 1
call agent”
```

3. **Interfaces**

```
set interface ethernet3 zone untrust_subscriber “ Our subscribers’ network”
set interface ethernet3 ip 2.2.2.0/24
set interface ethernet3 route
set interface ethernet4 zone trust_ca “ Our No. 1 call agent”
set interface ethernet4 ip 10.1.1.2/24
set interface ethernet4 route
```

4. **Policies**

```
set policy name Pol-CA-TO-Subscribers from trust_ca to untrust_subscriber
call_agent1 SubscriberSubNet mgcp-ua permit
set policy name Pol-Subscribers-To-CA from untrust_subscriber to trust_ca
SubscriberSubNet call_agent1 mgcp-ca permit
```

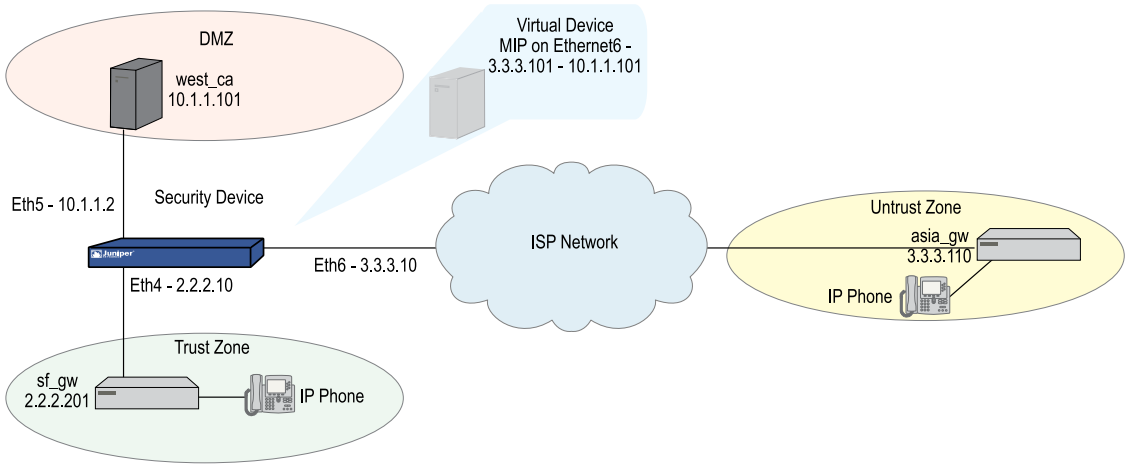
ISP-Hosted Service

In this example, (see Figure 307 on page 1167) an ISP located on the American west coast provides MGCP service to customers in Asia and San Francisco. Asia customers are in the Untrust zone, and supported by the gateway: asia_gw (3.3.3.110); San Francisco customers are in the Trust zone, and supported by the gateway: sf_gw (2.2.2.201). The call agent: west_ca (10.1.1.101) is in the DMZ.

After setting addresses for the gateways and the call agent, you configure the interfaces, putting ethernet4 and ethernet5, which are trusted, in route mode to allow them to stream media directly after call setup. To protect the IP address of the call agent in the DMZ from exposure, you place a MIP on ethernet6, that is, you map the IP address of the call agent (10.1.1.101) to an IP address from the pool of addresses on the ethernet6 interface, in this case: 3.3.3.101.

Finally, you create policies. To allow MGCP signaling between the call agent in the DMZ and the gateway in the Untrust zone, you create one policy for each direction, referencing the MIP that protects the call agent. You create another pair of policies to allow signaling between the call agent and the gateway in the Trust zone. A single policy is sufficient to allow bidirectional communication between gateways in the Trust and Untrust zones.

Figure 307: ISP-Hosted Service



WebUI

1. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: sf_gw
   Comment: gateway in asia
   IP Address/Domain Name:
   IP/Netmask: (select), 2.2.2.201/32
   zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: asia_gw
   Comment: gateway in asia
   IP Address/Domain Name:
   IP/Netmask: (select), 3.3.3.110/32
   zone: Untrust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: west_ca
   Comment: ca in west coast
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.101/32
   zone: DMZ

2. **Interfaces**

   Network > Interfaces > Edit (for ethernet4): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 2.2.2.10/24
Enter the following, then click **OK**:
Interface Mode: route

Network > Interfaces > Edit (for ethernet5): Enter the following, then click **Apply**:

Zone Name: DMZ
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.2/24
Enter the following, then click **OK**:
Interface Mode: route

Network > Interfaces > Edit (for ethernet6): Enter the following, then click **Apply**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 3.3.3.10/24
Enter the following, then click **OK**:
Interface Mode: NAT

3. **MIP**

Network > Interfaces > Edit (for ethernet6) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 3.3.3.101
Netmask: 255.255.255.255
Host IP Address: 10.1.1.101
Host Virtual Router Name: trust-vr

4. **Policies**

Policies > (From: DMZ To: Untrust) New: Enter the following, then click **OK**:

Source Address
  Address Book Entry: (select), west_ca
Destination Address
  Address Book Entry: (select), asia_gw
Service: MGCP-UA
Action: Permit

Policies > (From: Untrust To: DMZ) New: Enter the following, then click **OK**:

Source Address
  Address Book Entry: (select), asia_gw
Destination Address
  Address Book Entry: (select), west_ca
Service: MGCP-CA
Action: Permit

Policies > (From: Trust To: DMZ) New: Enter the following, then click **OK**:

Source Address
Address Book Entry: (select), sf_gw
Destination Address
Address Book Entry: (select), west_ca
Service: MGCP-CA
Action: Permit

Policies > (From: DMZ To: Trust) New: Enter the following, then click **OK**:

Source Address
Address Book Entry: (select), west_ca
Destination Address
Address Book Entry: (select), sf_gw
Service: MGCP-UA
Action: Permit

Policies > (From: Trust To: Untrust) New: Enter the following, then click **OK**:

Source Address
Address Book Entry: (select), sf_gw
Destination Address
Address Book Entry: (select), asia_gw
Service: MGCP-UA
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
DIP on: None (Use Egress Interface IP)

CLI

1. **Addresses**

```
set address trust sf_gw 2.2.2.201/32 " gateway in s.f."
set address untrust asia_gw 3.3.3.110/32 " gateway in asia"
set address dmz west_ca 10.1.1.101/32 " ca in west coast"
```

2. **Interfaces**

```
set interface ethernet4 ip 2.2.2.10/24
set interface ethernet4 route
set interface ethernet4 zone trust
set interface ethernet5 ip 10.1.1.2/24
set interface ethernet5 route
set interface ethernet5 zone dmz
set interface ethernet6 ip 3.3.3.10/24
set interface ethernet6 zone untrust
```

3. **Mapped IP Address**

```
set interface ethernet6 mip 3.3.3.101 host 10.1.1.101 netmask
255.255.255.255 vrouter trust-vr
```

4. **Policies**

```
set policy from dmz to untrust west_ca asia_gw mgcp-ua permit
set policy from untrust to dmz asia_gw mip(3.3.3.101) mgcp-ca permit
set policy from trust to dmz sf_gw west_ca mgcp-ca permit
set policy from dmz to trust west_ca sf_gw mgcp-ua permit
set policy from trust to untrust sf_gw asia_gw mgcp-ua nat src permit
```

Chapter 30

Skinny Client Control Protocol Application Layer Gateway

This chapter presents an overview of the Skinny Client Control Protocol (SCCP) Application Layer Gateway (ALG) and lists the firewall security features of the implementation. Examples of typical scenarios follow a summary of the SCCP architecture. This chapter includes the following sections:

Overview

Skinny Client Control Protocol (SCCP) is supported on security devices in Route, transparent, and Network Address Translation (NAT) modes. SCCP is a binary-based Application Layer protocol used for Voice-over-Internet Protocol (VoIP) call setup and control. In the SCCP architecture, a Cisco H.323 proxy, known as the Call Manager, does most of the processing. IP phones, also called End Stations, run the Skinny client and connect to a primary (and, if available, a secondary) Call Manager over TCP on port 2000 and register with the primary Call Manager. This connection is then used to establish calls coming to or from the client.

The SCCP ALG supports the following:

- Call flow from a Skinny client, through the Call Manager, to another Skinny client.
- Seamless failover—switches over all calls in process to the standby firewall during failure of the primary.
- VoIP signaling payload inspection—fully inspects the payload of incoming VoIP signaling packets based on related RFCs and proprietary standards. Any malformed packet attack is blocked by the ALG.
- SCCP signaling payload inspection—fully inspects the payload of incoming SCCP signaling packets in accordance with RFC 3435. Any malformed-packet attack is blocked by the ALG.
- Stateful processing—invokes the corresponding VoIP-based state machines to process the parsed information. Any out-of-state or out-of-transaction packet is identified and properly handled.

- Network Address Translation (NAT)—translates any embedded IP address and port information in the payload, based on the existing routing information and network topology, with the translated IP address and port number, if necessary.
- Pinhole creation and management for VoIP traffic—identifies IP address and port information used for media or signaling and dynamically opens (and closes) pinholes to securely stream the media.

SCCP Security

The SCCP ALG includes the following security features:

- Denial of Service (DoS) attack protection—The ALG performs stateful inspection at the UDP packet level, the transaction level, and the call level. Packets matching the SCCP message format, transaction state, and call state are processed. All other messages are dropped.
- Firewall policy enforcement between Cisco IP phones and the Call Manager (Intra-Cluster).
- Firewall policy enforcement between Call Managers (Inter-Cluster).
- Call Manager flood control—Protects the Call Manager from being flooded with new calls either by an already compromised connected client or by a faulty device.
- Firewall policy enforcement between gateways (media policy).
- Per-gateway SCCP connection flooding control.
- Seamless switchover/failover if calls, including calls in progress, are switched to the standby firewall in case of system failure.

About SCCP

The following sections give a brief overview of SCCP and how it works:

- SCCP Components on page 1172
- SCCP Transactions on page 1173
- SCCP Messages on page 1176

SCCP Components

The principle components of the SCCP VoIP architecture include the following:

- SCCP Client on page 1173
- Call Manager on page 1173
- Cluster on page 1173

SCCP Client

The SCCP client runs on an IP phone, also called an End Station, which uses SCCP for signaling and for making calls. In order for a Skinny client to make a call, it must first register with a Primary Call Manager (and a secondary, if available). The connection between the client and the Call Manager is over TCP on port 2000. This connection is then used to establish calls to or from the client. Transmission of media is over RTP, UDP, and IP.

Call Manager

The Call Manager is a Cisco H.323 server with overall control of all devices and communication in the SCCP VoIP network. Its functions include defining, monitoring and controlling SCCP groups, regions of numbers, and route plans;providing initialization, admission and registration of devices on the network; providing a redundant database that contains addresses, phone numbers, and number formats; and initiating contact with called devices or their agents to establish logical sessions in which voice communication can flow.

Cluster

A Cluster is a collection of SCCP clients and a Call Manager. The Call Manager in the cluster knows about all SCCP clients in the cluster. There can be more than one Call Manager for backup in a cluster. Call Manager behavior varies in each of the following cluster scenarios:

- Intra-Cluster, in which the Call Manager knows about each SCCP client, and the call is between SCCP clients of the same cluster.
- Inter-Cluster, in which the Call Manager needs to communicate with another Call Manager using H.323 for call setup.
- Inter-Cluster calls using the gatekeeper for admission control and address resolution.

  Call Manager behavior also varies with calls between an SCCP client and a phone in a Public Switched Telephone Network (PSTN), and with calls between an SCCP client and a phone in another administrative domain that is using H323.

SCCP Transactions

SCCP transactions are the processes that need to take place in order for an SCCP call to proceed. SCCP transactions include the following:

- "Client Initialization" on page 1174
- "Client Registration" on page 1174
- "Call Setup" on page 1175
- "Media Setup" on page 1175

Client Initialization

To initialize, the SCCP client needs to know the IP address of the Call Manager, its own IP address, and other information about the IP gateway and DNS servers. Initialization takes place on the local LAN. The client sends a Dynamic Host Control Protocol (DHCP) request to get an IP address, the DNS server address, and the TFTP server name and address. The client needs the TFTP server name to download the configuration file: sep*macaddr*.cnf. If the TFTP name is not given, the client uses the default filename in the IP phone. The client then downloads the configuration file .cnf (xml) from TFTP server. CNF files contain the IP address or addresses of the primary and secondary Cisco Call Manager. With this information, the client contacts the Call Manager to register.

Client Registration

The SCCP client, after initialization, registers with the Call Manager over a TCP connection on well-known default port 2000. The client registers by providing the Call Manager with its IP address, the MAC address of the phone, and other information, such as protocol and version. The client cannot initiate or receive calls until it is registered. Keepalive messages keep this TCP connection open between the client and Call Manager so that the client can initiate or receive calls at any time, provided that a policy on the security device allows this.

Table 75 on page 1174 lists SCCP messages and indicates messages that are of interest to the security device.

Table 75: SCCP Registration Messages

From Client	From Call Manager	Of Interest to Security Device
RegisterMessage		b
IPortMessage		b
	RegisterAckMessage	b
	CapabilititsRequest	
CapabilitiesResMessage		
ButtonTemplateReqMessage		
	ButtonTemplateResMessage	
SoftKeyTemplateReqMessage		
	SoftKeyTemplateResMessage	
LineStatReqMessage		b
	LineStatMessage	b

Call Setup

IP phone-to-IP phone call-setup using SCCP is always handled by the Call Manager. Messages for call setup are sent to the Call Manager, which returns messages appropriate to the status of the call. If call setup is successful, and a policy on the security device allows the call, the Call Manager sends the media setup messages to the client.

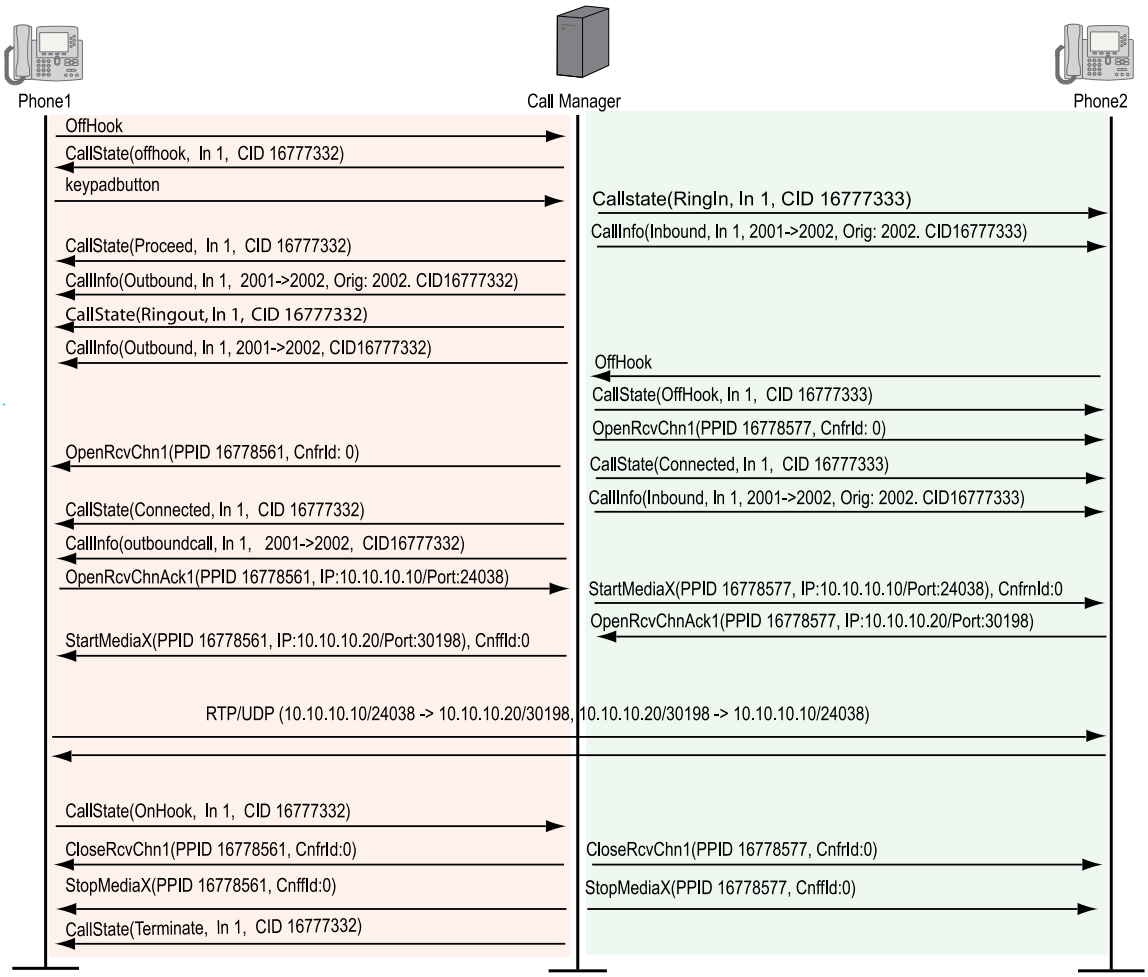
Media Setup

The Call Manager sends the IP address and port number of the called party to the calling party. The Call Manager also sends the media IP address and port number of the calling party to the called party. After media setup, media is transmitted directly between clients. When the call ends, the Call Manager is informed and terminates the media streams. At no time during this process does the Call Manager hand over call-setup function to the client. Media is streamed directly between clients through RTP/UDP/IP.

SCCP Control Messages and RTP Flow

Figure 308 on page 1176 shows the SCCP control messages used to set up and tear down a simple call between *Phone1* and *Phone2*. Except for the OffHook message initiating the call from *Phone1* and the OnHook message signaling the end of the call, all aspects of the call are controlled by the Call Manager.

Figure 308: Call Setup and Teardown



SCCP Messages

Table 76 on page 1176, Table 77 on page 1177, Table 78 on page 1177, and Table 79 on page 1177 list the SCCP call message IDs in the four intervals allowed by the security device.

Table 76: Station to Call Manager Messages

Message	Range
#define STATION_REGISTER_MESSAGE	0x00000001
#define STATION_IP_PORT_MESSAGE	0x00000002
#define STATION_ALARM_MESSAGE	0x00000020
#define STATION_OPEN_RECEIVE_CHANNEL_ACK	0x00000022

Table 77: Call Manager to Station Messages

Message	Range
#define STATION_START_MEDIA_TRANSMISSION	0x00000001
#define STATION_STOP_MEDIA_TRANSMISSION	0x00000002
#define STATION_CALL_INFO_MESSAGE	0x00000020
#define STATION_OPEN_RECEIVE_CHANNEL_ACK	0x00000022
#define STATION_CLOSE_RECEIVE_CHANNEL	0x00000106

Table 78: Call Manager 4.0 Messages and Post Skinny 6.2

Message	Range
#define STATION_REGISTER_TOKEN_REQ_MESSAGE	0x00000029
#define STATION_MEDIA_TRANSMISSION_FAILURE	0x0000002A
#define STATION_OPEN_MULTIMEDIA_RECEIVE_CHANNEL_ACK	0x00000031

Table 79: Call Manager to Station

Message	Range
#define STATION_OPEN_MULTIMEDIA_RECEIVE_CHANNEL	0x00000131
#define STATION_START_MULTIMEDIA_TRANSMISSION	0x00000132
#define STATION_STOP_MULTIMEDIA_TRANSMISSION	0x00000133
#define STATION_CLOSE_MULTIMEDIA_RECEIVE_CHANNEL	0x00000136

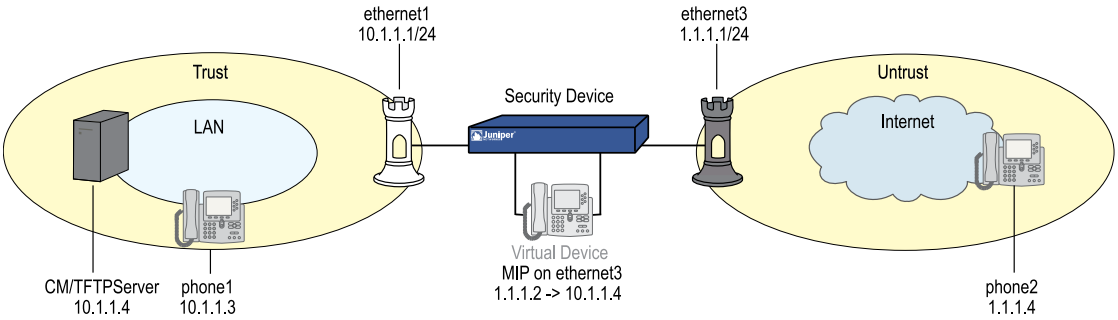
Examples

This section contains the following sample scenarios:

Example: Call Manager/TFTP Server in the Trust Zone

In this example, phone1 and the Call Manager/TFTP Server are on the ethernet1 interface in the Trust (private) zone, and phone2 is on the ethernet3 interface in the Untrust zone. You put a MIP for the Call Manager/TFTP Server on the ethernet3 interface, so that when phone2 boots up it can contact the TFTP Server and obtain the IP address of the Call Manager. (We recommend that you change the IP address of the Call Manager in the TFTP Server config file (sep <mac_addr>.cnf) to the MIP IP address of the Call Manager.) You then create a policy allowing SCCP traffic from the Untrust to the Trust zone and reference that MIP in the policy. You also create a policy from the Trust to the Untrust zone to allow phone1 to call out.

Figure 309: Call Manager/TFTP Server in the Private Zone



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

   Zone: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click **OK:**
   Interface Mode: route

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone: Untrust
   IP Address/Netmask: 1.1.1.1/24
   Interface Mode: Route

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: phone1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.3/24
   Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone2
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.4/24
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: CM-TFTP_Server
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.4/24
Zone: Trust

3. **MIP**

Network > Interfaces > Edit (for loopback.3) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 1.1.1.2
Netmask: 255.255.255.255
Host IP Address: 10.1.1.4
Host Virtual Router Name: trust-vr

4. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select) any
Destination Address:
Address Book Entry: (select) phone2
Service: SCCP
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): None (Use Egress Interface IP)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select), phone2
Destination Address:
Address Book Entry: (select), MIP(1.1.1.2)
Service: SCCP
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 route
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 route
```

2. **Addresses**

```
set address trust phone1 10.1.1.3/24
set address untrust phone2 1.1.1.4/24
set address trust cm-tftp_server 10.1.1.4/24
```

3. **MIP**

```
set interface ethernet3 mip 1.1.1.2 host 10.1.1.4
```

4. **Policies**

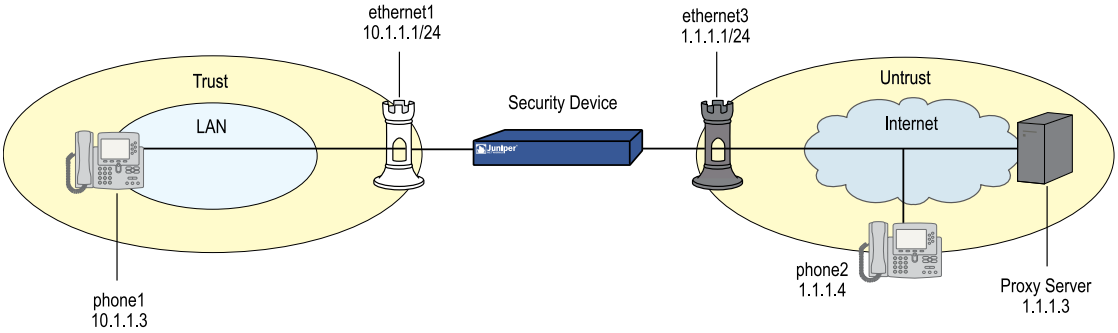
```
set policy from trust to untrust any phone2 sccp nat src permit
set policy from untrust to trust phone2 mip(1.1.1.2) sccp permit
save
```

NOTE: It is always more secure to specify a service explicitly, as shown in this example configuration, than to use the keyword **any**.

Example: Call Manager/TFTP Server in the Untrust Zone

In this example, phone1 is on the ethernet1 interface in the Trust zone, and phone2 and the Call Manager/TFTP Server are on the ethernet3 interface in the Untrust zone. After configuring interfaces and addresses, you create policy from the Trust zone to the Untrust. This allows phone1 to register with the Call Manager/TFTP Server in the Untrust zone.

Figure 310: Call Manager/TFTP Server in the Untrust Zone



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Enter the following, then click **OK**:

   Interface Mode: route

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24
   Interface Mode: Route

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: phone1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.3/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: phone2
   IP Address/Domain Name:
   IP/Netmask: (select), 1.1.1.4/24
   Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: CM/TFTP Server
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.3/24
Zone: Untrust

3. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address
Address Book Entry: (select) phone1
Destination Address
Address Book Entry: (select) any
Service: SCCP
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): None (Use Egress Interface IP)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 route
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 route
```

2. **Addresses**

```
set address trust phone1 10.1.1.3/24
set address untrust phone2 1.1.1.4/24
set address untrust cm-tftp_server 1.1.1.3/24
```

3. **Policies**

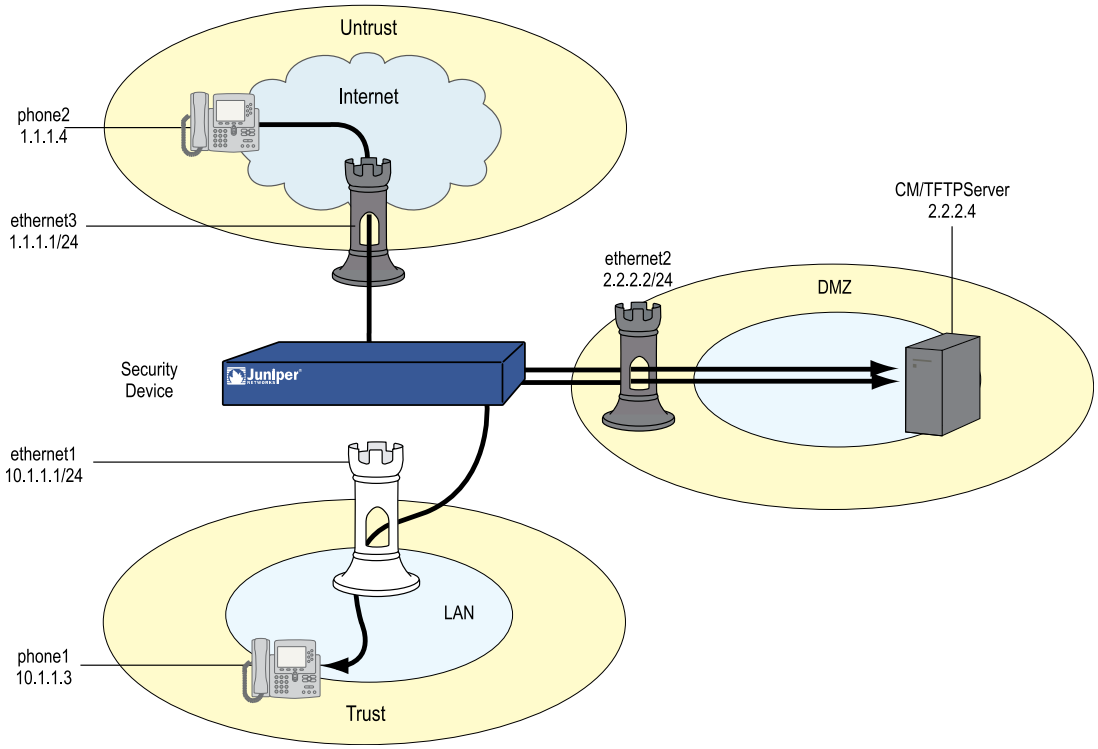
```
set policy from trust to untrust phone1 any sccp nat src permit
save
```

NOTE: It is always more secure to specify a service explicitly, as shown in this example configuration, than to use the keyword **any**.

Example: Three-Zone, Call Manager/TFTP Server in the DMZ

In this example, phone1 is on the ethernet1 interface in the Trust zone, phone2 is on the ethernet3 interface in the Untrust zone, and the Call Manager/TFTP Server is on the ethernet2 interface in the DMZ. For signaling, you create a policy from the Trust zone to the DMZ to allow phone1 to communicate with the Call Manager/TFTP Server, and you create a policy from the Untrust zone to the DMZ to allow phone2 to communicate with the Call Manager/TFTP Server. For transmission of media, you create a policy from Trust to Untrust to allow phone1 and phone2 to communicate directly. The arrows in Figure 311 on page 1183 show the flow of SCCP signaling traffic when phone2 in the Untrust zone places a call to phone1 in the Trust zone. After the session is initiated, the media flows directly between phone1 and phone2.

Figure 311: Call Manager/TFTP Server in the DMZ



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone: Trust
   Static IP: (select when this option is present)
   IP Address/Netmask: 10.1.1.1/24

Enter the following, then click OK:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: DMZ
Static IP: (select when this option is present)
IP Address/Netmask: 2.2.2.2/24

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 1.1.1.1/24

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone1
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.3/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone2
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.4/24
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: CM-TFTP_Server
IP Address/Domain Name:
IP/Netmask: (select), 2.2.2.4/24
Zone: DMZ

3. **Policies**

Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select), phone1
Destination Address:
Address Book Entry: (select), CM-TFTP_Server
Service: SCCP
Action: Permit

> Advanced: Enter the following, then click Return to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: Enable
(DIP on): None (Use Egress Interface IP)

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select), phone2
Destination Address:
Address Book Entry: (select), CM-TFTP_Server
Service: SCCP
Action: Permit

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select), phone1
Destination Address:
Address Book Entry: (select), phone2
Service: SCCP
Action: Permit

> Advanced: Enter the following, then click Return to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: Enable
(DIP on): None (Use Egress Interface IP)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 route
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 route
set interface ethernet2 zone dmz
set interface ethernet2 ip 2.2.2.2/24
set interface ethernet2 route
```

2. **Addresses**

```
set address trust phone1 10.1.1.3/24
set address untrust phone2 1.1.1.4/24
set address dmz cm-tftp_server 2.2.2.4
```

3. **Policies**

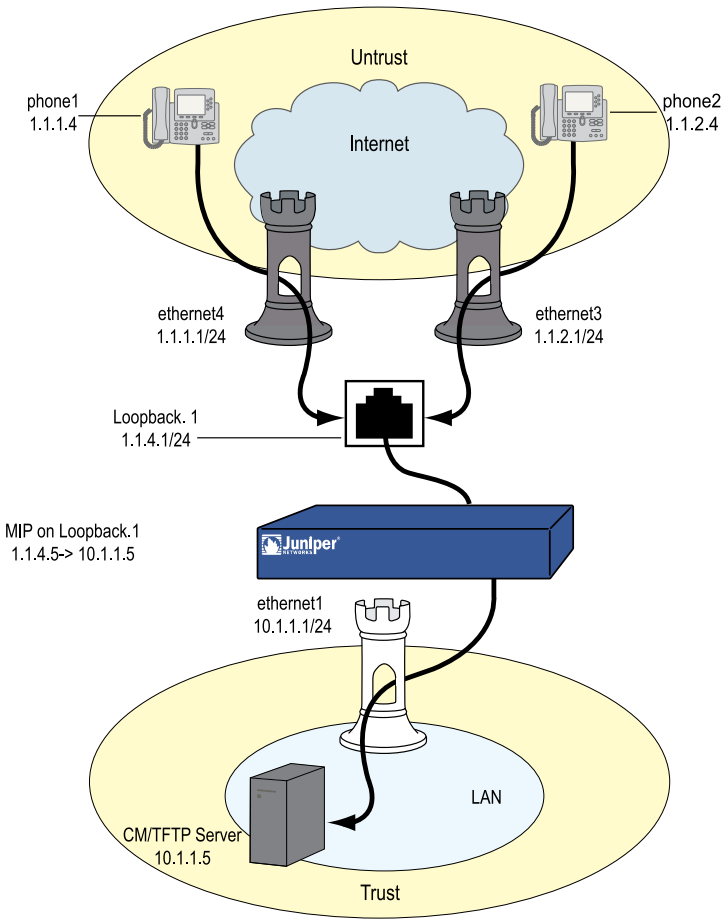
```
set policy from trust to dmz phone1 cm-tftp_server sccp nat src permit
set policy from untrust to dmz phone2 cm-tftp_server sccp permit
set policy from trust to untrust phone1 phone2 sccp nat src permit
save
```

NOTE: It is always more secure to specify a service explicitly, as shown in this example configuration, than to use the keyword any.

Example: Intrazone, Call Manager/TFTP Server in Trust Zone

In this example, phone1 is on the ethernet4 interface in the Untrust zone, phone2 is in a subnet on the ethernet3 interface in the Untrust zone, and the Call Manager/TFTP Server is on the ethernet1 interface in the Trust zone. To allow intrazone SCCP traffic between the two phones in the Untrust zone, you create a loopback interface, add ethernet3 and ethernet4 to a loopback group, then put a MIP on the loopback interface to the IP address of the Call Manager/TFTP Server. Creating a loopback interface enables you to use a single MIP for the Call Manager/TFTP Server in the Trust zone. (For more information about using loopback interfaces, see “MIP and the Loopback Interface” on page 1545 .) And finally, because intrazone blocking is on by default, you unset blocking in the Untrust zone to allow intrazone communication.

Figure 312: Intrazone, Call Manager/TFTP Server in Trust Zone



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone: Trust
   Static IP: (select when this option is present)
   IP Address/Netmask: 10.1.1.1/24

   Enter the following, then click **OK:**

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet4): Enter the following, then click **OK**:

   Zone: Untrust
   Static IP: (select when this option is present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone: Untrust
   Static IP: (select when this option is present)
   IP Address/Netmask: 1.1.2.1/24

   Network > Interfaces > New Loopback IF: Enter the following, then click **OK**:

   Interface Name: loopback.1
   Zone: Untrust (trust-vr)
   IP Address/Netmask: 1.1.4.1/24

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: CM-TFTP_Server
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.5/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: phone1
   IP Address/Domain Name:
   IP/Netmask: (select), 1.1.1.4/32
   Zone: Untrust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone2
IP Address/Domain Name:
IP/Netmask: (select), 1.1.2.4/32
Zone: Untrust

3. **Loopback Group**

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

As member of loopback group: (select) loopback.1
Zone Name: Untrust

Network > Interfaces > Edit (for ethernet4): Enter the following, then click **OK**:

As member of loopback group: (select) loopback.1
Zone Name: Untrust

4. **MIP**

Network > Interfaces > Edit (for loopback.1) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 1.1.4.5
Netmask: 255.255.255.255
Host IP Address: 10.1.1.5
Host Virtual Router Name: trust-vr

5. **Blocking**

Network > Zones > Edit (for Untrust): Enter the following, then click **OK**:

Block Intra-Zone Traffic: (clear)

6. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address:
Address Book Entry: (select), CM-TFTP_Server
Destination Address:
Address Book Entry: (select), Any
Service: SCCP
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: Enable
(DIP on): None (Use Egress Interface IP)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:

Address Book Entry: (select), MIP(1.1.4.5)
Service: SCCP
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 route
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.2.1/24
set interface ethernet3 route
set interface ethernet4 zone untrust
set interface ethernet4 ip 1.1.1.1/24
set interface ethernet4 route
set interface loopback.1 zone untrust
set interface loopback.1 ip 1.1.4.1/24
set interface loopback.1 route
```

2. **Addresses**

```
set address trust cm-tftp_server 10.1.1.5/32
set address untrust phone1 1.1.1.4/32
set address untrust phone2 1.1.2.4/32
```

3. **Loopback Group**

```
set interface ethernet3 loopback-group loopback.1
set interface ethernet4 loopback-group loopback.1
```

4. **MIP**

```
set interface loopback.1 mip 1.1.4.5 host 10.1.1.5
```

5. **Blocking**

```
unset zone untrust block
```

6. **Policies**

```
set policy from trust to untrust cm/tftp_server any sccp nat src permit
set policy from untrust to trust any mip(1.1.4.5) sccp permit
save
```

NOTE: Although, in this example, you unset blocking in the Untrust zone to allow intrazone communication, you can accomplish the same thing by creating the following policy:

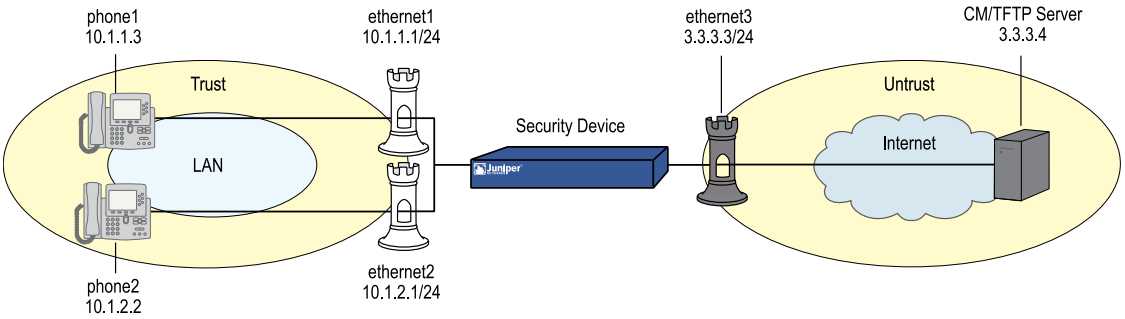
```
set policy from untrust to untrust any any sccp permit
```

Note, also, that it is always more secure to specify a service explicitly, as shown in this example configuration, than to use the keyword any.

Example: Intrazone, Call Manager/TFTP Server in Untrust Zone

In this example, phone1 is on the ethernet1 interface in the Trust zone, phone 2 is on the ethernet2 interface in a subnet in the Trust zone, and the Call Manager/TFTP Server is on the ethernet3 interface in the Untrust zone. After configuring interfaces and addresses, you create a policy from Trust to Untrust to allow phone1 and phone2 to register with the Call Manager/TFTP Server in the Untrust zone. Blocking is off by default in the Trust zone (as it is in custom zones you define), so it is not necessary to create. However, for greater security, you could optionally turn blocking off, and create a policy from Trust to Trust. This would allow you to specify the SCCP service, and restrict intrazone calls to phone1 and phone2.

Figure 313: Intrazone, Call Manager/TFTP Server in Trust Zone



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone: Trust
   Static IP: (select when this option is present)
   IP Address/Netmask: 10.1.1.1/24
   Enter the following, then click OK:
   Interface Mode: route

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **Apply**:

Zone: Trust
Static IP: (select when this option is present)
IP Address/Netmask: 10.1.2.1/24
Enter the following, then click **OK:**
Interface Mode: route
Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 3.3.3.3/24

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone1
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.3/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone2
IP Address/Domain Name:
IP/Netmask: (select), 10.1.2.2/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: CM/TFTP Server
IP Address/Domain Name:
IP/Netmask: (select), 3.3.3.4/24
Zone: Untrust

3. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click OK:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), CM/TFTP Server
Service: SCCP
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: Enable
(DIP on): None (Use Egress Interface IP)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet2 zone trust
set interface ethernet2 ip 10.1.2.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 3.3.3.3/24
set interface ethernet3 route
```

2. **Addresses**

```
set address trust phone1 10.1.1.3/24
set address trust phone2 10.1.2.2/24
set address untrust cm-tftp_server 3.3.3.4/24
```

3. **Policies**

```
set policy from trust to untrust any cm-tftp_server sccp nat src permit
save
```

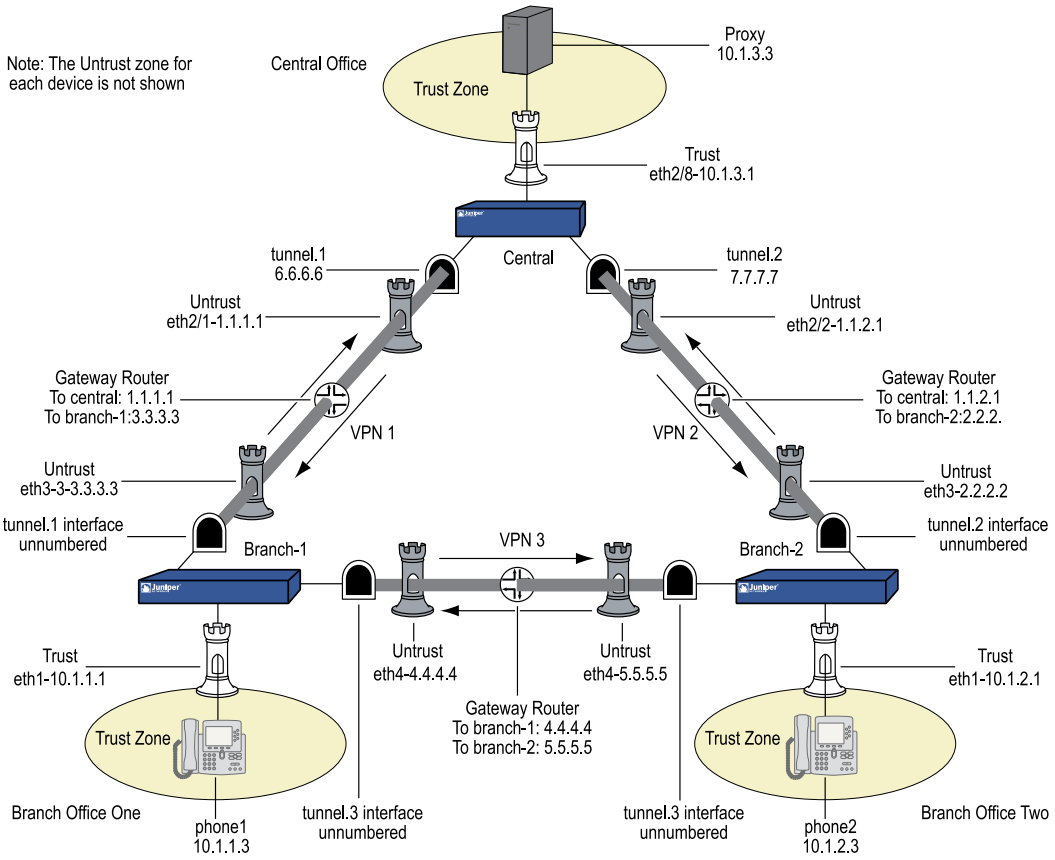
NOTE: It is always more secure to specify a service explicitly, as shown in this example configuration, than to use the keyword **any**.

Example: Full-Mesh VPN for SCCP

In this example, the central office and two branch offices are linked by a full-mesh VPN. Each site has a single security device. The Call Manager/TFTP Server is in the Trust zone at the Central Office, phone1 is in the Trust zone at Branch Office One, and phone2 is in the Trust zone at Branch Office Two. All interfaces connecting the devices are in their respective Untrust zones. On each device, you configure two tunnels, one to each of the other devices, to create a fully meshed network.

NOTE: The security devices used in this example must have at least three independently configurable interfaces available.

Figure 314: Full-Mesh VPN for SCCP



NOTE: It is always more secure to explicitly specify a service, as shown in this example configuration, than to use the keyword **any**.

WebUI (for Central)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2/1): Enter the following, then click **Apply**:

   Zone: Untrust
   Static IP: (select when this option is present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > Edit (for ethernet2/2): Enter the following, then click **Apply**:

   Zone: Untrust
   Static IP: (select when this option is present)
   IP Address/Netmask: 1.1.2.1/24

Network > Interfaces > Edit (for ethernet2/8): Enter the following, then click **Apply**:

Zone: Trust
Static IP: (select when this option is present)
IP Address/Netmask: 10.1.3.1/24
Enter the following, then click OK:
Interface mode: route

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

Tunnel Interface Name: 1
Zone (VR): Untrust
IP Address / Netmask: 6.6.6.6/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

Tunnel Interface Name: 2
Zone (VR): Untrust
IP Address / Netmask: 7.7.7.7/24

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: CM/TFTP Server
IPv4/Netmask: 10.1.3.3/32
Zone: Trust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: to-branch-1
Security Level: Standard
IPvc4/v6 Address/Hostname: 3.3.3.3
Preshare Key: netscreen
Outgoing Interface: ethernet2/1

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn-branch-1

Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Bind to: (select) Tunnel Interface, tunnel.1

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: to-branch-2
Security Level: Standard
IPvc4/v6 Address/Hostname: 2.2.2.2

Preshare Key: netscreen
Outgoing Interface: ethernet2/2

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn-branch-2

Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

4. **Routing**

Network > Routing > Destination > New: Enter the following, then click **OK:**

Network Address / Netmask: 10.1.1.0/24
Interface (select): tunnel.1

Network > Routing > Destination > New: Enter the following, then click **OK:**

Network Address / Netmask: 10.1.2.0/24
Interface (select): tunnel.2

5. **Policies**

Policies > (From: Trust, To: Untrust) New Enter the following, then click **OK:**

Source Address (select) Address Book Entry: CM/TFTP Server
Destination Address (select) Address Book Entry: Any-IPv4
Service: SCCP
Action: Permit

Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK:**

Source Address (select) Address Book Entry: Any-IPv4
Destination Address (select) Address Book Entry: CM/TFTP Server
Service: SCCP
Action: Permit

CLI (for Central)

1. **Interfaces**

```
set interface ethernet2/1 zone untrust
set interface ethernet2/1 ip 1.1.1.1/24
set interface ethernet2/2 zone untrust
set interface ethernet2/2 ip 1.1.2.1/24
set interface ethernet2/8 zone trust
set interface ethernet2/8 ip 10.1.3.1/24
set interface ethernet2/8 route
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 6.6.6.6/24
set interface tunnel.2 zone untrust
set interface tunnel.2 ip 7.7.7.7/24
```

2. **Address**

```
set address trust cm-tftp_server 10.1.3.3/32
```

3. **VPN**

```
set ike gateway to-branch-1 address 3.3.3.3 main outgoing-interface ethernet2/1
 preshare netscreen sec-level standard
set ike gateway to-branch-2 address 2.2.2.2 main outgoing-interface ethernet2/2
 preshare netscreen sec-level standard
set vpn vpn_branch-1 gateway to-branch-1 no-reply tunnel idletime 0 sec-level
standard
set vpn vpn-branch-1 id 1 bind interface tunnel.1
set vpn vpn-branch-2 gateway to-branch-2 no-reply tunnel idletime 0 sec-level
standard
set vpn vpn-branch-2 id 2 bind interface tunnel.2
```

4. **Routing**

```
set route 10.1.2.0/24 interface tunnel.2
set route 10.1.1.0/24 interface tunnel.1
```

5. **Policies**

```
set policy from trust to untrust cm-tftp_server any sccp permit
set policy from untrust to trust any cm-tftp_server sccp permit
save
```

WebUI (for Branch Office 1)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone: Trust
Static IP: (select when this option is present)
IP Address/Netmask: 10.1.1.1/24
Interface mode: route

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **Apply**:

Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 3.3.3.3/24

Network > Interfaces > Edit (for ethernet4): Enter the following, then click **Apply**:

Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 4.4.4.4/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

Tunnel Interface Name: 2
Zone (VR): Untrust
Unnumbered (select) Interface: ethernet3

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

Tunnel Interface Name: 3
Zone (VR): Untrust
Unnumbered (select) Interface: ethernet4

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: phone1
IPv4/Netmask: 10.1.1.3/32
Zone: V1-Trust

3. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: to-central
Security Level: Standard
IPvc4/v6 Address/Hostname: 1.1.2.1
Preshare Key: netscreen
Outgoing Interface: ethernet3

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn-central

Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Bind to (select): Tunnel Interface, tunnel.1

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: to-ns50
Security Level: Standard
IPvc4/v6 Address/Hostname: 5.5.5.5
Preshare Key: netscreen
Outgoing Interface: ethernet4

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn-ns50

Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Bind to (select): Tunnel Interface, tunnel.3

4. **Routing**

   Network > Routing > Destination > New: Enter the following, then click **OK:**

   Network Address / Netmask: 10.1.2.0/24
   Interface (select): tunnel.3

   Network > Routing > Destination > New: Enter the following, then click **OK:**

   Network Address / Netmask: 10.1.3.0/24
   Interface (select): tunnel.1

5. **Policies**

   Policies > (From: Trust, To: Untrust) > New: Enter the following, then click **OK:**

   Source Address (select) Address Book Entry: phone2
   Destination Address (select) Address Book Entry: Any-IPv4
   Service: SCCP
   Action: Permit

   Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK:**

   Source Address (select) Address Book Entry: Any-IPv4
   Destination Address (select) Address Book Entry: phone2
   Service: SCCP
   Action: Permit

CLI (for Branch Office 1)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 route
set interface ethernet3 zone untrust
set interface ethernet3 ip 3.3.3.3/24
set interface ethernet4 zone untrust
set interface ethernet4 ip 4.4.4.4/24
set interface tunnel.2 zone untrust
set interface tunnel.2 ip unnumbered interface ethernet3
set interface tunnel.3 zone untrust
set interface tunnel.3 ip unnumbered interface ethernet4
```

2. **Address**

```
set address trust phone1 10.1.1.3/32
```

3. **VPN**

```
set ike gateway to-central address 1.1.1.1 main outgoing-interface ethernet3
preshare netscreen sec-level standard
set ike gateway to-ns50 address 5.5.5.5 main outgoing-interface ethernet4
preshare netscreen sec-level standard
set vpn vpncentral gateway to-central no-replay tunnel idletime 0 sec-level
```

```
standard
set vpn vpncentral bind interface tunnel.1
set vpn vpn-ns50 gateway to-ns50 no-replay tunnel idletime 0 sec-level standard
set vpn vpn-ns50 bind interface tunnel.3
```

4. **Routes**

```
set route 10.1.2.0/24 interface tunnel.3
set route 10.1.3.0/24 interface tunnel.1
```

5. **Policies**

```
set policy from trust to untrust phone1 any sccp permit
set policy from untrust to trust any phone1 sccp permit
save
```

WebUI (for Branch Office 2)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

```
Zone: Trust
Static IP: (select when this option is present)
IP Address/Netmask: 10.1.2.1/24
Enter the following, then click OK:
Interface mode: route
```

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **Apply**:

```
Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 2.2.2.2/24
```

Network > Interfaces > Edit (for ethernet4): Enter the following, then click **Apply**:

```
Zone: Untrust
Static IP: (select when this option is present)
IP Address/Netmask: 4.4.4.4/24
```

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

```
Tunnel Interface Name: 2
Zone (VR): Untrust
Unnumbered (select) Interface: ethernet3
```

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

```
Tunnel Interface Name: 3
Zone (VR): Untrust
Unnumbered (select) Interface: ethernet4
```

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK:**

   Address Name: phone2
   IPv4/Netmask: 10.1.2.3/32
   Zone: Trust

3. **VPN**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: to-central
   Security Level: Standard
   IPvc4/v6 Address/Hostname: 1.1.2.1
   Preshare Key: netscreen
   Outgoing Interface: ethernet3

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: vpn-central

   Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

   Bind to (select): Tunnel Interface, tunnel.2

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: to-ns50
   Security Level: Standard
   IPvc4/v6 Address/Hostname: 4.4.4.4
   Preshare Key: netscreen
   Outgoing Interface: ethernet4

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: vpn-ns50

   Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

   Bind to (select): Tunnel Interface, tunnel.3

4. **Routing**

   Network > Routing > Destination > New: Enter the following, then click **OK:**

   Network Address / Netmask: 10.1.3.0/24
   Interface (select): tunnel.2

   Network > Routing > Destination > New: Enter the following, then click **OK:**

Network Address / Netmask: 10.1.1.0/24
Interface (select): tunnel.3

5. **Policies**

Policies > (From: Trust, To: Untrust) New Enter the following, then click **OK:**

Source Address (select) Address Book Entry: phone2
Destination Address (select) Address Book Entry: Any-IPv4
Service: SCCP
Action: Permit

Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK:**

Source Address (select) Address Book Entry: Any-IPv4
Destination Address (select) Address Book Entry: phone2
Service: SCCP
Action: Permit

CLI (for Branch Office 2)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.2.1/24
set interface ethernet1 route
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface ethernet4 zone untrust
set interface ethernet4 ip 4.4.4.4/24
set interface tunnel.2 zone untrust
set interface tunnel.2 ip unnumbered interface ethernet3
set interface tunnel.3 zone untrust
set interface tunnel.3 ip unnumbered interface ethernet4
```

2. **Address**

```
set address trust phone1 10.1.2.3/32
```

3. **VPN**

```
set ike gateway to-central address 1.1.1.1 Main outgoing-interface ethernet3
preshare netscreen sec-level standard
set ike gateway to-ns50 address 4.4.4.4 Main outgoing-interface ethernet4
preshare netscreen sec-level standard
set vpn vpncentral gateway to-central no-replay tunnel idletime 0 sec-level
standard
set vpn vpncentral id 4 bind interface tunnel.2
set vpn vpn-ns50 gateway to-ns50 no-replay tunnel idletime 0 sec-level standard
set vpn vpn-ns50 id 5 bind interface tunnel.3
```

4. **Routes**

```
set route 10.1.3.0/24 interface tunnel.1
set route 10.1.2.0/24 interface tunnel.3
```

5. **Policies**

```
set policy from trust to untrust phone2 any sccp permit
set policy from untrust to trust any phone2 sccp permit
save
```

Chapter 31

Apple iChat Application Layer Gateway

This chapter describes the Apple iChat application and provides examples for configuring the AppleiChat Application Layer Gateway (ALG) on a Juniper Networks security device. It contains the following sections:

Overview

Apple iChat is an Instant Messaging (IM) application that lets you chat with other iChat, Mac, or AOL Instant Messenger (AIM) users over the Internet using text, audio, or video. ScreenOS currently supports iChat applications up to version 3.15.

The iChat application uses standard ports to send data to its servers and clients. The AppleiChat ALG provides support for iChat applications by opening pinholes on Juniper Networks security device, thereby allowing the text, audio, and video calls to pass through the security device. Without the AppleiChat ALG, the ports are blocked and need to be opened manually, which exposes the network to attack on these ports.

Table 80 on page 1203 shows the standard ports iChat uses for various services.

Table 80: Standard iChat Service Ports

Port Number	Service Name	Protocol	Used For
5190	AOL	TCP	iChat and AOL instant messenger, file transfer
5678	SNATMAP server	UDP	Determining the external Internet addresses of hosts.
5060	Session Initiation Protocol (SIP)	UDP/TCP	Initiating audio/video (AV) chat invitations.
16384 16403	Real-Time Transport Protocol (RTP) /Real-Time Control Protocol (RTCP)	UDP	iChat audio RTP/RTCP video RTP/RTCP

For a list of well-known ports, see http://docs.info.apple.com/article.html?artnum=106439

The iChat service uses the AOL and SIP protocols for its audio/video operations. It uses the AIM protocol to connect to servers. SIP is used for setting audio/video sessions between IM clients after they successfully negotiate ports. The SIP ALG creates pinholes for audio/video sessions. SIP is a predefined service in ScreenOS and uses port 5060 as the destination port. During iChat operation, the security device creates separate sessions for AOL and SIP.

NOTE: The ALG does not open all ports when you enable the AppleiChat ALG on the security device. ALG opens pinholes only for the ports that are exchanged during iChat signaling messages.

The number of iChat sessions that the security device can handle is limited to the maximum number of Network Address Translation (NAT) cookies available for that particular security device.

NOTE: The NAT cookies available for a security device are shared by other ALGs like H.323 and P2P ALG.

You can view the maximum number of NAT cookies available for a particular device using the following CLI command:

```
get nat cookie
```

For information about running iChat in NAT mode, see http://docs.info.apple.com/article.html?artnum=93208

Configuring the AppleiChat ALG

You configure the AppleiChat ALG with the WebUI or the CLI.

WebUI

Security>ALG>Apple iChat. Select the following, then click **Apply:**
AppleiChat Enable (select)

CLI

```
set alg appleichat enable
```

When you enable the AppleiChat ALG functionality, the security device opens pinholes for the configured call-answer-time to establish the iChat audio/video session. The call-answer-time is the duration of time for which the security device opens the pinholes for establishing iChat audio/video session. The default value of call-answer-time is 32 seconds. When this timer expires, the device closes the pinholes. The range for configuring the call-answer-time is 20 to 90 seconds.

To configure a call-answer-time of 30 seconds:

WebUI

Security > ALG > AppleiChat. Enter the following, then click **Apply:**

Call-Answer-Time: 30

CLI

```
set alg appleichat call-answer-time 30
```

The iChat application fragments the packets it sends to the receiver based on the maximum segment size (MSS) of the receiver. The MSS is the maximum amount of data, in bytes, a device can receive as a single unfragmented frame. The MSS value depends on the network configuration of the receiver. The fragmented packet is reassembled at the ALG for address translation. By default, the reassembly option is disabled. You can enable reassembly with the WebUI or the CLI.

WebUI

Security > ALG > AppleiChat. Select the following, then click **Apply:**

Re-Assembly Enable (select)

CLI

```
set alg appleichat reassembly enable
```

Configuration Examples

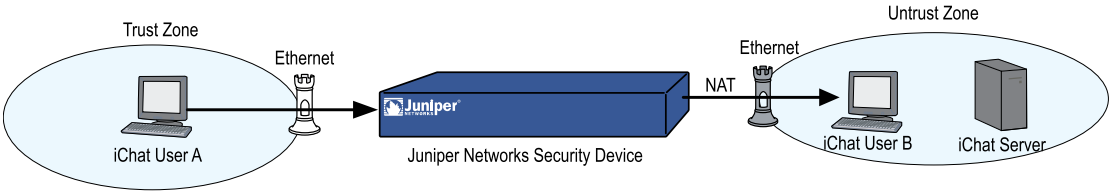
This section includes the following configuration scenarios:

- One iChat user on a private network, another iChat user on a public network, and an iChat server on a public network
- An intra-zone call between two iChat users within a private network
- Users across different firewalls

Scenario 1: Private–Public Network

In Figure 315 on page 1206, one iChat user is on a private network, another iChat user is on a public network, and the iChat server is on public network. There is a NAT between the private and the public network.

Figure 315: AppleiChat Scenario 1—Users on Public and Private Networks



NOTE: Because the administrator does not know the IP address details initially, we recommend that the user put **"ANY"** in the destination address field of the policy.

WebUI

1. **Configuration for Logging into the Server in NAT Mode**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click OK:

   Source Address
   Address Book Entry: (select), iChat UserA
   Destination Address
   Address Book Entry: (select), ANY
   Service: (select) AppleiChat
   Action: Permit

   > **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   NAT:
   Source Translation: (select)
   (DIP on): (select)

2. **Configuration for File Transfer from iChat UserA to iChat UserB in NAT Mode**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click OK:

   Source Address
   Address Book Entry: (select), iChat UserA
   Destination Address
   Address Book Entry: (select), iChatserver_IP_range
   Service: (select) AppleiChat
   Action: Permit

   > **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   NAT:
   Source Translation: (select)
   (DIP on): (select)

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click OK:

Source Address
Address Book Entry: (select), iChat UserB
Destination Address
Address Book Entry: (select), ANY
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

3. **Configuration for Making Audio/Video Calls from iChat UserB in NAT Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
Address Book Entry: (select), iChat UserA
Destination Address
Address Book Entry: (select), iChatserver_IP_range
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
Address Book Entry: (select), iChat UserA
Destination Address
Address Book Entry: (select), iChat UserB
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

4. **Configuration for Making Audio/Video Calls from iChat UserB in Route Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
Address Book Entry: (select), iChat UserA
Destination Address
Address Book Entry: (select), iChatserver_IP_range

Service: (select) AppleiChat
Action: Permit

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
  Address Book Entry: (select), iChat UserA
Destination Address
  Address Book Entry: (select), iChat UserB
Service: (select) AppleiChat
Action: Permit

5. **Configuration for Making Audio/Video Calls from iChat UserA in NAT Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
  Address Book Entry: (select), iChat UserA
Destination Address
  Address Book Entry: (select), iChatserver_IP_range
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
  Address Book Entry: (select), iChat UserA
Destination Address
  Address Book Entry: (select), iChat UserB
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

6. **Configuration for Making Audio/Video Calls from iChat UserA in Route Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
  Address Book Entry: (select), iChat UserA
Destination Address
  Address Book Entry: (select), iChatserver_IP_range
Service: (select) AppleiChat
Action: Permit

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
  Address Book Entry: (select), iChat UserA
Destination Address
  Address Book Entry: (select), iChat UserB
Service: (select) AppleiChat
Action: Permit

CLI

1. **Configuration for Logging into the Server in NAT Mode**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatserver_IP_range" apple-ichat
 nat src permit
```

NOTE: Policies for route/transparent mode are same except the "nat src" option in policy.

2. **Configuration for File Transfer from iChat UserA to iChat UserB in NAT Mode**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatserver_IP_range" apple-ichat
 nat src permit
set policy from trust to untrust "ichatUserB" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserB" "iChatserver_IP_range" apple-ichat
 nat src permit
```

3. **Configuration for Making Audio/Video Calls from iChat UserB in NAT Mode**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatserver_IP_range" apple-ichat
 nat src permit
set policy from trust to untrust "iChatUserA" "iChatUserB" apple-ichat nat src
permit
```

4. **Configuration for Making Audio/Video Calls from iChat UserB in Route Mode**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatserver_IP_range" apple-ichat
 permit
set policy from trust to untrust "iChatUserA" "iChatuserB" apple-ichat permit
```

5. **Configuration for Making Audio/Video Calls from iChat UserA in NAT Mode**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatserver_IP_range" apple-ichat
 nat src permit
set policy from trust to untrust "ichatuserA" "iChatUserB" apple-ichat nat src
permit
```

6. **Configuration for Making Audio/Video Calls from iChat UserA in Route Mode**

```
set policy from trust to untrust "ichatUserA" "ANY"  permit
```

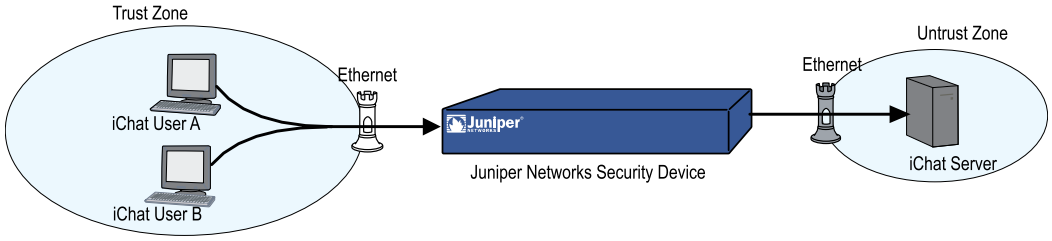
OR

```
set policy from trust to untrust "ichatUserA" "iChatserver_IP_range" apple-ichat
 permit
set policy from trust to untrust "ichatuserA" "iChatUserB" apple-ichat permit
```

Scenario 2: Intrazone Call Within Private Network

In the example shown in Figure 316 on page 1210, iChat userA and iChat userB are in the same network and behind a firewall. The iChat server is in public network. There is a NAT between the private and the public networks.

Figure 316: AppleiChat Scenario 2—Intrazone Call Within a Private Network



WebUI

1. **Configuring iChat userA to Log In iChat server in NAT Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

```
Source Address
   Address Book Entry: (select), iChat UserA
Destination Address
```

Address Book Entry: (select), any
Service: AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

2. **Configuration for File Transfer from iChat UserA to iChat UserB**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
Address Book Entry: (select), iChat UserA
Destination Address
Address Book Entry: (select), iChatserver_IP_range
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
Address Book Entry: (select), iChat UserB
Destination Address
Address Book Entry: (select), ANY
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

3. **Configuration for Making Audio/Video Calls from iChat UserA in NAT Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
Address Book Entry: (select), iChat UserA
Destination Address
Address Book Entry: (select), ANY
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

4. **Configuration for Making Audio/Video Calls from iChat UserA in Route Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
  Address Book Entry: (select), iChat UserA
Destination Address
  Address Book Entry: (select), ichatServer
Service: (select) AppleiChat
Action: Permit

5. **Configuration for Making Audio/Video Calls from iChat UserB in NAT Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
  Address Book Entry: (select), iChat UserB
Destination Address
  Address Book Entry: (select), iChatServer
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

6. **Configuration for Making Audio/Video Calls from iChat UserB in Route Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
  Address Book Entry: (select), iChat UserB
Destination Address
  Address Book Entry: (select), iChatServer
Service: (select) AppleiChat
Action: Permit

CLI

1. **Configuring iChat UserA to Log Into iChat Server in NAT Mode**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatServer_IP_range" apple-ichat
 nat src permit
```

2. **Configuration for File Transfer Between UserA and UserB**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatServer_IP_range" apple-ichat
 nat src permit
set policy from trust to untrust "ichatUserB" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserB" "iChatServer_IP_range" apple-ichat
 nat src permit
```

3. **Configuration for Making Audio/Video Calls from iChat UserA in NAT Mode**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatserver_IP_range" apple-ichat
 nat src permit
```

4. **Configuration for Making Audio/Video Calls from iChat UserA in Route Mode**

```
set policy from trust to untrust "ichatUserA" "iChatserver_IP_range" apple-ichat
 permit
```

5. **Configuration for Making Audio/Video Calls from iChat UserB in NAT Mode**

```
set policy from trust to untrust "ichatUserB" "iChatserver_IP_range" apple-ichat
 nat src permit
```

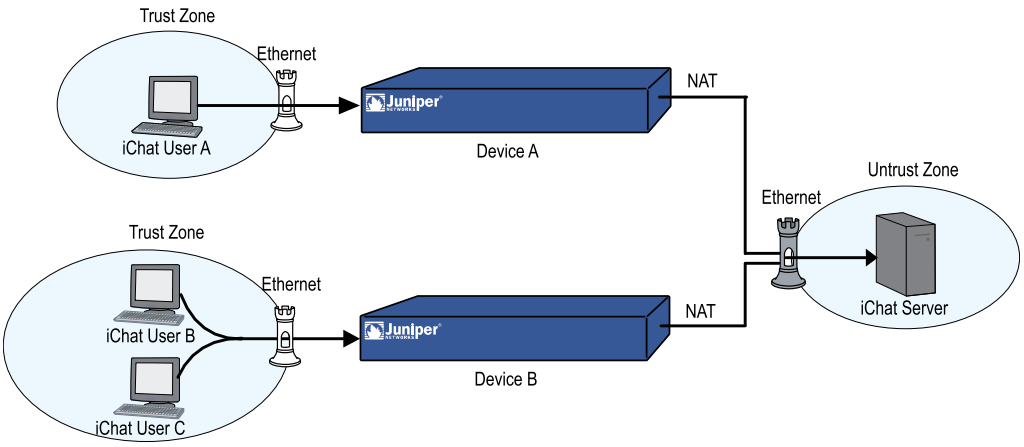
6. **Configuration for Making Audio/Video Calls from iChat UserB in Route Mode**

```
set policy from trust to untrust "ichatUserB" "iChatserver_IP_range" apple-ichat
 permit
```

Scenario 3: Users Across Different Networks

In Figure 317 on page 1214, iChat userA is on a private network and iChat userB and userC are on another private network. The iChat server is on a public network. There is NAT between private networks and the public network.

Figure 317: AppleiChat Scenario 3—Users Across Different Networks



WebUI

1. **Configuration on Firewall 1 for Login from iChat UserA in NAT Mode**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

   Source Address
   Address Book Entry: (select), iChat UserA
   Destination Address
   Address Book Entry: (select), any
   Service: (select) AppleiChat
   Action: Permit

   > **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   NAT:
   Source Translation: (select)
   (DIP on): (select)

2. **Configuration on Firewall 1 for File Transfer from iChat UserA to iChat UserB in NAT Mode**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

   Source Address
   Address Book Entry: (select), iChat UserA
   Destination Address
   Address Book Entry: (select), iChatserver_IP_range
   Service: (select) AppleiChat
   Action: Permit

   > **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

3. **Configuration on Firewall 1 for Making Audio/Video Calls from iChat UserA in NAT Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
  Address Book Entry: (select), iChat UserA
Destination Address
  Address Book Entry: (select), iChatserver_IP_range
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
  Address Book Entry: (select), iChat UserA
Destination Address
  Address Book Entry: (select), iChat UserB
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

4. **Configuration on Firewall 1 for Making Audio/Video Calls from iChat UserA in Route Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
  Address Book Entry: (select), iChat UserA
Destination Address
  Address Book Entry: (select), iChatserver_IP_range
Service:(select) AppleiChat
Action: Permit

5. **Configuration on Firewall 2 for Making Audio/Video Calls from iChat UserB in NAT Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
Address Book Entry: (select), iChat UserB
Destination Address
Address Book Entry: (select), iChat server
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
Address Book Entry: (select), iChat UserB
Destination Address
Address Book Entry: (select), ichatUserA_public
Service: (select) AppleiChat
Action: Permit

> **Advanced:** Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): (select)

6. **Configuration on Firewall 2 for Making Audio/Video Calls from iChat UserB in Route Mode**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
Address Book Entry: (select), iChat UserB
Destination Address
Address Book Entry: (select), iChatserver_IP_range
Service:(select) AppleiChat
Action: Permit

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK:**

Source Address
Address Book Entry: (select), iChat UserB
Destination Address
Address Book Entry: (select), iChatserver_IP_range
Service:(select) AppleiChat
Action: Permit

CLI

1. **Configuration on Firewall 1 for Login from iChat UserA in NAT Mode**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatServer_IP_range" apple-ichat
 nat src permit
```

2. **Configuration on Firewall 1 for File Transfer from iChat UserA to iChat UserB in NAT Mode**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatServer_IP_range" apple-ichat
 nat src permit
```

3. **Configuration on Firewall 1 for Making Audio/Video calls from iChat UserA in NAT mode**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatserver_IP_range" apple-ichat
 nat src permit
set policy from trust to untrust "iChatuserA" "iChatuserB_public"  apple-ichat
nat src permit
```

4. **Configuration on Firewall 1 for Making Audio/Video calls from iChat UserA in Route Mode**

```
set policy from trust to untrust "ichatUserA" "ANY" apple-ichat permit
```

OR

```
set policy from trust to untrust "ichatUserA" "iChatserver_IP_range" apple-ichat
 permit
set policy from trust to untrust "iChatUserA" "iChatuserB_public" apple-ichat
permit
```

5. **Configuration on Firewall 2 for Making Audio/Video Calls from iChat UserB in NAT Mode**

```
set policy from trust to untrust "ichatUserB" "ANY" apple-ichat nat src permit
```

OR

```
set policy from trust to untrust "ichatUserB" "iChatserver_IP_range" apple-ichat
 nat src permit
set policy from trust to untrust "iChatUserB" "ichatUserA_public" apple-ichat
nat src permit
```

6. **Configuration on Firewall 2 for Making Audio/Video Calls from iChat UserB in Route Mode**

```
set policy from trust to untrust "ichatUserB" "ANY" apple-ichat permit
```

OR

```
set policy from trust to untrust "ichatUserB" "iChatserver_IP_range" apple-ichat
 permit
set policy from trust to untrust ""iChatUserB" ichatUserA_public" apple-ichat
permit
```

Part 7
Routing

Routing contains the following chapters:

- "Static Routing" on page 1221 explains route tables and how to configure static routes for destination-based routing, Source Interface-Based Routing (SIBR), or source-based routing.
- "Routing" on page 1235 explains how to configure virtual routers on security devices and how to redistribute routing table entries between protocols or between virtual routers.
- "Open Shortest Path First" on page 1269 describes how to configure the OSPF dynamic routing protocol on security devices.
- "Routing Information Protocol" on page 1307 explains how to configure Routing Information Protocol l (RIP).
- "Border Gateway Protocol" on page 1337 explains how to configure Border Gateway Protocol (BGP).
- "Policy-Based Routing" on page 1373 describes policy based routing (PBR). PBR provides a flexible routing mechanism for data forwarding over networks that rely on Application Layer support such as for antivirus (AV), deep inspection (DI), or Web filtering.
- "Multicast Routing" on page 1391 explains multicast routing basics, including how to configure static multicast routes.
- "Internet Group Management Protocol" on page 1399 explains how to configure Internet Group Management Protocol (IGMP).
- "Protocol Independent Multicast" on page 1425 explains how to configure Protocol Independent Multicast-Sparse Mode (PIM-SM) and Protocol Independent Multicast-Source Specific Multicast (PIM-SSM).
- "ICMP Router Discovery Protocol" on page 1461 explains how to set up an Internet Control Message Protocol (ICMP) exchange between a host and a router.

Chapter 32
Static Routing

This chapter discusses static routing and explains when and how to set up static routes. It contains the following sections:

Overview

A static route is a manually configured mapping of an IP network address to a next-hop destination (another router) that you define on a Layer 3 forwarding device, such as a router.

For a network that has few connections to other networks, or for networks where inter-network connections are relatively unchanging, it is usually more efficient to define static routes rather than dynamic routes. ScreenOS retains static routes until you explicitly remove them. However, you can override static routes with dynamic route information if necessary.

You can view static routes in the ScreenOS routing table. To force load-balancing, you can configure Equal Cost Multi-Path (ECMP). To only use active gateways, you can set gateway tracking.

You should set at least a null route as a default route (network address 0.0.0.0/0). A default route is a catch-all entry for packets that are destined for networks other than those defined in the routing table.

How Static Routing Works

When a host sends packets to another host that resides on a different network, each packet header contains the address of the destination host. When a router receives a packet, it compares the destination address to all addresses contained in its routing table. The router selects the most specific route in the routing table to the destination address and, from the selected route entry, determines the next-hop to forward the packet.

NOTE: The most specific route is determined by first performing a bit-wise logical AND of the destination address and network mask for each entry in the routing table. For example, a bit-wise logical AND of the IP address 10.1.1.1 with the subnet mask 255.255.255.0 is 10.1.1.0. The route that has the highest number of bits set to 1 in the subnet mask is the most specific route (also called the "longest matching route").

Figure 318 on page 1222 represents a network that uses static routing and a sample IP packet. In this example, host 1 in network A wants to reach host 2 in network C. The packet to be sent contains the following data in the header:

- Source IP address
- Destination IP address
- Payload (message)

Figure 318: Static Routing Example

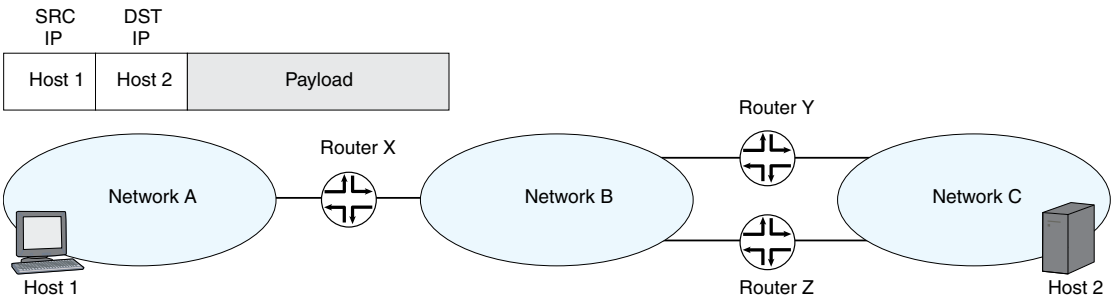


Table 81 on page 1222 summarizes the routing table of each router.

Table 81: Routing Table Summary for Routers X, Y, and Z

Router X		Router Y		Router Z	
Network	**Gateway**	**Network**	**Gateway**	**Network**	**Gateway**
Net A	Connected	Net A	Router X	Net A	Router X
Net B	Connected	Net B	Connected	Net B	Connected
Net C	Router Y	Net C	Connected	Net C	Connected

In Table 81 on page 1222, router X has a static route configured for network C with the gateway (next-hop) as router Y. When router X receives the packet destined for host 2 in network C, it compares the destination address in the packet with its routing table and finds that the last route entry in the table is the most specific route to the destination address. The last route entry specifies to send traffic destined for network C to router Y for delivery. Router Y receives the packet, and, because it knows that

network C is directly connected, it sends the packet through the interface connected to that network.

If router Y fails, or if the link between router Y and network C is unavailable, the packet cannot reach host 2. While there is another route for network C through router Z, that route has not been statically configured on router X, so router X does not detect the alternate route.

When to Configure Static Routes

You need to define at least a few static routes even when using dynamic routing protocols. You need to define static routes for conditions such as the following:

- You need to define a static route to add a default route (0.0.0.0/0) to the routing table for a virtual router (VR). For example, if you are using two VRs on the same security device, the trust-vr routing table could contain a default route that specifies the untrust-vr as the next hop. This allows traffic for destinations that are not in the trust-vr routing table to be routed to the untrust-vr. You can also define a default route in the untrust-vr to route to a specific IP address traffic for destinations not found in the untrust-vr routing table.
- If a network is not directly connected to the security device but is accessible through a router from an interface within a VR, you need to define a static route for the network with the IP address of the router. For example, the Untrust zone interface can be on a subnet with two routers that each connect to different Internet service providers (ISPs). You must define which router to use for forwarding traffic to specific ISPs.
- If you are using two VRs on the same security device, and inbound traffic arrives on an untrust-vr interface that is destined for a network connected to a trust-vr interface, you need to define a static entry in the untrust-vr routing table for the destination network with the trust-vr as the next hop. You can avoid setting a static route in this case by exporting the routes in the trust-vr to the untrust-vr.
- When the device is in transparent mode, you must define static routes that direct management traffic originating from the device itself (as opposed to user traffic traversing the firewall) to remote destinations. For example, you need to define static routes directing syslog, SNMP, and WebTrends messages to a remote administrator's address. You must also define routes that direct authentication requests to the RADIUS, SecurID, and LDAP servers, and URL checks to the Websense server.

NOTE: When the security device is in transparent mode, you must define a static route for management traffic from the device even if the destination is on the same subnet as the device.

- For outbound Virtual Private Network (VPN) traffic where there is more than one outgoing interface to the destination, you need to set a route for directing the outbound traffic through the desired interface to the external router.
- If an interface for a security zone in the trust-vr is NAT, and if you configured a Mapped IP (MIP) or Virtual IP (VIP) on that interface to receive incoming traffic

from a source in the untrust-vr routing domain, then you must create a route to the MIP or VIP in the untrust-vr that points to the trust-vr as the gateway.

- By default, the security device uses destination IP addresses to find the best route on which to forward packets. You can also enable source-based or SIBR tables on a VR. Both source-based and SIBR tables contain static routes that you configure on the VR.

Configuring Static Routes

To configure a static route, you need to define the following:

- Virtual router (VR) to which the route belongs.
- IP address and netmask of the destination network.
- Next hop for the route, which can be either another VR on the security device or a gateway (router) IP address. If you specify another VR, make sure that an entry for the destination network exists in the routing table of that VR.
- The interface through which the routed traffic is forwarded. The interface can be any ScreenOS-supported interface, such as a physical interface (for example, ethernet1/2) or a tunnel interface. You can also specify the Null interface for certain applications. See "Forwarding Traffic to the Null Interface" on page 1231.

Optionally, you can define the following elements:

- Route metric is used to select the active route when there are multiple routes to the same destination network, all with the same preference value. The default metric for static routes is 1.
- Route tag is a value that can be used as a filter when redistributing routes. For example, you can choose to import into a VR only those routes that contain specified tag values.
- Preference value for the route. By default, all static routes have the same preference value, which is set in the VR.
- Whether the route is permanent (kept active even if the forwarding interface is down or the IP address is removed from the interface).

This section contains the following examples:

Setting Static Routes

In Figure 319 on page 1226, a security device operating with its Trust zone interface in Network Address Translation (NAT) mode protects a multilevel network. There is both local and remote management (via Network and Security Manager). The security device sends SNMP traps and syslog reports to the local administrator (located on a network in the Trust zone) and it sends Network and Security Manager (NSM) reports to the remote administrator (located on a network in the Untrust zone). The device

uses a SecurID server in the Demilitarized Zone (DMZ) to authenticate users and a Websense server in the Trust zone to perform Web filtering.

NOTE: The following zones must be bound before this example can be completed: ethernet1 to the Trust zone, ethernet2 to the DMZ zone, and ethernet3 to the Untrust zone. The interface IP addresses are 10.1.1.1/24, 2.2.10.1/24, and 2.2.2.1/24, respectively.

The trust-vr and untrust-vr routing tables must contain routes for the following destinations:

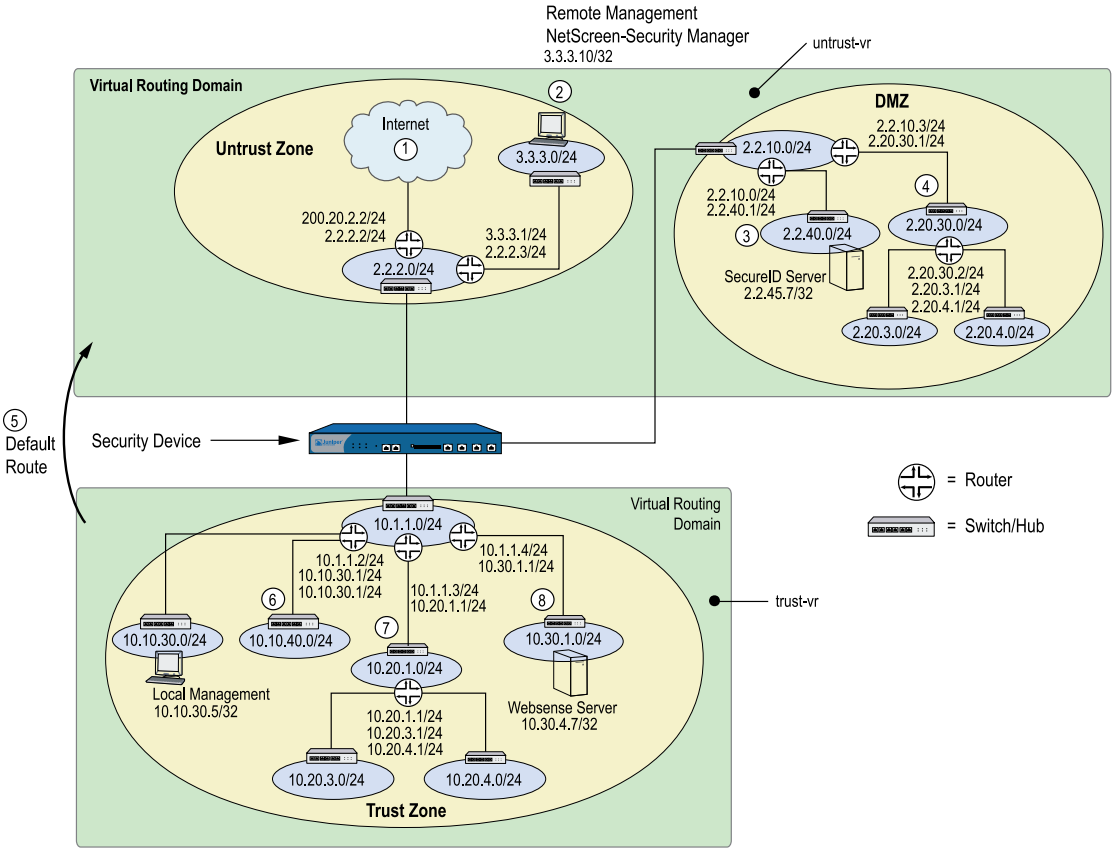
untrust-vr

1. Default gateway to the Internet (default route for the VR)
2. Remote administrator in the 3.3.3.0/24 subnet
3. 2.2.40.0/24 subnet in the DMZ
4. 2.20.0.0/16 subnet in the DMZ

trust-vr

1. untrust-vr for all addresses not found in the trust-vr routing table (default route for the VR)
2. 10.10.0.0/16 subnet in the Trust zone
3. 10.20.0.0/16 subnet in the Trust zone
4. 10.30.1.0/24 subnet in the Trust zone

Figure 319: Static Route Configuration



WebUI

1. **untrust-vr**

   Network > Routing > Destination > untrust-vr New: Enter the following to create the untrust default gateway, then click **OK**:

   Network Address/Netmask: 0.0.0.0/0
   Gateway: (select)
   Interface: ethernet3
   Gateway IP Address: 2.2.2.2

   Network > Routing > Destination > untrust-vr New: Enter the following to direct system reports generated by the security device to remote management, then click **OK**:

   Network Address/Netmask: 3.3.3.0/24
   Gateway: (select)
   Interface: ethernet3
   Gateway IP Address: 2.2.2.3

Network > Routing > Destination > untrust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 2.2.40.0/24
Gateway: (select)
Interface: ethernet2
Gateway IP Address: 2.2.10.2

Network > Routing > Destination > untrust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 2.20.0.0/16
Gateway: (select)
Interface: ethernet2
Gateway IP Address: 2.2.10.3

2. **trust-vr**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Next Hop Virtual Router Name: (select); untrust-vr

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.10.0.0/16
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 10.1.1.2

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.20.0.0/16
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 10.1.1.3

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.30.1.0/24
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 10.1.1.4

NOTE: To remove an entry, click **Remove**. A message appears prompting you to confirm the removal. Click **OK** to proceed or **Cancel** to cancel the action.

CLI

1. **untrust-vr**

```
set vrouter untrust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.2
set vrouter untrust-vr route 3.3.3.0/24 interface ethernet3 gateway 2.2.2.3
set vrouter untrust-vr route 2.2.40.0/24 interface ethernet2 gateway 2.2.10.2
set vrouter untrust-vr route 2.20.0.0/16 interface ethernet2 gateway 2.2.10.3
```

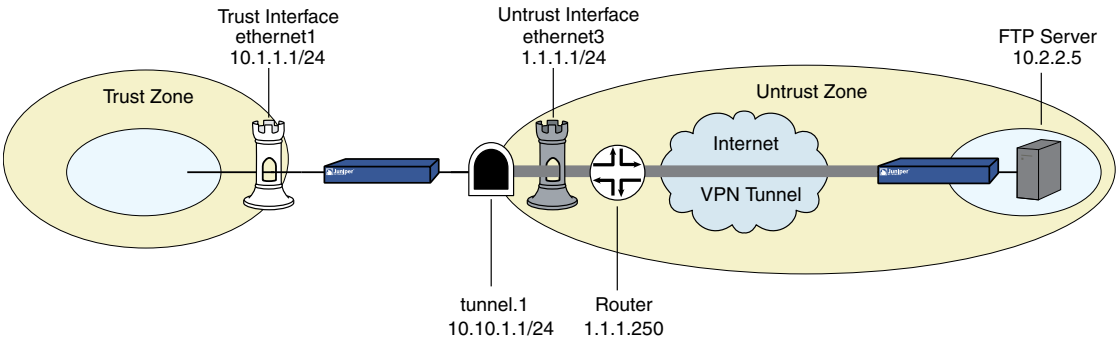
2. **trust-vr**

```
set vrouter trust-vr route 0.0.0.0/0 vrouter untrust-vr
set vrouter trust-vr route 10.10.0.0/16 interface ethernet1 gateway 10.1.1.2
set vrouter trust-vr route 10.20.0.0/16 interface ethernet1 gateway 10.1.1.3
set vrouter trust-vr route 10.30.1.0/24 interface ethernet1 gateway 10.1.1.4
save
```

Setting a Static Route for a Tunnel Interface

In Figure 320 on page 1228, a trusted host resides in a different subnet from the trusted interface. A File Transfer Protocol (FTP) server receives inbound traffic through a VPN tunnel. You need to set a static route to direct traffic exiting the tunnel interface to the internal router leading to the subnet where the server resides.

Figure 320: Static Route for a Tunnel Interface



WebUI

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

```
Network Address/Netmask: 10.2.2.5/32
Gateway: (select)
   Interface: tunnel.1
   Gateway IP Address: 0.0.0.0
```

NOTE: For **tunnel.1** to appear in the Interface drop-down list, you must first create the tunnel.1 interface.

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

CLI

```
set vrouter trust-vr route 10.2.2.5/32 interface tunnel.1
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
save
```

Adding Descriptions to Static Routes

When many static routes are configured on the security device, it becomes cumbersome for the user to identify what traffic routes through the device and to search for a specific route in a route table. To mitigate such difficulties, ScreenOS enables you to add a description to a static route you configure. The description can be 1 to 32 characters in length.

NOTE: You cannot add a description to a dynamic route.

You can add description to a static route through the WebUI or the CLI. In this example, you add a description to a destination-based static route.

WebUI

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.5/32
Gateway: (select)
Interface: ethernet0/0
Gateway IP Address: 2.2.2.250
Description: route_to_office

CLI

```
set vrouter trust-vr route 10.2.2.5/32 interface ethernet 0/0 gateway 2.2.2.250
description "route_to_office"
```

You can use the **get route**, **get vrouter**, and **get config** commands to display the descriptions you have added to static routes.

You cannot use the unset command to unset a description for a static route. To delete the description, you must delete the entire route using the unset command.

In the following example output, the **get route id 4** command shows the description you have added to a destination-based static route with ID 4.

```
device-> get route id 4
route in vr1:
------------------------------------------------------------
ID: 4
IP address/mask: 10.3.2.1/24
next hop (gateway): 0.0.0.0
preference: 20
metric: 1
description: 121555
outgoing interface: ethernet0/0
vsys name/id: Root/0
tag: 0
flag: 24002040/00100081
type: static
Redistributed to:
status: active (for 15 seconds)
------------------------------------------------------------
```

Enabling Gateway Tracking

The security device allows interface-independent static routes with gateways to be tracked for reachability. By default, static routes are not tracked, but you can configure a security device to track reachability for gateway routes. The security device uses the routing table information to determine the reachability of each gateway. The routing table displays the status of the route as active or inactive, depending on the ability of a route to reach the tracked gateway.

To add a static route with gateway tracking, you need to explicitly set the route at the virtual router (VR) level and at the gateway address. The security device tracks only the specified gateway IP and does not depend on the route interface.

NOTE: You can use gateway tracking only to track remote gateway addresses. Gateway tracking cannot be applied for the default gateway address of your local subnet. To enable gateway tracking on a static route, you must use the IP address, not the interface, to identify the gateway.

Use the CLI command given below to add a static route with a tracked gateway for IP address 1.1.1.254 with prefix 1.1.1.0 and a length of 24.

WebUI

Network > Routing > Destination: Click **New**, then enter the following:

IPv4/Netmask: 1.1.1.0/24
Gateway: (select)
Gateway IP Address: 1.1.1.254

CLI

```
set vrouter trust route 1.1.1.0/24 gateway 1.1.1.254
unset vrouter trust route 1.1.1.0/24 gateway 1.1.1.254
save
```

Routes with gateway tracking are only applied locally in the device, so if the device is participating in NSRP cluster it is necessary to manually set the route in the NSRP peer as well.

Forwarding Traffic to the Null Interface

You can configure static routes with the Null interface as the outgoing interface. The Null interface is always active, and traffic destined to the Null interface is always dropped. To make the route to the Null interface a *last resort* route, you need to define the route with a higher metric than other routes. The three purposes for using static routes that forward traffic to the Null interface are as follows:

- Preventing route lookup in other routing tables
- Preventing tunnel traffic from being sent on non-tunnel interfaces
- Preventing traffic loops

Preventing Route Lookup in Other Routing Tables

If SIBR is enabled, the security device by default performs route lookup in the SIBR table. (For information about configuring SIBR, see “Source Interface-Based Routing Table” on page 1241.) If the route is not found in the SIBR table and if source-based routing is enabled, the security device performs route lookup in the source-based routing table. If the route is not found in the source-based routing table, the security device performs route lookup in the destination-based routing table. If you want to prevent route lookup in either the source-based routing table or the destination-based routing table, you can create a default route in the SIBR table with the Null interface as the outgoing interface. Use a higher metric than other routes to ensure that this route is only used if no other source interface-based routes exist that match the route.

Preventing Tunnel Traffic from Being Sent on Non-Tunnel Interfaces

You can use static or dynamic routes with outgoing tunnel interfaces to encrypt traffic to specified destinations. If a tunnel interface becomes inactive, all routes defined on the interface become inactive. If there is an alternate route on a non-tunnel interface, traffic is sent unencrypted. To prevent traffic that is intended to be encrypted from being sent on a non-tunnel interface, define a static route to the same destination as the tunnel traffic with the Null interface as the outgoing interface. Assign this route a higher metric than the tunnel interface route so that the route only becomes active if the tunnel interface route is unavailable. If the tunnel interface becomes inactive, the route with the Null interface becomes active and traffic for the tunnel destination is dropped.

Preventing Loops Created by Summarized Routes

When the security device advertises summarized routes, the device might receive traffic for prefixes that are not in its routing tables. It might then forward the traffic based on its default route. The receiving router might then forward the traffic back to the security device because of the summarized route advertisement. To avoid such loops, you can define a static route for the summarized route prefix with the Null interface as the outgoing interface and a high route metric. If the security device receives traffic for prefixes that are in its summarized route advertisement but not in its routing tables, the traffic is dropped.

In this example, you set a NULL interface for the summarized route that you created to the network 2.1.1.0/24 in the previous example. Within the network 2.1.1.0/24 you have hosts 2.1.1.2, 2.1.1.3, and 2.1.1.4. Any packets addressed to 2.1.1.10 fall into the range for the summarized route. The security device accepts these packets but has nowhere to forward them except back out to the origin and this begins a network loop. To avoid this pattern, you set a NULL interface for this route. Setting a high preference and metric are important when setting a NULL interface.

WebUI

Network > Routing > Destination > trust-vr New: Enter the following and then click **OK**:

Network Address/Netmask: 2.1.1.0/24
Gateway: (Select)
Interface: Null
Gateway IP Address: 0.0.0.0
Preference: 255
Metric: 65535

CLI

```
set vrouter trust-vr route 2.1.1.0/24 interface null preference 255 metric 65535
save
```

Permanently Active Routes

Certain situations exist where you want a route to stay active in a routing table even if the physical interface associated with that route goes down or does not have an assigned IP address. For example, an XAuth server can assign an IP address to an interface on a security device whenever there is traffic that needs to be sent to the server. The route to the XAuth server needs to be kept active even when there is no IP address assigned on the interface so that traffic that is intended for the XAuth server is not dropped.

It is also useful to keep routes active through interfaces on which IP tracking is configured. IP tracking allows the security device to reroute outgoing traffic through a different interface if target IP addresses become unreachable through the original interface. Even though the security device may reroute traffic to another interface,

it still needs to be able to send ping requests on the original interface to determine if the targets become reachable again.

Changing Routing Preference with Equal Cost Multipath

You can also change the routing preference for static routes with Equal Cost Multipath (ECMP). See "Configuring Equal Cost Multipath Routing" on page 1259 for more information.

Chapter 33
Routing

This chapter describes routing and virtual router (VR) management. It contains the following sections:

Overview

Routing is the process of forwarding packets from one network to another toward a final destination. A router is a device that resides where one network meets another network and directs traffic between those networks.

By default, a security device enters the route operational mode and operate as a Layer-3 router. However, you can configure a security device to operate in transparent mode as a Layer 2 switch.

NOTE: For either operational mode, you need to manually configure some routes.

Juniper Networks security devices accomplish routing through a process called a virtual router (VR). A security device divides its routing component into two or more VRs with each VR maintaining its own list of known networks in the form of a routing table, routing logic, and associated security zones. A single VR can support one or more of the following:

- Static or manually configured routes
- Dynamic routes, such as those learned by a dynamic routing protocol
- Multicast routes, such as a route to a group of host machines

Juniper Networks security devices have two predefined VRs:

- trust-vr, which by default contains all the predefined security zones and any user-defined zones
- untrust-vr, which by default does not contain any security zones

You cannot delete the trust-vr or untrust-vr VRs. Multiple VRs can exist, but trust-vr remains the default VR. In the VR table an asterisk (*) designates trust-vr as the default VR in the command line interface (CLI). You can view the VR table with the **get vrouter** CLI command. To configure zones and interfaces within other VRs, you must specify the VR by name, such as untrust-vr. For more information about zones, see "Zones" on page 43.

Some security devices allow you to create additional custom VRs. By separating routing information into multiple VRs, you can control how much routing information is visible to other routing domains. For example, you can keep the routing information for all the security zones inside a corporate network on the predefined VR trust-vr, and the routing information for all the zones outside the corporate network on the other predefined VR untrust-vr. You can keep internal network routing information separate from untrusted sources outside the company because routing table details of one VR are not visible to the other.

Virtual Router Routing Tables

In a security device, each VR maintains its own routing tables. A routing table is an up-to-date list of known networks and directions for reaching them. When a security device processes an incoming packet, it performs a routing table lookup to find the appropriate interface that leads to the destination address.

Each route table entry identifies the destination network to which traffic can be forwarded. The destination network can be an IP network, subnetwork, supernet, or host. Each routing table entry can be unicast (packet sent to single IP address that references a single host machine) or multicast (packet sent to a single IP address that references multiple host machines).

Routing table entries can originate from the following sources:

- Directly connected networks (the destination network is the IP address that you assign to an interface in route mode)
- Dynamic routing protocols, such as Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), or Routing Information Protocol (RIP)
- Other routers or virtual routers in the form of imported routes
- Statically configured routes
- Host routes

NOTE: When you set an IP address for an interface in route mode, the routing table automatically creates a connected route to the adjacent subnet for traffic traversing the interface.

A VR supports three types of routing tables:

- The destination-based routing table allows the security device to perform route lookups based on the destination IP address of an incoming data packet. By default, the security device uses only destination IP addresses to find the best route on which to forward packets.

- The source-based routing table allows the security device to perform route lookups based on the source IP address of an incoming data packet. To add entries to the source-based routing table, you must configure static routes for specific source addresses on which the security device can perform route lookup. This routing table is disabled by default. See “Source-Based Routing Table” on page 1239.
- The SIBR table allows the security device to perform route lookups based on the interface on which a data packet arrives on the device. To add entries to the SIBR table, you must configure static routes for specific interfaces on which the VR performs route lookup. This routing table is disabled by default. See “Source Interface-Based Routing Table” on page 1241.

Destination-Based Routing Table

The destination-based routing table is always present in a VR. Additionally, you can enable source-based or SIBR tables, or both, in a VR. The following is an example of ScreenOS destination-based routing tables:

```
device->  get route

IPv4 Dest-Routes for <untrust-vr> (0 entries)
--------------------------------------------------------------------------------
H: Host C: Connected S: Static A: Auto-Exported
I: Imported R: RIP P: Permanent D: Auto-Discovered
iB: IBGP eB: EBGP O: OSPF E1: OSPF external type 1
E2: OSPF external type 2

IPv4 Dest-Routes for <trust-vr> (11 entries)
--------------------------------------------------------------------------------
   ID          IP-Prefix      Interface         Gateway   P Pref    Mtr     Vsys
--------------------------------------------------------------------------------
*   8          0.0.0.0/0         eth1/1     10.100.37.1   S   20      1     Root
*   7        1.1.1.1/32          eth1/2         0.0.0.0   H    0      0     Root
*   3     192.168.1.1/32            mgt         0.0.0.0   H    0      0     Root
*   2     192.168.1.0/24            mgt         0.0.0.0   C    0      0     Root
*   4     10.100.37.0/24         eth1/1         0.0.0.0   C    0      0     Root
*   5   10.100.37.170/32         eth1/1         0.0.0.0   H    0      0     Root
*   6        1.1.1.0/24          eth1/2         0.0.0.0   C    0      0     Root
*   9       11.3.3.0/24            agg1         0.0.0.0   C    0      0     Root
*  10       11.3.3.0/32            agg1         0.0.0.0   H    0      0     Root
*  11        3.3.3.0/24           tun.1         0.0.0.0   C    0      0     Root
*  12        3.3.3.0/32           tun.1         0.0.0.0   H    0      0     Root
```

For each destination network, the routing table contains the following information:

- The interface on the security device on which traffic for the destination network is forwarded.
- The next-hop, which can be either another VR on the security device or a gateway IP address (usually a router address).
- The protocol from which the route is derived. The protocol column of the routing table allows you know the route type:
  - Connected network (C)
  - Static (S)

  - Auto-exported (A)
  - Imported (I)
  - Dynamic routing protocols, such as RIP (R), Open Shortest Path First or OSPF (O), OSPF external type 1 or type 2 (E1 or E2, respectively), internal or external Border Gateway Protocol (iB or eB, respectively)
  - Permanent (P)
  - Host (H)

    A host-route entry with a 32-bit mask appears when you configure each interface with an IP address. The host route is always active in the route table so that route lookup always succeeds. The host routes automatically update with configured changes, such as interface IP address deletion, and they are never redistributed or exported. Host routes remove the possibility of wandering traffic and conserve processing capability.

- The *preference* is used to select the route to use when there are multiple routes to the same destination network. This value is determined by the protocol or the origin of the route. The lower the preference value of a route, the more likely the route is to be selected as the active route.

  You can modify the preference value for each protocol or route origin on a per-virtual router basis. See “Route Selection” on page 1254 for more information.
- The *metric* can also be used to select the route to use when there are multiple routes for the same destination network with the same preference value. The metric value for connected routes is always 0. The default metric value for static routes is 1, but you can specify a different value when defining a static route.
- The virtual system (vsys) to which this route belongs. For more information about virtual routers and vsys, see “Virtual Routers and Virtual Systems” on page 1250. In this example, no entries appear under the untrust-vr table header; eleven entries appear under the trust-vr table header.
- The *description* added to a route.

Most routing tables include a *default route* (network address 0.0.0.0/0), which is a catch-all entry for packets that are destined for networks other than those defined in the routing table.

For an example of destination based routing, see “Configuring Static Routes” on page 1224.

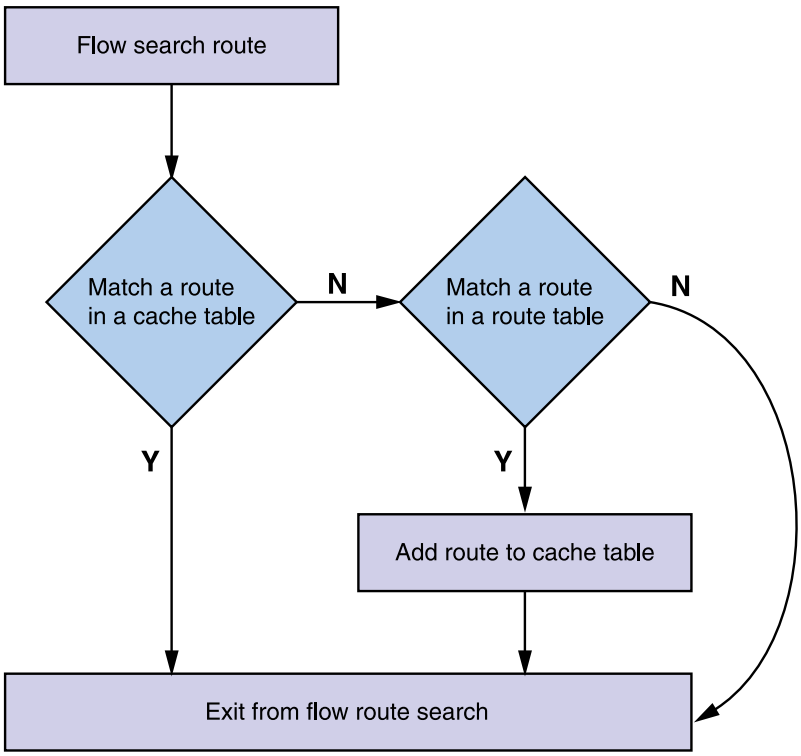
Route-cache

For every packet with an identical destination IP address, the device searches the route and ARP tables to get the same route and ARP entry, even if no route and ARP change happens. In order to avoid this, ScreenOS provides the feature of caching the route and ARP entries for a specific destination - IP. Users can enable this feature by using the CLI command:

```
set flow route-cache
```

This command enables the device to cache recently used route and ARP entries. When route-cache is enabled, the device first searches the cache table for identical matches. Only if it does not find a matching entry in the cache route, it searches the routing table as shown in Figure 321 on page 1239

Figure 321: Route-cache



The following points needs to be considered when enabling this feature:

- Route-cache works only for destination routes.
- Route-cache table does not work if any of the following features is enabled:
  - Source-based routing
  - Source interface-based routing
  - Equal cost multipath (ECMP) routing

Source-Based Routing Table

You can direct the security device to forward traffic based on the source IP address of a data packet instead of the destination IP address. This feature allows traffic from users on a specific subnet to be forwarded on one path while traffic from users on a different subnet is forwarded on another path. When source-based routing is enabled in a VR, the security device performs routing table lookup on the packet's source IP address in a source-based routing table. If the security device does not find a route for the source IP address in the source-based routing table, then the device uses the

packet's destination IP address for route lookup in the destination-based routing table.

You define source-based routes as statically configured routes on specified VRs. Source-based routes apply to the VR in which you configure them, but you can specify another VR as the next hop for a source-based route. You cannot, however, redistribute source-based routes into another VR or into a routing protocol.

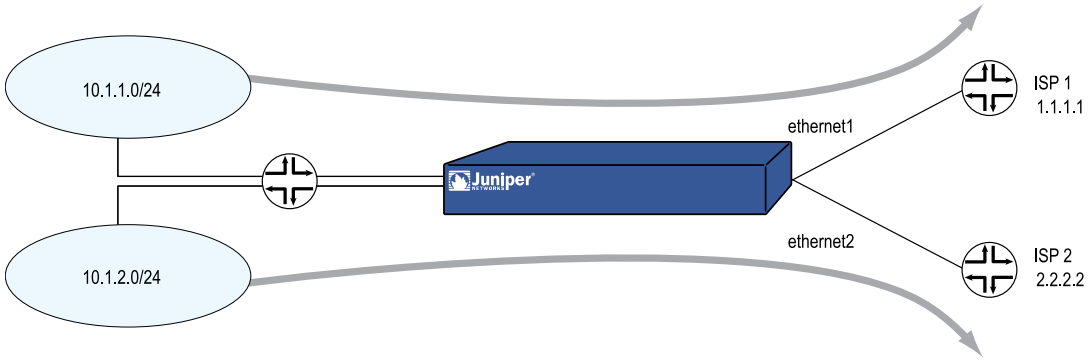
To use this feature:

1. Create one or more source-based routes by specifying the following information:
   - The name of the VR in which source-based routing applies.
   - The source IP address, which appears as an entry in the source-based routing table, on which the security device performs a routing table lookup.
   - The name of the outgoing interface on which the packet is forwarded.
   - The next-hop for the source-based route (If you have already specified a default gateway for the interface with the CLI **set interface** *interface* **gateway** *ip_addr* command, you do not need to specify the gateway parameter; the interface's default gateway is used as the next hop for the source-based route. You can also specify another VR as the next-hop for the source-based route with the **set vrouter** *vrouter* **route source** *ip_addr/netmask* **vrouter** *next-hop_vrouter*.)
   - The metric for the source-based route. (If there are multiple source-based routes with the same prefix, only the route with the lowest metric is used for route lookup; other routes with the same prefix are marked as "inactive.")
   - Optional description for the source-based route.
2. Enable source-based routing for the VR. The security device uses the source IP of the packet for route lookup in the source-based routing table. If no route is found for the source IP address, the destination IP address is used for the routing table lookup.

In Figure 322 on page 1241, traffic from users on the 10.1.1.0/24 subnetwork is forwarded to ISP 1, while traffic from users on the 10.1.2.0/24 subnetwork is forwarded to ISP 2. This configuration requires two entries in the default trust-vr VR routing table and enables source-based routing:

- The subnetwork 10.1.1.0/24, with ethernet3 as the forwarding interface, and ISP 1's router (1.1.1.1) as the next-hop
- The subnetwork 10.1.2.0/24, with ethernet4 as the forwarding interface, and ISP 2's router (2.2.2.2) as the next-hop

Figure 322: Source-Based Routing Example



WebUI

Network > Routing > Source Routing > New (for trust-vr): Enter the following, then click **OK**:

Network Address/Netmask: 10.1.1.0 255.255.255.0
Interface: ethernet3 (select)
Gateway IP Address: 1.1.1.1

Network > Routing > Source Routing > New (for trust-vr): Enter the following, then click **OK**:

Network Address/Netmask: 10.1.2.0 255.255.255.0
Interface: ethernet4 (select)
Gateway IP Address: 2.2.2.2

NOTE: In the WebUI, the default preference and metric value are 1.

Network > Routing > Virtual Routers > Edit (for trust-vr): Select **Enable Source Based Routing**, then click **OK**.

CLI

```
set vrouter trust-vr route source 10.1.1.0/24 interface ethernet3 gateway 1.1.1.1
metric 1
set vrouter trust-vr route source 10.1.2.0/24 interface ethernet4 gateway 2.2.2.2
metric 1
set vrouter trust-vr source-routing enable
save
```

Source Interface-Based Routing Table

Source interface-based routing (SIBR) allows the security device to forward traffic based on the source interface (the interface on which the data packet arrives on the security device). When SIBR is enabled in a virtual router (VR), the security device performs route lookup in an SIBR routing table. If the security device does not find

a route entry in the SIBR routing table for the source interface, it can perform route lookup in the source-based routing table (if source-based routing is enabled in the VR) or the destination-based routing table.

You define source interface-based routes as static routes for specified source interfaces. Source interface-based routes apply to the VR in which you configure them, but you can also specify another VR as the next hop for a source interface-based route. You cannot, however, export source interface-based routes into another VR or redistribute them into a routing protocol.

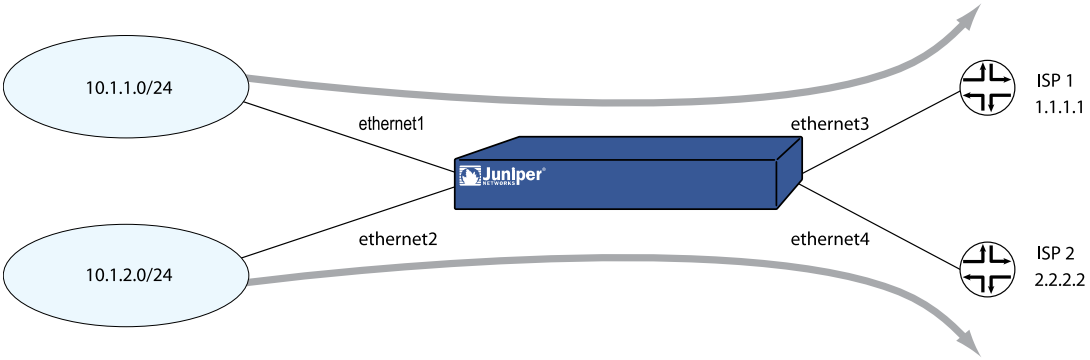
To use this feature:

1. Create one or more source interface-based routes by specifying the following information:
   - The name of the VR in which SIBR applies.
   - The source interface on which the security device performs a lookup in the SIBR table. (The interface appears as an entry in the routing table.)
   - The IP address and netmask prefix for the route.
   - The name of the outgoing interface on which the packet is forwarded.
   - The next-hop for the source interface-based route. (If you have already specified a default gateway for the interface with the CLI **set interface** *interface* **gateway** *ip_addr* command, you do not need to specify the gateway parameter; the interface's default gateway is used as the next hop for the source interface-based route. You can also specify another VR as the next-hop for the source-based route with the **set vrouter** *vrouter* **route source** *ip_addr/netmask* **vrouter** *next-hop_vrouter*.)
   - The metric for the source interface-based route. (If there are multiple source interface-based routes with the same prefix, only the route with the lowest metric is used for route lookup; other routes with the same prefix are marked as "inactive.")
   - Optional description for the source interface-based route.
2. Enable SIBR for the VR. The security device uses the source interface of the packet for route lookup in the SIBR table.

In Figure 323 on page 1243, traffic from users on the 10.1.1.0/24 subnetwork arrives on the security device on the ethernet1 interface and is forwarded to ISP 1, while traffic from users on the 10.1.2.0/24 subnetwork arrives on the device on ethernet2 and is forwarded to ISP 2. You need to configure two entries in the default trust-vr VR routing table and enable SIBR:

- The subnetwork 10.1.1.0/24, with ethernet1 as the source interface and ethernet3 as the forwarding interface, and ISP 1's router (1.1.1.1) as the next-hop
- The subnetwork 10.1.2.0/24, with ethernet2 as the source interface and ethernet4 as the forwarding interface, and ISP 2's router (2.2.2.2) as the next-hop

Figure 323: Source Interface-Based Routing Example



WebUI

Network > Routing > Source Interface Routing > New (for ethernet1): Enter the following, then click **OK**:

Network Address/Netmask: 10.1.1.0 255.255.255.0
Interface: ethernet3 (select)
Gateway IP Address: 1.1.1.1

Network > Routing > Source Interface Routing > New (for ethernet2): Enter the following, then click **OK**:

Network Address/Netmask: 10.1.2.0 255.255.255.0
Interface: ethernet4 (select)
Gateway IP Address: 2.2.2.2

NOTE: In the WebUI, the default preference and metric value are 1.

Network > Routing > Virtual Routers > Edit (for trust-vr): Select **Enable Source Interface Based Routing**, then click **OK**.

CLI

```
set vrouter trust-vr route source in-interface ethernet1 10.1.1.0/24 interface ethernet3
 gateway 1.1.1.1 metric 1
set vrouter trust-vr route source in-interface ethernet2 10.1.2.0/24 interface ethernet4
 gateway 2.2.2.2 metric 1
set vrouter trust-vr sibr-routing enable
save
```

Creating and Modifying Virtual Routers

This section contains various examples and procedures for modifying existing virtual routers (VRs) and for creating or deleting custom VRs.

Modifying Virtual Routers

You can modify a predefined or custom VR through either the WebUI or the CLI. For example, to modify the trust-vr VR:

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit

CLI

```
set vrouter trust-vr
```

You can modify the following parameters for VRs:

- Virtual router ID (see “Limiting the Number of Routing Table Entries” on page 1253).
- Maximum number of entries allowed in the routing table.
- Preference value for routes, based on protocol (see “Setting a Route Preference” on page 1254).
- Direct the VR to forward traffic based on the source IP address of a data packet (by default, the VR forwards traffic based on the destination IP address of a data packet. See “Source-Based Routing Table” on page 1239.)
- Enable or disable automatic route exporting to the untrust-vr for interfaces configured in route mode (for the trust-vr only).
- Add a default route with another VR as the next hop (for the trust-vr only).
- Make SNMP traps for the dynamic routing MIBs private (for the default root-level VR only).
- Allow routes on inactive interfaces to be considered for advertising (by default, only active routes defined on active interfaces can be redistributed to other protocols or exported to other VRs).
- Direct the VR to ignore overlapping subnet addresses for interfaces (by default, you cannot configure overlapping subnet IP addresses for interfaces in the same VR).
- Allow the VR to synchronize its configuration with the VR on its NetScreen Redundancy Protocol (NSRP) peer.

Assigning a Virtual Router ID

With dynamic routing protocols, each routing device uses a *unique* router identifier to communicate with other routing devices. The identifier can be in the form of a dotted decimal notation, like an IP address, or an integer value. If you do not define a specific virtual router ID (VR ID) before enabling a dynamic routing protocol, ScreenOS automatically selects the highest IP address of the active interfaces in the virtual router (VR) for the router identifier.

By default all security devices have IP address 192.168.1.1 assigned to the VLAN1 interface. If you do not specify a router ID before enabling a dynamic routing protocol on a security device, the IP address chosen for the router ID will likely be the default 192.168.1.1 address. This can cause a problem with routing because there cannot be multiple security VRs with the same VR ID in a routing domain. We recommend that you always explicitly assign a VR ID that is unique in the network. You can set the VR ID to the loopback interface address, as long as the loopback interface is not a Virtual Security Interface (VSI) in a NetScreen Redundancy Protocol (NSRP) cluster. (See “High Availability” on page 1763 for more information about configuring an NSRP cluster.)

In this example, you assign 0.0.0.10 as the router ID for the trust-vr.

NOTE: In the WebUI, you must enter the router ID in dotted decimal notation. In the CLI, you can enter the router ID either in dotted decimal notation (0.0.0.10) or you can simply enter 10 (this is converted by the CLI to 0.0.0.10).

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit: Enter the following, then click **OK**:

Virtual Router ID: Custom (select)
In the text box, enter 0.0.0.10

CLI

```
set vrouter trust-vr router-id 10
save
```

NOTE: You cannot assign or change a router ID if you have already enabled a dynamic routing protocol in the VR. If you need to change the router ID, you must first disable the dynamic routing protocol(s) in the VR. For information about disabling a dynamic routing protocol in the VR, see the appropriate chapter in this guide.

Forwarding Traffic Between Virtual Routers

When two VRs exist on a security device, traffic from zones in one VR is *not* automatically forwarded to zones in another VR even if there are policies that permit the traffic. If traffic must pass between VRs, you need to take one of these actions:

- Configure a static route in one VR that defines another VR as the next-hop for the route. This route can even be the default route for the VR. For example, you can configure a default route for the trust-vr with the untrust-vr as the next-hop. If the destination in an outbound packet does not match any other entries in the trust-vr routing table, it is forwarded to the untrust-vr. For information about configuring static routes, see "Configuring Static Routes" on page 1224.
- Export routes from the routing table in one VR into the routing table of another VR. You can export and import specific routes. You can also export all routes in the trust-vr routing table to the untrust-vr. This enables packets received in the untrust-vr to be forwarded to destinations in the trust-vr. For information, see "Exporting and Importing Routes Between Virtual Routers" on page 1265.

Configuring Two Virtual Routers

When multiple VRs exist within a security device, each VR maintains separate routing tables. By default, all predefined and user-defined security zones are bound to the trust-vr. This also means that all interfaces that are bound to those security zones also belong to the trust-vr. This section discusses how to bind a security zone (and its interfaces) to the untrust-vr VR.

You can bind a security zone to only one VR. You can bind multiple security zones to a single VR when there is no address overlap between zones. That is, all interfaces in the zones must be in route mode. Once a zone is bound to a VR, all the interfaces in that zone belong to the VR. You can change the binding of a security zone from one VR to another, however, you must first remove all interfaces from the zone. (For more information about binding and unbinding an interface to a security zone, see "Interfaces" on page 51.)

The following are the basic steps in binding a security zone to the untrust-vr VR:

1. Remove all interfaces from the zone that you want to bind to the untrust-vr. You cannot modify a zone-to-VR binding if there is an interface assigned to the zone. If you have assigned an IP address to an interface, you need to remove the address assignment before removing the interface from the zone.
2. Assign the zone to the untrust-vr VR.
3. Assign interface(s) back to the zone.

In the following example, the untrust security zone is bound by default to the trust-vr, and the interface ethernet3 is bound to the untrust security zone. (There are no other interfaces bound to the untrust security zone.) You must first set the IP address and netmask of the ethernet3 interface to 0.0.0.0, then change the bindings so that the untrust security zone is bound to the untrust-vr.

WebUI

1. **Unbind Interface from Untrust Zone**

   Network > Interfaces (ethernet3) > Edit: Enter the following, then click **OK**:

   Zone Name: Null
   IP Address/Netmask: 0.0.0.0/0

2. **Bind Untrust Zone to untrust-vr**

   Network > Zones (untrust) > Edit: Select **untrust-vr** from the Virtual Router Name drop-down list, then click **OK**.

3. **Bind Interface to Untrust Zone**

   Network > Interfaces (ethernet3) > Edit: Select **Untrust** from the Zone Name drop-down list, then click **OK**.

CLI

1. **Unbind Interface from Untrust Zone**

   ```
   unset interface ethernet3 ip
   unset interface ethernet3 zone
   ```

2. **Bind Untrust Zone to untrust-vr**

   ```
   set zone untrust vrouter untrust-vr
   ```

3. **Bind Interface to Untrust Zone**

   ```
   set interface eth3 zone untrust
   save
   ```

In the following example output, the **get zone** command shows the default interface, zone, and VR bindings. In the default bindings, the untrust zone is bound to the trust-vr.

```
device->  get zone
Total of 12 zones in vsys root. 7 policy configurable zone(s)
--------------------------------------------------------------
ID Name         Type    Attr    VR          Default-IF   VSYS
 0 Null         Null    Shared untrust-vr   null         Root
 1 Untrust      Sec(L3) Shared trust-vr     ethernet3    Root
 2 Trust        Sec(L3)        trust-vr     ethernet1    Root
 3 DMZ          Sec(L3)        trust-vr     ethernet2    Root
 4 Self         Func           trust-vr     self         Root
 5 MGT          Func           trust-vr     vlan1        Root
 6 HA           Func           trust-vr     null         Root
10 Global       Sec(L3)        trust-vr     null         Root
11 V1-Untrust   Sec(L2)        trust-vr     v1-untrust   Root
12 V1-Trust     Sec(L2)        trust-vr     v1-trust     Root
13 V1-DMZ       Sec(L2)        trust-vr     v1-dmz       Root
```

```
16 Untrust-Tun  Tun            trust-vr     null         Root
------------------------------------------------------------
```

You can choose to change the zone binding for the untrust-vr. Executing the **get zone** command shows the changed interface, zone, and VR bindings; in this case, the untrust zone is now bound to the untrust-vr.

```
device->  get zone
Total of 12 zones in vsys root. 7 policy configurable zone(s)
------------------------------------------------------------
ID Name         Type    Attr    VR          Default-IF   VSYS
 0 Null         Null    Shared untrust-vr   null         Root
 1 Untrust      Sec(L3) Shared untrust-vr   ethernet3    Root
 2 Trust        Sec(L3)        trust-vr     ethernet1    Root
 3 DMZ          Sec(L3)        trust-vr     ethernet2    Root
 4 Self         Func           trust-vr     self         Root
 5 MGT          Func           trust-vr     vlan1        Root
 6 HA           Func           trust-vr     null         Root
10 Global       Sec(L3)        trust-vr     null         Root
11 V1-Untrust   Sec(L2)        trust-vr     v1-untrust   Root
12 V1-Trust     Sec(L2)        trust-vr     v1-trust     Root
13 V1-DMZ       Sec(L2)        trust-vr     v1-dmz       Root
16 Untrust-Tun  Tun            trust-vr     null         Root
------------------------------------------------------------
```

Creating and Deleting Virtual Routers

Some security devices allow you to create custom VRs in addition to the two predefined VRs. You can modify all aspects of a user-defined VR, including the VR ID, the maximum number of entries allowed in the routing table, and the preference value for routes from specific protocols.

NOTE: Only certain security devices support custom VRs. To create custom VRs, you need a software license key.

Creating a Custom Virtual Router

In this example, you create a custom VR called trust2-vr and you enable automatic route exporting from the trust2-vr VR to the untrust-vr.

WebUI

Network > Routing > Virtual Routers > New: Enter the following, then click **OK**:

Virtual Router Name: trust2-vr
Auto Export Route to Untrust-VR: (select)

CLI

```
set vrouter name trust2-vr
set vrouter trust2-vr auto-route-export
save
```

Deleting a Custom Virtual Router

In this example, you delete an existing user-defined VR named trust2-vr.

WebUI

Network > Routing > Virtual Routers: Click **Remove** for the trust2-vr.

When the prompt appears asking you to confirm the removal, click **OK**.

CLI

```
unset vrouter trust2-vr
```

When the prompt appears asking you to confirm the removal (vrouter unset, are you sure? y/[n]), enter **Y**.

```
save
```

NOTE: You cannot delete the predefined untrust-vr and trust-vr VRs, but you can delete any user-defined VR. To modify the name of a user-defined VR or change the VR ID, you must first delete the VR and then recreate it with the new name or VR ID.

Dedicating a Virtual Router to Management

A management virtual router (MGT VR) supports the out-of-band management infrastructure and segments security device management traffic away from production traffic. By default, the ScreenOS TCP/IP stack first looks up routes for the self-initiated traffic in the default VR, and if routes are not found, it searches the route table in untrust-vr. When you designate a VR as a MGT VR, the TCP/IP stack looks up the routes in the MGT VR.

A MGT VR is valid in the root virtual system (vsys) only. A MGT VR is also not valid at Layer 2, because a virtual local area network (VLAN) interface at Layer 2 cannot be moved to another VR other than the trust-VR.

You can enable a new or an existing VR to be a MGT VR through either the WebUI or the CLI. In this example, the trust2-vr already exists.

WebUI

Network > Routing > Virtual Routers > Edit (for trust2-vr): Enter the following, then click **OK**.

```
Management VR (select)
```

CLI

```
set management-vrouter trust2-vr
save
```

You can bind a MGT zone, which belongs to the trust-vr by default, to another VR. Likewise, you can bind a MGT interface, which belongs to the MGT zone by default, to another zone.

Virtual Routers and Virtual Systems

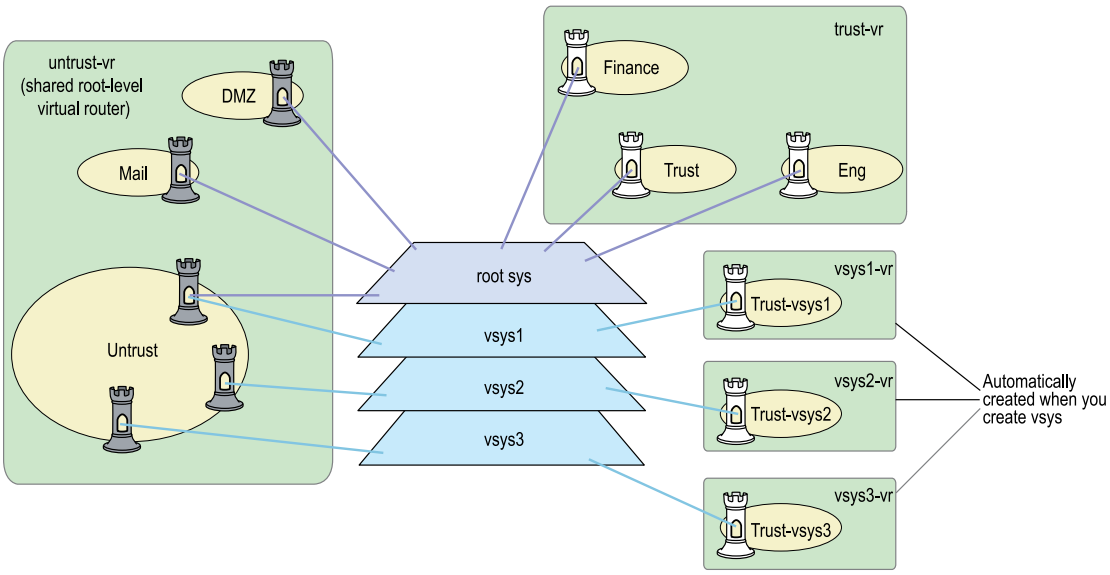
When a root-level administrator creates a vsys on virtual system-enabled systems, the vsys automatically has the following VRs available for its use:

- Any root-level VRs that have been defined as sharable. The untrust-vr is, by default, a shared VR that is accessible by any vsys. You can configure other root-level VRs to be sharable.
- A vsys-level VR. When you create a vsys, a vsys-level VR is automatically created that maintains the routing table for the Trust-*vsysname* zone. You can choose to name the VR vsysname-vr or a user-defined name. A vsys-level VR cannot be shared by other vsys.

NOTE: Only Juniper Networks security systems (NetScreen-5200, NetScreen-5400, ISG 1000, and ISG 2000) support vsys. To create vsys objects, you need a software license key.

You can define one or more custom VRs for a vsys. For more information about virtual systems, see “Virtual Systems” on page 1677. In Figure 324 on page 1251, each of the three vsys has two VRs associated with it: a vsys-level VR named vsysname-vr and the untrust-vr.

Figure 324: Virtual Routers Within a Vsys



Creating a Virtual Router in a Vsys

In this example, you define a custom VR vr-1a with the VR ID 10.1.1.9 for the vsys my-vsys1.

WebUI

Vsys > Configure > Enter (for my-vsys1) > Network > Routing > Virtual Routers > New: Enter the following, then click **Apply**:

Virtual Router Name: vr-1a
Virtual Router ID: Custom (select)
In the text box, enter 10.1.1.9

CLI

```
set vsys my-vsys1
(my-vsys1) set vrouter name vr-1a
(my-vsys1/vr-1a) set router-id 10.1.1.9
(my-vsys1/vr-1a) exit
(my-vsys1) exit
```

Enter **Y** at the following prompt:

```
Configuration modified, save? [y]/n
```

The vsys-level VR that is created when you create the vsys is the default VR for a vsys. You can change the default VR for a vsys to a custom VR. For example, you can make the custom VR vr-1a that you created previously in this example the default VR for the vsys my-vsys1:

WebUI

Vsys > Configure > Enter (for my-vsys1) > Network > Routing > Virtual Routers > Edit (for vr-1a): Select **Make This Vrouter Default-Vrouter for the System**, then click **Apply**.

CLI

```
set vsys my-vsys1
(my-vsys1) set vrouter vr-1a
(my-vsys1/vr-1a) set default-vrouter
(my-vsys1/vr-1a) exit
(my-vsys1) exit
```

Enter **Y** at the following prompt:

```
Configuration modified, save? [y]/n
```

The predefined Trust-*vsysname* security zone is bound by default to the vsys-level VR that is created when you created the vsys. However, you can bind the predefined Trust-*vsysname* security zone and any user-defined vsys-level security zone to any VR available to the vsys.

The untrust-vr is shared by default across all vsys. While vsys-level VRs are not sharable, you can define any root-level VR to be shared by the vsys. This allows you to define routes in a vsys-level VR that use a shared root-level VR as the next-hop. You can also configure route redistribution between a vsys-level VR and a shared root-level VR.

Sharing Routes Between Virtual Routers

In this example, the root-level VR my-router contains route table entries for the 4.0.0.0/8 network. If you configure the root-level VR my-router to be shareable by the vsys, then you can define a route in a vsys-level VR for the 4.0.0.0/8 destination with my-router as the next-hop. In this example, the vsys is my-vsys1, and the vsys-level VR is my-vsys1-vr.

WebUI

Network > Routing > Virtual Routers > New: Enter the following, then click **OK**:

```
Virtual Router Name: my-router
Shared and accessible by other vsys (select)
```

Vsys > Configure > Enter (for my-vsys1) > Network > Routing > Routing Entries > New (for my-vsys1-vr): Enter the following, then click **OK**:

```
Network Address/Netmask: 40.0.0.0 255.0.0.0
Next Hop Virtual Router Name: (select) my-router
```

CLI

```
set vrouter name my-router sharable
set vsys my-vsys1
(my-vsys1) set vrouter my-vsys1-vr route 40.0.0.0/8 vrouter my-router
(my-vsys1) exit
```

Enter **Y** at the following prompt:

```
Configuration modified, save? [y]/n
```

Limiting the Number of Routing Table Entries

Each VR is allocated the routing table entries it needs from a system-wide pool. The maximum number of entries available depends upon the security device and the number of VRs configured on the device. You can limit the maximum number of routing table entries that can be allocated for a specific VR. This helps prevent one VR from using up all the entries in the system.

NOTE: See the relevant product datasheet to determine the maximum number of routing table entries available on your Juniper Networks security device.

In this example, you set the maximum number of routing table entries for the trust-vr to 20.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr): Enter the following, then click **OK**:

```
Maximum Route Entry:
   Set limit at: (select), 20
```

CLI

```
set vrouter trust-vr max-routes 20
save
```

Routing Features and Examples

After configuring the required VRs for your network, you can determine which routing features you want to employ. These features affect routing behaviors and routing table data. These features are applicable to static routing and dynamic routing protocols.

This section explains the following topics:

- "Route Selection" on page 1254
- "Configuring Equal Cost Multipath Routing" on page 1259

- "Route Redistribution" on page 1261
- "Exporting and Importing Routes Between Virtual Routers" on page 1265

Route Selection

Multiple routes with the same prefix (IP address and mask) can exist in the routing table. Where the routing table contains multiple routes to the same destination, the preference values of each route are compared. The route that has the lowest preference value is selected. If the preference values are the same, the metric values are then compared. The route with the lowest metric value is then selected.

NOTE: If there are multiple routes to the same destination with the *same* preference values and the same metric values, then any one of those routes can be selected. In this case, selection of one specific route over another is not guaranteed or predictable.

Setting a Route Preference

A route preference is a weight added to the route that influences the determination of the best path for traffic to reach its destination. When importing or adding a route to the routing table, the VR adds a preference value — determined by the protocol by which the route is learned — to the route. A low preference value (a number closer to 0) is preferable to a high preference value (a number further from 0).

In a VR, you can set the preference value for routes according to protocol. Table 82 on page 1254 lists the default preference values for routes of each protocol.

Table 82: Default Route Preference Values

Protocol	Default Preference
Connected	0
Static	20
Auto-Exported	30
EBGP	40
OSPF	60
RIP	100
Imported	140
OSPF External Type 2	200
IBGP	250

You can also adjust the route preference value to direct traffic along preferred paths.

In this example, you specify a value of 4 as the preference for any “connected” routes added to the route table for the untrust-vr.

NOTE: If the route preference changes for any type of route (for example, OSPF type 1 routes), the new preference displays in the route table but the new preference does not take effect until the route is relearned (which can be achieved by disabling, then enabling, the dynamic routing protocol), or, in the case of static routes, deleted and added again.

Changing the route preference does not affect existing routes. To apply changes to existing routes, you need to delete the routes then re-add them. For dynamic routes, you need to disable the protocol then re-enable it or restart the device.

A route is connected when the router has an interface with an IP address in the destination network.

WebUI

Network > Routing > Virtual Routers > Edit (for untrust-vr): Enter the following, then click **OK**:

Route Preference:
Connected: 4

CLI

```
set vrouter untrust-vr preference connected 4
save
```

Route Metrics

Route metrics determine the best path a packet can take to reach a given destination. Routers use route metrics to weigh two routes to the same destination and determine the use of one route over the other. When there are multiple routes to the same destination network with the same preference value, the route with the lowest metric prevails.

A route metric can be based on any or a combination of the following elements:

- Number of routers a packet must traverse to reach a destination
- Relative speed and bandwidth of the path
- Dollar cost of the links making up the path
- Other factors

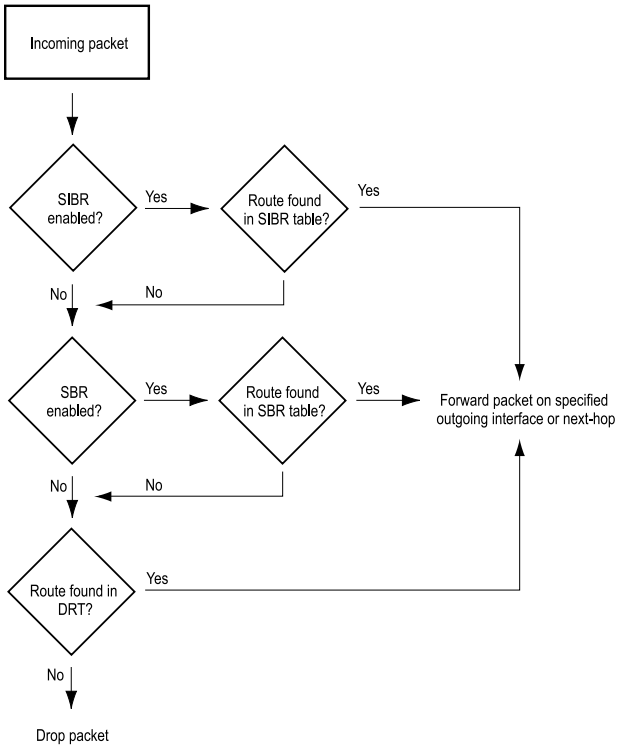
When routes are learned dynamically, the neighboring router from which the route originates provides the metric. The default metric for connected routes is always 0. The default metric for static routes is 1.

Changing the Default Route Lookup Sequence

If you enable both source-based routing and SIBR in a VR, the VR performs route lookup by checking the incoming packet against the routing tables in a specific order. This section describes the default route lookup sequence and how you can change the sequence by configuring preference values for each routing table.

If an incoming packet does not match an existing session, the security device performs First Packet Processing, a procedure that involves route lookup. Figure 325 on page 1256 shows the default route lookup sequence.

Figure 325: Default Route Lookup Sequence



1. If SIBR is enabled in the VR, the security device first checks the SIBR table for a route entry that matches the interface on which the packet arrived. If the security device finds a route entry for the source interface in the SIBR table, it forwards the packet as specified by the matching routing entry. If the security device does not find a route entry for the source interface in the SIBR table, the device checks to see if source-based routing is enabled in the VR.
2. If source based routing is enabled in the VR, the security device checks the source based routing table for a route entry that matches the source IP address of the packet. If the security device finds a matching route entry for the source IP address, it forwards the packet as specified by the entry. If the security device does not find a route entry for the source IP address in the source based routing table, the device checks the destination-based routing table.

3. The security device checks the destination-based routing table for a route entry that matches the destination IP address of the packet. If the security device finds a matching route entry for the destination IP address, it forwards the packet as specified by the entry. If the device does not find an exact matching route entry for the destination IP address but a default route configured for the VR, the device forwards the packet as specified by the default route. If the security device does not find a route entry for the destination IP address and there is no default route configured for the VR, the packet is dropped.

The order in which the security device checks routing tables for a matching route is determined by a preference value assigned to each routing table. The routing table with the highest preference value is checked first while the routing table with the lowest preference value is checked last. By default, the SIBR table has the highest preference value (3), the source based routing table has the next-highest preference value (2), and the destination-based routing table has the lowest preference value (1).

You can reassign new preference values to a routing table to change the order in which the security device performs route lookup in a VR. Remember that the device checks routing tables from the highest to lowest preference values.

In the following example, you enable both SIBR and source-based routing in the trust-vr. You want the security device to perform route lookups in the routing tables in the following order: source-based routing first, SIBR, and then destination-based routing. To configure this sequence of route table lookup, you need to configure source-based routing with a higher preference value than SIBR — in this example, you assign a preference value of 4 to source-based routing.

WebUI

Network > Routing > Virtual Router > Edit (for trust-vr): Enter the following, then click **OK**:

Route Lookup Preference (1-255): (select)
  For Source Based Routing: 4
Enable Source Based Routing: (select)
Enable Source Interface Based Routing: (select)

CLI

```
set vrouter trust-vr sibr-routing enable
set vrouter trust-vr source-routing enable
set vrouter trust-vr route-lookup preference source-routing 4
save
```

Route Lookup in Multiple Virtual Routers

You can specify another VR as the next-hop for a destination-based route entry only and not for a source-based or source interface-based route entry. For example, the default route in the destination-based routing table can specify the untrust-vr as the next-hop; then the untrust-vr entry can specify another VR, such as a DMZ. The device will check up to a total of three VRs. Where route lookup in one VR results in

a route lookup in another VR, the security device always performs a second route lookup in the destination-based route table.

In the example, you enable source-based routing in both the trust-vr and untrust-vr routing tables. The trust-vr has the following routing entries:

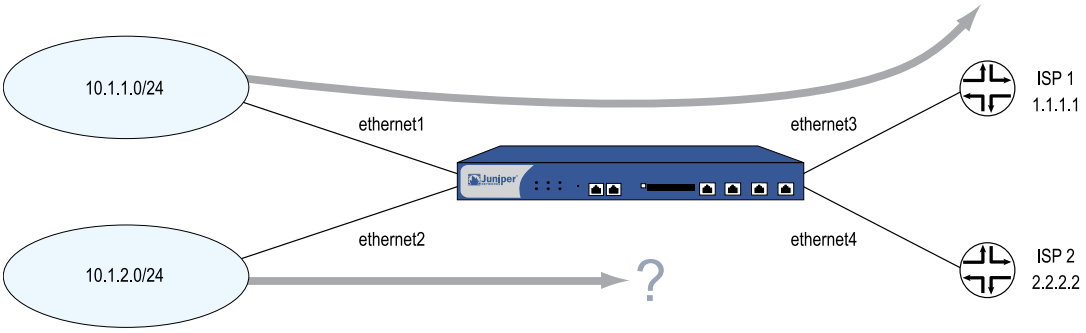
- A source-based routing entry for the subnetwork 10.1.1.0/24, with ethernet3 as the forwarding interface, and the router at 1.1.1.1 as the next-hop
- A default route, with the untrust-vr as the next-hop.

The untrust-vr has the following routing entries:

- A source-based routing entry for the subnetwork 10.1.2.0/24, with ethernet4 as the forwarding interface, and the router at 2.2.2.2 as the next-hop
- A default route, with ethernet3 as the forwarding interface and the router at 1.1.1.1 as the next-hop

Figure 326 on page 1258 shows how traffic from the subnetwork 10.1.2.0/24 will always be forwarded on ethernet3 to the router at 1.1.1.1.

Figure 326: Route Lookup in Multiple VRs



The source-based routing table for the trust-vr includes the following entry:

```
  ID          IP-Prefix      Interface         Gateway   P Pref    Mtr     Vsys
*  1        10.1.1.0/24           eth3       2.2.2.250   S   20      1     Root
```

The destination-based routing table for the untrust-vr includes the following entry:

```
  ID          IP-Prefix      Interface         Gateway   P Pref    Mtr     Vsys
*  1         0.0.0.0/24            n/a      untrust-vr   S   20      0     Root
```

Traffic from 10.1.2.0/24 subnetwork arrives on the security device on ethernet2. Because there is no matching source-based route entry, the security device performs route lookup in the destination-based routing table. The default route in the destination-based routing table specifies the untrust-vr as the next-hop.

Next, the security device does not check the source-based routing table for the untrust-vr to find the following entry:

```
  ID          IP-Prefix      Interface         Gateway   P Pref    Mtr     Vsys
*  1        10.1.2.0/24           eth4       2.2.2.250   S   20      1     Root
```

Instead, the security device checks the destination-based Routing Table and finds the following entry:

```
  ID          IP-Prefix      Interface         Gateway   P Pref    Mtr     Vsys
*  1         0.0.0.0/24           eth3       1.1.1.150   S   20      0     Root
```

In the untrust-vr, the security device performs route lookup in the destination-based routing table only, even though the source-based routing table in the untrust-vr contains an entry that would match the traffic. The matching route in the destination-based routing table (the default route) forwards the traffic out on the ethernet3 interface.

Configuring Equal Cost Multipath Routing

Juniper Networks security devices support equal cost multipath (ECMP) routing on a per-session basis. Routes of equal cost have the same preference and metric values. Once a security device associates a session with a route, the security device uses that route until a better route is learned or the current route becomes unusable. The eligible routes must have outgoing interfaces that belong to the same zone.

NOTE: If the outgoing interfaces do not belong to the same zone and the return packet goes to a zone other than the intended one, a session match cannot occur and the traffic may not go through.

NOTE: When ECMP is enabled and the outgoing interfaces are different and in NAT mode, applications, such as HTTP, that create multiple sessions will not work correctly. Applications, such as telnet or SSH, that create one session should work correctly.

ECMP assists with load-balancing among two to four routes to the same destination or increases the effective bandwidth usage among two or more destinations. When ECMP is enabled, security devices use the statically defined routes or dynamically learn multiple routes to the same destination through a routing protocol. The security device assigns routes of equal cost in rotating (round-robin) fashion.

Without ECMP, the security device only uses the first learned or defined route. Other routes that are of equal cost remain unused until the currently active route is no longer active.

NOTE: When using ECMP, if you have two security devices in a neighbor relationship and you notice packet loss and improper load-balancing, check the Address Resolution Protocol (ARP) configuration of the neighbor device to make sure the **arp always-on-dest** feature is disabled (default). For more information about ARP-related commands, see “Down Interfaces and Traffic Flow” on page 92.

For example, consider the following two routes that appear in the trust-vr destination-based routing table:

```
ID         IP-Prefix      Interface         Gateway   P Pref    Mtr     Vsys
*   8        0.0.0.0/0      ethernet3       1.1.1.250   C    0      1     Root
    9        0.0.0.0/0      ethernet2       2.2.2.250   S   20      1     Root
```

In this example, two default routes exist to provide connections to two different ISPs, and the goal is to use both default routes with ECMP.

The two routes have the same metric values; however, the first route is a connected route (C with a preference of 0). The security device acquired the first route through DHCP or PPP, and the device acquired the default route through manual configuration. The second route is a manually configured static route (S with an automatic preference of 20). With ECMP disabled, the security device forwards all traffic to the connected route on ethernet3.

To achieve load-balancing with both routes, you change the route preference of the static route to zero (0) to match the connected route by entering the **set vrouter** *trust-vr* **preference static 0** command and then enabling ECMP. With ECMP enabled, the security device load-balances the traffic by alternating between the two eligible ECMP routes. The following display shows the updated routing table.

```
ID         IP-Prefix      Interface         Gateway   P Pref    Mtr     Vsys
*  8        0.0.0.0/0       ethernet3       1.1.1.250    C   0      1     Root
*  9        0.0.0.0/0       ethernet2       2.2.2.250    S   0      1     Root
```

If you enable ECMP, and the security device finds more than one matching route of the same cost in a routing table, the device selects a different equal-cost route for each route lookup. With the routes shown above, the security device alternates between ethernet3 and ethernet2 to forward traffic to the 0.0.0.0/0 network.

If more than two equal-cost routes to the network exist, the security device selects from the routes in round-robin order up to the configured maximum so that the device selects a different ECMP route for each route lookup.

ECMP is disabled by default (the maximum number of routes is 1). To enable ECMP routing, you need to specify the maximum number of equal-cost routes on a per-virtual router basis. You can specify up to four routes. Once you set the maximum number of routes, the security device will not add or change routes even if more routes are learned.

In the following example, you set the maximum number of ECMP routes in the trust-vr to 2. Even though 3 or 4 routes of equal cost might exist within the same zone and in the routing table, the security device only alternates between the configured number of eligible routes. In this case, data only forwards along the 2 specified ECMP paths.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr): Enter the following, then click **OK**:

Maximum ECMP Routes:
Set Limit at: (select), 2

CLI

```
set vrouter trust-vr max-ecmp-routes 2
save
```

Route Redistribution

The routing table in a VR contains routes gathered by all dynamic routing protocols running in the VR, as well as static routes and directly connected routes. By default, a dynamic routing protocol (such as OSPF, RIP, or BGP) advertises to its neighbors or peers only the routes that meet the following conditions:

- The routes must be active in the routing table.
- The routes must be learned by the dynamic routing protocol.

NOTE: OSPF, RIP, and BGP also advertise connected routes for the ScreenOS interfaces on which these protocols are enabled.

To allow a dynamic routing protocol to advertise routes that were learned by another protocol, including statically configured routes, you need to *redistribute* routes from the source protocol into the advertising protocol.

You can redistribute routes learned from a routing protocol (including statically configured routes) into a different routing protocol in the same VR. This allows the receiving routing protocol to advertise the redistributed routes. When importing a route, the current domain has to translate all the information, particularly known routes, from the other protocol to its own protocol. For example, if a routing domain uses OSPF and it connects to a routing domain using BGP, the OSPF domain has to import all the routes from the BGP domain to inform all of its OSPF neighbors about how to reach devices in the BGP domain.

Routes are redistributed between protocols according to a *redistribution rule* defined by the system or network administrator. When a route is added to a routing table in a VR, all redistribution rules defined in the VR are applied one-by-one to the route to determine whether the route is to be redistributed. When a route is deleted from a routing table, all redistribution rules defined in the VR are applied one-by-one to the route to determine whether the route is to be deleted from another routing protocol within the VR. Note that all redistribution rules are applied to the added or deleted route. There is no concept of rule order or “first matching rule” for redistribution rules.

NOTE: You can only define one redistribution rule between any two protocols.

On the security device, you configure a *route map* to specify which routes are to be redistributed and the attributes of the redistributed routes.

Configuring a Route Map

A *route map* consists of a set of statements applied in sequential order to a route. Each statement in the route map defines a condition that is compared to the route. A route is compared to each statement in a specified route map in order of increasing sequence number until there is a match, then the action specified by the statement is applied. If the route matches the condition in the route map statement, the route is either permitted or rejected. A route map statement can also modify certain attributes of a matching route. There is an implicit deny at the end of every route map; that is, if a route does not match any entry in the route map, the route is rejected. “Initiating an AV Profile for Internal AV” on page 528 lists route map match conditions and gives a description of each.

Table 83: Route Map Match Conditions

Match Condition	Description
BGP AS Path	Matches a specified AS path access list. See “Route Filtering” on page 1263.
BGP Community	Matches a specified community list. See “Route Filtering” on page 1263.
OSPF route type	Matches OSPF internal, external type 1, or external type 2.
Interface	Matches a specified interface.
IP address	Matches a specified access list. See “Route Filtering” on page 1263.
Metric	Matches a specified route metric value.
Next-hop	Matches a specified access list. See “Route Filtering” on page 1263.
Tag	Matches a specified route tag value or IP address.

For each match condition, you specify whether a route that matches the condition is accepted (permitted) or rejected (denied). If a route matches a condition and is permitted, you can optionally set attribute values for the route. Table 84 on page 1262 lists route map attributes and descriptions of each.

Table 84: Route Map Attributes

Set Attributes	Description
BGP AS Path	Prepends a specified AS path access list to the path list attribute of the matching route.
BGP Community	Sets the community attribute of the matching route to the specified community list.
BGP local preference	Sets the local-pref attribute of the matching route to the specified value.
BGP weight	Sets the weight of the matching route.

Table 84: Route Map Attributes *(continued)*

Offset metric	Increments the metric of the matching route by the specified number. This increases the metric on a less desirable path. For RIP routes, you can apply the increment to either routes advertised (route-map out) or routes learned (route-map in). For other routes, you can apply the increment to routes that are exported into another VR.
OSPF metric type	Sets the OSPF metric type of the matching route to either external type 1 or external type 2.
Metric	Sets the metric of the matching route to the specified value.
Next-hop of route	Sets the next-hop of the matching route to the specified IP address.
Preserve metric	Preserves the metric of a matching route that is exported into another VR.
Preserve preference	Preserves the preference value of the matching route that is exported into another VR.
Tag	Sets the tag of the matching route to the specified tag value or IP address.

Route Filtering

Route filtering allows you to control which routes to permit into a VR, which routes to advertise to peers, and which routes to redistribute from one routing protocol to another. You can apply filters to incoming routes sent by a routing peer or to outgoing routes sent by the security VR to peer routers. You can use the following filtering mechanisms:

- Access list—See "Configuring an Access List" on page 1263 for information about configuring an access list.
- BGP AS-path access list—An AS-path attribute is a list of autonomous systems through which a route advertisement has passed and which is part of the route information. An AS-path access list is a set of regular expressions that represent specific ASs. You can use an AS-path access list to filter routes based on the AS through which the route has traversed. See "Configuring an AS-Path Access List" on page 1357 for information about configuring an AS-path access list.
- BGP community list—A community attribute contains identifiers for the communities to which a BGP route belongs. A BGP community list is a set of BGP communities that you can use to filter routes based on the communities to which a route belongs. See "BGP Communities" on page 1366 for information about configuring a BGP community list.

Configuring an Access List

An access list is a sequential list of statements against which a route is compared. Each statement specifies the IP address/netmask of a network prefix and the forwarding status (permit or deny the route). For example, a statement in an access list can allow routes for the 1.1.1.0/24 subnet. Another statement in the same access

list can deny routes for the 2.2.2.0/24 subnet. If a route matches a statement in the access list, the specified forwarding status is applied.

The sequence of statements in an access list is important because a route is compared to the first statement in the access list and then to subsequent statements until there is a match. If there is a match, all subsequent statements in the access list are ignored. You should sequence the more specific statements before less specific statements. For example, place the statement that denies routes for the 1.1.1.1/30 subnet before the statement that permits routes for the 1.1.1.0/24 subnet.

You can also use access lists to control the flow of multicast traffic. For information, see "Access Lists" on page 1394.

In this example, you create an access list on the trust-vr. The access list has the following characteristics:

- Identifier: 2 (you must specify an access list identifier when configuring the access list)
- Forwarding Status: permit
- IP Address/Netmask Filtering: 1.1.1.1/24
- Sequence Number: 10 (positions this statement relative to other statements in the access list)

WebUI

Network > Routing > Virtual Routers > Access List: > New (for trust-vr): Enter the following, then click **OK**:

```
Access List ID: 2
Sequence No: 10
IP/Netmask: 1.1.1.1/24
Action: Permit
```

CLI

```
set vrouter trust-vr access-list 2 permit ip 1.1.1.1/24 10
save
```

Redistributing Routes into OSPF

In this example, you redistribute specified BGP routes that have passed through the autonomous system 65000 into OSPF. You first configure an AS-path access list that allows routes that have passed through AS 65000. (For more information about configuring an AS-path access list, see "Configuring an AS-Path Access List" on page 1357.) Next, you configure a route map "rtmap1" to match routes in the AS path access list. Finally, in OSPF, you specify a redistribution rule that uses the route map "rtmap1" and then specifies BGP as the source protocol for the routes.

WebUI

1. **BGP AS-Path Access List**

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > AS Path: Enter the following, then click **Add**:

AS Path Access List ID: 1
Permit: (select)
AS Path String: _65000_

2. **Route Map**

Network > Routing > Virtual Routers > Route Map > New (for trust-vr): Enter the following, then click **OK**:

Map Name: rtmap1
Sequence No.: 10
Action: permit (select)
Match Properties:
AS Path: (select), 1

3. **Redistribution Rule**

Network > Routing > Virtual Router > Edit (for trust-vr) > Edit OSPF Instance > Redistributable Rules: Select the following, then click **Add**:

Route Map: rtmap1
Protocol: BGP

CLI

1. **BGP AS-Path Access List**

```
set vrouter trust-vr protocol bgp as-path-access-list 1 permit _65000_
```

2. **Route Map**

```
set vrouter trust-vr
device(trust-vr)-> set route-map name rtmap1 permit 10
device(trust-vr/rtmap1-10)-> set match as-path 1
device(trust-vr/rtmap1-10)-> exit
device(trust-vr)-> exit
```

3. **Redistribution Rule**

```
set vrouter trust-vr protocol ospf redistribute route-map rtmap1 protocol bgp
save
```

Exporting and Importing Routes Between Virtual Routers

If you have two VRs configured on a security device, you can allow specified routes in one VR to be learned by the other VR. To do this, you must define *export rules* on the source VR that will export routes to the destination VR. When exporting routes, a VR allows other VRs to learn about its network. On the destination VR, you can optionally configure *import rules* to control the routes that are allowed to be imported

from the source VR. If there are no import rules on the destination VR, all exported routes are accepted.

To export and import routes between VRs:

1. On the source VR, define an export rule.
2. (Optional) On the destination VR, define an import rule. While this step is optional, an import rule allows you to further control the routes that the destination VR accepts from the source VR.

On the security device, you configure an export or import rule by specifying the following:

- The destination VR (for export rules) or source VR (for import rules)
- The protocol of the routes to be exported/imported
- Which routes are to be exported/imported
- (Optional) New or modified attributes of the exported/imported routes

Configuring an export or import rule is similar to configuring a redistribution rule. You configure a *route map* to specify which routes are to be exported/imported and the attributes of the routes.

You can configure the trust-vr to automatically export all its route table entries to the untrust-vr. You can also configure a user-defined VR to automatically export routes to other VRs. Routes in networks directly connected to interfaces in NAT mode cannot be exported.

Configuring an Export Rule

In this example, OSPF routes for the 1.1.1.1/24 network in the trust-vr are exported to the untrust-vr routing domain. You first create an access list for the network prefix 1.1.1.1/24, which is then used in the route map "rtmap1" to filter for matches of routes for the 1.1.1.1/24 network. You then create a route export rule to export matching OSPF routes from the trust-vr to the untrust-vr.

WebUI

trust-vr

1. **Access List**

   Network > Routing > Virtual Routers > Access List: > New (for trust-vr): Enter the following, then click **OK**:

   Access List ID: 2
   Sequence No: 10
   IP/Netmask: 1.1.1.1/24
   Action: Permit

2. **Route Map**

Network > Routing > Virtual Routers > Route Map > New (for trust-vr): Enter the following, then click **OK**:

Map Name: rtmap1
Sequence No.: 10
Action: permit (select)
Match Properties:
Access List: (select), 2

3. **Export Rule**

Network > Routing > Virtual Routers > Export Rules > New (for trust-vr): Enter the following, then click **OK**:

Destination Virtual Router: untrust-vr
Route Map: rtmap1
Protocol: OSPF

CLI

trust-vr

1. **Access List**

```
set vrouter trust-vr
device(trust-vr)-> set access-list 2 permit ip 1.1.1.1/24 10
```

2. **Route Map**

```
device(trust-vr)-> set route-map name rtmap1 permit 10
device(trust-vr/rtmap1-10)-> set match ip 2
device(trust-vr/rtmap1-10)-> exit
```

3. **Export Rule**

```
device(trust-vr)-> set export-to vrouter untrust-vr route-map rtmap1 protocol ospf
device(trust-vr)-> exit
save
```

Configuring Automatic Export

You can configure the trust-vr to automatically export all of its routes to the untrust-vr.

CAUTION: This feature can override the isolation between the trust-vr and untrust-vr by making all trusted routes visible in the untrusted network.

If you define import rules for the untrust-vr, only routes that match the import rules are imported. In this example, the trust-vr automatically exports all routes to the untrust-vr, but an import rule on the untrust-vr allows only internal OSPF routes to be exported.

WebUI

trust-vr

Network > Routing > Virtual Router > Edit (for trust-vr): Select **Auto Export Route to Untrust-VR**, then click **OK**.

untrust-vr

Network > Routing > Virtual Router > Route Map (for untrust-vr) > New: Enter the following, then click **OK**:

Map Name: from-ospf-trust
Sequence No.: 10
Action: permit (select)
Route Type: internal-ospf (select)

CLI

trust-vr

```
set vrouter trust-vr auto-route-export
```

untrust-vr

```
set vrouter untrust-vr
device(untrust-vr)-> set route-map name from-ospf-trust permit 10
device(untrust-vr/from-ospf-trust-10)-> set match route-type internal-ospf
device(untrust-vr/from-ospf-trust-10)-> exit
device(untrust-vr)-> set import-from vrouter trust-vr route-map from-ospf-trust protocol
 ospf
device(untrust-vr)-> exit
save
```

Chapter 34

Open Shortest Path First

This chapter describes the Open Shortest Path First (OSPF) routing protocol on security devices. It contains the following sections:

Overview

The Open Shortest Path First (OSPF) routing protocol is an Interior Gateway Protocol (IGP) intended to operate within a single Autonomous System (AS). A router running OSPF distributes its state information (such as usable interfaces and neighbor reachability) by periodically flooding *link-state advertisements* (LSAs) throughout the AS.

Each OSPF router uses LSAs from neighboring routers to maintain a link-state database. The link-state database is a listing of topology and state information for the surrounding networks. The constant distribution of LSAs throughout the routing domain enables all routers in an AS to maintain identical link-state databases.

OSPF uses the link-state database to determine the best path to any network within the AS. This is done by generating a *shortest-path tree*, which is a graphical representation of the shortest path to any network within the AS. While all routers have the same link state database, they all have unique shortest-path trees because routers always generate the tree with themselves at the top of the tree.

Areas

By default, all routers are grouped into a single “backbone” area called area 0 (usually denoted as area 0.0.0.0). However, large geographically dispersed networks are typically segmented into multiple areas. As networks grow, link-state databases grow and dividing the link-state database into smaller groups allows for better scalability.

Areas reduce the amount of routing information passed throughout the network because a router only maintains a link-state database for the area in which it resides. No link-state information is maintained for networks or routers outside the area. A router connected to multiple areas maintains a link-state database for each area to which it is connected. Areas must be directly connected to area 0 except when creating a virtual link. For more information about virtual links, see “Virtual Links” on page 1283.

AS external advertisements describe routes to destinations in other ASs and are flooded throughout an AS. Certain OSPF areas can be configured as *stub areas*; AS external advertisements are not flooded into these areas. There are two common types of areas used in OSPF:

- **Stub area**—An area that receives route summaries from the backbone area but does not receive link-state advertisements from other areas for routes learned through non-OSPF sources (BGP, for example). A stub area can be considered a *totally stubby area* if no summary routes are allowed in the stub area.
- **Not So Stubby Area (NSSA)**—Like a normal stub area, NSSAs cannot receive routes from non-OSPF sources outside the current area. However, external routes learned within the area can be learned and passed to other areas.

Router Classification

Routers that participate in OSPF routing are classified according to their function or location in the network:

- **Internal Router**—A router with all interfaces belonging to the same area.
- **Backbone Router**—A router that has an interface in the backbone area.
- **Area Border Router**—A router that attaches to multiple areas is called an area border router (ABR). An ABR summarizes routes from non-backbone areas for distribution to the backbone area. On security devices running OSPF, the backbone area is created by default. If you create a second area in a virtual router, the device functions as an ABR.
- **AS Boundary Router**—When an OSPF area borders another AS, the router between the two autonomous systems is called an autonomous system boundary router (ASBR). An ASBR is responsible for advertising external AS routing information throughout an AS.

Hello Protocol

Two routers with interfaces on the same subnet are considered *neighbors*. Routers use the Hello protocol to establish and maintain these neighbor relationships. When

two routers establish bidirectional communication, they are said to have established an *adjacency*. If two routers do not establish an adjacency, they cannot exchange routing information.

In cases were there are multiple routers on a network, it is necessary to establish one router as the *designated router* (DR) and another as the *backup designated router* (BDR). The DR is responsible for flooding the network with LSAs that contain a list of all OSPF-enabled routers attached to the network. The DR is the only router that can form adjacencies with other routers on the network. Therefore, the DR is the only router on a network that can provide routing information to other routers. The BDR is responsible for becoming the designated router if the DR should fail.

Network Types

Juniper Networks security devices support the following OSPF network types:

- Broadcast
- Point-to-Point
- Point-to-Multipoint

Broadcast Networks

A *broadcast network* is a network that connects many routers together and can send, or broadcast, a single physical message to all the attached routers. Pairs of routers on a broadcast network are assumed to be able to communicate with each other. Ethernet is an example of a broadcast network.

On broadcast networks, the OSPF router dynamically detects its neighbor routers by sending hello packets to the multicast address 224.0.0.5. For broadcast networks, the Hello protocol elects a Designated Router and Backup Designated Router for the network.

A *non-broadcast network* is a network that connects many routers together but cannot broadcast messages to attached routers. On non-broadcast networks, OSPF protocol packets that are normally multicast need to be sent to each neighboring router. Juniper Networks security devices do not support OSPF on non-broadcast networks.

Point-to-Point Networks

A *point-to-point* network typically joins two routers over a Wide Area Network (WAN). An example of a point-to-point network is two security devices connected by an IPsec VPN tunnel. On point-to-point networks, the OSPF router dynamically detects neighbor routers by sending hello packets to the multicast address 224.0.0.5.

Point-to-Multipoint Networks

A *point-to-multipoint* network is a non-broadcast network where OSPF treats connections between routers as point-to-point links. No election of a designated router or LSA flooding exists for the network. A router in a point-to-multipoint network sends hello packets to all neighbors with which it can directly communicate.

NOTE: On security devices, OSPF point-to-multipoint configuration is only supported on tunnel interfaces, and you must disable route-deny for proper network operation. You cannot configure a physical Ethernet interface for point-to-multipoint connections. For more information, see "Point-to-Multipoint Tunnel Interface" on page 1293.

Link-State Advertisements

Each OSPF router sends out LSAs that define the local state information for the router. Additionally, there are other types of LSAs that a router can send out, depending upon the OSPF function of the router. Table 85 on page 1272 lists LSA types, where each type is flooded, and the contents of each type of LSA.

Table 85: LSA Types and Content Summary

LSA Type	Sent By	Flooded Throughout	Information Sent in LSA
Router LSA	All OSPF routers	Area	Describes the state of all router interfaces throughout the area.
Network LSA	Designated Router on broadcast and NBMA networks	Area	Contains a list of all routers connected to the network.
Summary LSA	Area Border Routers	Area	Describes a route to a destination outside the area but still inside the AS. There are two types: ■ Type 3 summary-LSAs describe routes to networks. ■ Type 4 summary-LSAs describe routes to AS boundary routers.
AS-External	Autonomous System Boundary Router	Autonomous System	Routes to networks in another AS. Often, this is the default route (0.0.0.0/0).

Basic OSPF Configuration

You create OSPF on a per-virtual router basis on a security device. If you have multiple virtual routers (VRs) in a system, you can enable multiple instances of OSPF, one instance for each VR.

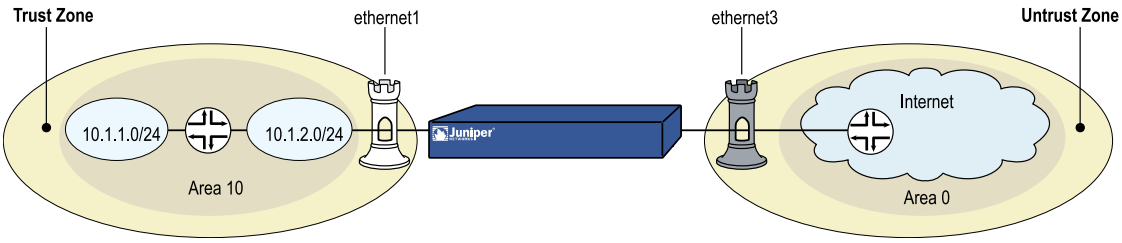
NOTE: Before you configure a dynamic routing protocol on the security device, you should assign a VR ID, as described in "Routing" on page 1235.

This section describes the following basic steps to configure OSPF in a VR on a security device:

1. Create and enable the OSPF routing instance in a VR. This step also automatically creates an OSPF backbone area, with an area ID of 0.0.0.0, which cannot be deleted.
2. (Optional) Unless all OSPF interfaces will be connected to the backbone area, you need to define a new OSPF area with its own area ID. For example, if the security device is to act as an ABR, you need to create a new OSPF area in addition to the backbone area. You can configure the new area as a normal, stub, or not-so-stubby area.
3. Assign one or more interfaces to each OSPF area. You must explicitly add interfaces to an OSPF area, including the backbone area.
4. Enable OSPF on each interface.
5. Verify that OSPF is properly configured and operating.

In this example, you configure the security device as an ABR connecting to area 0 through the ethernet3 interface and connecting to area 10 through the ethernet1 interface. See Figure 327 on page 1273.

Figure 327: OSPF Configuration Example



You can optionally configure other OSPF parameters, such as the following:

- Global parameters, such as virtual links, that are set at the VR level for the OSPF protocol (see “Global OSPF Parameters” on page 1282).
- Interface parameters, such as authentication, that are set on a per-interface basis for the OSPF protocol (see “Setting OSPF Interface Parameters” on page 1286).
- Security-related OSPF parameters that are set at either the VR level or on a per-interface basis (see “Security Configuration” on page 1288).

Creating and Removing an OSPF Routing Instance

You create and enable an OSPF routing instance on a specific VR on a security device. To remove an OSPF routing instance you disable the OSPF instance and then delete it. Creating the OSPF routing instance also automatically creates an OSPF backbone area. When you create and enable an OSPF routing instance on a VR, OSPF can transmit and receive packets on all OSPF-enabled interfaces in the VR.

Creating an OSPF Instance

In the following example, you first assign 0.0.0.10 as the router ID for the trust-vr. You then create an OSPF routing instance on the trust-vr. (For more information about VRs and configuring a VR on security devices, see “Routing” on page 1235.)

WebUI

1. **Router ID**

   Network > Routing > Virtual Router (trust-vr) > Edit: Enter the following, then click **OK**:

   Virtual Router ID: Custom (select)
   In the text box, enter 0.0.0.10

2. **OSPF Routing Instance**

   Network > Routing > Virtual Router (trust-vr) > Edit > Create OSPF Instance: Select **OSPF Enabled**, then click **OK**.

CLI

1. **Router ID**

   ```
   set vrouter trust-vr router-id 10
   ```

2. **OSPF Routing Instance**

   ```
   set vrouter trust-vr protocol ospf
   set vrouter trust-vr protocol ospf enable
   save
   ```

NOTE: In the CLI, you must first create the OSPF routing instance before you can enable it. Thus, you must issue two separate CLI commands to enable an OSPF routing instance.

Removing an OSPF Instance

In this example, you disable the OSPF routing instance in the trust-vr. OSPF stops transmitting and processing OSPF packets on all OSPF-enabled interfaces in the trust-vr.

WebUI

Network > Routing > Virtual Routers (trust-vr) > Edit > Edit OSPF Instance: Uncheck **OSPF Enabled**, then click **OK**.

Network > Routing > Virtual Routers (trust-vr) > Edit > Delete OSPF Instance, then click **OK** at the confirmation prompt.

CLI

```
unset vrouter trust-vr protocol ospf
deleting OSPF instance, are you sure? y/[n]
save
```

NOTE: In the CLI, you confirm the deletion of the OSPF instance.

Creating and Deleting an OSPF Area

Areas reduce the amount of routing information that needs to be passed through the network because an OSPF router maintains a link-state database only for the area it resides in. No link-state information is maintained for networks or routers outside the area.

All areas must be connected to area 0, which is created when you configure an OSPF routing instance on the virtual router. If you need to create an additional OSPF area, you can optionally define the area as a stub area or not-so-stubby area. See “Areas” on page 1270 for more information about these types of areas.

Table 86 on page 1275 lists area parameters, with descriptions of each parameter, and gives the default value for each.

Table 86: OSPF Areas Parameters and Default Values

Area Parameter	Description	Default Value
Metric for default route	(NSSA and stub areas only) Specifies the metric for the default route advertisement	1
Metric type for the default route	(NSSA area only) Specifies the external metric type (1 or 2) for the default route	1
No summary	(NSSA and stub areas only) Specifies that summary LSAs are not advertised into the area	Summary LSAs are advertised into the area
Range	(All areas) Specifies a range of IP addresses to be advertised in summary LSAs and whether they are advertised or not	—

Creating an OSPF Area

In the following example, you create an OSPF area with an area ID of 10.

WebUI

Network > Routing > Virtual Routers > Edit (trust-vr) > Edit OSPF Instance > Area: Enter the following, then click **OK**:

Area ID: 10
Type: normal (select)
Action: Add

CLI

```
set vrouter trust-vr protocol ospf area 10
save
```

Deleting an OSPF Area

Before you can delete an OSPF area, you must disable the OSPF process for the VR. In the following example, you stop the OSPF process and then delete an OSPF area with an area ID of 10.

WebUI

Network > Routing > Virtual Routers (trust-vr) > Edit > Edit OSPF Instance: Deselect **OSPF Enabled**, then click **OK**.

Network > Routing > Virtual Routers > Edit (trust-vr) > Edit OSPF Instance > Area: Click **Remove**.

CLI

```
unset vrouter trust-vr protocol ospf enable
unset vrouter trust-vr protocol ospf area 0.0.0.10
save
```

Assigning Interfaces to an OSPF Area

Once an area is created, you can assign one or more interfaces to the area, using either the WebUI or the CLI **set interface** command.

Assigning Interfaces to Areas

In the following example, you assign the ethernet1 interface to OSPF area 10 and assign the ethernet3 interface to OSPF area 0.

WebUI

Network > Routing > Virtual Routers > Edit (trust-vr) > Edit OSPF Instance > Area > Configure (Area 10): Use the **Add** button to move the ethernet1 interface from the Available Interface(s) column to the Selected Interfaces column. Click **OK**.

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance > Area > Configure (Area 0): Use the **Add** button to move the ethernet3 interface from the Available Interface(s) column to the Selected Interfaces column. Click **OK**.

CLI

```
set interface ethernet1 protocol ospf area 10
set interface ethernet3 protocol ospf area 0
save
```

Configuring an Area Range

By default, an ABR does not aggregate routes sent from one area to another area. Configuring an area range allows a group of subnets in an area to be consolidated into a single network address to be advertised in a single summary link advertisement to other areas. When you configure an area range, you specify whether to advertise or withhold the defined area range in advertisements.

In the following example, you define the following area ranges for area 10:

- 10.1.1.0/24 to be advertised
- 10.1.2.0/24 not to be advertised

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance > Area > Configure (0.0.0.10): Enter the following in the Area Range section, then click **Add**:

```
IP / Netmask: 10.1.1.0/24
Type: (select) Advertise
```

Enter the following in the Area Range section, then click **Add**:

```
IP / Netmask: 10.1.2.0/24
Type: (select) No Advertise
```

CLI

```
set vrouter trust-vr protocol ospf area 10 range 10.1.1.0/24 advertise
set vrouter trust-vr protocol ospf area 10 range 10.1.2.0/24 no-advertise
save
```

Enabling OSPF on Interfaces

By default, OSPF is disabled on all interfaces in the virtual router (VR). You must explicitly enable OSPF on an interface after you assign the interface to an area. When you disable OSPF on an interface, OSPF does not transmit or receive packets on the specified interface, but interface configuration parameters are preserved.

NOTE: If you disable the OSPF routing instance in the VR (see "Removing an OSPF Instance" on page 1274), OSPF stops transmitting and processing packets on all OSPF-enabled interfaces in the VR.

Enabling OSPF on Interfaces

In this example, you enable the OSPF routing instance on the ethernet1 interface (which was previously assigned to area 10) and on the ethernet3 interface (which was previously assigned to area 0).

WebUI

Network > Interfaces > Edit (for ethernet1) > OSPF: Select **Enable Protocol OSPF**, then click **Apply**.

Network > Interfaces > Edit (for ethernet3) > OSPF: Select **Enable Protocol OSPF**, then click **Apply**.

CLI

```
set interface ethernet1 protocol ospf enable
set interface ethernet3 protocol ospf enable
save
```

Disabling OSPF on an Interface

In this example, you disable the OSPF routing instance only on the ethernet1 interface. Any other interfaces in the trust-vr virtual router (VR) on which you have enabled OSPF are still able to transmit and process OSPF packets.

WebUI

Network > Interfaces > Edit (for ethernet1) > OSPF: Clear **Enable Protocol OSPF**, then click **Apply**.

CLI

```
unset interface ethernet1 protocol ospf enable
save
```

NOTE: If you disable the OSPF routing instance in the VR, OSPF stops transmitting and processing packets on all OSPF-enabled interfaces in the VR (see "Removing an OSPF Instance" on page 1274).

Verifying the Configuration

You can view the configuration you entered for the trust-vr by executing the following CLI command at the prompt:

```
device->  get vrouter trust-vr protocol ospf config
VR: trust-vr RouterId: 10.1.1.250
----------------------------------
set protocol ospf
set enable
set area 0.0.0.10 range 10.1.1.0 255.255.255.0 advertise
set area 0.0.0.10 range 10.1.2.0 255.255.255.0 no-advertise
set interface ethernet1 protocol ospf area 0.0.0.10
set interface ethernet1 protocol ospf enable
set interface ethernet3 protocol ospf area 0.0.0.0
set interface ethernet3 protocol ospf enable
```

You can verify that OSPF is running on the virtual router with the **get vrouter trust-vr protocol ospf** command.

```
device->  get vrouter trust-vr protocol ospf
VR: trust-vr RouterId: 10.1.1.250
----------------------------------
OSPF enabled
Supports only single TOS(TOS0) route
Internal Router
Automatic vlink creation is disabled
Numbers of areas is 2
Number of external LSA(s) is 0
SPF Suspend Count is 10 nodes
Hold time between SPFs is 3 second(s)
Advertising default-route lsa is off
Default-route discovered by ospf will be added to the routing table
RFC 1583 compatibility is disabled.
Hello packet flooding protection is not enabled
LSA flooding protection is not enabled
Area 0.0.0.0
        Total number of interfaces is 1, Active number of interfaces is 1
        SPF algorithm executed 2 times
        Number of LSA(s) is 1
Area 0.0.0.10
        Total number of interfaces is 1, Active number of interfaces is 1
        SPF algorithm executed 2 times
        Number of LSA(s) is 0
```

The highlighted areas show that OSPF is running and verify the active OSPF areas and active interfaces in each OSPF area.

NOTE: We recommend that you explicitly assign a router ID rather than use the default value. For information on setting a router ID, see "Routing" on page 1235.

You can verify that OSPF is enabled on the interfaces and see the state of the interfaces with the **get vrouter trust-vr protocol ospf interface** command.

```
device->  get vrouter trust-vr protocol ospf interface
VR: trust-vr RouterId: 10.1.1.250
---------------------------------
Interface   IpAddr        NetMask       AreaId        Status   State
-------------------------------------------------------------------------------
ethernet3   2.2.2.2       255.255.255.0 0.0.0.0       enabled Designated Router
ethernet1   10.1.1.1      255.255.255.0 0.0.0.10      enabled Up
```

You can configure the priority of the virtual router to be elected the Designated Router (DR) or the Backup Designated Router (BDR). In the example above, the State column lists the priority of the virtual router.

You can verify that the OSPF routing instance on the security device has established adjacencies with OSPF neighbors with the **get vrouter trust-vr protocol ospf neighbor** command.

```
device->  get vrouter trust-vr protocol ospf neighbor
VR: trust-vr RouterId: 10.1.1.250
---------------------------------
                Neighbor(s) on interface ethernet3 (Area 0.0.0.0)
IpAddr/If Index RouterId        Priority State    Options
-------------------------------------------------------------------------------
2.2.2.2        2.2.2.250              1 Full     E
Neighbor(s) on interface ethernet1 (Area 0.0.0.10)
IpAddr/If Index RouterId        Priority State    Options
-------------------------------------------------------------------------------
10.1.1.1       10.1.1.252             1 Full     E
```

In the State column in the example above, Full indicates full OSPF adjacencies with neighbors.

Redistributing Routes into Routing Protocols

Route redistribution is the exchange of route information between routing protocols. For example, you can redistribute the following types of routes into the OSPF routing instance in the same virtual router:

- Routes learned from BGP or RIP
- Directly connected routes
- Imported routes
- Statically configured routes

When you configure route redistribution, you must first specify a route map to filter the routes that are redistributed. For more information about creating route maps for route redistribution, see “Routing” on page 1235.

In the following example, you redistribute a route that originated from a BGP routing domain into the current OSPF routing domain. Both the CLI and WebUI examples assume that you previously created a route map called add-bgp.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance > Redistributable Rules: Enter the following, then click **Add**:

Route Map: add-bgp
Protocol: BGP

CLI

```
set vrouter trust-vr protocol ospf redistribute route-map add-bgp protocol bgp
save
```

Summarizing Redistributed Routes

In large internetworks where thousands of network addresses can exist, some routers might become overly congested with route information. Once you redistribute a series of routes from an external protocol to the current OSPF routing instance, you can bundle the routes into one generalized or summarized network route. By summarizing multiple addresses, you enable a series of routes to be recognized as one route, simplifying the lookup process.

An advantage to using route summarization in a large, complex network is that it can isolate topology changes from other routers. For example, if a specific link in a given domain is intermittently failing, the summary route would not change, so no router external to the domain would need to repeatedly modify its routing table due to the link failure.

In addition to creating fewer entries in the routing tables on the backbone routers, route summarization prevents the propagation of LSAs to other areas when one of the summarized networks goes down or comes up. You can also summarize inter-area routes or external routes.

Sometimes a summarized route can create opportunities for loops to occur. You can configure a route to a NULL interface to avoid loops. An example of creating a summarized route and then an example of setting a NULL interface follows this section.

Summarizing Redistributed Routes

In this example, you redistribute BGP routes into the current OSPF routing instance. You then summarize the set of imported routes under the network address 2.1.1.0/24.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance > Redistributable Rules: Enter the following, then click **Add**:

Route Map: add-bgp
Protocol: BGP

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance > Summary Import: Enter the following, then click **Add**:

IP/Netmask: 2.1.1.0/24

CLI

```
set vrouter trust-vr protocol ospf redistribute route-map add-bgp protocol bgp
set vrouter trust-vr protocol ospf summary-import ip 2.1.1.0/24
save
```

Global OSPF Parameters

This section describes optional OSPF global parameters that you can configure at the virtual router (VR) level. When you configure an OSPF parameter at the VR level, the parameter setting affects operations on all OSPF-enabled interfaces. You can modify global parameter settings through the OSPF routing protocol context in the CLI or by using the WebUI.

Table 87 on page 1282 lists global OSPF parameters and their default values.

Table 87: Global OSPF Parameters and Default Values

OSPF Global Parameters	Description	Default Value
Advertise default route	Specifies that an active default route (0.0.0.0/0) in the VR route table is advertised into all OSPF areas. You can also specify the metric value or whether the route's original metric is preserved, and the metric type (ASE type 1 or type 2). You can also specify that the default route is always advertised.	Default route is not advertised.
Reject default route	Specifies that any default route learned in OSPF is not added to the route table.	Default route learned in OSPF is added to the route table.
Automatic virtual link	Specifies that the VR is to automatically create a virtual link when it cannot reach the OSPF backbone.	Disabled.
Maximum hello packets	Specifies the maximum number of OSPF hello packets that the VR can receive in a hello interval.	10.
Maximum LSA packets	Specifies the maximum number of OSPF LSA packets that the VR can receive within the specified number of seconds.	No default.
RFC 1583 compatibility	Specifies that the OSPF routing instance is compatible with RFC 1583, an earlier version of OSPF.	OSPF version 2, as defined by RFC 2328.

Table 87: Global OSPF Parameters and Default Values *(continued)*

OSPF Global Parameters	Description	Default Value
Equal cost multipath routing (ECMP)	Specifies the maximum number of paths (1-4) to use for load-balancing with destinations that have multiple equal cost paths. See "Configuring Equal Cost Multipath Routing" on page 1259.	Disabled (1).
Virtual link configuration	Configures the OSPF area and router ID for the virtual link. You can optionally configure the authentication method, hello interval, retransmit interval, transmit delay, or neighbor dead interval for the virtual link.	No virtual link configured.

Advertising the Default Route

The default route, 0.0.0.0/0, matches every destination network in a routing table, although a more specific prefix overrides the default route.

In this example, you advertise the default route of the current OSPF routing instance.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance: Select **Advertising Default Route Enable**, then click **OK**.

NOTE: In the WebUI, the default metric (1) 62 must be manually entered, and the default metric-type is ASE type 1.

CLI

```
set vrouter trust-vr protocol ospf advertise-default-route metric 1 metric-type 1
save
```

Virtual Links

Although all areas should be connected directly to the backbone, sometimes you need to create a new area that cannot be physically connected to the backbone area. To solve this problem, you can configure a virtual link. A virtual link provides a remote area with a logical path to the backbone through another area.

You must configure the virtual link on the routers on both ends of the link. To configure a virtual link on the security device, you need to define:

- The ID of the OSPF area through which the virtual link will pass. You cannot create a virtual link that passes through the backbone area or a stub area.
- The ID of the router at the other end of the virtual link.

Table 88 on page 1284 lists optional parameters for virtual links.

Table 88: Optional Parameters for Virtual Links

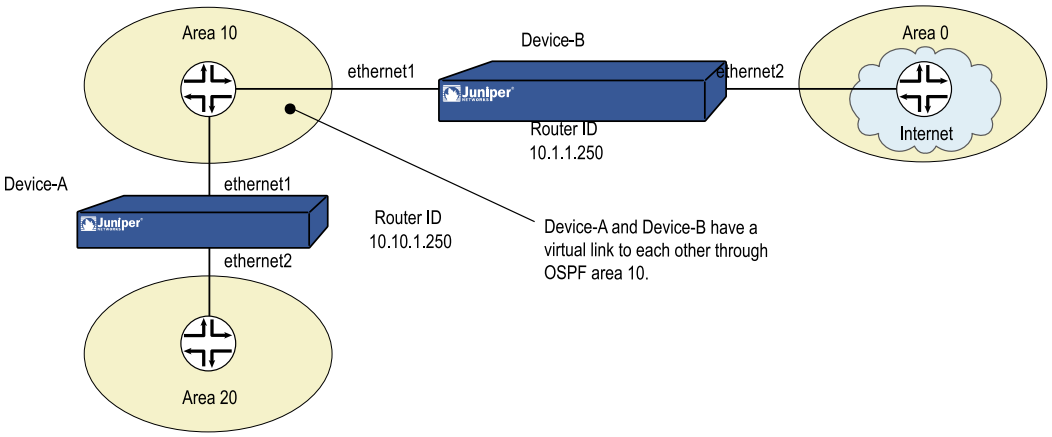
Virtual Link Parameter	Description	Default Value
Authentication	Specifies either clear text password or MD5 authentication.	No authentication
Dead interval	Specifies the number of seconds that elapses with no response from an OSPF neighbor before the neighbor is determined to be not running.	40 seconds
Hello interval	Specifies the number of seconds between OSPF hellos.	10 seconds
Retransmit interval	Specifies the number of seconds that elapses before the interface resends an LSA to a neighbor that did not respond to the original LSA.	5 seconds
Transmit delay	Specifies the number of seconds between transmissions of link-state update packets sent on an interface.	1 second

Creating a Virtual Link

In the following example, you create a virtual link through OSPF area 10 from Device-A with router ID 10.10.1.250 to Device-B with router ID 10.1.1.250. See "Routing" on page 1235 for information on how to configure router IDs on security devices.) You also configure the virtual link for a transit delay of 10 seconds. On each security device, you need to identify the router ID of the device at the other end of the virtual link.

Figure 328 on page 1284 shows the example network setup for a virtual link.

Figure 328: Creating a Virtual Link



NOTE: You must enable OSPF on both interfaces of each device and make sure that OSPF is running on the interfaces in devices A and B before the virtual link becomes active.

WebUI (Device-A)

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance > Virtual Link: Enter the following, then click **Add**:

Area ID: 10 (select)
Router ID: 10.1.1.250

> Configure: In the Transmit Delay field, type **10**, then click **OK**.

CLI (Device-A)

```
set vrouter trust-vr protocol ospf vlink area-id 10 router-id 10.1.1.250
set vrouter trust-vr protocol ospf vlink area-id 10 router-id 10.1.1.250 transit-delay
10
save
```

NOTE: In the CLI, you must first create the virtual link before you can configure any optional parameters for the virtual link. Thus, in the CLI example above, you must issue two separate commands to create and then configure the virtual link.

WebUI (Device-B)

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance > Virtual Link: Enter the following, then click **Add**:

Area ID: 10
Router ID: 10.10.1.250

> Configure: In the Transmit Delay field, type **10**, then click **OK**.

CLI (Device-B)

```
set vrouter trust-vr protocol ospf vlink area-id 10 router-id 10.10.1.250
set vrouter trust-vr protocol ospf vlink area-id 10 router-id 10.10.1.250 transit-delay
 10
save
```

Creating an Automatic Virtual Link

You can direct a virtual router (VR) to automatically create a virtual link for instances when it cannot reach the network backbone. Having the VR automatically create virtual links replaces the more time-consuming process of creating each virtual link manually. In the following example, you configure automatic virtual link creation.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance: Select **Automatically Generate Virtual Links**, then click **OK**.

CLI

```
set vrouter trust-vr protocol ospf auto-vlink
save
```

Setting OSPF Interface Parameters

This section describes OSPF parameters that you configure at the interface level. When you configure an OSPF parameter at the interface level, the parameter setting affects the OSPF operation only on the specific interface. You can modify interface parameter settings with **interface** commands in the CLI or by using the WebUI.

Table 89 on page 1286 lists optional OSPF interface parameters and their default values.

Table 89: Optional OSPF Interface Parameters and Default Values

OSPF Interface Parameter	Description	Default Value
Authentication	Specifies either clear text password or message digest 5 (MD5) authentication to verify OSPF communication on the interface. A clear text password requires password string of up to 8 digits, and an MD5 authentication password requires a password string of up to 16 digits. The MD5 password also requires that you configure key strings.	No authentication used.
Cost	Specifies the metric for the interface. The cost associated with an interface depends upon the bandwidth of the link to which the interface is connected. The higher the bandwidth, the lower (more desirable) the cost value.	1 for a 100MB or more link 10 for a 10MB link 100 for a 1MB link
Dead interval	Specifies the number of seconds that elapses with no response from an OSPF neighbor before OSPF determines the neighbor is not running.	40 seconds.
Hello interval	Specifies the interval, in seconds, at which OSPF sends out hello packets to the network.	10 seconds.
Link type	Specifies a tunnel interface as a point-to-point link or as a point-to-multipoint link. See "Point-to-Multipoint Tunnel Interface" on page 1293.	Ethernet interfaces are treated as broadcast interfaces. Tunnel interfaces bound to OSPF areas are point-to-point by default.

Table 89: Optional OSPF Interface Parameters and Default Values *(continued)*

Neighbor list	Specifies subnets, in the form of an access list, on which OSPF neighbors reside that are eligible to form adjacencies.	None (adjacencies are formed with all neighbors on the interface).
Passive interface	Specifies that the IP address of the interface is advertised into the OSPF domain as an OSPF route and not as an external route, but the interface does not transmit or receive OSPF packets. This option is useful when BGP is also enabled on the interface.	OSPF-enabled interfaces transmit and receive OSPF packets.
Priority	Specifies the priority for the virtual router to be elected the Designated Router or Backup Designated Router. The router with the larger priority value has the best chance (although not guaranteed) chance of being elected.	1.
Retransmit interval	Specifies the number of seconds that elapses before the interface resends an LSA to a neighbor that did not respond to the original LSA.	5 seconds.
Transit delay	Specifies the number of seconds between transmissions of link-state update packets sent on the interface.	1 second.
Demand circuit	(Tunnel interfaces only) Configures a tunnel interface as a demand circuit, per RFC 1793. See “Creating an OSPF Demand Circuit on a Tunnel Interface” on page 1292.	Disabled.
Reduce flooding	Specifies the reduction of LSA flooding on a demand circuit.	Disabled.
Ignore MTU	Specifies that any mismatches in maximum transmission unit (MTU) values between the local and remote interfaces that are found during OSPF database negotiations are ignored. This option should only be used when the MTU on the local interface is lower than the MTU on the remote interface.	Disabled.

NOTE: To form adjacencies, all OSPF routers in an area must use the same hello, dead, and retransmit interval values.

In the following example, you configure the following OSPF parameters for the ethernet1 interface:

- Increase the interval between OSPF hello messages to 15 seconds.
- Increase the interval between OSPF retransmissions to 7 seconds.
- Increase the interval between LSA transmissions to 2 seconds.

WebUI

Network > Interfaces > Edit (for ethernet1) > OSPF: Enter the following, then click **Apply**:

Hello Interval: 15
Retransmit Interval: 7
Transit Delay: 2

CLI

```
set interface ethernet1 protocol ospf hello-interval 15
set interface ethernet1 protocol ospf retransmit-interval 7
set interface ethernet1 protocol ospf transit-delay 2
save
```

Security Configuration

This section describes possible security problems in the OSPF routing domain and methods of preventing attacks.

NOTE: To make OSPF more secure, you should configure all routers in the OSPF domain to be at the same security level. Otherwise, a compromised OSPF router can bring down the entire OSPF routing domain.

Authenticating Neighbors

An OSPF router can be easily spoofed, since LSAs are not encrypted and most protocol analyzers provide decapsulation of OSPF packets. Authenticating OSPF neighbors is the best way to fend off these types of attacks.

OSPF provides both simple password and MD5 authentication to validate OSPF packets received from neighbors. All OSPF packets received on the interface that are not authenticated are discarded. By default, there is no authentication enabled on any OSPF interface.

MD5 authentication requires that the same key be used for both the sending and receiving OSPF routers. You can specify more than one MD5 key on the security device; each key is paired with a key identifier. If you configure multiple MD5 keys on the security device, you can then select the key identifier of the key that is to be used for authenticating communications with the neighbor router. This allows MD5 keys on pairs of routers to be changed periodically with minimal risk of packets being dropped.

Configuring a Clear-Text Password

In this example, you set a clear-text password 12345678 for OSPF on interface ethernet1.

WebUI

Network > Interfaces > Edit (for ethernet1) > OSPF: Enter the following, then click **Apply**:

Password: (select), 12345678

CLI

```
set interface ethernet1 protocol ospf authentication password 12345678
save
```

Configuring an MD5 Password

In the following example, you set the two different MD5 keys on interface ethernet1 and select one of the keys to be the active key. Each MD5 key can be 16 characters long. The key-id number must be between 0 and 255. The default key-id is 0 so you do not have to specify the key-id for the first MD5 key you enter.

WebUI

Network > Interfaces > Edit (for ethernet1) > OSPF: Enter the following, then click **Apply**:

Authentication:
MD5 Keys: (select)
1234567890123456
9876543210987654
Key ID: 1
Preferred: (select)

CLI

```
set interface ethernet1 protocol ospf authentication md5 1234567890123456
set interface ethernet1 protocol ospf authentication md5 9876543210987654 key-id
 1
set interface ethernet1 protocol ospf authentication md5 active-md5-key-id 1
save
```

Configuring an OSPF Neighbor List

Multi-access environments can allow devices, including routers, to be connected into a network relatively easily. This can cause stability or performance issues if the connected device is not reliable.

By default, the OSPF routing instance on a ScreenOS virtual router (VR) forms adjacencies with all OSPF neighbors communicating on an OSPF-enabled interface. You can limit the devices on an interface that can form adjacencies with the OSPF routing instance by defining a list of subnets that contain eligible OSPF neighbors. Only hosts or routers that reside in the specified subnets can form adjacencies with

the OSPF routing instance. To specify the subnets that contain eligible OSPF neighbors, define an access list for the subnets at the VR level.

In this example, you configure an access list that permits the hosts on subnet 10.10.10.130/27. You then specify the access list to configure eligible OSPF neighbors.

WebUI

Network > Routing > Virtual Router (trust-vr) > Access List > New: Enter the following, then click **OK**:

Access List ID: 4
Sequence No.: 10
IP/Netmask: 10.10.10.130/27
Action: Permit (select)

Network > Interfaces > Edit (for ethernet1) > OSPF: Enter the following, then click **Apply**:

Neighbor List: 4

CLI

```
set vrouter trust-vr access-list 4
set vrouter trust-vr access-list 4 permit ip 10.10.10.130/27 10
set interface ethernet1 protocol ospf neighbor-list 4
save
```

Rejecting Default Routes

In a Route Detour Attack, a router injects a default route (0.0.0.0/0) into the routing domain in order to detour packets to itself. The router can then either drop the packets, causing service disruption, or it can obtain sensitive information in the packets before forwarding them. On Juniper Networks security devices, OSPF by default accepts any default routes that are learned in OSPF and adds the default route to the routing table.

In the following example, you specify that a default route not be learned from OSPF.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance: Select the Do Not Add Default-route Learned in OSPF check box, then click **OK**.

CLI

```
set vrouter trust-vr protocol ospf reject-default-route
save
```

Protecting Against Flooding

A malfunctioning or compromised router can flood its neighbors with OSPF hello packets or with LSAs. Each router retrieves information from the LSAs sent by other routers on the network to distill path information for the routing table. LSA flood protection enables you to manage the number of LSAs entering the virtual router (VR). If the VR receives too many LSAs, the router fails because of LSA flooding. An LSA attack happens when a router generates an excessive number of LSAs in a short period of time, thus keeping other OSPF routers in the network busy running the SPF algorithm.

On VRs using ScreenOS, you can configure both the maximum number of hello packets per hello interval and the maximum number of LSAs that can be received on an OSPF interface within a certain interval. Packets that exceed a configured threshold are dropped. By default, the OSPF hello packet threshold is 10 packets per hello interval (the default hello interval for an OSPF interface is 10 seconds). There is no default LSA threshold; if you do not set an LSA threshold, all LSAs are accepted.

Configuring the Hello Threshold

In the following example, you configure a threshold of 20 packets per hello interval. The hello interval, which is configurable on each OSPF interface, is not changed from its default of 10 seconds.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance: Enter the following, then click **OK**:

```
Prevent Hello Packet Flooding Attack: On
Max Hello Packet: 20
```

CLI

```
set vrouter trust-vr protocol ospf hello-threshold 20
save
```

Configuring the LSA Threshold

In this example, you create an OSPF LSA flood attack threshold of 10 packets per 20 seconds.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPF Instance: Enter the following, then click **OK**:

```
LSA Packet Threshold Time: 20
Maximum LSAs: 10
```

CLI

```
set vrouter trust-vr protocol ospf lsa-threshold 20 10
save
```

Enabling Reduced Flooding

You can enable the reduce flooding feature to suppress LSA flooding on point-to-point interfaces—such as serial, tunnel, or Asynchronous Digital Subscriber Line (ADSL)—or broadcast interfaces—such as Ethernet. In the following example, you enable periodic LSA suppression without affecting hello packet flow for the tunnel.1 interface.

WebUI

Network > Interfaces > Edit (for tunnel.1) > OSPF: Enter the following, then click **Apply**:

Reduce Flooding: (select)

CLI

```
set interface tunnel.1 protocol ospf reduce-flooding
save
```

Creating an OSPF Demand Circuit on a Tunnel Interface

OSPF demand circuits, as defined in RFC 1793, are network segments where connect time or usage affects the cost of using such connections. On a demand circuit the traffic generated by OSPF needs to be limited to changes in network topology. On Juniper Networks security devices, only point-to-point interfaces, such as serial, tunnel, or ADSL Asynchronous Digital Subscriber Line (ADSL) interfaces, can be demand circuits; and, for proper operation, both ends of the tunnel must be manually configured as demand circuits.

On tunnel interfaces that are configured as demand circuits, the security device suppresses sending OSPF hello packets and periodic refreshment of LSA flooding to decrease overhead. After the OSPF neighbor reaches Full state (Hellos match and router and network LSAs reflect all adjacent neighbors), the security device suppresses periodic hello packets and LSA refreshes. The security device only floods LSAs in which content has changed.

In the following example, you configure the tunnel.1 interface as a demand circuit.

NOTE: You need to configure the remote peer's tunnel interface as a demand circuit. However, you do not need to configure reduced LSA flooding on the remote peer.

WebUI

Network > Interfaces > Edit > OSPF: Enter the following, then click **Apply**:

Demand Circuit: (select)

CLI

```
set interface tunnel.1 protocol ospf demand-circuit
save
```

Point-to-Multipoint Tunnel Interface

When you bind a tunnel interface to an OSPF area on a security device, you create a point-to-point OSPF tunnel by default. The point-to-point tunnel interface can form an adjacency with only one OSPF router at the remote end. If the local tunnel interface is to be bound to multiple tunnels, you must configure the local tunnel interface as a point-to-multipoint interface and *disable* the route-deny feature on the tunnel interface.

NOTE: You must configure a tunnel interface as a point-to-multipoint interface before enabling OSPF on the interface. Once you configure the interface as a point-to-multipoint interface, you cannot configure it as a demand circuit (see "Creating an OSPF Demand Circuit on a Tunnel Interface" on page 1292). You can, however, configure the interface for reduced LSA flooding.

For an example of binding multiple tunnels to a tunnel interface, see "Binding Automatic Route and NHTB Table Entries" on page 1008. The following sections include examples for:

- Setting the link-type (see "Setting the OSPF Link-Type" on page 1293)
- Setting the route-deny feature (see "Disabling the Route-Deny Restriction" on page 1294)
- Configuring a network with a point-to-multipoint tunnel interface (see "Creating a Point-to-Multipoint Network" on page 1294)

Setting the OSPF Link-Type

If you intend to form OSPF adjacencies on multiple tunnels, then you need to set the link type as Point-to-Multipoint (p2mp).

In the following example, you set the link type of tunnel.1 to point-to-multipoint (p2mp) to match your networking needs.

WebUI

Network > Interface (Edit) > OSPF: Select Point-to-Multipoint from the Link Type radio button list.

CLI

```
set interface tunnel.1 protocol ospf link-type p2mp
save
```

Disabling the Route-Deny Restriction

By default, the security device can potentially send and receive packets on the same interface unless explicitly configured not to send and receive packets on the same interface. In a point-to-multipoint scenario, you might desire this behavior. To configure the security device to send and receive on the same interface, you must disable the route-deny restriction. In this example, you disable the route-deny restriction through the CLI on the point-to multipoint tunnel interface tunnel.1.

WebUI

NOTE: You must use the CLI to set the route-deny feature.

CLI

```
unset interface tunnel.1 route-deny
save
```

Creating a Point-to-Multipoint Network

Figure 329 on page 1295 shows a medium-sized enterprise that has a central office (CO) in San Francisco and remote sites in Chicago, Los Angeles, Montreal, and New York. Each office has a single security device.

The following are the configuration requirements particular to the security device in the CO:

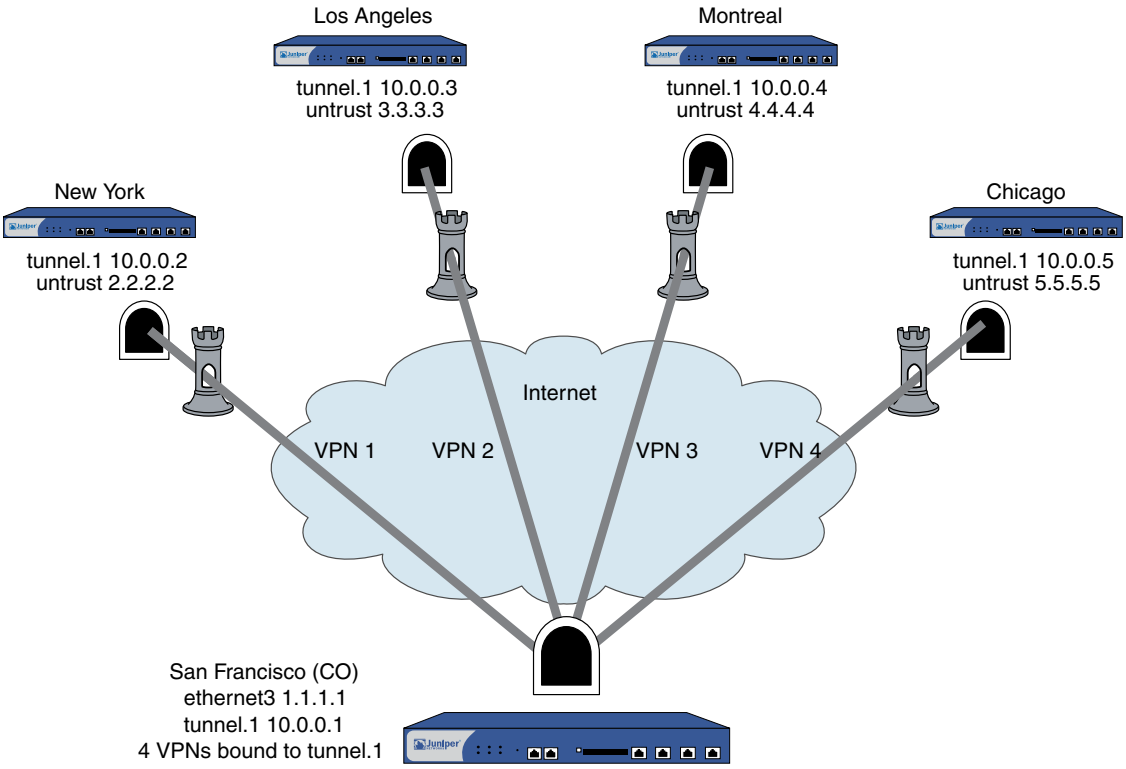
1. Configure the VR to run an instance of OSPF, enable OSPF, and then configure the tunnel.1 interface.
2. Configure the four VPNs and bind them to the tunnel.1 interface.

The following are the configuration requirements particular to the remote security devices:

1. Configure the VR to run an instance of OSPF and enable OSPF and then configure the tunnel.1 interface.
2. Configure the VPN and bind it to tunnel.1 interface.

Timer values for all of the devices must match for adjacencies to form. Figure 329 on page 1295 shows the described network scenario.

Figure 329: Point-to-MultiPoint Network Example



In Figure 329 on page 1295, four VPNs originate from the San Francisco security device and radiate out to remote offices in New York, Los Angeles, Montreal, and Chicago.

In this example, you configure the following settings on the CO security device:

1. Security Zone and Interfaces
2. VPN
3. Routes and OSPF

To complete the network configuration, you configure the following settings on each of the four remote office security devices:

1. Interface and OSPF
2. VPN
3. Policy

NOTE: The WebUI procedures are abbreviated due to the length of the example. The CLI portion of the example is complete. You can refer ahead to the CLI portion for the exact settings and values to use.

WebUI (Central Office Device)

1. **Security Zone and Interfaces**

   Network > Interfaces > **Click** New Tunnel IF and continue to the Configuration page.

   Network > Interfaces > Edit (for ethernet3) and configure IP address and zone.

   Network > Interface > Edit (for tunnel.1) > OSPF: Select Point-to-Multipoint from the Link Type radio button list.

2. **VPN**

   VPNs > AutoKey Advanced > Gateway

3. **Routes and OSPF**

   Network > Routing > Virtual Routers > Click **Edit** for the virtual router and configure OSPF parameters.

CLI (Central Office Device)

1. **Security Zone and Interfaces**

```
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.10.10.1/24
```

2. **VPN**

```
set ike gateway gw1 address 2.2.2.2 main outgoing-interface ethernet3 preshare
 ospfp2mp proposal pre-g2-3des-sha
set ike gateway gw2 address 3.3.3.3 main outgoing-interface ethernet3 preshare
 ospfp2mp proposal pre-g2-3des-sha
set ike gateway gw3 address 4.4.4.4 main outgoing-interface ethernet3 preshare
 ospfp2mp proposal pre-g2-3des-sha
set ike gateway gw4 address 5.5.5.5 main outgoing-interface ethernet3 preshare
 ospfp2mp proposal pre-g2-3des-sha
set vpn vpn1 gateway gw1 no-replay tunnel idletime 0 proposal g2-esp-3des-sha
set vpn vpn1 monitor rekey
set vpn1 id 1 bind interface tunnel.1
set vpn vpn2 gateway gw2 no-replay tunnel idletime 0 proposal g2-esp-3des-sha
set vpn vpn2 monitor rekey
set vpn2 id 2 bind interface tunnel.1
set vpn vpn3 gateway gw3 no-replay tunnel idletime 0 proposal g2-esp-3des-sha
```

```
set vpn vpn3 monitor rekey
set vpn3 id 3 bind interface tunnel.1
set vpn vpn4 gateway gw4 no-replay tunnel idletime 0 proposal g2-esp-3des-sha
set vpn vpn4 monitor rekey
set vpn4 id 4 bind interface tunnel.1
```

3. **Routes and OSPF**

```
set vrouter trust router-id 10
set vrouter trust protocol ospf
set vrouter trust protocol ospf enable
set interface tunnel.1 protocol ospf area 0
set interface tunnel.1 protocol ospf enable
set interface tunnel.1 protocol ospf link-type p2mp
unset interface tunnel.1 route-deny
save
```

NOTE: By default route-deny is disabled. However, if you enabled the route-deny feature at some point, then you need to disable the feature for the proper operation of the point-to-multipoint tunnel interface.

You can follow these steps to configure the remote office security device. Juniper Networks security devices learn about neighbors through LSAs.

To complete the configuration shown in Figure 329 on page 1295, you must repeat the following section for each remote device and change the IP addresses, gateway names and VPN names and set policies to match the network needs. For each remote site, the trust and untrust zones change.

NOTE: The WebUI procedures are abbreviated due to the length of the example. The CLI portion of the example is complete. You can refer ahead to the CLI portion for the exact settings and values to use.

WebUI (Remote Office Device)

1. **Interface and OSPF**

   Network > Interfaces > Click **New Tunnel IF** and continue to the Configuration page.

2. **VPN**

   VPNs > AutoKey Advanced > Gateway

3. **Policy**

   Policies (from All zones to All zones) > Click **New**.

CLI (Remote Office Device)

1. **Interface and OSPF**

```
set vrouter trust protocol ospf
set vrouter trust protocol ospf enable
set interface untrust ip 2.2.2.2/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.0.0.2/24
set interface tunnel.1 protocol ospf area 0
set interface tunnel.1 protocol ospf enable
```

2. **VPN**

```
set ike gateway gw1 address 1.1.1.1/24 main outgoing-interface untrust preshare
 ospfp2mp proposal pre-g2-3des-sha
set vpn vpn1 gateway gw1 no-replay tunnel idletime 0 proposal g2-esp-3des-sha
set vpn vpn1 monitor rekey
set vpn vpn1 id 1 bind interface tunnel.1
```

3. **Policy (configure as required)**

```
set policy id 1 from trust to untrust any any any permit
set policy id 2 from untrust to trust any any any permit
save
```

You can view the new changes with the **get vrouter** vrouter **protocol ospf config** command.

OSPFv3

OSPFv3 adds support for IPv6 in the Open Shortest Path First (OSPF) routing protocol, as detailed in RFC 2740. Most configuration and operational commands function essentially the same as in OSPFv2:

- All OSPFv3 operational and configuration commands include the identifier ospfv3 in place of the familiar ospf option. For example, get protocol ospf database in OSPFv2 becomes get protocol ospfv3 database in OSPFv3.
- OSPFv3 Router IDs, Area IDs, and LSA link-state IDs remain at the OSPFv2 IPv4 size of 32 bits.
- All the optional capabilities in OSPFv2 for IPv4, such as STUB areas, are supported in OSPFv3 for IPv6.

However, there are many significant changes to note about OSPFv3 for IPv6:

- Router link-state advertisements (LSAs) and Network LSAs no longer carry prefix information. In OSPFv3, these LSAs only carry topology information.

NOTE: Because addressing information in the LSA header, Router LSA, and Network LSA (Type 2) has been removed, the OSPFv3 protocol is designed to be network protocol independent.

- New and modified LSAs have been created to handle the flow of IPv6 addresses and prefixes in an OSPFv3 network. As a result, some show command output appears in a different format for OSPFv3. The LSAs that have been modified are:
  - Inter-area-Prefix LSA—This replaces the Network Summary or Type 3 LSA.
  - Inter-area Router LSA—This replaces the Autonomous System Boundary Router (ASBR) Summary or Type 4 LSA.
- OSPFv3 now runs on a per-link basis, instead of on a per-IP-subnet basis.
- IPv6 link-local addresses are used for OSPFv3 neighbor exchanges (except over virtual links).
- The flooding scope for LSAs has been generalized into three categories for OSPFv3:
  - Link-local scope—The OSPFv3 packet is flooded to the members of a link.
  - Area scope—The OSPFv3 packet is flooded to all members of an OSPFv3 area.
  - AS scope—The OSPFv3 packet is flooded to all members of an AS.
- Authentication has been removed from the OSPFv3 protocol itself and relies on the authentication header (AH) and Encapsulating Security Payload (ESP) portions of the IP Security (IPSec) protocol for all authentication tasks in IPv6.
- Neighboring routers are always identified by the 32-bit router ID in OSPFv3.

OSPFv3 Features

The following sections describe the OSPFv3 behaviors that the user can configure:

Multiple OSPFv3 Instances

As with other dynamic routing protocols, OSPFv3 is also enabled per virtual router (VR). There can be only one OSPFv3 instance in a VR. However, multiple OSPFv3 instances can be configured for multiple VRs within an AS.

OSPFv3 Route Preference

Route preference is used to select a route in the forwarding table when more than one protocol calculates the route to the same destination. The route with the lowest preference value will be selected.

Table 90: OSPFv3 Route Preference

Source	Route Preference
Connected	0
Static	20
Auto-exported	30
BGP	40
OSPFv2, OSPFv3	60
RIP, RIPng	100
EBGP	120
Imported	140
OSPFv2-e2, OSPFv3-e2	200

OSPFv3 Router ID

Each VR in an AS uses a router ID to communicate with other routing devices. In OSPFv3, neighboring routers on a given link are always identified by their OSPF router ID. You should always define a virtual-router ID before enabling OSPFv3. If you do not specify a router ID, ScreenOS selects the highest IPv4 address of an active interface in the VRs as the router ID. If a VR ID is not defined and no IPv4 address is available, the dynamic routing protocols cannot be enabled.

OSPFv3 Area Parameters

The OSPFv3 area parameters are similar to those of the OSPFv2 parameters. See section “Creating and Deleting an OSPF Area” on page 1275 for more details.

OSPFv3 Interface Paramters

The interface parameters that can be configured in OSPFv3–enabled interfaces are listed below:

Table 91: OSPFv3 Interface Paramters

OSPFv3	Description	Default Value
Ignore MTU	Specifies that any mismatches in maximum transmission unit (MTU) values between the local and remote interfaces that are found during OSPF database negotiations are ignored. This option should only be used when the MTU on the local interface is lower than the MTU on the remote interface.	Enabled
Passive Interface	Specifies that the IP address of the interface is advertised in the OSPF domain as an OSPF route and not as an external route, but the interface does not transmit or receive OSPF packets. This option is useful when BGP is also enabled on the interface.	Disabled
Link type	Specifies a tunnel interface as a point-to-point link or as a point-to-multipoint link.	Ethernet interfaces are treated as broadcast interfaces. Tunnel interfaces bound to OSPF areas are point-to-point by default.
Cost	Specifies the metric for the interface. The cost associated with an interface depends upon the bandwidth of the link to which the interface is connected. The higher the bandwidth, the lower (more desirable) the cost value.	1 for a 100 MB or greater link 10 for a 10 MB link 100 for a 1 MB link
Instance ID	Controls the selection of neighbors. Only routers that have the same interface ID can become neighbors.	0
Priority	Specifies the priority for the virtual router to be elected the designated router or backup designated router. The router with the larger priority value has the best (although not guaranteed) chance of being elected.	1
Dead interval	Specifies the number of seconds that elapses with no response from an OSPF neighbor before OSPF determines the neighbor is not running.	40 seconds

Table 91: OSPFv3 Interface Paramters *(continued)*

OSPFv3	Description	Default Value
Hello Interval	Specifies the interval, in seconds, at which OSPF sends out hello packets to the network.	10 seconds
Retransmit interval	Specifies the number of seconds that elapses before the interface resends an LSA to a neighbor that did not respond to the original LSA.	10 seconds
Transmit delay	Specifies the number of seconds between transmissions of link-state update packets sent on the interface.	1 second

Compared with OSPFv2 interface parameters, only instance ID is the new parameter.

Route Redistribution

IPv6 routes within the same VR as that of OSPFv3 can be redistributed into OSPFv3 routing instance. Redistributed routes can be summarized (based on the summary list) or filtered (based on the route map). The summary-import and route-redistribution settings can be configured for OSPFv3 as in OSPFv2.

Configuring OSPFv3

The following section explains the steps involved in configuring the OSPFv3 parameters using the CLI and WebUI.

To enable OSPFv3

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Create OSPFv3 Instance:

Select OSPFv3 Enabled, then click OK.

CLI

Initiating the OSPFv3 context can take up to four steps:

1. Enter the vrouter context by executing the set vroutercommand.

   set vrouter trust-vr

2. Enter the ospfv3context by executing the set protocol ospfv3command.

   device(trust-vr) -> set protocol ospfv3

3. Enable OSPFv3 (it is disabled by default).

```
device(trust-vr/ospfv3) -> set enable
```

To create an OSPFv3 area with area-id 10

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Create OSPFv3 Instance > Area.

Area ID: 10
Type: normal (select)
Action: Add

CLI

```
set vrouter trust-vr protocol ospfv3 area 10
save
```

To Assign Interfaces to OSPFv3 Areas

WebUI

Once an area is created, you can assign one or more interfaces to the area, using either the WebUI or the CLI set interface command. In the following example, you assign the ethernet2 interface to OSPF area 10 and assign the ethernet3 interface to OSPF area 0.

Network > Routing > Virtual Routers > Edit (trust-vr) > Edit OSPFv3 Instance > Area > Configure (Area 10): Use the Add button to move the ethernet2 interface from the Available Interface(s) column to the Selected Interfaces column. Click OK.

Network > Routing > Virtual Routers > Edit (trust-vr) > Edit OSPFv3 Instance > Area > Configure (Area 10): Use the Add button to move the ethernet3 interface from the Available Interface(s) column to the Selected Interfaces column. Click OK.

CLI

Enabling OSPFv3 on interfaces:

```
set interface ethernet1 protocol ospfv3 enable
set interface ethernet3 protocol ospfv3 enable
```

Assigning interfaces to OSPFv3 areas:

```
set interface ethernet1 protocol ospfv3 area 10
set interface ethernet3 protocol ospfv3 area 0
save
```

To Configure Area Range

In the following example, you define the following area ranges for area 10:

- 2001::1/64 to be advertised
- 2001:20::/64 no-advertise

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPFv3 Instance > Area > Configure (0.0.0.10): Enter the following in the Area Range section, then click Add:

IP / Netmask: 2001::1/64Type: (select) Advertise
IP / Netmask: 2001:20::/64Type: (select) No Advertise

CLI

```
set vrouter trust-vr protocol ospfv3 area 10 range 2001::1/64 advertise
save
```

To redistribute routes from BGP to OSPFv3

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPFv3 Instance > Redistributable Routes

Route Map: map1(select)
Protocol: BGP (select)
Action: Add

CLI

```
set vrouter trust-vr protocol ospfv3 redistribute route-map map1 protocol bgp
save
```

To configure OSPFv3 interface parameters

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit OSPFv3 Instance > Interfaces Enter the following, then click Apply:

Instance ID: 10
Hello Interval: 15
Retransmit Interval: 7
Transit Delay: 2

CLI

```
set interface ethernet1 protocol ospfv3 hello-interval 15 save
set interface ethernet1 protocol ospfv3 retransmit-interval 7
set interface ethernet1 protocol ospfv3 transit-delay 2
save
```

Monitoring OSPFv3

You can monitor the OSPFv3 configuration by using the following get commands:

Table 92: OSPFv3 get commands

Command	Description
get vrouter trust-vr protocol ospfv3	Displays general information of OSPFv3
get vrouter trust-vr protocol ospfv3 config	Displays the recently executed commands for ospfv3
get vrouter trust-vr protocol ospfv3 database	Displays the details of link state database or specified LSA
get vrouter trust-vr protocol ospfv3 interface	Displays information about OSPFv3 interfaces.
get vrouter trust-vr protocol ospfv3 neighbor	Displays information about ospfv3 neighbor.
get vrouter trust-vr protocol ospfv3 routes-redistribute	Displays information about the redistribution routes for OSPFv3.
get vrouter trust-vr protocol ospfv3 rules-redistribute	Displays information about the redistribution rules for OSPFv3.
get vr trust-vr protocol ospfv3 statistics	Displays OSPfv3 statistics.
get vr trust-vr protocol ospfv3 summary-import	Displays the summary routes that redistribute into OSPFv3
get vr trust-vr protocol ospfv3 vlink	Displays information about OSPFv3 virtual links.
get route protocol ospfv3	Displays route information in the routing table.

To get general information of OSPFv3:

```
device->get vrouter trust-vr protocol ospfv3
VR: trust-vr RouterId: 10.1.1.250
---------------------------------
OSPFv3 enabled
Internal Router
Numbers of areas is 2
Number of external LSA(s) is 0
Area 0.0.0.0
Total number of interfaces is 1, Active number of interfaces is 1
```

```
SPF algorithm executed 2 times
Number of LSA(s) is 1
Area 0.0.0.10
Total number of interfaces is 1, Active number of interfaces is 1
SPF algorithm executed 2 times
Number of LSA(s) is 0
```

To view the recently executed commands for OSPFv3:

```
device->  get vrouter trust-vr protocol ospfv3 config
VR: trust-vr RouterId: 10.1.1.250
---------------------------------
set protocol ospfv3
set enable
set area 0.0.0.10 range 2001:10::/64 advertise
set area 0.0.0.10 range 2001:20::/64 no-advertise
set interface ethernet1 protocol ospfv3 area 0.0.0.10
set interface ethernet1 protocol ospfv3 enable
set interface ethernet3 protocol ospfv3 area 0.0.0.0
set interface ethernet3 protocol ospfv3 enable
```

Chapter 35

Routing Information Protocol

This chapter describes the Routing Information Protocol (RIP) version 2 on Juniper Networks security devices. It contains the following sections:

Overview

Routing Information Protocol (RIP) is a distance vector protocol used as an Interior Gateway Protocol (IGP) in moderate-sized autonomous systems (AS). ScreenOS supports RIP version 2 (RIPv2), as defined by RFC 2453. While RIPv2 supports only simple password (plain text) authentication, the RIP implementation for ScreenOS also supports MD5 authentication extensions, as defined in RFC 2082.

NOTE: RIP is *not* supported over unnumbered tunnel interfaces. All interfaces that use RIP protocol must be numbered. Any attempt to configure and run an unnumbered interface using RIP may lead to unpredictable routing failure.

RIP manages route information within a small, homogeneous, network such as a corporate LAN. The longest path allowed in a RIP network is 15 hops. A metric value of 16 indicates an invalid or unreachable destination (this value is also referred to as “infinity” because it exceeds the 15-hop maximum allowed in RIP networks).

RIP is not intended for large networks or networks where routes are chosen based on real-time parameters such as measured delay, reliability, or load. RIP supports both point-to-point networks (used with VPNs) and broadcast/multicast Ethernet networks. RIP supports point-to-multipoint connections over tunnel interfaces with

or without a configured demand circuit. For more information about demand circuits, see “Demand Circuits on Tunnel Interfaces” on page 1328.

RIP sends out messages that contain the complete routing table to every neighboring router every 30 seconds. These messages are normally sent as multicasts to address 224.0.0.9 from the RIP port.

The RIP routing database contains one entry for every destination that is reachable through the RIP routing instance. The RIP routing database includes the following information:

- IPv4 address of a destination. Note that RIP does not distinguish between networks and hosts.
- IP address of the first router along the route to the destination (the next hop).
- Network interface used to reach the first router.
- Metric that indicates the distance, or cost, of getting to the destination. Most RIP implementations use a metric of 1 for each network.
- A timer that indicates the time that has elapsed since the database entry was last updated.

Basic RIP Configuration

You create RIP on a per-virtual router basis on a security device. If you have multiple virtual routers (VRs) within a system, you can enable multiple instances of RIP, one instance of either version 1 or 2 for each VR. By default, Juniper Networks security devices support RIP version 2.

NOTE: Before you configure a dynamic routing protocol on the security device, you should assign a VR ID, as described in “Routing” on page 1235.

This section describes the following basic steps to configure RIP on a security device:

1. Create the RIP routing instance in a VR.
2. Enable the RIP instance.
3. Enable RIP on interfaces that connect to other RIP routers.
4. Redistribute routes learned from different routing protocols (such as OSPF, BGP, or statically configured routes) into the RIP instance.

This section describes how to perform each of these tasks using either the CLI or the WebUI.

Optionally, you can configure RIP parameters such as the following:

- Global parameters, such as timers and trusted RIP neighbors, that are set at the VR level for RIP (see “Global RIP Parameters” on page 1316)

- Interface parameters, such as neighbor authentication, that are set on a per-interface basis for RIP (see "Configuring RIP Interface Parameters" on page 1318)
- Security-related RIP parameters, that are set at either the VR level or on a per-interface basis (see "Security Configuration" on page 1319)

Creating and Deleting a RIP Instance

You create and enable a RIP routing instance on a specific virtual router (VR) on a security device. When you create and enable a RIP routing instance on a VR, RIP transmits and receives packets on all RIP-enabled interfaces in the VR.

Deleting a RIP routing instance in a VR removes the corresponding RIP configurations for all interfaces that are in the VR.

For more information about VRs and configuring a VR on security devices, see "Routing" on page 1235.

Creating a RIP Instance

You create a RIP routing instance on the *trust-vr* and then enable RIP.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit: Enter a **Virtual Router ID** and then Select **Create RIP Instance**.

Select **Enable RIP**, then click **OK**.

CLI

1. **Router ID**

```
set vrouter trust-vr router-id 10
```

2. **RIP Routing Instance**

```
set vrouter trust-vr protocol rip
set vrouter trust-vr protocol rip enable
save
```

NOTE: In the CLI, creating a RIP routing instance is a two-step process. You create the RIP instance and then enable RIP.

Deleting a RIP Instance

In this example, you disable the RIP routing instance in the trust-vr. RIP stops transmitting and processing packets on all RIP-enabled interfaces of the trust-vr.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIP Instance: Deselect Enable RIP and then click **OK**.

Network > Routing > Virtual Router (trust-vr) > Edit > Delete RIP Instance and then click **OK** at the confirmation prompt.

CLI

```
unset vrouter trust-vr protocol rip enable
unset vrouter trust-vr protocol rip
save
```

Enabling and Disabling RIP on Interfaces

By default, RIP is disabled on all interfaces in the virtual router (VR) and you must explicitly enable it on an interface. When you disable RIP at the interface level, RIP does not transmit or receive packets on the specified interface. Interface configuration parameters are preserved when you disable RIP on an interface.

NOTE: If you disable the RIP routing instance in the VR (see "Deleting a RIP Instance" on page 1309), RIP stops transmitting and processing packets on all RIP-enabled interfaces in the VR.

Enabling RIP on an Interface

In this example, you enable RIP on the Trust interface.

WebUI

Network > Interface > Edit (for Trust) > RIP: Select Protocol RIP **Enable**, then click **Apply**.

CLI

```
set interface trust protocol rip enable
save
```

Disabling RIP on an Interface

In this example, you disable RIP on the Trust interface. To completely remove the RIP configuration enter the second CLI command before saving.

WebUI

Network > Interface (for Trust) > RIP: Clear Protocol RIP **Enable**, then click **Apply**.

CLI

```
unset interface trust protocol rip enable
unset interface trust protocol rip
save
```

Redistributing Routes

Route redistribution is the exchange of route information between routing protocols. For example, you can redistribute the following types of routes into the RIP routing instance in the same virtual router (VR):

- Routes learned from BGP
- Routes learned from OSPF
- Directly connected routes
- Imported routes
- Statically configured routes

You need to configure a route map to filter the routes that are redistributed. For more information about creating route maps for route redistribution, see “Routing” on page 1235.

Routes imported into RIP from other protocols have a default metric of 10. You can change the default metric (see “Global RIP Parameters” on page 1316).

In this example, you redistribute static routes that are in the subnetwork 20.1.0.0/16 to RIP neighbors in the trust-vr. To do this, you first create an access list to permit addresses in the 20.1.0.0/16 subnetwork. Then, configure a route map that permits addresses that match the access list you configured. Use the route map to specify the redistribution of static routes into the RIP routing instance.

WebUI

Network > Routing > Virtual Router (trust-vr) > Access List > New: Enter the following, then click **OK**:

```
Access List ID: 20
Sequence No.: 1
IP/Netmask: 20.1.0.0/16
Action: Permit (select)
```

Network > Routing > Virtual Router (trust-vr) > Route Map > New: Enter the following, then click **OK**:

```
Map Name: rtmap1
   Sequence No.: 1
   Action: Permit (select)
   Match Properties:
     Access List: (select), 20 (select)
```

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIP Instance > Redistributable Rules: Enter the following, then click **Add**:

Route Map: rtmap1 (select)
Protocol: Static (select)

CLI

```
set vrouter trust-vr access-list 20 permit ip 20.1.0.0/16 1
set vrouter trust-vr route-map name rtmap1 permit 1
set vrouter trust-vr route-map rtmap1 1 match ip 20
set vrouter trust-vr protocol rip redistribute route-map rtmap1 protocol static
save
```

Viewing RIP Information

After modifying RIP parameters, you can view the following types of RIP details:

- Database, which shows routing information
- Protocol, which gives RIP and interface details for a virtual router (VR)
- Neighbor

Viewing the RIP Database

You can verify RIP routing information from the CLI. You can choose to view a complete list of all RIP database entries or a single entry.

In this example, you view detailed information from the RIP database. You can choose to view all database entries or limit the output to a single database entry by appending the IP address and mask of the desired VR.

In this example, you specify the trust-vr and append the prefix and IP address 10.10.10.0/24 to view only a single table entry.

WebUI

NOTE: You must use the CLI to view the RIP database.

CLI

```
get vrouter trust-vr protocol rip database prefix 10.10.10.0/24
save
```

After you enter the following CLI command, you can view the RIP database entry:

```
device->  get vrouter trust-vr protocol rip database 10.10.10.0/24
VR: trust-vr
------------------------------------------------------------------------------
```

```
Total database entry: 3
Flags: Added in Multipath - M, RIP - R, Redistributed - I,
       Default (advertised) - D, Permanent - P, Summary - S,
       Unreachable - U, Hold - H
DBID Prefix        Nexthop    Ifp  Cost Flags Source
   7 10.10.10.0/24      20.20.20.1 eth1    2    MR 20.20.20.1
--------------------------------------------------------------------------------
```

The RIP database contains the following fields:

- DBID, the database identifier for the entry
- Prefix, the IP address and prefix
- Nexthop, the address of the next hop (router)
- Ifp, the type of connection (Ethernet or tunnel)
- Cost metric assigned to indicate the distance form the source

Flags can be one or more of the following: multipath (M), RIP (R), Redistributed (I), Advertised default (D), Permanent (P), Summary (S), Unreachable (U), or Hold (H).

In this example, the database identifier is 7, the IP address and prefix is 10.10.10.0/24, and the next hop is 20.20.20.1. It is an Ethernet connection with a cost of 2. The flags are M and R and indicate that this route is multipath and uses RIP.

Viewing RIP Details

You can view RIP details to verify that the RIP configuration matches your network needs. You can limit output to only the interface summary table by appending interface to the CLI command.

You can view complete RIP information to check a configuration or verify that saved changes are active.

WebUI

NOTE: You must use the CLI to view the RIP details.

CLI

```
get vrouter trust-vr protocol rip
```

This command produces output similar to the following output:

```
device->  get vrouter trust-vr protocol rip
VR: trust-vr
--------------------------------------------------------------------------------

State: enabled
Version: 2
Default metric for routes redistributed into RIP: 10
```

```
Maximum neighbors per interface: 16
Not validating neighbor in same subnet: disabled
RIP update transmission not scheduled
Maximum number of Alternate routes per prefix: 2
Advertising default route: disabled
Default routes learnt by RIP will not be accepted
Incoming routes filter and offset-metric: not configured
Outgoing routes filter and offset-metric: not configured
Update packet threshold is not configured
Total number of RIP interfaces created on vr(trust-vr): 1
Update| Invalid|   Flush| DC Retransmit| DC Poll| Hold Down (Timers in seconds)
---------------------------------------------------------------------------
    30|     180|     120|        5|      40| 90
Flags: Split Horizon - S, Split Horizon with Poison Reverse - P, Passive - I
       Demand Circuit - D
Interface   IP-Prefix     Admin       State    Flags   NbrCnt Metric Ver-Rx/Tx
---------------------------------------------------------------------------
tun.1       122.1.2.114/8 enabled    disabled       SD        1      1  v1v2/v1v
```

You can view RIP settings, packet details, RIP timer information, and a summarized interface table.

Viewing RIP Neighbor Information

You can view details about RIP neighbors for a virtual router (VR). You can retrieve a list of information about all neighbors or an entry for a specific neighbor by appending the IP address of the desired neighbor. You can check the status of a route and verify the connection between the neighbor and the security device from these statistics.

In the following example you view RIP neighbor information for the trust-vr.

WebUI

NOTE: You must use the CLI to view RIP neighbor information.

CLI

get vrouter trust-vr protocol rip neighbors

This command produces output similar to the following output:

```
device->  get vrouter trust-vr protocol rip neighbors
VR: trust-vr
-------------------------------------------------------------------------------
Flags : Static - S, Demand Circuit - T, NHTB - N, Down - D, Up - U, Poll - P,
        Demand Circuit Init - I
Neighbors on interface tunnel.1
-------------------------------------------------------------------------------
IpAddress       Version  Age          Expires      BadPackets BadRoutes Flags
-------------------------------------------------------------------------------
10.10.10.1      v2       -            -                     0         0 TSD
```

In addition to viewing the IP address and RIP version, you can view the following RIP neighbor information:

- Age of the entry
- Expiration time
- Number of bad packets
- Number of bad routes
- Flags: static (S), demand circuit (T), NHTB (N), down (D), up (U), poll (P), or demand circuit init (I)

Viewing RIP Details for a Specific Interface

You can view all pertinent RIP information for all interfaces and a summary of neighboring router details. Optionally, you can append the IP address of a specific neighbor to limit the output.

In the following example, you can view information about the tunnel.1 interface for the neighbor residing at IP address 10.10.10.2.

WebUI

NOTE: You must use the CLI to view the RIP interface details.

CLI

```
get interface tunnel.1 protocol rip neighbor 10.10.10.2
```

This command produces output similar to the following output:

```
device->  get interface tunnel.1 protocol rip
VR: trust-vr
------------------------------------------------------------------------------
Interface: tunnel.1, IP: 10.10.10.2/8, RIP: enabled, Router: enabled
Receive version v1v2, Send Version v1v2
State: Down, Passive: No
Metric: 1, Split Horizon: enabled, Poison Reverse: disabled
Demand Circuit: configured
Incoming routes filter and offset-metric: not configured
Outgoing routes filter and offset-metric: not configured
Authentication: none
Current neighbor count: 1
Update not scheduled
Transmit Updates: 0 (0 triggered), Receive Updates: 0
Update packets dropped because flooding: 0
Bad packets: 0, Bad routes: 0
Flags : Static - S, Demand Circuit - T, NHTB - N, Down - D, Up - U, Poll - P
Neighbors on interface tunnel.1
------------------------------------------------------------------------------
IpAddress       Version  Age          Expires      BadPackets BadRoutes Flags
```

```
------------------------------------------------------------------------------
10.10.10.1      -       -           -                   0         0 TSD
```

From this summary of information you can view the number of bad packets or bad routes present, verify any overhead that RIP adds to the connection, and view authentication settings.

Global RIP Parameters

This section describes RIP global parameters that you can configure at the virtual router (VR) level. When you configure a RIP parameter at the VR level, the parameter setting affects operations on all RIP-enabled interfaces. You can modify global parameter settings through the RIP routing protocol context in the CLI or by using the WebUI.

Table 93 on page 1316 lists the RIP global parameters and their default values.

Table 93: Global RIP Parameters and Default Values

RIP Global Parameter	Description	Default Value(s)
Default metric	Default metric value for routes imported into RIP from other protocols, such as OSPF and BGP.	10
Update timer	Specifies, in seconds, when to issue updates of RIP routes to neighbors.	30 seconds
Maximum packets per update	Specifies the maximum number of packets received per update.	No maximum
Invalid timer	Specifies, in seconds, when a route becomes invalid from the time a neighbor stops advertising the route.	180 seconds
Flush timer	Specifies, in seconds, when a route is removed from the time the route is invalidated.	120 seconds
Maximum neighbors	The maximum number of RIP neighbors allowed.	Depends on platform
Trusted neighbors	Specifies an access list that defines RIP neighbors. If no neighbors are specified, RIP uses multicasting or broadcasting to detect neighbors on an interface. See "Configuring Trusted Neighbors" on page 1320.	All neighbors are trusted
Allow neighbors on different subnet	Specifies that RIP neighbors on different subnets are allowed.	Disabled
Advertise default route	Specifies whether the default route (0.0.0.0/0) is advertised.	Disabled
Reject default routes	Specifies whether RIP rejects a default route learned from another protocol. See "Rejecting Default Routes" on page 1321.	Disabled
Incoming route map	Specifies the filter for routes to be learned by RIP.	None
Outgoing route map	Specifies the filter for routes to be advertised by RIP.	None

Table 93: Global RIP Parameters and Default Values *(continued)*

Maximum alternate routes	Specifies the maximum number of RIP routes for the same prefix that can be added into the RIP route database. See "Automatic Update" on page 569.	0
Summarize advertised routes	Specifies advertising of a summary route that corresponds to all routes that fall within a summary range. See "Enabling and Disabling a Prefix Summary" on page 1325.	None
RIP protocol version	Specifies the version of RIP the VR uses. You can override the version on a per-interface basis. See "Setting the RIP Version" on page 1324.	Version 2
Hold-timer	Prevents route flapping to the route table. You can specify a value between the minimum (three times the value of the update timer) and the maximum (sum of the update timer and the hold timer, not to exceed the value of the flush timer) values.	90 seconds
Retransmit timer	Specifies the retransmit interval of triggered responses over a demand circuit. You can set the retransmit timer and assign a retry count that matches your network needs.	5 seconds 10 retries
Poll-timer	Checks the remote neighbor for the demand circuit to see if that neighbor is up. You can configure the poll timer in minutes and assign a retry count that matches your network needs. A retry count of zero (0) means to poll forever.	180 seconds 0 retries

Advertising the Default Route

You can change the RIP configuration to include the advertisement of the default route (a non-RIP route) and change the metric associated with the default route present in a particular VR routing table.

By default, the default route (0.0.0.0/0) is not advertised to RIP neighbors. The following command advertises the default route to RIP neighbors in the trust-vr VR with a metric of 5 (you must enter a metric value). The default route must exist in the routing table.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIP Instance: Enter the following, then click **OK**:

Advertising Default Route: (select)
Metric: 5

CLI

```
set vrouter trust-vr protocol rip advertise-def-route metric number 5
save
```

NOTE: Refer to the *ScreenOS CLI Reference Guide: IPv4 Command Descriptions* for more information about global parameters that you can configure in the RIP routing protocol context.

Configuring RIP Interface Parameters

This section describes RIP parameters that you configure at the interface level. When you configure a RIP parameter at the interface level, the parameter setting affects the RIP operation only on the specific interface. You can modify interface parameter settings with **interface** commands in the CLI or by using the WebUI.

Table 94 on page 1318 lists the RIP interface parameters and their default values.

Table 94: RIP Interface Parameters and Default Values

RIP Interface Parameter	Description	Default Value
Split-horizon	Specifies whether to enable split-horizon (do not advertise routes learned from an interface in updates sent to the same interface). If split horizon is enabled with the poison-reverse option, routes that are learned from an interface are advertised with a metric of 16 in updates sent to the same interface.	Split-horizon is enabled. Poison reverse is disabled.
RIP metric	Specifies the RIP metric for the interface.	1
Authentication	Specifies either clear text password or MD5 authentication. See "Authenticating Neighbors by Setting a Password" on page 1319.	No authentication used.
Passive mode	Specifies that the interface is to receive but not transmit RIP packets.	No
Incoming route map	Specifies the filter for routes to be learned by RIP.	None.
Outgoing route map	Specifies the filter for routes to be advertised by RIP.	None.
RIP version for sending or receiving updates	Specifies the RIP version used for sending or receiving updates on the interface. The version of the interface used for sending updates does not need to be the same as the version for receiving updates. See "Setting the RIP Version" on page 1324.	Version configured for the virtual router.
Route summarization	Specifies whether route summarization is enabled on the interface. See "Enabling and Disabling a Prefix Summary" on page 1325.	Disabled.
Demand-circuit	Specifies the demand circuit on a specified tunnel interface. Only when changes occur, the security device sends update messages. See "Demand Circuits on Tunnel Interfaces" on page 1328.	None.
Static neighbor IP	Specifies the IP address of a manually assigned RIP neighbor.	None.

You can define incoming and outgoing route maps at the virtual router (VR) level or at the interface level. A route map that you define at the interface-level takes precedence over a route map defined at the VR-level. For example, if you define an incoming route map at the VR level and a different incoming route map at the interface level, the incoming route map defined at the interface level takes precedence. For more information, see "Configuring a Route Map" on page 1262.

In the following example, you configure the following RIP parameters for the trust interface:

- Set MD5 authentication, with the key 1234567898765432 and the key ID 215.
- Enable split horizon with poison reverse for the interface.

WebUI

Network > Interfaces > Edit (for Trust) > RIP: Enter the following, then click **OK**:

Authentication: MD5 (select)
Key: 1234567898765432
Key ID: 215
Split Horizon: Enabled with poison reverse (select)

CLI

```
set interface trust protocol rip authentication md5 1234567898765432 key-id 215
set interface trust protocol rip split-horizon poison-reverse
save
```

Security Configuration

This section describes possible security problems in the RIP routing domain and methods of preventing attacks.

NOTE: To make RIP more secure, you should configure all routers in the RIP domain to be at the same security level. Otherwise, a compromised RIP router can bring down the entire RIP routing domain.

Authenticating Neighbors by Setting a Password

A RIP router can be easily spoofed, since RIP packets are not encrypted and most protocol analyzers provide decapsulation of RIP packets. Authenticating RIP neighbors is the best way to fend off these types of attacks.

RIP provides both simple password and MD5 authentication to validate RIP packets received from neighbors. All RIP packets received on the interface that are not authenticated are discarded. By default, there is no authentication enabled on any RIP interface.

MD5 authentication requires that the same key be used for both the sending and receiving RIP routers. You can specify more than one MD5 key on the security device; each key is paired with a key identifier. If you configure multiple MD5 keys on the security device, you can then select the key identifier of the key that is to be used for authenticating communications with the neighbor router. This allows MD5 keys on pairs of routers to be changed periodically with minimal risk of packets being dropped.

In the following example, you set the two different MD5 keys on interface ethernet1 and select one of the keys to be the active key. The default key-id is 0 so you do not have to specify the key-id for the first MD5 key you enter.

WebUI

Network > Interfaces > Edit (for ethernet1) > RIP: Enter the following, then click **Apply**:

MD5 Keys: (select)
1234567890123456 (first key field)
9876543210987654 (second key field)
Key ID: 1
Preferred: (select)

CLI

```
set interface ethernet1 protocol rip authentication md5 1234567890123456
set interface ethernet1 protocol rip authentication md5 9876543210987654 key-id
 1
set interface ethernet1 protocol rip authentication md5 active-md5-key-id 1
save
```

Configuring Trusted Neighbors

Multi-access environments can allow devices, including routers, to be connected into a network relatively easily. This can cause stability or performance issues if the connected device is not reliable. To prevent this problem, you can use an access list to filter the devices that are allowed to become RIP neighbors. By default, RIP neighbors are limited to devices that are on the same subnet as the virtual router (VR).

In this example, you configure the following global parameters for the RIP routing instance running in the trust-vr:

- Maximum number of RIP neighbors is 1.
- The IP address of the trusted neighbor, 10.1.1.1, is specified in an access-list.

WebUI

Network > Routing > Virtual Router (trust-vr) > Access List > New: Enter the following, then click **OK**:

Access List ID: 10
Sequence No.: 1
IP/Netmask: 10.1.1.1/32
Action: Permit (select)

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIP Instance: Enter the following, then click **OK**:

Trusted Neighbors: (select), 10
Maximum Neighbors: 1

CLI

```
set vrouter trust-vr
device(trust-vr)-> set access-list 10 permit ip 10.1.1.1/32 1
device(trust-vr)-> set protocol rip
device(trust-vr/rip)-> set max-neighbor-count 1
device(trust-vr/rip)-> set trusted-neighbors 10
device(trust-vr/rip)-> exit
device(trust-vr)-> exit
save
```

Rejecting Default Routes

In a Route Detour Attack, a router injects a default route (0.0.0.0/0) into the routing domain in order to detour packets to itself. The router can then either drop the packets, causing service disruption, or it can obtain sensitive information in the packets before forwarding them. On Juniper Networks security devices, RIP by default accepts any default routes that are learned in RIP and adds the default route to the routing table.

In the following example, you configure the RIP routing instance running in trust-vr to reject any default routes that are learned in RIP.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIP Instance: Enter the following, then click **OK**:

Reject Default Route Learnt by RIP: (select)

CLI

```
set vrouter trust-vr protocol rip reject-default-route
save
```

Protecting Against Flooding

A malfunctioning or compromised router can flood its neighbors with RIP routing update packets. On virtual router (VRs), you can configure the maximum number of update packets that can be received on a RIP interface within an update interval to avoid flooding of update packets. All update packets that exceed the configured

update threshold are dropped. If you do not set an update threshold, all update packets are accepted.

You need to exercise care when configuring an update threshold when neighbors have large routing tables, as the number of routing updates can be quite high within a given duration because of flash updates. Update packets that exceed the threshold are dropped and valid routes may not be learned.

Configuring an Update Threshold

In this example, you set the maximum number of routing update packets that RIP can receive on an interface to 4.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIP Instance: Enter the following, then click **OK**:

Maximum Number Packets per Update Time: (select), 4

CLI

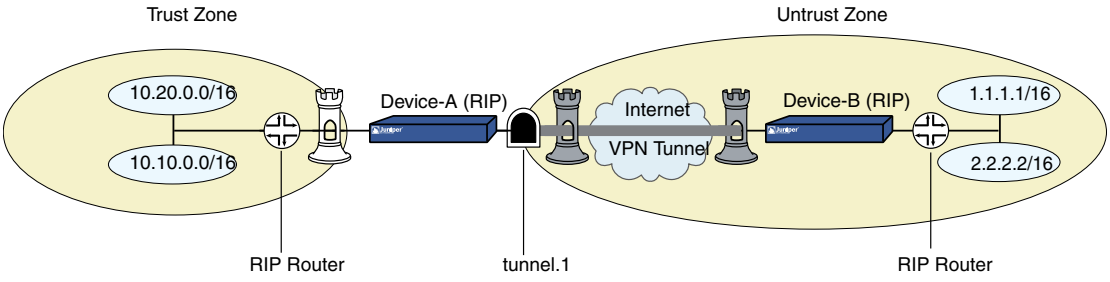
```
set vrouter trust-vr protocol rip threshold-update 4
save
```

Enabling RIP on Tunnel Interfaces

The following example creates and enables a RIP routing instance in trust-vr, on the Device-A device. You enable RIP on both the VPN tunnel interface and the Trust zone interface. Only routes that are in the subnet 10.10.0.0/16 are advertised to the RIP neighbor on Device-B. This is done by first configuring an access list that permits only addresses in the subnet 10.10.0.0/16, then specifying a route map *abcd* that permits routes that match the access list. You then specify the route map to filter the routes that are advertised to RIP neighbors.

Figure 330 on page 1322 shows the described network scenario.

Figure 330: Tunnel Interface with RIP Example



WebUI

Network > Routing > Virtual Router > Edit (for trust-vr) > Create RIP Instance: Select **Enable RIP**, then click **OK**.

Network > Routing > Virtual Router > Access List (for trust-vr) > New: Enter the following, then click **OK**:

Access List ID: 10
Sequence No.: 10
IP/Netmask: 10.10.0.0/16
Action: Permit

Network > Routing > Virtual Router > Route Map (for trust-vr) > New: Enter the following, then click **OK**:

Map Name: abcd
Sequence No.: 10
Action: Permit
Match Properties:
Access List: (select), 10

Network > Routing > Virtual Router > Edit (for trust-vr) > Edit RIP Instance: Select the following, then click **OK:**

Outgoing Route Map Filter: abcd

Network > Interfaces > Edit (for tunnel.1) > RIP: Enter the following, then click **Apply**:

Enable RIP: (select)

Network > Interfaces > Edit (for trust) > RIP: Enter the following, then click **Apply**:

Enable RIP: (select)

CLI

```
set vrouter trust-vr protocol rip
set vrouter trust-vr protocol rip enable
set interface tunnel.1 protocol rip enable
set interface trust protocol rip enable
set vrouter trust-vr access-list 10 permit ip 10.10.0.0/16 10
set vrouter trust-vr route-map name abcd permit 10
set vrouter trust-vr route-map abcd 10 match ip 10
set vrouter trust-vr protocol rip route-map abcd out
save
```

Optional RIP Configurations

This section describes various RIP features that you can configure.

Setting the RIP Version

On Juniper Networks security devices, you can configure the Routing Information Protocol (RIP) version for the virtual router (VR) and for each RIP interface that sends and receives updates. Per RFC 2453, the VR can run a version of RIP that differs from the instance of RIP running on a particular interface. You can also configure different RIP versions for sending updates and for receiving updates on a RIP interface.

On the VR, you can configure either RIP version 1 or version 2; the default is version 2. For sending updates on RIP interfaces, you can configure either RIP version 1, version 2, or version 1-compatible mode (described in RFC 2453). For receiving updates on RIP interfaces, you can configure either RIP version 1, version 2, or both version 1 and 2.

NOTE: Using both versions 1 and 2 at the same time is not recommended. Network complications can result between versions 1 and 2 of the protocol.

For both sending and receiving updates on RIP interfaces, the default RIP version is the version that is configured for the VR.

In the following example, you set RIP version 1 in trust-vr. For the interface ethernet3, you set RIP version 2 for both sending and receiving updates.

WebUI

Network > Routing > Virtual Router > Edit (for trust-vr) > Edit RIP Instance: Select V1 for Version, then click **Apply**.

Network > Interfaces > Edit (for ethernet3) > RIP: Select V2 for Sending and Receiving in Update Version, then click **Apply**.

CLI

```
set vrouter trust-vr protocol rip version 1
set interface ethernet3 protocol rip receive-version v2
set interface ethernet3 protocol rip send-version v2
save
```

To verify the RIP version in the VR and on RIP interfaces, you can enter the **get vrouter trust-vr protocol rip** command.

```
device->  get vrouter trust-vr protocol rip
VR: trust-vr
---------------------------------
State: enabled
Version: 1
Default metric for routes redistributed into RIP: 10
Maximum neighbors per interface: 512
Not validating neighbor in same subnet: disabled
Next RIP update scheduled after: 14 sec
```

```
Advertising default route: disabled
Default routes learnt by RIP will be accepted
Incoming routes filter and offset-metric: not configured
Outgoing routes filter and offset-metric: not configured
Update packet threshold is not configured
Total number of RIP interfaces created on vr(trust-vr): 1
Update Invalid Flush (Timers in seconds)
-------------------------------------------------------------
    30     180   120
Flags: Split Horizon - S, Split Horizon with Poison Reverse - P, Passive - I
       Demand Circuit - D
Interface   IP-Prefix        Admin      State    Flags   NbrCnt Metric Ver-Rx/Tx
-------------------------------------------------------------------------------
ethernet3   20.20.1.2/24     enabled    enabled  S            0      1   2/2
```

In the example above, the security device is running RIP version 1 on the trust-vr; but RIP version 2 is running on the ethernet3 interface for sending and receiving updates.

Enabling and Disabling a Prefix Summary

You can configure a summary route that encompasses a range of route prefixes to be advertised by RIP. The security device then advertises only one route that corresponds to a summary range instead of individually advertising each route that falls within the summary range. This can reduce the number of route entries sent in RIP updates and reduce the number of entries that RIP neighbors need to store in their routing tables. You enable route summarization on the RIP interface from which the device sends. You can choose to summarize routes on one interface and send routes without summarization on another interface.

NOTE: You cannot selectively enable summarization for a specific summary range; when you enable summarization on an interface, all configured summary routes appear on routing updates.

When configuring the summary route, you cannot specify multiple prefix ranges that overlap. You also cannot specify a prefix range that includes the default route. You can optionally specify a metric for the summary route. If you do not specify a metric, the largest metric for all routes that fall within the summary range is used.

Sometimes a summarized route can create opportunities for loops to occur. You can configure a route to a NULL interface to avoid loops. For more information about setting a NULL interface, see “Preventing Loops Created by Summarized Routes” on page 1232.

Enabling a Prefix Summary

In the following example, you configure a summary route 10.1.0.0/16, which encompasses the prefixes 10.1.1.0/24 and 10.1.2.0/24. To allow ethernet3 to send the summary route in RIP updates, you need to enable summarization on the interface.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit RIP Instance > Summary IP: Enter the following, then click **Add**:

Summary IP: 10.1.0.0
Netmask: 16
Metric: 1

Network > Interface > Edit (for ethernet3) > RIP: Select Summarization, then click **Apply**.

CLI

```
set vrouter trust-vr protocol rip summary-ip 10.1.0.0/16
set interface ethernet3 protocol rip summary-enable
save
```

Disabling a Prefix Summary

In the following example, you disable a prefix summary route for ethernet3 on the trust-vr.

WebUI

Network > Interfaces > Edit > RIP: Uncheck **Summarization**, then click **Apply**.

CLI

```
unset vrouter trust-vr protocol rip summary-ip 10.1.0.0/16
unset interface ethernet3 protocol rip summary-enable
save
```

Setting Alternate Routes

The security device maintains a RIP database for routes learned by the protocol and routes that are redistributed into RIP. By default, only the best route for a given prefix is maintained in the database. You can specify that one, two, or three alternate RIP routes for the same prefix can exist in the RIP database. If you allow alternate routes for a prefix in the RIP database, routes to the same prefix with a different next-hop or RIP source are added to the RIP database. This allows RIP to support demand circuits and fast failover.

NOTE: We recommend the use of alternate routes with demand circuits. For more information about demand circuits, see “Demand Circuits on Tunnel Interfaces” on page 1328.

Only the best route in the RIP database for a given prefix is added to the routing table of a virtual router (VR) and advertised in RIP updates. If the best route is removed

from the routing table of a VR, then RIP adds the next-best route for the same prefix from the RIP database. If a new route, which is better than the best existing route in the routing table of a VR, is added to the RIP database, then RIP updates to use the new better route to the routing table and stops using the old route. Depending upon the alternate route limit you configured, RIP may or may not delete the old route from the RIP database.

You can view the RIP database by issuing this CLI command: **get vrouter** *vrouter* **protocol rip database**. In the following example, the number of alternate routes for the RIP database is set to a number greater than 0. The RIP database shows two entries for the prefix 10.10.70.0/24 in the RIP database, one with a cost of 2 and the other with a cost of 4. The best route for the prefix, the route with the lowest cost, is included in the routing table of the VR.

```
device->  get vrouter trust-vr protocol rip database
VR: trust-vr
--------------------------------------------------------------------------------
Total database entry: 14
Flags: Added in Multipath - M, RIP - R, Redistributed - I
       Default (advertised) - D, Permanent - P, Summary - S
       Unreachable - U, Hold - H
DBID   Prefix               Nexthop            Interface  Cost Flags Source
--------------------------------------------------------------------------------

                                        .
                                        .
                                        .
  47   10.10.70.0/24         10.10.90.1         eth4          2 MR    10.10.90.1

  46   10.10.70.0/24         10.10.90.5         eth4          4 R     10.10.90.5
```

If equal cost multipath (ECMP) routing is enabled (see “Configuring Equal Cost Multipath Routing” on page 1259) and multiple routes of equal cost exist in the RIP database for a given prefix, then RIP adds multiple routes for the prefix into the routing table of the VR up to the ECMP limit. In some cases, the alternate route limit in the RIP database may result in RIP routes not being added to the routing table of the VR. If the ECMP limit is less than or equal to the alternate route limit in the RIP database, RIP routes that are not added to the routing table for the VR remain in the RIP database; these routes are added into the routing table for the VR only if a previously added route is either deleted or is no longer the “best” RIP route for the network prefix.

For example, if the ECMP limit is 2 and the alternate route limit in the RIP database is 3, there can be only two RIP routes for the same prefix with the same cost in the routing table for the VR. Additional same prefix/same cost routes in the RIP database can exist, but only two routes are added into the routing table for the VR.

In the following example, you set the number of alternate routes allowed for a prefix in the RIP database to 1 in trust-vr. This allows one “best” route and one alternate route for any given prefix in the RIP database in the VR.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit RIP Instance: Enter 1 in the Maximum Alternative Route field, then click **Apply**.

CLI

```
set vrouter trust-vr protocol rip alt-route 1
save
```

Demand Circuits on Tunnel Interfaces

A demand circuit is a point-to-point connection between two tunnel interfaces. Minimal network overhead in terms of messages pass between the demand circuit end points. Demand circuits for RIP, defined by RFC 2091 for wide area networks, support large numbers of RIP neighbors on VPN tunnels on Juniper Networks security devices.

Demand circuits for RIP eliminate the periodic transmission of RIP packets over the tunnel interface. To save overhead, the security device sends RIP information only when changes occur in the routing database. The security device also retransmits updates and requests until valid acknowledgements are received. The security device learns RIP neighbors through the configuration of static neighbors; and if the VPN tunnel goes down, RIP flushes routes learned from the neighbor's IP address.

Routes learned from demand circuits do not age with RIP timers because demand circuits are in a permanent state. Routes in permanent state are only removed under the following conditions:

- A formerly reachable route changes to unreachable in an incoming response
- The VPN tunnel goes down or the demand circuit is down due to an excessive number of unacknowledged retransmissions

On the security device, you can also configure a point-to-point or a point-to-multipoint tunnel interface as a demand circuit. You must disable route-deny (if configured) on a point-to-multipoint tunnel so that all routes can reach remote sites. Although not required, you can also disable split horizon on the point-to-multipoint interface with demand circuits. If you disable split horizon, the end points can learn about each other.

You must configure VPN monitoring with rekey on VPN tunnels in order to learn tunnel status.

After you configure the demand circuit and the static neighbor(s), you can set the RIP retransmit-timer, poll-timer, and hold-down-timer to conform to your network requirements.

Examples of how to configure a demand circuit and a static neighbor follow this section. A RIP network configuration example with demand circuits over point-to-multipoint tunnel interfaces begins on "Creating a Point-to-Multipoint Network" on page 1294.

In the following example, you configure tunnel.1 interface to be a demand circuit and save the configuration.

WebUI

Network > Interfaces > (Edit) RIP: Select **Demand Circuit**, then click **Apply.**

CLI

```
set interface tunnel.1 protocol rip demand-circuit
save
```

After enabling a demand circuit, you can check its status and timers with the **get vrouter*vrouter* protocol rip database** command. Table 95 on page 1329 lists suggestions for troubleshooting performance issues influenced by timer settings.

Table 95: Troubleshooting the Demand Circuit Retransmit Timer

Demand Circuit Performance	Suggestion
Relatively slow	You can reconfigure the retransmit timer to a higher value to reduce the number of retransmits.
Loss free	You can reconfigure the retransmit timer to lower retry count.
Congested and lossy	You can reconfigure the retransmit timer to a higher retry count to give the static neighbor more time to respond before forcing the static neighbor into a POLL state.

Configuring a Static Neighbor

A point-to-multipoint interface that is running RIP requires statically configured neighbors. For demand circuits manual configuration is the only way for a security device to learn neighbor addresses on point-to-multipoint interfaces. To configure a RIP static neighbor you enter the interface name and the IP address of the RIP neighbor.

In the following example you configure the RIP neighbor at IP address 10.10.10.2 of the tunnel.1 interface.

WebUI

Network > Interfaces > (Edit) RIP: Click **Static Neighbor IP** button to advance to the Static Neighbor IP table. Enter the IP address of the static neighbor, then click **Add**.

CLI

```
set interface tunnel.1 protocol rip neighbor 10.10.10.2
unset interface tunnel.1 protocol rip neighbor 10.10.10.2
save
```

Configuring a Point-to-Multipoint Tunnel Interface

RIP point-to-multipoint is supported on numbered tunnel interfaces for RIP versions 1 and 2.

CAUTION: RIP is *not* supported over unnumbered tunnel interfaces. All interfaces that use RIP protocol must be numbered. Any attempt to configure and run an unnumbered interface using RIP may lead to unpredictable routing failure.

You must disable split horizon on a point-to-multipoint interface tunnel that you configure with demand circuits so that messages reach all remote sites. For a point-to-multipoint tunnel interface without demand circuits, you can leave split horizon enabled (default). RIP dynamically learns about neighbors. RIP sends all transmitted messages to the multicast address 224.0.0.9 and reduplicates them to all tunnels as appropriate.

If you want to set up RIP as a point-to-multipoint tunnel with demand circuits, you must design your network in a hub-and-spoke configuration.

NOTE: In this example, we only reference the command line interfaces, and we do not discuss zones and other necessary configuration steps.

The network in this example is a medium-sized enterprise that has a central office (CO) in San Francisco and remote sites in Chicago, Los Angeles, Montreal, and New York. Each office has a single security device. See Figure 331 on page 1331.

The following are the configuration requirements particular to the security device in the CO:

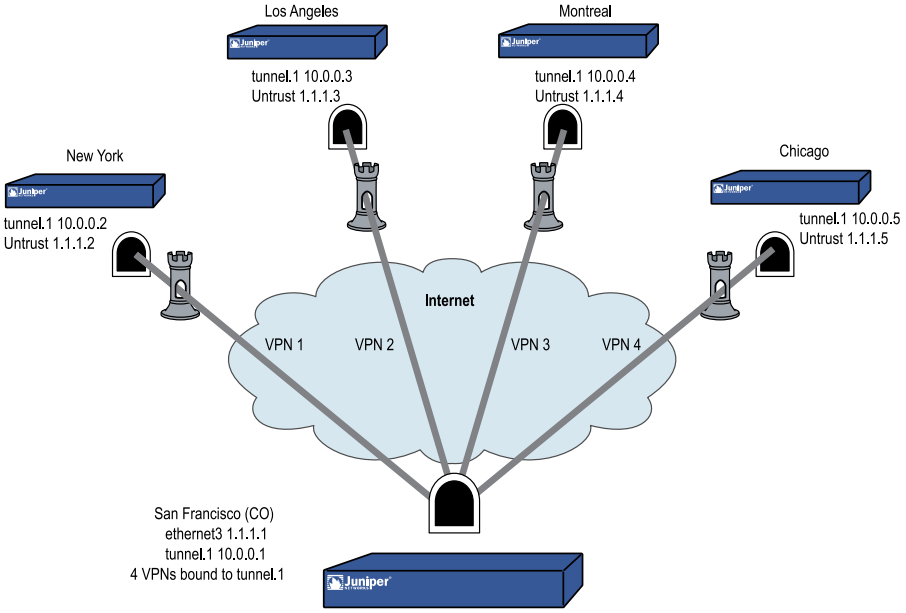
1. Configure the VR to run an instance of RIP, enable RIP, and then configure tunnel.1 interface.
2. Configure the four VPNs and bind them to tunnel.1 interface.
3. Configure RIP static neighbors on the CO security device.
4. Do not change the default timer values in this example.

The following are the configuration requirements particular to the remote Juniper Networks security devices:

1. Configure the VR to run an instance of RIP and enable RIP and then configure tunnel.1 interface.
2. Configure the VPN and bind it to tunnel.1 interface.
3. Do not configure static neighbors on the remote office security devices. The remote office devices only have one neighboring device that will be discovered by initial multicast requests.

NOTE: It is not necessary to change the default timer values in this example.

Figure 331: Point-to-MultiPoint with Tunnel Interface Network Example



In the network diagram shown in Figure 331 on page 1331, four VPNs originate from the San Francisco security device and radiate out to remote offices in New York, Los Angeles, Montreal, and Chicago.

In this example, you configure the following settings on the CO security device:

1. Security Zone and Interfaces
2. VPN
3. Routes and **RIP**
4. Static Neighbors
5. Summary Route

To be able to check the circuit status on the device in the CO, you must enable VPN monitoring.

To complete the network configuration, you configure the following settings on each of the four remote office security devices:

1. Security Zone and Interfaces
2. VPN
3. Routes and RIP

4. Static Neighbors
5. Summary Route

NOTE: The WebUI procedures are abbreviated due to the length of the example. The CLI portion of the example is complete. You can refer ahead to the CLI portion for the exact settings and values to use.

WebUI (Central Office Device)

1. **Security Zones and Interfaces**

   Network > Interfaces > Click **New Tunnel IF** and continue to the Configuration page.

   Network > Interfaces > Edit (for ethernet3)

2. **VPN**

   VPNs > AutoKey Advanced > Gateway

3. **Routes and RIP**

   Network > Routing > Virtual Routers > Click **Edit** for the virtual router and click **Create RIP Instance**, then enable RIP on the virtual router.

   Network > Interfaces > Edit > Click Edit and then click RIP, then enable RIP on the interface.

4. **Static Neighbors**

   Network > Interfaces > Edit > RIP > Static Neighbor IP and then **Add Neighbor IP Address.**

5. **Summary Route**

   Network > Routing > Virtual Router Edit (RIP) > Click **Summary IP** and configure the summary IP address.

CLI (Central Office Device)

1. **Security Zones and Interfaces**

```
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.0.0.1/24
```

2. **VPN**

```
set ike gateway gw1 address 1.1.1.2 main outgoing-interface ethernet3 preshare
 ripdc proposal pre-g2-3des-sha
```

```
set ike gateway gw2 address 1.1.1.3 main outgoing-interface ethernet3 preshare
 ripdc proposal pre-g2-3des-sha
set ike gateway gw3 address 1.1.1.4 main outgoing-interface ethernet3 preshare
 ripdc proposal pre-g2-3des-sha
set ike gateway gw4 address 1.1.1.5 main outgoing-interface ethernet3 preshare
 ripdc proposal pre-g2-3des-sha

set vpn vpn1 gateway gw1 no-replay tunnel idletime 0 proposal g2-esp-3des-sha
set vpn vpn1 monitor rekey
set vpn vpn1 bind interface tunnel.1

set vpn vpn2 gateway gw2 no-replay tunnel idletime 0 proposal g2-esp-3des-sha
set vpn vpn2 monitor rekey
set vpn vpn2 bind interface tunnel.1

set vpn vpn3 gateway gw3 no-replay tunnel idletime 0 proposal g2-esp-3des-sha
set vpn vpn3 monitor rekey
set vpn vpn3 bind interface tunnel.1

set vpn vpn4 gateway gw4 no-replay tunnel idletime 0 proposal g2-esp-3des-sha
set vpn vpn4 monitor rekey
set vpn vpn4 bind interface tunnel.1
```

3. **Routes and RIP**

```
set vrouter trust protocol rip
set vrouter trust protocol rip enable
set vrouter protocol rip summary-ip 100.10.0.0/16

set interface tunnel.1 protocol rip
set interface tunnel.1 protocol rip enable
set interface tunnel.1 protocol rip demand-circuit
```

4. **Static Neighbors**

```
set interface tunnel.1 protocol rip neighbor 10.0.0.2
set interface tunnel.1 protocol rip neighbor 10.0.0.3
set interface tunnel.1 protocol rip neighbor 10.0.0.4
set interface tunnel.1 protocol rip neighbor 10.0.0.5
```

5. **Summary Route**

```
set interface tunnel.1 protocol rip summary-enable
save
```

You can follow these steps to configure the remote office security device. When setting up the remote office, you do not need to configure static neighbors. In a demand circuit environment only one neighbor exists for the remote device, and the remote devices learns this neighbor's information when it sends a multicast message at startup.

To complete the configuration shown in the diagram on Figure 331 on page 1331, you must repeat this section for each remote device and change the IP addresses, gateway names and VPN names to match the network needs. For each remote site, the trust and untrust zones change.

NOTE: The WebUI procedures are abbreviated due to the length of the example. The CLI portion of the example is complete. You can refer ahead to the CLI portion for the exact settings and values to use.

WebUI (Remote Office Device)

1. **Security Zones and Interfaces**

   Network > Interfaces > **Click** New Tunnel IF and continue to the Configuration page.

   Network > Interfaces > Edit (for ethernet3)

2. **VPN**

   VPNs > AutoKey Advanced > Gateway

3. **Routes and RIP**

   Network > Routing > Virtual Routers > Click **Edit** for the virtual router and click **Create RIP Instance**, then enable RIP on the virtual router.

   Network > Interfaces > Edit > Click Edit and then click RIP, then enable RIP on the interface.

4. **Policy (configure as required)**

   Policies (from All zones to All zones) > Click **New**.

CLI (Remote Office Device)

1. **Interface and routing protocol**

```
set vrouter trust-vr protocol rip
set vrouter trust-vr protocol rip enable
set interface untrust ip 1.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.0.0.2/24
```

2. **VPN**

```
set ike gateway gw1 address 1.1.1.1/24 main outgoing-interface untrust preshare
 ripdc proposal pre-g2-3des-sha
set vpn vpn1 gateway gw1 no-replay tunnel idletime 0 proposal g2-esp-3des-sha
set vpn vpn1 monitor rekey
set vpn vpn1 id 1 bind interface tunnel.1
```

3. **Routes and RIP**

```
set interface tunnel.1 protocol rip
set interface tunnel.1 protocol rip demand-circuit
set interface tunnel.1 protocol rip enable
```

4. **Policy (configure as required)**

```
set policy id 1 from trust to untrust any any any permit
set policy id 2 from untrust to trust any any any permit
save
```

You can view the new changes with the **get vrouter** *vrouter* **protocol rip neighbors** command. Neighbors for a demand circuit appear in the neighbor table; neighbor information does not age or expire. You can view the RIP database with the **get vrouter** *vrouter* **protocol rip database** command. *P* for *permanent* appears next to demand circuit entries.

Chapter 36

Border Gateway Protocol

This chapter describes the Border Gateway Protocol (BGP) on Juniper Networks security devices. It contains the following sections:

Overview

The Border Gateway Protocol (BGP) is a path vector protocol that is used to carry routing information between Autonomous Systems (ASs). An AS is a set of routers that are in the same administrative domain.

The BGP routing information includes the sequence of AS numbers that a network prefix (a route) has traversed. The path information that is associated with the prefix is used to enable loop prevention and enforce routing policies. ScreenOS supports BGP version 4 (BGP-4), as defined in RFC 1771.

Two *BGP peers* establish a *BGP session* in order to exchange routing information. A BGP router can participate in BGP sessions with different peers. BGP peers must first establish a TCP connection between themselves to open a BGP session. Upon forming the initial connection, peers exchange entire routing tables. As routing table changes occur, BGP routers exchange update messages with peers. A BGP router maintains current versions of the routing tables of all the peers with which it has sessions, periodically sending keepalive messages to peers to verify the connections.

A BGP peer only advertises those routes that it is actively using. When a BGP peer advertises a route to its neighbor, it also includes path attributes that describe the characteristics of the route. A BGP router compares the path attributes and prefix to select the best route from all paths that are available to a given destination.

Multiprotocol BGP for IPv6

Beginning with version 6.2.0, ScreenOS supports multiprotocol BGP for Internet Protocol version 6 (IPv6). Multiprotocol BGP is an enhanced BGP that carries routing information for multiple Network Layer protocol address families, such as IPv6 and multicast.

NOTE: ScreenOS currently supports the IPv6 unicast address family only.

ScreenOS supports multiprotocol extensions for BGP-4, as defined in RFC 2858, and the use of BGP-4 multiprotocol extensions for IPv6 interdomain routing, as defined in RFC 2545.

Multiprotocol BGP extensions for IPv6 support the same features and functionality as IPv4 BGP and add support for the following features:

- IPv6 address families
- Network layer reachability information (NLRI)
- Next-hop path attributes that use IPv6 addresses

To use multiprotocol BGP for IPv6, you must enable the IPv6 environment variable. At the device level, start a CLI session with the device and enter the following commands:

```
set envar ipv6=yes
save
reset save-config yes
```

For more information, see “Enabling an IPv6 Environment” on page 2104 .

Keep the following points in mind when configuring BGP in IPv6:

- IPv6 addresses differ from IPv4 addresses.

  For information about the differences in notation, prefixes, and address types, see “IPv6 Addressing” on page 2090.

- The **ipv6** keyword needs to be added.

  If you issue commands without the **ipv4** or **ipv6** keywords, the IPv4 address family is configured by default.

Several examples showing address and keyword usage follow.

To enable the next-hop self feature for neighbor 6.6.6.6 on IPv6 routes advertised to the peer:

```
set protocol bgp ipv6 neighbor 6.6.6.6 nhself-enable
```

To enable the next-hop self feature for neighbor 2008::5 on IPv6 routes advertised to the peer:

```
set protocol bgp ipv6 neighbor 2008::5 nhself-enable
```

To enable the next-hop self feature for neighbor 6.6.6.6 on IPv4 routes advertised to the peer:

set protocol bgp ipv4 neighbor 6.6.6.6 nhself-enable
or
set protocol bgp neighbor 6.6.6.6 nhself-enable

The descriptions of functionality in this chapter apply to both IPv4 and IPv6. Many of the figures and examples reflect IPv4—they do not include the ipv4 keyword and they use IPv4 addressing. They apply to IPv6 if you add the ipv6 keyword and substitute the IPv4 addressing with that of IPv6. For examples of IPv6 commands, see the *ScreenOS CLI Reference Guide: IPv6 Command Descriptions*.

Support for IPv6 introduces the capability of using IPv4 peers to advertise IPv6 routes and, conversely, IPv6 peers to advertise IPv4 routes. For more information, see "Advertising IPv6 Routes Between IPv4 BGP Peers and IPv4 Routes Between IPv6 BGP Peers" on page 1351.

Types of BGP Messages

BGP uses four different types of messages to communicate with peers:

- **Open** messages identify BGP peers to each other to initiate the BGP session. These messages are sent after the peers establish a TCP session. During the exchange of open messages, BGP peers specify their protocol version, AS number, hold time and BGP identifier.
- **Update** messages announce routes to the peer and withdraw previously advertised routes.
- **Notification** messages indicate errors. The BGP session is terminated and then the TCP session is closed.

NOTE: The security device does not send a Notification message to a peer if, during the exchange of open messages, the peer indicates that it supports protocol capabilities that the security device does not support.

- **Keepalive** messages are used to maintain the BGP session. By default, the security device sends keepalive messages to peers at 60-second intervals. This interval is configurable.

Path Attributes

BGP path attributes are a set of parameters that describe the characteristics of a route. BGP couples the attributes with the route they describe, then compares all paths available to a destination to select the best route to use to reach the destination. The well-known mandatory path attributes are:

- **Origin** describes where the route was learned—it can be IGP, EGP, or incomplete.
- **AS-Path** contains a list of autonomous systems through which the route advertisement has passed.
- **Next-Hop** is the IP address of the router to which traffic for the route is sent.

The optional path attributes are:

- **Multi-Exit Discriminator** (MED) is a metric for a path where there are multiple links between ASs (the MED is set by one AS and used by another AS to choose a path).
- **Local-Pref** is a metric used to inform BGP peers of the local router's preference for the route.
- **Atomic-Aggregate** informs BGP peers that the local router selected a less-specific route from a set of overlapping routes received from a peer.
- **Aggregator** specifies the AS and router that performed aggregation of the route.
- **Communities** specifies one or more communities to which this route belongs
- **Cluster List** contains a list of the reflector clusters through which the route has passed

A BGP router can choose to add or modify the optional path attributes before advertising the route to peers.

External and Internal BGP

External BGP (EBGP) is used between autonomous systems, as when different ISP networks connect to each other or an enterprise network connects to an ISP network. Internal BGP (IBGP) is used within an AS, such as within an enterprise network. The main goal of IBGP is to distribute the routes learned from EBGP to routers in the AS. An IBGP router can readvertise routes that it learns from its EBGP peers to its IBGP peers, but it cannot advertise routes learned from IBGP peers to other IBGP peers. This restriction prevents route advertisement loops within the network, but means that an IBGP network must be fully meshed (that is, every BGP router in the network must have a session with every other router in the network).

Some path attributes are only applicable to EBGP or IBGP. For example, the MED attribute is only used for EBGP messages, while the LOCAL-PREF attribute is only present in IBGP messages.

Basic BGP Configuration

You create a BGP instance on a per-virtual router (VR) basis on a security device. If you have multiple VRs on a device, you can enable multiple instances of BGP—one instance for each VR.

NOTE: Before you configure a dynamic routing protocol on the security device, you should assign a virtual router ID, as described in "Routing" on page 1235.

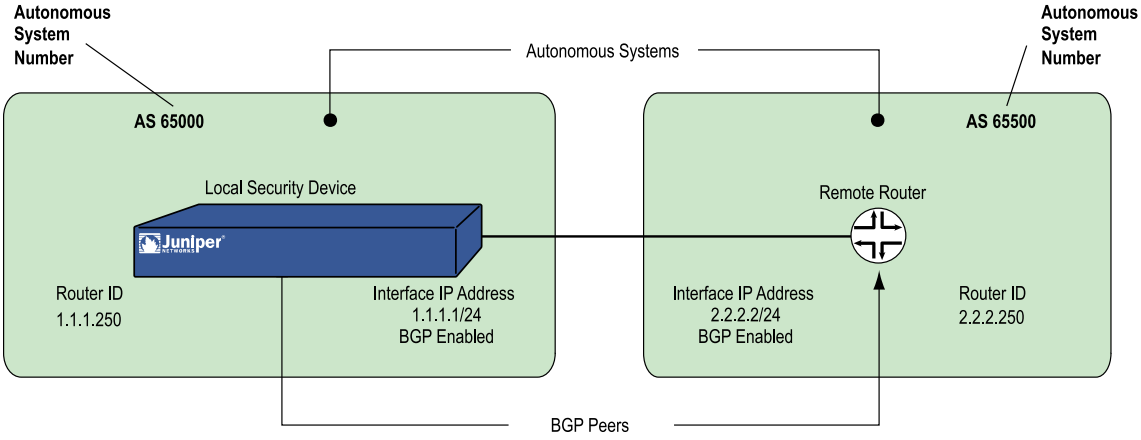
The five basic steps to configure BGP in a VR on a security device are:

1. Create and enable the BGP routing instance in a VR by first assigning an autonomous system number to the BGP instance, then enabling the instance.
2. Enable BGP on the interface that is connected to the peer.
3. Configure one or more remote BGP peers.

4. Enable each BGP peer.
5. Verify that BGP is properly configured and operating.

This section describes how to perform each of these tasks using either the CLI or the WebUI for the following example. Figure 332 on page 1341 (IPv4) and (IPv6) show the security device as a BGP peer in AS 65000. You need to configure the security device so that it can establish a BGP session with the peer in AS 65500.

Figure 332: IPv4 BGP Configuration Example



Creating and Enabling a BGP Instance

You create and enable a BGP routing instance on a specific virtual router (VR) on a security device. To create a BGP routing instance, you need to first specify the autonomous system number in which the VR resides. If the VR is an IBGP router, the autonomous system number must be the same as other IBGP routers in the network. When you enable the BGP routing instance on a VR, the BGP routing instance will be able to contact and establish a session with the BGP peers that you configure.

NOTE: Autonomous System (AS) numbers are globally unique numbers that are used to exchange EBGP routing information and to identify the AS. The following entities allocate AS numbers: the American Registry for Internet Numbers (ARIN), Réseaux IP Européens (RIPE), and Asia Pacific Network Information Center (APNIC). The numbers 64512 through 65535 are for private use and not to be advertised on the global Internet.

Creating a BGP Routing Instance

In the following example, you first assign 0.0.0.10 as the router ID for the trust-vr. You then create and enable a BGP routing instance on the trust-vr, which resides on the security device in AS 65000. (For more information about virtual routers and configuring a virtual router on security devices, see “Routing” on page 1235.)

WebUI

1. **Router ID**

   Network > Routing > Virtual Router (trust-vr) > Edit: Enter the following, then click **OK**:

   Virtual Router ID: Custom (select)
   In the text box, enter 0.0.0.10

2. **BGP Routing Instance**

   Network > Routing > Virtual Routers > Edit (for trust-vr) > Create BGP Instance: Enter the following, then click **OK**:

   AS Number (required): 65000
   BGP Enabled: (select)

CLI

1. **Router ID**

   ```
   set vrouter trust-vr router-id 10
   ```

2. **BGP Routing Instance**

   ```
   set vrouter trust-vr protocol bgp 65000
   set vrouter trust-vr protocol bgp enable
   save
   ```

Removing a BGP Instance

In this example, you disable and remove the BGP routing instance in the trust-vr. BGP stops sessions with all peers.

WebUI

Network > Routing > Virtual Routers (trust-vr) > Edit > Edit BGP Instance: Deselect BGP Enabled, then click **OK**.

Network > Routing > Virtual Routers (trust-vr) > Edit: Select **Delete BGP Instance**, then click **OK** at the confirmation prompt.

CLI

```
unset vrouter trust-vr protocol bgp enable
unset vrouter trust-vr protocol bgp 65000
save
```

Enabling and Disabling BGP on Interfaces

You must enable BGP on the interface on which the peer resides. (By default, interfaces on the security device are not bound to any routing protocol.)

Enabling BGP on Interfaces

In this example, you enable BGP on the interface ethernet4.

WebUI

Network > Interfaces > Edit > BGP: Select **Protocol BGP enable**, then click **OK**.

CLI

```
set interface ethernet4 protocol bgp
save
```

Disabling BGP on Interfaces

In this example, you disable BGP on the interface *ethernet4*. Other interfaces on which you have enabled BGP are still able to transmit and process BGP packets.

WebUI

Network > Interfaces > Configure (for ethernet4): Clear **Protocol BGP enable**, then click **OK**.

CLI

```
unset interface ethernet4 protocol bgp
save
```

Configuring BGP Peers and Peer Groups

Before two BGP devices can communicate and exchange routes, they need to identify each other so they can start a BGP session. You need to specify the IP addresses of the BGP peers and, optionally, configure parameters for establishing and maintaining the session. Peers can be either internal (IBGP) or external (EBGP) peers. For an EBGP peer, you need to specify the autonomous system in which the peer resides.

All BGP sessions are authenticated by checking the BGP peer identifier and the AS number advertised by the peers. A successful connection with a peer is logged. If anything goes wrong with the peer connection, a BGP notification message will either be sent to or received from the peer, which causes the connection to fail or close.

You can configure parameters for individual peer addresses. You can also assign peers to a *peer-group*, which then allows you to configure parameters for the peer-group as a whole.

NOTE: You cannot assign IBGP and EBGP peers to the same peer group.

Table 96 on page 1344 lists parameters you can configure for BGP peers and the default values. An "X" in the Peer column indicates a parameter you can configure for an individual peer IP address, and an "X" in the Peer Group column indicates a parameter you can configure for a peer group.

Table 96: BGP Peer and Peer Group Parameters and Default Values

BGP Parameter	Peer	Peer Group	Description	Default Value
Activate address family	X		Sets the address family for the neighbor (use before you enable the BGP neighbor). See "Enabling BGP Address Families for Neighbors" on page 1351.	On
Advertise default route	X		Advertises the default route in the virtual route to BGP peers.	Default route is not advertised
Advertisement interval	X		Configures the advertisement interval for a specific neighbor according to the address family.	Disabled
EBGP multihop	X	X	Number of nodes between local BGP and neighbor.	0 (disabled)
Force connect	X	X	Causes the BGP instance to drop an existing BGP connection with the specified peer and accept a new connection. This parameter is useful when connecting to a router that goes down, then comes back up and tries to reestablish BGP peering as it allows faster reestablishment of the peer connection. *Note:* You can use the **exec neighbor disconnect** command to cause the BGP instance to drop an existing BGP connection with the specified peer and accept a new connection. Using this exec command does not change the configuration of the BGP peer. For example, you can use this exec command if you change the configuration of the route map that is applied to the peer.	—
Hold time	X	X	Time elapsed without a message from a peer before the peer is considered down.	180 seconds
Keepalive	X	X	Time between keepalive transmissions.	1/3 of hold-time
Local preference	X		Configures the LOCAL_PREF value.	100
MD5 authentication	X	X	Configures MD-5 authentication.	Only peer identifier and AS number checked
MED	X		Configures MED attribute value.	0

Table 96: BGP Peer and Peer Group Parameters and Default Values *(continued)*

Next-hop self	X	X	For routes sent to the peer, the next hop path attribute is set to the IP address of the interface of the local virtual router.	Next hop attribute unchanged
Reflector client	X	X	Peer is a reflector client when the local BGP is set as the route reflector.	None
Reject default route	X		Ignores default route advertisements from BGP peers.	Default routes from peers are added to routing table
Remove private AS	X		Removes private AS numbers from the AS_PATH list before the routes are propagated to a BGP peer.	Private AS numbers in the AS-PATH for the peer are not removed.
Retry time	X	X	Time after a failed session attempt that the BGP session is reattempted.	120 seconds
Send community	X	X	Transmits community attribute to peer.	Community attribute not sent to peers
Weight	X	X	Priority of path between local BGP and peer.	100

You can configure some parameters at both the peer level and the protocol level (see "Configuring a Confederation" on page 1364). For example, you can configure the hold-time value for a specific peer at 210 seconds, while the default hold-time value at the protocol level is 180 seconds; the peer configuration takes precedence. You can set different MED values at the protocol level and at the peer level; the MED value you set at the peer level applies only to routes that are advertised to those peers.

Configuring a BGP Peer (IPv4)

In the following example, you configure and enable a BGP peer. This peer has the following attributes:

- IP address 1.1.1.250
- Resides in AS 65500

NOTE: You must enable each peer connection that you configure.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors: Enter the following, then click **Add**:

AS Number: 65500
Remote IP: 1.1.1.250

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors > Configure (for the peer you just added): Select **Peer Enabled**, then click **OK**.

CLI

```
set vrouter trust-vr protocol bgp neighbor 1.1.1.250 remote-as 65500
set vrouter trust-vr protocol bgp neighbor 1.1.1.250 enable
save
```

Configuring a BGP Peer (IPv6)

In the following example, you configure and enable a BGP peer. This peer has the following attributes:

- IP address 2001:0db8:3aaa::2
- Resides in AS 65500

NOTE: You must enable each peer connection that you configure.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors: Enter the following, then click **Add**:

```
AS Number: 65500
Remote IP: 2001:0db8:3aaa::2
```

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors > Configure (for the peer you just added): Select **Peer Enabled**, then click **OK**.

CLI

```
set vrouter trust-vr protocol bgp neighbor 2001:0db8:3aaa::2 remote-as 65500
set vrouter trust-vr protocol bgp neighbor 2001:0db8:3aaa::2 enable
save
```

Configuring an IBGP Peer Group (IPv4)

In the following example, you configure an IBGP peer group called **ibgp** that contains the IP addresses 10.1.2.250 and 10.1.3.250. Once you have defined a peer group, you can configure parameters (such as MD5 authentication) that apply to all members of the peer group.

NOTE: You must enable each peer connection that you configure. If you configure peers as part of a peer group, you still need to enable the peer connections one by one.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Peer Group: Enter **ibgp** for Group Name, then click **Add**.

> Configure (for ibgp): In the Peer authentication field, enter **verify03**, then click **OK**.

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors: Enter the following, then click **Add**:

AS Number: 65000
Remote IP: 10.1.2.250
Peer Group: ibgp (select)

Enter the following, then click **Add**:

AS Number: 65000
Remote IP: 10.1.3.250
Peer Group: ibgp (select)

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors > Configure (for 10.1.2.250): Select **Peer Enabled**, then click **OK**.

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors > Configure (for 10.1.3.250): Select **Peer Enabled**, then click **OK**.

CLI

```
set vrouter trust-vr protocol bgp neighbor peer-group ibgp
set vrouter trust-vr protocol bgp neighbor peer-group ibgp remote-as 65000
set vrouter trust-vr protocol bgp neighbor peer-group ibgp md5-authentication verify03
set vrouter trust-vr protocol bgp neighbor 10.1.2.250 remote-as 65000
set vrouter trust-vr protocol bgp neighbor 10.1.2.250 peer-group ibgp
set vrouter trust-vr protocol bgp neighbor 10.1.3.250 remote-as 65000
set vrouter trust-vr protocol bgp neighbor 10.1.3.250 peer-group ibgp
set vrouter trust-vr protocol bgp neighbor 10.1.2.250 enable
set vrouter trust-vr protocol bgp neighbor 10.1.3.250 enable
save
```

Configuring an IBGP Peer Group (IPv6)

In the following example, you configure an IBGP peer group called **ibgp** that contains the IP addresses 2001::2 and 2005::2. Once you have defined a peer group, you can configure parameters (such as MD5 authentication) that apply to all members of the peer group.

NOTE: You must enable each peer connection that you configure. If you configure peers as part of a peer group, you still need to enable the peer connections one by one.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Peer Group: Enter **ibgp** for Group Name, then click **Add**.

> Configure (for ibgp): In the Peer authentication field, enter **verify03**, then click **OK**.

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors: Enter the following, then click **Add**:

AS Number: 65000
Remote IP: 2001::2
Peer Group: ibgp (select)

Enter the following, then click **Add**:

AS Number: 65000
Remote IP: 2005::2
Peer Group: ibgp (select)

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors > Configure (for 2001::2): Select **Peer Enabled**, then click **OK**.

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors > Configure (for 2005::2): Select **Peer Enabled**, then click **OK**.

CLI

```
set vrouter trust-vr protocol bgp neighbor peer-group ibgpv6
set vrouter trust-vr protocol bgp neighbor peer-group ibgpv6 remote-as 65000
set vrouter trust-vr protocol bgp neighbor peer-group ibgpv6 md5-authentication
verify03
set vrouter trust-vr protocol bgp neighbor 2001::2 remote-as 65000
set vrouter trust-vr protocol bgp neighbor 2001::2 peer-group ibgpv6
set vrouter trust-vr protocol bgp neighbor 2005::2 remote-as 65000
set vrouter trust-vr protocol bgp neighbor 2005::2 peer-group ibgpv6
set vrouter trust-vr protocol bgp neighbor 2001::2 enable
set vrouter trust-vr protocol bgp neighbor 2005::2 enable
save
```

Verifying the BGP Configuration

You can review the configuration you entered through the WebUI or the CLI with the **get vrouter** *vrouter* **protocol bgp config** command.

```
device->  get vrouter trust-vr protocol bgp config
set protocol bgp 65000
set enable
set neighbor peer-group "ibgp"
set neighbor peer-group "ibgp" md5-authentication "cq1tu6gVNU5gvfsO60CsvxVPNnntO
PwY/g=="
set neighbor 10.1.2.250 remote-as 65000

output continues...

exit
```

You can verify that BGP is running on the virtual router by executing the **get vrouter** *vrouter* **protocol bgp** command.

```
device->  get vrouter trust-vr protocol bgp
Admin State:           enable
Local Router ID:       10.1.1.250
Local AS number:       65000
Hold time:             180
Keepalive interval:    60 = 1/3 hold time, default
Local MED is:          0
Always compare MED:    disable
Local preference:      100
Route Flap Damping:    disable
IGP synchronization:   disable
Route reflector:       disable
Cluster ID:            not set (ID = 0)
Confederation based on RFC 1965
Confederation:         disable (confederation ID = 0)
Member AS:             none
Origin default route: disable
Ignore default route: disable
```

You can view the administrative state of the virtual router (VR) and the router ID, as well as all other configured parameters particular to BGP.

NOTE: We recommend that you explicitly assign a router ID rather than use the default. For information on setting a router ID, see "Routing" on page 1235.

You can verify that a BGP peer or peer group is enabled and see the state of the BGP session by executing the **get vrouter** vrouter **protocol bgp neighbor** command.

```
device-> get vrouter trust-vr protocol bgp neighbor
Peer AS Remote IP   Local IP     Wt Status   State  ConnID
  65500 1.1.1.250    0.0.0.0    100 Enabled ACTIVE          up
Total 1 BGP peers shown
```

In this example you can verify that the BGP peer is enabled and the session is active.

The state can be one of the following:

- **Idle**—The first state of the connection
- **Connect**—BGP is waiting for successful TCP transport connection

- **Active**—BGP is initiating a transport connection
- **OpenSent**—BGP is waiting for an OPEN message from the peer
- **OpenConfirm**—BGP is waiting for a KEEPALIVE or NOTIFICATION message from the peer
- **Established**—BGP is exchanging UPDATE packets with the peer

NOTE: A session state that continually changes between the Active and Connect may indicate a problem with the connection between the peers.

There are also extended BGP configuration choices, including viewing neighbors, enabling address families, and advertising IPv6 routes.

Viewing BGP Advertised and Received Routes for Neighbors

You can view advertised routes or received routes for all neighbors or for specific neighbors with the **get vrouter** command. Each BGP received route can come from only one fixed neighbor, but each BGP route can be advertised to multiple BGP neighbors.

In the following example, the **get vrouter** *vrouter* **protocol bgp ipv4 rib-in neighbor** *neighbor_address* **received** command shows the received routes for the specified IPv4 neighbor (1.1.1.10):

```
ISG2k-> get vr trust protocol bgp ipv4 rib nei 1.1.1.10 received
i: IBGP route, e: EBGP route, >: best route, *: valid route
               Prefix         Nexthop    Wt  Pref   Med Orig    AS-Path
-------------------------------------------------------------------------------
>e*    100.65.37.0/30        1.1.1.10   100   100     0  INC     10
>e*    100.66.38.0/30        1.1.1.10   100   100     0  INC     10
>e*    100.67.39.0/30        1.1.1.10   100   100     0  INC     10
>e*    100.68.32.0/30        1.1.1.10   100   100     0  INC     10
>e*    100.69.33.0/30        1.1.1.10   100   100     0  INC     10
>e*    100.70.34.0/30        1.1.1.10   100   100     0  INC     10
>e*    100.71.35.0/30        1.1.1.10   100   100     0  INC     10
>e*    100.72.44.0/30        1.1.1.10   100   100     0  INC     10
>e*    100.73.45.0/30        1.1.1.10   100   100     0  INC     10
Total IPv4 routes received: 9
```

In the following example, the **get vrouter** *vrouter* **protocol bgp ipv6 rib-in neighbor** *neighbor_address* **advertised** command shows the advertised routes for the specified IPv6 neighbor (2008::5):

```
ISG2k-> get vr trust protocol bgp ipv6 rib nei 2008::5 advertised
i: IBGP route, e: EBGP route, >: best route, *: valid route
                                        Prefix        Wt  Pref   Med Orig
                                       Nexthop    AS-Path
-------------------------------------------------------------------------------
>e*                                2222::3/128       100   100     0  IGP
                                      1044::2      10
>e*                                 2005::/64       100   100     0  IGP
                                      1044::2      10
```

```
>e*                                3abd::/64        100    100      0  IGP
                                     1044::2       10
>i                                 3abc::/64      32768    100      0  IGP
                                          ::
>i                                 2ddd::/64      32768    100      0  IGP
                                          ::
Total IPv6 routes advertised: 5
```

For both examples, the column headings are described as follows:

- Prefix—the prefix for the routing table entry
- Next-hop—the IP address of the next hop
- Wt—the assigned path weight
- Pref—the local preference for the route
- Med—the multi-exit discriminator for the route
- Orig—the origin of the route
- AS-Path—the value of the AS-path attribute

Enabling BGP Address Families for Neighbors

By default, the IPv4 address family is supported for an IPv4 neighbor and the IPv6 address family is supported for an IPv6 neighbor when the neighbor is created. You can set or unset the address family for the neighbor by using the **set protocol bgp** *address_family* **neighbor** *ip_address* **activate** command. Use this command before you enable the BGP neighbor so that the new address family can be negotiated through open messages at the session start stage.

The following example configures the address family for one IPv4 peer.

CLI

```
device(trust-vr)-> set protocol bgp neighbor 40.0.0.40 remote-as 20
device(trust-vr)-> set protocol bgp ipv6 neighbor 40.0.0.40 activate
device(trust-vr)-> set protocol bgp neighbor 40.0.0.40 enable
```

The following example configures the address family for one IPv6 peer.

CLI

```
device(trust-vr)-> set protocol bgp neighbor 3abc::6 remote-as 20
device(trust-vr)-> set protocol bgp ipv4 neighbor 3abc::6 activate
device(trust-vr)-> set protocol bgp neighbor 3abc::6 enable
```

Advertising IPv6 Routes Between IPv4 BGP Peers and IPv4 Routes Between IPv6 BGP Peers

If an IPv4 network is connecting two separate IPv6 networks, you can use IPv4 peers to advertise the IPv6 routes. To do so, configure the peering by using the IPv4 addresses within the IPv6 address family. Because the advertised next hop will usually

be unreachable, set the next hop with a static route or with an inbound route map. Advertising IPv4 routes between two IPv6 peers uses the same model.

The following example advertises IPv6 routes between IPv4 peers when the IPv4 network is connecting two separate IPv6 networks. Peering is configured using IPv4 addresses in the IPv6 address family configuration mode. The inbound route map named rtmap6 sets the next hop because the advertised next hop is likely to be unreachable.

CLI

```
device-> set vrouter trust
device(trust-vr)-> set protocol bgp 45000
device(trust-vr/bgp)-> set enable

device(trust-vr/bgp)-> set neighbor 10.1.1.2 remote-as 45002
device(trust-vr/bgp)-> set ipv6 neighbor 10.1.1.2 activate
device(trust-vr/bgp)-> set ipv6 neighbor 10.1.1.2 route-map rtmapv6 in
device(trust-vr/bgp)-> set neighbor 10.1.1.2 enable
device(trust-vr/bgp)-> exit
device(trust-vr)-> exit
device-> set vrouter trust
device(trust-vr)-> set access-list ipv6 1 permit ip ::/0 1
device(trust-vr)-> set route-map ipv6 name rtmapv6 permit 10
device(trust-vr/rtmapv6-10)-> set match ip 1
device(trust-vr/rtmapv6-10)-> set next-hop  2007:3bbb::5
device(trust-vr/rtmapv6-10)-> exit
device(trust-vr)->
```

The following example advertises IPv4 routes between IPv6 peers when the IPv6 network is connecting two separate IPv4 networks. In this case, peering is configured using IPv6 addresses in the IPv4 address family configuration mode. As in the previous example, the inbound route map **(rtmapv4)** sets the next hop because the advertised next hop is likely to be unreachable.

CLI

```
device-> set vrouter trust
device(trust-vr)-> set protocol bgp 65000
device(trust-vr/bgp)-> set enable

device(trust-vr/bgp)-> set neighbor 2001:0db8:3aaa::2 remote-as 45002
device(trust-vr/bgp)-> set ipv4 neighbor  2001:0db8:3aaa::2 activate
device(trust-vr/bgp)-> set ipv4 neighbor  2001:0db8:3aaa::2 route-map
rtmapv4 in
device(trust-vr/bgp)-> set neighbor 2001:0db8:3aaa::2 enable
device(trust-vr/bgp)-> exit
device(trust-vr)-> exit
device-> set vrouter trust
device(trust-vr)-> set access-list 2 permit ip 0.0.0.0/0 2
device(trust-vr)-> set route-map name rtmapv4 permit 1
device(trust-vr/rtmapv4-10)-> set match ip 2
device(trust-vr/rtmapv4-10)-> set next-hop  10.5.5.5
```

```
device(trust-vr/rtmapv4-10)-> exit
device(trust-vr)->
```

Security Configuration

This section describes possible security problems in the BGP routing domain and methods of preventing attacks.

NOTE: To make BGP more secure, you should configure all routers in the BGP domain to be at the same security level. Otherwise, a compromised BGP router can bring down the entire BGP routing domain.

Authenticating BGP Neighbors

A BGP router can be easily spoofed, since BGP packets are not encrypted and most protocol analyzers provide decapsulation of BGP packets. Authenticating BGP peers is the best way to fend off these types of attacks.

BGP provides MD5 authentication to validate BGP packets received from peers. MD5 authentication requires that the same key be used for both the sending and receiving BGP routers. All BGP packets received from the specified peer that are not authenticated are discarded. By default, only the peer identifier and AS number are checked for a BGP peer.

In the following example, you first configure a BGP peer with the remote IP address 1.1.1.250 in AS 65500. You then configure the peer for MD5 authentication using the key 1234567890123456.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Neighbors: Enter the following, then click **Add**:

AS Number: 65500
Remote IP: 1.1.1.250

> Configure (for Remote IP 1.1.1.250): Enter the following, then click **OK**:

Peer Authentication: Enable (select)
MD5 password: 1234567890123456
Peer Enabled: (select)

CLI

```
set vrouter trust-vr
(trust-vr)-> set protocol bgp
(trust-vr/bgp)-> set neighbor 1.1.1.250 remote-as 65500
(trust-vr/bgp)-> set neighbor 1.1.1.250 md5-authentication 1234567890123456
(trust-vr/bgp)-> set neighbor 1.1.1.250 enable
(trust-vr/bgp)-> exit
```

```
(trust-vr)-> exit
save
```

Rejecting Default Routes

In a route detour attack, a router injects a default route (it can be an IPv4 or IPv6 default route) into the routing domain in order to detour packets to itself. The router can then either drop the packets, causing service disruption, or it can remove sensitive information in the packets before forwarding them. On security devices, BGP by default accepts any default routes that are sent from BGP peers and adds the default route to the routing table.

In this example, you configure the BGP routing instance running in the trust-vr to ignore any default routes that are sent from BGP peers.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Edit BGP Instance: Enter the following, then click **OK**:

Ignore default route from peer: (select)

CLI

```
set vrouter trust-vr protocol bgp reject-default-route
save
```

Optional BGP Configurations

This section describes the parameters you can configure for the BGP routing protocol in the virtual router. You can configure these parameters with either the CLI BGP context commands or the WebUI. This section explains some of the more complex parameter configurations. Table 97 on page 1354 describes BGP parameters and their default values.

Table 97: Optional BGP Parameters and Default Values

BGP Protocol Parameter	Description	Default Value
Advertise default route	Advertise the default route in the virtual router to BGP peers.	Default route not advertised
Aggregate	Create aggregated routes. See "Route Aggregation" on page 1367.	—
Always compare MED	Compare MED values in routes.	Disabled
AS confederation	Create confederations. See "Configuring a Confederation" on page 1364.	—
AS path access list	Create an AS path access list to permit or deny routes. See "Configuring an AS-Path Access List" on page 1357.	—

Table 97: Optional BGP Parameters and Default Values *(continued)*

Community list	Create community lists. See "BGP Communities" on page 1366.	—
Equal cost multipath (ECMP)	Equal cost multiple routes can be added to provide load-balancing capabilities. See "Configuring Equal Cost Multipath Routing" on page 1259.	Disabled (default = 1)
Flap damping	Block advertisement of a route until it becomes stable.	Disabled
Hold time	Time elapsed without a message from a peer before the peer is considered down.	180 seconds
Keepalive	Time between keepalive transmissions.	1/3 of hold-time
Local preference	Configure LOCAL_PREF metric.	100
MED	Configure MED attribute value.	0
Network	Add static network and subnetwork entries into BGP. BGP advertises these static routes to all BGP peers. See "Adding Routes to BGP" on page 1357.	—
Route redistribution	Import routes into BGP from other routing protocols. See "Redistributing Routes into BGP" on page 1355.	—
Reflector	Configure the local BGP instance as a route reflector to clients. See "Configuring Route Reflection" on page 1362.	Disabled
Reject default route	Ignore default route advertisements from BGP peers. See "Rejecting Default Routes" on page 1354.	Default routes from peers are added to routing table
Retry time	Time after an unsuccessful BGP session establishment with a peer that session establishment is retried.	12 seconds
Synchronization	Enable synchronization with an IGP, such as OSPF or RIP.	Disabled

Redistributing Routes into BGP

Route redistribution is the exchange of route information between routing protocols. For example, you can redistribute the following types of routes into the BGP routing instance in the same virtual router:

- Routes learned from OSPF (IPv4) or RIP (IPv4 and IPv6)
- Directly connected routes

- Imported routes
- Statically configured routes

When you configure route redistribution, you must first specify a route map to filter the routes that are redistributed. For more information about creating route maps for route redistribution, see “Routing” on page 1235.

In the following example, you redistribute a route that originated from a Routing Information Protocol (RIP) routing domain into the current Border Gateway Protocol (BGP) routing domain. Both the CLI and WebUI examples assume that you previously created a route map called add-rip.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Redist. Rules: Enter the following, then click **Add**:

```
Route Map: add-rip
Protocol: rip
```

CLI

```
set vrouter trust-vr protocol bgp redistribute route-map add-rip protocol rip
save
```

Maximum Routes for Redistribution

For each virtual router in BGP, 17000 routes for redistribution are available.

The increase in redistributable routes in BGP to 17000 applies to the following NetScreen–5000 platforms:

- NetScreen-5000 Series using Management Module 2 and NS-5000-8G2 and NS-5000-2XGE Secure Port Modules
- NetScreen-5000 Series using Management Module 3 (NS-5000-MGT3) and NS-5000-8G2-G4 and NS-5000-2XGE-G4 Secure Port Modules

Each virtual router (VR) can redistribute up to 17000 routes into BGP.

- The 17000 redistributable routes limit is not restricted to the static routes for each VR. The increased limit encompasses all source protocols, including static routes, connect routes, and RIP and OSPF routes that can be redistributed into BGP protocol.
- In ScreenOS 6.2 and earlier releases, the maximum redistributed route value of 6000 was used by both BGP and OSPF. Beginning with the ScreenOS 6.3.0 release, the increase in maximum redistributed routes to 17000 applies to BGP. However, OSPF continues to have 6000 maximum redistribute routes.
- Because of the task schedule order impact, the maximum redistributed routes might exceed the value of 17000. Therefore, the actual maximum redistributed routes into BGP can be greater than or equal to 17000, but not less than 17000.

Configuring an AS-Path Access List

The AS-path attribute contains a list of the autonomous systmes (ASs) through which a route has traversed. BGP prepends the local AS number to the AS-path attribute when a route passes through the AS. You can use an *AS-path* access list to filter routes based on the AS-path information. An AS-path access list consists of a set of regular expressions that define AS-path information and whether the routes that match the information are permitted or denied. For example, you can use an AS-path access list to filter routes that have passed through a particular AS or routes that originated in a particular AS.

Regular expressions are a way to define a search for specific patterns in the AS-path attribute. You can use special symbols and characters in constructing a regular expression. For example, to match routes that have passed through AS 65000, use the regular expression **_65000_** (the underscores match any characters before or after 65000). You can use the regular expression 65000$ to match routes that originated in AS 65000 (the dollar sign matches the end of the AS-path attribute, which would be the AS where the route originated).

The following example configures an AS-path access list for the trust-vr that allows routes that have passed through AS 65000 but does not allow routes that originated in AS 65000.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > AS Path: Enter the following, then click **Add**:

```
AS Path Access List ID: 2
Deny: (select)
AS Path String: 65000$
```

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > AS Path: Enter the following, then click **Add**:

```
AS Path Access List ID: 2
Permit: (select)
AS Path String: _65000_
```

CLI

```
set vrouter trust-vr protocol bgp as-path-access-list 2 deny 65000$
set vrouter trust-vr protocol bgp as-path-access-list 2 permit _65000_
save
```

Adding Routes to BGP

To allow BGP to advertise network routes, you need to redistribute the routes from the source protocol into the advertising protocol (BGP) in the same virtual router (VR). You can also add static routes directly into BGP. If the network prefix is reachable from the VR, BGP advertises this route to peers without requiring that the route be

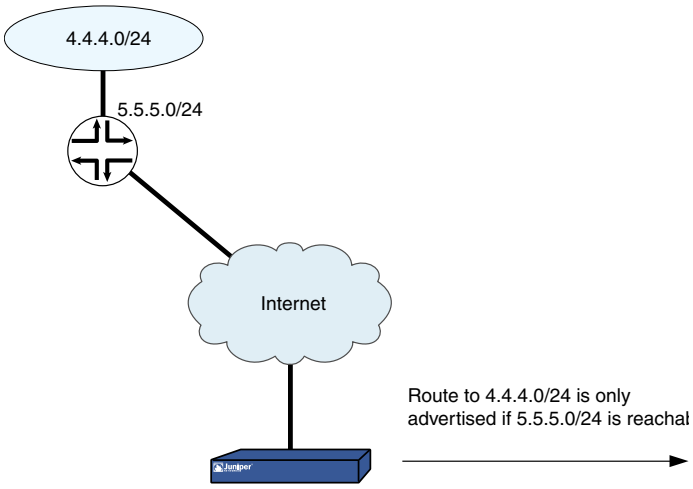
redistributed into BGP. When you add a network prefix into BGP, you can specify several options:

- By selecting **By default** to the check reachability option, you can specify whether the network prefix must be reachable from the VR before BGP advertises the route to peers.
- By selecting **Yes** to the check reachability option, you can specify whether a *different* network prefix must be reachable from the VR before BGP advertises the route to peers. For example, if the prefix you want BGP to advertise must be reached through a specific router interface, you want to ensure that the router interface is reachable before BGP advertises the network to peers. If the router interface you specify is reachable, BGP advertises the route to its peers. If the router interface you specify is not reachable, the route is not added to BGP and is consequentially not advertised to BGP peers. If the router interface you specify becomes unreachable, BGP withdraws the route from its peers.
- By selecting **No Check** to the check reachability option, you can specify that the network prefix always be advertised whether reachable from the VR or not. By default, the network prefix must be reachable from the VR before BGP advertises the route to peers. If you enable check reachability, the route can be connected.
- You can assign a *weight* value to the network prefix. The weight is an attribute that you can assign locally to a route; it is not advertised to peers. If there is more than one route to a destination, the route with the highest weight value is preferred.
- You can set the attributes of the route to those specified in a route map (see "Configuring a Route Map" on page 1262). BGP advertises the route with the route attributes specified in the route map.

Conditional Route Advertisement

In the following example, you add a static route to the network 4.4.4.0/24. You specify that the router interface 5.5.5.0/24 must be reachable from the virtual router in order for BGP to advertise the 4.4.4.0/24 route to peers. If the 5.5.5.0/24 network is not reachable, BGP does not advertise the 4.4.4.0/24 network. See Figure 333 on page 1359.

Figure 333: Conditional BGP Route Advertisement Example



WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Networks: Enter the following, then click **Add**:

IP/Netmask: 4.4.4.0/24
Check Reachability:
Yes: (select), 5.5.5.0/24

CLI

```
set vrouter trust-vr protocol bgp network 4.4.4.0/24 check 5.5.5.0/24
save
```

Setting the Route Weight

In the following example, you set a weight value of 100 for the route 4.4.4.0/24. (You can specify a weight value between 0 and 65535.)

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Networks: Enter the following, then click **Add**:

IP/Netmask: 4.4.4.0/24
Weight: 100

CLI

```
set vrouter trust-vr protocol bgp network 4.4.4.0/24 weight 100
save
```

Setting Route Attributes

In the following example, you configure a route map setattr that sets the metric for the route to 100. You then configure a static route in BGP that uses the route map *setattr*. (You do not need to set the route map to match the network prefix of the route entry.)

WebUI

Network > Routing > Virtual Router > Route Map (for trust-vr) > New: Enter the following, then click **OK**:

Map Name: setattr
Sequence No.: 1
Action: Permit (select)
Set Properties:
Metric: (select), 100

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Networks: Enter the following, then click **Add**:

IP/Netmask: 4.4.4.0/24
Route Map: setattr (select)

CLI

```
set vrouter trust-vr route-map name setattr permit 1
set vrouter trust-vr route-map setattr 1 metric 100
set vrouter trust-vr protocol bgp network 4.4.4.0/24 route-map setattr
save
```

Route-Refresh Capability

The BGP route-refresh feature as defined in RFC 2918 provides a soft reset mechanism that allows the dynamic exchange of route refresh requests and routing information between BGP peers and the subsequent readvertisement of the outbound or inbound routing table.

Routing policies for a BGP peer using route maps might impact inbound or outbound routing table updates because whenever a route policy change occurs, the new policy takes effect only after the BGP session is reset. A BGP session can be cleared through a hard or soft reset.

NOTE: A hard reset is disruptive because active BGP sessions are torn down and brought back up.

A soft reset allows the application of a new or changed policy without clearing an active BGP session. The route-refresh feature allows a soft reset to occur on a per-neighbor basis and does not require preconfiguration or extra memory.

A dynamic inbound soft reset is used to generate inbound updates from a neighbor. An outbound soft reset is used to send a new set of updates to a neighbor. Outbound resets don't require preconfiguration or routing table update storage.

The route refresh feature requires that both BGP peers advertise route-refresh feature support in the OPEN message. If the route-refresh method is successfully negotiated, either BGP peer can use the route-refresh feature to request full routing information from the other end.

NOTE: Using the **get neighbor** *ip_addr* command, an administrator can check whether the route-refresh capability is negotiated. The command also displays counters, such as the number of times the route-refresh request is sent or received.

Requesting an Inbound Routing Table Update

In this example, you request the inbound routing table of the neighboring peer at 10.10.10.10 (IPv4) or 2001:0db8:3aaa::2 (IPv6) to be sent to the trust-vr of the local BGP peer by using the **soft-in** command.

NOTE: If the route refresh feature is not available, the command reports an exception when the administrator tries to use it.

WebUI

This feature is not available in the WebUI.

CLI (IPv4)

```
clear vrouter trust-vr protocol bgp neighbor 10.10.10.10 soft-in
```

CLI (IPv6)

For routes of the IPv4 address family:

```
clear vrouter trust-vr protocol bgp neighbor 2001:0db8:3aaa::2 soft-in
```

For routes of the IPv6 address family:

```
clear vrouter trust-vr protocol bgp ipv6 neighbor 2001:0db8:3aaa::2 soft-in
```

Requesting an Outbound Routing Table Update

In this example, you send the full routing table for the trust-vr through updates from the local BGP peer to the neighboring peer at 10.10.10.10 (IPv4) or 2001:0db8:3aaa::2 (IPv6) by using the soft-out command.

WebUI

This feature is not available in the WebUI.

CLI (IPv4)

```
clear vrouter trust-vr protocol bgp neighbor 10.10.10.10 soft-out
```

CLI (IPv6)

For routes of the IPv4 address family:

```
clear vrouter trust-vr protocol bgp neighbor 2001:0db8:3aaa::2 soft-out
```

For routes of the IPv6 address family:

```
clear vrouter trust-vr protocol bgp ipv6 neighbor 2001:0db8:3aaa::2 soft-out
```

Configuring Route Reflection

Because an IBGP router cannot readvertise routes learned from one IBGP peer to another IBGP peer (see "External and Internal BGP" on page 1340), a *full mesh* of IBGP sessions is required where each router in a BGP AS is a peer to every other router in the AS.

NOTE: Having a full mesh does not mean each pair of routers needs to be directly connected, but each router needs to be able to establish and maintain an IBGP session with every other router.

A full-mesh configuration of IBGP sessions does not scale well. For example, in an AS with eight routers, each of the eight routers would need to peer with the seven other routers, which can be calculated with this formula:

$$x \cdot (x - 1)/2$$

For an AS containing 8 routers, the number of full-mesh IBGP sessions would be 28.

Route reflection is a method for solving the IBGP scalability problem (described in RFC 1966). A *route reflector* is a router that passes IBGP learned routes to specified IBGP neighbors (*clients*), thus eliminating the need for full-mesh sessions. The route reflector and its clients make up a *cluster*, which you can further identify with a cluster ID. Routers outside of the cluster treat the entire cluster as a single entity, instead of interfacing with each individual router in full mesh. This arrangement greatly reduces overhead. The clients exchange routes with the route reflector, while the route reflector reflects routes between clients.

The local virtual router (VR) of the security device can act as a route reflector and can be assigned a cluster ID. If you specify a cluster ID, the BGP routing instance appends the cluster ID to the Cluster-List attribute of a route. The cluster ID helps

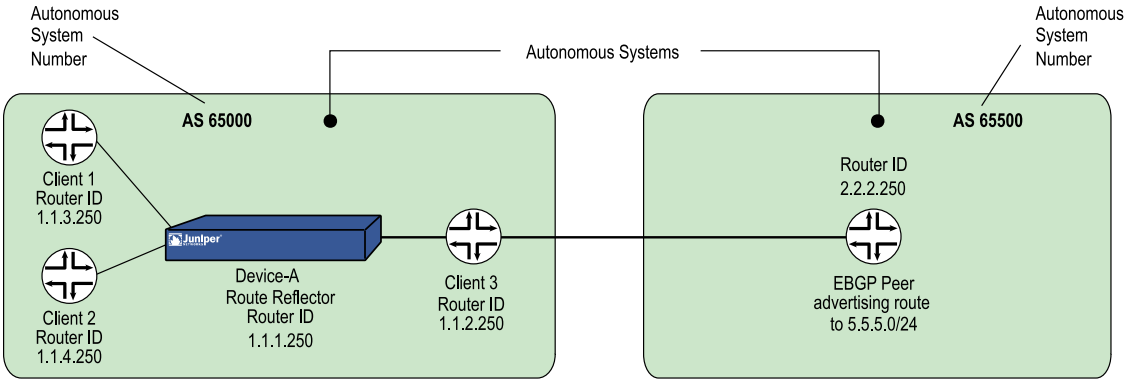
prevent routing loops as the local BGP routing instance drops a route when its cluster ID appears in the route's cluster list.

NOTE: Before you can configure a cluster ID, the BGP routing instance must be disabled.

After you set up a route reflector on the local VR, you then define the route reflector's clients. You can specify individual IP addresses or a peer-group for the clients. You do not need to configure anything on the clients.

In the following example, the EBGP router advertises the 5.5.5.0/24 prefix to Client 3. Without route reflection, Client 3 advertises the route to Device-A, but Device-A does not readvertise that route to Clients 1 and 2. If you configure Device-A as the route reflector with Clients 1, 2, and 3 as its clients, Device-A readvertises routes received from Client 3 to Clients 1 and 2. See Figure 334 on page 1363.

Figure 334: BGP Route Reflection Example



WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance: Enter the following, then click **Apply**:

Route reflector: Enable
Cluster ID: 99

> Neighbors: Enter the following, then click **Add**:

AS Number: 65000
Remote IP: 1.1.2.250

Enter the following, then click **Add**:

AS Number: 65000
Remote IP: 1.1.3.250

Enter the following, then click **Add**:

AS Number: 65000
Remote IP: 1.1.4.250

> Configure (for Remote IP 1.1.2.250): Select **Reflector Client**, then click **OK**.

> Configure (for Remote IP 1.1.3.250): Select **Reflector Client**, then click **OK**.

> Configure (for Remote IP 1.1.4.250): Select **Reflector Client**, then click **OK**.

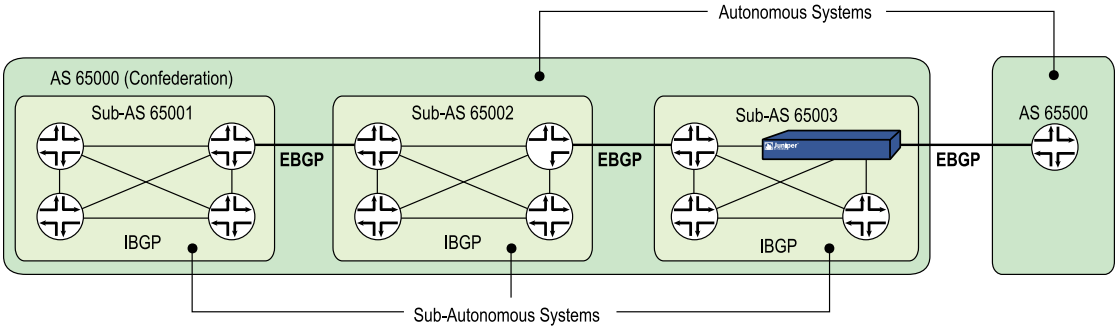
CLI

```
set vrouter trust-vr protocol bgp reflector
set vrouter trust-vr protocol bgp reflector cluster-id 99
set vrouter trust-vr protocol bgp neighbor 1.1.2.250 remote-as 65000
set vrouter trust-vr protocol bgp neighbor 1.1.2.250 reflector-client
set vrouter trust-vr protocol bgp neighbor 1.1.3.250 remote-as 65000
set vrouter trust-vr protocol bgp neighbor 1.1.3.250 reflector-client
set vrouter trust-vr protocol bgp neighbor 1.1.4.250 remote-as 65000
set vrouter trust-vr protocol bgp neighbor 1.1.4.250 reflector-client
save
```

Configuring a Confederation

Like route reflection (see "Configuring Route Reflection" on page 1362), *confederations* are another approach to solving the problem of full-mesh scaling in an IBGP environment and are described in RFC 1965. A confederation splits an autonomous system into several smaller ASs, with each sub-AS a fully meshed IBGP network. A router outside the confederation sees the entire confederation as a single AS with a single identifier; the sub-AS networks are not visible outside the confederation. Sessions between routers in two different sub-ASs in the same confederation, known as EIBGP sessions, are essentially EBGP sessions between autonomous systems, but the routers also exchange routing information as if they were IBGP peers. Figure 335 on page 1364 illustrates BGP confederations.

Figure 335: BGP Confederations



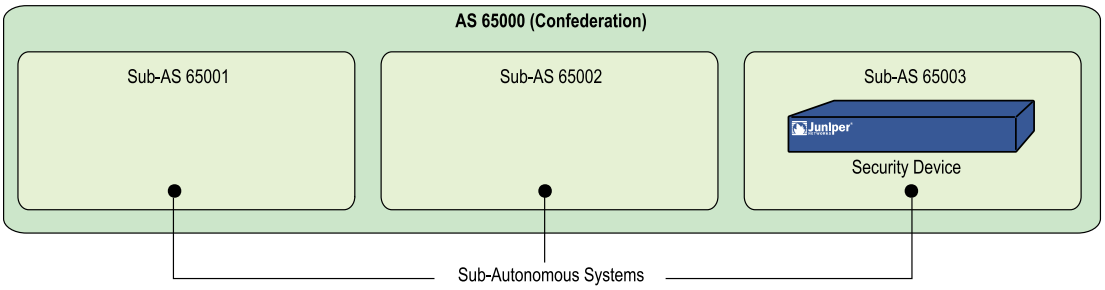
For each router in a confederation, you need to specify the following:

- The sub-AS number (this is the AS number that you specify when you create the BGP routing instance)
- The confederation to which the sub-AS belongs (this is the AS number that is visible to BGP routers outside the confederation)
- The peer sub-AS numbers in the confederation
- Whether the confederation supports RFC 1965 (the default) or RFC 3065

NOTE: The AS-Path attribute (see "Path Attributes" on page 1339) is normally composed of a sequence. RFC 3065 allows for the AS-Path attribute to include the member ASs in the local confederation traversed by the routing update.

Figure 336 on page 1365 shows the security device as a BGP router in sub-AS 65003 that belongs to the confederation 65000. The peer sub-ASs in confederation 65000 are 65002 and 65003.

Figure 336: BGP Confederation Configuration Example



WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Create BGP Instance: Enter the following, then click **Apply**:

AS Number (required): 65003

> Confederation: Enter the following, then click **Apply:**

Enable: (select)
ID: 65000
Supported RFC: RFC 1965 (select)

Enter the following, then click **Add**:

Peer member area ID: 65001

Enter the following, then click **Add**:

Peer member area ID: 65002

> Parameters: Select **BGP Enable**

CLI

```
set vrouter trust-vr protocol bgp 65003
set vrouter trust-vr protocol bgp confederation id 65000
set vrouter trust-vr protocol bgp confederation peer 65001
set vrouter trust-vr protocol bgp confederation peer 65002
set vrouter trust-vr protocol bgp enable
save
```

BGP Communities

The communities path attribute provides a way of grouping destinations (called communities), which a BGP router can then use to control the routes it accepts, prefers, or redistributes to peers. A BGP router can either append communities to a route (if the route does not have a communities path attribute) or modify the communities in a route (if the route contains a communities path attribute). The communities path attribute provides an alternative to distributing route information based on IP address prefixes or AS path attribute. You can use the communities path attribute in many ways, but its primary purpose is to simplify configuration of routing policies in complex networking environments.

RFC 1997 describes the operation of BGP communities. An AS administrator can assign the same community to a set of routes that require the same routing decisions; this is sometimes called *route coloring*. For example, you can assign one community value to routes that receive access to the Internet and a different community value to routes that do not.

There are two forms of communities:

- A *specific community* consists of the AS identifier and a community identifier. The community identifier is defined by the AS administrator.
- A *well-known community* signifies special handling for routes that contain these community values. The following are well-known community values that you can specify for BGP routes on the security device:
  - **no-export**: Routes with this communities path attribute are not advertised outside a BGP confederation.
  - **no-advertise**: Routes with this communities path attribute are not advertised to other BGP peers.
  - **no-export-subconfed**: Routes with this communities path attribute are not advertised to EBGP peers.

You can use a route map to filter routes that match a specified community list, remove or set the communities path attributes in routes, or add or delete communities from the route.

For example, if an ISP provides Internet connectivity to its customers, then all routes from those customers can be assigned a specific community number. Those customer routes are then advertised to peer ISPs. Routes from other ISPs are assigned different community numbers and are not advertised to peer ISPs.

Display Format of BGP Community lists

Beginning with ScreenOS 6.3.0, the configuration file displays the BGP community lists in a new format. This new format is in compliance with RFC-1997. The new format of the community file is as follows:

AA NN

where:

- AA = A number that identifies the autonomous system.
- NN = A number that identifies the community within the autonomous system.

For example, in 65535 300, 65535 identifies the autonomous system, and 300 identifies the community within the autonomous system 65535.

In earlier releases, the display of community lists in the configuration file was in the 4-byte number format. A sample of the earlier format:

```
ssg20(trust-vr/bgp)-> get config | include community-list
set community-list 30 permit 2031615
set community-list 4 permit 66191372
```

With the implementation of the new format for community list display, the list displays as follows:

```
isg2000-C(trust-vr/bgp)(M)-> get config | include community-list
set community-list 1 permit as 65535 300
set community-list 45 permit as 1000 11
```

The administrator can configure the BGP community lists using both the older format of 4-byte number and new format (autonomous system and community number).

Route Aggregation

Aggregation is a technique for summarizing ranges of routing addresses (known as *contributing routes*) into a single route entry. There are various optional parameters you can set when configuring an aggregated route. This section presents examples of aggregate route configuration.

Aggregating Routes with Different AS Paths

When you configure an aggregate route, you can specify that the AS-Set field in the BGP AS-Path path attribute includes the AS paths of all contributing routes. To specify this, use the AS-Set option in the aggregate route configuration.

NOTE: If you use the **AS-Set** option with an aggregated route, a change in a contributing route can cause the path attribute in the aggregated route to also change. This causes BGP to readvertise the aggregated route with the changed path attribute.

WebUI (IPv4)

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Aggregate Address: Enter the following, then click **Apply**:

Aggregate State: Enable (select)
IP/Netmask: 1.0.0.0/8
AS-Set: (select)

WebUI (IPv6)

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Aggregate Address: Enter the following, then click **Apply**:

IPv6 Aggregate State: Enable (select)
IP/Prefix: 2aaa:77::/32
AS-Set: (select)

CLI (IPv4)

```
set vrouter trust protocol bgp
set vrouter trust protocol bgp aggregate
set vrouter trust protocol bgp aggregate 1.0.0.0/8 as-set
set vrouter trust protocol bgp enable
save
```

CLI (IPv6)

```
set vrouter trust protocol bgp
set vrouter trust protocol bgp ipv6 aggregate
set vrouter trust protocol bgp ipv6 aggregate 2aaa:77::/32 as-set
set vrouter trust protocol bgp enable
save
```

NOTE: You must enable BGP aggregation before enabling BGP.

Suppressing More-Specific Routes in Updates

When you configure an aggregate route, you can specify that more-specific routes be filtered out from routing updates. (A BGP peer prefers a more-specific route, if advertised, to an aggregate route.) You can suppress more-specific routes in one of two ways:

- Use the **Summary-Only** option in the aggregate route configuration to suppress all more-specific routes.
- Use the **Suppress-Map** option in the aggregate route configuration to suppress routes that are specified by a route map.

In the following example, BGP advertises the aggregate route 1.0.0.0/8, but more-specific routes are filtered out from outgoing route updates.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Aggregate Address: Enter the following, then click **Apply**:

Aggregate State: Enable (select)
IP/Netmask: 1.0.0.0/8
Suppress Option: Summary-Only (select)

CLI

```
set vrouter trust protocol bgp aggregate 1.0.0.0/8 summary-only
save
```

In the next example, you want routes in the 1.2.3.0/24 range to be filtered out from updates that include the aggregate route 1.0.0.0/8. To do this, you first configure an access list that specifies the routes to be filtered out (1.2.3.0/24). You then configure a route map *noadvert* to permit routes 1.2.3.0/24. You then configure an aggregate route 1.0.0.0/8 and specify the route map *noadvert* as a suppress option for outgoing updates.

WebUI

Network > Routing > Virtual Router > Access List (for trust-vr) > New: Enter the following, then click **OK**:

Access List ID: 1
Sequence No.: 777
IP/Netmask: 1.2.3.0/24
Action: Permit (select)

Network > Routing > Virtual Router > Route Map (for trust-vr) > New: Enter the following, then click **OK**:

Map Name: noadvert
Sequence No.: 2
Action: Permit (select)
Match Properties:
  Access List (select), 1 (select)

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Aggregate Address: Enter the following, then click **Apply**:

Aggregate State: Enable (select)
IP/Netmask: 1.0.0.0/8
Suppress Option: Route-Map (select), noadvert (select)

CLI

```
set vrouter trust-vr access-list 1 permit ip 1.2.3.0/24 777
set vrouter trust-vr route-map name noadvert permit 2
```

```
set vrouter trust-vr route-map noadvert 2 match ip 1
set vrouter trust protocol bgp aggregate 1.0.0.0/8 suppress-map noadvert
save
```

Selecting Routes for Path Attribute

When you configure an aggregated route, you can specify which routes should or should not be used to build the BGP AS-Path path attribute of the aggregated route. Use the **Advertise-Map** option in the aggregate route configuration to select the routes. You can use this option with the AS-Set option to select routes that are advertised with the AS-Set attribute.

In the following example, you configure an aggregate route 1.0.0.0/8 to be advertised with the AS-Set attribute. The advertised **AS-Set** attribute consists of all more-specific routes that fall into the prefix range 1.5.0.0/16, but not the routes that fall into the prefix range 1.5.6.0/24; you configure the prefix ranges to be included and excluded in the route map *advertset*.

WebUI

Network > Routing > Virtual Router > Access List (for trust-vr) > New: Enter the following, then click **OK**:

```
Access List ID: 3
Sequence No.: 888
IP/Netmask: 1.5.6.0/24
Action: Deny (select)
```

Network > Routing > Virtual Router > Access List (for trust-vr) > New: Enter the following, then click **OK**:

```
Access List ID: 3
Sequence No.: 999
IP/Netmask: 1.5.0.0/16
Action: Permit (select)
```

Network > Routing > Virtual Router > Route Map (for trust-vr) > New: Enter the following, then click **OK**:

```
Map Name: advertset
Sequence No.: 4
Action: Permit (select)
Match Properties:
   Access List (select), 3 (select)
```

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Aggregate Address: Enter the following, then click **Apply**:

```
Aggregate State: Enable (select)
IP/Netmask: 1.0.0.0/8
Advertise Map: advertset (select)
```

CLI

```
set vrouter trust-vr access-list 3 deny ip 1.5.6.0/24 888
set vrouter trust-vr access-list 3 permit ip 1.5.0.0/16 999
set vrouter trust-vr route-map name advertset permit 4
set vrouter trust-vr route-map advertset 4 match ip 3
set vrouter trust protocol bgp aggregate 1.0.0.0/8 advertise-map advertset
save
```

Changing Attributes of an Aggregated Route

When you configure an aggregated route, you can set the attributes of the aggregated route based upon a specified route map. In the following example, you configure an aggregated route 1.0.0.0/8 (IPv4) or 2ccc::/16 (IPv6) that is advertised with the metric 1111 in outgoing updates.

WebUI (IPv4)

Network > Routing > Virtual Router > Route Map (for trust-vr) > New: Enter the following, then click **OK**:

```
Map Name: aggmetric
Sequence No.: 5
Action: Permit (select)
Set Properties: (select)
Metric: 1111
```

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Aggregate Address: Enter the following, then click **Apply**:

```
Aggregate State: Enable (select)
IP/Netmask: 1.0.0.0/8
Attribute Map: aggmetric (select)
```

WebUI (IPv6)

Network > Routing > Virtual Router > Route Map (for trust-vr) > New: Enter the following, then click **OK**:

```
Map Name: aggmetricv6
Sequence No.: 5
Action: Permit (select)
Set Properties: (select)
Metric: 1111
```

Network > Routing > Virtual Routers > Edit (for trust-vr) > Edit BGP Instance > Aggregate Address: Enter the following, then click **Apply**:

```
IPv6 Aggregate State: Enable (select)
IP/Prefix: 2ccc::/16
Attribute Map: aggmetricv6 (select)
```

CLI (IPv4)

```
set vrouter trust-vr route-map name aggmetric permit 5
set vrouter trust-vr route-map aggmetric 5 metric 1111
set vrouter trust protocol bgp aggregate 1.0.0.0/8 attribute-map aggmetric
save
```

CLI (IPv6)

```
set vrouter trust-vr route-map ipv6 name aggmetricv6 permit 5
set vrouter trust-vr route-map aggmetricv6 5 metric 1111
set vrouter trust protocol bgp ipv6 aggregate 2ccc::/16 attribute-map aggmetricv6
save
```

Chapter 37

Policy-Based Routing

This chapter describes policy based routing (PBR). PBR provides a flexible routing mechanism for data forwarding over networks that rely on Application Layer support such as for antivirus (AV), deep inspection (DI), or Web filtering.

This chapter contains the following sections:

Policy Based Routing Overview

PBR enables you to implement policies that selectively cause packets to take different paths. PBR provides a routing mechanism for networks that rely on Application Layer support, such as antivirus (AV), deep inspection (DI), or antispam, Web filtering, and/or that require an automatic way to specific applications.

When a packet enters the security device, ScreenOS checks for PBR as the first part of the route-lookup process, and the PBR check is transparent to all non-PBR traffic. PBR is enabled at the interface level and configured within a virtual router context; but you can choose to bind PBR policies to an interface, a zone, a virtual router (VR), or a combination of interface, zone, or VRs.

You use the following three building blocks to create a PBR policy:

- Extended access lists
- Match groups
- Action groups

Extended Access-Lists

Extended access-lists list the match criteria you define for PBR policies. PBR match criteria determine the path of a particular data traffic flow. Match criteria include the following:

- Source IP address
- Destination IP address
- Source port
- Destination port
- Protocol, such as HTTP
- Quality of Service (QoS) priority (optional)

Match Groups

Match groups provide a way to organize (by group, name and priority) extended access lists. Match groups associate an extended access-list ID number with a unique match group name and a match-group ID number. This match-group ID number defines the order in which you want the security device to process the extended ACL lists. You can assign multiple extended access-lists to the same match-group.

Action Groups

Action groups specify the route that you want a packet to take. You specify the "action" for the route by defining the next interface, the next-hop, or both.

Each configured action entry is monitored for reachability as follows:

NOTE: Monitoring reachability does not refer to Layer 3 tracking or Layer 2 Address Resolution Protocol (ARP) lookups.

- **Next-Interface Only Reachability**

  If you associate the action entry with only a next-interface, link state determines reachability.

  If the next-interface is up, the action entry is reachable. Any interface including all the logical interfaces, such as tunnel, aggregate, or redundant, that are visible in the VR in which the policy resides are candidates for next-interface.

  For example, if you configure the action entry with a NULL interface, the action entry is reachable all the time. With a NULL interface as the next interface, PBR lookup always succeeds; so, ScreenOS stops the route lookup and discards the packet(s).

- **Next-Hop Only Reachability**

  If you associate the action group with a next-hop only, that next-hop must be reachable through a route entry in the destination routes routing table. The configured next-hop is reachable as long as a valid route exists in the destination routes routing table to resolve the next-hop.

- **Next-Interface** and **Next-Hop Reachability**

If you configure both next-interface and next-hop reachability, the configured next-hop must be reachable through the configured next-interface.

If the next-hop is reachable through the next-interface, the action entry is reachable. Any interface including all the logical interfaces, such as tunnel, aggregate, or redundant, that are visible in the VR in which the policy resides are candidates to be a next-interface.

If the next hop is reachable but the next interface is a NULL interface, ScreenOS drops the packet. If you configure the action entry with a NULL interface as the next interface and the next hop as a static route, ScreenOS passes the packet(s) to the static route.

At the time of configuration, you also assign a sequence number to specify the order in which you want the action group entry processed.

Route Lookup with PBR

When you enable PBR on an interface, ScreenOS checks all traffic sent to that interface for PBR. When a packet enters the security device, ScreenOS checks the in-interface for a PBR policy configuration. If PBR is enabled on that in-interface, the following actions are applied to the packet:

1. ScreenOS applies the PBR policy bound to the in-interface to the packet.
2. If no interface-level PBR policy exists, then ScreenOS applies the PBR policy bound to the zone associated with the in-interface to the packet.
3. If no zone-level PBR policy exists, then ScreenOS applies the PBR policy bound to the VR associated with the in-interface to the packet.

ScreenOS locates the match group and then processes the action group entries. The first reachable action entry from the action-group with a valid route is used to forward the packet. If no reachable route exists among the action entries, then a regular route lookup is performed.

If the action entry is reachable, ScreenOS performs a route lookup with the preferred interface as the next-interface (if specified) and the next-hop as the IP address (if specified) instead of using the destination IP. If a route matches the indicated next-interface and next-hop, ScreenOS forwards the packet. Otherwise, ScreenOS uses the destination IP address.

NOTE: For more information about route lookup, see *"Fundamentals" on page 15*.

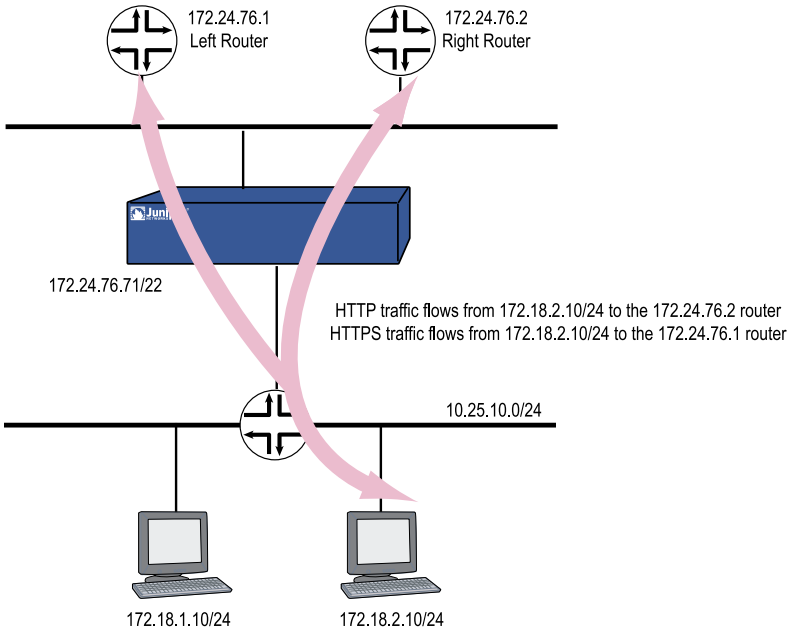
Configuring PBR

Figure 337 on page 1376 shows one way PBR differentiates service-traffic paths by sending HTTP traffic along one path and HTTPS traffic along another. Figure 337 on page 1376 shows two nodes, one at 172.18.1.10 and another at 172.18.2.10. When the security device receives HTTP traffic, ScreenOS routes the traffic through the

172.24.76.1 router; and when the security device receives HTTPS traffic, ScreenOS routes the traffic through the 172.24.76.2 router.

The opposite is true for the 172.18.2.10 node. HTTP traffic from the 172.18.2.10 node flows to the 172.24.76.2 router, and HTTPS traffic flows to the 172.24.76.1 router.

Figure 337: Routing HTTP and HTTPS Traffic with Policy Based Routing



Configuring an Extended Access List

You can configure an extended access list with the web user interface (WebUI) or the command line interface (CLI) from within a virtual router context. First, you configure the extended access list on the ingress virtual router (VR).

In this example on Figure 337 on page 1376, the ingress VR is the trust-vr. If you are using the CLI, you need to enter the virtual router context. This example requires two access lists: 10 and 20. The access sequence number is a number from 1 to 99. Entries 1 and 2 are required for each extended access list.

NOTE: Optionally, you can also add a type of service (TOS) number, which is a number from 1 to 255. A TOS number is not required in this example.

Access list 10 defines the source IP address as 172.18.1.10, the destination port as 80, and the protocol as TCP. The destination point for access list 10 defines the destination IP address as 172.18.2.10, the destination port as 443, and the protocol as TCP.

Access list 20 defines the source IP address as 172.18.2.10, the destination port as 443, and the protocol as TCP. The destination point for access list 10 defines the destination IP address as 172.18.1.10, the destination port as 80, and the protocol as TCP.

In the CLI after configuring the extended access list, you exit the virtual router context. The WebUI example only shows the creation of access list 10.

WebUI

Network > Routing > PBR > Extended ACL List: Select the virtual router from the drop-down list, then click **New** to view the Configuration page.

Enter the following information to create access list 10 entries:

Extended ACL ID: 10
Sequence No.: 1
Source IP Address/Netmask: 172.18.1.10/32
Destination Port: 80-80
Protocol: TCP

Click **OK**. ScreenOS returns you to a list of access lists.

Click **New** to configure a second entry for access list 10 and enter the following information:

Extended ACL ID : 10
Sequence No.: 2
Source IP Address/Netmask: 172.18.2.10/32
Destination Port: 443-443
Protocol: TCP

Click **OK**. ScreenOS returns you to a list of access lists.

CLI

```
set vrouter trust-vr
set access-list extended 10 src-ip 172.18.1.10/32 dest-port 80-80 protocol tcp
entry 1
set access-list extended 10 src-ip 172.18.2.10/32 dest-port 443-443 protocol tcp
 entry 2
set access-list extended 20 src-ip 172.18.2.10/32 dest-port 80-80 protocol tcp
entry 1
set access-list extended 20 src-ip 172.18.1.10/32 dest-port 443-443 protocol tcp
 entry 2
exit
```

Configuring a Match Group

You can configure a match group from within a virtual router context.

In the example on Figure 337 on page 1376, you need to configure two match-groups: Left Router and Right Router. You bind extended access list 10 to Left Router and

extended access list 20 to Right Router. A match group name is a unique identifier of no more than 31 alphanumeric characters.

The ingress VR is the trust-vr. If you are using the CLI, you need to enter the virtual router context. In the CLI after configuring the extended access list, you exit the virtual router context.

The WebUI example only shows the creation of a match group for Left Router.

WebUI

Network > Routing > PBR > Match Group > Select the correct virtual router from the drop-down list, then click **New** to view the Match Group Configuration page. Enter the following information to configure Left Router:

Match Group Name: left_router
Sequence No.: 1
Extended ACL: Select 10 from the drop-down list.

CLI

```
set vrouter trust-vr
set match-group name left_router
set match-group left ext-acl 10 match-entry 1
set match-group name right_router
set match-group right ext-acl 20 match-entry 1
exit
```

Configuring an Action Group

You can configure an action group within a virtual routing context.

In the example on Figure 337 on page 1376 two different action groups are possible: the security device can forward to traffic to the left router or the right router. For this reason, you need to configure two different action groups.

To configure these two action-groups, you perform the following tasks:

1. Enter the virtual routing context. In this example, the virtual router is the trust-vr.
2. Name the action-group with a meaningful, unique name. In this example, the names **action-right** and **action-left** are descriptive of the possible traffic flows.
3. Configure the action-group details. In this example, you set the next-hop address for each action-group and then assign a number to indicate the processing priority. In this example, the priority of each action-group is 1.

WebUI

Network > Routing > PBR > Action Group > Click **New** to view the Configuration page

CLI

```
set vrouter trust-vr
set action-group name action-right
set action-group action-right next-hop 172.24.76.2 action-entry 1
set action-group name action-left
set action-group action-left next-hop 172.24.76.1 action-entry 1
exit
```

Configuring a PBR Policy

You can configure a PBR policy from within a virtual router context.

Each PBR policy needs to have a unique name. In this example, the policy is named **redirect-policy**.

A PBR policy can contain a match group name and action group name. In this example, traffic can flow two different ways, so two different statements are required: **action-left** with sequence number 1 and **action-right** with sequence number 2. The policy statement with sequence number 1 is processed first.

WebUI

Network > Routing > PBR > Policy > Click **New** to view the Configuration page

CLI

```
set vrouter trust-vr
set pbr policy name redirect-policy
set pbr policy redirect-policy match-group left action-group action-left 1
set pbr policy redirect-policy match-group right action-group action-right 2
exit
```

Binding a PBR Policy

You can bind a PBR policy to an interface, a zone, or a virtual router from within a virtual router context.

Binding a PBR Policy to an Interface

You can bind the PBR policy **redirect-policy** to the ingress interface. In this example, the interface is the **trust** interface.

WebUI

Network > Routing > PBR > Policy Binding

CLI

```
set interface trust pbr redirect-policy
```

Binding a PBR Policy to a Zone

You can bind the PBR policy **redirect-policy** to a zone. In this example, the zone is the **Trust** zone.

WebUI

Network > Routing > PBR > Policy Binding

CLI

```
set zone trust pbr redirect-policy
```

Binding a PBR Policy to a Virtual Router

You can bind the PBR policy **redirect-policy** to a virtual router. In this example, the virtual router is the **trust-vr**.

WebUI

Network > Routing > PBR > Policy Binding

CLI

```
set vrouter trust-vr pbr redirect-policy
```

Viewing PBR Output

You can view PBR-related information with the WebUI or the CLI.

Viewing an Extended Access List

You can view the entire list of extended access lists from the WebUI or the CLI.

In the CLI you can specify to view one particular extended access list. In the second CLI example, the sample output shows that two extended access lists exist in the trust-vr, but the user indicated extended access list 2. As specified, ScreenOS returned two access-list entries, 10 and 20, for the second extended access list only.

WebUI

Network > Routing > PBR > Access List Ext

CLI 1

```
get vrouter trust-vr pbr access-list configuration
```

Sample output:

```
set access-list extended 1 src-ip 172.16.10.10/32 dest-ip 192.169.10.10/32
dest-port 80-80 protocol tcp entry 1
set access-list extended 1 src-port 200-300 entry 2
set access-list extended 2 dest-port 500-600 protocol udp entry 10
set access-list extended 2 dest-ip 50.50.50.0/24 protocol udp entry 20
```

CLI 2

get vrouter trust-vr pbr access-list 2

Sample output:

```
PBR access-list: 2 in vr: trust-vr, number of entries: 2
------------------------------------------------
PBR access-list entry: 10
------------------------
dest port range 500-600
protocols: udp
PBR access-list entry: 20
------------------------
dest ip-address 50.50.50.0/24
protocols: udp
```

Viewing a Match Group

You can view match group details from the WebUI or the CLI.

WebUI

Network > Routing > PBR > Match Group

CLI

get vrouter trust-vr pbr match-group config

Sample output:

```
set match-group name pbr1_mg
set match-group pbr1_mg ext-acl 1 match-entry 1
set match-group name pbr1_mg2
set match-group pbr1_mg2 ext-acl 2 match-entry 10
```

Viewing an Action Group

You can view action group details from the WebUI or the CLI.

WebUI

Network > Routing > PBR > Action Group

CLI 1

```
get vrouter trust-vr pbr action-group configuration
```

Sample output:

```
set action-group name pbr1_ag
set action-group pbr1_ag next-interface ethernet2 next-hop 10.10.10.2 action-entry
 1
set action-group name pbr1_ag2
set action-group pbr1_ag2 next-hop 30.30.30.30 action-entry 10
set action-group pbr1_ag2 next-interface ethernet3 action-entry 20
set action-group pbr1_ag2 next-interface ethernet3 next-hop 60.60.60.60 action-entry
 30
```

CLI 2

```
get vrouter trust-vr pbr match-group name pbr1_ag2
```

Sample output:

```
device-> get vr tr pbr action-group name pbr1_ag2
PBR action-group: pbr1_ag2 in vr: trust-vr number of entries: 3
------------------------------------------------
PBR action-group entry: 10
next-interface: N/A, next-hop: 30.30.30.30
------------------------
PBR action-group entry: 20
next-interface: ethernet3, next-hop: 0.0.0.0
------------------------
PBR action-group entry: 30
next-interface: ethernet3, next-hop: 60.60.60.60
------------------------
```

Viewing a PBR Policy Configuration

You can view PBR policy configuration details from the WebUI or the CLI. In the CLI you can choose to view the configuration or you can enter the policy name to view a single policy configuration.

WebUI

Network > Routing > PBR > Policy

CLI

```
get vrouter trust-vr pbr policy config
```

Sample output:

```
set pbr policy name pbr1_policy
set pbr policy pbr1_policy match-group pbr1_mg2 action-group pbr1_ag2 50
set pbr policy pbr1_policy match-group pbr1_mg action-group pbr1_ag 256
```

CLI

get vrouter trust-vr pbr policy name pbr1_policy

Sample output:

```
PBR policy: pbr1_policy in vr: trust-vr number of entries: 2
-----------------------------------------------
PBR policy entry: 50
match-group: pbr1_mg2, action-group: pbr1_ag2
------------------------
PBR policy entry: 256
match-group: pbr1_mg, action-group: pbr1_ag
------------------------
```

Viewing a Complete PBR Configuration

You can view a PBR configuration from the WebUI or the CLI.

WebUI

Network > Routing > PBR > Access List Ext
Network > Routing > PBR > Match Group
Network > Routing > PBR > Action Group
Network > Routing > PBR > Policy

CLI

get vrouter trust-vr pbr configuration

Sample output:

```
set access-list extended 1 src-ip 172.16.10.10/32 dest-ip 192.169.10.10/32
dest-port 80-80 protocol tcp entry 1
set access-list extended 1 src-port 200-300 entry 2
set access-list extended 2 dest-port 500-600 protocol udp entry 10
set access-list extended 2 dest-ip 50.50.50.0/24 protocol udp entry 20
set match-group name pbr1_mg
set match-group pbr1_mg ext-acl 1 match-entry 1
set match-group name pbr1_mg2
set match-group pbr1_mg2 ext-acl 2 match-entry 10
set action-group name pbr1_ag
set action-group pbr1_ag next-interface ethernet2 next-hop 10.10.10.2 action-entry
 1
set action-group name pbr1_ag2
set action-group pbr1_ag2 next-hop 30.30.30.30 action-entry 10
set action-group pbr1_ag2 next-interface ethernet3 action-entry 20
set action-group pbr1_ag2 next-interface ethernet3 next-hop 60.60.60.60
action-entry 30
set pbr policy name pbr1_policy
```

```
set pbr policy pbr1_policy match-group pbr1_mg2 action-group pbr1_ag2 50
set pbr policy pbr1_policy match-group pbr1_mg action-group pbr1_ag 256
```

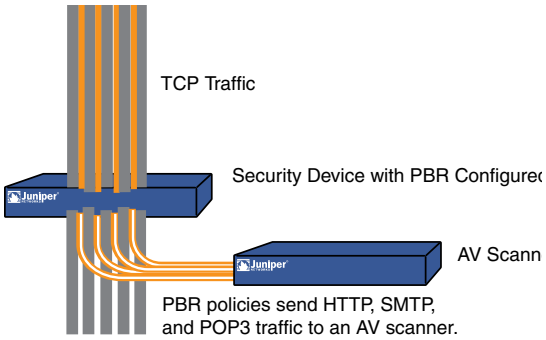
Advanced PBR Example

PBR allows you to define and offload only the types of traffic that ScreenOS needs to process. In processing specific types of traffic, such as traffic requiring antivirus (AV) scanning, the network does not get bottlenecked by scanning packet types that do not need to be scanned for viruses.

NOTE: You could also configure PBR to send traffic specific for antispam, deep inspection (DI), intrusion detection and prevention (IDP), Web filtering, or caching.

You can combine several types of Juniper Networks security devices to work together to provide services while keeping network processing speed fast and AV scanning manageable. Figure 338 on page 1384 shows a security device running PBR to segregate AV traffic from all other traffic (right).

Figure 338: Selective Routing by Traffic Type



For example, if you want use PBR to offload only HTTP, SMTP, and POP3 traffic for AV processing, at a minimum you need to use at least one security device with four available 10/100 interfaces to provide routing and one security device to provide the application (AV) support.

NOTE: If you have only three 10/100 interfaces available, you can place a switch between the two security devices and use VLAN tagging (802.1q) to set up the same paths for the traffic.

In the following example, you perform the following steps to set up the security device that provides the routing paths:

1. Configure routing.
2. Configure policy based routing.
3. Bind the PBR policies to the appropriate interfaces.

The next sections explain each of these steps. The examples show only CLI commands and output.

For information about configuring AV, see "Attack Detection and Defense Mechanisms" on page 431.

Routing

In this example, you need to create two custom zones:

- **av-dmz-1** for the trust-vr
- **av-dmz-2** for the untrust-vr

To set up the zones, enter the following commands:

```
set zone name av-dmz-1
set zone name av-dmz-2
```

Using the information shown in Table 98 on page 1385, you set up four 10/100 Ethernet interfaces.

Table 98: Interface Configuration for Routing

Interface Name	Zone	Virtual Router	IPv4 Address
E1	trust	trust-vr	10.251.10.0/24
E2	av-dmz-1	trust-vr	192.168.100.1/24
E3	av-dmz-2	untrust-vr	192.168.101.1/24
E4	untrust	untrust-vr	172.24.76.127/24

To set up the interfaces, enter the following commands:

```
set interface e1 zone trust vrouter trust-vr ip 10.251.10.0/24
set interface e2 zone av-dmz-1 vrouter trust-vr ip 192.168.100.1/24
set interface e3 zone av-dmz-2 vrouter untrust-vr ip 192.168.101.1/24
set interface e4 zone untrust vrouter untrust-vr ip 172.24.76.127/24
```

After setting up the zones, interfaces and routes, you need to perform the following two tasks:

1. Configure a static route from the untrust-vr to the trust-vr. Assign a gateway IP address of 10.251.10.0/24 and a preference value of 20 to the entry:

   ```
   set vrouter "untrust-vr"
   set route 10.251.10.0/24 vrouter "trust-vr" preference 20
   exit
   ```

2. Configure the NULL interface with a preference value greater than zero (0) from the Trust interface to the Untrust interface:

```
set vrouter "trust-vr"
set route 0.0.0.0/0 vrouter "untrust-vr" preference 20
exit
```

You can verify the changes with the **get route** command:

```
Routing Table:
IPv4 Dest-Routes for <untrust-vr> (6 entries)
------------------------------------------------------------------------------
   ID          IP-Prefix      Interface         Gateway   P Pref    Mtr     Vsys
------------------------------------------------------------------------------
*   6          0.0.0.0/0           eth4     172.24.76.1   C    0      1     Root
*   3     10.251.10.0/24            n/a        trust-vr   S   20      0     Root
*   4     172.24.76.0/22          eth4         0.0.0.0   C    0      0     Root
*   2   192.168.101.1/32          eth3         0.0.0.0   H    0      0     Root
*   5   172.24.76.127/32          eth4         0.0.0.0   H    0      0     Root
*   1   192.168.101.0/24          eth3         0.0.0.0   C    0      0     Root

IPv4 Dest-Routes for <trust-vr> (5 entries)
------------------------------------------------------------
   ID          IP-Prefix      Interface         Gateway   P Pref    Mtr     Vsys
------------------------------------------------------------
*   5          0.0.0.0/0            n/a      untrust-vr   S   20      0     Root
*   1     10.251.10.0/24          eth1         0.0.0.0   C    0      0     Root
*   4   192.168.100.1/32          eth2         0.0.0.0   H    0      0     Root
*   3   192.168.100.0/24          eth2         0.0.0.0   C    0      0     Root
*   2     10.251.10.1/32          eth1         0.0.0.0   H    0      0     Root
```

You are now ready to configure PBR.

PBR Elements

After you configure the interfaces and routes, you configure PBR. For PBR to work correctly, you must configure the following items for the trust-vr:

- Extended access list
- Match group
- Action group
- PBR policy

Extended Access Lists

For this example, you determine that you want to send HTTP (port 80), SMTP (port 110), and POP3 (port 25) traffic for AV processing. To send these three types of packets to a security device, you set up an extended access list in the trust-vr.

NOTE: You do not need to set up an extended access list for the return traffic because the security device performs a session lookup before a route lookup and then applies a PBR policy as necessary. Return traffic has an existing session.

When any client in the 10.251.10.0/24 subnet initiates traffic that uses TCP to port 80, 110, or 25, you want ScreenOS to match that traffic to extended access list criteria and to perform the action associated with the access list. The action forces ScreenOS to route the traffic as you indicate and not like other traffic. Each access list needs three entries, one for each kind of TCP traffic that you are targeting.

To configure the extended access list for the trust-vr, enter the following commands:

```
set vrouter "trust-vr"
set access-list extended 10 src-ip 10.251.10.0/24 dest-port 80-80 protocol tcp
entry 1
set access-list extended 10 src-ip 10.251.10.0/24 dest-port 110-110 protocol
tcp entry 2
set access-list extended 10 src-ip 10.251.10.0/24 dest-port 25-25 protocol tcp
entry 3
exit
```

Match Groups

A match group associates an extended access list with a meaningful name that gets referenced in the PBR policy. You first enter a virtual router context, then create a match group, and finally add an entry that associates the newly created match group name with an access list and entry number.

To create match groups in the trust-vr, enter the following commands:

```
set vrouter trust-vr
set match-group name av-match-trust-vr
set match-group av-match-trust-vr ext-acl 10 match-entry 1
exit
```

Action Group

Next, you create an action-group, which indicates where to send the packet. For this example, you create an action group for the trust-vr with the action set to send the traffic to the next hop.

CAUTION: If the action is to send traffic to the next interface, the link-state change will activate/deactivate the routing policy.

With next hop, the action resolves with Address Resolution Protocol (ARP).

For the trust-vr, you redirect traffic with the next hop statement through 192.168.100.254 by entering the following commands:

```
set vrouter trust-vr
   set action-group name av-action-redirect-trust-vr set action-group
av-action-redirect-trust-vr next-hop 192.168.100.254 action-entry 1
exit
```

PBR Policies

Next, you define the PBR policy, which requires the following elements:

- PBR policy name
- Match group name
- Action group name

To configure the PBR policy, enter the following commands:

```
set vrouter trust-vr
set pbr policy name av-redirect-policy-trust-vr
set pbr policy av-redirect-policy-trust-vr match-group av-match-trust-vr action-group
 av-action-redirect-trust-vr 1
exit
```

Interface Binding

Finally, you bind the PBR policy to the ingress interface, e1.

To bind the PBR policy to its ingress interface, enter the following commands:

```
set interface e1 pbr av-redirect-policy-trust-vr
```

Advanced PBR with High Availability and Scalability

Using the previous PBR example as a foundation, you can add resilience to your network with high availability (HA) and/or scalability.

Resilient PBR Solution

A robust PBR solution might include the following device configurations:

- Two security devices that provide networking
- Two other security devices that provide AV scanning

Each pair of devices runs NetScreen Redundancy Protocol (NSRP) in an Active/Passive configuration to provide failover protection. For the two security devices that are performing routing, one device takes over the routing function if a hardware failure occurs. In the case of the pair that is providing the AV scanning, if a failure occurs in one of the devices, the other device takes over the scanning function.

NOTE: For more information, see “Virtual Private Networks” on page 705.

Scalable PBR Solution

PBR solutions scale well. If you need more capacity, you can add more security devices. By dividing the /24 subnet into two /25 subnets, you can configure one extended access list for the lower /25 subnet and another extended access list for the higher /25 subnet, then add two security devices to provide scanning services in the DMZ.

You can also implement load balancing if you create an active/active NSRP configuration. One device could process traffic from the lower /25 subnet, and the other device could process traffic from the higher /25 subnet. Each device backs up the other.

Chapter 38

Multicast Routing

This chapter introduces basic multicast routing concepts. It contains the following sections:

Overview

Enterprises use multicast routing to transmit traffic, such as data or video streams, from one source to a group of receivers simultaneously. Any host can be a source, and the receivers can be anywhere on the Internet.

IP multicast routing provides an efficient method for forwarding traffic to multiple hosts because multicast-enabled routers transmit multicast traffic only to hosts that want to receive the traffic. Hosts must signal their interest in receiving multicast data, and they must join a multicast group in order to receive the data. Multicast-enabled routers forward multicast traffic only to receivers interested in receiving the traffic.

Multicast routing environments require the following elements to forward multicast information:

- A mechanism between hosts and routers to communicate multicast group membership information. Security devices support Internet Group Management Protocol (IGMP) versions 1, 2, and 3. Routers and hosts use IGMP to transmit membership information only, not to forward or route multicast traffic. (For information about IGMP, see "Internet Group Management Protocol" on page 1399.)
- A multicast routing protocol to populate the multicast route table and forward data to hosts throughout the network. Juniper Networks security devices support Protocol Independent Multicast-Sparse-Mode (PIM-SM) and Protocol Independent Multicast-Source-Specific Mode (PIM-SSM). (For information about PIM-SM and PIM-SSM, see "Protocol Independent Multicast" on page 1425.)

  Alternatively, you can use the IGMP Proxy feature to forward multicast traffic without the CPU overhead of running a multicast routing protocol. (For more information, see "IGMP Proxy" on page 1407.)

The following sections introduce basic concepts used in multicast routing.

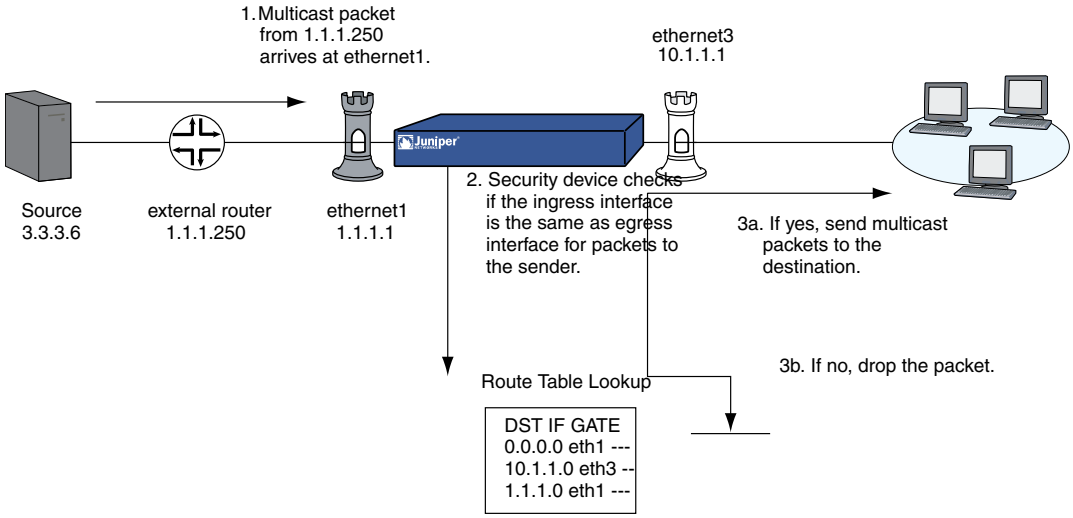
Multicast Addresses

When a source sends multicast traffic, the destination address is a multicast group address. Multicast group addresses are Class D addresses from 224.0.0.0 to 239.255.255.255.

Reverse Path Forwarding

When a multicast router receives multicast packets, it uses a process called reverse path forwarding (RPF) to check the validity of the packets. Before creating a multicast route, the router performs a route lookup on the unicast route table to check if the interface on which it received the packet (ingress interface) is the same interface it must use to send packets back to the sender. If it is, the router creates the multicast route entry and forwards the packet to the next hop router. If it is not, the router drops the packet. Multicast routers do not perform this RPF check for static routes. Figure 339 on page 1392 shows the security device and the multicast packet processing flow.

Figure 339: Reverse Path Forwarding



Multicast Routing on Security Devices

Juniper Networks security devices have two predefined virtual routers (VRs): a trust-vr and an untrust-vr. Each virtual router is a separate routing component with its own unicast and multicast route tables. (For information about unicast route tables, see "Static Routing" on page 1221.) When the security device receives an incoming multicast packet, it does a route lookup using the routes in the multicast route table.

Multicast Routing Table

The multicast route table is populated by multicast static routes or routes learned through a multicast routing protocol. The security device uses the information from

the multicast route table to forward multicast traffic. Security devices maintain a multicast routing table for each routing protocol in a virtual router.

The multicast routing table contains information specific to the routing protocol plus the following information:

- Each entry starts with the forwarding state. The forwarding state can be in one of the following formats: (*, G) or (S, G). The (*, G) format is called a "star comma G" entry where the * indicates any source and G is a specific multicast group address. The (S, G) format is called an "S comma G" entry, where S is the source IP address and G is the multicast group address.
- The upstream and downstream interfaces.
- The reverse path forwarding (RPF) neighbor.

Following is an example of a PIM-SM multicast routing table in the trust-vr virtual router:

```
trust-vr - PIM-SM routing table
------------------------------------------------------------------------------
Register - R, Connected members - C, Pruned - P, Pending SPT Alert - G
Forward - F, Null - N, Negative Cache - E, Local Receivers - L
SPT - T, Proxy-Register - X, Imported - I, SGRpt state - Y, SSM Range Group - S
Turnaround Router - K
------------------------------------------------------------------------------
Total PIM-SM mroutes: 2
(*, 236.1.1.1)  RP 20.20.20.10         00:06:24/-         Flags: LF
  Zone            : Untrust
  Upstream        : ethernet1/2         State          : Joined
  RPF Neighbor    : local               Expires        : -
  Downstream      :
  ethernet1/2  00:06:24/00:02:57  Join         0.0.0.0          FC
(20.20.20.200/24, 236.1.1.1)           00:06:24/00:00:36  Flags: TXLF  Register Prune
  Zone            : Untrust
  Proxy register  : (10.10.10.1, 238.1.1.1) of zone Trust
  Upstream        : ethernet1/1         State          : Joined
  RPF Neighbor    : local               Expires        : -
  Downstream      :
  ethernet1/2  00:06:24/-         Join         236.1.1.1          20.20.20.200 FC
```

Configuring a Static Multicast Route

You can define a static multicast route from a source to a multicast group (S, G) or wildcard either the source or multicast group, or both. Static multicast routes are typically used to support multicast data forwarding from the hosts on interfaces in IGMP router proxy mode to the routers upstream on the interfaces in IGMP host mode. (For more information, see "IGMP Proxy" on page 1407.) You can also use static multicast routes to support inter-domain multicast forwarding. You can create a static route for an (S, G) pair with any input and output interface. You can also create a static route and wildcard either the source or multicast group, or both by entering 0.0.0.0. When you configure a static route, you can also specify the original multicast group address and a different multicast group address on the outgoing interface.

In this example, you configure a static multicast route from a source with IP address 20.20.20.200 to the multicast group 238.1.1.1. Configure the security device to translate the multicast group from 238.1.1.1 to 238.2.2.1 on the outgoing interface.

WebUI

Network > Routing > MCast Routing > New: Enter the following, then click **OK**:

Source IP: 20.20.20.200
MGroup: 238.1.1.1
Incoming Interface: ethernet1(select)
Outgoing Interface: ethernet3(select)
Translated MGroup: 238.2.2.1

CLI

```
set vrouter trust-vr mroute mgroup 238.1.1.1 source 20.20.20.200 iif ethernet1 oif
 ethernet3 out-group 238.2.2.1
save
```

Access Lists

An access list is a sequential list of statements against which a route is compared. Each statement specifies the IP address/netmask of a network prefix and the forwarding status (permit or deny the route). In multicast routing, a statement can also contain a multicast group address. In multicast routing, you create access lists to permit multicast traffic for specified multicast groups or hosts. Therefore, the action or forwarding status is always Permit. You cannot create access lists to deny certain groups or hosts. (For additional information about access lists, see "Configuring an Access List" on page 1263.)

Configuring Generic Routing Encapsulation on Tunnel Interfaces

Encapsulating multicast packets in unicast packets is a common method for transmitting multicast packets across a non-multicast-aware network and through IPsec tunnels. Generic Routing Encapsulation (GRE) version 1 is a mechanism that encapsulates any type of packet within IPv4 unicast packets. Juniper Networks security devices support GREv1 for encapsulating IP packets in IPv4 unicast packets. For additional information about GRE, refer to RFC 1701, *Generic Routing Encapsulation (GRE).*

On security devices, you enable GRE encapsulation on tunnel interfaces.

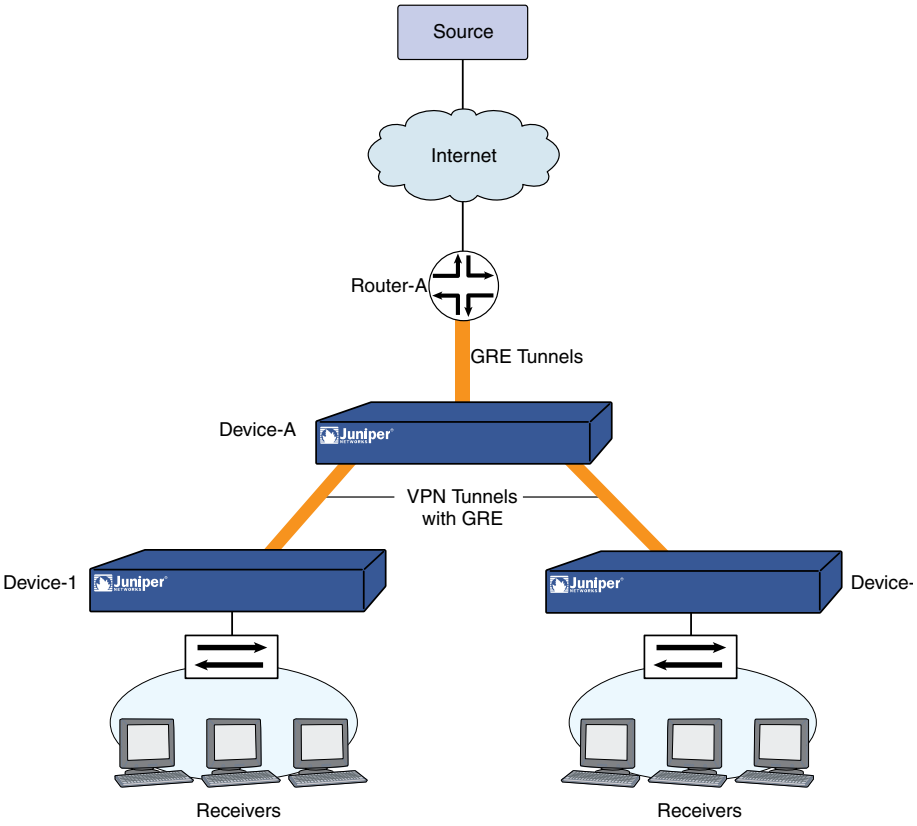
NOTE: You can enable GRE on a tunnel interface that is bound to a loopback interface as long as the loopback interface is on the same zone as the outgoing interface. For information about loopback interfaces, see "Loopback Interfaces" on page 75.

You must enable GRE when you transmit multicast packets through an IPsec VPN tunnel between a Juniper Networks security device and a third-party device or router.

Security devices have platform-specific limitations on the number of outgoing interfaces through which they can transmit multicast packets. In large hub-and-spoke VPN environments where the security device is the hub, you can avoid this limitation by creating a GRE tunnel between the router upstream of the hub-site security device to security devices at the spokes.

In Figure 340 on page 1395, Router-A is upstream of Device-A. Router-A has two GRE tunnels which terminate at Device-1 and Device-2. Device-A is connected to Device-1 and Device-2 through VPN tunnels. Before Router-A transmits multicast packets, it first encapsulates them in IPv4 unicast packets. Device-A receives these packets as unicast packets and sends them through to Device-1 and Device-2.

Figure 340: GRE on Tunnel Interfaces



In this example, you configure the tunnel interface on Device-1. You perform the following steps:

1. Create the tunnel.1 interface and bind it to ethernet3 and to the Untrust zone on the trust-vr.
2. Enable GRE encapsulation on tunnel.1.
3. Specify the local and remote endpoints of the GRE tunnel.

This example shows the GRE configuration for the security device only. (For information about VPNs, see *"Virtual Private Networks" on page 705*.)

WebUI

Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet3 (trust-vr)

Network > Interfaces > Tunnel (tunnel.1): Enter the following, then click **Apply**:

Encap: GRE (select)
Local Interface: ethernet3
Destination IP: 3.3.3.1

CLI

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.1 tunnel encap gre
set interface tunnel.1 tunnel local-if ethernet3 dst-ip 3.3.3.1
save
```

Multicast Policies

By default, Juniper Networks security devices do not permit multicast control traffic, such as IGMP or PIM messages, to cross security devices. To permit multicast control traffic between zones, you must configure a multicast policy that specifies the following:

- **Source**—The zone from which traffic initiates
- **Destination**—The zone to which traffic is sent
- **Multicast group**—The multicast group for which you want the security device to permit multicast control traffic. You can specify one of the following:
  - The multicast group IP address
  - An access list that defines the multicast group(s) that hosts can join
  - The keyword **any**, to allow multicast control traffic for any multicast group
- **Multicast control traffic**—The type of multicast control message: IGMP messages or PIM messages. (For information about IGMP, see "Internet Group Management Protocol" on page 1399. For information about PIM, see "Protocol Independent Multicast" on page 1425.)

In addition, you can specify the following:

- **Translated multicast address**—The security device can translate a multicast group address in an internal zone to a different address on the egress interface. To translate a group address, you must specify both the original multicast address and the translated multicast group address in the multicast policy.

- **Bi-directional**—You can create a bidirectional policy to apply it to both directions of traffic.

NOTE: Multicast policies control the flow of multicast control traffic only. To allow data traffic (both unicast and multicast) to pass between zones, you must configure firewall policies. (For information about policies, see "Virtual Private Networks" on page 705.)

You do not sequence multicast policies, as you would firewall policies. Thus, the latest multicast policy does not overwrite an earlier one, should there be a conflict. Instead, the security device selects the longest match to resolve any conflict, as used by other routing protocols. When it finds a smaller subnet to match the request, it uses that policy.

NOTE: For an example of how to configure a multicast policy for IGMP messages, see "Creating a Multicast Group Policy for IGMP" on page 1411. For an example of how to configure a multicast policy for PIM messages, see "Defining a Multicast Group Policy for PIM-SM" on page 1435.

Chapter 39

Internet Group Management Protocol

This chapter describes the Internet Group Management Protocol (IGMP) multicast protocol on Juniper Networks security devices. It contains the following sections:

Overview

The Internet Group Management Protocol (IGMP) multicast protocol is used between hosts and routers to establish and maintain multicast group memberships in a network. security devices support the following versions of IGMP:

- IGMPv1, as defined in RFC 1112, *Host Extensions for IP Multicasting*, defines the basic operations for multicast group memberships.
- IGMPv2, as defined in RFC 2236, *Internet Group Management Protocol*, Version 2, expands on the functionality of IGMPv1.
- IGMPv3, as defined in RFC 3376, *Internet Group Management Protocol, Version 3*, adds support for source filtering. Hosts running IGMPv3 indicate which multicast groups they want to join and the sources from which they expect to receive multicast traffic. IGMPv3 is required when you run Protocol Independent Multicast in Source-Specific Multicast (PIM-SSM) mode. (For more information, see "PIM-SSM" on page 1431.)

IGMP provides a mechanism for hosts and routers to maintain multicast group memberships. Multicast routing protocols, such as PIM, then process the membership information from IGMP, create entries in the multicast routing table and forward multicast traffic to hosts throughout the network.

The following sections explain the different types of IGMP messages that hosts and routers exchange to maintain group membership information throughout the network. Hosts and routers running newer versions of IGMP can operate with those running older IGMP versions.

Hosts

Hosts send IGMP messages to join multicast groups and maintain their memberships in those groups. Routers learn which hosts are members of multicast groups by

listening to these IGMP messages on their local networks. Table 99 on page 1400 lists the IGMP messages that hosts send and the destination of the messages.

Table 99: IGMP Host Messages

IGMP Version	IGMP Message	Destination
IGMPv1 and v2	A host sends a membership report when it first joins a multicast group and periodically, once it is a member of the group. The membership report indicates which multicast group the host wants to join.	IP address of the multicast group the host wants to join
IGMPv3	A host sends a membership report when it first joins a multicast group and periodically, once it is a member of the group. The membership report contains the multicast group address, the filter-mode, which is either include or exclude, and a list of sources. If the filter-mode is include, then packets from the addresses in the source list are accepted. If the filter mode is exclude, then packets from sources other than those in the source list are accepted.	224.0.0.22
IGMPv2	A host sends a Leave Group message when it wants to leave the multicast group and stop receiving data for that group.	"all routers group" (224.0.0.2)

Multicast Routers

Routers use IGMP to learn which multicast groups have members on their local network. Each network selects a designated router, called the querier. There is usually one querier for each network. The querier sends IGMP messages to all hosts in the network to solicit group membership information. When the hosts respond with their membership reports, the routers take the information from these messages and update their list of group memberships on a per-interface basis. IGMPv3 routers maintain a list which includes the multicast group address, filter-mode (either include or exclude), and the source list.

NOTE: With IGMPv1, each multicast routing protocol determines the querier for a network. With IGMPv2 and v3, the router interface with the lowest IP address in the network is the querier.

Table 100 on page 1400 describes the messages that a querier sends and destinations.

Table 100: IGMP Querier Messages

IGMP Version	IGMP Message	Destination
IGMPv1, v2 and v3	The querier periodically sends general queries to solicit group membership information.	"all hosts" group (224.0.0.1)
IGMPv2 and v3	The querier sends a group-specific query when it receives an IGMPv2 Leave Group message or an IGMPv3 membership report that indicates a change in group membership. If the querier does not receive a response within a specified interval, then it assumes there are no more members for that group on its local network and stops forwarding multicast traffic for that group.	The multicast group that the host is leaving

Table 100: IGMP Querier Messages *(continued)*

IGMPv3	The querier sends a group-and-source-specific query to verify whether there are any receivers for that particular group and source.	The multicast group that the host is leaving

IGMP on Security Devices

On some routers, IGMP is automatically enabled when you enable a multicast routing protocol. On Juniper Networks security devices, you must explicitly enable IGMP and a multicast routing protocol.

Enabling and Disabling IGMP on Interfaces

IGMP is disabled by default on all interfaces. You must enable IGMP in router mode on all interfaces that are connected to hosts. When in router mode, the security device runs IGMPv2 by default. You can change the default and run IGMPv1, IGMPv2 and v3, or only IGMPv3.

Enabling IGMP on an Interface

In this example, you enable IGMP in router mode on the ethernet1 interface, which is connected to a host.

WebUI

Network > Interfaces > Edit (for ethernet1) > IGMP: Enter the following, then click **Apply**:

IGMP Mode: Router (select)
Protocol IGMP: Enable (select)

CLI

```
set interface ethernet1 protocol igmp router
set interface ethernet1 protocol igmp enable
save
```

Disabling IGMP on an Interface

In this example, you disable IGMP on the ethernet1 interface. The security device maintains the IGMP configuration, but disables it.

WebUI

Network > Interfaces > Edit (for ethernet1) > IGMP: Clear **Protocol IGMP Enable**, then click **Apply**.

CLI

```
unset interface ethernet1 protocol igmp enable
save
```

To delete the IGMP configuration, enter the **unset interface** *interface* **protocol igmp router** command.

Configuring an Access List for Accepted Groups

There are some security issues you must consider when running IGMP. Malicious users can forge IGMP queries, membership reports, and leave messages. On security devices, you can restrict multicast traffic to known hosts and multicast groups only. In addition, you can also specify the allowed queriers in your network. You set these restrictions by creating access lists and then applying them to an interface.

An access list is a sequential list of statements that specifies an IP address and a forwarding status (permit or deny). In IGMP, access lists must always have a forwarding status of **permit** and must specify one of the following:

- Multicast groups that hosts can join
- Hosts from which the IGMP router interface can receive IGMP messages
- Queriers from which the IGMP router interface can receive IGMP messages

After you create an access list, you apply it to an interface. Once you apply an access list to an interface, that interface accepts traffic only from those specified in the access list. Therefore, to deny traffic from a particular multicast group, host or querier, simply exclude it from the access list. (For additional information about access lists, see “Configuring an Access List” on page 1263.)

In this example, you create an access list on the trust-vr. The access list specifies the following:

- Access list ID is 1.
- Permit traffic for multicast group 224.4.4.1/32.
- Sequence Number of this statement is 1.

After you create the access list, allow the hosts on ethernet1 to join the multicast group specified in the access list.

WebUI

Network > Routing > Virtual Routers > Access List: > New (for trust-vr): Enter the following, then click **OK**:

```
Access List ID: 1
Sequence No: 1
IP/Netmask: 224.4.4.1/32
Action: Permit (select)
```

Network > Interfaces > Edit (for ethernet1) > IGMP: Enter the following, then click **OK**:

Accept Group's Access List ID: 1

CLI

```
set vrouter trust-vr access-list 1 permit ip 224.4.4.1/32 1
set interface ethernet1 protocol igmp accept groups 1
save
```

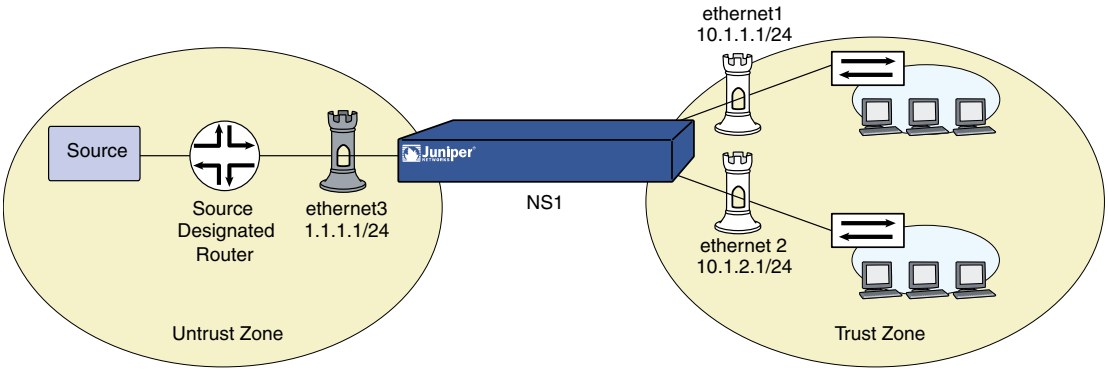
Configuring IGMP

To run IGMP on a Juniper Networks security device, you simply enable it in router mode on the interfaces that are directly connected to hosts. To ensure the security of your network, use access lists to limit multicast traffic to known multicast groups, hosts, and routers.

In Figure 341 on page 1403, the hosts in the Trust zone protected by the security device NS1 are potential receivers of the multicast stream from the source in the Untrust zone. The interfaces ethernet1 and ethernet2 are connected to the hosts. The multicast source is transmitting data to the multicast group 224.4.4.1. Perform the following steps to configure IGMP on the interfaces that are connected to the hosts:

1. Assign IP addresses to the interfaces and bind them to zones.
2. Create an access list that specifies the multicast group 224.4.4.1/32.
3. Enable IGMP in router mode on ethernet1 and ethernet2.
4. Restrict the interfaces (ethernet1 and ethernet2) to receiving IGMP messages for the multicast group 224.4.4.1/32.

Figure 341: IGMP Configuration Example



WebUI

1. **Zones and Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

Zone Name: Trust
IP Address/Netmask: 10.1.1.1/24

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: Trust
IP Address/Netmask: 10.1.2.1/24

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
IP Address/Netmask: 1.1.1.1/24

2. **Access List**

Network > Routing > Virtual Routers > Access List: > New (for trust-vr): Enter the following, then click **OK**:

Access List ID: 1
Sequence No: 1
IP/Netmask: 224.4.4.1/32
Action: Permit

3. **IGMP**

Network > Interfaces > Edit (for ethernet1) > IGMP: Enter the following, then click **Apply**:

IGMP Mode: Router (select)
Protocol IGMP: Enable (select)
Accept Group's Access List ID: 1

Network > Interfaces > Edit (for ethernet2) > IGMP: Enter the following, then click **Apply**:

IGMP Mode: Router (select)
Protocol IGMP: Enable (select)
Accept Group's Access List ID: 1

CLI

1. **Zones and Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet2 zone trust
set interface ethernet2 ip 10.2.1.1/24
```

2. **Access List**

```
set vrouter trust access-list 1 permit ip 224.4.4.1/32 1
```

3. **IGMP**

```
set interface ethernet1 protocol igmp router
set interface ethernet1 protocol igmp accept groups 1
set interface ethernet1 protocol igmp enable
set interface ethernet2 protocol igmp router
set interface ethernet2 protocol igmp accept groups 1
set interface ethernet2 protocol igmp enable
save
```

After you configure IGMP on ethernet1 and ethernet2, you must configure a multicast routing protocol, such as PIM, to forward multicast traffic. (For information about PIM, see "Protocol Independent Multicast" on page 1425.)

Verifying an IGMP Configuration

To verify connectivity and ensure that IGMP is running properly, there are a number of **exec** and **get** commands that you can use.

- To send either general queries or group-specific queries on a particular interface, use the **exec igmp interface** *interface* **query** command. For example, to send a general query from ethernet2, enter the following command:

  exec igmp interface ethernet2 query

  To send a group-specific query from ethernet2 to the multicast group 224.4.4.1, enter the following command:

  exec igmp interface ethernet2 query 224.4.4.1

- To send a membership report on a particular interface, use the **exec igmp interface** *interface* **report** command. For example, to send a membership report from ethernet2, enter the following command:

  exec igmp interface ethernet2 report 224.4.4.1

You can review the IGMP parameters of an interface by entering the following command:

```
device->  get igmp interface
Interface trust support IGMP version 2 router. It is enabled.
IGMP proxy is disabled.
Querier IP is 10.1.1.90, it has up 23 seconds. I am the querier.
There are 0 multicast groups active.
    Inbound Router access list number: not set
    Inbound Host access list number: not set
    Inbound Group access list number: not set
    query-interval: 125 seconds
    query-max-response-time 10 seconds
    leave-interval 1 seconds
    last-member-query-interval 1 seconds
```

This output lists the following information:

- IGMP version (2)
- Querier status (I am the querier.)

- Set and unset parameters

To display information about multicast groups, enter the following CLI command:

```
device->  get igmp group
total groups matched: 1
multicast group  interface    last reporter   expire ver
*224.4.4.1       trust        0.0.0.0         ------ v2
```

IGMP Operational Parameters

When you enable IGMP in router mode on an interface, the interface starts up as a querier. As the querier, the interface uses certain defaults which you can change. When you set parameters on this level, it affects only the interface that you specify. Table 101 on page 1406 lists the IGMP querier interface parameters and their defaults.

Table 101: IGMP Querier Interface Parameters and Default Values

IGMP Interface Parameters	Description	Default Value
General query interval	The interval at which the querier interface sends general queries to the "all hosts" group (224.0.0.1).	125 seconds
Maximum response time	The maximum time between a general query and a response from the host.	10 seconds
Last Member Query Interval	The interval at which the interface sends a Group-Specific query. If it does not receive a response after the second Group-Specific query, then it assumes there are no more members for that group on its local network.	1 second

By default, an IGMPv2/v3-enabled router accepts only IGMP packets with a router-alert IP option, and drops packets that do not have this option. IGMPv1 packets do not have this option and consequently, a security device running IGMPv2/v3 drops IGMPv1 packets by default. You can configure the security device to stop checking IGMP packets for the router-alert IP option and accept all IGMP packets, allowing backward compatibility with IGMPv1 routers. For example, to allow the ethernet1 interface to accept all IGMP packets:

WebUI

Network > Interfaces > Edit (for ethernet1) > IGMP: Select the following, then click **OK**:

Packet Without Router Alert Option: Permit (select)

CLI

```
set interface ethernet1 protocol igmp no-check-router-alert
save
```

IGMP Proxy

Routers listen for and send IGMP messages to their connected hosts only; they do not forward IGMP messages beyond their local network. You can allow interfaces on a Juniper Networks security device to forward IGMP messages one hop beyond its local network by enabling IGMP proxy. IGMP proxy enables interfaces to forward IGMP messages upstream toward the source without the CPU overhead of a multicast routing protocol.

When you run IGMP proxy on a security device, interfaces connected to hosts function as routers and those connected to upstream routers function as hosts. The host and router interfaces are typically in different zones. To allow IGMP messages to pass between zones, you must configure a multicast policy. Then, to allow multicast data traffic to pass between zones, you must also configure a firewall policy.

On devices that support multiple virtual systems, you must configure one interface in the root virtual system (vsys) and the other interface in a separate vsys. Then, create a multicast policy to allow multicast control traffic between the two virtual systems. (For information about virtual systems, see “Virtual Private Networks” on page 705.)

As the interfaces forward IGMP membership information, they create entries in the multicast route table of the virtual router to which the interfaces are bound, building a multicast distribution tree from the receivers to the source. The following sections describe how the IGMP host and router interfaces forward IGMP membership information upstream toward the source, and how they forward multicast data downstream from the source to the receiver.

Membership Reports Upstream to the Source

When a host connected to a router interface on a security device joins a multicast group, it sends a membership report to the multicast group. When the router interface receives the membership report from the attached host, it checks if it has an entry for the multicast group. The security device then takes one of the following actions:

- If the router interface has an entry for the multicast group, it ignores the membership report.
- If the router interface does not have an entry for the multicast group, it checks if there is a multicast policy for the group that specifies to which zone(s) the router interface should send the report.
  - If there is no multicast policy for the group, the router interface does not forward the report.
  - If there is a multicast policy for the group, the router interface creates an entry for the multicast group and forwards the membership report to the proxy host interface in the zone specified in the multicast policy.

When a proxy host interface receives the membership report, it checks if it has a (*, G) entry for that multicast group.

- If it has a (*, G) entry for the group, the host interface adds the router interface to the list of egress interfaces for that entry.
- If it does not have a (*, G) entry for that group, it creates such an entry; the ingress interface is the proxy host interface and the egress interface is the router interface. Then, the proxy host interface forwards the report to its upstream router.

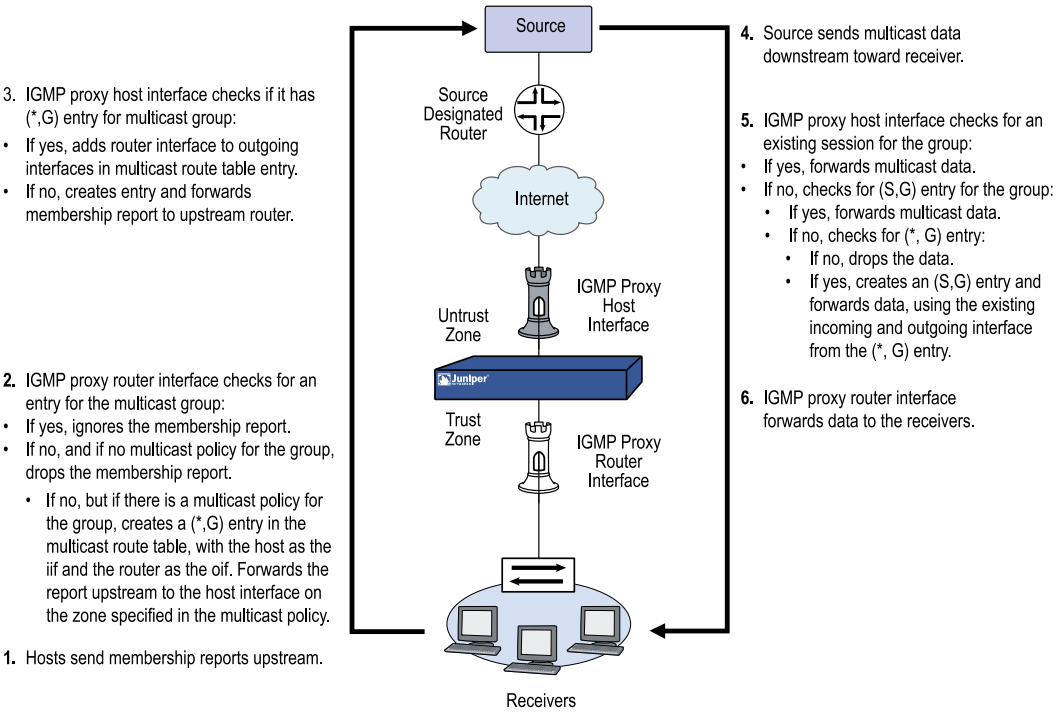
Multicast Data Downstream to Receivers

When the host interface on the security device receives multicast data for a multicast group, it checks if there is an existing session for that group.

- If there is a session for the group, the interface forwards the multicast data based on the session information.
- If there is no session for the group, the interface checks if the group has an (S, G) entry in the multicast route table.
  - If there is an (S, G) entry, the interface forwards the multicast data accordingly.
  - If there is no (S, G) entry, the interface checks if there is a (*, G) entry for the group.
  - If there is no (*, G) entry for the group, the interface drops the packet.
  - If there is a (*, G) entry for the group, the interface creates an (S, G) entry. When the interface receives subsequent multicast packets for that group, it forwards the traffic to the router interface (the egress interface), which in turn forwards the traffic to its connected host.

Figure 342 on page 1409 shows an example of an IGMP proxy host configuration.

Figure 342: IGMP Proxy Host Configuration



Configuring IGMP Proxy

This section describes the basic steps required to configure IGMP proxy on a Juniper Networks security device:

1. Enable IGMP in host mode on upstream interfaces. IGMP proxy is enabled by default on host interfaces.
2. Enable IGMP in router mode on downstream interfaces.
3. Enable IGMP proxy on router interfaces.
4. Configure a multicast policy that allows multicast control traffic to pass between zones.
5. Configure a policy to pass data traffic between zones.

Configuring IGMP Proxy on an Interface

When you run IGMP proxy on a security device, you configure the downstream interface in router mode and the upstream interface in host mode. (Note that an interface can either be in host mode or router mode, not both.) Additionally, for a router interface to forward multicast traffic, it must be the querier in the local network. To allow a non-querier interface to forward multicast traffic, you must specify the keyword **always** when you enable IGMP on the interface.

By default, an IGMP interface accepts IGMP messages from its own subnet only. It ignores IGMP messages from external sources. You must enable the security device to accept IGMP messages from sources in other subnets when you run IGMP proxy.

In this example, the interface ethernet1 has an IP address of 10.1.2.1/24 and is connected to the upstream router. You configure the following on ethernet1:

- Enable IGMP in host mode.
- Allow it to accept IGMP messages from all sources, regardless of subnet.

The interface ethernet3 has an IP address of 10.1.1.1/24 and is connected to the hosts. You configure the following on ethernet3:

- Enable IGMP in router mode.
- Allow it to forward multicast traffic even if it is a non-querier.
- Allow it to accept IGMP messages from sources on other subnets.

WebUI

1. **Zones and Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

   Zone Name: Trust
   IP Address/Netmask: 10.1.2.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **Apply**:

   Zone Name: Trust
   IP Address/Netmask: 10.1.1.1/24

2. **IGMP**

   Network > Interfaces > Edit (for ethernet1) > IGMP: Enter the following, then click **Apply**:

   IGMP Mode: Host (select)
   Protocol IGMP: Enable (select)
   Packet From Different Subnet: Permit (select)

   Network > Interfaces > Edit (for ethernet3) > IGMP: Enter the following, then click **OK**:

   IGMP Mode: Router (select)
   Protocol IGMP: Enable (select)
   Packet From Different Subnet: Permit (select)
   Proxy: (select)
   Always (select)

CLI

1. **Zones and Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.2.1/24
set interface ethernet3 zone trust
set interface ethernet1 ip 10.1.1.1/24
```

2. **IGMP**

```
set interface ethernet1 protocol igmp host
set interface ethernet1 protocol igmp enable
set interface ethernet1 protocol igmp no-check-subnet
set interface ethernet3 protocol igmp router
set interface ethernet3 protocol igmp proxy
set interface ethernet3 protocol igmp proxy always
set interface ethernet3 protocol igmp enable
set interface ethernet3 protocol igmp no-check-subnet
save
```

Multicast Policies for IGMP and IGMP Proxy Configurations

Normally, a security device exchanges IGMP messages with its connected hosts only. With IGMP Proxy, security devices might need to send IGMP messages to a host or router in another zone. To allow IGMP messages across zones, you must configure a multicast policy that specifically allows this. When you create a multicast policy, you must specify the following:

- **Source**—The zone from which traffic is initiated
- **Destination**—The zone to which traffic is sent
- **Multicast group**—Can be a multicast group, an access list that specifies multicast groups, or "any"

In addition, you can specify that the policy is bidirectional to apply the policy to both directions of traffic.

Creating a Multicast Group Policy for IGMP

In this example, the router interface is on the Trust zone and the host interface is in the Untrust zone. You define a multicast policy that allows IGMP messages for the multicast group 224.2.202.99/32 to pass between the Trust and Untrust zones. You use the keyword bi-directional to allow traffic in both directions.

WebUI

MCast Policies (From: Trust, To: Untrust) > New: Enter the following, then click **OK**:

MGroup Address: IP/Netmask (select) 224.2.202.99/32
Bidirectional: (select)
IGMP Message: (select)

CLI

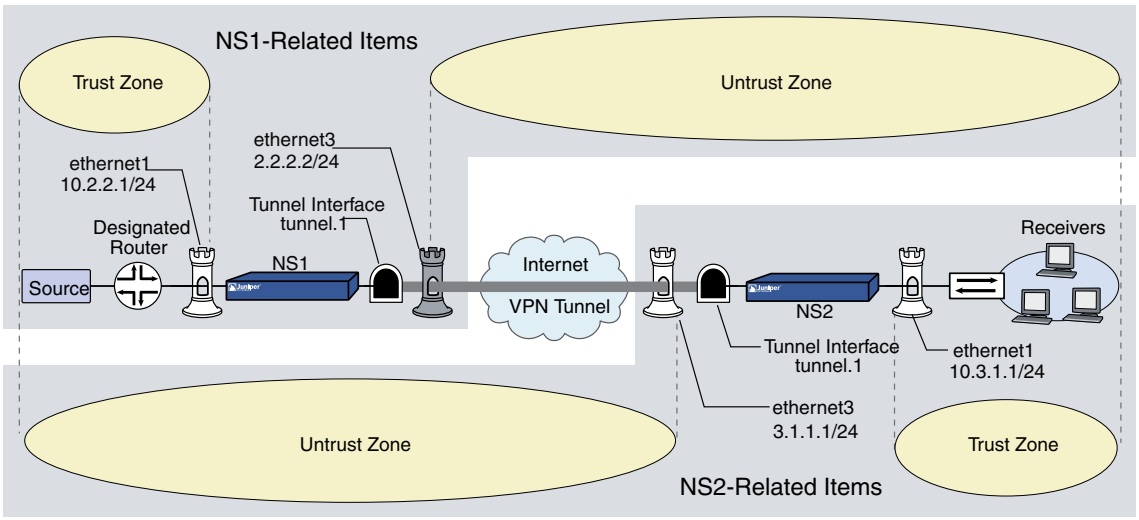
set multicast-group-policy from trust mgroup 224.2.202.99/32 to untrust
igmp-message bi-directional
save

Creating an IGMP Proxy Configuration

As shown in Figure 343 on page 1413, you configure IGMP proxy on the security devices NS1 and NS2. They are connected to each other through a VPN tunnel. Perform the following steps on the security devices at both locations:

1. Assign IP addresses to the physical interfaces bound to the security zones.
2. Create the address objects.
3. Enable IGMP on the host and router interfaces, and enable IGMP proxy on the router interface. (IGMP proxy is enabled by default on host interfaces.)
   a. Specify the keyword **always** on ethernet1 of NS1 to enable it to forward multicast traffic even if it is a non-querier.
   b. By default, an IGMP interface accepts IGMP packets from its own subnet only. In the example, the interfaces are on different subnets. When you enable IGMP, allow the interfaces to accept IGMP packets (queries, membership reports, and leave messages) from any subnet.
4. Set up routes.
5. Configure the VPN tunnel.
6. Configure a firewall policy to pass data traffic between zones.
7. Configure a multicast policy to pass IGMP messages between zones. In this example, you restrict multicast traffic to one multicast group (224.4.4.1/32).

Figure 343: IGMP Proxy Configuration Between Two Devices



WebUI (NS1)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.2.2.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   IP Address/Netmask: 2.2.2.2/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Unnumbered: (select)
   Interface: ethernet3 (trust-vr)

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: branch
   IP Address/Domain Name:
   IP/Netmask: (select), 10.3.1.0/24
   Zone: Untrust

3. **IGMP**

   Network > Interfaces > Edit (for ethernet1) > IGMP: Enter the following, then click **Apply**:

   IGMP Mode: Host (select)
   Protocol IGMP: Enable (select)
   Packet From Different Subnet: Permit (select)

   Network > Interfaces > Edit (for tunnel.1) > IGMP: Enter the following, then click **Apply**:

   IGMP Mode: Router (select)
   Protocol IGMP: Enable (select)
   Packet From Different Subnet: Permit (select)
   Proxy (select): Always (select)

4. **Routes**

   Network > Routing > Routing Entries > New: Enter the following, then click **OK**:

   Network Address / Netmask: 10.3.1.0 / 24
   Gateway (select):
   Interface: tunnel.1 (select)

5. **VPN**

   VPN > AutoKey Advanced > Gateway > New: Enter the following, then click OK.

   Gateway Name: To_Branch
   Security Level: Compatible
   Remote Gateway Type:
   Static IP Address: (select), IP Address/Hostname: 3.1.1.1
   Preshared Key: fg2g4h5j
   Outgoing Interface: ethernet3

   > > Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

   Security Level: Compatible
   Phase 1 Proposal (For Compatible Security Level): pre-g2-3des-sha
   Mode (Initiator): Main (ID Protection)

6. **Policy**

   Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), branch
   Destination Address:
   Address Book Entry: (select), any (select)
   Service: any
   Action: Permit

7. **Multicast Policy**

   MCast Policies > (From: Trust, To: Untrust) > New: Enter the following, then click **OK**:

   Mgroup Address: IP/Netmask (select): 224.4.4.1/32
   Bidirectional: (select)
   IGMP Message: (select)

WebUI (NS2)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

   Zone Name: Trust
   IP Address/Netmask: 10.3.1.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   IP Address/Netmask: 3.1.1.1/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Unnumbered: (select)
      Interface: ethernet3 (trust-vr)

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: mgroup1
   IP Address/Domain Name:
   IP/Netmask: (select), 224.4.4.1/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: source-dr
   IP Address/Domain Name:
      IP/Netmask: (select), 10.2.2.1/24
   Zone: Untrust

3. **IGMP**

   Network > Interfaces > Edit (for ethernet1) > IGMP: Enter the following, then click **Apply**:

IGMP Mode: Router (select)
Protocol IGMP: Enable (select)
Proxy (select): Always (select)

Network > Interfaces > Edit (for tunnel.1) > IGMP: Enter the following, then click **Apply**:

IGMP Mode: Host (select)
Protocol IGMP: Enable (select)
Packet From Different Subnet: Permit (select)

4. **Routes**

Network > Routing > Routing Entries > New (trust-vr): Enter the following, then click **OK**:

Network Address / Netmask: 10.2.2.0 / 24
Gateway (select):
  Interface: tunnel.1 (select)

5. **VPN**

VPN > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_Corp
Security Level: Compatible
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 1.1.1.1
Preshared Key: fg2g4hvj
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Compatible
Phase 1 Proposal (For Compatible Security Level): pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)

6. **Policy**

Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), source-dr
Destination Address:
  Address Book Entry: (select), mgroup1
Service: ANY
Action: Permit

7. **Multicast Policy**

MCast Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK**:

Mgroup Address: IP/Netmask (select): 224.4.4.1/32
Bidirectional: (select)
IGMP Message: (select)

CLI (NS1)

1. **Interfaces**

   Set interface ethernet1 zone trust
   set interface ethernet1 ip 10.2.2.1/24
   set interface ethernet3 zone untrust
   set interface ethernet3 ip 2.2.2.2/24
   set interface tunnel.1 zone untrust
   set interface tunnel.1 ip unnumbered interface ethernet3

2. **Addresses**

   set address untrust branch1 10.3.1.0/24

3. **IGMP**

   set interface ethernet1 protocol igmp host
   set interface ethernet1 protocol igmp enable
   set interface ethernet1 protocol igmp no-check-subnet
   set interface tunnel.1 protocol igmp router
   set interface tunnel.1 protocol igmp proxy
   set interface tunnel.1 protocol igmp proxy always
   set interface tunnel.1 protocol igmp enable
   set interface tunnel.1 protocol igmp no-check-subnet

4. **Routes**

   set route 10.3.1.0/24 interface tunnel.1

5. **VPN Tunnel**

   set ike gateway To_Branch address 3.1.1.1 main outgoing-interface ethernet3
   preshare fg2g4h5j proposal pre-g2-3des-sha
   set vpn Corp_Branch gateway To_Branch sec-level compatible
   set vpn Corp_Branch bind interface tunnel.1
   set vpn Corp_Branch proxy-id local-ip 10.2.2.0/24 remote-ip 10.3.1.0/24 any

6. **Policies**

   set policy name To_Branch from untrust to trust branch1 any any permit

7. **Multicast Policies**

   set multicast-group-policy from trust mgroup 224.4.4.1/32 to untrust
   igmp-message bi-directional
   save

CLI (NS2)

1. **Interfaces**

```
Set interface ethernet1 zone trust
set interface ethernet1 ip 10.3.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 3.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

2. **Addresses**

```
set address trust mgroup1 224.4.4.1/32
set address untrust source-dr 10.2.2.1/24
```

3. **IGMP**

```
set interface ethernet1 protocol igmp router
set interface ethernet1 protocol igmp proxy
set interface ethernet1 protocol igmp proxy always
set interface ethernet1 protocol igmp enable
set interface tunnel.1 protocol igmp host
set interface tunnel.1 protocol igmp enable
set interface tunnel.1 protocol igmp no-check-subnet
```

4. **Routes**

```
set route 10.2.2.0/24 interface tunnel.1
```

5. **VPN Tunnel**

```
set ike gateway To_Corp address 2.2.2.2 main outgoing-interface ethernet3
preshare fg2g4hvj proposal pre-g2-3des-sha
set vpn Branch_Corp gateway To_Corp sec-level compatible
set vpn Branch_Corp bind interface tunnel.1
set vpn Branch_Corp proxy-id local-ip 10.3.1.0/24 remote-ip 10.2.2.0/24 any
```

6. **Policy**

```
set policy from untrust to trust source-dr mgroup1 any permit
```

7. **Multicast Policy**

```
set multicast-group-policy from untrust mgroup 224.4.4.1/32 to trust
igmp-message bi-directional
save
```

Setting Up an IGMP Sender Proxy

In IGMP proxy, the multicast traffic usually travels downstream from the host interface to the router interface. In certain situations, the source can be in the same network as the router interface. When a source connected to an interface that is on the same

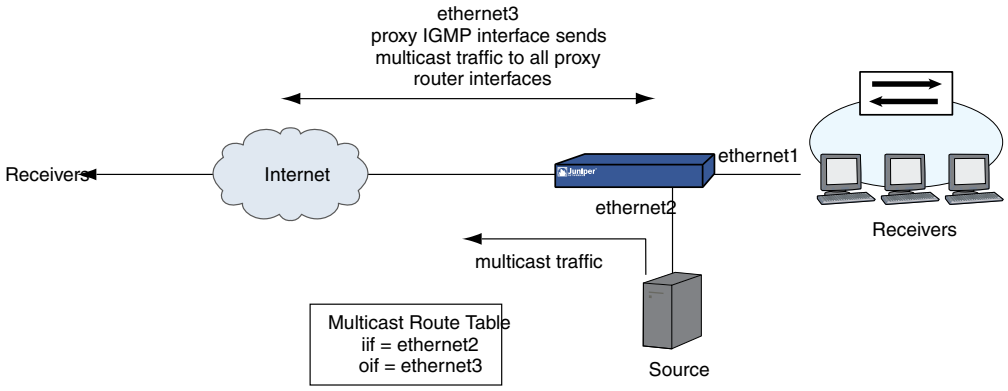
network as the IGMP router proxy interface sends multicast traffic, the security device checks for the following:

- A multicast group policy allowing traffic from the source zone to the zone of the IGMP proxy host interface
- An access list for acceptable sources

If there is no multicast policy between the source zone and the zone of the proxy IGMP interface or if the source is not on the list of acceptable sources, the security device drops the traffic. If there is a multicast policy between the source zone and the zone of the proxy IGMP interface, and the source is on the list of acceptable sources, then the device creates an (S,G) entry for that interface in the multicast route table; the incoming interface is the interface to which the source is connected and the outgoing interface is the IGMP proxy host interface. The security device then sends the data upstream to the IGMP proxy host interface which sends the data to all its connected proxy router interfaces, except to the interface connected to the source.

Figure 344 on page 1419 shows an example of IGMP sender proxy.

Figure 344: IGMP Sender Proxy



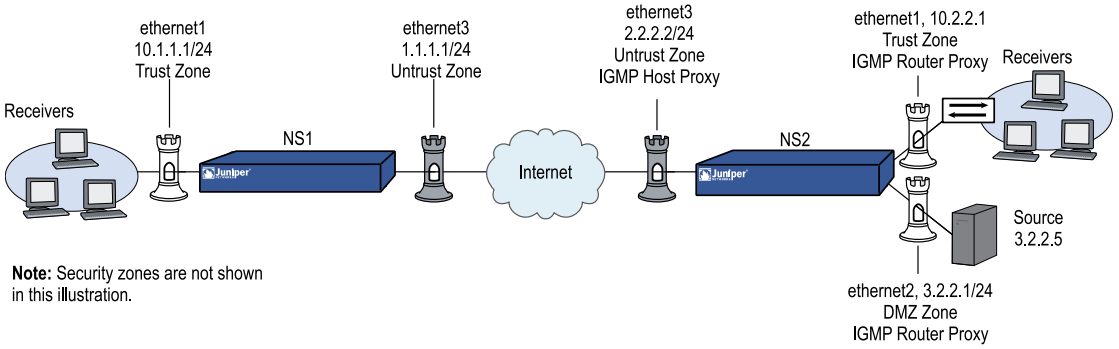
In Figure 345 on page 1420, the source is connected to the ethernet2 interface, which is bound to the DMZ zone on NS2. It is sending multicast traffic to the multicast group 224.4.4.1/32. There are receivers connected to the ethernet1 interface bound to the Trust zone on NS2. Both ethernet1 and ethernet2 are IGMP proxy router interfaces. The ethernet3 interface bound to the Untrust zone of NS2 is an IGMP proxy host interface. There are also receivers connected to the ethernet1 interface bound to the Trust zone on NS1. Perform the following steps on NS2:

1. Assign IP addresses to the interfaces bound to the security zones.
2. Create the address objects.
3. On ethernet1 and ethernet2:
   a. Enable IGMP in router mode and enable IGMP proxy.
   b. Specify the keyword **always** to enable the interfaces to forward multicast traffic even if they are not queriers.

4. Enable IGMP in host mode on ethernet3.
5. Set up the default route.
6. Configure firewall policies between the zones.
7. Configure multicast policies between the zones.

NOTE: This example includes only the configuration for NS2, not the configuration for NS1.

Figure 345: IGMP Sender Proxy Network Example



WebUI (NS2)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.2.2.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select this option when present)
   IP Address/Netmask: 3.2.2.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 2.2.2.2/24

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

Address Name: mgroup1
IP Address/Domain Name:
IP/Netmask: (select), 224.4.4.1/32
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

Address Name: source-dr
IP Address/Domain Name:
IP/Netmask: (select), 3.2.2.5/32
Zone: DMZ

Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

Address Name: proxy-host
IP Address/Domain Name:
IP/Netmask: (select), 2.2.2.2/32
Zone: Untrust

3. **IGMP**

Network > Interfaces > Edit (for ethernet1) > IGMP: Enter the following, then click **Apply**:

IGMP Mode: Router (select)
Protocol IGMP: Enable (select)
Proxy (select): Always (select)

Network > Interfaces > Edit (for ethernet2) > IGMP: Enter the following, then click **Apply**:

IGMP Mode: Router (select)
Protocol IGMP: Enable (select)
Proxy (select): Always (select)

Network > Interfaces > Edit (for ethernet3) > IGMP: Enter the following, then click **Apply**:

IGMP Mode: Host (select)
Protocol IGMP: Enable (select)
Packet From Different Subnet: Permit (select)

4. **Route**

Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 2.2.2.250

5. **Policy**

   Policies > (From: DMZ, To: Trust) > New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: source-dr
   Destination Address:
   Address Book Entry: (select), mgroup1
   Service: ANY
   Action: Permit

   Policies > (From: DMZ, To: Untrust) > New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), source-dr
   Destination Address:
   Address Book Entry: (select), mgroup1
   Service: ANY
   Action: Permit

   Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), proxy-host
   Destination Address:
   Address Book Entry: (select), mgroup1
   Service: ANY
   Action: Permit

6. **Multicast Policy**

   MCast Policies > (From: DMZ, To: Untrust) > New: Enter the following, then click **OK**:

   Mgroup Address: IP/Netmask (select): 224.4.4.1/32
   Bidirectional: (select)
   IGMP Message: (select)

   MCast Policies > (From: DMZ, To: Trust) > New: Enter the following, then click **OK**:

   Mgroup Address: IP/Netmask (select): 224.4.4.1/32
   Bidirectional: (select)
   IGMP Message: (select)

MCast Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK**:

Mgroup Address: IP/Netmask (select): 224.4.4.1/32
Bidirectional: (select)
IGMP Message: (select)

CLI (NS2)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet2 zone dmz
set interface ethernet2 ip 3.2.2.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
```

2. **Addresses**

```
set address trust mgroup1 224.4.4.1/32
set address dmz source-dr 3.2.2.5/32
set address untrust proxy-host 2.2.2.2/32
```

3. **IGMP**

```
set interface ethernet1 protocol igmp router
set interface ethernet1 protocol igmp proxy always
set interface ethernet1 protocol igmp enable
set interface ethernet2 protocol igmp router
set interface ethernet2 protocol igmp proxy always
set interface ethernet2 protocol igmp enable
set interface ethernet3 protocol igmp host
set interface ethernet3 protocol igmp no-check-subnet
set interface ethernet3 protocol igmp enable
```

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
```

5. **Policies**

```
set policy from dmz to trust source-dr mgroup1 any permit
set policy from dmz to untrust source-dr mgroup1 any permit
set policy from untrust to trust proxy-host mgroup1 any permit
```

6. **Multicast Policies**

```
set multicast-group-policy from dmz mgroup 224.4.4.1/32 to untrust
igmp-message bi-directional
set multicast-group-policy from dmz mgroup 224.4.4.1/32 to trust igmp-message
 bi-directional
set multicast-group-policy from trust mgroup 224.4.4.1/32 to untrust
igmp-message bi-directional
save
```

Chapter 40

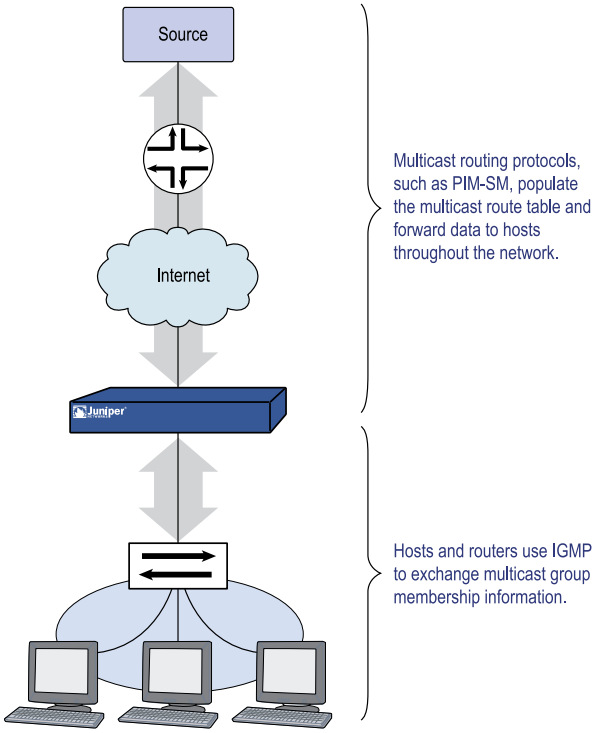
Protocol Independent Multicast

This chapter explains how to configure Protocol Independent Multicast-Sparse Mode (PIM-SM) and Protocol Independent Multicast-Source Specific Multicast (PIM-SSM) on Juniper Networks security devices. It includes the following sections:

Overview

Protocol Independent Multicast (PIM) is a multicast routing protocol that runs between routers. Whereas the Internet Group Management Protocol (IGMP) runs between hosts and routers to exchange multicast group membership information, PIM runs between routers to forward multicast traffic to multicast group members throughout the network. (For information about IGMP, see "Internet Group Management Protocol" on page 1399.)

Figure 346: IGMP



When you run PIM, you must also configure either static routes or a dynamic routing protocol. PIM is called *protocol independent* because it uses the route table of the underlying unicast routing protocol to perform its RPF (reverse path forwarding) checks, but does not depend on the functionality of the unicast routing protocol. (For information about RPF, see "Reverse Path Forwarding" on page 1392.)

PIM can operate in the following modes:

- PIM-Dense Mode (PIM-DM) floods multicast traffic throughout the network and then prunes routes to receivers that do not want to receive the multicast traffic.
- PIM-Sparse Mode (PIM-SM) forwards multicast traffic only to those receivers that request it. Routers running PIM-SM can use the shared path tree or shortest path tree (SPT) to forward multicast information. (For more information, see "Multicast Distribution Trees" on page 1427.)
- PIM-Source Specific Multicast Mode (PIM-SSM) is derived from PIM-SM. Like PIM-SM, it forwards multicast traffic to interested receivers only. Unlike PIM-SM, it immediately forms an SPT to the source.

  Juniper Networks security devices support PIM-SM, as defined in *draft-ietf-pim-sm-v2-new-06*; and PIM-SSM as defined in RFC 3569, *An Overview of Source-Specific Multicast (SSM)*. For information about PIM-SM, see "PIM-SM" on page 1427 For information about PIM-SSM, see "PIM-SSM" on page 1431.

PIM-SM

PIM-SM is a multicast routing protocol that forwards multicast traffic to interested receivers only. It can use either a shared distribution tree or the shortest path tree (SPT) to forward multicast traffic throughout the network. (For information about multicast distribution trees, see "Multicast Distribution Trees" on page 1427.) By default, PIM-SM uses the shared distribution tree with a rendezvous point (RP) at the root of the tree. All sources in a group send their packets to the RP, and the RP sends data down the shared distribution tree to all receivers in a network. When a configured threshold is reached, the receivers can form an SPT to the source, decreasing the time it takes the receivers to receive the multicast data.

NOTE: By default, Juniper Networks security devices switch to the SPT upon receiving the first byte.

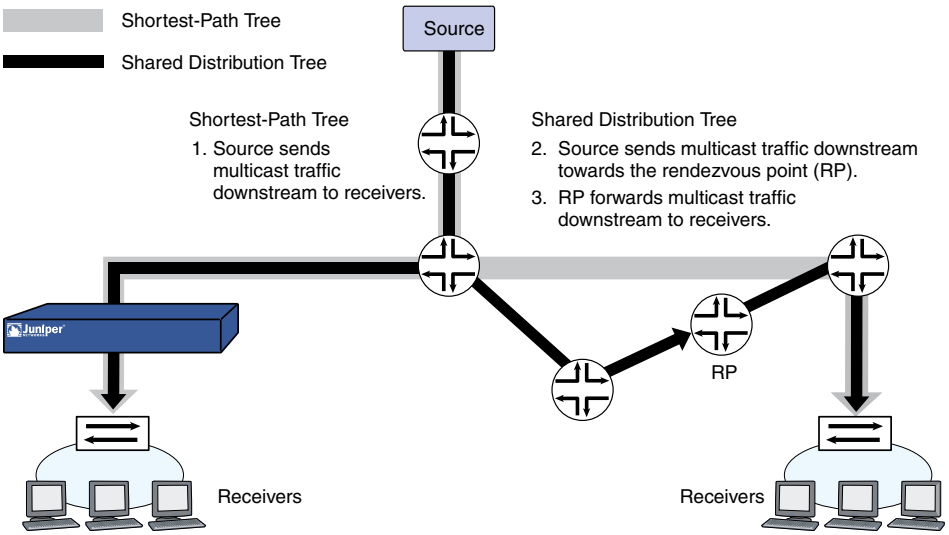
Regardless of which tree is used to distribute traffic, only receivers that explicitly join a multicast group can receive the traffic for that group. PIM-SM uses the unicast routing table to perform its reverse path forwarding (RPF) lookups when it receives multicast control messages, and it uses the multicast routing table to send multicast data traffic to receivers.

Multicast Distribution Trees

Multicast routers forward multicast traffic downstream from the source to the receivers through a multicast distribution tree. There are two types of multicast distribution trees:

- Shortest-Path Tree (SPT)—The source is at the root of the tree and forwards the multicast data downstream to each receiver. This is also referred to as a source-specific tree.
- Shared Distribution Tree—The source transmits the multicast traffic to the rendezvous point (RP), which is typically a router at the core of the network. The RP then forwards the traffic downstream to receivers on the distribution tree.

Figure 347: PIM



Designated Router

When there are multiple multicast routers in a multi-access local area network (LAN), the routers elect a designated router (DR). The DR on the LAN of the source is responsible for sending the multicast packets from the source to the RP and to the receivers that are on the source-specific distribution tree. The DR on the LAN of the receivers is responsible for forwarding join-prune messages from the receivers to the RP, and for sending multicast data traffic to the receivers in the LAN. Receivers send join-prune messages when they want to join or leave a multicast group.

The DR is selected through an election process. Each PIM-SM router in a LAN has a DR priority that is user configurable. PIM-SM routers advertise their DR priorities in hello messages they periodically send their neighbors. When the routers receive the hello messages, they select the router with the highest DR priority as the DR for the LAN. If multiple routers have the highest DR priority, then the router with the highest IP address becomes the DR of the LAN.

Mapping Rendezvous Points to Groups

A rendezvous point (RP) sends multicast packets for specific multicast groups. A PIM-SM domain is a group of PIM-SM routers that have the same RP-group mappings. There are two ways to map multicast groups to an RP: statically and dynamically.

Static RP Mapping

To create a static mapping between an RP and a multicast group, you must configure the RP for the multicast group on each router in the network. Each time the address of the RP changes, you must reconfigure the RP address.

Dynamic RP Mapping

PIM-SM also provides a mechanism for dynamically mapping RPs to multicast groups. First, you configure candidate rendezvous points (C-RPs) for each multicast group. Then, the C-RPs send Candidate-RP advertisements to one router in the LAN, called the bootstrap router (BSR). The advertisements contain the multicast group(s) for which the router is to be an RP and the priority of the C-RP.

The BSR collects these C-RP advertisements and sends them out in a BSR message to all routers in the domain. The routers collect these BSR messages and use a well-known hash algorithm to select one active RP per multicast group. If the selected RP fails, then the router selects a new RP-group mapping from among the candidate RPs. For information about the BSR selection process, refer to *draft-ietf-pim-sm-bsr-03.txt*.

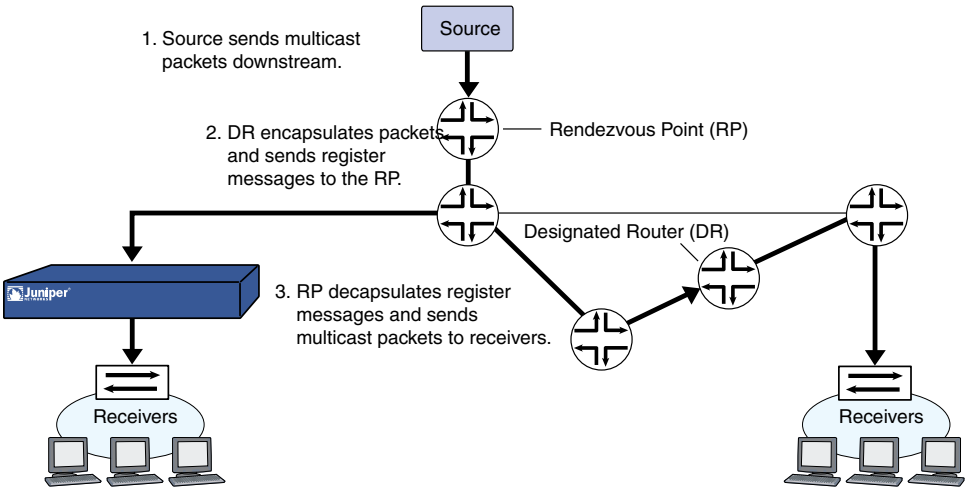
Forwarding Traffic on the Distribution Tree

This section describes how PIM-SM routers send join messages toward the rendezvous point (RP) of a multicast group and how the RP sends multicast data to the receivers in the network. In a multicast networking environment, a security device can function as an RP, a designated router either in the source network or the receivers' network, or an intermediate router.

Source Sends Data to a Group

When a source starts sending multicast packets, it transmits the packets on the network. When the designated router (DR) on that local area network (LAN) receives the multicast packets, it looks up the outgoing interface and next-hop IP address toward the RP in the unicast route table. Then the DR encapsulates the multicast packets in unicast packets, called register messages, and forwards them to the next hop IP address. When the RP receives the register messages, it decapsulates the packets and sends the multicast packets down the distribution tree toward the receivers.

Figure 348: Source Sending Data

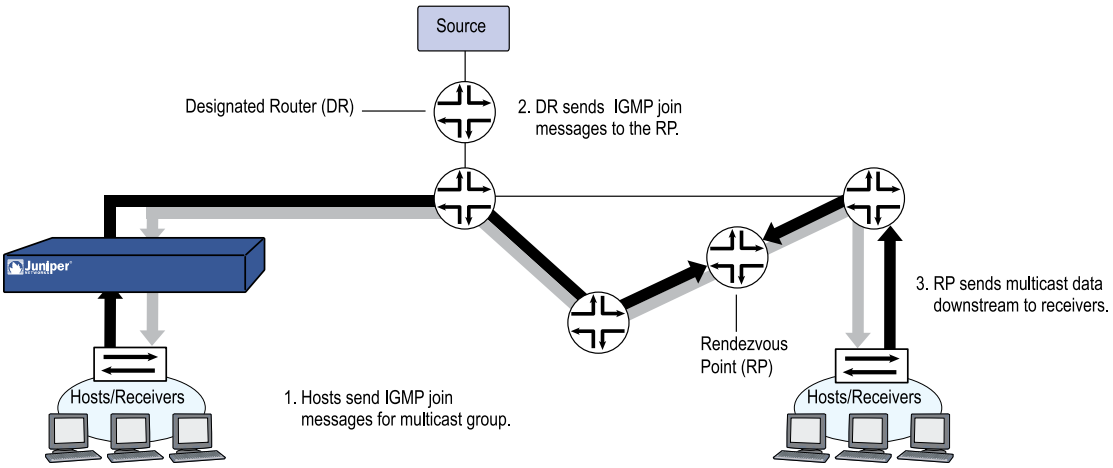


If the data rate from the source DR reaches a configured threshold, the RP sends a PIM-SM join message toward the source DR so the RP can receive the native multicast data, instead of the register messages. When the source DR receives the join message, it sends the multicast packets and the register messages toward the RP. When the RP receives the multicast packets from the DR, it sends the DR a register-stop message. When the DR receives the register-stop message, it stops sending the register messages and sends the native multicast data, which the RP then sends downstream to the receivers.

Host Joins a Group

When a host joins a multicast group, it sends an IGMP join message to that multicast group. When the DR on the LAN of the host receives the IGMP join message, it looks up the RP for the group. It creates a (*,G) entry in the multicast route table and sends a PIM-SM join message to its RPF neighbor upstream toward the RP. When the upstream router receives the PIM-SM join message, it performs the same RP lookup process and also checks if the join message came from an RPF neighbor. If it did, then it forwards the PIM-SM join message toward the RP. This continues until the PIM-SM join message reaches the RP. When the RP receives the join message, it sends the multicast data downstream toward the receiver.

Figure 349: Host Joining a Group



Each downstream router performs an RPF check when it receives the multicast data. Each router checks if it received the multicast packets from the same interface it uses to send traffic toward the RP. If the RPF check is successful, the router then looks for a matching (*, G) forwarding entry in the multicast route table. If it finds the (*, G) entry, it places the source in the entry, which becomes an (S, G) entry, and forwards the multicast packets downstream. This process continues down the distribution tree until the host receives the multicast data.

When the traffic rate reaches a configured threshold, the DR on the LAN of the host can form the shortest-path tree directly to the multicast source. When the DR starts receiving traffic directly from the source, it sends a source-specific prune message upstream toward the RP. Each intermediate router “prunes” the link to the host off the distribution tree, until the prune message reaches the RP, which then stops sending the multicast traffic down that particular branch of the distribution tree.

PIM-SSM

In addition to PIM-SM, security devices also support PIM-Source-Specific Multicast (SSM). PIM-SSM follows the source-specific model (SSM) where multicast traffic is transmitted to channels, not just multicast groups. A channel consists of a source and multicast group. A receiver subscribes to a channel with a known source and multicast group. The receivers provide information about the source through IGMPv3. The designated router on the LAN sends messages to the source and not to a rendezvous point (RP).

The IANA has reserved the multicast address range 232/8 for the SSM service in IPv4. If IGMPv3 is running on a device along with PIM-SM, PIM-SSM operations are guaranteed within this address range. The security device handles IGMPv3 membership reports for multicast groups within the 232/8 address range as follows:

- If the report contains a filter-mode of include, the device sends the report directly to the sources in the source list.
- If the report contains a filter mode of exclude, the device drops the report. It does not process (*,G) reports for multicast groups in the 232/8 address range.

The steps for configuring PIM-SSM on a security device are the same as those for configuring PIM-SM with the following differences:

- You must configure IGMPv3 on interfaces connected to receivers. (IGMPv2 is enabled by default on security devices.)
- When you configure a multicast group policy, allow join-prune messages. (Bootstrap messages are not used.)
- You do not configure an Rendezvous Point.

The next sections explain how to configure PIM-SM on security devices.

Configuring PIM-SM on Security Devices

Juniper Networks security devices have two predefined virtual routers (VRs): a trust-vr and an untrust-vr. Each virtual router is a separate routing component with its own route tables. Protocol Independent Multicast-Sparse Mode (PIM-SM) uses the route table of the virtual router on which it is configured to look up the reverse path forwarding (RPF) interface and next-hop IP address. Therefore, to run PIM-SM on a security device, you must first configure either static routes or a dynamic routing protocol on a virtual router, and then configure PIM-SM on the same virtual router. (For information about virtual routers, see "Routing" on page 1235.) Security devices support the following dynamic routing protocols:

- Open Shortest Path First (OSPF)—For more information, see "Open Shortest Path First" on page 1269.
- Routing Information Protocol (RIP)—For more information, see "Routing Information Protocol" on page 1307.
- Border Gateway Protocol (BGP)—For more information, see "Border Gateway Protocol" on page 1337.

The following sections describe the basic steps for configuring PIM-SM on a security device:

- Creating and enabling a PIM-SM instance in a VR
- Enabling PIM-SM on interfaces
- Configuring a multicast policy to allow PIM-SM messages to cross the security device

Enabling and Deleting a PIM-SM Instance for a VR

You can configure one PIM-SM instance for each VR. PIM-SM uses the unicast route table of the VR to perform its RPF check. After you create and enable a PIM-SM routing instance on a VR, you can then enable PIM-SM on the interfaces in the VR.

Enabling PIM-SM Instance

In this example, you create and enable a PIM-SM instance for the trust-vr virtual router.

WebUI

Network > Routing > Virtual Routers > Edit (for trust-vr) > Create PIM Instance: Select **Protocol PIM: Enable**, then click **Apply**.

CLI

```
device-> set vrouter trust-vr
device(trust-vr)-> set protocol pim
device(trust.vr/pim)-> set enable
device(trust.vr/pim)-> exit
device(trust-vr)-> exit
save
```

Deleting a PIM-SM Instance

In this example, you delete the PIM-SM instance in the trust-vr virtual router. When you delete the PIM-SM instance in a virtual router, the security device disables PIM-SM on the interfaces and deletes all PIM-SM interface parameters.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Delete PIM Instance, then click **OK** at the confirmation prompt.

CLI

```
unset vrouter trust-vr protocol pim
deleting PIM instance, are you sure? y/[n] y
save
```

Enabling and Disabling PIM-SM on Interfaces

By default, PIM-SM is disabled on all interfaces. After you create and enable PIM-SM in a virtual router, you must enable PIM-SM on the interfaces within that virtual router that transmit multicast traffic. If an interface is connected to a receiver, you must also configure IGMP in router mode on that interface. (For information about IGMP, see "Internet Group Management Protocol" on page 1399.)

When you enable PIM-SM on an interface that is bound to a zone, PIM-SM is automatically enabled in the zone to which that interface belongs. You can then configure PIM-SM parameters for that zone. Similarly, when you disable PIM-SM parameters on interfaces in a zone, then all PIM-SM parameters related to the zone are automatically deleted.

Enabling PIM-SM on an Interface

In this example, you enable PIM-SM on the ethernet1 interface.

WebUI

Network > Interfaces > Edit (for ethernet1) > PIM: Enter the following, then click **Apply**:

PIM Instance: (select)
Protocol PIM: Enable (select)

CLI

```
set interface ethernet1 protocol pim
set interface ethernet1 protocol pim enable
save
```

Disabling PIM-SM on an Interface

In this example, you disable PIM-SM on the ethernet1 interface. Note that any other interfaces on which you have enabled PIM-SM are still transmitting and processing PIM-SM packets.

WebUI

Network > Interfaces > Edit (for ethernet1) > PIM: Clear **Protocol PIM Enable**, then click **Apply**.

CLI

```
unset interface ethernet1 protocol pim enable
save
```

Multicast Group Policies

By default, security devices do not allow multicast control traffic, such as PIM-SM messages, to pass between zones. You must configure a multicast group policy to allow PIM-SM messages between zones. Multicast group policies control two types of PIM-SM messages: static-RP-BSR messages and join-prune messages.

Static-RP-BSR Messages

Static-RP-BSR messages contain information about static rendezvous points (RPs) and dynamic RP-group mappings. Configuring a multicast policy that allows static RP mappings and bootstrap (BSR) messages between zones enables the security device to share RP-group mappings across zones within a virtual router or between two virtual routers. Routers are able to learn about RP-group mappings from other zones, so you do not have to configure RPs in all zones.

When the security device receives a BSR message, it verifies that it came from its reverse path forwarding (RPF) neighbor. Then it checks if there are multicast policies for the multicast groups in the BSR message. It filters out groups not allowed in the

multicast policy and sends the BSR message for the allowed groups to all destination zones that are allowed by the policy.

Join-Prune Messages

Multicast group policies also control join-prune messages. When the security device receives a join-prune message for a source and group or source and RP on its downstream interface, it looks up the RPF neighbor and interface in the unicast routing table.

- If the RPF interface is on the same zone as the downstream interface, then multicast policy validation is not necessary.
- If the RPF interface is on another zone, then the security device checks if there is a multicast policy that allows join-prune messages for the group between the zone of the downstream interface and the zone of the RPF interface.
  - If there is a multicast policy that allows join-prune messages between the two zones, the security device forwards the message to the RPF interface.
  - If there is no multicast policy that allows join-prune messages between the two zones, then it drops the join-prune message.

Defining a Multicast Group Policy for PIM-SM

In this example, you define a bi-directional multicast group policy that allows all PIM-SM messages between the Trust and Untrust zones for group 224.4.4.1.

WebUI

Policies (From: Trust, To: Untrust) > New: Enter the following, then click **OK**:

MGroup Address: IP/Netmask (select) 224.4.4.1/32
Bidirectional: (select)
PIM Message: (select)
BSR-Static RP: (select)
Join/Prune: (select)

CLI

```
set multicast-group-policy from trust mgroup 224.4.4.1/32 to untrust pim-message
 bsr-static-rp join-prune bi-directional
save
```

Setting a Basic PIM-SM Configuration

A security device can function as a rendezvous point (RP), source designated router (DR), receiver DR, and intermediate router. It cannot function as a bootstrap router.

You can configure PIM-SM on one virtual router (VR) or across two VRs. Perform the following steps to configure PIM-SM on one virtual router:

1. Configure zones and interfaces.
2. Configure either static routes or a dynamic routing protocol such as Routing Information Protocol (RIP), Border Gateway Protocol (BGP) or Open Shortest Path First (OSPF) on a specific virtual router on the security device.
3. Create a firewall policy to pass unicast and multicast data traffic between zones.
4. Create and enable a PIM-SM routing instance on the same virtual router on which you configured the static routes or a dynamic routing protocol.
5. Enable PIM-SM on interfaces forwarding traffic upstream toward the source or RP, and downstream toward the receivers.
6. Enable IGMP on interfaces connected to hosts.
7. Configure a multicast policy to permit PIM-SM messages between zones.

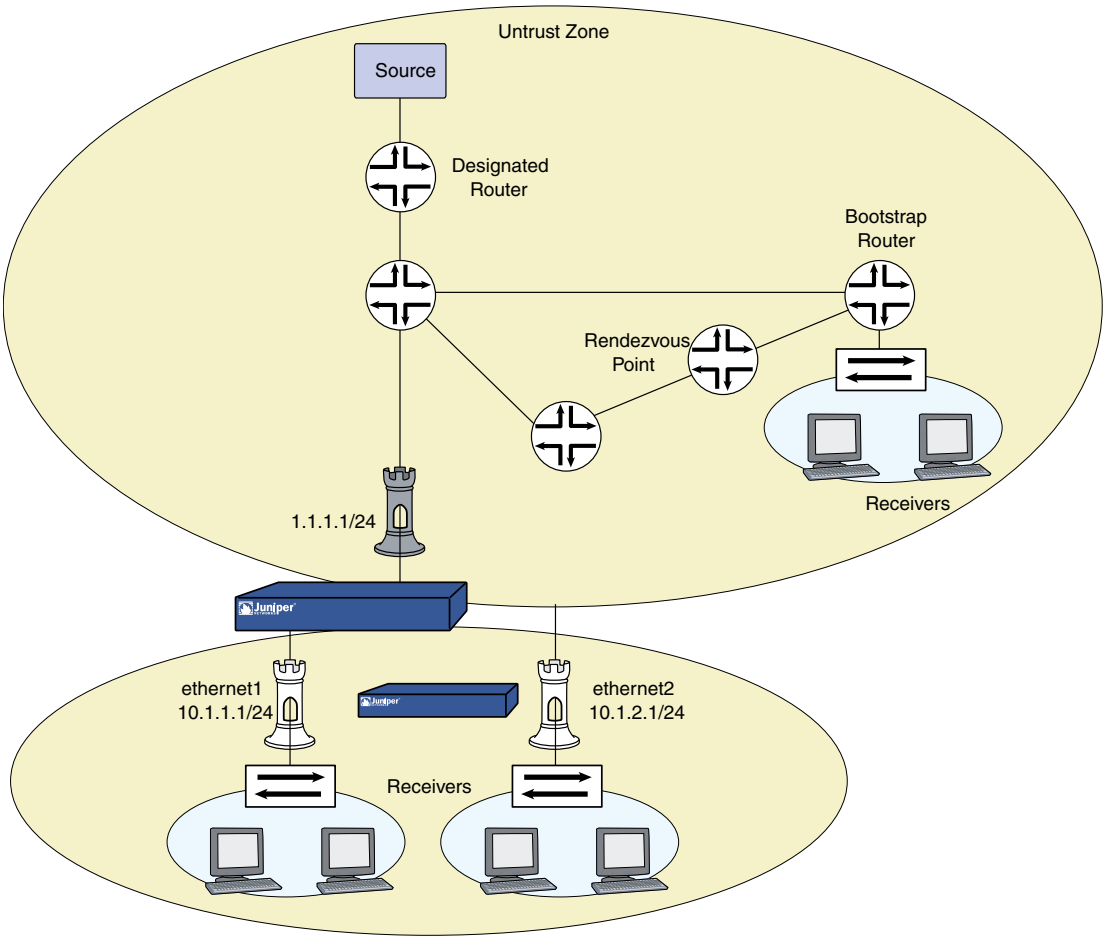
When you configure PIM-SM across two VRs, you must configure the RP in the zone of the VR in which the RP is located. Then, configure a multicast group policy allowing join-prune and BSR-static-RP messages between the zones in each VR. You must also export unicast routes between the two VRs to ensure the accuracy of the reverse path forwarding (RPF) information. For information about exporting routes, see "Exporting and Importing Routes Between Virtual Routers" on page 1265.

NOTE: If a security device is configured with multiple VRs, all VRs must have the same PIM-SM options.

Some Juniper Networks security devices support multiple virtual systems. (For information about virtual systems, see *"Virtual Private Networks" on page 705*.) When you configure PIM-SM in a virtual system, it is the same as configuring PIM-SM in the root system. When you configure PIM-SM on two virtual routers that are each in a different virtual system, then you must configure a proxy RP. (For information about configuring a proxy RP, see "Configuring a Proxy Rendezvous Point" on page 1449.)

In this example, you configure PIM-SM in the trust-vr. You want hosts in the Trust zone to receive multicast traffic for the multicast group 224.4.4.1/32. You configure RIP as the unicast routing protocol in the trust-vr and create a firewall policy to pass data traffic between the Trust and Untrust zones. You create a PIM-SM instance in the trust-vr and enable PIM-SM on ethernet1 and ethernet2 in the Trust zone, and on ethernet3 in the Untrust zone. All interfaces are in route mode. Then, you configure IGMP on ethernet1 and ethernet2, which are connected to receivers. Finally, create a multicast policy that permits static-RP-BSR and join-prune messages between the zones.

Figure 350: Basic PIM-SM Configuration



WebUI

1. **Zones and Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT
   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.2.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.2.1/24
Select the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
IP Address/Netmask: 1.1.1.1/24

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

Address Name: mgroup1
IP Address/Domain Name:
IP/Netmask: (select), 224.4.4.1/32
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

Address Name: source-dr
IP Address/Domain Name:
IP/Netmask: (select), 6.6.6.1/24
Zone: Untrust

3. **IGMP**

Network > Interfaces > Edit (for ethernet1) > IGMP: Enter the following, then click **OK**:

IGMP Mode: Router (select)
Protocol IGMP: Enable (select)

Network > Interfaces > Edit (for ethernet2) > IGMP: Enter the following, then click **OK**:

IGMP Mode: Router (select)
Protocol IGMP: Enable (select)

4. **RIP**

Network > Routing > Virtual Router (trust-vr) > Edit > Create RIP Instance: Select **Enable RIP**, then click **OK**.

Network > Interfaces > Edit (for ethernet3) > RIP: Enter the following, then click **Apply**:

RIP Instance: (select)
Protocol RIP: Enable (select)

5. **PIM-SM**

   Network > Routing > Virtual Router (trust-vr) > Edit > Create PIM Instance: Select the following, then click **OK**.

   Protocol PIM: Enable (select)

   Network > Interfaces > Edit (for ethernet1) > PIM: Enter the following, then click **Apply**:

   PIM Instance: (select)
   Protocol PIM: Enable (select)

   Network > Interfaces > Edit (for ethernet2) > PIM: Enter the following, then click **Apply**:

   PIM Instance: (select)
   Protocol PIM: Enable (select)

   Network > Interfaces > Edit (for ethernet3) > PIM: Enter the following, then click **Apply**:

   PIM Instance: (select)
   Protocol PIM: Enable (select)

6. **Policy**

   Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK**:

   Source Address:
     Address Book Entry: (select), source-dr
   Destination Address:
     Address Book Entry: (select), mgroup1
   Service: any
   Action: Permit

7. **Multicast Policy**

   MCast Policies (From: Trust, To: Untrust) > New: Enter the following and click **OK**:

   MGroup Address: IP/Netmask (select) 224.4.4.1/32
   Bidirectional: (select)
   PIM Message: (select)
   BSR Static RP: (select)
   Join/Prune: (select)

CLI

1. **Zones and Interfaces**

   ```
   set interface ethernet1 zone trust
   set interface ethernet1 ip 10.1.1.1/24
   set interface ethernet1 nat
   ```

```
set interface ethernet2 zone trust
set interface ethernet2 ip 10.1.2.1/24
set interface ethernet2 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Addresses**

```
set address trust mgroup1 224.4.4.1/32
set address untrust source-dr 6.6.6.1/24
```

3. **IGMP**

```
set interface ethernet1 protocol igmp router
set interface ethernet1 protocol igmp enable
set interface ethernet2 protocol igmp router
set interface ethernet2 protocol igmp enable
```

4. **RIP**

```
set vrouter trust-vr protocol rip
set vrouter trust-vr protocol rip enable
set interface ethernet3 protocol rip enable
```

5. **PIM-SM**

```
set vrouter trust-vr protocol pim
set vrouter trust-vr protocol pim enable
set interface ethernet1 protocol pim
set interface ethernet1 protocol pim enable
set interface ethernet2 protocol pim
set interface ethernet2 protocol pim enable
set interface ethernet3 protocol pim
set interface ethernet3 protocol pim enable
```

6. **Policy**

```
set policy from untrust to trust source-dr mgroup1 any permit
```

7. **Multicast Policy**

```
set multicast-group-policy from trust mgroup 224.4.4.1/32 any to untrust
pim-message bsr-static-rp join bi-directional
save
```

Verifying the Configuration

To verify the PIM-SM configuration, execute the following command:

```
device->  get vrouter trust protocol pim
PIM-SM enabled
Number of interfaces : 1
SPT threshold        : 1 Bps
PIM-SM Pending Register Entries Count : 0
Multicast group accept policy list: 1
```

```
Virtual Router trust-vr - PIM RP policy
-------------------------------------------------
Group Address      RP access-list
Virtual Router trust-vr - PIM source policy
-------------------------------------------------
Group Address      Source access-list
```

To view the multicast route entries, execute the following command:

```
device->  get igmp group
total groups matched: 1
multicast group  interface    last reporter   expire ver
*224.4.4.1       trust        0.0.0.0         ------ v2

device->get vrouter trust protocol pim mroute
trust-vr - PIM-SM routing table
------------------------------------------------------------------------------
Register - R, Connected members - C, Pruned - P, Pending SPT Alert - G
Forward - F, Null - N, Negative Cache - E, Local Receivers - L
SPT - T, Proxy-Register - X, Imported - I, SGRpt state - Y, SSM Range Group - S
Turnaround Router - K
------------------------------------------------------------------------------
Total PIM-SM mroutes: 2

(*, 236.1.1.1)  RP 20.20.20.10         01:54:20/-         Flags: LF
  Zone            : Untrust
  Upstream        : ethernet1/2         State          : Joined
  RPF Neighbor    : local               Expires        : -
  Downstream      :
  ethernet1/2  01:54:20/-         Join         0.0.0.0          FC

(10.10.10.1/24, 238.1.1.1)             01:56:35/00:00:42  Flags: TLF  Register
Prune
  Zone            : Trust
  Upstream        : ethernet1/1         State          : Joined
  RPF Neighbor    : local               Expires        : -
  Downstream      :
  ethernet1/2  01:54:20/-         Join         236.1.1.1          20.20.20.200
FC
```

You can verify the following in each route entry:

- The (S, G) state or (*, G) forwarding state
- If the forwarding state is (*, G), the RP IP address; If the forwarding state is (S, G), the source IP address
- Zone that owns the route
- The “join” status and the incoming and outgoing interfaces
- Timer values

To view the rendezvous points in each zone, execute the following command:

```
device->  get vrouter trust protocol pim rp
Flags : I - Imported, A - Always(override BSR mapping)
        C - Static Config, P - Static Proxy
------------------------------------------------------------------------------
```

```
Trust
  238.1.1.1/32       RP: 10.10.10.10     192    Static  -     C
    Registering : 0
    Active Groups : 1
               238.1.1.1
Untrust
  236.1.1.1/32       RP: 20.20.20.10     192    Static  -     P
    Registering : 0
    Active Groups : 1
               236.1.1.1
```

To verify that there is a Reverse Path Forwarding neighbor, execute the following command:

```
device->  get vrouter trust protocol pim rpf
Flags : RP address - R, Source address - S
Address         RPF Interface      RPF Neighbor    Flags
-----------------------------------------------------------
10.10.11.51     ethernet3          10.10.11.51     R
10.150.43.133   ethernet3          10.10.11.51     S
```

To view the status of join-prune messages the security device sends to each neighbor in a virtual router, execute the following command:

```
device->  get vrouter untrust protocol pim join
Neighbor        Interface       J/P          Group           Source
-------------------------------------------------------------------------
1.1.1.1         ethernet4:1    (S,G)     J  224.11.1.1      60.60.0.1
                               (S,G)     J  224.11.1.1      60.60.0.1
```

Configuring Rendezvous Points

You can configure a static rendezvous point (RP) when you want to bind a specific RP to one or more multicast groups. You can configure multiple static RPs, with each RP mapped to a different multicast group.

You must configure a static RP when there is no bootstrap router in the network. Although a security device can receive and process bootstrap messages, it does not function as a bootstrap router.

You can configure a virtual router as a candidate RP (C-RP) when you want to map RPs dynamically to multicast groups. You can create one C-RP for each zone.

Configuring a Static Rendezvous Point

When you configure a static RP, you specify the following:

- The zone of the static RP
- IP address of the static RP
- An access list that defines the multicast groups of the static RP (For more information, see “Access Lists” on page 1394.)

To ensure that the multicast groups in the access list always use the same RP, include the keyword **always**. If you do not include this keyword, and the security device discovers another RP dynamically mapped to the same multicast groups, it uses the dynamic RP.

In this example, you create an access list for the multicast group 224.4.4.1, and then create a static RP for that group. The IP address of the static RP is 1.1.1.5/24. You specify the keyword **always** to ensure that the security device always uses the same RP for that.

WebUI

Network > Routing > Virtual Routers > Access List: > New (for trust-vr): Enter the following, then click **OK**:

Access List ID: 2
Sequence No.: 1
IP/Netmask: 224.4.4.1/32
Action: Permit

Network > Routing > Virtual Router (trust-vr) > Edit > Edit PIM Instance > RP Address > New: Select the following, then click **OK**:

Zone: Trust (select)
Address: 1.1.1.5
Access List: 2
Always: (select)

CLI

```
set vrouter trust-vr access-list 2 permit ip 224.4.4.1/32 1
set vrouter trust-vr protocol pim zone trust rp address 1.1.1.5 mgroup-list 2 always
save
```

Configuring a Candidate Rendezvous Point

When you configure a virtual router as a C-RP, you specify the following:

- The zone in which the C-RP is configured
- IP address of the interface that is advertised as the C-RP
- An access list that defines the multicast groups of the C-RP
- The advertised C-RP priority

In this example, you enable PIM-SM on the ethernet1 interface which is bound to the Trust zone. You create an access list that defines the multicast groups of the C-RP. Then you create a C-RP in the Trust zone of the trust-vr. You set the priority of the C-RP to 200.

WebUI

Network > Interfaces > Edit (for ethernet1) > PIM: Enter the following, then click **Apply**:

PIM Instance: (select)
Protocol PIM: Enable (select)

Network > Routing > Virtual Routers > Access List: > New (for trust-vr): Enter the following, then click **OK**:

Access List ID: 1
Sequence No.: 1
IP/Netmask: 224.2.2.1/32
Action: Permit

Select Add Seq No: Enter the following, then click **OK**:

Sequence No.: 2
IP/Netmask: 224.3.3.1/32
Action: Permit

Network > Routing > Virtual Router (trust-vr) > Edit > Edit PIM Instance > RP Candidate > Edit (Trust Zone): Select the following, then click **OK**.

Interface: ethernet1 (select)
Access List: 1 (select)
Priority: 200

CLI

```
set interface ethernet1 protocol pim
set interface ethernet1 protocol pim enable
set vrouter trust-vr access-list 1 permit ip 224.2.2.1/32 1
set vrouter trust-vr access-list 1 permit ip 224.3.3.1/32 2
set vrouter trust-vr protocol pim zone trust rp candidate interface ethernet1 mgroup-list
 1 priority 200
save
```

Security Considerations

When you run PIM-SM, there are certain options that you can set at the virtual router (VR) level to control traffic to and from the VR. Settings defined at the VR level affect all PIM-SM-enabled interfaces in the VR.

When an interface receives multicast control traffic (IGMP or PIM-SM messages) from another zone, the security device first checks if there is a multicast policy that allows the traffic. If the security device finds a multicast policy that allows the traffic, it checks the virtual router for any PIM-SM options that apply to the traffic. For example, if you configure the virtual router to accept join-prune messages from multicast groups specified in an access list, the security device checks if the traffic is for a

multicast group on the list. If it is, then the device allows the traffic. If it is not, then the device drops the traffic.

Restricting Multicast Groups

You can restrict a VR to forward PIM-SM join-prune messages for a particular set of multicast groups only. You specify the allowed multicast groups in an access list. When you use this feature, the VR drops join-prune messages for groups that are not in the access list.

In this example, you create an access list with ID number 1 that allows the following multicast groups: 224.2.2.1/32 and 224.3.3.1/32. Then you configure the trust-vr to accept join-prune messages from the multicast groups in the access list.

WebUI

Network > Routing > Virtual Routers > Access List: > New (for trust-vr): Enter the following, then click **OK**:

Access List ID: 1
Sequence No: 1
IP/Netmask: 224.2.2.1/32
Action: Permit

Select Add Seq No: Enter the following, then click **OK**:

Sequence No: 2
IP/Netmask: 224.3.3.1/32
Action: Permit

Network > Routing > Virtual Router (trust-vr) > Edit > Edit PIM Instance: Select the following, then click **Apply**:

Access Group: 1 (select)

CLI

```
set vrouter trust-vr access-list 1 permit ip 224.2.2.1/32 1
set vrouter trust-vr access-list 1 permit ip 224.3.3.1/32 2
set vrouter trust-vr protocol pim accept-group 1
save
```

Restricting Multicast Sources

You can control the sources from which a multicast group receives data. You identify the allowed source(s) in an access list, then link the access list to multicast groups. This prevents unauthorized sources from sending data into your network. When you use this feature, the security device drops multicast data from sources not in the list. If the virtual router is the rendezvous point in the zone, it checks the access list before accepting a register message from a source. The security device drops register messages that are not from an allowed source.

In this example, you first create an access list with ID number 5 that specifies the allowed source, 1.1.1.1/32. Then you configure the trust-vr to accept multicast data for the multicast group 224.4.4.1/32 from the source specified in the access list.

WebUI

Network > Routing > Virtual Routers > Access List: > New (for trust-vr): Enter the following, then click **OK**:

Access List ID: 5
Sequence No: 1
IP/Netmask: 1.1.1.1/32
Action: Permit

Network > Routing > Virtual Router (trust-vr) > Edit > Edit PIM Instance > MGroup: Select the following, then click **Add**:

MGroup: 224.4.4.1/32
Accept Source: 5 (select)

CLI

```
set vrouter trust-vr access-list 5 permit ip 1.1.1.1/32 1
set vrouter trust-vr protocol pim mgroup 224.4.4.1/32 accept-source 5
save
```

Restricting Rendezvous Points

You can control which rendezvous points (RPs) are mapped to a multicast group. You identify the allowed RP(s) in an access list, then link the access list to the multicast groups. When the virtual router (VR) receives a bootstrap message for a particular group, it checks its list of allowed RPs for that group. If it does not find a match, then it does not select an RP for the multicast group.

In this example, you create an access list with ID number 6 that specifies the allowed RP, 2.1.1.1/32. Then you configure the trust-vr to accept the RPs in the access list for the multicast group, 224.4.4.1/32.

WebUI

Network > Routing > Virtual Routers > Access List: > New (for trust-vr): Enter the following, then click **OK**:

Access List ID: 6
Sequence No: 1
IP/Netmask: 2.1.1.1/32
Action: Permit

Network > Routing > Virtual Router (trust-vr) > Edit > Edit PIM Instance > MGroup: Select the following, then click **Add**:

MGroup: 224.4.4.1/32
Accept RP: 6 (select)

CLI

```
set vrouter trust-vr access-list 6 permit ip 2.1.1.1/32 1
set vrouter trust-vr protocol pim mgroup 224.4.4.1/32 accept-rp 6
save
```

PIM-SM Interface Parameters

You can change certain defaults for each interface on which you enable PIM-SM. When you set parameters on this level, it affects only the interface that you specify.

Table 102 on page 1447 describes the PIM-SM interface parameters and their defaults.

Table 102: PIM-SIM Parameters

PIM-SIM Interface Parameters	Description	Default Value
Neighbor policy	Controls neighbor adjacencies. For additional information, see "Defining a Neighbor Policy" on page 1447.	Disabled
Hello interval	Specifies the interval at which the interface sends hello messages to its neighboring routers.	30 seconds
Designates router priority	Specifies the priority of the interface for the designated router election.	1
Join-Prune interval	Specifies the interval, in seconds, at which the interface sends join-prune messages.	60 seconds
Bootstrap border	Specifies that the interface is a bootstrap border. For additional information, see "Defining a Bootstrap Border" on page 1448.	Disabled

Defining a Neighbor Policy

You can control the neighbors with which an interface can form an adjacency. PIM-SM routers periodically send hello messages to announce themselves as PIM-SM routers. If you use this feature, the interface checks its list of allowed or disallowed neighbors and forms adjacencies with those that are allowed.

In this example, you create an access list that specifies the following:

- ID number is 1.
- The first statement permits 2.1.1.1/24.
- The second statement permits 2.1.1.3/24.

Then you specify that ethernet 1 can form an adjacency with the neighbors in the access list.

WebUI

Network > Routing > Virtual Routers > Access List: > New (for trust-vr): Enter the following, then click **OK**:

Access List ID: 1
Sequence No: 1
IP/Netmask: 2.1.1.1/24
Action: Permit

Select Add Seq No: Enter the following and click **OK**:

Sequence No: 2
IP/Netmask: 2.1.1.3/24
Action: Permit

Network > Interfaces > Edit (for ethernet1) > PIM: Enter the following, then click **Apply**:

Accepted Neighbors: 1

CLI

```
set vrouter trust-vr access-list 1 permit ip 2.1.1.1/24 1
set vrouter trust-vr access-list 1 permit ip 2.1.1.3/24 2
set interface ethernet1 protocol pim neighbor-policy 1
save
```

Defining a Bootstrap Border

An interface that is a bootstrap (BSR) border receives and processes BSR messages, but it does not forward these messages to other interfaces even if there is a multicast group policy allowing BSR messages between zones. This ensures that the RP-to-group mappings always stay within a zone.

In this example, you configure ethernet1 as a bootstrap border.

WebUI

Network > Interfaces > Edit (for ethernet1) > PIM: Select **Bootstrap Border**, then click **Apply**.

CLI

```
set interface ethernet1 protocol pim boot-strap-border
save
```

Configuring a Proxy Rendezvous Point

A PIM-SM domain is a group of PIM-SM routers that have the same rendezvous point (RP)-group mappings. In a PIM-SM domain with dynamic RP-group mappings, PIM-SM routers in a domain listen to messages from the same bootstrap router (BSR) to select their RP-group mappings. In a PIM-SM domain with static RP-group mappings, you must configure the static RP on each router in the domain. (For information about RP-group mappings, see "Configuring Rendezvous Points" on page 1442.)

On Juniper Networks security devices, interfaces bound to a Layer-3 zone can run either in NAT mode or in route mode. To run PIM-SM on a device with interfaces operating in different modes, each zone must be in a different PIM-SM domain. For example, if interfaces in the Trust zone are in NAT mode and interfaces in the Untrust zone are in route mode, each zone must be in a different PIM-SM domain. In addition, when configuring PIM-SM across two virtual routers that are in two different virtual systems, each virtual router must be in a separate PIM-SM domain.

You can advertise multicast groups from one PIM-SM domain to another by configuring a proxy RP. A proxy RP acts as the RP for multicast groups learned from other PIM-SM domains either through a static RP or through bootstrap messages allowed by the multicast group policy. It functions as the root of the shared tree for receivers in its domain and it can form the shortest path tree to the source.

You can configure one proxy RP per zone in a virtual router. To configure a proxy RP in a zone, you must configure a candidate-RP (C-RP) in that zone. The security device then advertises the IP address of the C-RP as the IP address of the proxy RP. When you configure the C-RP, do not specify any multicast group in the multicast group list. This enables the C-RP to act as the proxy RP for any group imported from other zones. If you specify multicast groups, then the C-RP functions as the real RP for the groups specified in the list.

If there is a BSR in the zone, the proxy RP advertises itself as the RP for the multicast groups imported from other zones. If there is no BSR in the zone of the proxy RP, then the proxy RP functions as the static RP for the multicast groups imported from other zones. You must then configure the IP address of the C-RP as the static RP on all the other routers in the zone.

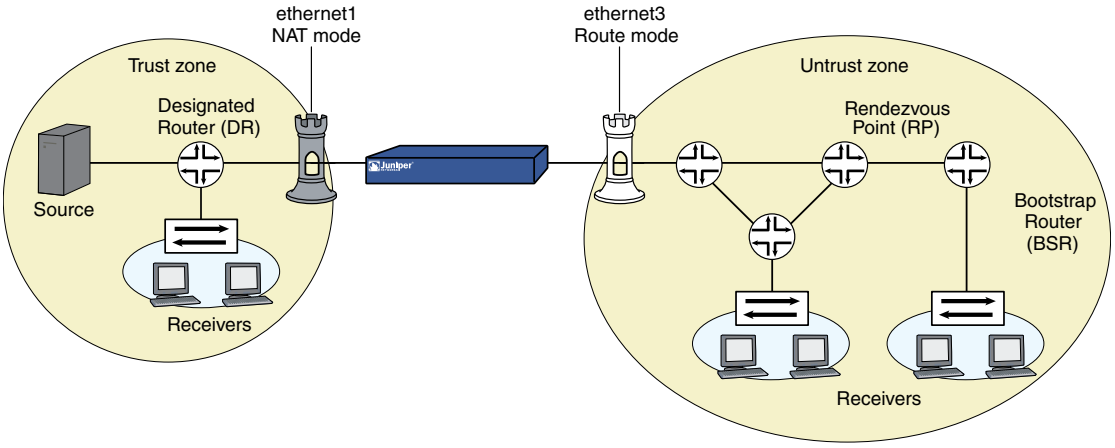
Proxy RP supports the use of Mapped IPs (MIP) for source address translation. A MIP is a direct one-to-one mapping of one IP address to another. You can configure a MIP when you want the security device to translate a private address in a zone whose interfaces are in NAT mode to another address. When a MIP host in the zone of a proxy RP sends a register message, the security device translates the source IP address to the MIP address and sends a new register message to the real RP. When the security device receives a join-prune message for a MIP address, the device maps the MIP to the original source address and sends it to the source.

Proxy RP also supports the translation of multicast group addresses between zones. You can configure a multicast policy that specifies the original multicast group address and the translated multicast group address. When the security device receives a join-prune message on an interface in the zone of the proxy RP, it translates the multicast group, if required, and sends the join message to the real RP.

Consider the following scenario:

- ethernet1 in the Trust zone is in NAT mode, and ethernet3 in the Untrust zone is in route mode.
- There is a MIP for the source in the Trust zone.
- The source in the Trust zone sends multicast traffic to the multicast group 224.4.4.1/32.
- There are receivers in both the Trust and Untrust zones.
- There is a multicast policy that allows PIM-SM messages between the Trust and Untrust zones.
- The Trust zone is configured as the proxy RP.
- The RP and BSR are in the Untrust zone.

Figure 351: Proxy Rendezvous Point Example



Following is the data flow:

1. Source sends data to the multicast group 224.4.4.1/32.
2. The designated router (DR) encapsulates the data and sends Register messages toward the RP.
3. The RP proxy in the Trust zone receives the Register message, and changes the original source IP address to the IP address of the MIP. It then forwards the message toward the RP for the multicast group.
4. The proxy RP sends (*, G) joins to the real RP.
5. Receivers in the Trust zone send join messages to the proxy RP.
6. Proxy RP sends multicast packets to receivers in the Trust zone.

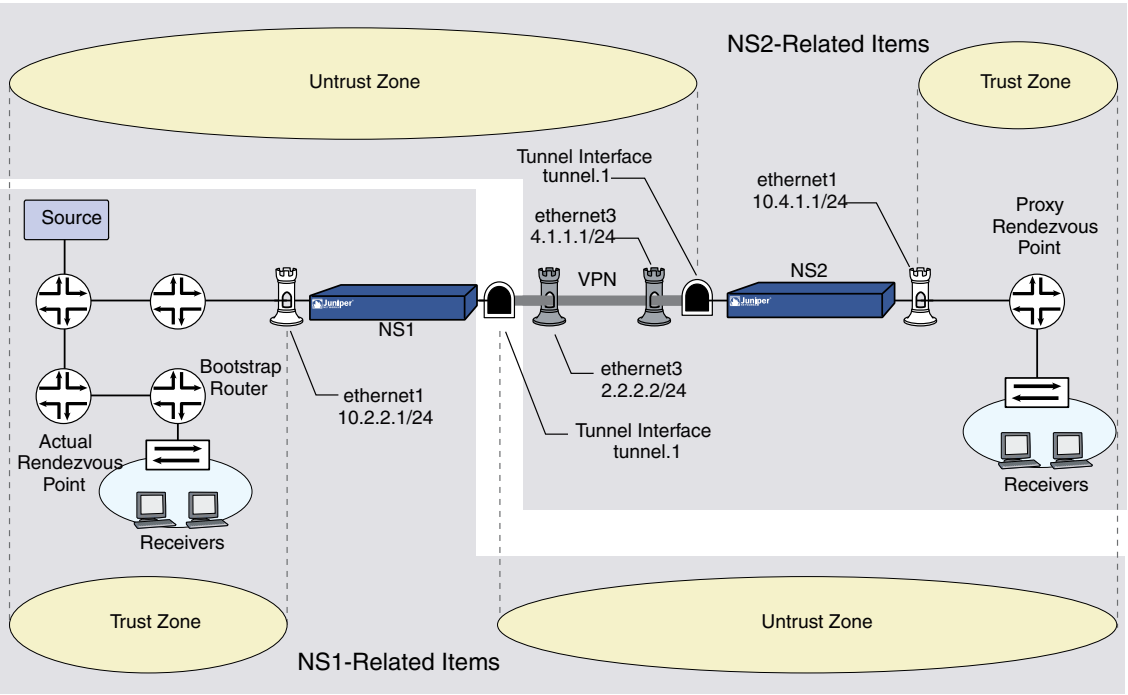
To configure a proxy RP, you must do the following:

1. Create a PIM-SM instance on a specific virtual router.
2. Enable PIM-SM on the appropriate interfaces.

3. Configure a candidate RP in the zone of the proxy RP.
4. Configure the proxy RP.

In this example, the security devices NS1 and NS2 are connected through a VPN tunnel. Both devices are running the dynamic routing protocol, BGP. You configure PIM-SM on ethernet1 and tunnel.1 on NS1 and on NS2. Then, on NS2, you configure ethernet1 as a static RP and create a proxy RP in the Trust zone of the trust-vr.

Figure 352: Proxy RP Configuration Example



WebUI (NS1)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.2.2.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 2.2.2.2/24

Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Unnumbered: (select)
Interface: ethernet3 (trust-vr)

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

Address Name: mgroup1
IP Address/Domain Name:
IP/Netmask: (select), 224.4.4.1/32
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

Address Name: branch
IP Address/Domain Name:
IP/Netmask: (select), 10.4.1.0/24
Zone: Untrust

3. **PIM-SM**

Network > Routing > Virtual Router (trust-vr) > Edit > Create PIM Instance: Select **Protocol PIM: Enable**, then click **OK**.

Network > Interfaces > Edit (for ethernet1) > PIM: Enter the following, then click **Apply**:

PIM Instance: (select)
Protocol PIM: Enable (select)

Network > Interfaces > Edit (for tunnel.1) > PIM: Enter the following, then click **Apply**:

PIM Instance: (select)
Protocol PIM: Enable (select)

4. **VPN**

VPN > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**.

Gateway Name: To_Branch
Security Level: Compatible
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 4.1.1.1
Preshared Key: fg2g4h5j
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Compatible
Phase 1 Proposal (For Compatible Security Level): pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)

5. **BGP**

Network > Routing > Virtual Router (trust-vr) > Edit: Enter the following, then click **OK**:

Virtual Router ID: Custom (select)
In the text box, enter 0.0.0.10

Network > Routing > Virtual Router (trust-vr) > Edit: Select **Create BGP Instance**.

AS Number (required): 65000
BGP Enabled: (select)

Network > Routing > Virtual Router (trust-vr) > Edit > Edit BGP Instance > Neighbors: Enter the following, then click **Add**:

AS Number: 65000
Remote IP: 4.1.1.1
Outgoing Interface: ethernet3

Network > Routing > Virtual Router (trust-vr) > Edit > Edit BGP Instance > Neighbors > Configure (for the peer you just added): Select **Peer Enabled** and then click **OK**.

Network > Routing > Virtual Router (trust-vr) > Edit > Edit BGP Instance > Networks: Enter **2.2.2.0/24** in the IP/Netmask field, then click **Add**. Then enter 10.2.2.0/24 in the IP/Netmask field, and click **Add** again.

Network > Interfaces > Edit (for ethernet3) > BGP: Enter the following, then click **Apply**:

Protocol BGP: Enable (select)

6. **Policy**

Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), branch
Destination Address:
  Address Book Entry: (select), mgroup1
Service: any
Action: Permit

7. **Multicast Policy**

MCast Policies (From: Trust, To: Untrust) > New: Enter the following and click **OK**:

MGroup Address: IP/Netmask (select) 224.4.4.1/32
Bidirectional: (select)

PIM Message: (select)
BSR Static IP: (select)
Join/Prune: (select)

WebUI (NS2)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.4.1.1/24
   Select **NAT**, then click **Apply**.

   > IGMP: Enter the following, then click **Apply**:

   IGMP Mode: Router
   Protocol IGMP: Enable (select)

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 4.1.1.1/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Unnumbered: (select)
      Interface: ethernet3 (trust-vr)

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: mgroup1
   IP Address/Domain Name:
      IP/Netmask: (select), 224.4.4.1/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: corp
   IP Address/Domain Name:
      IP/Netmask: (select), 2.2.2.0/24
   Zone: Untrust

3. **PIM-SM**

Network > Routing > Virtual Router (trust-vr) > Edit > Create PIM Instance: Select **Protocol PIM: Enable**, then click **OK**.

Network > Interfaces > Edit (for ethernet1) > PIM: Enter the following, then click **Apply**:

PIM Instance: (select)
Protocol PIM: Enable (select)

Network > Interfaces > Edit (for tunnel.1) > PIM: Enter the following, then click **Apply**:

PIM Instance: (select)
Protocol PIM: Enable (select)

Network > Routing > Virtual Router (trust-vr) > Edit > Edit PIM Instance > RP Address > New: Select the following, then click **OK**:

Zone: Trust (select)
Address:10.4.1.1/24

4. **VPN**

VPN > AutoKey Advanced > Gateway > New: Enter the following, then click OK:

Gateway Name: To_Corp
Security Level: Compatible
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 2.2.2.2
Preshared Key: fg2g4h5j
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Compatible
Phase 1 Proposal (For Compatible Security Level): pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)

5. **BGP**

Network > Routing > Virtual Router (trust-vr) > Edit: Enter the following, then click **OK**:

Virtual Router ID: Custom (select)
In the text box, enter 0.0.0.10

Network > Routing > Virtual Router (trust-vr) > Edit: Select **Create BGP Instance**.

AS Number (required): 65000
BGP Enabled: (select)

Network > Routing > Virtual Router (trust-vr) > Edit > Edit BGP Instance > Neighbors: Enter the following, then click **Add**:

AS Number: 65000
Remote IP: 2.2.2.2
Outgoing Interface: ethernet3

Network > Routing > Virtual Router (trust-vr) > Edit > Edit BGP Instance > Neighbors > Configure (for the peer you just added): Select **Peer Enabled** and then click **OK**.

Network > Routing > Virtual Router (trust-vr) > Edit > Edit BGP Instance > Networks:

In the IP/Netmask field, enter **4.1.1.0/24**, then click **Add**.

In the IP/Netmask field, enter **10.4.1.0/24**, then click **Add**.

Network > Interfaces > Edit (for ethernet3) > BGP: Select **Protocol BGP: Enable**, then click **Apply**.

6. **Policy**

   Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), corp
   Destination Address:
   Address Book Entry: (select), mgroup1
   Service: any
   Action: Permit

7. **Multicast Policy**

   MCast Policies (From: Trust, To: Untrust) > New: Enter the following and click **OK**:

   MGroup Address: IP/Netmask (select) 224.4.4.1/32
   Bidirectional: (select)
   PIM Message: (select)
   BSR Static IP: (select)
   Join/Prune: (select)

CLI (NS1)

1. **Interfaces**

   ```
   Set interface ethernet1 zone trust
   set interface ethernet1 ip 10.2.2.1/24
   set interface ethernet3 zone untrust
   set interface ethernet3 ip 2.2.2.2/24
   set interface tunnel.1 zone untrust
   set interface tunnel.1 ip unnumbered interface ethernet3
   ```

2. **Addresses**

   ```
   set address trust mgroup1 224.4.4.1/32
   set address untrust branch 10.4.1.0/24
   ```

3. **PIM-SM**

```
set vrouter trust-vr
set vrouter trust-vr protocol pim enable
set interface ethernet1 protocol pim
set interface ethernet1 protocol pim enable
set interface tunnel.1 protocol pim
set interface tunnel.1 protocol pim enable
```

4. **VPN Tunnel**

```
set ike gateway To_Branch address 4.1.1.1 main outgoing-interface ethernet3
preshare fg2g4h5j proposal pre-g2-3des-sha
set vpn Corp_Branch gateway To-Branch3 sec-level compatible
set vpn Corp_Branch bind interface tunnel.1
set vpn Corp_Branch proxy-id local-ip 10.2.2.0/24 remote-ip 10.4.1.0/24
```

5. **BGP**

```
set vrouter trust-vr router-id 10
set vrouter trust-vr protocol bgp 6500
set vrouter trust-vr protocol bgp enable
set vrouter trust-vr protocol bgp neighbor 4.1.1.1
set vrouter trust-vr protocol bgp network 2.2.2.0/24
set vrouter trust-vr protocol bgp network 10.2.2.0/24
set interface ethernet3 protocol bgp enable
set interface ethernet3 protocol bgp neighbor 4.1.1.1
```

6. **Policy**

```
set policy name To-Branch from untrust to trust branch any any permit
```

7. **Multicast Policy**

```
set multicast-group-policy from trust mgroup 224.4.4.1/32 any to untrust
pim-message bsr-static-rp join bi-directional
save
```

CLI (NS2)

1. **Interfaces**

```
set interface ethernet 1 zone trust
set interface ethernet 1 ip 10.4.1.1/24
set interface ethernet 1 protocol igmp router
set interface ethernet 1 protocol igmp enable
set interface ethernet 3 zone untrust
set interface ethernet 3 ip 4.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
```

2. **Addresses**

```
set address trust mgroup1 224.4.4.1/32
set address untrust corp 2.2.2.0/24
```

3. **PIM-SM**

```
set vrouter trust protocol pim
set interface ethernet1 protocol pim
set interface ethernet1 protocol pim enable
set interface tunnel.1 protocol pim
set interface tunnel.1 protocol pim enable
set vrouter trust protocol pim zone trust rp proxy
set vrouter trust protocol pim zone trust rp candidate interface ethernet1
set vrouter trust protocol pim enable
```

4. **VPN Tunnel**

```
set ike gateway To_Corp address 2.2.2.2 main outgoing-interface ethernet3
preshare fg2g4h5j proposal pre-g2-3des-sha
set vpn Branch_Corp gateway To_Corp sec-level compatible
set vpn Branch_Corp bind interface tunnel.1
set vpn Branch_Corp proxy-id local-ip 10.4.1.0/24 remote-ip 10.2.2.0/24
```

5. **BGP**

```
set vrouter trust-vr router-id 10
set vrouter trust-vr protocol bgp 6500
set vrouter trust-vr protocol bgp enable
set vrouter trust-vr protocol bgp neighbor 2.2.2.2
set vrouter trust-vr protocol bgp network 4.1.1.0/24
set vrouter trust-vr protocol bgp network 10.4.1.0/24
set interface ethernet3 protocol bgp neighbor 2.2.2.2
```

6. **Policy**

```
set policy name To-Corp from untrust to trust corp any any permit
```

7. **Multicast Policy**

```
set multicast-group-policy from trust mgroup 224.4.4.1/32 any to untrust
pim-message bsr-static-rp join bi-directional
save
```

PIM-SM and IGMPv3

PIM-SM can operate with interfaces running Internet Group Management Protocol (IGMP) version 1, 2 or 3. When you run PIM-SM with interfaces running IGMPv1 or v2, hosts receiving data for a multicast group can receive data from any source that sends data to the multicast group. IGMPv1 and v2 membership reports only indicate which multicast groups the hosts wants to join. They do not contain information about the sources of the multicast traffic. When PIM-SM receives IGMPv1 and v2 membership reports, it creates (*,G) entries in the multicast route table, allowing any source to send to the multicast group. This is called the any-source-multicast model (ASM), where receivers join a multicast group, with no knowledge of the source that sends data to the group. The network maintains information about the source.

Hosts running IGMPv3 indicate which multicast groups they want to join and the sources from which they expect to receive multicast traffic. The IGMPv3 membership

reports contain the multicast group address, the filter-mode, which is either include or exclude, and a list of sources.

If the filter-mode is include, then receivers accept multicast traffic only from the addresses in the source list. When PIM-SM receives an IGMPv3 membership report with a source list and a filter mode of include, it creates (S,G) entries in the multicast route table for all sources in the source list.

If the filter mode is exclude, then receivers do not accept multicast traffic from the sources in the list; they accept multicast traffic from all other sources. When PIM-SM receives an IGMPv3 membership report with source list and a filter mode of exclude, then it creates a (*,G) for the group and sends a prune message for sources in the source list. In this case, you might need to configure a rendezvous point if the receivers do not know the address of the source.

Chapter 41

ICMP Router Discovery Protocol

This chapter explains Internet Control Messages Protocol (ICMP) Router Discovery Protocol as defined in RFC 1256. It contains the following sections:

Overview

ICMP Router Discovery Protocol (IRDP) is an ICMP message exchange between a host and a router. The security device is the router and advertises the IP address of a specified interface periodically or on demand. If the host is configured to listen, you can configure the security device to send periodic advertisements. If the host explicitly sends router solicitations, you can configure the security device to respond on demand.

NOTE: IRDP is not available on all platforms. Check your datasheet to see if this feature is available on your security device.

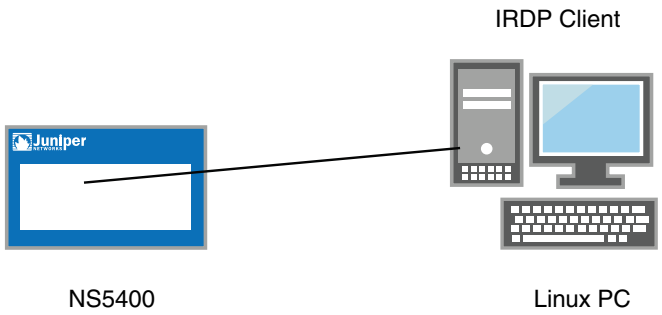
ScreenOS supports IRDP on a per-interface basis. You must assign an IP address before IRDP becomes available on that interface. By default, this feature is disabled. You can configure this feature in a high availability (HA) environment using NetScreen Redundancy Protocol (NSRP).

Beginning with the ScreenOS 6.3.0 release, IRDP support is available on all platforms; however, IRDP support is available only on an Ethernet interface with an IP address. Support for the following IRDP features is available on a logical interface when the mapped hardware type is Ethernet:

- Aggregate interface
- Redundant interface
- Tunnel interface
- Sub-interface
- NSRP VSI interface

Of the router and host modes of implementation described in RFC 1256, ScreenOS supports only the router mode implementation for IRDP. Support for unicast advertisement is not available.

The following illustration shows an example of how this can be deployed:



In this example:

- NS 5400 is used as IRDP router.
- The Linux PC is used as IRDP client, which can send IRDP Router Solicit packets.
- On NS 5400:

```
set interface ethernet2/1 protocol irdp enable
set interface ethernet2/1 protocol irdp 2.2.2.8 advertise
set interface ethernet2/1 protocol irdp broadcast-address
set interface ethernet2/1 protocol irdp init-adv-packet 4
set interface ethernet2/1 protocol irdp init-adv-interval 31
```

Configuring ICMP Router Discovery Protocol

You can enable and disable IRDP and configure or view IRDP settings with the WebUI or the CLI.

Enabling ICMP Router Discovery Protocol

When you enable IRDP on an interface, ScreenOS initiates an immediate IRDP advertisement to the network. For information about configuring an interface, see “Interfaces” on page 51.

In the following example, you configure IRDP for the Trust interface.

WebUI

Network > Interfaces (edit) > IRDP: Select the IRDP Enable check box.

CLI

```
set interface trust protocol irdp enable
```

Configuring ICMP Router Discovery Protocol from the WebUI

To configure IRDP from the WebUI:

Network > Interface > Edit > IRDP: Enter the desired settings, then click **OK**.

Table 103 on page 1463 lists the IRDP parameters, default values, and available settings.

Table 103: IRDP WebUI Settings

Parameter	Default Settings	Alternative Settings
IPv4 address	■ Primary and secondary IP addresses-advertised ■ Management and webauth IP addresses-not advertised	Advertise—you can add a preference value (-1through 2147483647)
Broadcast-address	Disabled	Enabled
Init Advertise Interval	16 seconds	1 through 32 seconds
Init Advertise Packet	3	1 through 5
Lifetime	three times the Max Advertise Interval value	Max Advertise Interval value through 9000 seconds
Max Advertise Interval	600 seconds	4 through 1800 seconds
Min Advertise Interval	75% of the Max Advertise Interval value	3 through Max Advertise Interval value
Response Delay	2 seconds	0 through 4 seconds

Configuring ICMP Router Discovery Protocol from the CLI

You can configure various IRDP parameters from the CLI to control how advertisement and solicitation behavior occurs.

Advertising an Interface

By default, ScreenOS advertises the primary IP address of the security device; however, the IP address is not advertised for WebAuth and management.

You can also associate a preference status for a security device. The preference status is a number from -1 through 2147483647. Higher numbers have greater preference. You can assign different preference values for different security devices. For example, you can assign a higher preference number for the security device that primarily handles network traffic. For a backup security device, you can assign a lower preference number.

To advertise the Untrust interface with an IP address of 10.10.10.10 with a preference of 250, enter the following commands:

```
set interface untrust protocol irdp 10.10.10.10 advertise
set interface untrust protocol irdp 10.10.10.10 preference 250
save
```

Broadcasting the Address

By default, except for the initial broadcast advertisement message when IRDP is enabled, the interface does not send broadcast advertisements. The default address is 224.0.0.1 (all hosts on the network).

To configure the default broadcast address for the Untrust interface, enter the following command:

```
set interface untrust protocol irdp broadcast-address
```

Setting a Maximum Advertisement Interval

The maximum advertisement interval is the maximum number of seconds that passes between ICMP advertisements. This interval can be a value from 4 through 1800 seconds. The default value is 600 seconds.

To set the maximum advertisement interval to be 800 seconds for the Untrust interface, enter the following commands:

```
set interface untrust protocol irdp max-adv-interval 800
save
```

Setting a Minimum Advertisement Interval

The minimum advertisement interval is the lower limit (in seconds) of the advertisement period, which is calculated to be 75 percent of the maximum advertisement value. The value range for the minimum advertisement interval is 3 through the maximum advertisement value. When you change the maximum advertisement value, the minimum advertisement interval value is automatically calculated.

When you set the maximum advertisement interval to 800 seconds, ScreenOS automatically recalculates the minimum advertisement interval to be 600 seconds.

To set the minimum advertisement interval value to 500 seconds for the Untrust interface, enter the following commands:

```
set interface untrust protocol irdp min-adv-interval 500
save
```

Setting an Advertisement Lifetime Value

By default, the advertisement lifetime value is three times the maximum advertisement interval. You can set the advertisement lifetime value. The value range

is the maximum advertisement interval value (4 through 1800 seconds) through 9000 seconds.

To set the advertisement lifetime value to 5000 seconds for the Untrust interface, enter the following commands:

```
set interface untrust protocol untrust lifetime 5000
save
```

Setting a Response Delay

By default, the security device waits 0 to 2 seconds before responding to a client-solicitation request. You can change the response delay setting to no delay (0 seconds) to up to a four-second response delay. For example, if you configure the response delay to 4 seconds, the security device waits 0 to 4 seconds before responding.

To set a delay the response delay value to 4 seconds to the Untrust interface, enter the following commands:

```
set interface untrust protocol irdp response-delay 4
save
```

Setting an Initial Advertisement Interval

The Initial Advertise Interval is the number of seconds during the IRDP startup period allocated for advertisement. By default, this interval is 16 seconds. The value range for this interval is 1 through 32 seconds.

To set the Initial Advertise Interval to 24 seconds for the Untrust interface, enter the following commands:

```
set interface untrust protocol irdp init-adv-interval
save
```

Setting a Number of Initial Advertisement Packets

By default. the security device sends out three advertisement packets during the specified startup period. You can change this setting to be 1 through 5.

To change the number of initial packets sent to 5, enter the following commands:

```
set interface untrust protocol irdp init-adv-packet 5
save
```

Configuration Example

The following example illustrates the IRDP configuration:

```
ssg5-serial-> get config | in irdp
set interface ethernet0/1 protocol irdp enable
set interface ethernet0/1 protocol irdp broadcast-address
set interface ethernet0/1 protocol irdp max-adv-interval 1000
```

```
set interface ethernet0/1 protocol irdp min-adv-interval 100
set interface ethernet0/1 protocol irdp init-adv-packet 5
set interface ethernet0/1 protocol irdp init-adv-interval 10
set interface ethernet0/1 protocol irdp response-delay 4
```

Disabling IRDP

You can disable an interface from running IRDP; however, when you do so, ScreenOS deletes all related memory from the original configuration.

To disable the Trust interface from running IRDP, enter the following command:

```
unset interface trust protocol irdp enable
```

Viewing IRDP Settings

You can view IRDP information from the WebUI or the CLI.

To view IRDP settings, enter the **get irdp** or **get irdp interface** *interface_name* commands.

WebUI

Network > Interface > Edit > IRDP: You can view whether IRDP is enabled.

CLI 1

```
device> get irdp
```

```
Total 1 IRDP instance enabled
-----------------------------------------------------------------
interface    dest-addr       lifetime adv-interval Next-Adv(sec)
-----------------------------------------------------------------
untrust      255.255.255.255     6000 450 to 600   358
```

CLI 2

```
device-> get irdp interface untrust
```

```
IRDP enabled on untrust:
advertisement interval       :   450 to 600 sec
next advertisement in        :   299 sec
advertisement lifetime       :   6000 sec
advertisement address        :   255.255.255.255
initial advertise interval   :   16 sec
initial advertise packet     :   3
solicitation response delay  :   4 sec
10.100.37.90                 :   pref 250, advertise YES
```

Part 8
Address Translation

Address Translation focuses on the various methods available in ScreenOS to perform address translation. This guide contains the following chapters:

- "Address Translation" on page 1469 gives an overview of the various translation options, which are covered in detail in subsequent chapters.
- "Source Network Address Translation" on page 1481 describes NAT-src, the translation of the source IP address in a packet header, with and without Port Address Translation (PAT).
- "Destination Network Address Translation" on page 1499 describes NAT-dst, the translation of the destination IP address in a packet header, with and without destination port address mapping. This section also includes information about the packet flow when doing NAT-src, routing considerations, and address shifting.
- "Mapped and Virtual Addresses" on page 1535 describes the mapping of one destination IP address to another based on IP address alone (Mapped IP) or based on destination IP address and destination port number (Virtual IP).

NOTE: For coverage of interface-based Source Network Address Translation—referred to simply as *NAT*—see "NAT Mode" on page 116.

Chapter 42
Address Translation

ScreenOS provides many methods for performing source and destination IP and port address translation. This chapter describes the various address translation methods available and contains the following sections:

Introduction to Address Translation

ScreenOS provides several mechanisms for applying Network Address Translation (NAT). NAT includes the translation of the Internet Protocol (IP) address in an IP packet header and, optionally, the translation of the port number in the Transmission Control Protocol (TCP) segment or User Datagram Protocol (UDP) datagram header. The translation can involve the source address (and, optionally, the source port number), the destination address (and, optionally, the destination port number), or a combination of translated elements.

Source Network Address Translation

When performing Source Network Address Translation (NAT-src), the security device translates the original source IP address to a different address. The translated address can come from a Dynamic IP (DIP) pool or from the egress interface of the security device. If the security device draws the translated address from a DIP pool, it can do so either arbitrarily or deterministically; that is, it can draw any address from the DIP pool at random, or it can consistently draw a specific address in relation to the original source IP address.

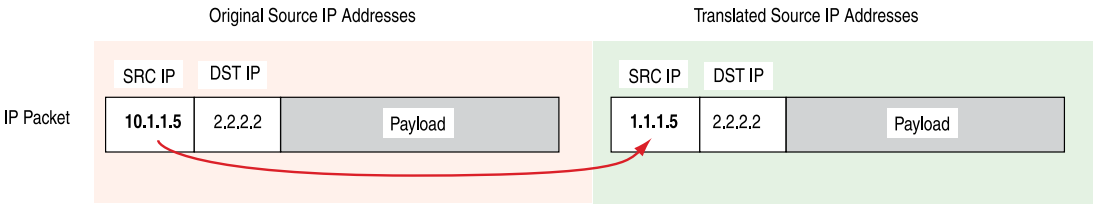
NOTE: Deterministic address translation uses a technique called address shifting, which is explained later in this chapter. For information about address shifting that applies to NAT-src, see "NAT-Src from a DIP Pool with Address Shifting" on page 1492. For information about address shifting that applies to NAT-dst, see "NAT-Src and NAT-Dst in the Same Policy" on page 1522.

If the translated address comes from the egress interface, the security device translates the source IP address in all packets to the IP address of that interface. You can configure the security device to apply NAT-src at either the interface level or at the

policy level. If you configure a policy to apply NAT-src and the ingress interface is in NAT mode, the policy-based NAT-src settings override the interface-based NAT. (This chapter focusses on policy-based NAT-src. For details on interface-based NAT-src—or NAT alone—see *"NAT Mode" on page 116*. For more information about DIP pools, see *"Dynamic IP Pools" on page 177*.)

NOTE: You can use policy-based NAT-src when the ingress interface is in Route or NAT mode. If it is in NAT mode, the policy-level NAT-src parameters supersede the interface-level NAT parameters.

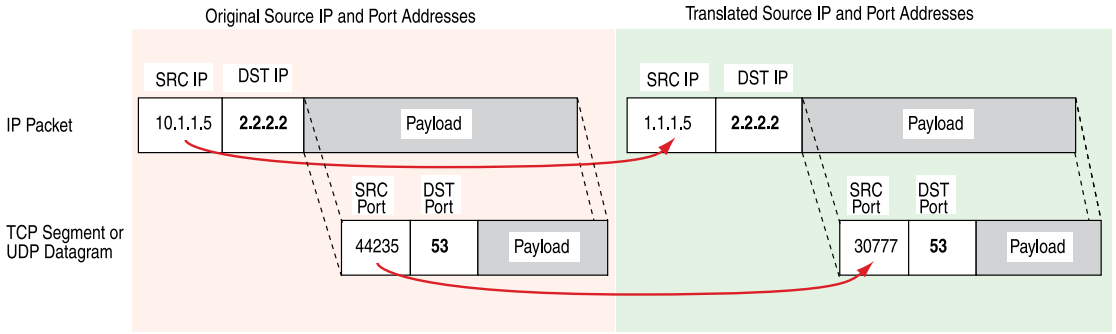
Figure 353: Source IP Address Translation



With policy-based NAT-src, you can optionally choose to have the security device perform Port Address Translation (PAT) on the original source port number. When PAT is enabled, the security device can translate up to 64,500 different IP addresses to a single IP address with up to 64,500 different port numbers. The security device uses the unique, translated port number to maintain session state information for traffic to and from the same, single IP address. For interface-based NAT-src—or just NAT—PAT is enabled automatically. Because the security device translates all original IP addresses to the same translated IP address (that of the egress interface), the security device uses the translated port number to identify each session to which a packet belongs. Similarly, if a DIP pool consists of only one IP address and you want the security device to apply NAT-src to multiple hosts using that address, then PAT is required for the same reason.

NOTE: With PAT enabled, the security device maintains a pool of free port numbers to assign along with addresses from the DIP pool. The figure of up to 64,500 is derived by subtracting 1023, the numbers reserved for the well-known ports, from the maximum number of ports, which is 65,535. Thus, when the security device performs NAT-src with a DIP pool containing a single IP address and PAT is enabled, the security device can translate the original IP addresses of up to 64,500 hosts to a single IP address and translate each original port number to a unique port number.

Figure 354: Source IP and Source Port Address Translation



For custom applications that require a specific source port number to operate properly, performing PAT causes such applications to fail. To provide for such cases, you can disable PAT.

NOTE: For more information about NAT-src, see "Source Network Address Translation" on page 1481.

Destination Network Address Translation

Screen OS offers the following three mechanisms for performing Destination Network Address Translation (NAT-dst):

- **Policy-based NAT-dst**: see Policy-Based NAT-Dst on page 1471
- **MIP**: see Mapped Internet Protocol on page 1474
- **VIP**: see Virtual Internet Protocol on page 1474

All three options translate the original destination IP address in an IP packet header to a different address. WIth policy-based NAT-dst and VIPs, you can optionally enable port mapping.

NOTE: For information about port mapping, see the "Policy-Based NAT-Dst" on page 1471 and "Destination Network Address Translation" on page 1499.

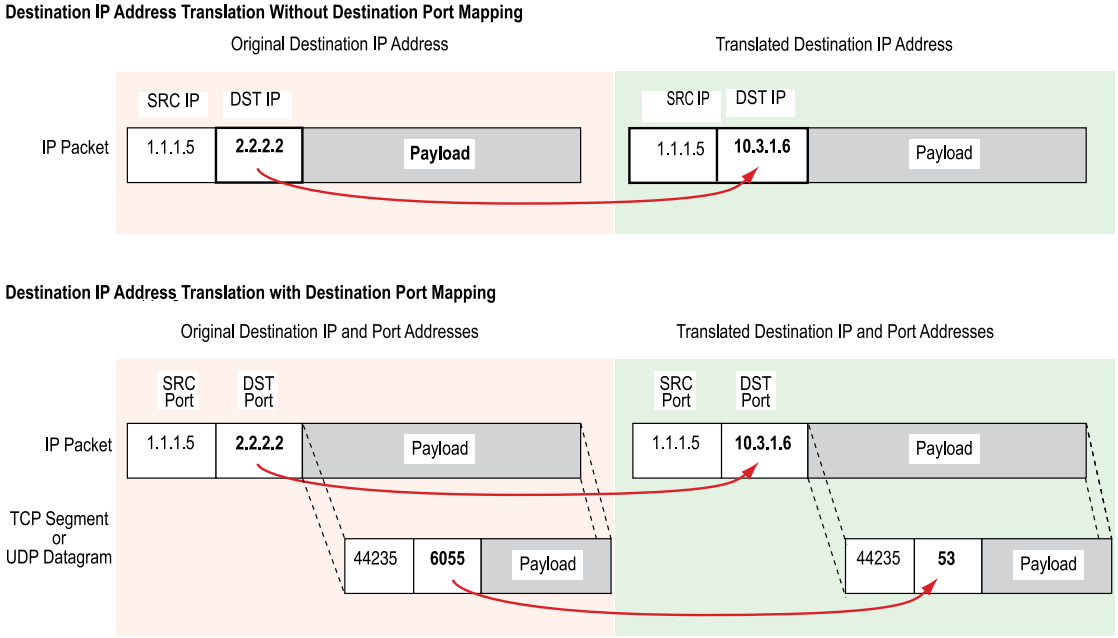
ScreenOS does not support the use of policy-based NAT-dst in combination with MIPs and VIPs. If you have configured a MIP or VIP, the security device applies the MIP or VIP to any traffic to which a policy-based NAT-dst configuration also applies. In other words, MIPs and VIPs disable policy-based NAT-dst if the security device is accidentally configured to apply both to the same traffic.

Policy-Based NAT-Dst

You can configure a policy to translate one destination IP address to another address, one IP address range to a single IP address, or one IP address range to another IP

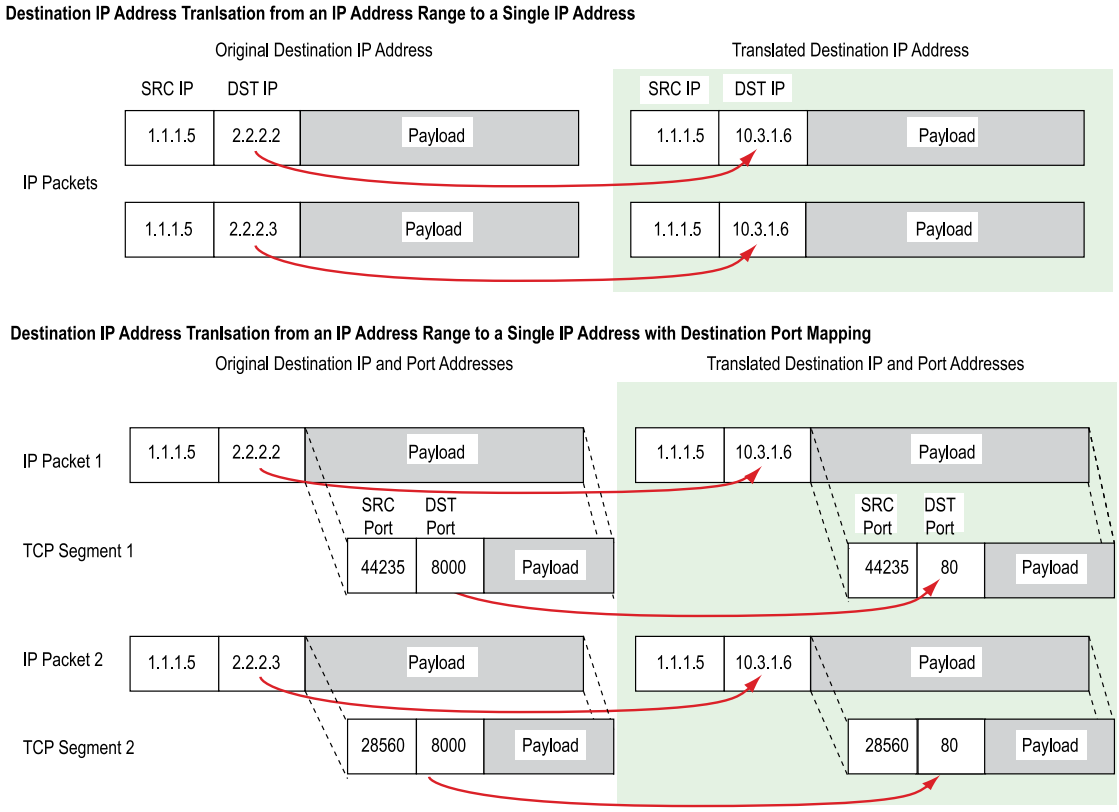
address range. When a single destination IP address translates to another IP address or an IP address range translates to a single IP address, ScreenOS can support NAT-dst with or without port mapping. Port mapping is the deterministic translation of one original destination port number to another specific number, unlike PAT, which translates any original source port number randomly assigned by the initiating host to another number randomly assigned by the security device.

Figure 355: Destination IP Address Translation



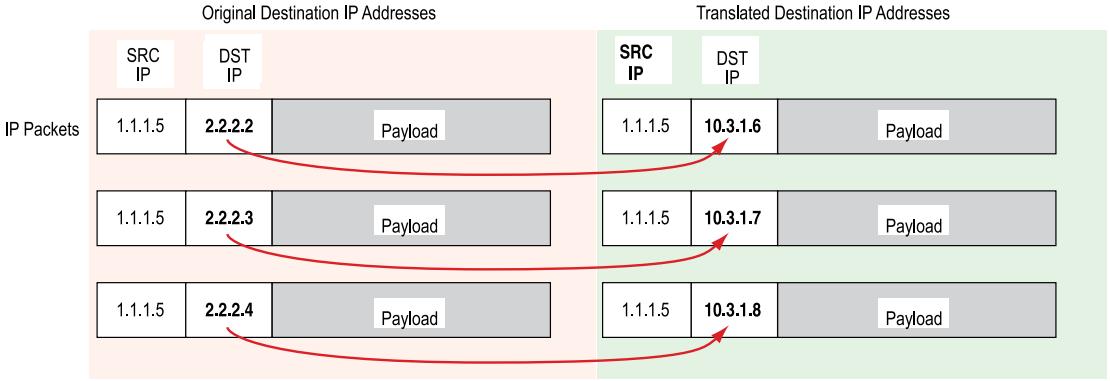
When you configure a policy to perform NAT-dst to translate an address range to a single address, the security device translates any destination IP address from within the user-defined range of original destination addresses to a single address. You can also enable port mapping.

Figure 356: NAT-Dst from an IP Address Range to a Single IP Address



When you configure a policy to perform NAT-dst for an address range, the security device uses address shifting to translate a destination IP address from within a range of original destination addresses to a known address in another range of addresses.

Figure 357: NAT-Dst with Address Shifting



When performing NAT-dst for a range of IP addresses, the security device maintains a mapping of each IP address in one address range to a corresponding IP address in another address range.

NOTE: You can combine NAT-src and NAT-dst within the same policy. Each translation mechanism operates independently and unidirectionally. That is, if you enable NAT-dst on traffic from zone1 to zone2, the security device does not perform NAT-src on traffic originating from zone2 and destined to zone1 unless you specifically configure it to do so. For more information, see "Directional Nature of NAT-Src and NAT-Dst" on page 1478. For more information about NAT-dst, see "Destination Network Address Translation" on page 1499.

Mapped Internet Protocol

A mapped Internet Protocol (MIP) is a mapping of one IP address to another IP address. You define one address in the same subnet as an interface IP address. The other address belongs to the host to which you want to direct traffic. Address translation for a MIP behaves bidirectionally, so that the security device translates the destination IP address in all traffic coming to a MIP to the host IP address and source IP address in all traffic originating from the host IP address to the MIP address. MIPs do not support port mapping. For more information about MIPs, see "Mapped IP Addresses" on page 1535.

Virtual Internet Protocol

A virtual Internet Protocol (VIP) is a mapping of one IP address to another IP address based on the destination port number. A single IP address defined in the same subnet as an interface can host mappings of several services—identified by various destination port numbers—to as many hosts. VIPs also support port mapping. Unlike MIPs, address translation for a VIP behaves unidirectional. The security device translates the destination IP address in all traffic coming to a VIP to a host IP address. The security device does not translate the original source IP address in outbound traffic from a VIP host to that of the VIP address. Instead, the security device applies interface-based or policy-based NAT-src if you have previously configured it. Otherwise, the security device does not perform any NAT-src on traffic originating from a VIP host. For more information about VIPs, see "Virtual IP Addresses" on page 1552.

NOTE: You can define a VIP to be the same as an interface IP address. This ability is convenient when the security device only has one assigned IP address and when the IP address is assigned dynamically.

Whereas the address translation mechanisms for MIPs and VIPs are bidirectional, the capabilities provided by policy-based NAT-src and NAT-dst separate address translation for inbound and outbound traffic, providing better control and security. For example, if you use a MIP to a Web server, whenever that server initiates outbound traffic to get an update or patch, its activity is exposed, which might provide information for a vigilant attacker to exploit. The policy-based address translation methods allow you to define a different address mapping when the Web server receives traffic (using NAT-dst) than when it initiates traffic (using NAT-src). By thus keeping its activities hidden, you can better protect the server from anyone attempting to gather information in preparation for an attack. Policy-based NAT-src and NAT-dst

offer a single approach that can duplicate and surpass the functionality of interface-based MIPs and VIPs.

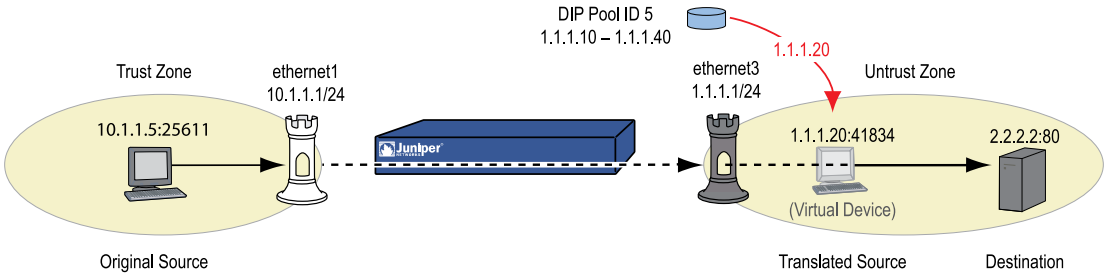
Policy-Based Translation Options

ScreenOS provides the following ways to apply Source Network Address Translation (NAT-src) and Destination Network Address Translation (NAT-dst). Note that you can always combine NAT-src with NAT-dst within the same policy.

Example: NAT-Src from a DIP Pool with PAT

The security device translates the original source IP address to an address drawn from a Dynamic IP (DIP) pool. The security device also applies source Port Address Translation (PAT). For more information, see "NAT-Src from a DIP Pool with PAT Enabled" on page 1484.

Figure 358: NAT-Src with Port Address Translation

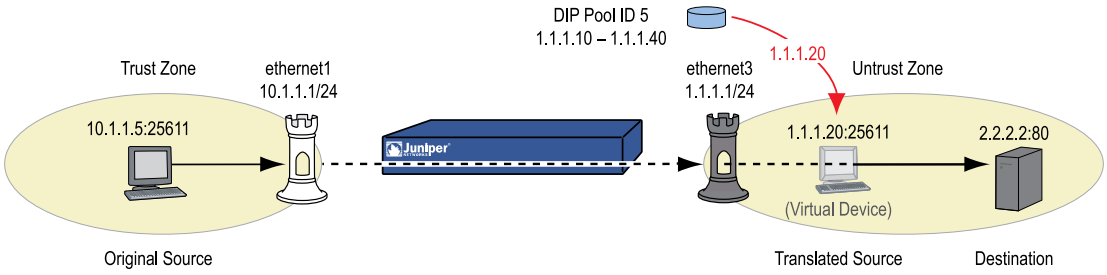


NOTE: In Figure 358 on page 1475 and in subsequent figures, a "virtual device" is used to indicate a translated source or destination address when that address does not belong to an actual device.

Example: NAT-Src From a DIP Pool Without PAT

The security device translates the original source IP address to an address drawn from a DIP pool. The security device does not apply source PAT. For more information, see "NAT-Src from a DIP Pool with PAT Disabled" on page 1490.

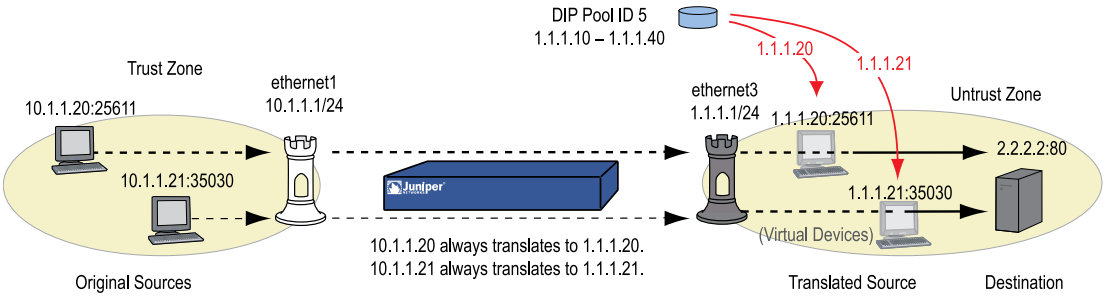
Figure 359: NAT-Src Without Port Address Translation



Example: NAT-Src from a DIP Pool with Address Shifting

The security device translates the original source IP address to an address drawn from a dynamic IP (DIP) pool, consistently mapping each original address to a particular translated address. The security device does not apply source Port Address Translation (PAT). For more information, see "NAT-Src from a DIP Pool with Address Shifting" on page 1492.

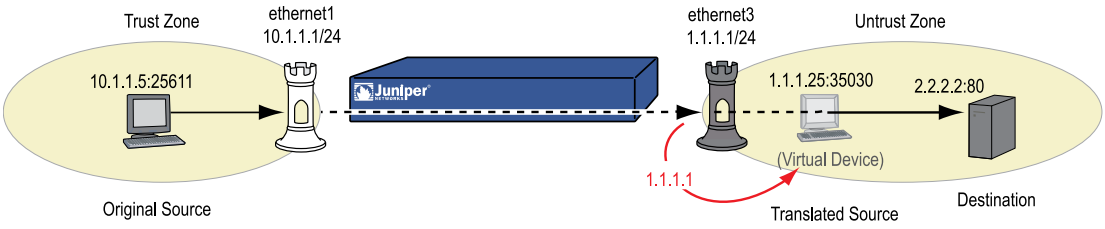
Figure 360: NAT-Src with Address Shifting



Example: NAT-Src from the Egress Interface IP Address

The security device translates the original source IP address to the address of the egress interface. The security device applies source PAT as well. For more information, see "NAT-Src from the Egress Interface IP Address" on page 1496.

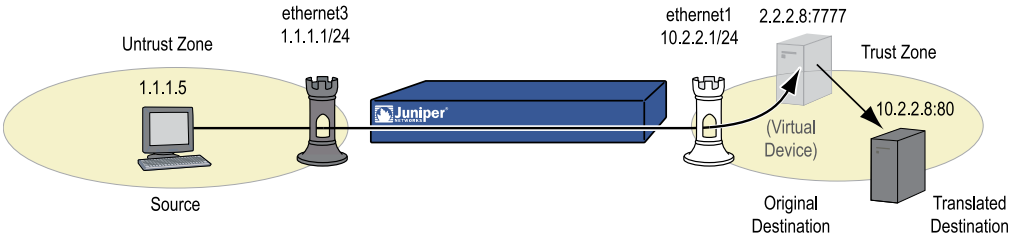
Figure 361: NAT-Src Using the Egress Interface IP Address



Example: NAT-Dst to a Single IP Address with Port Mapping

The security device performs Destination Network Address Translation (NAT-dst) and destination port mapping. For more information, see "NAT-Dst with Port Mapping" on page 1518.

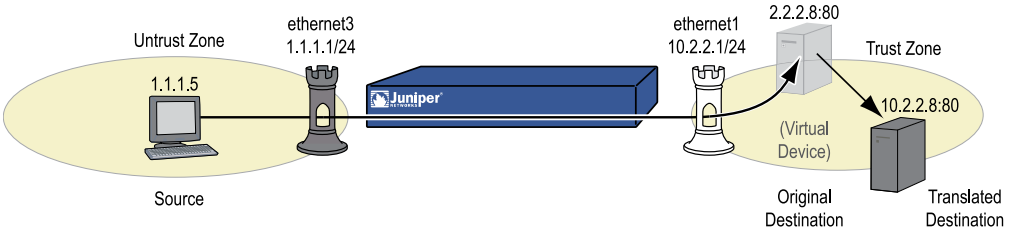
Figure 362: NAT-Dst with Port Mapping



Example: NAT-Dst to a Single IP Address Without Port Mapping

The security device performs NAT-dst but does not change the original destination port number. For more information, see “Destination Network Address Translation” on page 1499.

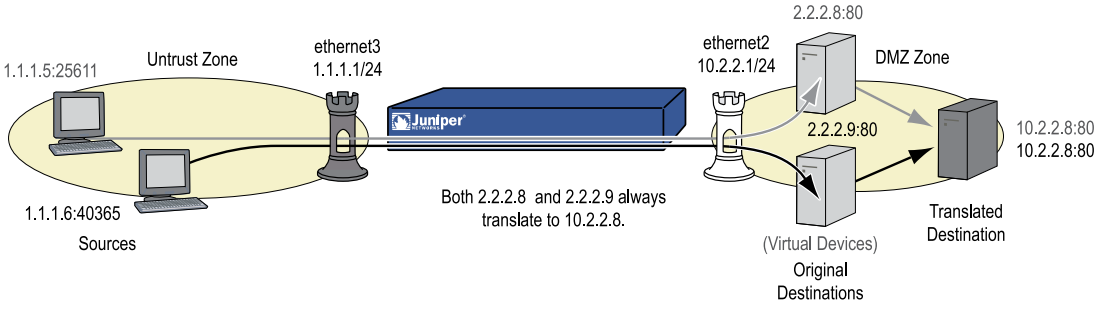
Figure 363: NAT-Dst Without Port Mapping



Example: NAT-Dst from an IP Address Range to a Single IP Address

The security device performs NAT-dst to translate a range of IP addresses to a single IP address. If you also enable port mapping, the security device translates the original destination port number to another number. For more information, see “NAT-Dst—Many-to-One Mapping” on page 1512.

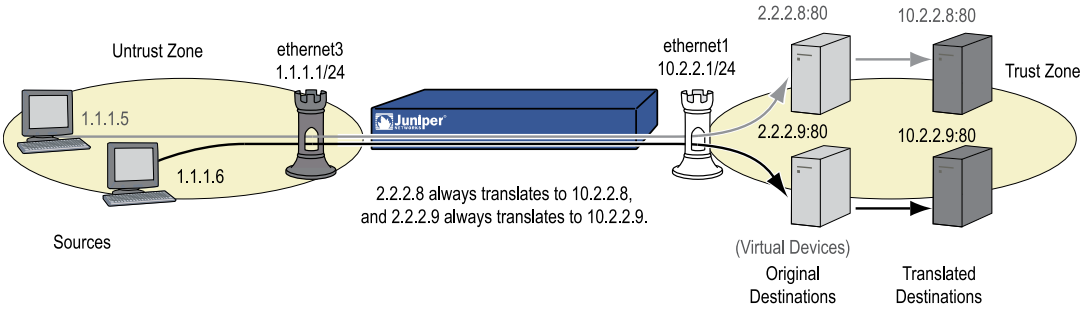
Figure 364: NAT-Dst from an Address Range to a Single IP Address



Example: NAT-Dst Between IP Address Ranges

When you apply NAT-dst for a range of IP addresses, the security device maintains a consistent mapping of an original destination address to a translated address within the specified range using a technique called address shifting. Note that address shifting does not support port mapping. For more information, see “NAT-Dst—Many-to-Many Mapping” on page 1515.

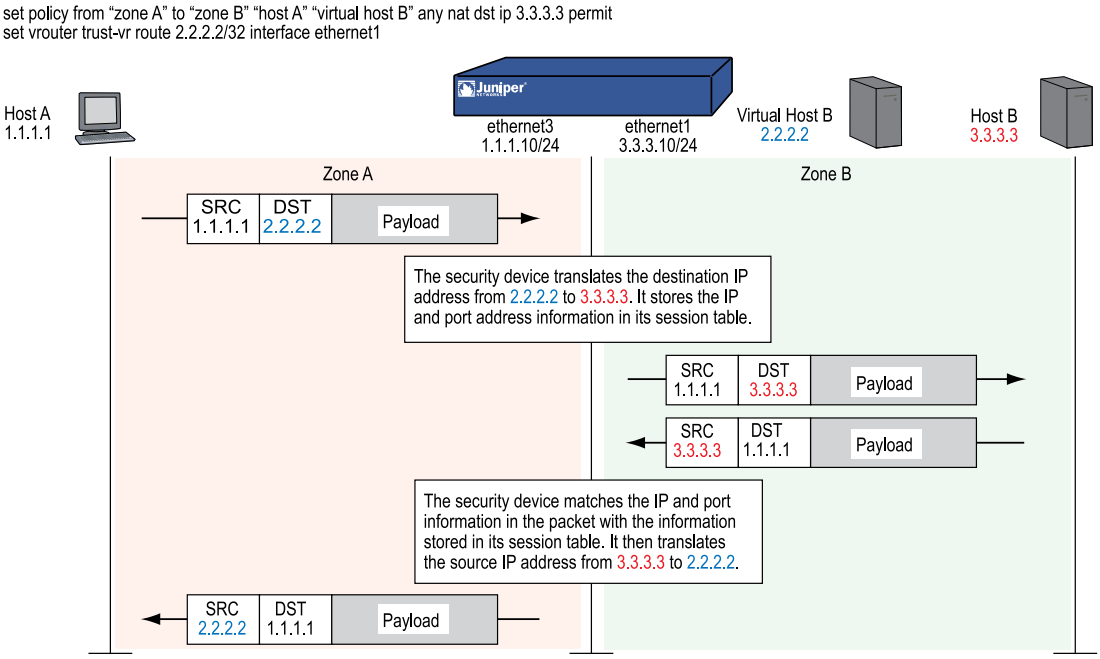
Figure 365: NAT-Dst Between Address Ranges



Directional Nature of NAT-Src and NAT-Dst

The application of NAT-src is separate from that of NAT-dst. You determine their applications on traffic by the direction indicated in a policy. For example, if the security device applies a policy requiring NAT-dst for traffic sent from host A to virtual host B, the security device translates the original destination IP address from 2.2.2.2 to 3.3.3.3. (It also translates the source IP address from 3.3.3.3 to 2.2.2.2 in responding traffic.)

Figure 366: Packet Flow for NAT-Dst

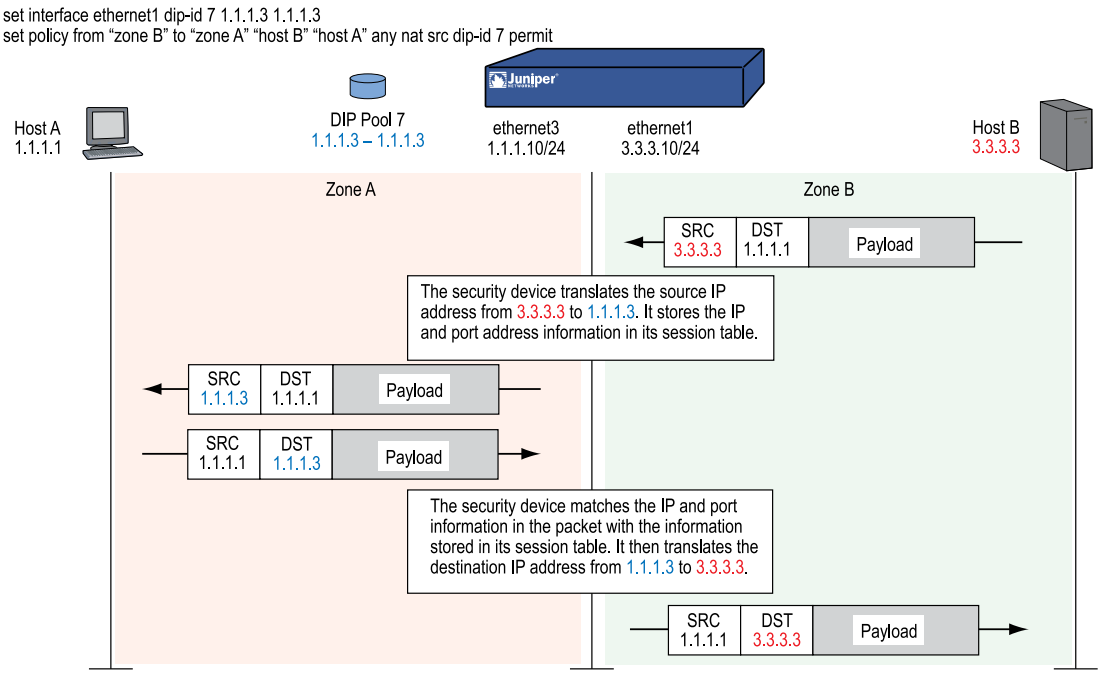


NOTE: You must set a route to 2.2.2.2/32 (virtual host B) so the security device can do a route lookup to determine the destination zone. For more about NAT-dst routing issues, see "Routing for NAT-Dst" on page 1503.

However, if you only create the above policy specifying NAT-dst from host A to host B, the security device does not translate the original source IP address of host B if host B initiates traffic to host A, rather than responding to traffic from host A. For the security device to do translate the source IP address of host B when it initiates traffic to host A, you must configure a second policy from host B to host A specifying NAT-src. (This behavior differs from that of MIPs. See "Mapped IP Addresses" on page 1535.)

NOTE: To retain focus on the IP address translation mechanisms, Port Address Translation (PAT) is not shown. If you specify fixed port numbers for a DIP pool consisting of a single IP address, then only one host can use that pool at a time. The policy above specifies only "host B" as the source address. If "host B" is the only host that uses DIP pool 7, then it is unnecessary to enable PAT.

Figure 367: Packet Flow for Source IP Address Translation



Chapter 43

Source Network Address Translation

ScreenOS provides many methods for performing Source Network Address Translation (NAT-src) and Source Port Address Translation (PAT). This chapter describes the various address translation methods available and contains the following sections:

Introduction to NAT-Src

It is sometimes necessary for the security device to translate the original source IP address in an IP packet header to another address. For example, when hosts with private IP addresses initiate traffic to a public address space, the security device must translate the private source IP address to a public one. Also, when sending traffic from one private address space through a VPN to a site using the same addresses, the security devices at both ends of the tunnel must translate the source and destination IP addresses to mutually neutral addresses.

NOTE: For information about public and private IP addresses, see *"Public IP Addresses" on page 63* and *"Private IP Addresses" on page 64*.

A dynamic IP (DIP) address pool provides the security device with a supply of addresses from which to draw when performing Source Network Address Translation (NAT-src). When a policy requires NAT-src and references a specific DIP pool, the security device draws addresses from that pool when performing the translation.

NOTE: The DIP pool must use addresses within the same subnet as the default interface in the destination zone referenced in the policy. If you want to use a DIP pool with addresses outside the subnet of the destination zone interface, you must define a DIP pool on an extended interface. For more information, see *"Using DIP in a Different Subnet" on page 181*.

The DIP pool can be as small as a single IP address, which, if you enable Port Address Translation (PAT), can support up to 64,500 hosts concurrently. Although all packets receiving a new source IP address from that pool get the same address, they each get a different port number. The unique port number assigned for each IP address can be used only once and can support up to 62463 sessions per IP address. By maintaining a session table entry that matches the original address and port number with the translated address and port number, the security device can track which packets belong to which session and which sessions belong to which hosts.

NOTE: When PAT is enabled, the security device also maintains a pool of free port numbers to assign along with addresses from the DIP pool. The figure of up to 64,500 is derived by subtracting 1023, the numbers reserved for the well-known ports, from the maximum number of ports, which is 65,535.

The DIP pool supports more ports per session only if two packets have different destination IP addresses. The security device translates different source IP addresses and port numbers to a single IP address and port number without any conflict as long as the destination IP packets are different.

To enable a DIP pool to support more ports per session, you create port pools. A port pool consists of all available ports for an IP address. You override the port pool for a group of destination IP addresses that have the same hash value. The number of times you override the port pool of an IP address is determined by the scale-size. You can configure the scale-size using the following CLI:

set interface interface [ext ip ip_addr/mask] dip id_num ip_addr1 [ip_addr2] [random-port | incoming] [scale-size *number*]

By default, scale-size is 1. The maximum scale-size for an interface cannot exceed the dip-scale-size value specified in the vsys profile.

After you configure the scale-size, an IP address will have multiple port pools. When the packets arrive, screenOS calculates the hash value using the destination IP address and the scale-size. Based on the hash value, a port number is allocated from the port pool. Every port pool will have 62463 single ports. Hence, every IP address can support up to scale-size* 62463 sessions.

In this example, you assign ethernet3/1 an IP Address Range–1.1.1.23 to 1.1.1.26 with DIP ID 5. Set the Scale Size to 2 and the DIP Scale Size of the vsys profile to 2.

WebUI

Network > Interface > Edit (for ethernet3/1) > DIP: Enter the following, then click **OK**:

ID: 5
IP Address Range: 1.1.1.23 ~ 1.1.1.26
Port Translation: (select)
Scale Size: 2

Vsys > Profile > Edit: Enter the following, then click **OK**:

DIP Scale Size: 2

CLI

```
set interface etherent3/1 dip 5 1.1.1.23 1.1.1.26 scale-size 2
get interface ethernet3/1 dip 5 detail

set vsys-profile name dip-scale-size 2
get vsys-profile
```

After you configure the scale-size, every IP address supports up to scale-size* 62463 sessions.

In transparent mode, the current version of ScreenOS supports only policy based NAT-src with the dip pool built on the extended VLAN interface. To perform the address translation, you must configure a DIP pool on the VLAN interface and use the extended interface option to define an address range for the DIP pool. For more information, see *"Using DIP in a Different Subnet" on page 181*.

In the following example, you configure various DIP pools such as fix-port, port-xlate, and ip-shift on the vlan1 interface.

WebUI

1. **Interfaces**

   Network > Interface > Edit (vlan1): Enter the following, then click **OK:**

   Zone Name: VLAN
   Ip Address/ Netmask: 10.10.10.1/24

2. **DIP**

   Network > Interfaces > Edit (for vlan1) > DIP > New: Enter the following, then click **OK:**

   ID: 21
   IP Address Range (select), 20.20.20.1 ~ 20.20.20.10
   Port translation (select)
   IP Shift (select), From 5.5.5.1 To 20.20.20.50 ~ 20.20.20.59
   In the same subnet as the extended IP (select)
   Extended Ip/Netmask: 20.20.20.1/24

CLI

```
set interface vlan1 ip 10.10.10.1/24
set interface vlan1 ext ip 20.20.20.1/24 dip 20 20.20.20.1 20.20.20.10
set interface vlan1 ext ip 20.20.20.1/24 dip 21 20.20.20.30 20.20.20.39 fix-port
set interface vlan1 ext ip 20.20.20.1/24 dip 22 shift-from 5.5.5.1 to 20.20.20.50
 20.20.20.59
save
```

NOTE: In transparent mode, ScreenOS supports only policy based NAT-src on incoming packets.

If you use NAT-src but do not specify a DIP pool in the policy, the security device translates the source address to that of the egress interface in the destination zone. In such cases, PAT is required and automatically enabled.

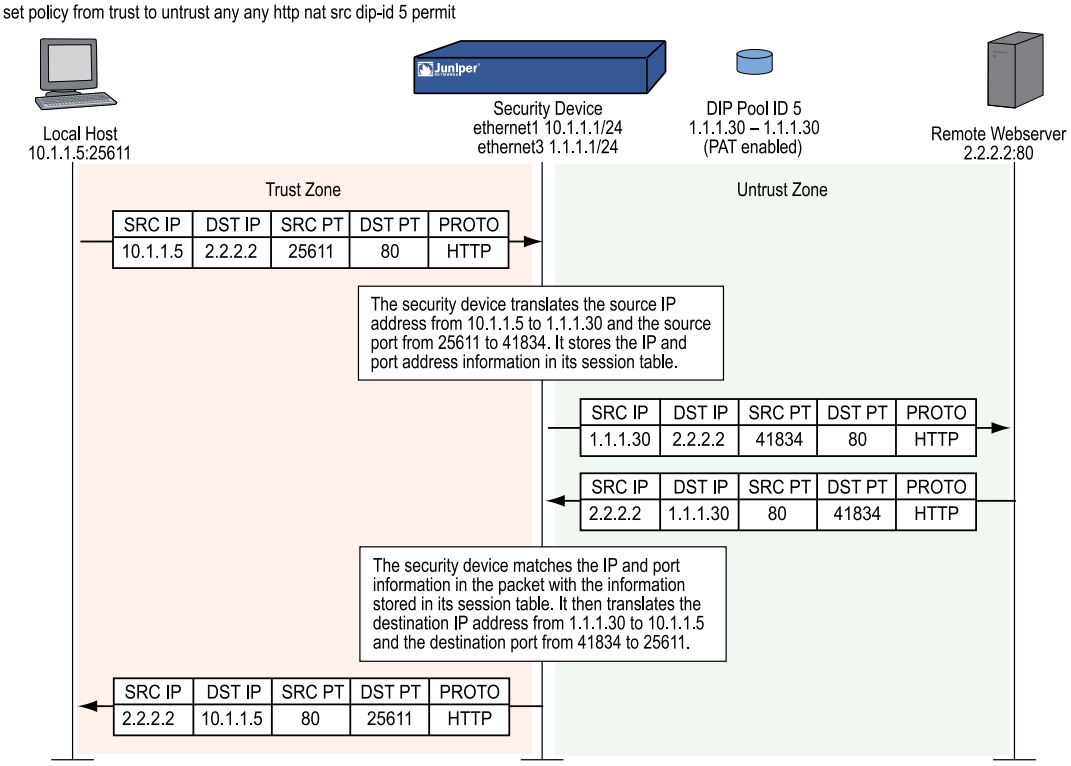
For applications requiring that a particular source port number remain fixed, you must disable PAT and define a DIP pool with a range of IP addresses large enough for each concurrently active host to receive a different translated address. For fixed-port DIP, the security device assigns one translated source address to the same host for all its concurrent sessions. In contrast, when the DIP pool has PAT enabled, the security device might assign a single host different addresses for different concurrent sessions—unless you define the DIP as sticky (see *"Sticky DIP Addresses" on page 180*).

NAT-Src from a DIP Pool with PAT Enabled

When applying Source Network Address Translation (NAT-src) with Port Address Translation (PAT), the security device translates IP addresses and port numbers, and performs stateful inspection as illustrated in Figure 368 on page 1485 (note that only the elements in the IP packet and TCP segment headers relevant to NAT-src are shown).

NOTE: You can add a maximum of three IP address ranges for a fixed-port DIP pool. The IP address ranges should not overlap. When the first address range is exhausted, the security device attempts to process the NAT request using the second address range. When the second address range is exhausted, the security device attempts to process the NAT request using the third address range. Note that the total range of all IP addresses defined in the fixed-port DIP pool must not exceed the permitted address scope of the subnet. For more information, see *"Creating a DIP Pool with PAT" on page 178*.

Figure 368: NAT-Src Using a DIP Pool with PAT Enabled



Example: NAT-Src with PAT Enabled

In this example, you define a DIP pool 5 on ethernet3, an interface bound to the Untrust zone. The DIP pool contains a single IP address—1.1.1.30—and has PAT enabled by default.

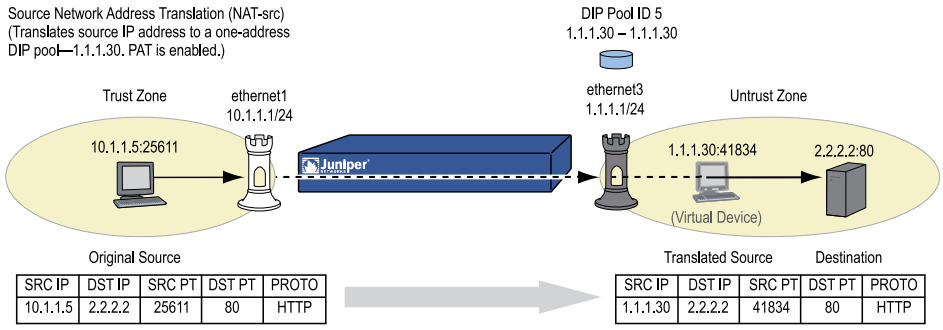
NOTE: When you define a DIP pool, the security device enables PAT by default. To disable PAT, you must add the key word fix-port to the end of the CLI command, or clear the Port Translation option on the DIP configuration page in the WebUI. For example, **set interface ethernet3 dip 5 1.1.1.30 1.1.1.30 fix-port**, or Network > Interfaces > Edit (for ethernet3) > DIP: ID: 5; Start: 1.1.1.30; End: 1.1.1.30; Port Translation: (clear).

You then set a policy that instructs the security device to perform the following tasks:

- Permit HTTP traffic from any address in the Trust zone to any address in the Untrust zone
- Translate the source IP address in the IP packet header to 1.1.1.30, which is the sole entry in DIP pool 5

- Translate the original source port number in the TCP segment header or UDP datagram header to a new, unique number
- Send HTTP traffic with the translated source IP address and port number out ethernet3 to the Untrust zone

Figure 369: NAT-Src with PAT Enabled



SRC IP	DST IP	SRC PT	DST PT	PROTO
10.1.1.5	2.2.2.2	25611	80	HTTP

SRC IP	DST IP	SRC PT	DST PT	PROTO
1.1.1.30	2.2.2.2	41834	80	HTTP

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **DIP**

   Network > Interfaces > Edit (for ethernet3) > DIP > New: Enter the following, then click **OK**:

   ID: 5
   IP Address Range: (select), 1.1.1.30 ~ 1.1.1.30
   Port Translation: (select)
   In the same subnet as the interface IP or its secondary IPs: (select)

3. **Policy**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: HTTP
   Action: Permit

   > Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   NAT:
   Source Translation: (select)
   (DIP on): 5 (1.1.1.30 - 1.1.1.30)/X-late

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **DIP**

```
set interface ethernet3 dip 5 1.1.1.30 1.1.1.30
```

3. **Policy**

```
set policy from trust to untrust any any http nat src dip-id 5 permit
save
```

NAT-Src from a DIP Pool with PAT Disabled

Certain configurations or situations may require you to perform Source Network Address Translation (NAT-src) for the IP address without performing Port Address Translation (PAT) for the source port number. For example, a custom application might require a specific number for the source port address. In such a case, you can define a policy instructing the security device to perform NAT-src without performing PAT.

Example: NAT-Src with PAT Disabled

In this example, you define a DIP pool 6 on ethernet3, an interface bound to the Untrust zone. The DIP pool contains a range of IP addresses from 1.1.1.50 to 1.1.1.150. You disable PAT. You then set a policy that instructs the security device to perform the following tasks:

- Permit traffic for a user-defined service named “e-stock” from any address in the Trust zone to any address in the Untrust zone

NOTE: It is assumed that you have previously defined the user-defined service “e-stock”. This fictional service requires that all e-stock transactions originate from specific source port numbers. For this reason, you must disable PAT for DIP pool 6.

- Translate the source IP address in the IP packet header to any available address in DIP pool 6
- Retain the original source port number in the TCP segment header or UDP datagram header
- Send e-stock traffic with the translated source IP address and original port number out ethernet3 to the Untrust zone

WebUI

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24
Select the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

2. **DIP**

Network > Interfaces > Edit (for ethernet3) > DIP > New: Enter the following, then click **OK**:

ID: 6
IP Address Range: (select), 1.1.1.50 ~ 1.1.1.150
Port Translation: (clear)
In the same subnet as the interface IP or its secondary IPs: (select)

3. **Policy**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: e-stock
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
DIP on: (select), 6 (1.1.1.50 - 1.1.1.150)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **DIP**

```
set interface ethernet3 dip 6 1.1.1.50 1.1.1.150 fix-port
```

3. **Policy**

```
set policy from trust to untrust any any e-stock nat src dip-id 6 permit
save
```

NAT-Src from a DIP Pool with Address Shifting

You can define a one-to-one mapping from an original source IP address to a translated source IP address for a range of IP addresses. Such a mapping ensures that the security device always translates a particular source IP address from within that range to the same translated address within a DIP pool. There can be any number of addresses in the range. You can even map one subnet to another subnet, with a consistent one-to-one mapping of each original address in one subnet to its translated counterpart in the other subnet.

One possible use for performing NAT-src with address shifting is to provide greater policy granularity on another security device that receives traffic from the first one. For example, the admin for Device-A at site A defines a policy that translates the source addresses of its hosts when communicating with Device-B at site B through a site-to-site VPN tunnel. If Device-A applies NAT-src using addresses from a DIP pool without address shifting, the Device-B admin can only configure generic policies regarding the traffic it can allow from site A. Unless the Device-B admin knows the specific translated IP addresses, he can only set inbound policies for the range of source addresses drawn from the Device-A DIP pool. On the other hand, if the Device-B admin knows what the translated source addresses are (because of address shifting), the Device-B admin can now be more selective and restrictive with the policies he sets for inbound traffic from site A.

Note that it is possible to use a DIP pool with address shifting enabled in a policy that applies to source addresses beyond the range specified in the pool. In such cases, the security device passes traffic from all source addresses permitted in the policy, applying NAT-src with address shifting to those addresses that fall within the DIP pool range but leaving those addresses that fall outside the DIP pool range unchanged. If you want the security device to apply NAT-src to all source addresses, make sure that the range of source addresses is smaller or the same size as the range of the DIP pool.

NOTE: The security device does not support source Port Address Translation (PAT) with address shifting.

Example: NAT-Src with Address Shifting

In this example, you define DIP pool 10 on ethernet3, an interface bound to the Untrust zone. You want to translate five addresses between 10.1.1.11 and 10.1.1.15 to five addresses between 1.1.1.101 and 1.1.1.105, and you want the relationship between each original and translated address to be consistent:

Table 104: NAT-Src with Address Shifting

Original Source IP Address	Translated Source IP Address
10.1.1.11	1.1.1.101
10.1.1.12	1.1.1.102
10.1.1.13	1.1.1.103
10.1.1.14	1.1.1.104
10.1.1.15	1.1.1.105

You define addresses for five hosts in the Trust zone and added them to an address group named "group1". The addresses for these hosts are 10.1.1.11, 10.1.1.12, 10.1.1.13, 10.1.1.14, and 10.1.1.15. You configure a policy from the Trust zone to the Untrust zone that references that address group in a policy to which you apply NAT-src with DIP pool 10. The policy instructs the security device to perform NAT-src whenever a member of group1 initiates HTTP traffic to an address in the Untrust zone. Furthermore, the security device always performs NAT-src from a particular IP address—such as 10.1.1.13—to the same translated IP address—1.1.1.103.

You then set a policy that instructs the security device to perform the following tasks:

- Permit HTTP traffic from group1 in the Trust zone to any address in the Untrust zone
- Translate the source IP address in the IP packet header to its corresponding address in DIP pool 10
- Send HTTP traffic with the translated source IP address and port number out ethernet3 to the Untrust zone

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **DIP**

   Network > Interfaces > Edit (for ethernet3) > DIP > New: Enter the following, then click **OK**:

   ID: 10
   IP Shift: (select)
   From: 10.1.1.11
   To: 1.1.1.101 ~ 1.1.1.105
   In the same subnet as the interface IP or its secondary IPs: (select)

3. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: host1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.11/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: host2
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.12/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: host3
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.13/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: host4
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.14/32
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: host5
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.15/32
   Zone: Trust

   Policy > Policy Elements > Addresses > Group > (for Zone: Trust) New: Enter the following group name, move the following addresses, then click **OK**:

Group Name: group1

Select **host1** and use the < < button to move the address from the Available Members column to the Group Members column.

Select **host2** and use the < < button to move the address from the Available Members column to the Group Members column.

Select **host3** and use the < < button to move the address from the Available Members column to the Group Members column.

Select **host4** and use the < < button to move the address from the Available Members column to the Group Members column.

Select **host5** and use the < < button to move the address from the Available Members column to the Group Members column.

4. **Policy**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), group1
Destination Address:
Address Book Entry: (select), Any
Service: HTTP
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Source Translation: (select)
(DIP on): 10 (1.1.1.101 - 1.1.1.105)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **DIP**

```
set interface ethernet3 dip 10 shift-from 10.1.1.11 to 1.1.1.101 1.1.1.105
```

3. **Addresses**

```
set address trust host1 10.1.1.11/32
set address trust host2 10.1.1.12/32
set address trust host3 10.1.1.13/32
set address trust host4 10.1.1.14/32
```

```
set address trust host5 10.1.1.15/32
set group address trust group1 add host1
set group address trust group1 add host2
set group address trust group1 add host3
set group address trust group1 add host4
set group address trust group1 add host5
```

4. **Policy**

```
set policy from trust to untrust group1 any http nat src dip-id 10 permit
save
```

NAT-Src from the Egress Interface IP Address

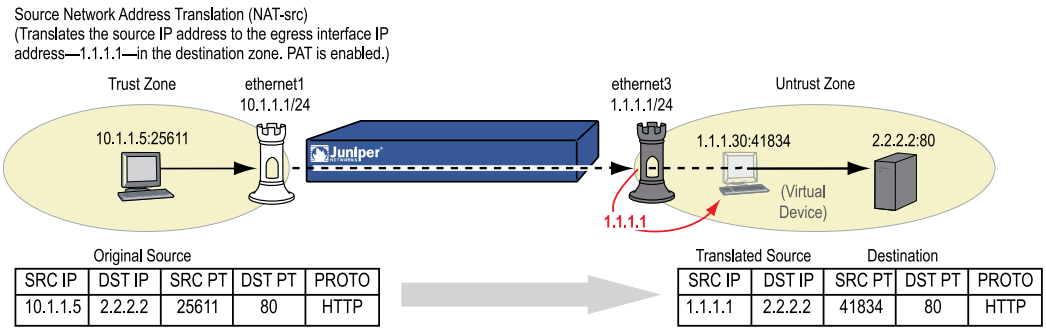
If you apply NAT-src to a policy but do not specify a DIP pool, then the security device translates the source IP address to the address of the egress interface. In such cases, the security device always applies PAT.

Example: NAT-Src Without DIP

In this example, you define a policy that instructs the security device to perform the following tasks:

- Permit HTTP traffic from any address in the Trust zone to any address in the Untrust zone
- Translate the source IP address in the IP packet header to 1.1.1.1, which is the IP address of ethernet3, the interface bound to the Untrust zone, and thus the egress interface for traffic sent to any address in the Untrust zone
- Translate the original source port number in the TCP segment header or UDP datagram header to a new, unique number
- Send traffic with the translated source IP address and port number out ethernet3 to the Untrust zone

Figure 370: NAT-Src Without DIP



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Policy**

   Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: HTTP
   Action: Permit

   > Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   NAT:
   Source Translation: (select)
   (DIP on): None (Use Egress Interface IP)

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Policy**

```
set policy from trust to untrust any any http nat src permit
save
```

Chapter 44

Destination Network Address Translation

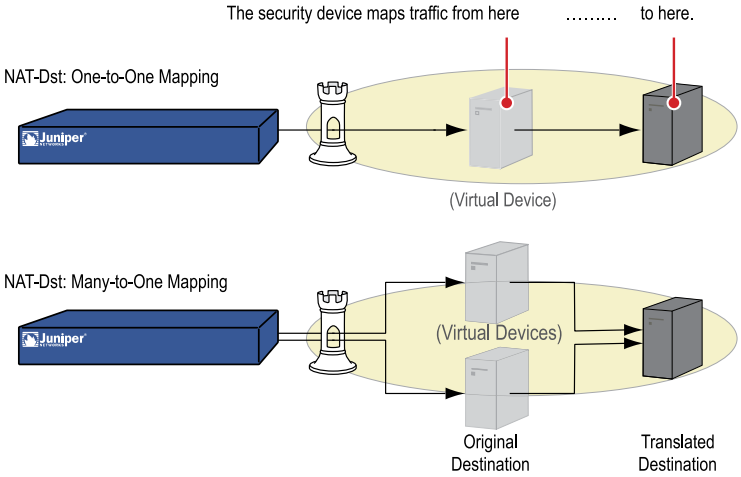
ScreenOS provides many methods for performing Destination Network Address Translation (NAT-dst) and destination port address mapping. This chapter describes the various address-translation methods available and contains the following sections:

NOTE: For information about destination address translation using a mapped IP (MIP) or virtual IP (VIP) address, see "Mapped and Virtual Addresses" on page 1535.

Introduction to NAT-Dst

You can define policies to translate the destination address from one IP address to another. Perhaps you need the security device to translate one or more public IP addresses to one or more private addresses. The relationship of the original destination address to the translated destination address can be a one-to-one relationship, a many-to-one relationship, or a many-to-many relationship. Figure 371 on page 1500 depicts the concepts of one-to-one and many-to-one NAT-dst relationships.

Figure 371: NAT-Dst—One-to-One and Many-to-One

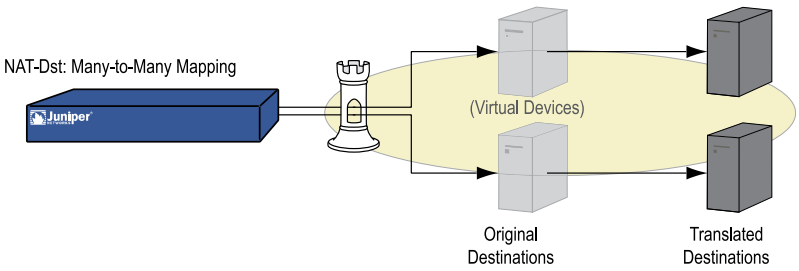


Note: The original and the translated destination IP addresses must be in the same security zone.

Both of the configurations shown in Figure 371 on page 1500 support destination port mapping. Port mapping is the deterministic translation of one original destination port number to another specific number. The relationship of the original-to-translated number in port mapping differs from Port Address Translation (PAT). With port mapping, the security device translates a predetermined original port number to another predetermined port number. With PAT, the security device translates a randomly assigned original source port number to another randomly assigned number.

You can translate a range of destination addresses to another range—such as one subnet to another—with address shifting, so that the security device consistently maps each original destination address to a specific translated destination address. Note that security does not support port mapping with address shifting. Figure 372 on page 1500 depicts the concept of a many-to-many relationship for NAT-dst.

Figure 372: NAT-Dst—Many-to-Many



There must be entries in the route table for both the original destination IP address and the translated destination IP address. The security device performs a route lookup using the original destination IP address to determine the destination zone for a subsequent policy lookup. It then performs a second route lookup using the translated address to determine where to send the packet. To ensure that the routing decision is in accord with the policy, both the original destination IP address and the translated IP address must be in the same security zone. (For more information about the

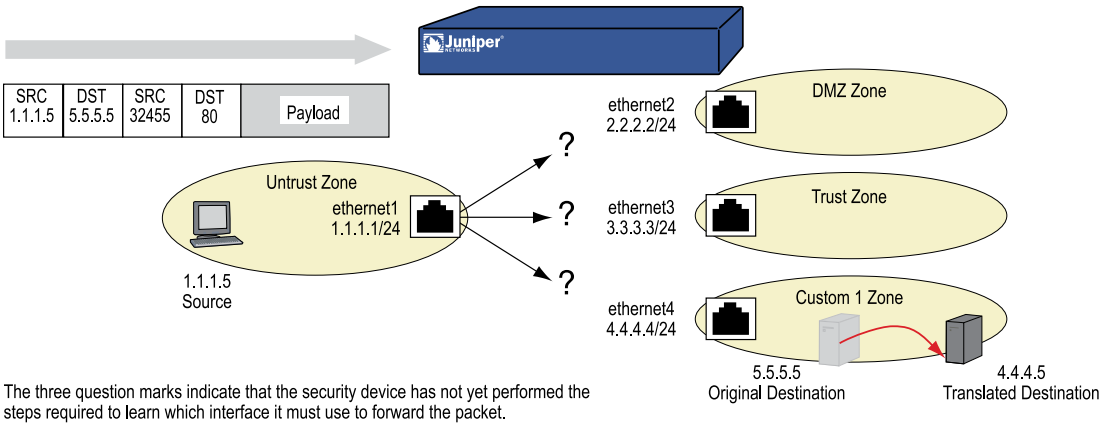
relationship of the destination IP address, route lookup, and policy lookup, see “Packet Flow for NAT-Dst” on page 1501.)

Packet Flow for NAT-Dst

The following steps describe the path of a packet through a security device and the various operations that it performs when applying NAT-dst:

1. An HTTP packet with source IP address:port number 1.1.1.5:32455 and destination IP address:port number 5.5.5.5:80 arrives at ethernet1, which is bound to the Untrust zone.

Figure 373: NAT-Dst Packet Flow—Packet Arrival



The three question marks indicate that the security device has not yet performed the steps required to learn which interface it must use to forward the packet.

2. If you have enabled SCREEN options for the Untrust zone, the security device activates the SCREEN module at this point. SCREEN checking can produce one of the following three results:

   - If a SCREEN mechanism detects anomalous behavior for which it is configured to block the packet, the security device drops the packet and makes an entry in the event log.
   - If a SCREEN mechanism detects anomalous behavior for which it is configured to record the event but not block the packet, the security device records the event in the SCREEN counters list for the ingress interface and proceeds to the next step.
   - If the SCREEN mechanisms detect no anomalous behavior, the security device proceeds to the next step.

     If you have not enabled any SCREEN options for the Untrust zone, the security device immediately proceeds to the next step.

3. The session module performs a session lookup, attempting to match the packet with an existing session.

   If the packet does not match an existing session, the security device performs First Packet Processing, a procedure involving the remaining steps.

If the packet matches an existing session, the security device performs Fast Processing, using the information available from the existing session entry to process the packet. Fast Processing bypasses all but the last step because the information generated by the bypassed steps has already been obtained during the processing of the first packet in the session.

4. The address-mapping module checks if a mapped IP (MIP) or virtual IP (VIP) configuration uses the destination IP address 5.5.5.5.

NOTE: The security device checks if the destination IP address is used in a VIP configuration only if the packet arrives at an interface bound to the Untrust zone.

If there is such a configuration, the security device resolves the MIP or VIP to the translated destination IP address and bases its route lookup on that. It then does a policy lookup between the Untrust and Global zones. If it finds a policy match that permits the traffic, the security device forwards the packet out the egress interface determined in the route lookup.

If 5.5.5.5 is not used in a MIP or VIP configuration, the security device proceeds to the next step.

5. To determine the destination zone, the route module does a route lookup of the original destination IP address; that is, it uses the destination IP address that appears in the header of the packet that arrives at ethernet1. (The route module uses the ingress interface to determine which virtual router to use for the route lookup.) It discovers that 5.5.5.5/32 is accessed through ethernet4, which is bound to the Custom1 zone.

trust-vr Route Table			
To Reach:	**Use Interface:**	**In Zone:**	**Use Gateway:**
0.0.0.0/0	ethernet1	Untrust	1.1.1.250
1.1.1.0/24	ethernet1	Untrust	0.0.0.0
2.2.2.0/24	ethernet2	DMZ	0.0.0.0
3.3.3.0/24	ethernet3	Trust	0.0.0.0
4.4.4.0/24	ethernet4	Custom1	0.0.0.0
5.5.5.5/32	**ethernet4**	Custom1	**0.0.0.0**

6. The policy engine does a policy lookup between the Untrust and Custom1 zones (as determined by the corresponding ingress and egress interfaces). The source and destination IP addresses and the service match a policy redirecting HTTP traffic from 5.5.5.5 to 4.4.4.5.

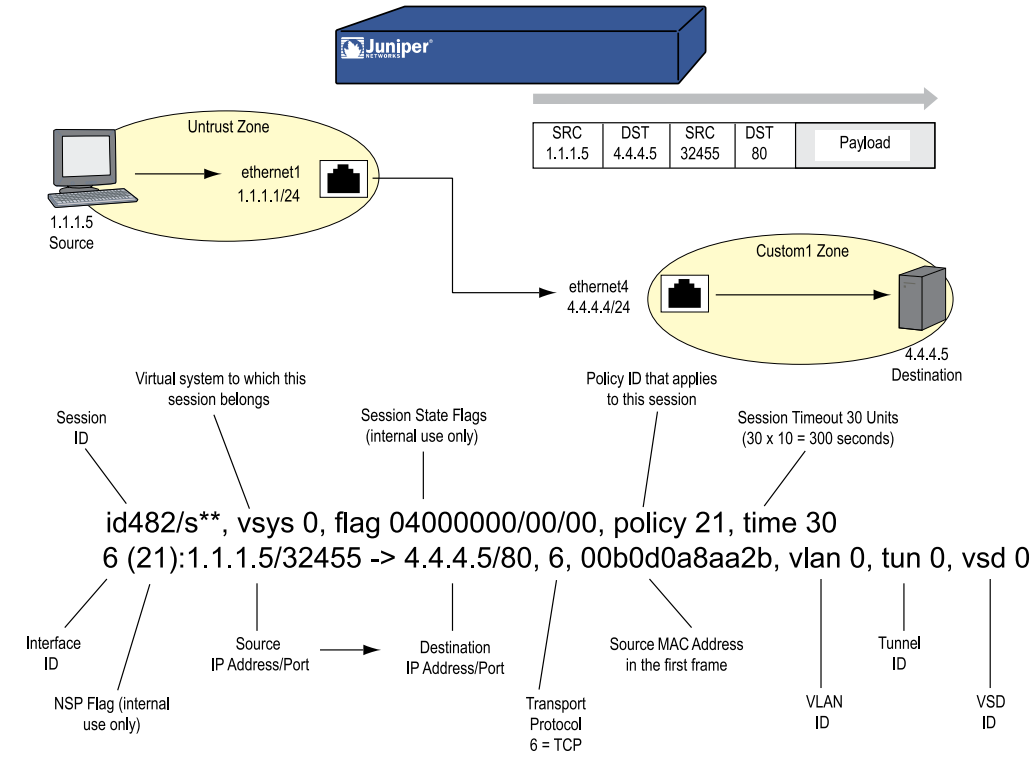
set policy from untrust to custom1 any v-server1 http nat dst ip 4.4.4.5 permit

(You have previously defined the address “v-server1” with IP address 5.5.5.5/32. It is in the Custom1 zone.)

The security device translates the destination IP address from 5.5.5.5 to 4.4.4.5. The policy indicates that neither NAT-Src nor PAT-dst is required.

7. The security device does a second route lookup using the translated IP address and discovers that 4.4.4.5/32 is accessed through ethernet4.
8. The address-mapping module translates the destination IP address in the packet header to 4.4.4.5. The security device then forwards the packet out ethernet4 and makes an entry in its session table (unless this packet is part of an existing session and an entry already exists).

Figure 374: NAT-Dst Packet Flow—Packet Forwarding



Note: Because this session does not involve a virtual system, VLAN, VPN tunnel, or virtual security device (VSD), the setting for all these ID numbers is zero.

Routing for NAT-Dst

When you configure addresses for NAT-dst, the security device must have routes in its routing table to both the original destination address that appears in the packet header and the translated destination address (that is, the address to which the security device redirects the packet). As explained in “Packet Flow for NAT-Dst” on page 1501, the security device uses the original destination address to do a route lookup, and thereby determine the egress interface. The egress interface in turn provides the destination zone—the zone to which the interface is bound—so that the security device can do a policy lookup. When the security device finds a policy match, the

policy defines the mapping of the original destination address to the translated destination address. The security device then performs a second route lookup to determine the interface through which it must forward the packet to reach the new destination address. In summary, the route to the original destination address provides a means to perform the policy lookup, and the route to the translated destination address specifies the egress interface through which the security device is to forward the packet.

In the following three scenarios, the need to enter static routes differs according to the network topology surrounding the destination addresses referenced in this policy:

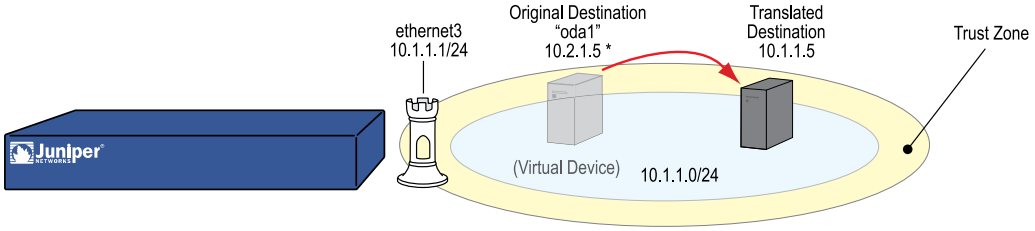
set policy from untrust to trust any oda1 http nat dst ip 10.1.1.5 permit

in which "oda1" is the original destination address 10.2.1.5, and the translated destination address is 10.1.1.5.

Example: Addresses Connected to One Interface

In this scenario, the routes to both the original and translated destination addresses direct traffic through the same interface, ethernet3. The security device automatically adds a route to 10.1.1.0/24 through ethernet3 when you configure the IP address of the ethernet3 interface as 10.1.1.1/24. To complete the routing requirements, you must add an additional route to 10.2.1.5/32 through ethernet3.

Figure 375: Original and Translated Addresses Using the Same Egress Interface



* Although 10.2.1.5 is not in the 10.1.1.0/24 subnet, because its route does not specify a gateway, it is illustrated as if it is in the same connected subnet as the 10.1.1.0/24 address space.

WebUI

Network > Routing > Destination > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.2.1.5/32
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 0.0.0.0

CLI

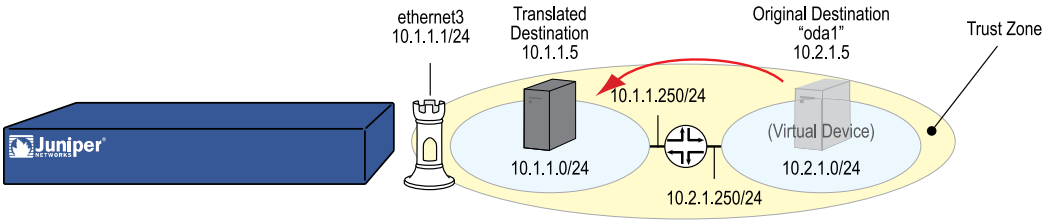
```
set vrouter trust-vr route 10.2.1.5/32 interface ethernet3
save
```

Example: Addresses Connected to One Interface But Separated by a Router

In this scenario, the routes to both the original and translated destination addresses direct traffic through ethernet3. The security device automatically adds a route to 10.1.1.0/24 through ethernet3 when you configure the IP address of the ethernet3 interface as 10.1.1.1/24. To complete the routing requirements, you must add a route to 10.2.1.0/24 through ethernet3 and the gateway connecting the 10.1.1.0/24 and the 10.2.1.0/24 subnets.

NOTE: Because this route is required to reach any address in the 10.2.1.0/24 subnet, you have probably already configured it. If so, no extra route needs to be added just for the policy to apply NAT-dst to 10.2.1.5.

Figure 376: Original and Translated Addresses Separated by a Router



WebUI

Network > Routing > Destination > (trust-vr) New: Enter the following, then click **OK**:

Network Address / Netmask: 10.2.1.0/24
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 10.1.1.250

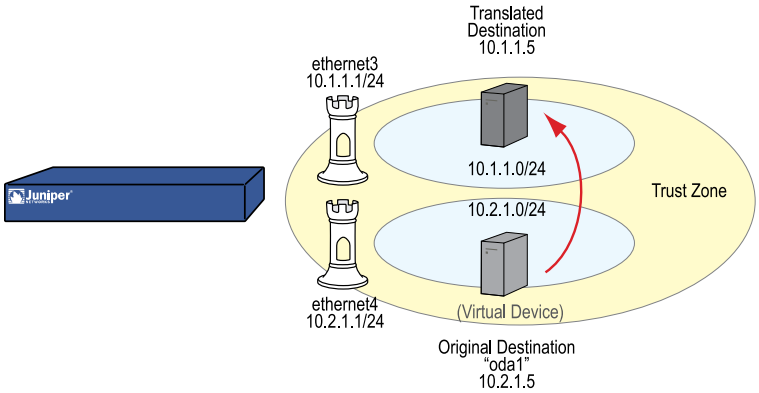
CLI

```
set vrouter trust-vr route 10.2.1.0/24 interface ethernet3 gateway 10.1.1.250
save
```

Example: Addresses Separated by an Interface

In this scenario, two interfaces are bound to the Trust zone: ethernet3 with IP address 10.1.1.1/24 and ethernet4 with IP address 10.2.1.1/24. The security device automatically adds a route to 10.1.1.0/24 through ethernet3 and 10.2.1.0/24 through ethernet4 when you configure the IP addresses of these interfaces. By putting the original destination address in the 10.2.1.0/24 subnet and the translated destination address in the 10.1.1.0/24 subnet, you do not have to add any other routes for the security device to apply NAT-dst from 10.1.1.5 to 10.2.1.5.

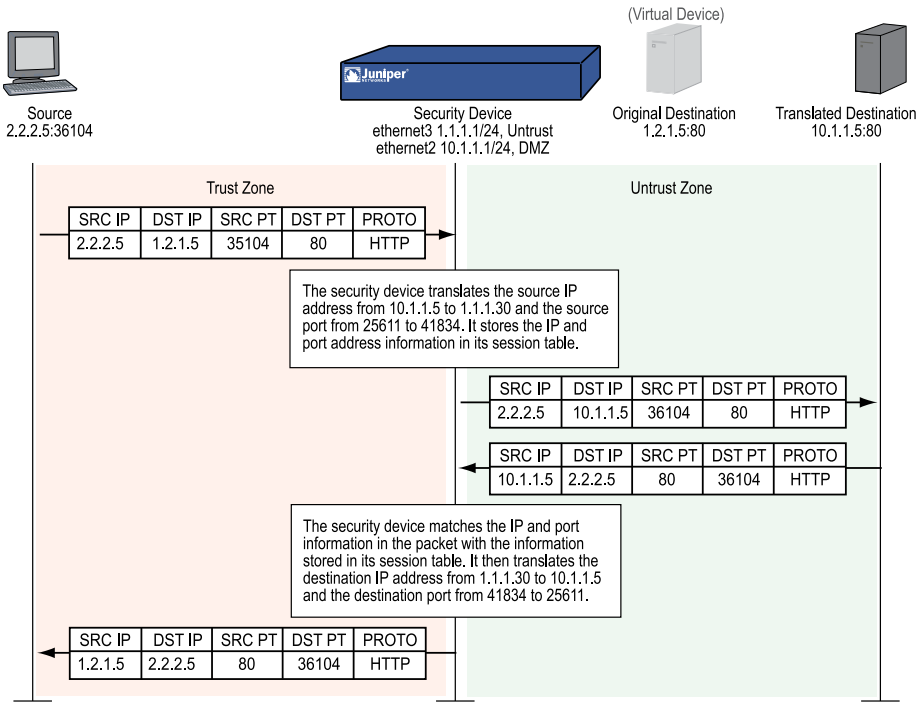
Figure 377: Original and Translated Addresses Using Different Egress Interfaces



NAT-Dst—One-to-One Mapping

When applying Destination Network Address Translation (NAT-dst) without PAT, the security device translates the destination IP address and performs stateful inspection as illustrated in Figure 378 on page 1506 (note that only the elements in the IP packet and TCP segment headers relevant to NAT-dst are shown).

Figure 378: One-to-One NAT-Dst



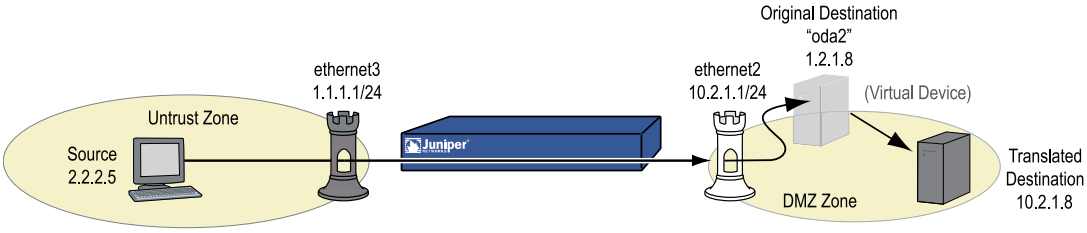
Example: One-to-One Destination Translation

In this example, you set a policy to provide one-to-one Destination Network Address Translation (NAT-dst) without changing the destination port addresses. The policy instructs the security device to perform the following tasks:

- Permit both FTP and HTTP traffic (defined as the service group “http-ftp”) from any address in the Untrust zone to a the original destination address named “ oda2” with address 1.2.1.8 in the DMZ zone
- Translate the destination IP address in the IP packet header from 1.2.1.8 to 10.2.1.8
- Leave the original destination port number in the TCP segment header as is (80 for HTTP and 21 for FTP)
- Forward HTTP and FTP traffic to 10.2.1.8 in the DMZ zone

You bind ethernet3 to the Untrust zone and assign it IP address 1.1.1.1/24. You bind ethernet2 to the DMZ and assign it IP address 10.2.1.1/24. You also define a route to the original destination address 1.2.1.8 through ethernet2. Both the Untrust and DMZ zones are in the trust-vr routing domain.

Figure 379: NAT-Dst—One-to-One



SRC IP	DST IP	SRC PT	DST PT	PROTO
2.2.2.5	1.2.1.5	25611	80	HTTP

SRC IP	DST IP	SRC PT	DST PT	PROTO
1.2.1.5	10.2.1.8	80	36104	HTTP

SRC IP	DST IP	SRC PT	DST PT	PROTO
2.2.2.5	1.2.1.8	40365	23	FTP

SRC IP	DST IP	SRC PT	DST PT	PROTO
2.2.2.5	10.2.1.8	40365	21	FTP

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: DMZ
Static IP: (select this option when present)
IP Address/Netmask: 10.2.1.1/24

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: oda2
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.1.8/32
   Zone: DMZ

3. **Service Group**

   Policy > Policy Elements > Services > Groups: Enter the following group name, move the following services, then click **OK**:

   Group Name: HTTP-FTP

   Select **HTTP** and use the < < button to move the service from the Available Members column to the Group Members column.

   Select **FTP** and use the < < button to move the service from the Available Members column to the Group Members column.

4. **Route**

   Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

   Network Address / Netmask: 1.2.1.8/32
   Gateway: (select)
   Interface: ethernet2
   Gateway IP Address: 0.0.0.0

5. **Policy**

   Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), oda2
   Service: HTTP-FTP
   Action: Permit

   > Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   NAT:
   Destination Translation: (select)
   Translate to IP: (select), 10.2.1.8
   Map to Port: (clear)

CLI

1. **Interfaces**

```
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet2 zone dmz
set interface ethernet2 ip 10.2.1.1/24
```

2. **Address**

```
set address dmz oda2 1.2.1.8/32
```

3. **Service Group**

```
set group service http-ftp
set group service http-ftp add http
set group service http-ftp add ftp
```

4. **Route**

```
set vrouter trust-vr route 1.2.1.8/32 interface ethernet2
```

5. **Policy**

```
set policy from untrust to dmz any oda2 http-ftp nat dst ip 10.2.1.8 permit
save
```

Translating from One Address to Multiple Addresses

The security device can translate the same original destination address to different translated destination addresses specified in different policies, depending on the type of service or the source address specified in each policy. You might want the security device to redirect HTTP traffic from 1.2.1.8 to 10.2.1.8, and FTP traffic from 1.2.1.8 to 10.2.1.9 (see the following example). Perhaps you want the security device to redirect HTTP traffic sent from host1 to 1.2.1.8 over to 10.2.1.8, but HTTP traffic sent from host2 to 1.2.1.8 over to 10.2.1.37. In both cases, the security device redirects traffic sent to the same original destination address to different translated addresses.

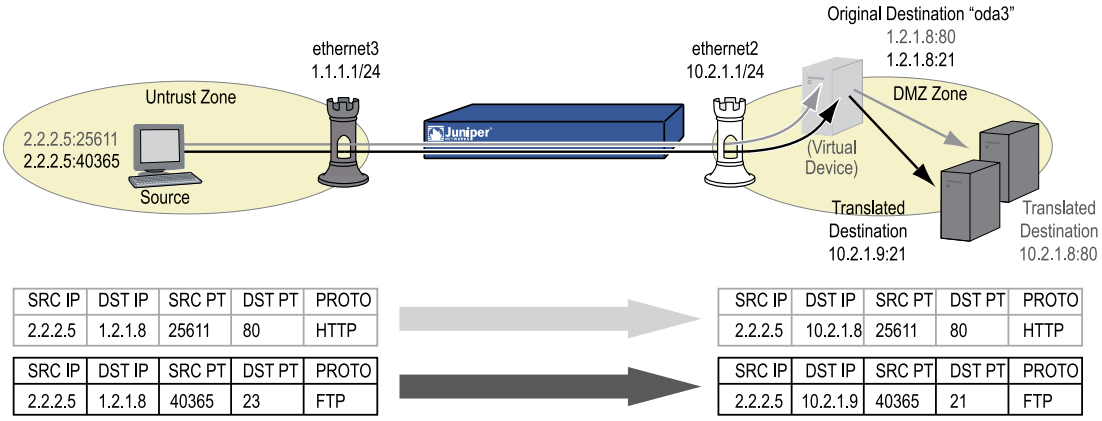
Example: One-to-Many Destination Translation

In this example, you create two policies that use the same original destination address (1.2.1.8), but that direct traffic sent to that address to two different translated destination addresses based on the service type. These policies instruct the security device to perform the following tasks:

- Permit both FTP and HTTP traffic from any address in the Untrust zone to a user-defined address named “oda3” in the DMZ zone
- For HTTP traffic, translate the destination IP address in the IP packet header from 1.2.1.8 to 10.2.1.8

- For FTP traffic, translate the destination IP address from 1.2.1.8 to 10.2.1.9
- Leave the original destination port number in the TCP segment header as is (80 for HTTP, 21 for FTP)
- Forward HTTP traffic to 10.2.1.8 and FTP traffic to 10.2.1.9 in the DMZ zone

Figure 380: NAT-Dst—One-to-Many



You bind ethernet3 to the Untrust zone and assign it IP address 1.1.1.1/24. You bind ethernet2 to the DMZ, and assign it IP address 10.2.1.1/24. You also define a route to the original destination address 1.2.1.8 through ethernet2. Both the Untrust zone and the DMZ zone are in the trust-vr routing domain.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select this option when present)
   IP Address/Netmask: 10.2.1.1/24

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: oda3
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.1.8/32
   Zone: DMZ

3. **Route**

   Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

   Network Address / Netmask: 1.2.1.8/32
   Gateway: (select)
   Interface: ethernet2
   Gateway IP Address: 0.0.0.0

4. **Policies**

   Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), oda3
   Service: HTTP
   Action: Permit

   > Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   NAT:
   Destination Translation: (select)
   Translate to IP: (select), 10.2.1.8
   Map to Port: (clear)

   Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), oda3
   Service: FTP
   Action: Permit

   > Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   NAT:
   Destination Translation: (select)
   Translate to IP: (select), 10.2.1.9
   Map to Port: (clear)

CLI

1. **Interfaces**

```
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet2 zone dmz
set interface ethernet2 ip 10.2.1.1/24
```

2. **Address**

   set address dmz oda3 1.2.1.8/32

3. **Route**

   set vrouter trust-vr route 1.2.1.8/32 interface ethernet2

4. **Policies**

   set policy from untrust to dmz any oda3 http nat dst ip 10.2.1.8 permit
   set policy from untrust to dmz any oda3 ftp nat dst ip 10.2.1.9 permit
   save

NAT-Dst—Many-to-One Mapping

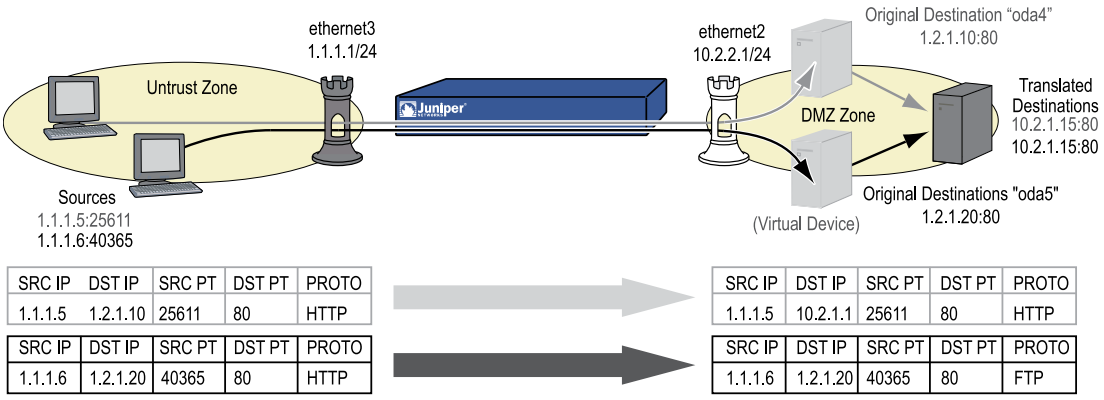
The relationship of the original destination address to the translated destination address can also be a many-to-one relationship. In this case, the security device forwards traffic sent to several original destination addresses to a single translated destination address. Optionally, you can also specify destination port mapping.

Example: Many-to-One Destination Translation

In this example, you create a policy that redirects traffic sent to different original destination addresses (1.2.1.10 and 1.2.1.20) to the same translated destination address. This policy instructs the security device to perform the following tasks:

- Permit HTTP traffic from any address in the Untrust zone to a user-defined address group named “oda45” with addresses “oda4” (1.2.1.10) and “oda5” (1.2.1.20) in the DMZ zone
- Translate the destination IP addresses in the IP packet header from 1.2.1.10 and 1.2.1.20 to 10.2.1.15
- Leave the original destination port number in the TCP segment header as is (80 for HTTP)
- Forward the HTTP traffic to 10.2.1.15 in the DMZ zone

Figure 381: NAT-Dst—Many-to-One



You bind ethernet3 to the Untrust zone and assign it IP address 1.1.1.1/24. You bind ethernet2 to the DMZ and assign it IP address 10.2.1.1/24. You also define a route to the original destination addresses 1.2.1.10 and 1.2.1.20 through ethernet2. Both the Untrust zone and the DMZ zone are in the trust-vr routing domain.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select this option when present)
   IP Address/Netmask: 10.2.1.1/24

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: oda4
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.1.10/32
   Zone: DMZ

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: oda5
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.1.20/32
   Zone: DMZ

   Policy > Policy Elements > Addresses > Groups > (for Zone: DMZ) New: Enter the following group name, move the following addresses, then click **OK**:

   Group Name: oda45

   Select **oda4** and use the < < button to move the address from the Available Members column to the Group Members column.

   Select **oda5** and use the < < button to move the address from the Available Members column to the Group Members column.

3. **Routes**

   Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 1.2.1.10/32
Gateway: (select)
Interface: ethernet2
Gateway IP Address: 0.0.0.0

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 1.2.1.20/32
Gateway: (select)
Interface: ethernet2
Gateway IP Address: 0.0.0.0

4. **Policy**

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), oda45
Service: HTTP
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Destination Translation: (select)
Translate to IP: (select), 10.2.1.15
Map to Port: (clear)

CLI

1. **Interfaces**

```
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet2 zone dmz
set interface ethernet2 ip 10.2.1.1/24
```

2. **Addresses**

```
set address dmz oda4 1.2.1.10/32
set address dmz oda5 1.2.1.20/32
set group address dmz oda45 add oda4
set group address dmz oda45 add oda5
```

3. **Routes**

```
set vrouter trust-vr route 1.2.1.10/32 interface ethernet2
set vrouter trust-vr route 1.2.1.20/32 interface ethernet2
```

4. **Policy**

```
set policy from untrust to dmz any oda45 http nat dst ip 10.2.1.15 permit
save
```

NAT-Dst—Many-to-Many Mapping

You can use Destination Network Address Translation (NAT-dst) to translate one range of IP addresses to another range. The range of addresses can be a subnet or a smaller set of addresses within a subnet. ScreenOS employs an address shifting mechanism to maintain the relationships among the original range of destination addresses after translating them to the new range of addresses. For example, if the range of original addresses is 10.1.1.1 – 10.1.1.50 and the starting address for the translated address range is 10.100.3.101, then the security device translates the addresses as follows:

- 10.1.1.1 – 10.100.3.101
- 10.1.1.2 – 10.100.3.102
- 10.1.1.3 – 10.100.3.103

  …
- 10.1.1.48 – 10.100.3.148
- 10.1.1.49 – 10.100.3.149
- 10.1.1.50 – 10.100.3.150

NOTE: When configuring Destination Network Address Translation (NAT-dst), do not specify the address group entry as the destination.

If, for example, you want to create a policy that applies the above translations to HTTP traffic from any address in zone A to an address object named “addr1-50” , which contains all the addresses from 10.1.1.1 to 10.1.1.50, in zone B, you can enter the following CLI command:

```
set policy id 1 from zoneA to zoneB any addr1-50 http nat dst ip 10.100.3.101
10.100.3.150 permit
```

If any host in zone A initiates HTTP traffic to an address within the defined range in zone B, such as 10.1.1.37, then the security device applies this policy and translates the destination address to 10.100.3.137.

The security device only performs NAT-dst if the source and destination zones, the source and destination addresses, and the service specified in the policy all match these components in the packet. For example, you might create another policy that permits traffic from any host in zone A to any host in zone B and position it after policy 1 in the policy list:

```
set policy id 1 from zoneA to zoneB any addr1-50 http nat dst ip 10.100.3.101
10.100.3.150 permit
set policy id 2 from zoneA to zoneB any any any permit
```

If you have these two policies configured, the following kinds of traffic sent from a host in zone A to a host in zone B bypass the NAT-dst mechanism:

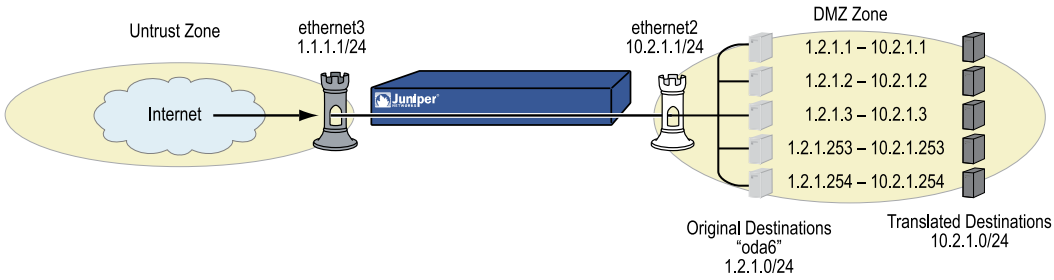
- A zone A host initiates non-HTTP traffic to 10.1.1.37 in zone B. The security device applies policy 2 because the service is not HTTP and passes the traffic without translating the destination address.
- A zone A host initiates HTTP traffic to 10.1.1.51 in zone B. The security device also applies policy 2 because the destination address is not in the addr1-50 address group, and passes the traffic without translating the destination address.

Example: Many-to-Many Destination Translation

In this example, you configure a policy that applies NAT-dst when any kind of traffic is sent to any host in a subnet, instructing the security device to perform the following tasks:

- Permit all traffic types from any address in the Untrust zone to any address in the DMZ zone
- Translate the original destination address named “oda6” from the 1.2.1.0/24 subnet to a corresponding address in the 10.2.1.0/24 subnet
- Leave the original destination port number in the TCP segment header as is
- Forward HTTP traffic to the translated address in the DMZ

Figure 382: NAT-Dst—Many-to-Many



You bind ethernet3 to the Untrust zone and assign it IP address 1.1.1.1/24. You bind ethernet2 to the DMZ and assign it IP address 10.2.1.1/24. You also define a route to the original destination address subnet (1.2.1.0/24) through ethernet2. Both the Untrust zone and the DMZ zone are in the trust-vr routing domain.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: DMZ
Static IP: (select this option when present)
IP Address/Netmask: 10.2.1.1/24

2. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

Address Name: oda6
IP Address/Domain Name:
IP/Netmask: (select), 1.2.1.0/24
Zone: DMZ

3. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 1.2.1.0/24
Gateway: (select)
Interface: ethernet2
Gateway IP Address: 0.0.0.0

4. **Policy**

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), oda6
Service: Any
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Destination Translation: (select)
Translate to IP Range: (select), 10.2.1.0 – 10.2.1.254

CLI

1. **Interfaces**

```
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet2 zone dmz
set interface ethernet2 ip 10.2.1.1/24
```

2. **Address**

```
set address dmz oda6 1.2.1.0/24
```

3. **Route**

```
set vrouter trust-vr route 1.2.1.0/24 interface ethernet2
```

4. **Policy**

```
set policy from untrust to dmz any oda6 any nat dst ip 10.2.1.0 10.2.1.254
permit
save
```

NAT-Dst with Port Mapping

When you configure the security device to perform Destination Network Address Translation (NAT-dst), you can optionally enable port mapping. One reason to enable port mapping is to support multiple server processes for a single service on a single host. For example, one host can run two Web servers—one at port 80 and another at port 8081. For HTTP service 1, the security device performs NAT-dst without port mapping (dst port 80 -> 80).

For HTTP service 2, the security device performs NAT-dst to the same destination IP address with port mapping (dst port 80 -> 8081). The host can sort HTTP traffic to the two Web servers by the two distinct destination port numbers.

NOTE: ScreenOS does not support port mapping for NAT-dst with address shifting. See “NAT-Dst—Many-to-Many Mapping” on page 1515.

Example: NAT-Dst with Port Mapping

In this example, you create two policies that perform NAT-dst and port mapping on Telnet traffic from the Trust and Untrust zones to a Telnet server in the DMZ zone. These policies instruct the security device to perform the following tasks:

- Permit Telnet from any address in the Untrust and Trust zones to 1.2.1.15 in the DMZ zone
- Translate the original destination IP address named “oda7” from 1.2.1.15 to 10.2.1.15
- Translate the original destination port number in the TCP segment header from 23 to 2200
- Forward Telnet traffic to the translated address in the DMZ zone

You configure the following interface-to-zone bindings and address assignments:

- ethernet1: Trust zone, 10.1.1.1/24
- ethernet2: DMZ zone, 10.2.1.1/24.
- ethernet3: Untrust zone, 1.1.1.1/24.

You define an address entry "oda7" with IP address 1.2.1.15/32 in the DMZ zone. You also define a route to the original destination address 1.2.1.15 through ethernet2. The Trust, Untrust, and DMZ zones are all in the trust-vr routing domain.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

   Zone Name: DMZ
   Static IP: (select this option when present)
   IP Address/Netmask: 10.2.1.1/24

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following information, then click **OK**:

   Address Name: oda7
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.1.15/32
   Zone: DMZ

3. **Route**

   Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

   Network Address / Netmask: 1.2.1.15/32
   Gateway: (select)
   Interface: ethernet2
   Gateway IP Address: 0.0.0.0

4. **Policies**

   Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any

Destination Address:
Address Book Entry: (select), oda7
Service: Telnet
Action: Permit

> Advanced: Enter the following, then click Return to set the advanced options and return to the basic configuration page:

NAT:
Destination Translation: (select)
Translate to IP: (select), 10.2.1.15
Map to Port: (select), 2200

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), oda7
Service: Telnet
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

NAT:
Destination Translation: (select)
Translate to IP: (select), 10.2.1.15
Map to Port: (select), 2200

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet2 zone dmz
set interface ethernet2 ip 10.2.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Address**

```
set address dmz oda7 1.2.1.15/32
```

3. **Route**

```
set vrouter trust-vr route 1.2.1.15/32 interface ethernet2
```

4. **Policies**

```
set policy from trust to dmz any oda7 telnet nat dst ip 10.2.1.15 port 2200
permit
set policy from untrust to dmz any oda7 telnet nat dst ip 10.2.1.15 port 2200
```

```
permit
save
```

Using proxy-arp-entry to import the NAT—DST traffic to the right VSI

The administrator can enable importing the ARP traffic to the correct VSI by setting the proxy ARP entry. On adding a proxy ARP entry on an interface, ScreenOS imports the traffic that is destined to the IP range using this interface.

When the ARP request arrives at the physical interface, the system runs a search for the proxy-arp-entry in each VSI interface. After identifying the VSI interface on which the proxy-arp-entry is identified, the system sends the ARP response using the MAC address of the VSI interface.

This CLI takes precedence over the existing set arp nat-dst command. This means that when the proxy-arp-entry is defined and matched, then the system does not respond to the ARP request via the physical interface.

Because the proxy-arp-entry command allows the customer to have better control of the device, the command “set arp nat-dstis not recommended.

Advantages of the proxy-arp-entry command over the arp dst-nat command

- Unlike the set arp dst-nat command, the administrator can configure the proxy-arp-entry on any interface that belongs to non-root VSYS. The administrator of any non-root VSYS can configure proxy-arp-entry on any interface that belongs to or is shared to the VSYS.
- The proxy-arp-entry can be configured on any VSI interface. This feature is supported by the three modes (AA/AP/Lite) in NSRP environment.

Configuration Example

The following example illustrates the configuration of the proxy-arp-entry:

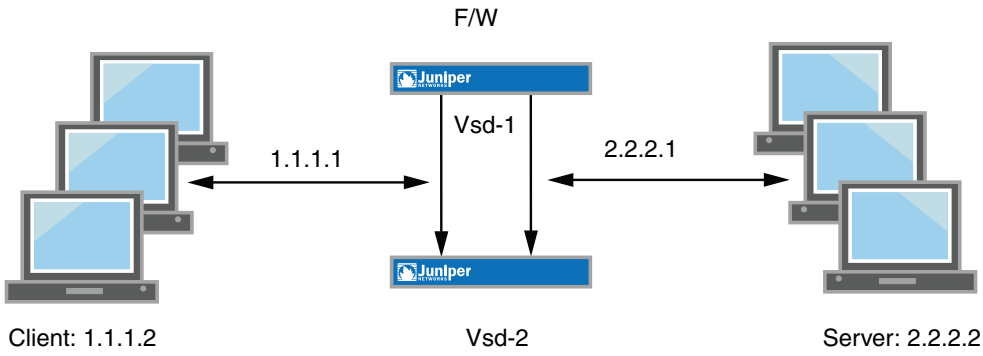
```
set int e0/1:1 proxy-arp-entry 3.3.3.10 3.3.3.12
set int e0/1:2 proxy-arp-entry 3.3.3.13 3.3.3.

get int e0/1:1 proxy-arp-entry
interface ip_low ip_high
ethernet0/1:1 3.3.3.10 3.3.3.12
ethernet0/1:2 3.3.3.13 3.3.3.15

get proxy-arp-entry
interface ip_low ip_high
ethernet0/1:1 3.3.3.10 3.3.3.12
ethernet0/1:2 3.3.3.13 3.3.3.15
```

The following example illustrates the usage of proxy-arp-entry:

Figure 383: Proxy ARP Entry




```
Set interfaces e1:1 proxy-arp-entry 1.1.1.5.1.1.17
Set policy from untrust to untrust any *1.1.1.5* any nat dst ip 2.2.2.2 permit
```

In the above example:

1. ScreenOS device responds to the ARP request that asks for the MAC of 1.1.1.5 with ethernet1:1's MAC address.
2. The consequent traffic is imported into the VSD group 1 (ethernet1:1).

If one device crashes, then the backup device sends a gratuitous ARP to the client. The consequent traffic is imported into the backup VSD group of another box.

NAT-Src and NAT-Dst in the Same Policy

You can combine Source Network Address Translation (NAT-Src) and Destination Network Address Translation (NAT-dst) in the same policy. This combination provides you with a method of changing both the source and the destination IP addresses at a single point in the data path.

Example: NAT-Src and NAT-Dst Combined

In the example shown in Figure 384 on page 1524, you configure a security device (Device-1) that is between a service provider's customers and server farms. The customers connect to Device-1 through ethernet1, which has IP address 10.1.1.1/24 and is bound to the Trust zone. Device-1 then forwards their traffic through one of two route-based VPN tunnels to reach the servers they want to target. The tunnel interfaces that are bound to these tunnels are in the Untrust zone. Both the Trust and Untrust zones are in the trust-vr routing domain.

NOTE: Policy-based VPNs do not support NAT-dst. You must use a route-based VPN configuration with NAT-dst.

Because the customers might have the same addresses as those of the servers to which they want to connect, Device-1 must perform both Source and Destination Network Address Translation (NAT-Src and NAT-dst). To retain addressing independence and flexibility, the security devices protecting the server farms—Device-A and Device-B—perform NAT-dst. The service provider instructs the customers and the server farm admins to reserve addresses 10.173.10.1–10.173.10.7, 10.173.20.0/24, 10.173.30.0/24, 10.173.40.0/24, and 10.173.50.0/24 for this purpose. These addresses are used as follows:

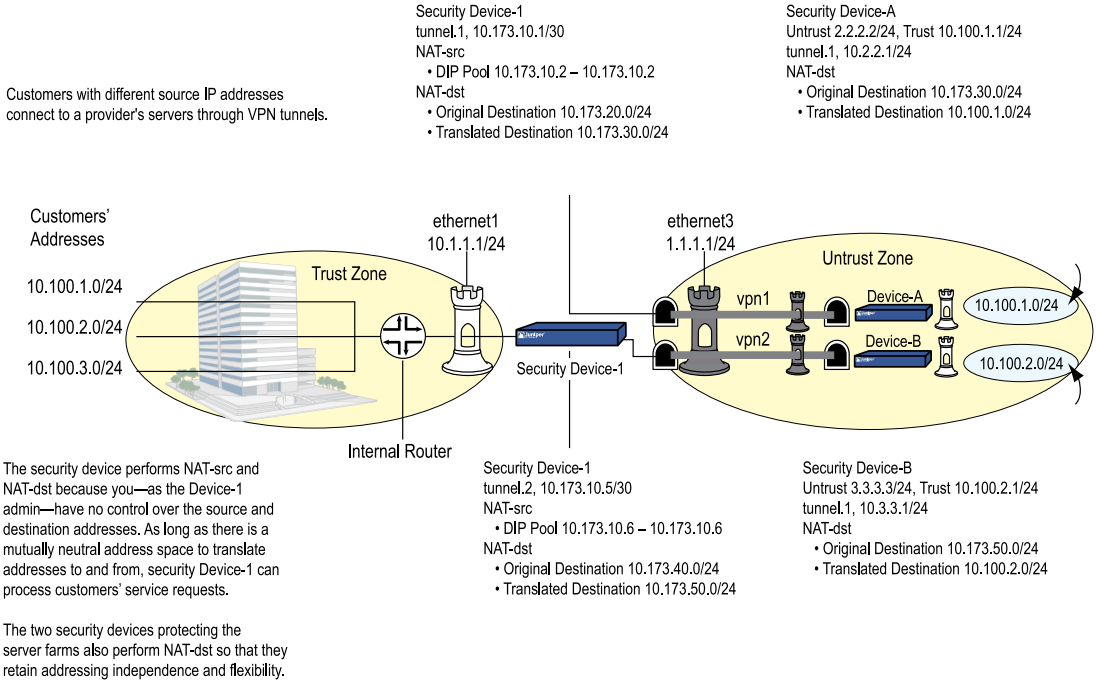
- The two tunnel interfaces have the following address assignments:
  - tunnel.1, 10.173.10.1/30
  - tunnel.2, 10.173.10.5/30
- Each tunnel interface supports the following DIP pools with PAT enabled:
  - tunnel.1, DIP ID 5: 10.173.10.2–10.173.10.2
  - tunnel.2, DIP ID 6: 10.173.10.6–10.173.10.6
- When Device-1 performs NAT-dst, it translates original destination addresses with address shifting as follows:
  - 10.173.20.0/24 to 10.173.30.0/24
  - 10.173.40.0/24 to 10.173.50.0/24

NOTE: For information about address shifting when performing NAT-dst, see “NAT-Dst—Many-to-Many Mapping” on page 1515.

The configurations for both tunnels—vpn1 and vpn2—use the following parameters: AutoKey IKE, preshared key (“device1” for vpn1, and “ device2” for vpn2), and the security level predefined as “ Compatible” for both Phase 1 and Phase 2 proposals. (For details about these proposals, see *“Tunnel Negotiation” on page 715*.) The proxy ID for both vpn1 and vpn2 is 0.0.0.0/0 - 0.0.0.0/0 - any.

NOTE: The configuration for Device-1 is provided first. The VPN configurations for Device-A and Device-B follow and are included for completeness.

Figure 384: NAT-Src and NAT-Dst Combined



WebUI (Security Device-1)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT
   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Fixed IP: (select)
   IP Address / Netmask: 10.173.10.1/30

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)

Fixed IP: (select)
IP Address / Netmask: 10.173.10.5/30

2. **DIP Pools**

   Network > Interfaces > Edit (for tunnel.1) > DIP > New: Enter the following, then click **OK**:

   ID: 5
   IP Address Range: (select), 10.173.10.2 ~ 10.173.10.2
   Port Translation: (select)
   In the same subnet as the interface IP or its secondary IPs: (select)

   Network > Interfaces > Edit (for tunnel.2) > DIP > New: Enter the following, then click **OK**:

   ID: 6
   IP Address Range: (select), 10.173.10.6 ~ 10.173.10.6
   Port Translation: (select)
   In the same subnet as the interface IP or its secondary IPs: (select)

3. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: serverfarm-A
   IP Address/Domain Name:
   IP/Netmask: (select), 10.173.20.0/24
   Zone: Untrust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: serverfarm-B
   IP Address/Domain Name:
   IP/Netmask: (select), 10.173.40.0/24
   Zone: Untrust

4. **VPNs**

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: vpn1
   Security Level: Compatible
   Remote Gateway: Create a Simple Gateway: (select)
   Gateway Name: gw-A
   Type: Static IP: (select), Address/Hostname: 2.2.2.2
   Preshared Key: device1
   Security Level: Compatible
   Outgoing Interface: ethernet3

NOTE: The outgoing interface does not have to be in the same zone to which the tunnel interface is bound, although in this case they are in the same zone.

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to Tunnel Interface: (select), tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn2
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: gw-B
Type: Static IP: (select), Address/Hostname: 3.3.3.3
Preshared Key: device2
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to Tunnel Interface: (select), tunnel.2
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

5. **Routes**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.173.20.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.173.30.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.173.40.0/24
Gateway: (select)
Interface: tunnel.2
Gateway IP Address: 0.0.0.0

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.173.50.0/24
Gateway: (select)
Interface: tunnel.2
Gateway IP Address: 0.0.0.0

6. **Policies**

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), serverfarm-A
Service: ANY
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Source Translation: (select)
(DIP on): 5 (10.173.10.2–10.173.10.2)/X-late
Destination Translation: (select)
Translate to IP Range: (select), 10.173.30.0 – 10.173.30.255

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), serverfarm-B
Service: ANY
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Source Translation: (select)
(DIP on): 6 (10.173.10.6–10.173.10.6)/X-late
Destination Translation: (select)
Translate to IP Range: (select), 10.173.50.0 – 10.173.50.255

CLI (Security Device-1)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.173.10.1/30
set interface tunnel.2 zone untrust
set interface tunnel.2 ip 10.173.10.5/30
```

2. **DIP Pools**

```
set interface tunnel.1 dip-id 5 10.173.10.2 10.173.10.2
set interface tunnel.2 dip-id 6 10.173.10.6 10.173.10.6
```

3. **Addresses**

```
set address untrust serverfarm-A 10.173.20.0/24
set address untrust serverfarm-B 10.173.40.0/24
```

4. **VPNs**

```
set ike gateway gw-A ip 2.2.2.2 main outgoing-interface ethernet3 preshare
device1 sec-level compatible
set vpn vpn1 gateway gw-A sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
set ike gateway gw-B ip 3.3.3.3 main outgoing-interface ethernet3 preshare
device2 sec-level compatible
set vpn vpn2 gateway gw-B sec-level compatible
set vpn vpn2 bind interface tunnel.2
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
set vrouter trust-vr route 10.173.20.0/24 interface tunnel.1
set vrouter trust-vr route 10.173.30.0/24 interface tunnel.1
set vrouter trust-vr route 10.173.40.0/24 interface tunnel.2
set vrouter trust-vr route 10.173.50.0/24 interface tunnel.2
```

6. **Policies**

```
set policy top from trust to untrust any serverfarm-A any nat src dip-id 5 dst ip
10.173.30.0 10.173.30.255 permit
set policy top from trust to untrust any serverfarm-B any nat src dip-id 6 dst ip
10.173.50.0 10.173.50.255 permit
save
```

WebUI (Security Device-A)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.100.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 2.2.2.2/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Fixed IP: (select)
   IP Address / Netmask: 10.2.2.1/24

2. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: serverfarm-A
   IP Address/Domain Name:
   IP/Netmask: (select), 10.173.30.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: customer1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.173.10.2/32
   Zone: Untrust

3. **VPN**

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: vpn1
   Security Level: Compatible
   Remote Gateway: Create a Simple Gateway: (select)
   Gateway Name: gw-1

Type: Static IP: (select), Address/Hostname: 1.1.1.1
Preshared Key: device1
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to Tunnel Interface: (select), tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

4. **Routes**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 2.2.2.250

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.173.10.2/32
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.173.30.0/24
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 0.0.0.0

5. **Policy**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), customer1
Destination Address:
Address Book Entry: (select), serverfarm-A
Service: ANY
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Destination Translation: (select)
Translate to IP Range: (select), 10.100.1.0 – 10.100.1.255

CLI (Security Device-A)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.100.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.2.2.1/24
```

2. **Addresses**

```
set address trust serverfarm-A 10.173.30.0/24
set address untrust customer1 10.173.10.2/32
```

3. **VPN**

```
set ike gateway gw-1 ip 1.1.1.1 main outgoing-interface ethernet3 preshare
device1 sec-level compatible
set vpn vpn1 gateway gw-1 sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
set vrouter trust-vr route 10.173.10.2/32 interface tunnel.1
set vrouter trust-vr route 10.173.30.0/24 interface ethernet1
```

5. **Policy**

```
set policy top from untrust to trust customer1 serverfarm-A any nat dst ip
10.100.1.0 10.100.1.255 permit
save
```

WebUI (Security Device-B)

1. **Interfaces**

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.100.2.1/24
Select the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 3.3.3.3/24
Tunnel Interface Name: tunnel.1
Zone (VR): Untrust (trust-vr)
Fixed IP: (select)
IP Address / Netmask: 10.3.3.1/24

2. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: serverfarm-B
IP Address/Domain Name:
IP/Netmask: (select), 10.173.50.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: customer1
IP Address/Domain Name:
IP/Netmask: (select), 10.173.10.6/32
Zone: Untrust

3. **VPN**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: gw-1
Type: Static IP: (select), Address/Hostname: 1.1.1.1
Preshared Key: device2
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to Tunnel Interface: (select), tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

4. **Routes**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 3.3.3.250

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.173.10.6/32
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.173.50.0/24
Gateway: (select)
Interface: ethernet1
Gateway IP Address: 0.0.0.0

5. **Policy**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), customer1
Destination Address:
Address Book Entry: (select), serverfarm-B
Service: ANY
Action: Permit
Position at Top: (select)

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Policy configuration page:

NAT:
Destination Translation: (select)
Translate to IP Range: (select), 10.100.2.0 – 10.100.2.255

CLI (Security Device-B)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.100.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 3.3.3.3/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.3.3.1/24
```

2. **Addresses**

```
set address trust serverfarm-B 10.173.50.0/24
set address untrust customer1 10.173.10.6/32
```

3. **VPN**

```
set ike gateway gw-1 ip 1.1.1.1 main outgoing-interface ethernet3 preshare
device2 sec-level compatible
set vpn vpn2 gateway gw-1 sec-level compatible
set vpn vpn2 bind interface tunnel.1
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

4. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 3.3.3.250
set vrouter trust-vr route 10.173.10.6/32 interface tunnel.1
set vrouter trust-vr route 10.173.50.0/24 interface ethernet1
```

5. **Policy**

```
set policy top from untrust to trust customer1 serverfarm-B any nat dst ip
10.100.2.0 10.100.2.255 permit
save
```

Chapter 45

Mapped and Virtual Addresses

ScreenOS provides many methods for performing destination IP address and destination Port Address Translation (PAT). This chapter describes how to use mapped IP (MIP) and virtual IP (VIP) addresses and is organized into the following sections:

Mapped IP Addresses

Mapped IP (MIP) is a direct one-to-one mapping of one IP address to another. The security device forwards incoming traffic destined for a MIP to the host with the address to which the MIP points. Essentially, a MIP is static destination address translation, mapping the destination IP address in an IP packet header to another static IP address. When a MIP host initiates outbound traffic, the security device translates the source IP address of the host to that of the MIP address. This bidirectional translation symmetry differs from the behavior of source and destination address translation (see "Directional Nature of NAT-Src and NAT-Dst" on page 1478).

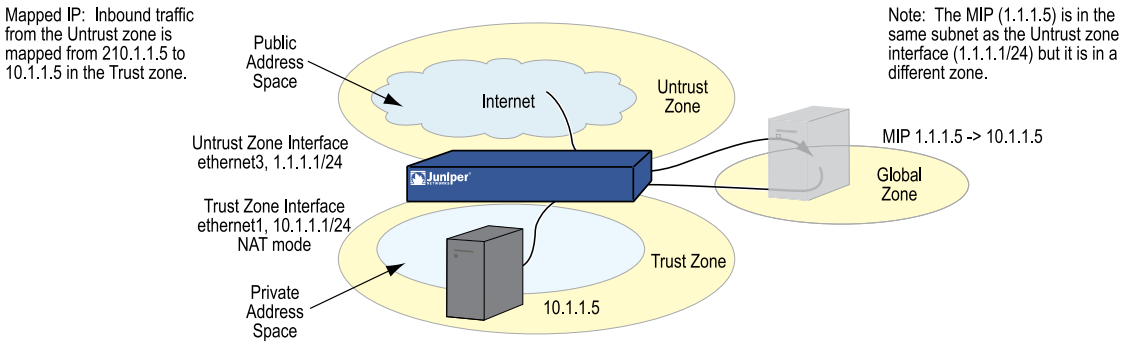
MIPs allow inbound traffic to reach private addresses in a zone whose interface is in NAT mode. MIPs also provide part of the solution to the problem of overlapping address spaces at two sites connected by a VPN tunnel. (For the complete solution to this problem, see *"VPN Sites with Overlapping Addresses" on page 863*.)

NOTE: An overlapping address space is when the IP address range in two networks is partially or completely the same.

You can create a MIP in the same subnet as a tunnel interface with an IP address/netmask, or in the same subnet as the IP address/netmask of an interface bound to a Layer 3 (L3) security zone. Although you configure MIPs for interfaces bound to tunnel zones and security zones, the MIP that you define is stored in the Global zone.

NOTE: An exception is a MIP defined for an interface in the Untrust zone. That MIP can be in a different subnet from an Untrust zone interface IP address. However, if that is the case, you must add a route on the external router pointing to an Untrust zone interface so that incoming traffic can reach the MIP. Also, you must define a static route on the security device associating the MIP with the interface that hosts it.

Figure 385: Mapped IP Address



NOTE: On some security devices, a MIP can use the same address as an interface, but a MIP address cannot be in a DIP pool.
You can map an address-to-address or subnet-to-subnet relationship. When a subnet-to-subnet mapped IP configuration is defined, the netmask is applied to both the mapped IP subnet and the original IP subnet.

MIP and the Global Zone

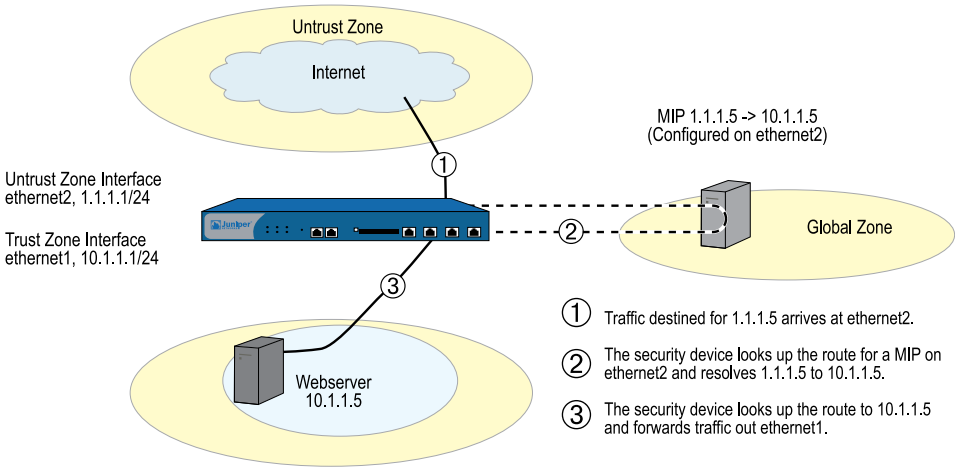
Setting a MIP for an interface in any zone generates an entry for the MIP in the Global zone address book. The Global zone address book stores all MIPs, regardless of the zone to which their interfaces belong. You can use these MIP addresses as the destination addresses in policies between any two zones, and as the destination addresses when defining a Global policy. (For information about Global policies, see *"Global Policies" on page 200*.) Although the security device stores MIPs in the Global zone, you can use either the Global zone or the zone with the address to which the MIP points as the destination zone in a policy referencing a MIP.

Example: MIP on an Untrust Zone Interface

In this example, you bind ethernet1 to the Trust zone and assign it IP address 10.1.1.1/24. You bind ethernet2 to the Untrust zone and assign it IP address 1.1.1.1/24. Then you configure a MIP to direct incoming HTTP traffic destined for 1.1.1.5 in the Untrust zone to a Web server at 10.1.1.5 in the Trust zone. Finally, you create a policy permitting HTTP traffic from the any address in the Untrust zone to the MIP—and consequently to the host with the address to which the MIP points—in the Trust zone. All security zones are in the trust-vr routing domain.

NOTE: No address book entry is required for a MIP or for the host to which it points.

Figure 386: MIP on Untrust Zone Interface



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **MIP**

   Network > Interfaces > Edit (for ethernet2) > MIP > New: Enter the following, then click **OK**:

   Mapped IP: 1.1.1.5
   Netmask: 255.255.255.255
   Host IP Address: 10.1.1.5
   Host Virtual Router Name: trust-vr

3. **Policy**

   Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), MIP(1.1.1.5)
Service: HTTP
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet2 zone untrust
set interface ethernet2 ip 1.1.1.1/24
```

2. **MIP**

```
set interface ethernet2 mip 1.1.1.5 host 10.1.1.5 netmask 255.255.255.255
 vrouter trust-vr
```

NOTE: By default, the netmask for a MIP is 32 bits (255.255.255.255), mapping the address to a single host. You can also define a MIP for a range of addresses. For example, to define 1.1.1.5 as a MIP for the addresses 10.1.10.129–10.1.10.254 within a class C subnet through the CLI, use the following syntax: **set interface** *interface* **mip 1.1.1.5 host 10.1.10.128 netmask 255.255.255.128**. Be careful not to use a range of addresses that includes the interface or router addresses.
The default virtual router is the trust-vr. You do not have to specify that the virtual router is the trust-vr or that the MIP has a 32-bit netmask. These arguments are included in this command to provide symmetry with the WebUI configuration.

3. **Policy**

```
set policy from untrust to trust any mip(1.1.1.5) http permit
save
```

Example: Reaching a MIP from Different Zones

Traffic from different zones can still reach a MIP through other interfaces than the one on which you configured the MIP. To accomplish this, you must set a route on the routers in each of the other zones that points inbound traffic to the IP address of their respective interfaces to reach the MIP.

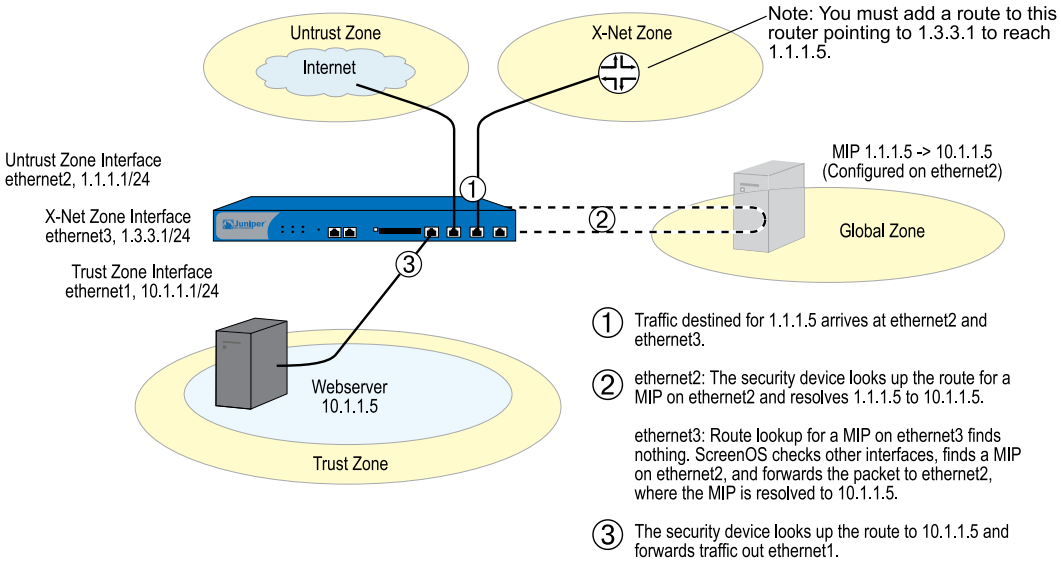
NOTE: If the MIP is in the same subnet as the interface on which you configured it, you do not have to add a route to the security device for traffic to reach the MIP via a different interface. However, if the MIP is in a different subnet than the IP address of its interface (which is possible only for a MIP on an interface in the Untrust zone), you must add a static route to the security device routing table. Use the **set vrouter** name_str **route** ip_addr **interface** interface command (or its equivalent in the WebUI), where name_str is the virtual router to which the specified interface belongs, and interface is interface on which you configured the MIP.

In this example, you configure a MIP (1.1.1.5) on the interface in the Untrust zone (ethernet2, 1.1.1.1/24) to map to a Web server in the Trust zone (10.1.1.5). The interface bound to the Trust zone is ethernet1 with IP address 10.1.1.1/24.

You create a security zone named X-Net, bind ethernet3 to it, and assign the interface the IP address 1.3.3.1/24. You define an address for 1.1.1.5 for use in a policy to allow HTTP traffic from any address in the X-Net zone to the MIP in the Untrust zone. You also configure a policy to allow the HTTP traffic to pass from the Untrust zone to the Trust zone. All security zones are in the trust-vr routing domain.

NOTE: You must enter a route on the router in the X-Net zone directing traffic destined for 1.1.1.5 (MIP) to 1.3.3.1 (IP address of ethernet3).

Figure 387: Reaching a MIP from Different Zones



WebUI

1. **Interfaces and Zones**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24
Select the following, then click **OK**:
Interface Mode: NAT

Network > Interfaces > Edit (for ethernet2): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 1.1.1.1/24

Network > Zones > New: Enter the following, then click **OK**:

Zone Name: X-Net
Virtual Router Name: trust-vr
Zone Type: Layer 3

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Zone Name: X-Net
IP Address/Netmask: 1.3.3.1/24

2. **Address**

Policy > Policy Elements > Addresses > List New: Enter the following, then click **OK**:

Address Name: 1.1.1.5
IP Address/Domain Name:
IP/Netmask: (select), 1.1.1.5/32
Zone: Untrust

3. **MIP**

Network > Interfaces > Edit (for ethernet2) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 1.1.1.5
Netmask: 255.255.255.255
Host IP Address: 10.1.1.5
Host Virtual Router Name: trust-vr

4. **Policies**

Policies > (From: X-Net, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), 1.1.1.5
Service: HTTP
Action: Permit

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), MIP(1.1.1.5)
Service: HTTP
Action: Permit

CLI

1. **Interfaces and Zones**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet2 zone untrust
set interface ethernet2 ip 1.1.1.1/24
set zone name X-Net
set interface ethernet3 zone X-Net
set interface ethernet3 ip 1.3.3.1/24
```

2. **Address**

```
set address untrust "1.1.1.5"  1.1.1.5/32
```

3. **MIP**

```
set interface ethernet2 mip 1.1.1.5 host 10.1.1.5 netmask 255.255.255.255
 vrouter trust-vr
```

NOTE: By default, the netmask for a MIP is 32 bits (255.255.255.255) and the default virtual router is the trust-vr. You do not have to specify them in the command. These arguments are included here to provide symmetry with the WebUI configuration.

4. **Policies**

```
set policy from X-Net to untrust any "1.1.1.5"  http permit
set policy from untrust to trust any mip(1.1.1.5) http permit
save
```

Example: Adding a MIP to a Tunnel Interface

In this example, the IP address space for the network in the Trust zone is 10.1.1.0/24 and the IP address for the tunnel interface "tunnel.8" is 10.20.3.1. The physical IP address for a server on the network in the Trust zone is 10.1.1.25. To allow a remote site whose network in the Trust zone uses an overlapping address space to access the local server through a VPN tunnel, you create a MIP in the same subnet as the tunnel.8 interface. The MIP address is 10.20.3.25/32. (For a more complete example of a MIP with a tunnel interface, see *"VPN Sites with Overlapping Addresses" on page 863*.)

WebUI

Network > Interfaces > Edit (for tunnel.8) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 10.20.3.25
Netmask: 255.255.255.255
Host IP Address: 10.1.1.25
Host Virtual Router Name: trust-vr

CLI

```
set interface tunnel.8 mip 10.20.3.25 host 10.1.1.25 netmask 255.255.255.255
 vrouter trust-vr
save
```

NOTE: By default, the netmask for a MIP is 32 bits (255.255.255.255) and the default virtual router is the trust-vr. You do not have to specify them in the command. These arguments are included here to provide symmetry with the WebUI configuration. When the remote administrator adds the address for the server to his Untrust zone address book, he must enter the MIP (10.20.3.25), not the physical IP address (10.1.1.25) of the server.
The remote administrator also needs to apply policy-based NAT-src (using DIP) on the outgoing packets bound for the server through the VPN so that the local administrator can add an Untrust zone address that does not conflict with the local Trust zone addresses. Otherwise, the source address in the incoming policy would seem to be in the Trust zone.

MIP-Same-as-Untrust

As IPv4 addresses become increasingly scarce, ISPs are becoming increasingly reluctant to give their customers more than one or two IP addresses. If you only have one IP address for the interface bound to the Untrust zone—the interface bound to the Trust zone is in Network Address Translation (NAT) mode—you can use the Untrust zone interface IP address as a mapped IP (MIP) to provide inbound access to an internal server or host, or to a VPN or L2TP tunnel endpoint.

A MIP maps traffic arriving at the one address to another address; so by using the Untrust zone interface IP address as a MIP, the security device maps all inbound traffic using the Untrust zone interface to a specified internal address. If the MIP on the Untrust interface maps to a VPN or L2TP tunnel endpoint, the device automatically forwards the IKE or L2TP packets that it receives to the tunnel endpoint, as long as there is no VPN or L2TP tunnel configured on the Untrust interface.

If you create a policy in which the destination address is a MIP using the Untrust zone interface IP address and you specify HTTP as the service in the policy, you lose Web management of the security device via that interface (because all inbound HTTP traffic to that address is mapped to an internal server or host). You can still manage the device via the Untrust zone interface using the WebUI by changing the port

number for Web management. To change the Web management port number, do the following:

1. Admin > Web: Enter a registered port number (from 1024 to 65,535) in the HTTP Port field. Then click **Apply**.
2. When you next connect to the Untrust zone interface to manage the device, append the port number to the IP address—for example, http://209.157.66.170:5000.

Example: MIP on the Untrust Interface

In this example, you select the IP address of the Untrust zone interface (ethernet3, 1.1.1.1/24) as the MIP for a Web server whose actual IP address is 10.1.1.5 in the Trust zone. Because you want to retain Web management access to the ethernet3 interface, you change the web management port number to 8080. You then create a policy permitting HTTP service (on the HTTP default port number—80) from the Untrust zone to the MIP—and consequently to the host with the address to which the MIP points—in the Trust zone.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Enter the following, then click **OK**:

   NAT: (select)

NOTE: By default, any interface that you bind to the Trust zone is in NAT mode. Consequently, this option is already enabled for interfaces bound to the Trust zone.

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **HTTP Port**

   Configuration > Admin > Management: Type **8080** in the HTTP Port field, then click **Apply**.

   (The HTTP connection is lost.)

3. **Reconnection**

Reconnect to the security device, appending 8080 to the IP address in the URL address field in your browser. (If you are currently managing the device via the untrust interface, enter **http://1.1.1.1:8080**.)

4. **MIP**

   Network > Interface > Edit (for ethernet3) > MIP > New: Enter the following, then click **OK**:

   Mapped IP: 1.1.1.1
   Netmask: 255.255.255.255
   Host IP Address: 10.1.1.5
   Host Virtual Router Name: trust-vr

NOTE: The netmask for a MIP using an Untrust zone interface IP address must be 32 bits.

5. **Route**

   Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

   Network Address / Netmask: 0.0.0.0/0
   Gateway: (select)
   Interface: ethernet3
   Gateway IP Address: 1.1.1.250

6. **Policy**

   Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), MIP(1.1.1.1)
   Service: HTTP
   Action: Permit

CLI

1. **Interfaces**

   ```
   set interface ethernet1 zone trust
   set interface ethernet1 ip 10.1.1.1/24
   set interface ethernet1 nat
   set interface ethernet3 zone untrust
   set interface ethernet3 ip 1.1.1.1/24
   ```

2. **HTTP Port**

   ```
   set admin port 8080
   ```

3. **MIP**

```
set interface ethernet3 mip 1.1.1.1 host 10.1.1.5 netmask 255.255.255.255
 vrouter trust-vr
```

NOTE: By default, the netmask for a MIP is 32 bits (255.255.255.255) and the default virtual router is the trust-vr. You do not have to specify them in the command. These arguments are included here to provide symmetry with the WebUI configuration.

4. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

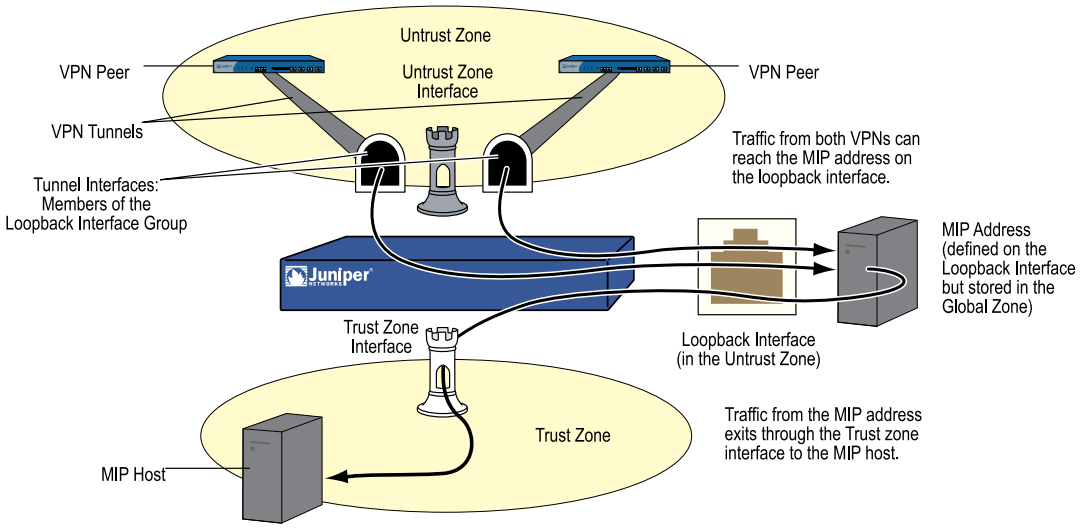
5. **Policy**

```
set policy from untrust to trust any mip(1.1.1.1) http permit
save
```

MIP and the Loopback Interface

Defining a MIP on the loopback interface allows a MIP to be accessed by a group of interfaces. The primary application for this is to allow access to a host through one of several VPN tunnels using a single MIP address. The MIP host can also initiate traffic to a remote site through the appropriate tunnel.

Figure 388: MIP on the Loopback Interface



You can think of the loopback interface as a resource holder that contains a MIP address. You configure a loopback interface with the name loopback.id_num (where id_num is an index number that uniquely identifies the interface in the device) and assign an IP address to the interface (see *"Loopback Interfaces" on page 75*). To allow other interfaces to use a MIP on the loopback interface, you then add the interfaces as members of the loopback group.

The loopback interface and its member interfaces must be in different IP subnets in the same zone. Any type of interface can be a member of a loopback group as long as the interface has an IP address. If you configure a MIP on both a loopback interface and one of its member interfaces, the loopback interface configuration takes precedence. A loopback interface cannot be a member of another loopback group.

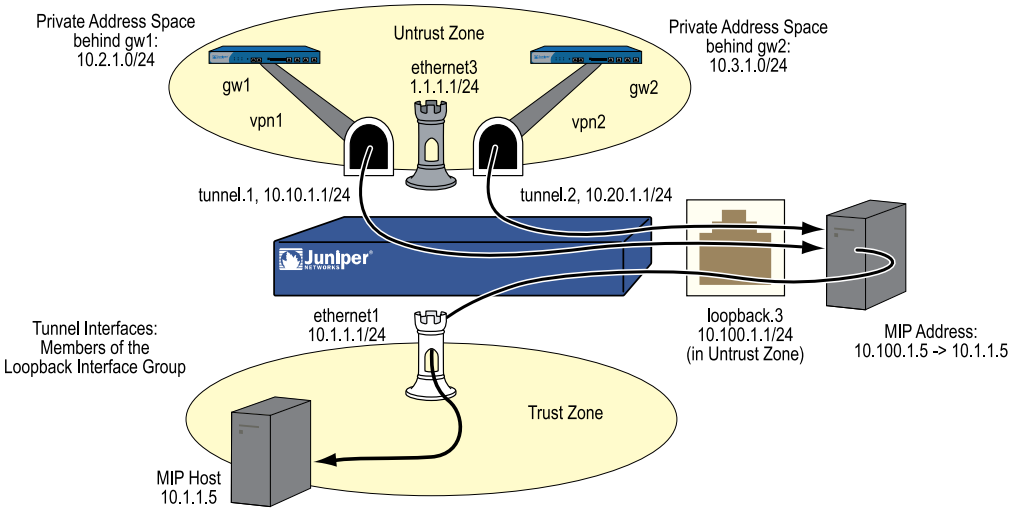
Example: MIP for Two Tunnel Interfaces

In this example, you configure the following interfaces:

- ethernet1, Trust zone, 10.1.1.1/24
- ethernet3, Untrust zone, 1.1.1.1/24
- tunnel.1, Untrust zone, 10.10.1.1/24, bound to vpn1
- tunnel.2, Untrust zone, 10.20.1.1/24, bound to vpn2
- loopback.3, Untrust zone, 10.100.1.1/24

The tunnel interfaces are members of the loopback.3 interface group. The loopback.3 interface contains MIP address 10.100.1.5, which maps to a host at 10.1.1.5 in the Trust zone.

Figure 389: MIP for Two Tunnel Interfaces



When a packet destined for 10.100.1.5 arrives at through a VPN tunnel to tunnel.1, the security device searches for the MIP on the loopback interface loopback.3. When it finds a match on loopback.3, the security device translates the original destination IP (10.100.1.5) to the host IP address (10.1.1.5) and forwards the packet through ethernet1 to the MIP host. Traffic destined for 10.100.1.5 can also arrive through a VPN tunnel bound to tunnel.2. Again, the security device finds a match on loopback.3 and translates the original destination IP 10.100.1.5 to 10.1.1.5 and forwards the packet to the MIP host.

You also define addresses, VPN tunnels, routes, and policies as needed to complete the configuration. All security zones are in the trust-vr routing domain.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Enter the following, then click **OK**:

   NAT: (select)

NOTE: By default, any interface that you bind to the Trust zone is in NAT mode. Consequently, this option is already enabled for interfaces bound to the Trust zone.

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

   Network > Interfaces > New Loopback IF: Enter the following, then click **OK**:

   Interface Name: loopback.3
   Zone: Untrust (trust-vr)
   IP Address / Netmask: 10.100.1.1/24

   Network > Interfaces > New Tunnel IF: Enter the following, then click **Apply**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Fixed IP: (select)
   IP Address / Netmask: 10.10.1.1/24

   Select **loopback.3** in the Member of Loopback Group drop-down list, then click **OK**.

   Tunnel Interface Name: tunnel.2
   Zone (VR): Untrust (trust-vr)
   Fixed IP: (select)
   IP Address / Netmask: 10.20.1.1/24

   Select **loopback.3** in the Member of Loopback Group drop-down list, then click **OK**.

2. **MIP**

   Network > Interfaces > Edit (for loopback.3) > MIP > New: Enter the following, then click **OK**:

Mapped IP: 10.100.1.5
Netmask: 255.255.255.255
Host IP Address: 10.1.1.5
Host Virtual Router Name: trust-vr

3. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: local_lan
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

Policy > Policy Elements > Addresses > New: Enter the following, then click **OK**:

Address Name: peer-1
IP Address/Domain Name:
IP/Netmask: (select), 10.2.1.0/24
Zone: Untrust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: peer-2
IP Address/Domain Name:
IP/Netmask: (select), 10.3.1.0/24
Zone: Untrust

4. **VPNs**

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: gw1
Type: Static IP: (select), Address/Hostname: 2.2.2.2
Preshared Key: device1
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to Tunnel Interface: (select), tunnel.1
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn2
Security Level: Compatible
Remote Gateway: Create a Simple Gateway: (select)
Gateway Name: gw2
Type: Static IP: (select), Address/Hostname: 3.3.3.3
Preshared Key: device2
Security Level: Compatible
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Bind to Tunnel Interface: (select), tunnel.2
Proxy-ID: (select)
Local IP / Netmask: 0.0.0.0/0
Remote IP / Netmask: 0.0.0.0/0
Service: ANY

5. **Routes**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.2.1.0/24
Gateway: (select)
Interface: tunnel.1
Gateway IP Address: 0.0.0.0

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 10.3.1.0/24
Gateway: (select)
Interface: tunnel.2
Gateway IP Address: 0.0.0.0

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address / Netmask: 0.0.0.0/0
Gateway: (select)
Interface: ethernet3
Gateway IP Address: 1.1.1.250

6. **Policies**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), peer-1
Destination Address:
Address Book Entry: (select), MIP(10.100.1.5)
Service: ANY
Action: Permit

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), peer-2
Destination Address:
Address Book Entry: (select), MIP(10.100.1.5)
Service: ANY
Action: Permit

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), local_lan
Destination Address:
Address Book Entry: (select), Any
Service: ANY
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface loopback.3 zone untrust
set interface loopback.3 ip 10.100.1.1/24
set interface tunnel.1 zone untrust
set interface tunnel.1 ip 10.10.1.1/24
set interface tunnel.1 loopback-group loopback.3
set interface tunnel.2 zone untrust
set interface tunnel.2 ip 10.20.1.1/24
set interface tunnel.2 loopback-group loopback.3
```

2. **MIP**

```
set interface loopback.3 mip 10.100.1.5 host 10.1.1.5 netmask
255.255.255.255 vrouter trust-vr
```

NOTE: By default, the netmask for a MIP is 32 bits (255.255.255.255) and the default virtual router is the trust-vr. You do not have to specify them in the command. These arguments are included here to provide symmetry with the WebUI configuration.

3. **Addresses**

```
set address trust local_lan 10.1.1.0/24
set address untrust peer-1 10.2.1.0/24
set address untrust peer-2 10.3.1.0/24
```

4. **VPNs**

```
set ike gateway gw1 address 2.2.2.2 outgoing-interface ethernet3 preshare
device1 sec-level compatible
```

```
set vpn vpn1 gateway gw1 sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
set ike gateway gw2 address 3.3.3.3 outgoing-interface ethernet3 preshare
device2 sec-level compatible
set vpn vpn2 gateway gw2 sec-level compatible
set vpn vpn2 bind interface tunnel.2
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

5. **Routes**

```
set vrouter trust-vr route 10.2.1.0/24 interface tunnel.1
set vrouter trust-vr route 10.3.1.0/24 interface tunnel.2
set vrouter untrust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

6. **Policies**

```
set policy top from untrust to trust peer-1 mip(10.100.1.5) any permit
set policy top from untrust to trust peer-2 mip(10.100.1.5) any permit
set policy from trust to untrust local_lan any any permit
save
```

MIP Grouping

Assigning a MIP to an interface sets aside a range of IP addresses to use as destination addresses for packets that pass through the interface. You can then use the MIP in a policy, which makes the address recognizable to interfaces receiving incoming packets or usable for interfaces transmitting outgoing packets.

However, there may be situations when it is necessary to invoke multiple MIPs in a single policy. For example, a single address range may not provide enough addresses when there are many destination hosts or gateways in a particular network topology. Such a solution requires *MIP grouping*, which allows you to design *multi-cell policies*. Such policies invoke multiple address ranges as possible source addresses.

Example: MIP Grouping with Multi-Cell Policy

In the following example, you create a policy that uses two different MIP address definitions (1.1.1.3 and 1.1.1.4).

NOTE: For this example, assume that the policy ID for the generated policy is 104.

WebUI

Network > Interfaces > Edit > MIP (List)

Policies > New

CLI

```
set interface ethernet1/2 mip 1.1.1.3 host 2.2.2.2
set interface ethernet1/2 mip 1.1.1.4 host 3.3.3.3
set policy from untrust to trust any mip(1.1.1.3) any permit
set policy id 104
(policy:104)-> set dst-address mip(1.1.1.4)
(policy:104)-> exit
get policy id 104
```

NOTE: After executing the last CLI command in this example, look for the following output for confirmation:
2 destinations: “MIP(1.1.1.3)” , “ MIP(1.1.1.4)”

Virtual IP Addresses

A virtual IP (VIP) address maps traffic received at one IP address to another address based on the destination port number in the TCP or UDP segment header. For example,

- An HTTP packet destined for 1.1.1.3:80 (that is, IP address 1.1.1.3 and port 80) might get mapped to a Web server at 10.1.1.10.
- An FTP packet destined for 1.1.1.3:21 might get mapped to an FTP server at 10.1.1.20.
- An SMTP packet destined for 1.1.1.3:25 might get mapped to a mail server at 10.1.1.30.

The destination IP addresses are the same. The destination port numbers determine the host to which the security device forwards traffic.

Figure 390: Virtual IP Address

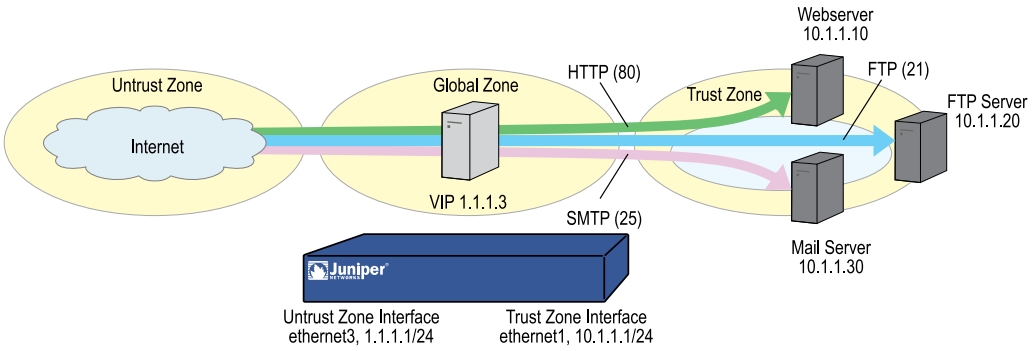


Table 105: Virtual IP Forwarding Table

Interface IP in Untrust Zone	VIP in Global Zone	Port	Forward to	Host IP in Trust Zone
1.1.1.1/24	1.1.1.3	80 (HTTP)		10.1.1.10
1.1.1.1/24	1.1.1.3	21 (FTP)		10.1.1.20
1.1.1.1/24	1.1.1.3	25 (SMTP)		10.1.1.30

The security device forwards incoming traffic destined for a VIP to the host with the address to which the VIP points. However, when a VIP host initiates outbound traffic, the security device only translates the original source IP address to another address if you have previously configured NAT on the ingress interface or NAT-src in a policy that applies to traffic originating from that host. Otherwise, the security device does not translate the source IP address on traffic originating from a VIP host.

You need the following information to define a virtual IP:

- The IP addresses for the servers that process the requests
- The type of service you want the security device to forward from the VIP to the IP address of the host

NOTE: You can only set a VIP on an interface in the Untrust zone.

Some notes about VIPs:

- You can use virtual port numbers for well-known services when running multiple server processes on a single machine. For example, if you have two FTP servers on the same machine, you can run one server on port 21 and the other on port 2121. Only those who know the virtual port number in advance and append it to the IP address in the packet header can gain access to the second FTP server.
- You can map predefined services and user-defined services. A single VIP can distinguish between custom services that have the same source and destination port numbers but different transports.
- You can configure VIPs and MIPs in any combination on a Layer 3 interface: VIPs and MIPs can use the same interface IP or the same IP on the same interface. If a conflict occurs, traffic bound to the device always is given priority over VIPs and MIPs.
- Custom services can use any destination port number or number range from 1 to 65,535, not just from 1024 to 65,535.
- A single VIP can support custom services with multiple port entries by creating multiple service entries under that VIP—one service entry in the VIP for each port entry in the service. By default, you can use single-port services in a VIP. To be able to use multiple-port services in a VIP, you must first issue the CLI command **set vip multi-port**, and then reset the security device. (See “Example: VIP with Custom and Multiple-Port Services” on page 1557.)

- The host to which the security device maps VIP traffic must be reachable from the trust-vr. If the host is in a routing domain other than that of the trust-vr, you must define a route to reach it.
- The IP address for the VIP can be in the same subnet as an interface or can have the same address as that interface

VIP and the Global Zone

Setting a VIP for an interface in the Layer 3 zone generates an entry in the Global zone address book. The Global zone address book keeps all the VIPs of all interfaces, regardless of the zone to which the interface belongs. You can use these VIP addresses as the destination address in policies between any two zones, and as the destination address in Global policies.

Example: Configuring Virtual IP Servers

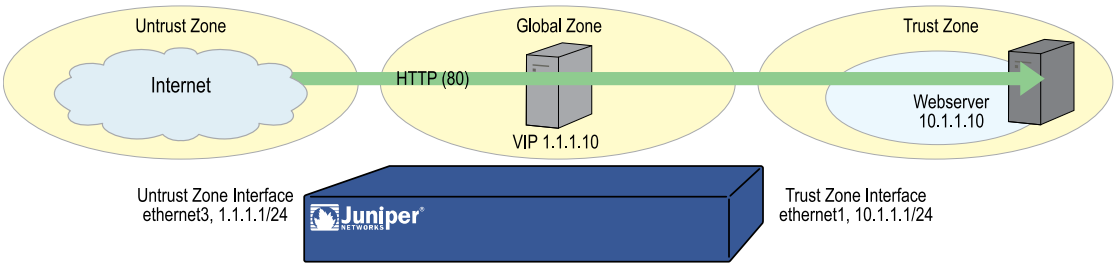
In this example, you bind interface ethernet1 to the Trust zone and assign it IP address 10.1.1.1/24. You bind interface ethernet3 to the Untrust zone and assign it IP address 1.1.1.1/24.

Then, you configure a VIP at 1.1.1.10 to forward inbound HTTP traffic to a Web server at 10.1.1.10, and you create a policy permitting traffic from the Untrust zone to reach the VIP—and consequently to the host with the address to which the VIP points—in the Trust zone.

Because the VIP is in the same subnet as the Untrust zone interface (1.1.1.0/24), you do not need to define a route for traffic from the Untrust zone to reach it. Also, no address book entry is required for the host to which a VIP forwards traffic. All security zones are in the trust-vr routing domain.

NOTE: If you want HTTP traffic from a security zone other than the Untrust zone to reach the VIP, you must set a route for 1.1.1.10 on the router in the other zone to point to an interface bound to that zone. For example, imagine that ethernet2 is bound to a user-defined zone, and you have configured a router in that zone to send traffic destined for 1.1.1.10 to ethernet2. After the router sends traffic to ethernet2, the forwarding mechanism in the security device locates the VIP on ethernet3, which maps it to 10.1.1.10, and sends it out ethernet1 to the Trust zone. This process is similar to that described in “Example: Reaching a MIP from Different Zones” on page 1538. You must also set a policy permitting HTTP traffic from the source zone to the VIP in the Trust zone.

Figure 391: Virtual IP Server



WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24
   Select the following, then click **OK**:
   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **VIP**

   Network > Interfaces > Edit (for ethernet3) > VIP: Enter the following address, then click **Add**:

   Virtual IP Address: 1.1.1.10

   Network > Interfaces > Edit (for ethernet3) > VIP > New VIP Service: Enter the following, then click **OK**:

   Virtual IP: 1.1.1.10
   Virtual Port: 80
   Map to Service: HTTP (80)
   Map to IP: 10.1.1.10

3. **Policy**

   Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), ANY
   Destination Address:
   Address Book Entry: (select), VIP(1.1.1.10)
   Service: HTTP
   Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **VIP**

```
set interface ethernet3 vip 1.1.1.10 80 http 10.1.1.10
```

3. **Policy**

```
set policy from untrust to trust any vip(1.1.1.10) http permit
save
```

Example: Editing a VIP Configuration

In this example, you modify the virtual IP (VIP) server configuration you created in the previous example. To restrict access to the Web server, you change the virtual port number for HTTP traffic from 80 (the default) to 2211. Now, only those that know to use port number 2211 when connecting to the Web server can access it.

WebUI

Network > Interfaces > Edit (for ethernet3) > VIP > Edit (in the VIP Services Configure section for 1.1.1.10): Enter the following, then click **OK**:

Virtual Port: 2211

CLI

```
unset interface ethernet3 vip 1.1.1.10 port 80
set interface ethernet3 vip 1.1.1.10 2211 http 10.1.1.10
save
```

Example: Removing a VIP Configuration

In this example, you delete the VIP configuration that you previously created and modified. Before you can remove a VIP, you must first remove any existing policies associated with it. The ID number for the policy that you created in “Example: Configuring Virtual IP Servers” on page 1554 is 5.

WebUI

Policies > (From: Untrust, To: Trust) > Go: Click **Remove** for policy ID 5.

Network > Interfaces > Edit (for ethernet3) > VIP: Click **Remove** in the VIP Configure section for 1.1.1.10.

CLI

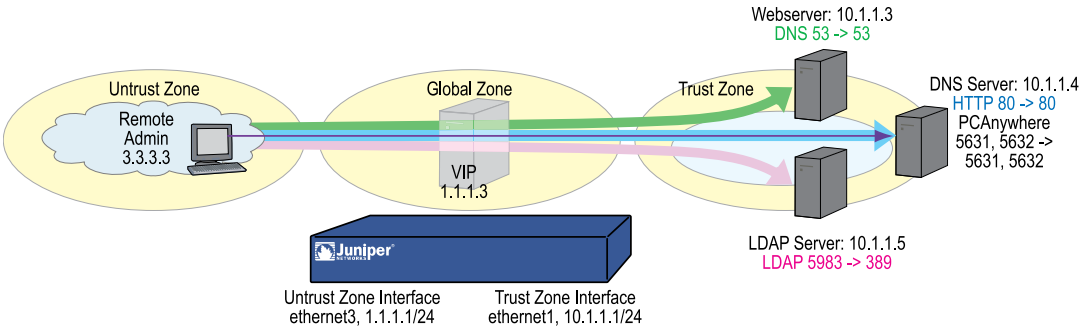
```
unset policy id 5
unset interface ethernet3 vip 1.1.1.10
save
```

Example: VIP with Custom and Multiple-Port Services

In the following example, you configure a VIP at 1.1.1.3 to route the following services to the following internal addresses:

Service	Transport	Virtual Port Number	Actual Port Number	Host IP Address
DNS	TCP, UDP	53	53	10.1.1.3
HTTP	TCP	80	80	10.1.1.4
PCAnywhere	TCP, UDP	5631, 5632	5631, 5632	10.1.1.4
LDAP	TCP, UDP	5983	389	10.1.1.5

Figure 392: VIP with Custom and Multiple-Port Services



The VIP routes DNS lookups to the DNS server at 10.1.1.3, HTTP traffic to the Web server at 10.1.1.4, and authentication checks to the database on the LDAP server at 10.1.1.5. For HTTP, DNS, and PCAnywhere, the virtual port numbers remain the same as the actual port numbers. For LDAP, a virtual port number (5983) is used to add an extra level of security to the LDAP authentication traffic.

For managing the HTTP server remotely, you define a custom service and name it PCAnywhere. PCAnywhere is a multiple-port service that sends and listens for data on TCP port 5631 and status checks on UDP port 5632.

You also enter the address of the Remote Admin at 3.3.3.3 in the Untrust zone address book, and configure policies from the Untrust zone to the Trust zone for all

the traffic that you want to use the VIPs. All security zones are in the trust-vr routing domain.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **Apply**:

   Zone Name: Trust
   Static IP: (select this option when present)
   IP Address/Netmask: 10.1.1.1/24

   Select the following, then click **OK**:

   Interface Mode: NAT

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   Zone Name: Untrust
   Static IP: (select this option when present)
   IP Address/Netmask: 1.1.1.1/24

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: Remote Admin
   IP Address/Domain Name:
   IP/Netmask: (select), 3.3.3.3/32
   Zone: Untrust

3. **Custom Service**

   Policy > Policy Elements > Services > Custom > New: Enter the following, then click **OK**:

   Service Name: PCAnywhere
   No 1:
   Transport protocol: TCP
   Source Port Low: 0
   Source Port High: 65535
   Destination Port Low: 5631
   Destination Port High: 5631
   No 2:
   Transport protocol: UDP
   Source Port Low: 0
   Source Port High: 65535
   Destination Port Low: 5632
   Destination Port High: 5632

4. **VIP Address and Services**

NOTE: To enable the VIP to support multiple-port services, you must use enter the CLI command **set vip multi-port**, save the configuration, and then reboot the device.

Network > Interfaces > Edit (for ethernet3) > VIP: click here to configure: Type **1.1.1.3** in the Virtual IP Address field, then click **Add**.

> New VIP Service: Enter the following, then click **OK**:

Virtual IP: 1.1.1.3
Virtual Port: 53
Map to Service: DNS
Map to IP: 10.1.1.3

> New VIP Service: Enter the following, then click **OK**:

Virtual IP: 1.1.1.3
Virtual Port: 80
Map to Service: HTTP
Map to IP: 10.1.1.4

> New VIP Service: Enter the following, then click **OK**:

Virtual IP: 1.1.1.3
Virtual Port: 5631
Map to Service: PCAnywhere
Map to IP: 10.1.1.4

NOTE: For multiple-port services, enter the lowest port number of the service as the virtual port number.

> New VIP Service: Enter the following, then click **OK**:

Virtual IP: 1.1.1.3
Virtual Port: 5983
Map to Service: LDAP
Map to IP: 10.1.1.5

NOTE: Using nonstandard port numbers adds another layer of security, preventing common attacks that check for services at standard port numbers.

5. **Policies**

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), VIP(1.1.1.3)

Service: DNS
Action: Permit

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), VIP(1.1.1.3)
Service: HTTP
Action: Permit

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), VIP(1.1.1.3)
Service: Custom-LDAP
Action: Permit

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Remote Admin
Destination Address:
Address Book Entry: (select), VIP(1.1.1.3)
Service: PCAnywhere
Action: Permit

CLI

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Address**

```
set address untrust “Remote Admin”  3.3.3.3/32
```

3. **Custom Service**

```
set service pcanywhere protocol udp src-port 0-65535 dst-port 5631-5631
set service pcanywhere + tcp src-port 0-65535 dst-port 5632-5632
set service custom-ldap protocol tcp src-port 0-65535 dst-port 5983-5983
```

4. **VIP Address and Services**

```
set vip multi-port
save
reset
```

```
System reset, are you sure? y/[n]
set interface ethernet3 vip 1.1.1.3 53 dns 10.1.1.3
set interface ethernet3 vip 1.1.1.3 + 80 http 10.1.1.4
set interface ethernet3 vip 1.1.1.3 + 5631 pcanywhere 10.1.1.4
set interface ethernet3 vip 1.1.1.3 + 5983 ldap 10.1.1.5
```

NOTE: For multiple-port services, enter the lowest port number of the service as the virtual port number.

5. **Policies**

```
set policy from untrust to trust any vip(1.1.1.3) dns permit
set policy from untrust to trust any vip(1.1.1.3) http permit
set policy from untrust to trust any vip(1.1.1.3) custom-ldap permit
set policy from untrust to trust “Remote Admin”  vip(1.1.1.3) pcanywhere permit
save
```

NAT—dst Port Range Mapping

Instead of mapping individual ports between virtual IP and real server IP, you can map a range of ports between them by using the port-range VIP entry feature. You can enable this feature by using the **set interface** command:

set interface *interface* vip { *ip_address* | *interface_ip* } port-range *port1 – port2* server-ip *ip-address2* port-range *portx - porty* [protocol TCP | UDP] [manual]

The port-range VIP entry is considered a single entry. The range of ports is from 1 – 65535.

For example, to map ports from 3 to 20, to ports 43 to 60, using IP address 10.10.10.100 and server IP 10.42.62.100

```
set interface ethernet3 vip 10.10.10.100 port-range 3-20 server-ip 10.42.62.100
  port-range 43-60
```

NOTE: This feature does not support ALG.

The virtual port-range size must be the same size as the real port. Port ranges can also be configured within the same virtual IP.

The following table lists an example where IPs 10.10.10.100:4500 are always mapped to IPs 20.20.20.110:5500.

Original Packet	Mapped Packet
-> 10.10.10.10/4500	-> 20.20.20.110/5500
-> 10.10.10.10/4501	-> 20.20.20.110/5501

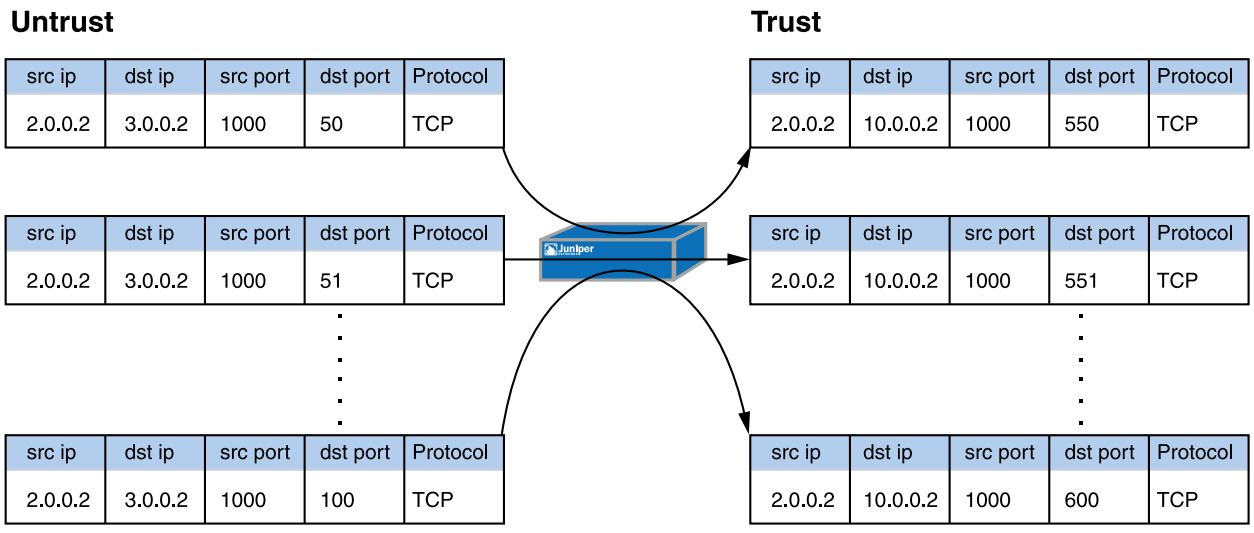
Original Packet	Mapped Packet
…	…
- > 10.10.10.10/4600	- > 20.20.20.110/5600

…

- > 10.10.10.100/4500	- > 20.20.20.200/5500
- > 10.10.10.100/4501	- > 20.20.20. 200/5501
…	…
- > 10.10.10.100/4600	- > 20.20.20. 200/5600

Figure 393 on page 1562 shows how multiple ports can be mapped using one CLI command.

Figure 393: NAT—dst Port Range Mapping with VIP




```
set  interface ethernet3 vip 3.0.0.2 50-100 10.0.0.2 550-600
```

Part 9
User Authentication

User Authentication describes the methods in ScreenOS for authenticating different types of users. It provides an introduction to user authentication, presents the two locations that can store user profiles—the internal database and an external authentication server—and provides numerous examples for configuring authentication, IKE, XAuth, and L2TP users and user groups. Some other aspects of user authentication are also covered, such as changing login banners, creating multiple-type users (such as an IKE/XAuth user, for example), and using group expressions in policies applying authentication.

This guide contains the following chapters:

- "Authentication" on page 1565 details the various authentication methods and uses that ScreenOS supports.
- "Authentication Servers" on page 1577 presents the options of using one of four possible types of external authentication server—RADIUS, SecurID, TACACS+, or LDAP—or the internal database and shows how to configure the security device to work with each type.
- "Infranet Authentication" on page 1607 details how the security device is deployed in a unified access control (UAC) solution. Juniper Networks UAC secures and ensures the delivery of applications and services across an enterprise infranet.
- "Authentication Users" on page 1615 explains how to define profiles for authentication users and how to add them to user groups stored either locally or on an external RADIUS authentication server.
- "IKE, XAuth, and L2TP Users" on page 1637 explains how to define IKE, XAuth, and L2TP users. Although the XAuth section focuses primarily on using the security device as an XAuth server, it also includes a subsection on configuring select security devices to act as an XAuth client.
- "Extensible Authentication for Wireless and Ethernet Interfaces" on page 1661 explains the options available for and examples of how to use the Extensible Authentication Protocol to provide authentication for Ethernet and wireless interfaces.

Chapter 46

Authentication

After a general introduction to the different types of authentication that are available for different types of network users, this chapter contains a brief section on admin user authentication. It then provides information on combining different user types, the use of group expressions, and how to customize the banners that appear on HTTP, FTP, L2TP, Telnet, and XAuth login prompts. The final section describes how to create a large, 4Kbyte banner that pre-empts all individually defined administrative access and firewall authentication banners. This chapter contains the following sections:

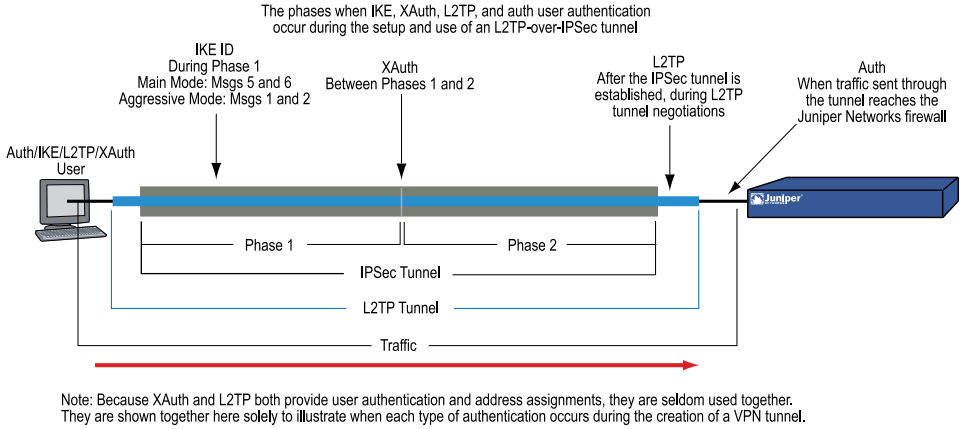
User Authentication Types

The following chapters describe the different types of users and user groups that you can create and how to use them when configuring policies, IKE gateways, and L2TP tunnels:

- "Authentication Users" on page 1615
- "IKE Users and User Groups" on page 1637
- "XAuth Users and User Groups" on page 1640
- "L2TP Users and User Groups" on page 1656

The security device authenticates the different types of users at different stages in the connection process. IKE, XAuth, L2TP, and auth user authentication techniques occur at different times during the creation of an L2TP-over-IPsec VPN tunnel. See Figure 394 on page 1566.

Figure 394: Authentication During L2TP-over-IPsec VPN Tunnel



Admin Users

Admin users are the administrators of a security device. There are five kinds of admin users:

- Root admin
- Root-level read/write admin
- Root-level read-only admin
- Vsys admin
- Vsys read-only admin

NOTE: For information about the privileges of each type of admin user and for examples of the creation, modification, and removal of admin users, see “Administration” on page 309 .

Although the profile of the root user of a security device must be stored in the local database, you can store vsys users and root-level admin users with read/write and read-only privileges either in the local database or on an external auth server.

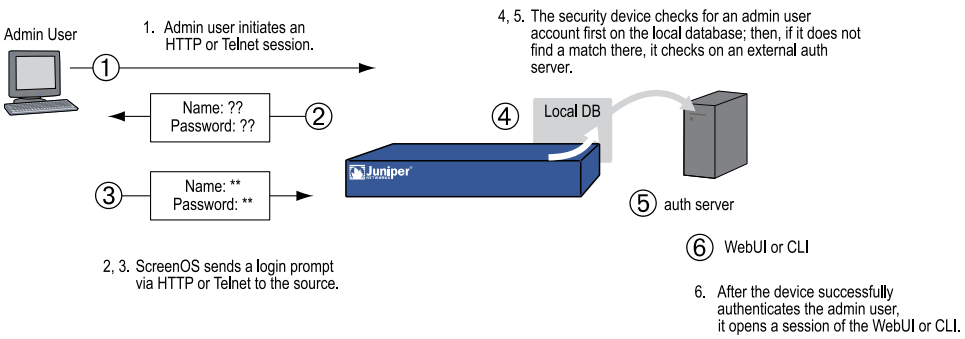
If you store admin user accounts on an external RADIUS auth server and you load the RADIUS dictionary file on the auth server, you can elect to query admin privileges defined on the server. Optionally, you can specify a privilege level to be applied globally to all admin users stored on that auth server. You can specify either read/write or read-only privileges. If you store admin users on an external SecurID or LDAP auth server, or on a RADIUS server without the RADIUS dictionary file, you cannot

define their privilege attributes on the auth server. Therefore, you must assign a privilege level to them on the security device.

If set on the security device:	And the RADIUS server is loaded with the RADIUS dictionary file, then:	And a SecurID, an LDAP, or a RADIUS server without the RADIUS dictionary file, then:
Get privileges from RADIUS server	Assign appropriate privileges	Root-level or vsys-level admin login fails
Assign read/write privileges to external admin	Assign root-level or vsys-level read/write privileges	Assign root-level read/write privileges Vsys admin login fails
Assign read-only privileges to external admin	Assign root-level or vsys-level read-only privileges	Assign root-level read-only privileges Vsys admin login fails

Figure 395 on page 1567 shows the admin authentication process.

Figure 395: Admin Authentication Process



Handling Admin Authentication Failures

You must be a root admin user to configure this feature. To minimize the chances of an unauthorized user will log into a device, you can limit the number of unsuccessful login attempts allowed and lock the unauthorized user's account for a specified period if the unsuccessful login attempts exceed the limit.

This restriction also protects against certain types of attacks such as automated dictionary attacks. By default, the device allows up to three unsuccessful login attempts and has a lockout time of one minute. The security device automatically unlocks the locked user account after the period expires. When the lockout time is set to **0**, the security device locks the user account permanently. However, a root administrator or a read-write security administrator can unlock the account.

In this example, you set the maximum number of authentication failures **5** and the user account lockout time to **60** minutes.

To set the number of login attempts and the lockout time:

WebUI

Configuration > Admin > Management: Enter the following, then click **Apply:**
Max Login Attempts: 5
Lock the Admin accounts on authentication failure: 60

CLI

```
set admin access attempts 5
set admin access lock-on-failure 60
save
```

When the authentication attempt fails for the fifth successive time, the security device prevents the admin user from accessing the device and locks the user account for 60 minutes. You can lock the user account for a maximum of 1440 minutes.

NOTE: Only a root admin can set the maximum number of login attempts. However, a root admin or a read-write admin with security role attribute can unlock the locked user account.

Clearing the Admin Lock

Only a root administrator can unlock a locked admin user's account. To unlock the user account:

WebUI

Configuration > Admin > Administrators: In the Admin Name field, enter the username, then click Clear.

CLI

```
clear admin lock name_str
```

Multiple-Type Users

You can combine auth, IKE, L2TP, XAuth users to create the following combinations to store on the local database:

- Auth/IKE user
- Auth/IKE/XAuth user
- Auth/L2TP user
- IKE/XAuth user

- Auth/IKE/L2TP user
- L2TP/XAuth user
- IKE/L2TP user
- IKE/L2TP/XAuth user
- Auth/XAuth user
- Auth/IKE/L2TP/XAuth user

Although you can make all of the above combinations when defining multiple-type user accounts on the local database, consider the following points before creating them:

- Combining an IKE user type with any other user type limits the potential to scale. You must store an IKE user account on the local database. If you create auth/IKE, IKE/L2TP, and IKE/XAuth user accounts and then the number of users grows beyond the capacity of the local database, you will not be able to relocate these accounts to an external auth server. If you separate IKE user accounts from other types of accounts, you have the flexibility to move the non-IKE user accounts to an external auth server should the need arise to do so.
- L2TP and XAuth provide the same services: remote user authentication and IP, DNS server, and WINS server address assignments. It is not recommended to use L2TP and XAuth together for an L2TP-over-IPsec tunnel. Not only do the two protocols accomplish the same goals, but the L2TP address assignments overwrite the XAuth address assignments after Phase 2 IKE negotiations complete and L2TP negotiations take place.
- If you create a multiple-type user account on the local database combining auth/L2TP or auth/XAuth, the same username and password must be used for both logins.

  Although it is more convenient to create a single multiple-type user account, separating the user types into two single accounts allows you to increase security. For example, you can store an auth user account on an external auth server and an XAuth user account on the local database. You can then assign different login usernames and passwords to each account and reference the XAuth user in the IKE gateway configuration and the auth user in the policy configuration. The dialup VPN user must authenticate himself twice, potentially with two completely different usernames and passwords.

Group Expressions

A group expression is a statement that you can use in policies to conditionalize the requirements for authentication. Group expressions allow you to combine users, user groups, or other group expressions as alternatives for authentication (“a” OR “ b”), or as requirements for authentication (“ a” AND “ b”). You can also use group expressions to exclude a user, user group, or another group expression (NOT “ c”).

NOTE: Although you define group expressions on the security device (and store them on the local database), the users and user groups that you reference in the group expressions must be stored on an external RADIUS server. A RADIUS server allows a user to belong to more than one user group. The local database does not permit this.

Group expressions make use of the three operators OR, AND, and NOT. The objects in the expression to which OR, AND, and NOT relate can be an auth user, an auth user group, or a previously defined group expression. Table 106 on page 1570 lists objects, group expressions, and examples.

Table 106: Group Expression Examples

Object	Expression	Example
Users	OR	A policy specifies that the user be *a* OR *b*, so the security device authenticates if the user matches either condition *a* or *b*.
	AND	AND in a group expression requires that at least one of the two expression objects be either a user group or a group expression. (It is illogical to require a user to be user *a* AND user *b*.) If the authentication aspect of a policy requires that the user be *a* AND a member of group *b*, then the security device authenticates the user only if those two conditions are met.
	NOT	A policy specifies that the user be anyone except user *c* (NOT *c*), then the security device authenticates as long as the user is not *c*.
User groups	OR	A policy specifies that the user belong to group *a* OR group *b*, so the security device authenticates if the user belongs to either group.
	AND	A policy requires that the user belong to group *a* AND group *b*, so the security device authenticates the user only if he or she belongs to both groups.
	NOT	A policy specifies that the user belong to any group other than group “c” (NOT “ c”), so the security device authenticates the user as long as the user does not belong to that group.
Group expressions	OR	A policy specifies that the user fit the description of group expression *a* OR group expression *b*, so the security device authenticates the user if either group expression applies.
	AND	A policy specifies that the user fit the description of group expression *a* AND group expression *b*, so the security device allows authentication only if both group expressions apply to the user.
	NOT	A policy specifies that the user not fit the description of group expression *c* (NOT *c*), so the security device allows authentication only if the user does not fit that group expression.

Example: Group Expressions (AND)

In this example, you create a group expression “s + m” that states “ sales AND marketing” . You have previously created the auth user groups “ sales” and “ marketing” on an external RADIUS auth server named “ radius1” and populated them with users. (For an example on how to configure an external RADIUS auth server, see “Example: RADIUS Auth Server” on page 1597.) You then use that group expression in an intrazone policy whose authentication component requires a user be a member of both user groups to be able to access the confidential contents on a server named “ project1” (10.1.1.70).

NOTE: For an intrazone policy to work properly, the source and destination addresses must be in different subnets connected to the security device through interfaces that are both bound to the same zone. There cannot be any other routing device beside the security device that can route traffic between the two addresses. For more information about intrazone policies, see “Policies” on page 197.

WebUI

1. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: project1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.70/32
   Zone: Trust

2. **Group Expression**

   Policy > Policy Elements > Group Expressions > New: Enter the following, then click **OK**:

   Group Expression: s+m
   AND: (select), sales AND marketing

3. **Policy**

   Policy > Policies > (From: Trust, To: Trust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), project1
   Service: ANY
   Action: Permit
   Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

```
Authentication: (select)
Auth Server: (select)
   Use: radius1
Group Expression: (select), External Group Expression - s+m
```

CLI

1. **Address**

```
set address trust project1 10.1.1.70/32
```

2. **Group Expression**

```
set group-expression s+m sales and marketing
```

3. **Policy**

```
set policy top from trust to trust any project1 any permit auth server radius1
group-expression s+m
save
```

Example: Group Expressions (OR)

In this example, you create a group expression “a/b” that states “ amy OR basil” . You have previously created auth user accounts “ amy” and “ basil” on an external RADIUS auth server named “ radius1.” (For an example on how to configure an external RADIUS auth server, see “Example: RADIUS Auth Server” on page 1597.) You then use that group expression in a policy from the Trust zone to the DMZ. The authentication component of the policy requires the user to be either amy or basil to be able to access the Web server named “ web1” at 210.1.1.70.

WebUI

1. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

```
Address Name: web1
IP Address/Domain Name
      IP/Netmask: (select), 210.1.1.70/32
Zone: DMZ
```

2. **Group Expression**

Policy > Policy Elements > Group Expressions > New: Enter the following, then click **OK**:

```
Group Expression: a/b
```

OR: (select), amy OR basil

3. **Policy**

Policy > Policies > (From: Trust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), web1
Service: ANY
Action: Permit
Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
Auth Server: (select)
Use: radius1
Group Expression: (select), External Group Expression - a/b

CLI

1. **Address**

```
set address trust project1 210.1.1.70/32
```

2. **Group Expression**

```
set group-expression a/b amy or basil
```

3. **Policy**

```
set policy top from trust to dmz any web1 any permit auth server radius1
group-expression
 a/b
save
```

Example: Group Expressions (NOT)

In this example, you create a group expression "-temp" that states " NOT temp" . You have previously created a local auth user group " temp" on an external RADIUS auth server named " radius1." (For an example on how to configure an external RADIUS auth server, see "Example: RADIUS Auth Server" on page 1597.) You then use that group expression in a policy from the Trust zone to the Untrust zone that allows Internet access to all full-time employees, but not to temporary contractors. The authentication component of the policy requires everyone in the Trust zone to be authenticated except the users in " temp," who are denied access to the Untrust zone.

WebUI

1. **Group Expression**

   Policy > Policy Elements > Group Expressions > New: Enter the following, then click **OK**:

   Group Expression: -temp
   OR: (select), NOT temp

2. **Policy**

   Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: HTTP
   Action: Permit
   Position at Top: (select)

   > Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   Authentication: (select)
   Auth Server: (select)
   Use: Local
   Group Expression: (select), External Group Expression - -temp

CLI

1. **Group Expression**

   set group-expression -temp not temp

2. **Policy**

   set policy top from trust to untrust any any any permit auth server radius1
   group-expression -temp
   save

Banner Customization

A banner is a message that appears on a monitor in different places depending on the type of login:

- At the top of a Telnet or console display when an admin user connects to the security device

NOTE: You can include an additional banner line under a Telnet or console banner. The second banner line remains the same for both Telnet and console login displays although the Telnet banner can differ from the console banner. To create a secondary banner, enter the following command: **set admin auth banner secondary** string.

- At the top of a browser screen after an auth user has successfully logged into a WebAuth address
- Before or after a Telnet, an FTP, or an HTTP login prompt, success message, and fail message for auth users

All of the banners, except that for a console login, already have default messages. You can customize the messages that appear on the banners to better suit the network environment in which you use the security device.

Example: Customizing a WebAuth Banner

In this example, you change the message that appears in the browser to indicate that an auth user has successfully authenticated himself after successfully logging in via WebAuth. The new message is "Authentication approved."

WebUI

Configuration > Admin > Banners > WebAuth: In the Success Banner field, type **Authentication approved**, then click **Apply**.

CLI

```
set webauth banner success "Authentication approved"
save
```

Login Banner

The size of the login banner is increased to a maximum of 4Kbytes. This provides space for terms of use statements, which are presented before administrators and authenticated users log into the security device and into protected resources behind the device. The login banner is a clear text ASCII file you create and store on the security device, the file must be called **usrterms.txt**. You activate the banner by restarting of the system. If the banner file is greater than 4Kbytes, the security device will not accept it and will continue using existing banners entered through the CLI and the WebUI.

When activated, the login banner is used globally by the root system and all virtual systems (vsys). You cannot differentiate or customize between or within a vsys. The login banner pre-empts all individually defined administrative access banners and firewall authentication banners. After entering a username and password, the user must click the **Login** button. Pressing the **Enter** key will not log the user into the device.

Example: Creating a Login Banner

Use the SCP utility to securely copy the banner file to the security device. With the following command, an administrator with username **netscreen** copies the banner file **my_large_banner.txt** to a security device at IP address 1.1.1.2. The banner file must be saved on the security device as **usrterms.txt**.

```
linux:~#scp my_large_banner.txt netscreen@1.1.1.2:useterms.txt
```

You must restart the device to activate the new banner. To modify the banner file, create a new file and overwrite the existing one with the new one.

To remove the banner, issue the following command on the security device:

```
device-> delete file usrterms.txt
```

This disables the login banner feature after you restart the device.

Chapter 47
Authentication Servers

This chapter examines different kinds of authentication servers—the local database built into every security device, and external RADIUS, SecurID, and LDAP authentication servers. This chapter includes the following sections:

Authentication Server Types

You can configure the security device to use the local database or one or more external authentication servers to verify the identities of the following types of users:

- Auth
- IKE
- L2TP
- XAuth
- Admin
- 802.1x

NOTE: IKE user accounts must be stored on the local database. The only external server to support L2TP and XAuth remote setting assignments and admin privilege assignments is RADIUS.

In addition to its local database, a security device supports external RADIUS, SecurID, LDAP, and TACACS+ servers. You can use each kind of authentication server to authenticate auth users, L2TP users, XAuth users, and admin users. ScreenOS also supports WebAuth, an alternative authentication scheme for auth users. (For a

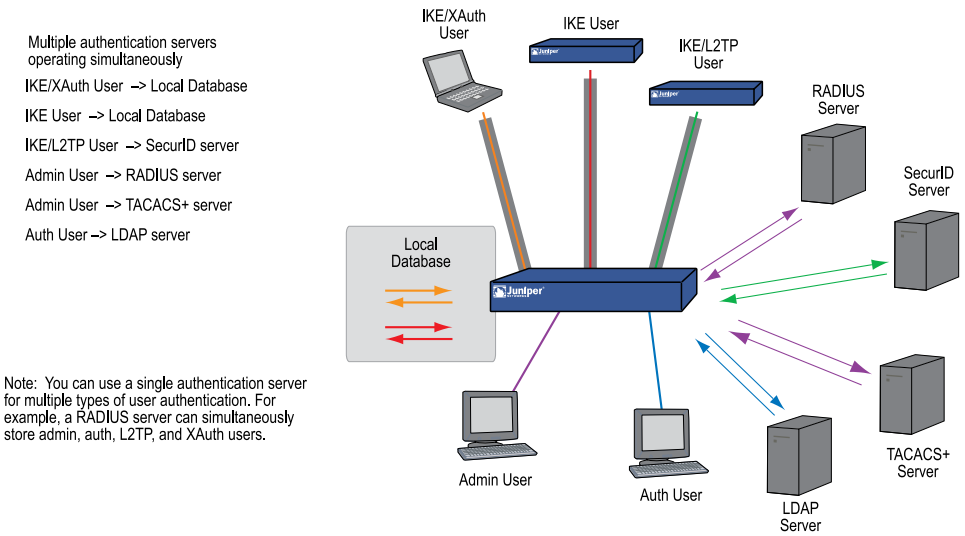
WebAuth example, see “Example: WebAuth + SSL Only (External User Group)” on page 1633.) Any auth server that contains auth user account types is eligible to be the default WebAuth auth server. Table 107 on page 1578 lists supported servers types and authentication features.

Table 107: Authentication Server Type, User Types, and Features

	Supported User Types and Features									
			L2TP Users		XAuth Users		Admin Users			
Server Type	Auth Users	IKE Users	Auth	Remote Settings	Auth	Remote Settings	Auth	Privileges	User Groups	Group Expressions
Local	X	X	X	X	X	X	X	X	X	
RADIUS	X		X	X	X	X	X	X	X	X
SecurID	X		X		X		X			
LDAP	X		X		X		X			
TACACS+							X	X		

On most Juniper Networks security devices, you can simultaneously employ up to 10 primary authentication servers per system—root system and virtual system—in any combination of types. This total includes the local database and excludes backup authentication servers. A RADIUS or LDAP server supports two backup servers, and a SecurID server supports one backup server; so, for example, you might use the local database and nine different primary RADIUS servers, with each RADIUS server having two backup servers assigned to it. See Figure 396 on page 1578.

Figure 396: Types of Authentication Servers

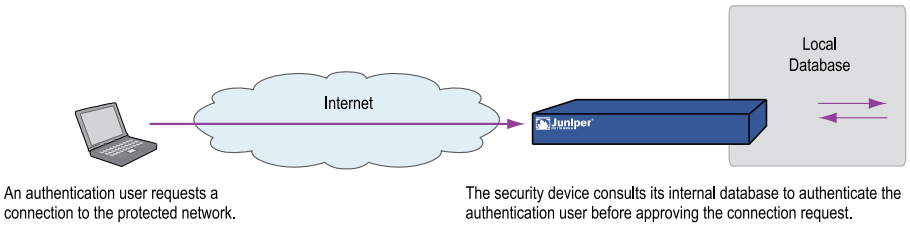


The following sections explain the local database and each authentication server in detail.

Local Database

All Juniper Networks security devices support a built-in user database for authentication. When you define a user on the security device, the security device enters the username and password in its local database. See Figure 397 on page 1579.

Figure 397: Local Authentication



The local database supports the following types of users and authentication features:

- Users:
  - Auth
  - IKE
  - L2TP
  - XAuth
  - Admin
  - 802.1x
- Authentication features:
  - Admin privileges
  - WebAuth
  - User groups
  - Group expressions

NOTE: You define the group expressions on the security device, but the users and user groups must be stored on an external RADIUS auth server. For more information about group expressions, see “Group Expressions” on page 1569.

The local database is the default authentication server (auth server) for all types of authentication. For instructions on how to add users and user groups to the local database via the WebUI and CLI, see “Authentication Users” on page 1615 and “IKE, XAuth, and L2TP Users” on page 1637.

Example: Local Database Timeout

By default, the local database authentication timeout for both admins and auth users is 10 minutes. In this example, you change it to never time out for admins and to time out after 30 minutes for auth users.

WebUI

Configuration > Admin > Management: Clear the Enable Web Management Idle Timeout check box, then click **Apply**.
Configuration > Auth > Servers > Edit (for Local): Enter **30** in the Timeout field, then click **Apply**.

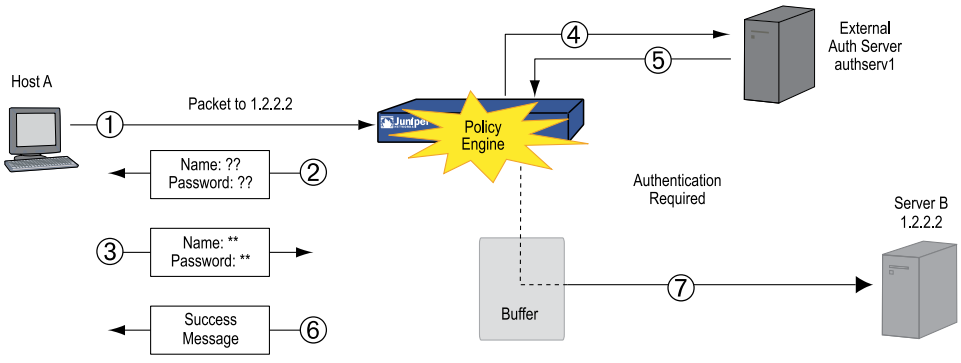
CLI

```
set admin auth timeout 0
set auth-server Local timeout 30
save
```

External Authentication Servers

A security device can connect to one or more external authentication servers, or *auth servers*, on which you store user accounts. When the security device receives a connection request that requires authentication verification, the security device requests an authentication check from the external auth server specified in the policy, L2TP tunnel configuration, or IKE gateway configuration. The security device then acts as a relay between the user requesting authentication and the auth server granting authentication. Figure 398 on page 1580 shows the steps to a successful authentication check by an external auth server.

Figure 398: External Auth Server



1. Host A sends an FTP, an HTTP, or a Telnet TCP SYN packet to 1.2.2.2.
2. The security device intercepts the packet, notes that its corresponding policy requires authentication from authserv1, buffers the packet, and prompts the user for a username and password.
3. The user replies with a username and password.

4. The security device relays the login information to authserv1.
5. Authserv1 sends back a notification of success to the security device.
6. The security device informs the auth user of his or her login success.
7. The security device then forwards the packet from its buffer to its destination of 1.2.2.2.

Auth Server Object Properties

A security device treats each auth server as an object that it can reference in policies, IKE gateways, and L2TP tunnels. The properties described in Table 108 on page 1581define and uniquely identify an auth server object.

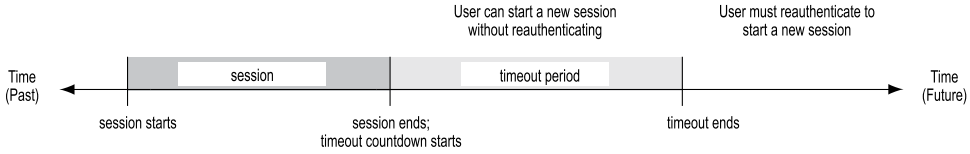
Table 108: Auth Server Object Properties

Property	Description	
Object name	A name string, such as authserv1. (The only predefined auth server is Local.)	
ID number	You can set the ID number or allow the security device to set it automatically. If you set an ID number, you must choose one that is not already in use.	
Type	RADIUS, SecurID, LDAP, TACACS+.	
Server name	The IP address or domain name of the server.	
Backup1	The IP address or domain name of a primary backup server.	
Backup2	The IP address or domain name of a secondary backup server.	
Account Type	One or more of the following types of users: Auth, L2TP, 802.1x, XAuth; or Admin by itself.	
Timeout value	The timeout value is idle timeout, and takes on a different meaning if it is for an auth user or if it is for an admin user.	
	Auth user	The timeout countdown begins after the first authenticated session completes. If the user initiates a new session before the countdown reaches the timeout threshold, the timeout countdown resets. The default timeout value is 10 minutes, the maximum is 255 minutes. To disable the timeout feature, set the timeout value to 0. See Figure 399 on page 1582.
	Admin user	If the length of idle time reaches the timeout threshold, the security device terminates the admin session. To continue managing the security device, the admin must reconnect to the device and reauthenticate himself. The default timeout value is 10 minutes, the maximum is 1000 minutes. To disable the timeout feature, set the timeout value to 0. See Figure 400 on page 1582.

Table 108: Auth Server Object Properties *(continued)*

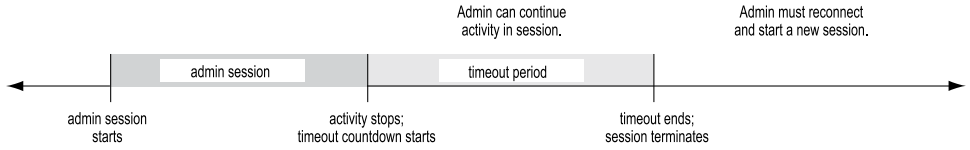
Property	Description
Forced Timeout	Forced timeout, unlike idle timeout, does not depend on the idleness of the user, but on an absolute timeout after which access for the authenticated user is terminated. The auth table entry for the user is removed, as are all associated sessions for the auth table entry. The default is 0 (disabled), the range is 0 to 10000 (6.9 days).

Figure 399: Auth Server Object Properties



NOTE: User authentication timeout is not the same as session idle timeout. If no activity occurs in a session for a predefined length of time, the security device automatically removes the session from its session table.

Figure 400: Admin Timeout Property



In addition to the above properties that apply to all auth server objects, each server has a few others specific to itself. These are explained in "Auth Server Types" on page 1582

Auth Server Types

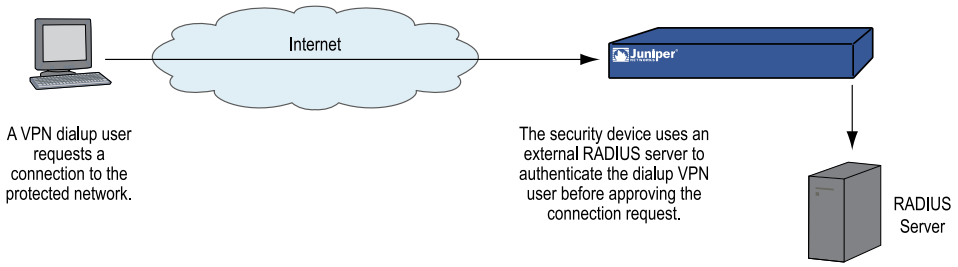
In addition to the internal database, security devices support four types of external auth servers:

- Remote Authentication Dial-In User Service (RADIUS)
- SecurID
- Lightweight Directory Access Protocol (LDAP)
- Terminal Access Controller Access Control System Plus (TACACS+)

Remote Authentication Dial-In User Service

The Remote Authentication Dial-In User Service (RADIUS) is a protocol for an authentication server that can support up to tens of thousands of users.

Figure 401: Using RADIUS as an External Auth Server



The RADIUS client (that is, the security device) authenticates users through a series of communications between the client and the server. Basically, RADIUS asks the person logging in to enter his or her username and password. It then compares these values to those in its database, and once a user is authenticated, the client provides the user with access to the appropriate network services.

To configure the security device for RADIUS, you must specify the IP address of the RADIUS server and define a shared secret—the same as that defined on the RADIUS server. The shared secret is a password the RADIUS server uses to generate a key to encrypt traffic between the security and RADIUS devices.

RADIUS Auth Server Object Properties

In addition to the generic auth server properties listed in “Auth Server Object Properties” on page 1581, a RADIUS server also makes use of the properties described in Table 109 on page 1583.

Table 109: Radius Auth Server Object Properties

Property	Description
Shared Secret	The secret (password) shared between the security device and the RADIUS server. The devices use this secret to encrypt the user’s password that it sends to the RADIUS server.
RADIUS Port	The port number on the RADIUS server to which the security device sends authentication requests. The default port number is 1645.
RADIUS Retry Timeout	The interval (in seconds) that the security device waits before sending another authentication request to the RADIUS server if the previous request does not elicit a response. The default is three seconds.

Supported User Types and Features

A RADIUS server supports the following types of users and authentication features:

- Auth users
- L2TP users (authentication and remote settings)
- Admin users (authentication and privilege assignments)

- User groups
- XAuth users (authentication and remote settings)

  The XAuth module provides support for the Session-timeout and Idle-timeout attributes retrieved from the RADIUS server described in Table 110 on page 1584.

Table 110: XAuth Attribute Support

Attribute	Description
Session-timeout	If the Session-timeout attribute is a non-zero value, the phase-1/phase-2 security association (SA) and the XAuth are both terminated when the timeout value is reached.
Idle-timeout	If the Idle-timeout attribute is a non-zero value, it takes preference over the local phase-2 SA idle time configuration and member SA hold time. If the Idle-timeout value is 0, the local phase-2 SA idle time and member SA hold time is used.
XAuth Accounting Start	The XAuth Accounting start is sent to the external RADIUS server after the user is authenticated correctly.
XAuth Accounting Stop	The XAuth Accounting stop is sent to the external RADIUS server when the XAuth connection is torn down. All phase-1/phase-2 SA and XAuth connection is terminated under the following conditions: ■ RADIUS server Session-timeout attribute is reached ■ RADIUS server Session-timeout attribute is not configured The XAuth session lifetime is used instead. ■ RADIUS server Idle-timeout attribute is reached on all Phase-2 SAs ■ Client disconnect is detected via dead peer detection (DPD) or heartbeat. ■ Locally configured phase-2 SA idle time or the member SA hold time is reached because RADIUS server is not providing an Idle-timeout attribute.

A RADIUS server can support all of the user types and features that the local database supports except IKE users. Among the four types of external auth servers, RADIUS is the only one at this time with such broad support. For a RADIUS server to support such ScreenOS-specific attributes as admin privileges, user groups, and remote L2TP and XAuth IP address, and DNS and WINS server address assignments, you must load a RADIUS dictionary file that defines these attributes onto the RADIUS server.

NOTE: ScreenOS uses the standard RADIUS attribute for IP address assignments. If you only want to use RADIUS for IP address assignments, you do not have to load the ScreenOS vendor-specific attributes (VSAs).

RADIUS Dictionary File

A dictionary file defines vendor-specific attributes (VSAs) that you can load onto a RADIUS server. After defining values for these VSAs, ScreenOS can then query them when a user logs into a security device. ScreenOS VSAs include admin privileges, user groups, and remote L2TP and XAuth IP address, and DNS and WINS server address assignments. There are two RADIUS dictionary files, one for Cisco RADIUS servers and one for Funk Software RADIUS servers. If you are using a Microsoft RADIUS server, there is no dictionary file. You must configure it as outlined in Bi-Directional Remote VPN using xAuth and Firewall Authentication with Microsoft Internet Authentication Service (IAS), which you can download from http://kb.juniper.net/kb/documents/public/kbdocs/ns10382/ns10382.pdf

Each RADIUS dictionary file contains the specific information described in Table 111 on page 1585.

Table 111: RADIUS Dictionary File Contents

Field	Description
Vendor ID	The ScreenOS vendor ID (VID; also called an "IETF number") is 3224. The VID identifies a specific vendor for a particular attribute. Some types of RADIUS server require you to enter the VID for each attribute entry, while other types only require you to enter it once and then apply it globally. Refer to your RADIUS server documentation for further information.
Attribute Name	The attribute names describe individual ScreenOS-specific attributes, such as NS-Admin-Privilege, NS-User-Group, NS-Primary-DNS-Server, and so on.
Attribute Number	The attribute number identifies an individual vendor-specific attribute. ScreenOS-specific attribute numbers fall into two ranges: ■ ScreenOS: 1 – 199 ■ Global PRO: 200 and above For example, the ScreenOS attribute number for user groups is 3. The Global PRO attribute number for user groups is 200.
Attribute Type	The attribute type identifies the form in which attribute data (or "value") appears—a string, an IP address, or an integer.

The RADIUS server automatically receives the above information when you load the RADIUS dictionary file onto it. To make new data entries, you must manually enter a value in the form indicated by the attribute type. For example, an entry for a read-write admin appears as follows:

VID	Attribute Name	Attribute Number	Attribute Type	Value
3224	NS-Admin-Privileges	1	data = int4 (i.e., integer)	2 (2 = all privileges)

To download a dictionary file, go to
http://www.juniper.net/customers/csc/research/netscreen_kb/downloads/dictionary/funk_radius.zip
or
http://www.juniper.net/customers/csc/research/netscreen_kb/downloads/dictionary/cisco_radius.zip

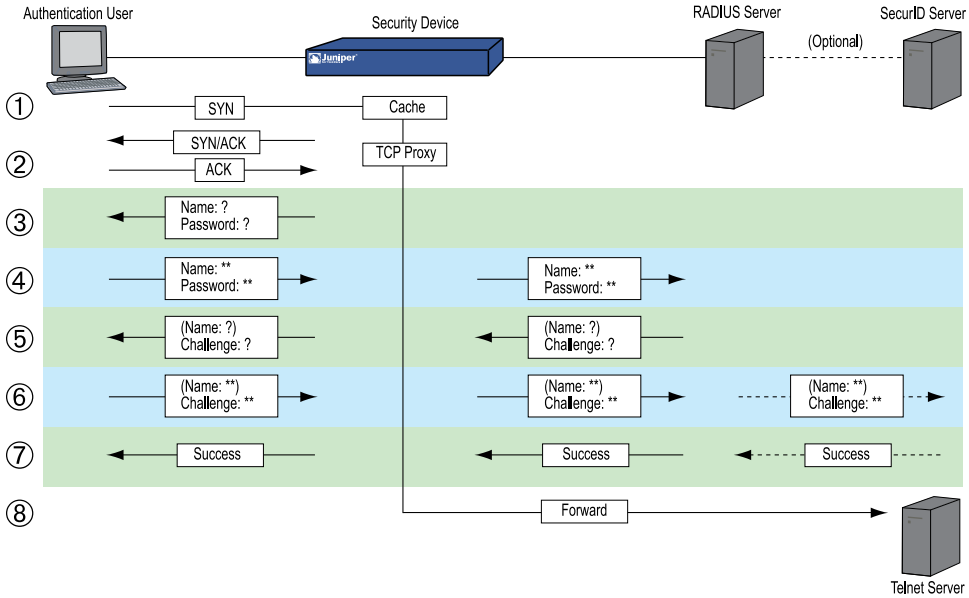
Log in and save the file to a local drive.

NOTE: All new installations of Funk Steel Belted RADIUS has the RADIUS firewall dictionary file already loaded on the RADIUS server.

RADIUS Access Challenge

Juniper Networks security devices can now process access-challenge packets from an external RADIUS server when an authentication user attempts to log in via Telnet. Access challenge presents an additional condition to the login process after the approval of a username and password. After an authentication user responds to a login prompt with the correct username and password, the RADIUS server sends an access challenge to the security device, which then forwards it to the user. When the user replies, the security device sends a new access request with the user's response to the RADIUS server. If the user's response is correct, the authentication process concludes successfully. Figure 402 on page 1586 lists the steps required for an authentication user who wants to telnet to a server.

Figure 402: RADIUS Access-Challenge Sequence



1. An authentication user sends a SYN packet to initiate a TCP connection for a Telnet session to a Telnet server.
2. A security device intercepts the packet, checks its policy list, and determines that this session requires user authentication. The security device caches the SYN packet and proxies the TCP 3-way handshake with the user.

3. The security device prompts the user to log in with a username and password.
4. The authentication user enters his or her username and password and sends it to the security device. The security device then sends an access request with the login information to a RADIUS server.
5. If the information is correct, the RADIUS server sends the security device an access challenge with a reply-message attribute that prompts the user to provide a response to a challenge. (The access challenge can optionally prompt the authentication to provide a username again. The second username can be the same as the first or a different one.) The security device then sends the user another login prompt that contains the content of the reply-message attribute.
6. The authentication user enters his or her challenge response (and, optionally, a username) and sends it to the security device. The security device then sends a second access request, with the user's challenge response, to the RADIUS server.

   If the RADIUS server needs to authenticate the challenge response via another auth server—for example, if a SecurID server must authenticate a token code—the RADIUS server sends the access request to the other auth server.
7. If the RADIUS server forwarded the challenge response to another auth server and that server sends an access accept, or, if the RADIUS server itself approves the challenge response, the RADIUS server sends an access-accept message to the security device. The security device then notifies the authentication user that his or her login is successful.
8. The security device forwards the initial SYN packet to its original destination: the Telnet server.

NOTE: ScreenOS does not support access challenge with L2TP at the time of this release.

Supported RADIUS Enhancements for Auth and XAuth Users

ScreenOS supports RADIUS enhancements through the Authentication and Extended Authentication (XAuth) modules with the following attributes:

- "NS Access Service Type" on page 1588
- "Framed Pool and Framed IP Address" on page 1588
- "Account Session ID" on page 1589
- "Calling Station ID" on page 1589
- "Called Station ID" on page 1589
- "Compatibility RFC-2138" on page 1589
- "Username" on page 1590
- "Separator" on page 1590
- "Fail-Over" on page 1590

NS Access Service Type

The **NS-Access-Service-Type** attribute provides information about the service type. The security device adds this attribute to each Access-Request indicating the type of service required by the user. This attribute is enabled by default.

If the RADIUS module receives the request from a Telnet, FTP, or HTTP Authentication module, it sets the value to WEB-AUTH (2). If the RADIUS module receives the request from the XAuth module, it sets the value to VPN-IPSEC (3).

The device includes the ns-access-service-type and value in the Access-Request message. If the RADIUS server determines that the requesting user is allowed to access to the service, it sends an Access-Accept message. If the service-type is not applicable to the requesting user, the RADIUS server sends an Access-Reject message.

The RADIUS server does not include the ns-access-service-type attribute in the Access-Response messages.

Framed Pool and Framed IP Address

The RADIUS server includes the **framed-pool** attribute in the Access-Accept message. When the Framed-Pool attribute is included, the device allocates an IP address to the user from this pool. However, the device does not send the Framed-Pool attribute in Access-Request messages.

Table 112 on page 1588 shows how the device handles the framed-pool and framed-ip-address attributes. The RADIUS enhancements also includes the ability to handle address pools at the virtual system (VSYS) level.

Table 112: Supported Attributes

Supported Attributes	Resolution
Framed-Pool attribute and the Framed-IP-Address attribute are both included in the Access-Accept message.	The Framed-Pool attribute is always ignored by the RADIUS server unless the Framed-IP-Address value is 0xFFFFFFFE (255.255.255.254). Then, the device allocates an address from the Framed-Pool attribute sent by the RADIUS server.
Framed-Pool attribute and the Framed-IP-Address attribute are both absent from the Access-Accept message.	The device does not assign an IP address to the end user.
Framed-IP-Address attribute is included in the Access-Accept message and it has a value of 0xFFFFFFFE (255.255.255.254). Framed-Pool attribute is absent.	The device allocates an IP address from the default IP address pool that is configured for that VSYS.

Table 112: Supported Attributes *(continued)*

Supported Attributes	Resolution
The pool sent out in the Framed-Pool attribute is not configured, or it does not have any IP addresses.	The following error messages are generated and the negotiation is terminated: ■ Login failed: IP pool needed but not configured. ■ Login failed: No more IP address available in IP pool. In both scenarios, the client receives the following message: No more IP address available in IP pool

Account Session ID

The **acct-session-id** uniquely identifies the accounting session. Each time an XAuth user connects to the device and the device authenticates the user, the device establishes a new acct-session-id, which identifies the accounting session. The accounting session lasts between the time the device sends the RADIUS server an Accounting-Start message, and the time it sends an Accounting-Stop message. To identify the user, each RADIUS access or request message may contain the calling-station-id (described below).

The **acct-session-id length** *number* is the length of the account-session-id in bytes. The default length of this value is 11 bytes. The number setting is for accommodating some RADIUS servers, which may have problems with the default length. You can set the length of acct-session-id from 6 bytes to 10 bytes, inclusive. To restore the default setting, execute the following command:

unset auth-server *name_str* **radius attribute acct-session-id** *number*

Calling Station ID

The calling-station-id attribute identifies the originator of the call. For example, this value might consist of the phone number of the user originating the call.

Called Station ID

The called-station-id attribute identifies the destination or receiver of the call. For example, this value might consist of the phone number of the user originating the call.

Compatibility RFC-2138

The **compatibility rfc-2138** attribute makes RADIUS accounting comply with RFC 2138, as compared with RFC 2865. For operations where RFC 2865 (the most recent standard) and RFC 2138 are mutually exclusive, the command works in accordance with RFC 2138, instead of RFC 2865. In cases where the behavior is additive, the command works compatibly with both RFC 2865 and RFC 2138.

Username

The **username s**pecifies a domain name for a particular auth server, or a portion of a username from which to strip characters. If you specify a domain name for the auth server, it must be present in the username during authentication.

Separator

The device uses a separator character to identify where stripping occurs. Stripping removes all characters to the right of each instance of the specified character, plus the character itself. The device starts with the right most separator character. An example of a separator command is as follows:

set auth-server *name_str* **username separator** *string* **number** *number*

where:

- *name_str* is the name of the authentication server.
- *string* is the character separator.
- *number* is the number of character separator instances with which to perform the character stripping.

If the specified number of separator characters (*number*) exceeds the actual number of separator characters in the username, the command stops stripping at the last available separator character.

NOTE: The device performs domain-name matching before stripping.

Fail-Over

The **fail-over** attribute specifies the revert-interval value (expressed in seconds) that must pass after an authentication attempt before the device attempts authentication through backup authentication servers.

When an authentication request sent to a primary server fails, the device tries the backup servers. If authentication via a backup server is successful, and the revert interval has elapsed, the device sends subsequent authentication requests to the backup server. Otherwise, it resumes sending the requests to the primary server. The range is 0 seconds (disabled) to 86400 seconds.

The following is an example of the fail-over and revert-interval commands:

set auth-server *name_str* **fail-over revert-interval** *number*

where:

- *name_str* is the name of the authentication server.
- *number* is the length of time (expressed in seconds).

NOTE: This feature applies to RADIUS and LDAP servers only.

SecurID

Instead of a fixed password, SecurID combines two factors to create a dynamically changing password. SecurID issues a credit-card-sized device, known as an *authenticator* (Figure 403 on page 1591) a *token code* with an LCD window that displays a randomly generated string of numbers that changes every minute. The user also has a personal identification number (PIN). When logging on the user enters a username and PIN along with the current token code.

Figure 403: SecurID Token

The authenticator performs an algorithm known only by RSA to create the values that appear in the LCD window. When the user to be authenticated enters the PIN and the string of numbers on the card, the ACE server, which also performs the same algorithm, compares the values received with those in its database. If they match, the authentication is successful.

The relationship between the security device and RSA SecurID ACE server is similar to that of a security device and a RADIUS server. That is, the security device acts as a client, forwarding authentication requests to the external server for approval and relaying login information between the user and the server. SecurID differs from RADIUS in that the user's password involves a continually changing token code.

SecurID ACE Server Cluster

RSA supports a primary server and up to 10 replica servers where each server can process authentication requests. The primary and slave server can be configured with DNS or static IP. Replica servers are IP addresses that are dynamically provisioned by server upon requests from firewall. To avoid conflict in processing authentication requests, the following functions are supported:

- Name locking

  The security device sends a username lock request to the server. The server locks the username and denies access to other servers in the realm. Once the security

device sends the token code, the user is authenticated and can access the server. This avoids both unauthorized access and duplicate authentication. Username is unlocked once the authentication process is complete as either failure or success.

- Load balancing and failover

  Load balancing occurs automatically and is determined by the security device during run time. This functionality helps the agent select the best server to communicate with.

Multiple Server Cluster Instances

ScreenOS supports multiple server cluster instances if the following parameters are met:

- Master and replica1 server must be in the same realm.
- At least one primary and slave server must have static IP and must be available during system startup.
- All servers in the cluster must be running the same version of RSA authentication manager.

NOTE: If auth servers are configured with different versions of RSA authentication manager, the node-secret stored in the outgoing interface as a key must be different. The agent uses the node-secret to communicate with servers with different versions.

SecurID Auth Server Object Properties

In addition to the generic auth server properties listed in "Auth Server Object Properties" on page 1581, a SecurID server also makes use of the properties described in Table 113 on page 1592.

Table 113: SecurID Auth Server Object Properties

Property	Description
Authentication Port	The port number on the SecurID ACE server to which the security device sends authentication requests. The default port number is 5500.
Encryption Type	The algorithm used for encrypting communications between the security device and the SecurID ACE server—either SDI or DES.
Client Retries	The number of times that the SecurID client (the security device) tries to establish communication with the SecurID ACE server before aborting the attempt.
Client Timeout	The length of time in seconds that the security device waits between authentication retry attempts.

Table 113: SecurID Auth Server Object Properties *(continued)*

Property	Description
Use Duress	An option that prevents or allows use of a different PIN. When this option is enabled, and a user enters a previously determined duress PIN, the security device sends a signal to the SecurID ACE server indicating that the user is performing the login against his or her will; that is, while under duress. The SecurID ACE server permits access that one time, and then it denies any further login attempts by that user until he or she contacts the SecurID administrator. Duress mode is available only if the SecurID ACE server supports this option.

Supported User Types and Features

A SecurID ACE server supports the following types of users and authentication features:

- Auth users
- L2TP users (user authentication; L2TP user receives default L2TP settings from the security device)
- XAuth users (user authentication; no support for remote setting assignments)
- Admin users (user authentication; admin user receives default privilege assignment of read-only)

At present, a SecurID ACE server cannot assign L2TP or XAuth remote settings or ScreenOS admin privileges, although you can use a SecurID server to store L2TP, XAuth, and admin user accounts for authentication purposes. Also, ScreenOS does not provide user group support when it's used with SecurID.

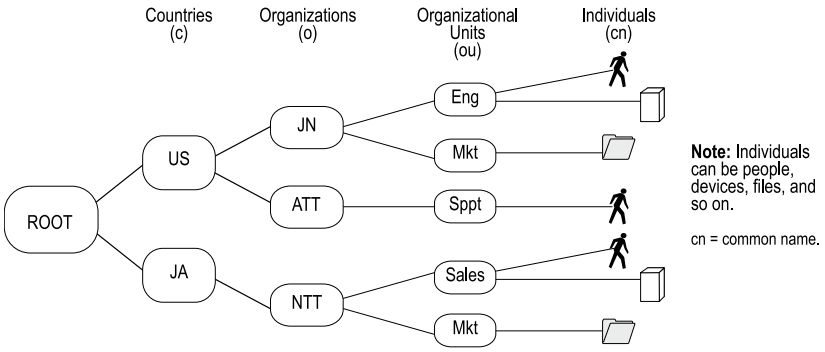
Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) is a directory server standard developed at the University of Michigan in 1996. LDAP is a protocol for organizing and accessing information in a hierarchical structure resembling a branching tree. Its purpose is twofold:

- To locate resources, such as organizations, individuals, and files on a network
- To help authenticate users attempting to connect to networks controlled by directory servers

The basic LDAP structure branches from countries to organizations to organizational units to individuals. There can also be other, intermediary levels of branching, such as "states" and " counties." Figure 404 on page 1594 shows an example of the branching organizational structure of LDAP.

Figure 404: LDAP Hierarchical Structure



NOTE: For information about LDAP, refer to RFC 1777, *Lightweight Directory Access Protocol*.

You can configure the security device to link to a Lightweight Directory Access Protocol (LDAP) server. This server uses the LDAP hierarchical syntax to identify each user uniquely.

LDAP Auth Server Object Properties

In addition to the generic auth server properties listed in "Auth Server Object Properties" on page 1581, an LDAP server also makes use of the properties described in Table 114 on page 1594.

Table 114: LDAP Auth Server Object Properties

Property	Description
LDAP Server Port	The port number on the LDAP server to which the security device sends authentication requests. The default port number is 389. Note: If you change the LDAP port number on the security device, also change it on the LDAP server.
Common Name Identifier	The identifier used by the LDAP server to identify the individual entered in a LDAP server. For example, an entry of "uid" means " user ID" and " cn" for " common name."
Distinguished Name (dn)	The path used by the LDAP server before using the common name identifier to search for a specific entry. (For example, c = us;o = juniper, where "c" stands for " country" and " o" for " organization.")

Supported User Types and Features

An LDAP server supports the following types of users and authentication features:

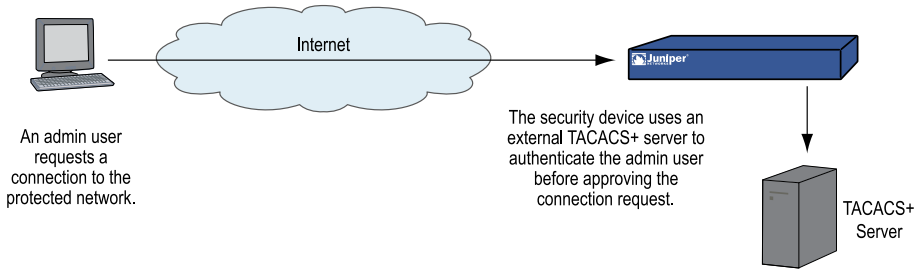
- Auth users
- L2TP users (user authentication; L2TP user receives default L2TP settings from the security device)
- XAuth users (user authentication; no support for remote setting assignments)
- Admin users (user authentication; admin user receives default privilege assignment of read-only)

At present, an LDAP server cannot assign L2TP or XAuth remote settings or ScreenOS admin privileges, although you can use an LDAP server to store L2TP, XAuth, and admin user accounts for authentication purposes. Also, ScreenOS does not provide user group support when used with LDAP.

Terminal Access Control Access Control System Plus (TACACS+)

Terminal Access Controller Access Control System Plus (TACACS+) is a security application that provides centralized validation of users.TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation.

Figure 405: Authenticating to a TACACS+ Server



The TACACS+ client (that is, the security device) authenticates users through a series of communications between the client and the server. Basically, TACACS+ asks the person logging in to enter his or her username and password. It then compares these values to those in its database, and once a user is authenticated, the client provides the user with access to the appropriate network services.

To configure the security device for TACACS+, you must specify the IP address of the TACACS+ server and define a shared secret—the same as that defined on the TACACS+ server. The shared secret is a password the TACACS+ server uses to generate a key to encrypt traffic between the security and TACACS+ devices.

TACACS+ provides for separate and modular authentication, authorization, and accounting facilities. The TACACS+ support in ScreenOS allows the TACACS+ server to provide authentication and authorization independently as follows:

- Authentication and authorization

  TACACS+ separates authentication and authorization functions. Remote authentication is supported for admin users only. Authenticated administrators

are assigned a privilege level and vsys. ScreenOS supports user-level authorization only.

- Configure up to two TACACS+ servers

  If the primary TACACS+ server is not reachable, then the backup TACACS+ server is queried. For more information, see "Prioritizing Admin Authentication" on page 1596.

TACACS+Server Object Properties

In addition to the generic auth server properties listed in "Auth Server Object Properties" on page 1581, a TACACS+ server also makes use of the properties described in Table 115 on page 1596.

Table 115: TACACS+Server Object Properties

Property	Description
Shared Secret	The secret (password) shared between the security device and the TACACS+ server. The devices use this secret to encrypt the user's password that it sends to the TACACS+ server.
TACACS+ Port	The port number on the TACACS+ server to which the security device sends authentication requests. The default port number is 49.
TACACS+ Retry Timeout	The interval (in seconds) that the security device waits before sending another authentication request to the TACACS+ server if the previous request does not elicit a response. The default is three seconds.
Client Retries	The number of times that the TACACS+ client (that is, the security device) tries to establish communication with the TACACS+ server before aborting the attempt.
Client Timeout	The length of time in seconds that the security device waits between authentication retry attempts.
Encryption	TACACS+ encrypts the entire payload of the client server exchange.

Prioritizing Admin Authentication

ScreenOS lets you prioritize the authentication process with regards to local and remote authentication services. The admin defines the sequence in which the authentication service is queried and the action to take if the initial attempt fails.

- Assigning priorities to authentication services

  Set higher priority to authenticate to the remote auth server over the local database. Root-privileged admins can define the sequence of the authentication service (local and remote) in which admin authentication service is attempted. The root admin sets one of the authentication services as primary. The other automatically becomes a secondary service.

- Defining fallback behavior

If the primary authentication service fails, you can configure the device to authenticate to the secondary service (default) or bypass it. The above action is defined differently for root-privileged and non-root privileged admins.

- Accepting externally authenticated admins

  Configure the security device to accept or not to accept root-privileged admins authenticated by a remote auth server.

- Default authentication

  If remote authentication fails and local authentication is disabled, then device resorts to allowing the root-privileged admins to be authenticated locally. This fallback procedure is restricted to the serial console.

Defining Auth Server Objects

Before you can refer to external authentication (auth) servers in policies, IKE gateways, and L2TP tunnels, you must first define the auth server objects. The following examples illustrate how to define auth server objects for a RADIUS server, a SecurID server, an LDAP server, and a TACACS+ server.

Example: RADIUS Auth Server

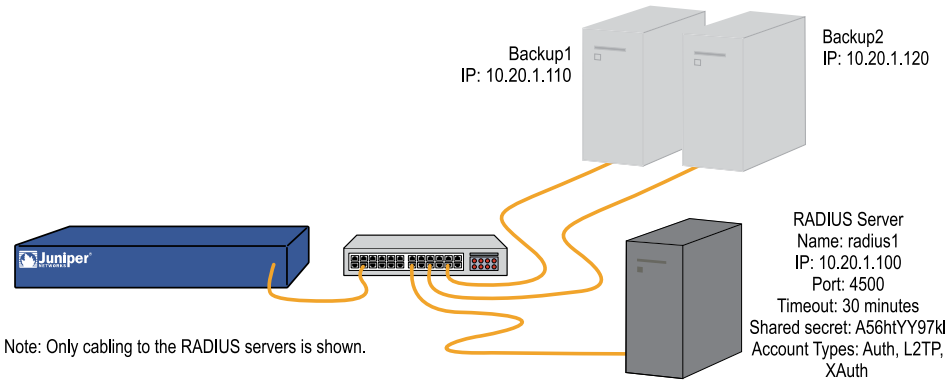
In the following example, you define an auth server object for a RADIUS server. You specify its user account types as auth, L2TP, and XAuth. You name the RADIUS server "radius1" and accept the ID number that the security device automatically assigns it. You enter its IP address, which is 10.20.1.100, and change its port number from the default (1645) to 4500. You define its shared secret as " A56htYY97kl" . You also assign its two backup servers the IP addresses 10.20.1.110 and 10.20.1.120.

You change the authentication idle timeout value from the default (10 minutes) to 30 minutes and the RADIUS retry timeout from 3 seconds to 4 seconds. But because, with this setting, the session could theoretically remain open indefinitely (as long as one keystroke is sent every 30 minutes) you limit total session time by setting forced-timeout to 60 minutes. With this setting, after one hour the auth table entry for the user is removed, as are all associated sessions for the auth table entry, and the user needs to reauthenticate.

You also load the RADIUS dictionary file on the RADIUS server so that it can support queries for the following vendor-specific attributes (VSAs): user groups, admin privileges, and remote L2TP and XAuth settings.

In Figure 406 on page 1598, The security device sends auth, L2TP, and XAuth authentication requests to the primary RADIUS server, "radius1" , at 10.20.1.100. If the security device loses network connectivity with the primary RADIUS server, it redirects authentication requests to backup1 at 10.20.1.110. If the security device cannot reach backup1, it redirects authentication requests to backup2 at 10.20.1.120.

Figure 406: RADIUS Backup Example



WebUI

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: radius1
IP/Domain Name: 10.20.1.100
Backup1: 10.20.1.110
Backup2: 10.20.1.120
Timeout: 30
Forced Timeout: 60
Account Type: Auth, L2TP, XAuth
RADIUS: (select)
    RADIUS Port: 4500
    Retry Timeout: 4 (seconds)
    Shared Secret: A56htYY97kl

Load the RADIUS dictionary file on the RADIUS server.

NOTE: For more information, see “RADIUS Dictionary File” on page 1585. For instructions on how to load the dictionary file onto a RADIUS server, refer to the documentation for your specific server.

CLI

```
set auth-server radius1 type radius
set auth-server radius1 account-type auth l2tp xauth
set auth-server radius1 server-name 10.20.1.100
set auth-server radius1 backup1 10.20.1.110
set auth-server radius1 backup2 10.20.1.120
set auth-server radius1 forced-timeout 60
set auth-server radius1 timeout 30
set auth-server radius1 radius port 4500
set auth-server radius1 radius timeout 4
set auth-server radius1 radius secret A56htYY97kl
save
```

NOTE: The order in which you enter the account types is important. For example, if you first enter **set auth-server radius1 account-type l2tp**, then your only subsequent choice is **xauth**; you cannot enter **auth** after **l2tp**. The correct order is easily remembered because it is alphabetical.

Changing the port number helps deter potential attacks targeted at the default RADIUS port number (1645).

Load the RADIUS dictionary file on the RADIUS server.

NOTE: For more information, see "RADIUS Dictionary File" on page 1585. For instructions on how to load the dictionary file onto a RADIUS server, refer to the documentation for your specific server.

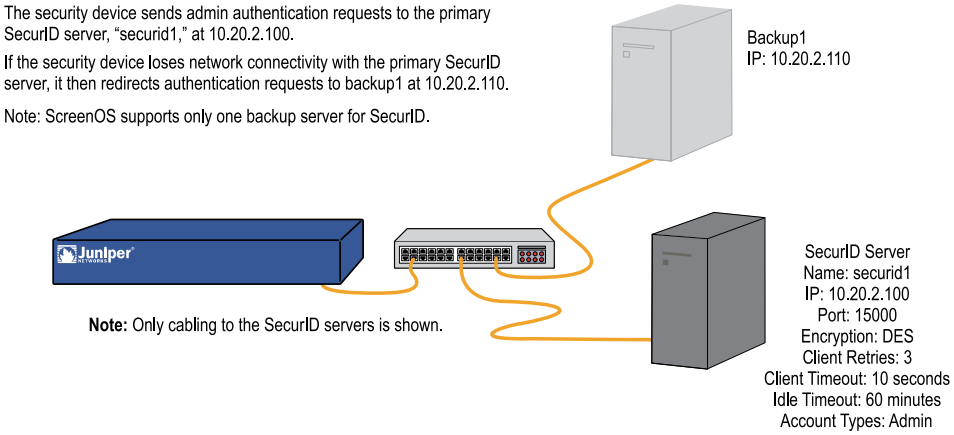
Example: SecurID Auth Server

In the following example, you configure an auth server object for a SecurID ACE server. You specify its user account type as admin. You name the server "securid1" and accept the ID number that the security device automatically assigns it. You enter the IP address of the primary server, which is 10.20.2.100, and the IP address of a backup server: 10.20.2.110. You change its port number from the default (5500) to 15000. The security device and the SecurID ACE server protect the authentication information using DES encryption. There are three allowable retries and a client timeout value of 10 seconds. You change the idle timeout value from the default (10 minutes) to 60 minutes. The **Use Duress** setting is disabled. See Figure 407 on page 1600.

NOTE: The client timeout value is the length of time in seconds that the SecurID client (that is, the security device) waits between authentication retry attempts.

The idle timeout value is the length of idle time in minutes that can elapse before the security device automatically terminates an inactive admin session. (For information comparing the timeout value as applied to admin users and other user types, see "Auth Server Object Properties" on page 1581.)

Figure 407: SecurID Backup Example



WebUI

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: securid1
IP/Domain Name: 10.20.2.100
Backup1: 10.20.2.110
Timeout: 60
Account Type: Admin
SecurID: (select)
    Client Retries: 3
    Client Timeout: 10 seconds
    Authentication Port: 15000
    Encryption Type: DES
    User Duress: No

CLI

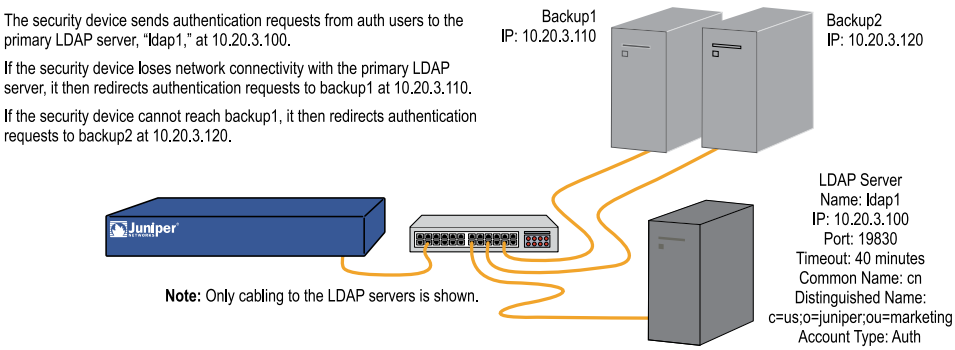
```
set auth-server securid1 type securid
set auth-server securid1 server-name 10.20.2.100
set auth-server securid1 backup1 10.20.2.110
set auth-server securid1 timeout 60
set auth-server securid1 account-type admin
set auth-server securid1 securid retries 3
set auth-server securid1 securid timeout 10
set auth-server securid1 securid auth-port 15000
set auth-server securid1 securid encr 1
set auth-server securid1 securid duress 0
save
```

Example: LDAP Auth Server

In the following example, you configure an auth server object for an LDAP server. You specify its user account type as **auth**. You name the LDAP server "ldap1" and accept the ID number that the security device automatically assigns it. You enter its IP address, which is 10.20.3.100, and change its port number from the default (389)

to 19830. You change the idle timeout value from the default (10 minutes) to 40 minutes. You also assign its two backup servers the IP addresses 10.20.3.110 and 10.20.3.120. The LDAP common name identifier is **cn**, and the Distinguished Name is **c = us;o = juniper;ou = marketing**. See Figure 408 on page 1601.

Figure 408: LDAP Backup Example



WebUI

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: ldap1
IP/Domain Name: 10.20.3.100
Backup1: 10.20.3.110
Backup2: 10.20.3.120
Timeout: 40
Account Type: Auth
LDAP: (select)
    LDAP Port: 4500
    Common Name Identifier: cn
    Distinguished Name (dn): c=us;o=juniper;ou=marketing

CLI

```
set auth-server ldap1 type ldap
set auth-server ldap1 account-type auth
set auth-server ldap1 server-name 10.20.3.100
set auth-server ldap1 backup1 10.20.3.110
set auth-server ldap1 backup2 10.20.3.120
set auth-server ldap1 timeout 40
set auth-server ldap1 ldap port 15000
set auth-server ldap1 ldap cn cn
set auth-server ldap1 ldap dn c=us;o=juniper;ou=marketing
save
```

Example: TACACS+ Auth Server

In the following example, you define an auth server object for a TACACS+ server. You name the TACACS+ server "tacacs1" and accept the ID number that the security device automatically assigns it. You enter its IP address, which is 1.1.1.1. You define

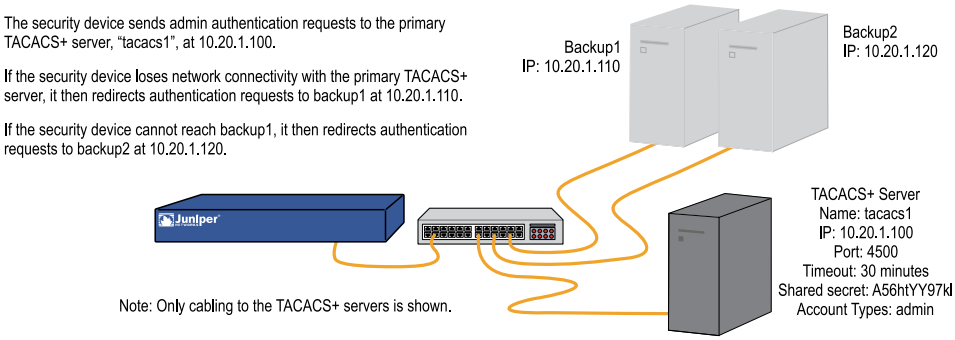
its shared secret as " A56htYY97kl" . You also assign its two backup servers the IP addresses 2.2.2.2 and 3.3.3.3.

You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your device are available.

You change the authentication idle timeout value from the default (10 minutes) to 30 minutes and the TACACS+ retry timeout from 3 seconds to 4 seconds. But because, with this setting, the session could theoretically remain open indefinitely (as long as one keystroke is sent every 30 minutes) you limit total session time by setting forced-timeout to 60 minutes. With this setting, after one hour the auth table entry for the user is removed, as are all associated sessions for the auth table entry, and the user needs to reauthenticate.

In Figure 409 on page 1602, The security device sends admin authentication requests to the primary TACACS+ server, "tacacs1" , at 1.1.1.1. If the security device loses network connectivity with the primary TACACS+ server, it redirects authentication requests to backup1 at 2.2.2.2. If the security device cannot reach backup1, it redirects authentication requests to backup2 at 3.3.3.3.

Figure 409: TACACS+ Backup Example



WebUI

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: tacacs1
IP/Domain Name: 1.1.1.1
Backup1: 2.2.2.2
Backup2: 3.3.3.3
Timeout: 10
Forced Timeout: 60
Account Type: admin
TACACS: (select)
    TACACS Port: 49
    Retry Timeout: 4 (seconds)
    Shared Secret: A56htYY97kl

CLI

```
set auth-server tacacs1 type tacacs
```

```
set auth-server tacacs1 account-type auth l2tp xauth
set auth-server tacacs1 server-name 1.1.1.1
set auth-server tacacs1 backup1 2.2.2.2
set auth-server tacacs1 backup2 3.3.3.3
set auth-server tacacs1 forced-timeout 60
set auth-server tacacs1 tacacs port 49
set auth-server tacacs1 tacacs secret A56htYY97kl
set admin auth server tacacs1
save
```

NOTE: Changing the port number helps deter potential attacks targeted at the default TACACS port number (49).

Defining Default Auth Servers

By default, the local database is the default auth server for all user types. You can specify external auth servers to be the default auth servers for one or more of the following user types:

- Admin
- Auth
- L2TP
- XAuth
- 802.1x

Then, if you want to use the default auth server for a specific user type when configuring authentication in policies, L2TP tunnels, or IKE gateways, you do not have to specify an auth server in every configuration. The security device refers to the appropriate auth servers that you previously appointed to be the defaults.

Example: Changing Default Auth Servers

In this example, you use the RADIUS, SecurID, and LDAP auth server objects that you created in the previous examples:

- radius1 (“Example: RADIUS Auth Server” on page 1597)
- securid1 (“Example: SecurID Auth Server” on page 1599)
- ldap1 (“Example: LDAP Auth Server” on page 1600)

You then assign the local database, radius1, securid1, and ldap1 as the default servers for the following user types:

- Local: Default auth server for XAuth users
- radius1: Default auth server for L2TP users
- securid1: Default auth server for admin users
- ldap1: Default auth server for auth users

NOTE: By default, the local database is the default auth server for all user types. Therefore, unless you have previously assigned an external auth server as the default server for XAuth users, you do not need to configure it as such.

WebUI

VPNs > AutoKey Advanced > XAuth Settings: Select **Local** from the Default Authentication Server drop-down list, then click **Apply**.

VPNs > L2TP > Default Settings: Select **radius1** from the Default Authentication Server drop-down list, then click **Apply**.

Configuration > Admin > Administrators: Select **Local/securid1** from the Admin Auth Server drop-down list, then click **Apply**.

Configuration > Auth > Firewall: Select **ldap1** from the Default Auth Server drop-down list, then click **Apply**.

CLI

```
set xauth default auth server Local
set l2tp default auth server radius1
set admin auth server securid1
set auth default auth server ldap1
save
```

NOTE: By default, the local database is the default auth server for all user types. Therefore, unless you have previously assigned an external auth server as the default server for XAuth users, you do not need to configure it as such.

Configuring a Separate External Accounting Server

By default, authentication and accounting are performed in the RADIUS auth server. You can configure separate RADIUS servers for accounting and authentication for the following user types:

- XAuth
- L2TP

XAUTH and L2TP users can disable the default accounting and configure a different RADIUS server for accounting.

NOTE: This feature is available only for XAuth and L2TP users in RADIUS auth servers.

Example: Configuring a Separate Accounting Server

In this example, you disable the default RADIUS accounting for L2TP users and configure separate RADIUS authentication (radius1) and accounting (radius2) servers.

1. **Disabling Accounting**

```
set l2tp default accounting off
```

2. **Configuring Authentication and Accounting Servers**

```
set auth-server radius1 type radius
set auth-server radius1 server-name 10.1.1.1
set auth-server radius1 account-type l2tp
set auth-server radius1 radius secret mysecret
set auth-server radius2 type radius
set auth-server radius2 server-name 10.1.1.2
set auth-server radius2 account-type l2tp
set auth-server radius2 radius secret mysecret
```

3. **Separating Authentication and Accounting Servers**

```
set l2tp default auth server radius1
set l2tp default accounting server radius2
```

NOTE: As of this release, ScreenOS supports separate authentication and accounting for RADIUS servers.

Chapter 48

Infranet Authentication

This chapter details how a Juniper Networks security device configured as an Infranet Enforcer is deployed with the Unified Access Control (UAC) solution.

Within a UAC deployment, the Juniper Networks Infranet Controller serves as the policy decision point where access is granted or denied and access privileges are controlled. UAC coordinates network authentication and policy rules for Odyssey Access Client (OAC) endpoints to provide granular network access control.

The Infranet Enforcer, deployed in front of servers and resources that you want to protect, serves as the policy enforcement point in the network.

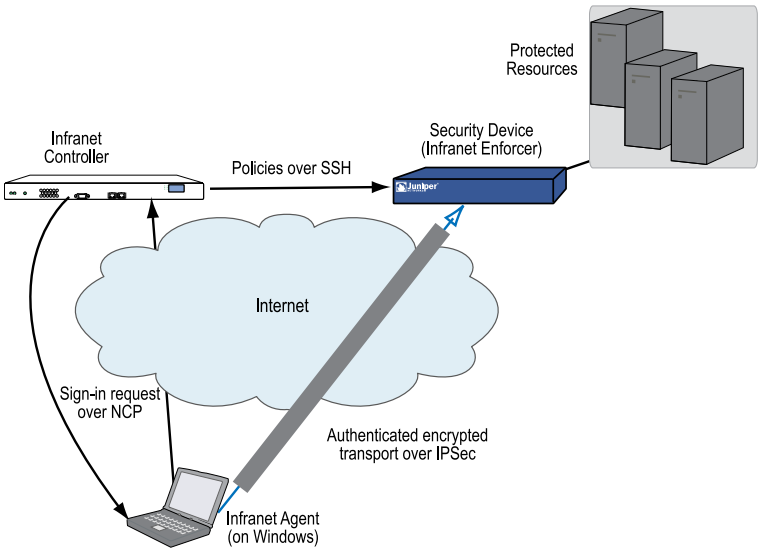
For more information about configuring and deploying UAC, see the Unified Access Control Administration Guide.

This chapter includes the following sections:

Unified Access Control Solution

The Infranet Enforcer and the Infranet Controller work together to provide endpoint security and firewall services to ensure that only authorized end users can access protected resources. See Figure 410 on page 1608.

Figure 410: Deploying the Infranet Enforcer with Unified Access Control



Access control on the Infranet Controller is policy-based. When specifying the resources that endpoints can access, you configure three types of policies: infranet-auth, resource access, and Host Checker. The Infranet Controller permits access to endpoints based on successful authentication, level of trust, and the status or health of the endpoint.

Infranet-auth policies enforce traffic permissions between source and destination zones on the Infranet Enforcer. You create basic infranet-auth policies on the Infranet Controller, and then you push the policies to the Infranet Enforcer to configure more complex policy options.

There are two types of infranet-auth policies:

- A source IP policy contains source and destination IP addresses to permit the Enforcer to route cleartext traffic between the source and destination zones.
- An IPsec policy contains source and destination information and other parameters to permit the Infranet Enforcer to set up a virtual private network (VPN) for encrypted traffic between the source and destination zones.

Resource access policies that you create on the Infranet Controller specify which users are allowed or denied access to a set of protected resources based on the level of trust that you assign to individual user roles.

You create authentication realms and assign roles to individual users. You configure role-mapping rules to associate authentication realms with roles on the Infranet Controller.

You use resource access policies to specify which roles should be allowed or denied access to individual resources based on the level of protection the resources require.

The Infranet Enforcer performs the action specified in the resource access policies that you create on the Infranet Controller, and you can configure the resource access policy to direct the Infranet Enforcer to notify users when they are denied access to a particular resource.

You can create any number of roles on the Infranet Controller to specify granular access restrictions for endpoints. For example, you can create an *employee* role with full access and a *contractor* role with access limitations.

You can also use resource access policies to apply additional firewall actions such as antivirus, antispam, Web filtering, and Intrusion Detection and Protection (IDP) to endpoints that access resources. You configure the resource access policy to include these actions on the Infranet Controller and then set up corresponding security policies on the Infranet Enforcer.

In addition to the policies that permit the Infranet Enforcer and the Infranet Controller to work together, you can use Host Checker policies on the Infranet Controller to ensure that the network is not compromised by endpoints that do not meet minimum security requirements.

Host Checker is a highly configurable tool that allows you to require endpoints to meet specific security requirements to access protected resources. For example, you can specify rules that require endpoints to run a current OS patch, to have an updated antivirus version, or to have a specific application running. If the endpoint does not meet the security requirements, Host Checker can display a page that instructs the user how to bring the endpoint into compliance.

How the Security Device Works with the Infranet Controller

This section explains how the Juniper Networks security device (in a UAC configuration, known as the Infranet Enforcer) and the Infranet Controller work together to establish communications and enforce security policies.

To allow the Infranet Enforcer and the Infranet Controller to communicate, you must import security certificates into each device. You create a certificate signing request (CSR) for a server certificate and then sign the CSR using your Certificate Authority (CA) or by using OpenSSL.

You import the signed server certificate into the Infranet Controller, and you import the certificate of the CA that signed the Infranet Controller's server certificate into the Infranet Enforcer.

The Infranet Enforcer initially connects with the Infranet Controller over an SSH connection that uses the NetScreen Address Change Notification (NACN) protocol. You specify the following items on the Infranet Controller in an Infranet Enforcer connection policy to initiate communication:

- NACN password
- Administrator name and password for signing into the Infranet Enforcer using SSH
- Serial number(s) of the Infranet Enforcer(s)

You can deploy multiple Infranet Enforcers to work with a single Infranet Controller. You can create auth table policies on the Infranet Controller to specify the resources that can be reached through each Infranet Enforcer.

With auth table policies you can automatically create auth table entries on specific Infranet Enforcers when a user authenticates. Auth table entries contain information about the user, including the user's role(s) and source IP address.

Auth table entries permit the Infranet Controller to provide information about each user that allows the Infranet Enforcer to enforce the policies you have created. Alternately, you can use dynamic auth table allocation to allow auth table entries to be provisioned only when a user requests access to a specific resource.

You can configure the Infranet Enforcer to work with up to three Infranet Controllers in a cluster environment, but each Enforcer supports only one Infranet Controller. See "Infranet Controller Clustering" on page 1612.

1. At startup, the Infranet Enforcer contacts the Infranet Controller over an SSL connection using the NACN protocol.
2. After the Infranet Enforcer successfully establishes an NACN connection with the Infranet Controller, the Infranet Controller opens an SSH connection with the Enforcer to push policy information to the Enforcer. All communication between the Infranet Controller and the Enforcer is over the SSH connection.
3. When the Infranet Controller authenticates a user and verifies that the user's computer complies with endpoint security policies, the Infranet Controller can share user authentication information with the Enforcer. This information includes auth table entries for authenticated users.

   You can configure the Infranet Controller to dynamically provision auth table entries only when a user attempts to access a resource protected by the Enforcer. See "Dynamic Auth Table Allocation" on page 1610.
4. When the Enforcer detects traffic from a user that matches an infranet-auth policy, it uses the user's auth table entry along with the resource access policies that apply to the protected resource to determine whether to allow the user to access the protected resource.

   When IDP is enabled, administrators can configure the security device to inspect traffic using either user roles or source IP addresses. When you select user-role-based IDP inspection, the security device will start checking the user-role-based policies first; only if a match is not found will it then search for IP-based rules.
5. The Infranet Controller sends commands to the Infranet Enforcer to remove policies or auth table entries and deny access to resources as necessary. This can occur when the user's computer becomes noncompliant with endpoint security policies or loses its connection with the Infranet Controller.

Dynamic Auth Table Allocation

Depending on the Infranet Enforcer you are using, there are limits to the number of auth tables the security device can store. Dynamic auth table allocation allows you to limit the number of auth tables on the Infranet Enforcer.

Normally, the Infranet Controller sends auth table information to each connected Infranet Enforcer when a user authenticates. Dynamic auth table allocation enables the Infranet Controller to provision auth table entries to a specific Infranet Enforcer only after access to a specific resource is requested.

With dynamic auth table allocation, when an endpoint attempts to access a resource, the Infranet Enforcer sends a message to the Infranet Controller with the source IP address and the destination IP address the endpoint was trying to access, the policy that caused the Enforcer to drop the packet, and an ID that uniquely identifies the Infranet Enforcer to the Infranet Controller.

When the Infranet Controller receives notification that a packet has been dropped, it searches through its resource access policies to determine if the user can access the resource. If the request matches a policy, the Infranet Controller sends an auth table entry to the Infranet Enforcer that sent the notification.

Supporting a Unified Access Control Solution in a Virtual System Configuration

To support the Infranet Controller with a virtual system (vsys) configuration, you must add the Infranet Controller in the root-vsys and share the connection to the vsys in the security device. This shared connection enables the Infranet Controller to support vsys-based authentications.

You can configure infranet-auth policies and resource access policies specific to each vsys on the Infranet Controller. The Infranet Controller pushes the configuration details to each vsys that it is configured to support.

NOTE: You use the **exec bulkcli vsys vsys_name bulkcli_string** command to send vsys-specific configuration details for RADIUS server, VPN, user, IKE, and IPsec policies as multi-line commands from the Infranet Controller to the Infranet Enforcer. For more information about this command, see the ScreenOS CLI Reference Guide: IPv4 Command Descriptions.

How the Infranet Controller Works with Multiple Vsys

This section describes how different virtual systems communicate with the Infranet Controller to gain access to protected resources:

- The Infranet Enforcer sends a dynamic auth table allocation message to the Infranet Controller when a user attempts to access resources protected by the security device. These messages include the source address, source interface, destination address, policy ID, and vsys ID of the endpoint attempting to reach the protected resource.
- Upon receiving the message, the Infranet Controller relays the auth table information of the corresponding vsys to the Enforcer.
- Based on the auth table information and infranet-auth policy, the Infranet Controller performs the authentication check to determine whether to allow the user access to the protected resource.

For users who want to access resources such as a Web server, and if the policy demands Infranet-auth redirect-only, the security device redirects the HTTP request to the Infranet Controller, which in this case is permitted to do the authentication for the user.

The security device redirects traffic that is destined for non-standard ports also. To enable this, you configure the infranet policy as shown in the following example:

```
set service captive protocol tcp src-port 0-65535 dst-port 10000-10000
set policy id 1 from untrust to trust any any captive permit infranet-auth
redirect-unauthenticated
set policy id 1 application http

set policy id 1 from z1 to z2 any any HTTP permit infranet-auth redirect-unauthenticated
```

NOTE: For the HTTP redirect to succeed, the user must send a message to the Infranet Controller indicating the destination address, destination URL, policy ID, enforcer ID, and vsys name of the HTTP traffic.

Infranet Controller Clustering

You can configure up to three Infranet Controllers in a cluster to ensure continuous network protection if one of the Infranet Controllers fails. The Infranet Enforcer communicates with only one Infranet Controller at a time; the other Infranet Controllers are used for backup. If the Enforcer cannot connect to the first Infranet Controller, it tries the next one in its configuration list until a connection can be made. Infranet Controllers configured with the Infranet Enforcer should all be members of the same Infranet Controller cluster.

An Infranet Controller cluster can communicate with a cluster of Infranet Enforcers that synchronize using NetScreen Redundancy Protocol (NSRP) and keepalive messages between two types of clusters. The runtime objects (RTOs) such as auth table entries and infranet auth policies that the Infranet Controller sends to the Infranet Enforcer are synchronized with all nodes in an Infranet Enforcer cluster.

A newly joined node establishes connection with an Infranet Controller only after it has synchronized the RTOs with its peers in the cluster to ensure that all peers in an Infranet Enforcer cluster have the same state at the end of the synchronization. Any infranet auth table or auth policy updates from the Infranet Controller are then synchronized across all security devices in the cluster. For more information about RTO Synchronization, see *"High Availability" on page 1763*.

Infranet Enforcers in the cluster maintain the infranet auth table entries and infranet auth policies for two minutes after the keepalive timeout to allow for establishing a new connection to the Infranet Controller (or another node in the Infranet Controller cluster if the previously connected Infranet Controller has failed).

Viewing the Configuration of an Infranet Controller Instance

You can view the configuration of an Infranet Controller instance through the WebUI and the CLI, which includes the following information:

- Name of the Infranet Controller instance
- IP address or domain name of the Infranet Controller
- Port number (should always be 11122)
- Timeout (60 seconds by default)
- Redirect URL for Captive Portal

The WebUI also allows you to view the NACN password and CA parameters.

WebUI

Select Configuration > Infranet Auth > Controllers from the left navigation bar.

Select Configuration > Infranet Auth > General Settings from the left navigation bar.

CLI

```
get infranet controller name controller1
```

Chapter 49

Authentication Users

An authentication user (or *auth user*) is a network user who must provide a username and password for authentication when initiating a connection across the firewall. You can store an auth user account on the local database or on an external RADIUS, SecurID, or LDAP server.

You can put several auth user accounts together to form an auth user group, which you can store on the local database or on a RADIUS server. A single auth user account can be in up to four user groups on the local database or on a RADIUS server. If you create an external user group on a RADIUS server, you must also create an identical—but unpopulated—user group on the security device. For example, if you define an auth user group named "au_grp1" on a RADIUS server named " rs1" and add 10 members to the group, then on the security device you need to define an auth user group also named " au_grp1," identify it as an external user group, but add no members to it. When you reference the external auth user group " au_grp1" and auth server " rs1" in a policy, the security device can properly query the specified RADIUS server when traffic matching the policy provokes an authentication check. This chapter contains the following sections:

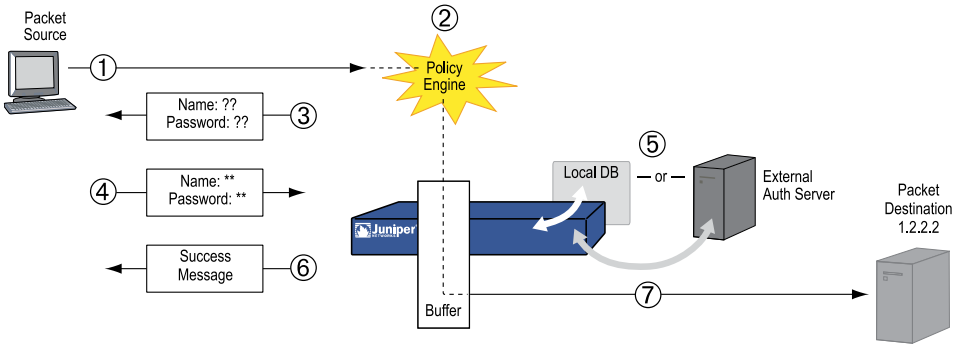
Referencing Auth Users in Policies

After you define an auth user, you can then create a policy that requires the user to authenticate himself or herself through one of two authentication schemes. The first scheme authenticates users when FTP, HTTP, or Telnet traffic matching a policy requiring authentication reaches the security device. In the second scheme, users authenticate themselves before sending traffic (of any kind—not just FTP, HTTP, or Telnet) to which a policy requiring user authentication applies.

Run-Time Authentication

When a user attempts to initiate an HTTP, an FTP, or a Telnet connection request to which a policy requiring authentication applies, the security device intercepts the request and prompts the user to enter a name and password (see "User Authentication" on page 208). Before granting permission, the security device validates the username and password by checking them against those stored in the local database or on an external auth server. See Figure 411 on page 1616.

Figure 411: Policy Lookup for a User

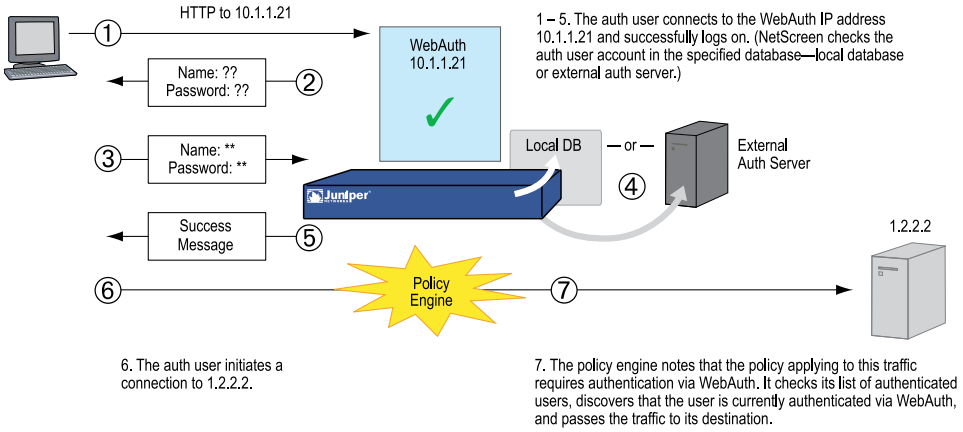


1. An auth user sends an FTP, an HTTP, or a Telnet packet to 1.2.2.2.
2. The security device intercepts the packet, notes that its policy requires authentication from either the local database or an external auth server, and buffers the packet.
3. The security device prompts the user for login information via FTP, HTTP, or Telnet.
4. The user replies with a username and password.
5. The security device either checks for an auth user account on its local database or it sends the login information to the external auth server as specified in the policy.
6. Finding a valid match (or receiving notice of such a match from the external auth server), the security device informs the user that the login has been successful.
7. The security device forwards the packet from its buffer to its destination of 1.2.2.2.

Pre-Policy Check Authentication (WebAuth)

Before sending traffic to an intended destination, an auth user initiates an HTTP session to the IP address hosting the WebAuth feature on the security device and authenticates himself or herself. After the security device authenticates the user, he or she can then send traffic to the destination as permitted by a policy requiring authentication via WebAuth. See Figure 412 on page 1617.

Figure 412: WebAuth Example



Some details about WebAuth:

- You can leave the default WebAuth auth server as the local database or you can choose an external auth server for the role. The main requirement for a WebAuth auth server is that the auth server must have auth user account-types.
- The WebAuth address must be in the same subnet as the interface that you want to use to host it. For example, if you want auth users to connect to WebAuth via ethernet3, which has IP address 1.1.1.1/24, then you can assign WebAuth an IP address in the 1.1.1.0/24 subnet.
- You can put a WebAuth address in the same subnet as the IP address of any physical interface, subinterface, or virtual security interface (VSI). (For information about different types of interfaces, see “Interfaces” on page 51.)
- If you want to use WebAuth while in transparent mode, you can put a WebAuth address in the same subnet as the VLAN1 IP address.
- You can put WebAuth addresses on multiple interfaces.
- If you have multiple interfaces bound to the same security zone, you can put a WebAuth address in a subnet on one interface, and traffic from the same zone but using a different interface can still reach it.
- You should be aware that after a security device authenticates a user at a particular source IP address, it subsequently permits traffic—as specified in the policy requiring authentication via WebAuth—from any other user at that same address. This might be the case if the users originate traffic from behind a NAT device that changes all original source addresses to a single translated address.
- You can direct the device to accept only SSL (HTTPS) connections for WebAuth sessions.

The security device redirects HTTP traffic to the WebAuth IP address (the WebAuth server) configured for the incoming interface. The security device also redirects traffic that is destined for non-standard ports.To set the security device to automatically redirect HTTP traffic to the WebAuth server:

WebUI

Policy > Policies > Edit > Advanced: Select the following options, then click OK:

WebAuth (Local) (select)
Redirect unauthenticated traffic (select)

CLI

```
set policy from zone1 to zone2 any any any permit webauth
redirect-unauthenticated
```

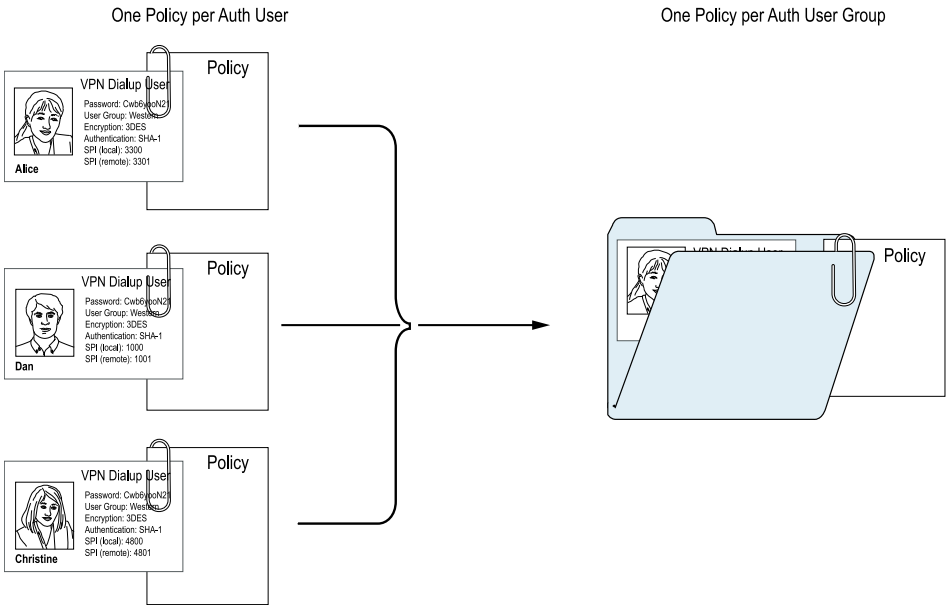
Referencing Auth User Groups in Policies

To manage a number of auth users, you can create auth user groups and store them either on the local security device or on an external RADIUS server.

NOTE: If you store users in groups on a RADIUS server, you must create unpopulated external user groups on the security device with names that correspond with those of the user groups you create on the RADIUS server.

Rather than manage each user individually, you can gather users into a group, so that any changes made to the group propagate to each group member. An auth user can be a member of up to four user groups on the local database or on a RADIUS server. An auth user who belongs to more than one group is required to supply a username and password only once, before being granted access to the resources defined for each group in which the user is a member. See Figure 413 on page 1619.

Figure 413: Auth User Groups



Example: Run-Time Authentication (Local User)

In this example, you define a local auth user named louis with password iDa84rNk and an address named host1 in the Trust zone address book. You then configure two outgoing policies: one that denies all outbound traffic, and another from host1 requiring louis to authenticate himself. (Louis must initiate all outbound traffic from host1.) The security device denies outbound access from any other address, as well as unauthenticated traffic from host1.

WebUI

1. **Local Auth User and Address**

   Objects > Users > Local > New: Enter the following, then click **OK**:

   User Name: louis
   Status: Enable
   Authentication User: (select)
   User Password: iDa84rNk
   Confirm Password: iDa84rNk

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: host1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.4/32
   Zone: Trust

2. **Policies**

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: ANY
Action: Deny

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), host1
Destination Address:
    Address Book Entry: (select), Any
Service: ANY
Action: Permit
Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
Auth Server: (select)
    Use: Local
User: (select), Local Auth User - louis

CLI

1. **Local User and Address**

```
set user louis password iDa84rNk
set address trust host1 10.1.1.4/32
```

NOTE: By default, a user to whom you assign a password is classified as an auth user.

2. **Policies**

```
set policy from trust to untrust any any any deny
set policy top from trust to untrust host1 any any permit auth user louis
save
```

Example: Run-Time Authentication (Local User Group)

In this example, you define a local user group named auth_grp1. You add previously created auth users louis and lara to the group. Then you configure a policy referencing auth_grp1. The policy provides FTP-GET and FTP-PUT privileges for auth_grp1, with

address name “auth_grp1” (IP address 10.1.8.0/24) in the Trust zone to access an FTP server named “ ftp1” (IP address 1.2.2.3/32) in the DMZ zone.

NOTE: When you create a user group in the local database, its user type remains undefined until you add a user to it. At that point, the user group takes the type or types of users that you add to it. You can create a multiple-type user group by adding auth, IKE, L2TP, and XAuth user types. You cannot combine Admin users with any other user type.

WebUI

1. **Local User Group and Members**

   Objects > Users > Local Groups > New: Enter **auth_grp1** in the Group Name field, do the following, then click **OK**:

   Select **louis** and use the < < button to move him from the Available Members column to the Group Members column.

   Select **lara** and use the < < button to move her from the Available Members column to the Group Members column.

2. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: auth_grp1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.8.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: ftp1
   IP Address/Domain Name:
   IP/Netmask: (select), 1.2.2.3/32
   Zone: DMZ

3. **Policy**

   Policy > Policies > (From: Trust; To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), auth_grp1
   Destination Address:
   Address Book Entry: (select), ftp1
   Service: FTP
   Action: Permit
   Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

```
Authentication: (select)
Auth Server: (select)
    Use: Local
User Group: (select), Local Auth Group - auth_grp1
```

CLI

1. **Local User Group and Members**

```
set user-group auth_grp1 location local
set user-group auth_grp1 user louis
set user-group auth_grp1 user lara
```

2. **Address**

```
set address trust auth_grp1 10.1.8.0/24
set address dmz ftp1 1.2.2.3/32
```

3. **Policy**

```
set policy top from trust to dmz auth_grp1 ftp1 ftp permit auth user-group
auth_grp1
save
```

Example: Run-Time Authentication (External User)

In this example, you define an external LDAP auth server named "x_srv1" with the following attributes:

- Account type: auth
- IP address: 10.1.1.100
- Backup1 IP address: 10.1.1.110
- Backup2 IP address: 10.1.1.120
- Authentication timeout: 60 minutes
- LDAP port number: 14500
- Common name identifier: cn
- Distinguished name: c = us;o = netscreen

You load the auth user "euclid" with password eTcS114u on the external auth server. You then configure an outgoing policy that requires authentication on auth server x_srv1 for external user euclid.

WebUI

1. **Auth Server**

Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

Name: x_srv1
IP/Domain Name: 10.1.1.100
Backup1: 10.1.1.110
Backup2: 10.1.1.120
Timeout: 60
Account Type: Auth
LDAP: (select)
    LDAP Port: 14500
    Common Name Identifier: cn
    Distinguished Name (dn): c=us;o=netscreen

2. **External User**

Define the auth user "euclid" with password eTcS114u on the external LDAP auth server x_serv1.

3. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: euc_host
IP Address/Domain Name:
    IP/Netmask: (select), 10.1.1.20/32
Zone: Trust

4. **Policy**

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: ANY
Action: Permit
Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
Auth Server: (select)
    Use: x_srv1
User: (select), External User
External User: euclid

CLI

1. **Auth Server**

```
set auth-server x_srv1
set auth-server x_srv1 type ldap
set auth-server x_srv1 account-type auth
set auth-server x_srv1 server-name 10.1.1.100
set auth-server x_srv1 backup1 10.1.1.110
set auth-server x_srv1 backup2 10.1.1.120
set auth-server x_srv1 timeout 60
set auth-server x_srv1 ldap port 14500
set auth-server x_srv1 ldap cn cn
set auth-server x_srv1 ldap dn c=us;o=netscreen
```

2. **External User**

   Define the auth user “euclid” with password eTcS114u on the external LDAP auth server x_serv1.

3. **Address**

   ```
   set address trust euc_host 10.1.1.20/32
   ```

4. **Policy**

   ```
   set policy top from trust to untrust euc_host any any auth server x_srv1 user
   euclid
   save
   ```

Example: Run-Time Authentication (External User Group)

In this example, you configure an external RADIUS auth server named “radius1” and define an external auth user group named “ auth_grp2.” You define the external auth user group auth_grp2 in two places:

1. External RADIUS auth server “radius1”
2. Security device

NOTE: The RADIUS auth server configuration is nearly identical to that in “Example: RADIUS Auth Server” on page 1597, except that in this example you only specify “auth” as the user account type.

You populate the auth user group “auth_grp2” with auth users on the RADIUS server only, leaving the group unpopulated on the security device. The members in this group are accountants who require exclusive access to a server at IP address 10.1.1.80. You create an address book entry for the server and name the address “ midas.” You then configure an intrazone policy permitting only authenticated traffic from auth_grp2 to midas, both of which are in the Trust zone. (For more information on intrazone policies, see “Policies” on page 197.)

RADIUS Server

1. Load the RADIUS dictionary file on the RADIUS server.

NOTE: For information on the RADIUS dictionary file, see "RADIUS Dictionary File" on page 1585. For instructions on loading the dictionary file onto a RADIUS server, refer to the RADIUS server documentation.

If you are using a Microsoft IAS RADIUS server, there is no dictionary file to load. Instead, define the correct vendor-specific attributes (VSAs) on the server.

2. After you define auth user accounts on the RADIUS server, use the ScreenOS user group VSA to create the user group "auth_grp2" and apply it to the auth user accounts that you want to add to that group.

WebUI

1. **Auth Server**

   Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

   Name: radius1
   IP/Domain Name: 10.20.1.100
   Backup1: 10.20.1.110
   Backup2: 10.20.1.120
   Timeout: 30
   Account Type: Auth
   RADIUS: (select)
   RADIUS Port: 4500
   Shared Secret: A56htYY97kl

2. **External User Group**

   Objects > Users > External Groups > New: Enter the following, then click **OK**:

   Group Name: auth_grp2
   Group Type: Auth

3. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: midas
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.80/32
   Zone: Trust

4. **Policy**

   Policy > Policies > (From: Trust, To: Trust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any

Destination Address:
Address Book Entry: (select), midas
Service: ANY
Action: Permit
Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
Auth Server: (select)
Use: radius1
User Group: (select), External Auth Group - auth_grp2

CLI

1. **Auth-Server**

```
set auth-server radius1 type radius
set auth-server radius1 account-type auth
set auth-server radius1 server-name 10.20.1.100
set auth-server radius1 backup1 10.20.1.110
set auth-server radius1 backup2 10.20.1.120
set auth-server radius1 timeout 30
set auth-server radius1 radius port 4500
set auth-server radius1 radius secret A56htYY97kl
```

2. **External User Group**

```
set user-group auth_grp2 location external
set user-group auth_grp2 type auth
```

3. **Address**

```
set address trust midas 10.1.1.80/32
```

4. **Policy**

```
set policy top from trust to trust any midas any permit auth server radius1
user-group auth_grp2
save
```

Example: Local Auth User in Multiple Groups

In this example, you define a local auth user named Mary. Mary is a sales manager who needs access to two servers: server A, which is for the salespeople (sales_reps group), and server B, which is for the managers (sales_mgrs group). To provide access to both, you add Mary to the two user groups. You then create two policies, one for each group.

NOTE: This example does not show the configuration for the other group members.

WebUI

1. **Local User**

   Objects > Users > Local > New: Enter the following, then click **OK**:

   User Name: mary
   Status: Enable
   Authentication User: (select)
   User Password: iFa8rBd
   Confirm Password: iFa8rBd

2. **Local User Groups and Member**

   Objects > Users > Local Groups > New: Enter **sales_mgrs** in the Group Name field, do the following, then click **OK**:

   Select **mary** and use the < < button to move her from the Available Members column to the Group Members column.

   Objects > Users > Local Groups > New: Enter **sales_reps** in the Group Name field, do the following, then click **OK**:

   Select **mary** and use the < < button to move her from the Available Members column to the Group Members column.

3. **Addresses**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: sales
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.8.0/24
   Zone: Trust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: server_a
   IP Address/Domain Name:
   IP/Netmask: (select), 1.1.1.5/32
   Zone: Untrust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: server_b
   IP Address/Domain Name:
   IP/Netmask: (select), 1.1.1.6/32
   Zone: Untrust

4. **Policies**

Policy > Policies > (From: Trust; To: Untrust) > New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), sales
Destination Address:
    Address Book Entry: (select), server_a
Service: FTP
Action: Permit
Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
Auth Server: (select)
    Use: Local
User Group: (select), Local Auth Group - sales_reps

Policy > Policies > (From: Trust; To: Untrust) > New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), sales
Destination Address:
    Address Book Entry: (select), server_b
Service: FTP
Action: Permit
Position at Top: (select)

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
Auth Server: (select)
    Use: Local
User Group: (select), Local Auth Group - sales_mgrs

CLI

1. **Local User**

```
set user mary password iFa8rBd
```

2. **Local User Groups and Member**

```
set user-group sales_mgrs location local
set user-group sales_mgrs user mary
set user-group sales_reps location local
set user-group sales_reps user mary
```

3. **Addresses**

```
set address trust sales 10.1.8.0/24
set address untrust server_a 1.1.1.5/32
```

```
set address untrust server_b 1.1.1.6/32
```

4. **Policy**

```
set policy top from trust to untrust sales server_a ftp permit auth user-group
sales_reps
set policy top from trust to untrust sales server_b ftp permit auth user-group
sales_mgrs
save
```

Example: WebAuth (Local User Group)

In this example, you require users to preauthenticate themselves via the WebAuth method before initiating outbound traffic to the Internet. You create a user group named "auth_grp3" in the local database on the security device. You then create auth user accounts for everyone in the Trust zone and add them to auth_grp3.

The Trust zone interface uses ethernet1 and has IP address 10.1.1.1/24. You assign 10.1.1.50 as the WebAuth IP address, and you use keep the local database as the default WebAuth server. Consequently, before a user can initiate traffic to the Internet, he or she must first make an HTTP connection to 10.1.1.50 and log in with a username and password. The security device then checks the username and password against those in its database and either approves or rejects the authentication request. If it approves the request, the authenticated user has 30 minutes to initiate traffic to the Internet. After terminating that initial session, the user has another 30 minutes to initiate another session before the security device requires him or her to reauthenticate himself or herself.

WebUI

1. **WebAuth**

   Configuration > Auth > WebAuth: Select **Local** from the WebAuth Server drop-down list, then click **Apply**.

   Network > Interfaces > Edit (for ethernet1): Select **WebAuth**, and in the WebAuth IP field enter **10.1.1.50**.

   Configuration > Auth > Auth Servers > Edit (for Local): Enter 30 in the Timeout field, then click **Apply**.

2. **User Group**

   Objects > Users > Local Groups > New: Enter **auth_grp3** in the Group Name field, do the following, then click **OK**:

   Select **user name** and use the < < button to move that user from the Available Members column to the Group Members column.

   Repeat the selection process, adding auth users until the group is complete.

3. **Policy**

Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: ANY
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
WebAuth: (select)
User Group: (select), Local Auth Group - auth_grp3

CLI

1. **WebAuth**

```
set webauth server Local
set interface ethernet1 webauth-ip 10.1.1.50
set interface ethernet1 webauth
set auth-server Local timeout 30
```

2. **User Group**

```
set user-group auth_grp3 location local
```

NOTE: The security device determines a local user group type by the type of members that you add to it. To make auth_grp3 an auth user group, add an auth user to the group.

Use the following command to add auth users to the user group you have just created:

```
set user-group auth_grp3 user name_str
```

3. **Policy**

```
set policy top from trust to untrust any any any permit webauth user-group
auth_grp3
save
```

Example: WebAuth (External User Group)

WebAuth is a method for pre-authenticating users before they initiate traffic across the firewall. In this example, you create a policy requiring authentication via the WebAuth method for all outgoing traffic.

You create an auth user group named “auth_grp4” on both the RADIUS server “ radius1” and on the security device. On the RADIUS server, you create user accounts for everyone in the Trust zone and add them to auth_grp4.

NOTE: Nearly the same RADIUS server settings are used here as in “Example: RADIUS Auth Server” on page 1597, except that in this example you only specify “auth” as the user account type.

The Trust zone interface uses ethernet1 and has IP address 10.1.1.1/24. You assign 10.1.1.50 as the WebAuth IP address, and you use the external RADIUS auth-server “radius1” as the default WebAuth server. Consequently, before a user can initiate traffic to the Internet, he or she must first make an HTTP connection to 10.1.1.50 and log in with a username and password. The security device then relays all WebAuth user authentication requests and responses between “ radius1” and the users attempting to log in.

RADIUS Server

1. Load the RADIUS dictionary file on the RADIUS server.

NOTE: For information on the RADIUS dictionary file, see “RADIUS Dictionary File” on page 1585. For instructions on loading the dictionary file onto a RADIUS server, refer to the RADIUS server documentation.

2. Enter user group “auth_grp4” on the auth-server “ radius1” , and then populate it with auth user accounts.

WebUI

1. **Auth-Server**

   Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

   Name: radius1
   IP/Domain Name: 10.20.1.100
   Backup1: 10.20.1.110
   Backup2: 10.20.1.120
   Timeout: 30
   Account Type: Auth
   RADIUS: (select)
   RADIUS Port: 4500
   Shared Secret: A56htYY97k

2. **WebAuth**

   Configuration > Auth > WebAuth: Select **radius1** from the WebAuth Server drop-down list, then click **Apply**.

Network > Interfaces > Edit (for ethernet1): Select **WebAuth**, in the WebAuth IP field enter **10.10.1.50**, then click **OK**.

3. **User Group**

   Objects > Users > External Groups > New: Enter the following, then click **OK**:

   Group Name: auth_grp4
   Group Type: Auth

4. **Policy**

   Policy > Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select), Any
   Destination Address:
   Address Book Entry: (select), Any
   Service: ANY
   Action: Permit

   > Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

   Authentication: (select)
   WebAuth: (select)
   User Group: (select), External Auth Group - auth_grp4

CLI

1. **Auth-Server**

   ```
   set auth-server radius1 type radius
   set auth-server radius1 account-type auth
   set auth-server radius1 server-name 10.20.1.100
   set auth-server radius1 backup1 10.20.1.110
   set auth-server radius1 backup2 10.20.1.120
   set auth-server radius1 timeout 30
   set auth-server radius1 radius port 4500
   set auth-server radius1 radius secret A56htYY97kl
   ```

2. **WebAuth**

   ```
   set webauth server radius1
   set interface ethernet1 webauth-ip 10.1.1.50
   set interface ethernet1 webauth
   ```

3. **User Group**

   ```
   set user-group auth_grp4 location external
   set user-group auth_grp4 type auth
   ```

4. **Policy**

```
set policy top from trust to untrust any any any permit webauth user-group
auth_grp4
save
```

Example: WebAuth + SSL Only (External User Group)

In this example, you combine WebAuth with Secure Sockets Layer (SSL) technologies to provide security for the usernames and passwords that users transmit when logging in. WebAuth makes use of the same certificate that secures administrative traffic to the security device for management via the WebUI. (For more information about SSL, see "Secure Sockets Layer" on page 315.)

The configuration for WebAuth using an external auth server plus SSL involves the following steps:

- You define an external RADIUS auth-server "radius1" and create an auth user group named " auth_grp5" on both the RADIUS server and on the security device. On the RADIUS server, you create user accounts for all auth users in the Untrust zone and add them to auth_grp5.

NOTE: Nearly identical RADIUS server settings are used here as in "Example: RADIUS Auth Server" on page 1597, except that you only specify "auth" as the user account type here.

- The Untrust zone interface uses ethernet3 and has IP address 1.1.1.1/24. You assign 1.1.1.50 as the WebAuth IP address, instruct the device to accept only SSL connections for WebAuth authentication requests, and define the external RADIUS auth-server "radius1" as the default WebAuth server.
- You specify the following SSL settings:
  - IDX number (1 in this example) of a certificate that you have previously loaded on the security device
  - DES_SHA-1 ciphers
  - SSL port number 2020
- You then configure an incoming policy requiring authentication via the WebAuth + SSL method for all traffic from the Untrust to Trust zones.

NOTE: For information on how to obtain and load digital certificates onto a security device, see "Public Key Cryptography" on page 741.

Consequently, before a user can initiate traffic to the internal network, he or she must first make an HTTPS connection to https://1.1.1.50:2020 and log in with a username and password. The security device then relays all WebAuth user authentication requests and responses between "radius1" and the user attempting to log in.

RADIUS Server

1. Load the RADIUS dictionary file on the RADIUS server.

NOTE: For information on the dictionary file, see “RADIUS Dictionary File” on page 1585. For instructions on loading the dictionary file onto a RADIUS server, refer to the RADIUS server documentation.

2. Enter user group “auth_grp5” on the auth-server “ radius1,” and then populate it with auth user accounts.

WebUI

1. **Auth-Server**

   Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

   Name: radius1
   IP/Domain Name: 10.20.1.100
   Backup1: 10.20.1.110
   Backup2: 10.20.1.120
   Timeout: 30
   Account Type: Auth
   RADIUS: (select)
   RADIUS Port: 4500
   Shared Secret: A56htYY97k

2. **WebAuth**

   Configuration > Auth > WebAuth: Select **radius1** from the WebAuth Server drop-down list, then click **Apply**.

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

   WebAuth: (select)
   IP: 1.1.1.50
   SSL Only: (select)

3. **SSL**

   Configuration > Admin > Management: Enter the following, then click **OK**:

   HTTPS (SSL) Port: 2020
   Certificate: (select the certificate that you previously loaded)
   Cipher: DES_SHA-1

4. **User Group**

   Objects > Users > External Groups > New: Enter the following, then click **OK**:

Group Name: auth_grp5
Group Type: Auth

5. **Policy**

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
    Address Book Entry: (select), Any
Destination Address:
    Address Book Entry: (select), Any
Service: ANY
Action: Permit

> Advanced: Enter the following, then click **Return** to set the advanced options and return to the basic configuration page:

Authentication: (select)
WebAuth: (select)
User Group: (select), External Auth Group - auth_grp5

CLI

1. **Auth-Server**

```
set auth-server radius1 type radius
set auth-server radius1 account-type auth
set auth-server radius1 server-name 10.20.1.100
set auth-server radius1 backup1 10.20.1.110
set auth-server radius1 backup2 10.20.1.120
set auth-server radius1 timeout 30
set auth-server radius1 radius port 4500
set auth-server radius1 radius secret A56htYY97kl
```

Load the RADIUS dictionary file on the RADIUS server.

NOTE: For information on the RADIUS dictionary file, see “RADIUS Dictionary File” on page 1585. For instructions on loading the dictionary file onto a RADIUS server, refer to the RADIUS server documentation.

2. **WebAuth**

```
set webauth server radius1
set interface ethernet3 webauth-ip 1.1.1.50
set interface ethernet3 webauth ssl-only
```

3. **SSL**

```
set ssl port 2020
set ssl cert 1
set ssl encrypt des sha-1
```

```
set ssl enable
```

4. **User Group**

```
set user-group auth_grp5 location external
set user-group auth_grp5 type auth
```

5. **Policy**

```
set policy top from untrust to trust any any any permit webauth user-group
auth_grp5
save
```

Chapter 50

IKE, XAuth, and L2TP Users

This chapter covers the three types of users involved with tunneling protocols—Internet Key Exchange (IKE) users, XAuth users, and Layer 2 Transport Protocol (L2TP) users. It contains the following sections:

NOTE: For more information and examples for IKE and L2TP, see *"Virtual Private Networks" on page 705*.

IKE Users and User Groups

An IKE user is a remote VPN user with a dynamically assigned IP address. The user—actually, the user's device—authenticates itself by sending either a certificate or preshared key together with an IKE ID during Phase 1 negotiations with the security device.

The IKE ID can be an email address, an IP address, a domain name, or ASN1-DN string. A security device authenticates an IKE user if the user sends either of the following:

- A **certificate** in which one or more of the values that appear in the distinguished name (DN) fields or in the SubAltName field is the same as the user's IKE ID configured on the security device
- A **preshared key** and an **IKE ID**, and the security device can successfully generate an identical preshared key from the received IKE ID and a preshared key seed value stored on the security device

NOTE: An example of an IKE ID using the Abstract Syntax Notation, version 1, distinguished name (ASN1-DN) format is
CN = fiona,OU = it,O = juniper,L = sunnyvale,ST = ca,C = us,E = fiona@juniper.net.

You reference an IKE user or user group in an AutoKey IKE gateway configuration. By gathering IKE users that require similar gateway and tunnel configurations into

a group, you only need to define one gateway referencing the group (and one VPN tunnel referencing that gateway), instead of one gateway and tunnel for each IKE user.

It is often impractical to create separate user accounts for every host. In such cases, you can create an IKE user group that has only one member, referred to as a group IKE ID user. The IKE ID of that user contains a set of values that must be present in the dialup IKE users' IKE ID definitions. If the IKE ID of a remote dialup IKE user matches the IKE ID of the group IKE ID user, ScreenOS authenticates that remote user. For more information, see "Group IKE ID" on page 911.

NOTE: You can only store IKE user and IKE user group accounts on the local database.

Example: Defining IKE Users

In this example, you define four IKE users, Amy, Basil, Clara, and Desmond, each with a different kind of IKE ID.

- Amy – email address (user-fully qualified domain name or U-FQDN): amy@juniper.net
- Basil – IP address: 3.3.1.1
- Clara – fully qualified domain name (FQDN): www.juniper.net
- Desmond – ASN1-DN string:
  CN = des,OU = art,O = juniper,L = sunnyvale,ST = ca,C = us,E = des@juniper.net

WebUI

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Amy
Status: Enable
IKE User: (select)
Simple Identity: (select)
IKE ID Type: AUTO
IKE Identity : amy@juniper.net

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Basil
Status: Enable
IKE User: (select)
Simple Identity: (select)
IKE ID Type: AUTO
IKE Identity: 3.3.1.1

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Clara
Status: Enable
IKE User: (select)

Simple Identity: (select)
IKE ID Type: AUTO
IKE Identity: www.juniper.net

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: Desmond
Status: Enable
IKE User: (select)
Use Distinguished Name for ID: (select)
CN: des
OU: art
Organization: juniper
Location: sunnyvale
State: ca
Country: us
E-mail: des@juniper.net

CLI

```
set user Amy ike-id u-fqdn amy@juniper.net
set user Basil ike-id ip 3.3.1.1
set user Clara ike-id fqdn www.juniper.net
set user Desmond ike-id wildcard
CN=des,OU=art,O=juniper,L=sunnyvale,ST=ca,C=us,E=des@juniper.net
save
```

Example: Creating an IKE User Group

In this example, you create a user group named ike_grp1. It becomes an IKE user group when you add IKE user Amy to it. You then add the other three IKE uses that you defined in "Example: Defining IKE Users" on page 1638.

WebUI

Objects > Users > Local Groups > New: Enter **ike_grp1** in the Group Name field, do the following, then click **OK**:

Select **Amy** and use the < < button to move her from the Available Members column to the Group Members column.

Select **Basil** and use the < < button to move him from the Available Members column to the Group Members column.

Select **Clara** and use the < < button to move her from the Available Members column to the Group Members column.

Select **Desmond** and use the < < button to move him from the Available Members column to the Group Members column.

CLI

```
set user-group ike_grp1 location local
```

```
set user-group ike_grp1 user amy
set user-group ike_grp1 user basil
set user-group ike_grp1 user clara
set user-group ike_grp1 user desmond
save
```

Referencing IKE Users in Gateways

After you define an IKE user or IKE user group, you can then reference it in an IKE gateway configuration when the remote IKE gateway is a dialup user or dialup user group.

To see examples that reference IKE users in gateway configurations, see:

- Policy-Based Dialup VPN, AutoKey IKE on page 888
- Creating a Group IKE ID (Certificates) on page 915
- Group IKE ID with Certificates on page 911

XAuth Users and User Groups

The XAuth protocol is composed of two components: remote VPN user authentication (username plus password) and TCP/IP address assignments (IP address, netmask, DNS server, and WINS server assignments). ScreenOS supports the application of either component by itself or both components in concert.

NOTE: The assigned netmask is always 255.255.255.255 and cannot be modified.

An XAuth user or user group is one or more remote users who authenticate themselves when connecting to the security device via an AutoKey IKE VPN tunnel and optionally receive TCP/IP settings from the security device. Whereas the authentication of IKE users is actually the authentication of VPN gateways or clients, the authentication of XAuth users is the authentication of the individuals themselves. XAuth users must enter information that only they are supposed to know—their username and password.

The ScreenOS-Remote client can use the TCP/IP settings it receives to create a virtual adapter when sending VPN traffic—while using the TCP/IP network adapter settings provided by the ISP or network admin for non-VPN traffic. By assigning known IP addresses to remote users, you can define routes on the security device to those addresses via specific tunnel interfaces. Then the security device can ensure that return routing reaches the remote user's IP address through the VPN tunnel, not via the default gateway. Address assignments also allow a downstream firewall to reference those addresses when creating policies. You can control the length of time that an IP address is associated with an individual XAuth user with the XAuth lifetime setting.

NOTE: A virtual adapter is the TCP/IP settings (IP address, DNS server addresses, WINS server addresses) that the security device assigns to a remote user for the duration of a VPN tunnel connection. Only ScreenOS-Remote clients support virtual adapter functionality. Juniper Networks security platforms do not.

ScreenOS supports the following aspects of XAuth:

- Authentication of local XAuth users and external XAuth users
- Authentication of local XAuth user groups and external XAuth user groups if stored on a RADIUS auth server
- IP, DNS server, and WINS server address assignments from an IP address pool for local XAuth users and external XAuth users stored on a RADIUS auth server

  To configure the security device to use default XAuth settings stored on an external RADIUS server, do either of the following:

  - WebUI: On the VPNs > AutoKey Advanced > XAuth Settings page, select **Query Client Settings on Default Server**.
  - CLI: Enter the **set xauth default auth server** name_str **query-config** command.

  The security device can also use gateway-specific XAuth settings stored on an external RADIUS server. When configuring a specific IKE gateway, do either of the following:

  - WebUI: On the VPNs > AutoKey Advanced > Gateway > New > Advanced page, select the name of the RADIUS server from the External Authentication drop-down list, and then select **Query Remote Setting**.
  - CLI: Enter the **set ike gateway** name_str **xauth server** name_str **query-config** command.
- Authentication only without address assignments, address assignments only without authentication (**set ike gateway** name_str **xauth bypass-auth**), and both authentication and address assignments in combination
- Authentication and accounting on different RADIUS auth servers. For more information about configuring a separate RADIUS accounting server for XAuth users, see “Configuring a Separate External Accounting Server” on page 1604.

Event Logging for IKE Mode

When a remote user accesses the network through Internet Key Exchange (IKE), ScreenOS authenticates the user with XAuth; ScreenOS records the event details in the traffic log. The log details include the following:

- Gateway IP address
- Username
- Number of session retries
- Allocated client IP address from the local IP pool or RADIUS server

For more information about viewing the traffic log, see *"Administration" on page 307*.

XAuth Users in IKE Negotiations

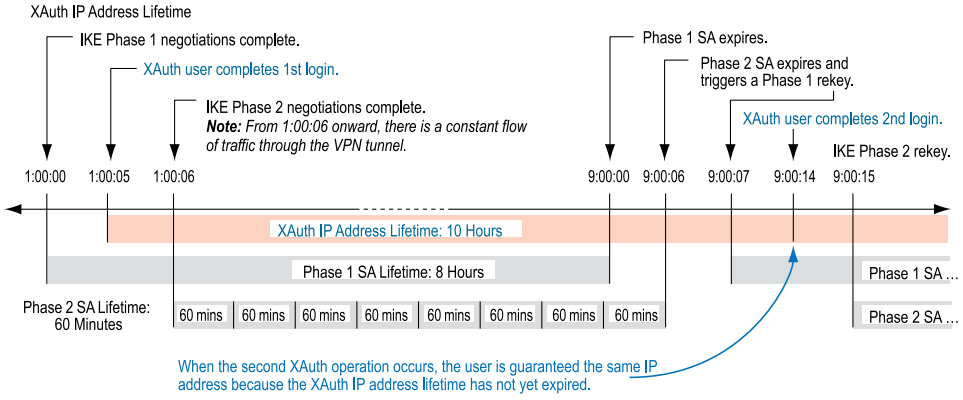
ScreenOS supports XAuth, version 6 (v6). To confirm that both parties in Phase 1 IKE negotiations support XAuth v6, they each send the following vendor ID to each other in the first two Phase 1 messages: 0x09002689DFD6B712. This vendor ID number is specified in the XAuth Internet draft, draft-beaulieu-ike-xauth-02.txt.

After the completion of Phase 1 negotiations, the security device sends a login prompt to the XAuth user at the remote site. If the XAuth user successfully logs on with the correct username and password, the security device assigns an IP address, 32-bit netmask, DNS server addresses, and WINS server addresses to the user, and the two parties continue with Phase 2 negotiations.

The XAuth user has 60 seconds to complete the login process. If the first login attempt fails, the XAuth user can make up to four more attempts, having 60 seconds for each attempt. If the user fails after five consecutive attempts, the security device stops providing a login prompt and severs the session.

At a minimum, the XAuth-assigned IP address belongs to a user for the duration specified as the XAuth address lifetime. The IP address might belong to the XAuth user longer, depending on when the Phase 1 and Phase 2 security associations (SAs) rekey. Figure 414 on page 1642 shows the relationship of Phase 1 and Phase 2 rekey operations and the XAuth IP address lifetime.

Figure 414: Phases 1 and 2 Rekey Operations and XAuth IP Address Lifetime



1. The Phase 1 SA is set with an 8-hour lifetime and expires after the first 8 hours.
2. The Phase 2 SA lifetime is set for 60 minutes. Because there is a 5-second delay during the initial IKE negotiations while the XAuth user enters his username and password, the eighth Phase 2 SA expires 8 hours and 6 seconds (5 seconds for the XAuth login + 1 second for Phase 2 negotiations) after Phase 1 negotiations complete.
3. Because there is active VPN traffic, the expiration of the eighth Phase 2 SA causes the Phase 1 SA, which expired 6 seconds prior, to rekey; that is, Phase 1 IKE negotiations (or "renegotiations") occur.

4. After Phase 1 IKE renegotiations complete, the security device prompts the XAuth user to log in again.

NOTE: To avoid repeating further logins after the initial login, configure the VPN tunnel with any idletime other than 0 with the CLI command: **set vpn** name **gateway** name **idletime** number (in minutes). If there is VPN activity at the completion of Phase 1 IKE renegotiations, the security device does not prompt the XAuth user to log in again. This option enables the user to download large files, transmit or receive streaming media, of participate in web conferences without interruption.

5. Because the XAuth address lifetime (10 hours) exceeds the Phase 1 SA lifetime, the user keeps the same IP address—although the user might get a different address after the next Phase 1 rekey occurs.

   If the XAuth address lifetime had been shorter than the Phase 1 SA lifetime, the security device would have assigned the user another IP address, which might or might not have been the same as the previously assigned address.

NOTE: If it is crucial that a user always be assigned the same IP address, you can specify an address in the user configuration. The security device then assigns this address instead of assigning one at random from an IP pool. Note that such an address must not be in an IP pool or it might get assigned to another user and be unavailable when needed.

To change the address lifetime, do either of the following:

- (WebUI) VPNs > AutoKey Advanced > XAuth Settings: Enter a number (minutes) in the Reserve Private IP for XAuth User field, then click **Apply**.
- (CLI) set xauth lifetime *number*

  To effectively disable the address lifetime feature, enter a value of 1—the minimum value allowed.

Example: XAuth Authentication (Local User)

In this example, you define an XAuth user named x1 with password aGgb80L0ws on the local database.

You then reference this user in a remote IKE gateway configuration to a peer at IP 2.2.2.2. You name the remote gateway **gw1**, specify main mode and the proposal **pre-g2-3des-sha** for Phase 1 negotiations, and use the preshared key **juniper1**. You name the VPN tunnel **vpn1** and specify the Compatible set of proposals for Phase 2 negotiations. You choose the Untrust zone interface ethernet3 as the outgoing interface.

WebUI

1. **XAuth User**

Objects > Users > Local > New: Enter the following, then click **OK**:

User Name: x1
Status: Enable
XAuth User: (select)
User Password: iDa84rNk
Confirm Password: iDa84rNk

2. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: gw1
Security Level: Custom
Remote Gateway Type:
Static IP Address: (select), Address/Hostname: 2.2.2.2
Preshared Key: juniper1
Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom: (select)
Phase 1 Proposal: pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)
XAuth Server: (select)
Local Authentication: (select)
User: (select), x1

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn1
Security Level: Compatible
Remote Gateway Tunnel: gw1

CLI

1. **XAuth User**

```
set user x1 password aGgb80L0ws
set user x1 type xauth
unset user x1 type auth
```

NOTE: The CLI command **set user** name_str **password** pswd_str creates an auth user. To create an XAuth-only user, you must define the user as an XAuth user (**set user** name_str type **xauth**), and then remove the auth user definition (**unset user** name_str **type auth**).

2. **VPN**

```
set ike gate gw1 ip 2.2.2.2 main outgoing-interface ethernet3 preshare juniper1
 proposal pre-g2-3des-sha
set ike gateway gw1 xauth server Local user x1
set vpn vpn1 gateway gw1 sec-level compatible
save
```

Example: XAuth Authentication (Local User Group)

In this example, you create a user group named xa-grp1 on the local database and add the XAuth user "x1" that you created in the previous example, "Example: XAuth Authentication (Local User)" on page 1643. When you add that user to the group, it automatically becomes an XAuth user group.

You then reference this group in a remote IKE gateway configuration to a peer at IP 2.2.2.2. You name the remote gateway "gw2," specify main mode and the proposal pre-g2-3des-sha for Phase 1 negotiations, and use the preshared key " juniper2." You name the VPN tunnel " vpn2" and specify the " Compatible" set of proposals for Phase 2 negotiations. You choose the Untrust zone interface ethernet3 as the outgoing interface.

WebUI

1. **XAuth User Group**

   Objects > Users > Local Groups > New: Enter **xa-grp1** in the Group Name field, do the following, then click **OK**:

   Select **x1** and use the < < button to move him from the Available Members column to the Group Members column.

2. **VPN**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: gw2
   Security Level: Custom
   Remote Gateway Type:
   Static IP Address: (select), Address/Hostname: 2.2.2.2
   Preshared Key: juniper2
   Outgoing Interface: ethernet3

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

   Phase 1 Proposal: pre-g2-3des-sha
   Mode (Initiator): Main (ID Protection)
   XAuth Server: (select)
   Local Authentication: (select)
   User Group: (select), xa-grp1

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: vpn2

Security Level: Compatible
Remote Gateway Tunnel:
Predefined: (select), gw2

CLI

1. **XAuth User Group**

```
set user-group xa-grp1 location local
set user-group xa-grp1 user x1
```

2. **VPN**

```
set ike gate gw2 ip 2.2.2.2 main outgoing-interface ethernet3 preshare juniper2
 proposal pre-g2-3des-sha
set ike gateway gw2 xauth server Local user-group xa-grp1
set vpn vpn2 gateway gw2 sec-level compatible
save
```

Example: XAuth Authentication (External User)

In this example, you reference an XAuth user named xa-1 with password iNWw10bd01 that you have previously loaded on an external SecurID auth server. This example uses almost the same configuration of the SecurID auth server as defined in "Example: SecurID Auth Server" on page 1599, except that here you define the account type as XAuth.

You reference XAuth user xa-1 in a remote IKE gateway configuration to a peer at IP 2.2.2.2. You name the remote gateway **gw3**, specify main mode and the proposal **pre-g2-3des-sha** for Phase 1 negotiations, and use the preshared key **juniper3**. You name the VPN tunnel **vpn3** and specify the proposal **g2-esp-3des-sha** for Phase 2 negotiations. You choose the Untrust zone interface ethernet3 as the outgoing interface.

WebUI

1. **External SecurID Auth Server**

   Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

   Name: securid1
   IP/Domain Name: 10.20.2.100
   Backup1: 10.20.2.110
   Timeout: 60
   Account Type: XAuth
   SecurID: (select)
   Client Retries: 3
   Client Timeout: 10 seconds
   Authentication Port: 15000
   Encryption Type: DES
   User Duress: No

2. **XAuth User**

   Define the auth user "xa-1" with password iNWw10bd01 on the external SecurID auth server securid1.

3. **VPN**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: gw3
   Security Level: Custom
   Remote Gateway Type:
   Static IP Address: (select), Address/Hostname: 2.2.2.2
   Preshared Key: juniper3
   Outgoing Interface: ethernet3

   > Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

   Phase 1 Proposal: pre-g2-3des-sha
   Mode (Initiator): Main (ID Protection)
   XAuth Server: (select)
   External Authentication: (select), securid1
   User: (select)
   Name: xa-1

   VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

   VPN Name: vpn3
   Security Level: Compatible
   Remote Gateway Tunnel:
   Predefined: (select), gw3

CLI

1. **External SecurID Auth Server**

```
set auth-server securid1 type securid
set auth-server securid1 server-name 10.20.2.100
set auth-server securid1 backup1 10.20.2.110
set auth-server securid1 timeout 60
set auth-server securid1 account-type xauth
set auth-server securid1 securid retries 3
set auth-server securid1 securid timeout 10
set auth-server securid1 securid auth-port 15000
set auth-server securid1 securid encr 1
set auth-server securid1 securid duress 0
```

2. **XAuth User**

   Define the auth user "xa-1" with password iNWw10bd01 on the external SecurID auth-server securid1.

3. **VPN**

```
set ike gate gw3 ip 2.2.2.2 main outgoing-interface ethernet3 preshare juniper3
 proposal pre-g2-3des-sha
set ike gateway gw3 xauth server securid1 user xa-1
set vpn vpn3 gateway gw3 sec-level compatible
save
```

Example: XAuth Authentication (External User Group)

In this example, you configure an external RADIUS auth server named “radius1” and define an external auth user group named “ xa-grp2.” You define the external XAuth user group xa-grp2 in two places:

1. External RADIUS auth server “radius1”
2. Security device

NOTE: The RADIUS auth server configuration is nearly identical to that in “Example: RADIUS Auth Server” on page 1597, except that in this example you only specify “xauth” as the user account type.

You populate the XAuth user group “xa-grp2” with XAuth users on the RADIUS server only, leaving the group unpopulated on the security device. The members in this group are resellers at a remote site who require access to FTP servers in the corporate LAN. You add an entry in the Untrust zone address book for the remote site with IP address 10.2.2.0/24 and the name reseller1. You also enter an address in the Trust zone address book for the FTP server “ rsl-srv1” with IP address 10.1.1.5/32.

You configure a VPN tunnel to 2.2.2.2 to authenticate XAuth users in the user group xa-grp2. You name the remote gateway “gw4,” specify main mode and the proposal pre-g2-3des-sha for Phase 1 negotiations, and use the preshared key “ juniper4.” You name the VPN tunnel “ vpn4” and specify the “ Compatible” set of proposals for Phase 2 negotiations. You choose the Untrust zone interface ethernet3 as the outgoing interface.

Finally, you create a policy permitting FTP traffic from reseller1 in the Untrust zone via vpn4 to rsl-svr1 in the Trust zone.

RADIUS Server

1. Load the RADIUS dictionary file on the RADIUS server.

NOTE: For information on the RADIUS dictionary file, see “RADIUS Dictionary File” on page 1585. For instructions on loading the dictionary file onto a RADIUS server, refer to the RADIUS server documentation.

2. Enter auth user group “xa-grp2” on the external auth server “ radius1“ , and then populate it with XAuth user accounts.

WebUI

1. **Auth Server**

   Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

   Name: radius1
   IP/Domain Name: 10.20.1.100
   Backup1: 10.20.1.110
   Backup2: 10.20.1.120
   Timeout: 30
   Account Type: XAuth
   RADIUS: (select)
   RADIUS Port: 4500
   Shared Secret: A56htYY97kl

2. **External User Group**

   Objects > Users > External Groups > New: Enter the following, then click **OK**:

   Group Name: xa-grp2
   Group Type: XAuth

3. **Address**

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: reseller1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.2.2.0/24
   Zone: Untrust

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: rsl-svr1
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.5/32
   Zone: Trust

4. **VPN**

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

   Gateway Name: gw4
   Security Level: Custom
   Remote Gateway Type:
   Static IP Address: (select), Address/Hostname: 2.2.2.2
   Preshared Key: juniper4
   Outgoing Interface: ethernet3

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Phase 1 Proposal: pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)
XAuth Server: (select)
External Authentication: (select), securid1
User Group: (select)
Name: xa-grp2

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: vpn4
Security Level: Compatible
Remote Gateway:
Predefined: (select), gw4

5. **Policy**

Policy > Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), reseller1
Destination Address:
Address Book Entry: (select), rsl-svr1
Service: FTP-Get
Action: Tunnel
Tunnel VPN: vpn4
Modify matching bidirectional VPN policy: (clear)
Position at Top: (select)

CLI

1. **Auth Server**

```
set auth-server radius1 type radius
set auth-server radius1 account-type xauth
set auth-server radius1 server-name 10.20.1.100
set auth-server radius1 backup1 10.20.1.110
set auth-server radius1 backup2 10.20.1.120
set auth-server radius1 timeout 30
set auth-server radius1 radius port 4500
set auth-server radius1 radius secret A56htYY97kl
```

2. **External User Group**

```
set user-group xa-grp2 location external
set user-group xa-grp2 type xauth
```

3. **Address**

```
set address untrust reseller1 10.2.2.0/24
set address trust rsl-svr1 10.1.1.5/32
```

4. **VPN**

```
set ike gate gw4 ip 2.2.2.2 main outgoing-interface ethernet3 preshare juniper4
 proposal pre-g2-3des-sha
set ike gateway gw4 xauth server radius1 user-group xa-grp2
set vpn vpn4 gateway gw4 sec-level compatible
```

5. **Policy**

```
set policy top from untrust to trust reseller1 rsl-svr1 ftp-get tunnel vpn vpn4
save
```

Example: XAuth Authentication and Address Assignments (Local User Group)

In this example, you set up both authentication and IP, DNS server, and WINS server IP address assignments for an IKE/XAuth user group stored on the local database.

NOTE: You can also use an external RADIUS auth server for XAuth user authentication and address assignments.You can use an external SecurID or LDAP auth server for XAuth authentication only (not for address assignments). For IKE user authentication, you can only use the local database.

When an IKE/XAuth user makes a dialup VPN connection to the security device, the security device authenticates the IKE user (that is, the client device) using an IKE ID and an RSA certificate during Phase 1 negotiations. The security device then authenticates the XAuth user (that is, the individual using the device) using a username and password and assigns IP, DNS server, and WINS server IP addresses between Phase 1 and Phase 2 negotiations.

You create a local user group ixa-grp1. You then define two IKE/XAuth users named “ixa-u1” (password: ccF1m84s) and “ ixa-u2” (password: C113g1tw) and add them to the group, thereby defining the group type as IKE/XAuth. (The addition of other IKE/XAuth users to the group is not included in the example.)

You create a DIP pool named xa-pool1 with an address range from 10.2.2.1 to 10.2.2.100. This is the pool of addresses from which the security device draws an IP address when assigning one to an XAuth user.

NOTE: The DIP pool must be in a different address space than that of the zone to which the XAuth user directs traffic to avoid routing problems and duplicate address assignments.

You configure the following XAuth default settings:

- Set the XAUTH address timeout to 480 minutes.
- Select the local database as the default auth server.

- Enable Challenge Handshake Authentication Protocol (CHAP), in which the security device sends a challenge (encryption key) to the remote client, who uses the key to encrypt his or her login name and password.
- Select xa-pool1 as the default DIP pool.
- Define the primary and secondary DNS server IP addresses as 10.1.1.150 and 10.1.1.151, respectively.
- Define the primary and secondary WINS server IP addresses as 10.1.1.160 and 10.1.1.161, respectively.

You configure an IKE gateway named "ixa-gw1," referencing user group ixa-grp1 and using the default XAuth auth server settings. You then configure a VPN tunnel name named " ixa-tun1" and a policy permitting traffic from ixa-grp1 to the Trust zone (IP address 10.1.1.0/24) via VPN tunnel ixa-tun1.

WebUI

1. **IKE/XAuth Users and User Group**

   Objects > Users > Local > New: Enter the following, then click **OK**:

   User Name: ixa-u1
   Status: Enable
   IKE User: (select)
   Simple Identity: (select)
   IKE ID Type: AUTO
   IKE Identity: u1@juniper.net
   XAuth User: (select)
   User Password: ccF1m84s
   Confirm Password: ccF1m84s

   Objects > Users > Local > New: Enter the following, then click **OK**:

   User Name: ixa-u2
   Status: Enable
   IKE User: (select)
   Simple Identity: (select)
   IKE ID Type: AUTO
   IKE Identity: u2@juniper.net
   XAuth User: (select)
   User Password: C113g1tw
   Confirm Password: C113g1tw

   Objects > Users > Local Groups > New: Enter **ixa-grp1** in the Group Name field, do the following, then click **OK**:

   Select **ixa-u1** and use the < < button to move him from the Available Members column to the Group Members column.

   Select **ixa-u2** and use the < < button to move him from the Available Members column to the Group Members column.

2. **IP Pool**

Objects > IP Pools > New: Enter the following, then click **OK**:

IP Pool Name: xa-pool1
Start IP: 10.2.2.1
End IP: 10.2.2.100

3. **Default XAuth Auth Server**

VPNs > AutoKey Advanced > XAuth Settings: Enter the following, then click **Apply**:

Reserve Private IP for XAuth User: 480 Minutes
Default Authentication Server: Local
Query Client Settings on Default Server: (clear)
CHAP: (select)
IP Pool Name: xa-pool1
DNS Primary Server IP: 10.1.1.150
DNS Secondary Server IP: 10.1.1.151
WINS Primary Server IP: 10.1.1.160
WINS Secondary Server IP: 10.1.1.161

4. **Address**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: Trust_zone
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

5. **VPN**

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: ixa-gw1
Security Level: Custom
Remote Gateway Type:
Dialup User Group: (select)
Group: ixa-grp1

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Phase 1 Proposal: rsa-g2-3des-sha
Mode (Initiator): Aggressive
Outgoing Interface: ethernet3
XAuth Server: (select)
User Default: (select)
User Group: (select), ixa-grp1

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: ixa-vpn1
Security Level: Compatible

Remote Gateway:
Predefined: (select), ixa-gw1

6. **Policy**

Policy > Policies > (From: Untrust; To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Dial-Up VPN
Destination Address:
Address Book Entry: (select), Trust_zone
Service: ANY
Action: Tunnel
Tunnel VPN: ixa-vpn1
Modify matching bidirectional VPN policy: (clear)
Position at Top: (select)

CLI

1. **IKE/XAuth Users and User Group**

```
set user-group ixa-grp1 location local
set user ixa-u1 type ike xauth
set user ixa-u1 ike-id u-fqdn u1@ns.com
set user ixa-u1 password ccF1m84s
unset user ixa-u1 type auth
set user ixa-u2 type ike xauth
set user ixa-u2 ike-id u-fqdn u2@juniper.net
set user ixa-u2 password C113g1tw
unset user ixa-u2 type auth
```

2. **IP Pool**

```
set ippool xa-pool1 10.2.2.1 10.2.2.100
```

3. **Default XAuth Auth Server**

```
set xauth lifetime 480
set xauth default auth server Local chap
set xauth default ippool xa-pool1
set xauth default dns1 10.1.1.150
set xauth default dns2 10.1.1.151
set xauth default wins1 10.1.1.160
set xauth default wins2 10.1.1.161
```

4. **Address**

```
set address trust Trust_zone 10.1.1.0/24
```

5. **VPN**

```
set ike gateway ixa-gw1 dialup ixa-grp1 aggressive outgoing-interface ethernet3
 proposal rsa-g2-3des-sha
set ike gateway ixa-gw1 xauth server Local user-group ixa-grp1
```

```
set vpn ixa-vpn1 gateway ixa-gw1 sec-level compatible
```

6. **Policy**

```
set policy top from untrust to trust "Dial-Up VPN"  Trust_zone any tunnel vpn
ixa-vpn1
save
```

XAuth Client

An XAuth client is a remote user or device that connects to an XAuth server via an AutoKey IKE VPN tunnel. A security device can act as an XAuth client, responding to authentication requests from a remote XAuth server. After the completion of Phase 1 negotiations, the remote XAuth server sends a login prompt to the security device. If the security device acting as an XAuth client successfully logs in with the correct username and password, Phase 2 negotiations commence.

To configure the security device as an XAuth client, you must specify the following:

- IKE gateway name
- XAuth username and password

You can configure the following types of XAuth authentication:

- **Any** — Allows either Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP)
- **CHAP** — Allows CHAP only

Example: Security Device as an XAuth Client

In this example, you first configure a remote IKE gateway gw1 with IP address 2.2.2.2. You specify the standard security level and use the preshared key juniper1. You then configure an XAuth client for the IKE gateway with the username beluga9 and the password 1234567. You also require CHAP authentication for the client.

WebUI

VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

```
Gateway Name: gw1
Security Level: Standard (select)
Remote Gateway Type:
      Static IP Address: (select), Address/Hostname: 2.2.2.2
Preshared Key: juniper1
Outgoing Interface: Untrust
```

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

```
XAuth Client: (select)
User Name: beluga9
```

Password: 1234567
Allowed Authentication Type: (select), CHAP Only

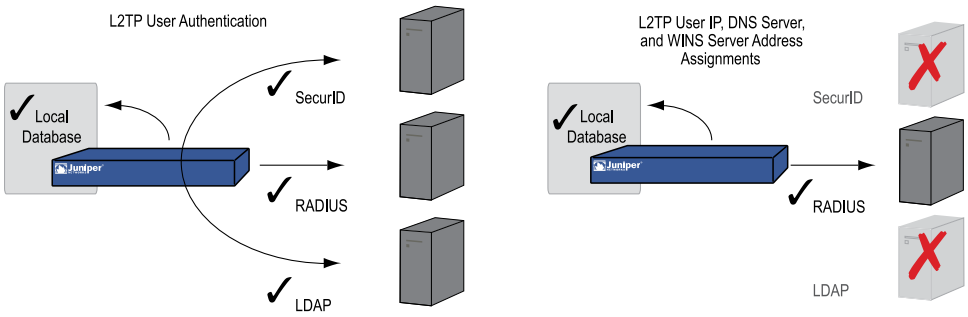
CLI

```
set ike gateway gw1 ip 2.2.2.2 Main outgoing-interface untrust preshare juniper1
sec-level standard
set ike gateway gw1 xauth client chap username beluga9 password 1234567
save
```

L2TP Users and User Groups

Layer 2 Tunneling Protocol (L2TP) provides a means for authenticating remote users and assigning IP, DNS server, and WINS server addresses. You can configure the security device to use either the local database or an external auth server to authenticate L2TP users. To make IP, DNS server, and WINS server address assignments, you can configure the security device to use either the local database or a RADIUS server (loaded with the RADIUS dictionary file—see "RADIUS Dictionary File" on page 1585). See Figure 415 on page 1656.

Figure 415: Authenticating Users with L2TP



You can even use a combination of auth servers, a different one for each of the two aspects of L2TP. For example, you might use a SecurID server to authenticate an L2TP user but make the address assignments from the local database. The following example illustrates the application of two auth servers to handle both components of L2TP. For other examples, along with a detailed examination of L2TP, see "Layer 2 Tunneling Protocol" on page 933.

NOTE: You can configure separate RADIUS servers for accounting and authentication for L2TP users. For more information about configuring a separate RADIUS accounting server for L2TP users, see "Configuring a Separate External Accounting Server" on page 1604.

Example: Local and External L2TP Auth Servers

In this example, you set up an external SecurID auth server to authenticate L2TP users, and you use the local database to assign L2TP users with IP, DNS server, and WINS server addresses.

The external SecurID auth server is securid1. It is nearly identical to the auth server configuration in "Example: SecurID Auth Server" on page 1599 except that the account type is L2TP. The SecurID auth server parameters are as follows:

- Name: securid1
- IP Address: 10.20.2.100
- Backup IP Address: 10.20.2.110
- Port: 15000
- Client Retries: 3
- Client Timeout: 10 seconds
- Idle Timeout: 60 minutes
- Account Type: L2TP
- Encryption: DES

The L2TP default settings are as follows:

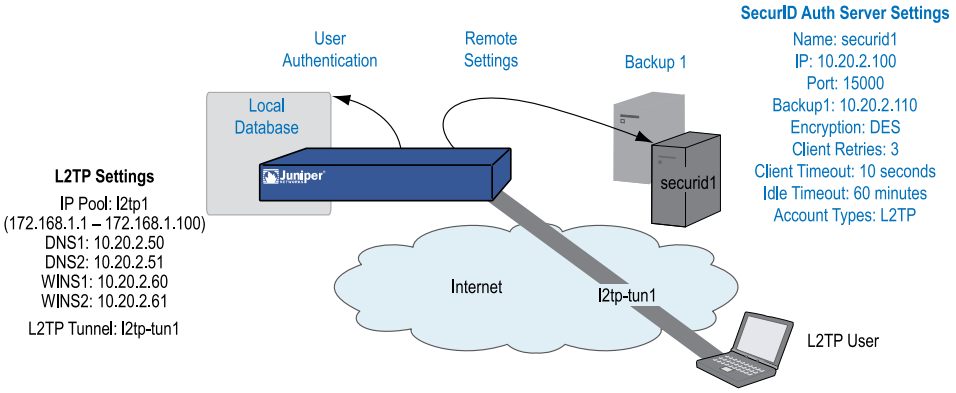
- IP Pool: I2tp1 (172.168.1.1 – 172.168.1.100)
- DNS Primary Server IP: 10.20.2.50
- DNS Secondary Server IP: 10.20.2.51
- PPP Authentication: CHAP
- WINS Primary Server IP: 10.20.2.60
- WINS Secondary Server IP: 10.20.2.61

After configuring the security device with the above settings, you create an L2TP tunnel named "l2tp-tun1" that references securid1 for authentication and uses the default settings for address assignments.

You must also set up the SecurID server as shown above and populate it with L2TP users. Figure 416 on page 1658 shows the L2TP settings, SecurID Auth server settings, and network setup.

NOTE: An L2TP-only configuration is not secure. To add security to an L2TP tunnel, we recommend that you combine it with an IPsec tunnel, which must be in transport mode, as shown in "Configuring L2TP-over-IPsec" on page 945.

Figure 416: Local and External L2TP Servers



WebUI

1. **Auth Server**

   Configuration > Auth > Auth Servers > New: Enter the following, then click **OK**:

   Name: securid1
   IP/Domain Name: 10.20.2.100
   Backup1: 10.20.2.110
   Timeout: 60
   Account Type: L2TP
   SecurID: (select)
   Client Retries: 3
   Client Timeout: 10 seconds
   Authentication Port: 15000
   Encryption Type: DES
   Use Duress: No

2. **IP Pool**

   Objects > IP Pools > New: Enter the following, then click **OK**:

   IP Pool Name: l2tp1
   Start IP: 172.168.1.1
   End IP: 172.168.1.100

3. **L2TP Default Settings**

   VPNs > L2TP > Default Settings: Enter the following, then click **Apply**:

   Default Authentication Server: Local
   IP Pool Name: l2tp1
   PPP Authentication: CHAP
   DNS Primary Server IP: 10.20.2.50
   DNS Secondary Server IP: 10.20.2.51
   WINS Primary Server IP: 10.20.2.60
   WINS Secondary Server IP: 10.20.2.61

4. **L2TP Tunnel**

   VPNs > L2TP > Tunnel > New: Enter the following, then click **OK**:

   Name: l2tp-tun1
   Use Custom Settings: (select)
   Authentication Server: securid1
   Query Remote Settings: (clear)
   Dialup User: (select), Allow Any

CLI

1. **Auth Server**

```
set auth-server securid1 type securid
set auth-server securid1 server-name 10.20.2.100
set auth-server securid1 backup1 10.20.2.110
set auth-server securid1 timeout 60
set auth-server securid1 account-type l2tp
set auth-server securid1 securid retries 3
set auth-server securid1 securid timeout 10
set auth-server securid1 securid auth-port 15000
set auth-server securid1 securid encr 1
set auth-server securid1 securid duress 0
```

2. **IP Pool**

```
set ippool l2tp1 172.168.1.1 172.168.1.100
```

3. **L2TP Default Settings**

```
set l2tp default auth server Local
set l2tp default ippool l2tp1
set l2tp default ppp-auth chap
set l2tp dns1 10.20.2.50
set l2tp dns1 10.20.2.51
set l2tp wins1 10.20.2.60
set l2tp wins2 10.20.2.61
```

4. **L2TP Tunnel**

```
set l2tp l2tp-tun1
set l2tp l2tp-tun1 auth server securid1
save
```

Chapter 51

Extensible Authentication for Wireless and Ethernet Interfaces

This chapter explains the options available for and examples of using Extensible Authentication Protocol (EAP) to provide authentication for Ethernet and wireless interfaces. It contains the following sections:

Overview

EAP is an authentication framework that supports multiple authentication methods. EAP typically runs directly over data link layers, such as Point-to-Point Protocol (PPP) or IEEE 802, without requiring Layer 3 addressing.

IEEE 802.1X works for port-based access control, and IKEv2 uses it as an option for authentication. EAP functions in a security device configured in transparent or route (with or without Network Address Translation enabled) mode. NetScreen Redundancy Protocol (NSRP) supports EAP in networks with high availability. Log messages and SNMP support are also available.

802.1X support is available for all platforms. In addition, 802.1X for IPv6 supports NetScreen Redundancy Protocol (NSRP).

EAP functions as the authentication portion of PPP, which operates at Layer 2. EAP authenticates a supplicant, or client, after the supplicant sends proper credentials and the authentication server, usually a RADIUS server, defines the user-level permissions. When you use EAP, all authentication information passes through the security device (known as a pass-through method of EAP authentication). All user information is stored on the authentication server.

If you use a RADIUS server for authentication that supports vendor-specific attributes (), you can use the zone-verification feature, which verifies the zones a client is a member of.

Supported EAP Types

The EAP types described in Table 116 on page 1662 are supported.

Table 116: EAP Types

Type	Description
EAP-TLS (Transport Layer Security)	The most common EAP derivative and is supported by most RADIUS servers. EAP-TLS uses certificates for user and server authentication and for dynamic session key generation.
EAP-TTLS (Tunneled Transport Layer Security)	Requires only a server-side certificate and a valid username and password for authentication. Steel-Belted RADIUS supports TTLS.
EAP-PEAP (Protected EAP)	Designed to compensate for the lack of features in EAP-TLS and reduce management complexity. It requires only server-side certificates and a valid username and password. It provides support for key exchange, session resumption, fragmentation, and reassembly. Steel-Belted RADIUS and Microsoft IAS support Protected EAP.
EAP-MD5 (Message Digest Algorithm 5)	Algorithm that uses a challenge and response process to verify MD5 hashes.

Enabling and Disabling 802.1X Authentication

By default, 802.X authentication is disabled. You can enable 802.1X authentication for Ethernet and wireless interfaces using the WebUI or CLI. To enable 802.1X authentication on an Ethernet interface, you modify the interface's configuration. 802.1X on a wireless interface is automatically enabled after you create and configure an SSID and then bind it to the wireless interface.

Ethernet Interfaces

Use one of the following procedures to enable 802.1X on the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (ethernet1) > 802.1X: Click **Enable**.

CLI

To enable 802.1X on the ethernet1 interface, enter the following command:

```
set interface ethernet1 dot1x
```

To disable 802.1X on the ethernet1 interface, enter the following command:

```
unset interface ethernet1 dot1x
```

Wireless Interfaces

Use one of the following procedures to enable 802.1X on a wireless interface. The following procedures create an SSID named **hr**, using WPA as the authentication method and TKIP as the encryption method. The authentication server is a predefined RADIUS server named **radius1**. The SSID is then bound to wireless interface 0/1.

WebUI

Wireless > SSID > Click **New**. Enter the following, then click **OK**.

SSID Name: hr
WPA Based Authentication and Encryption Methods: Select **WPA**, **TKIP**.
Auth Server: Select **radius1.**
Wireless Interface Binding: Select **wireless0/1**.

Depending on the security device, you need to activate the changes you configured by clicking the Activate Changes button at the top of the page or by selecting Wireless > Activate Changes.

To disable 802.1X on a wireless interface, select **none** from the Wireless Interface Binding list.

CLI

To enable 802.1X on a wireless interface, you must create and configure an SSID and then bind it to the wireless interface.

```
set ssid name hr
set ssid hr authentication wpa encryption tkip auth-server radius1
set ssid hr interface wireless0/1
```

To disable 802.1X on the wireless interface, enter one of the following command, which unbinds the SSID to the interface:

```
unset ssid hr interface
```

For security devices with two radio transceivers, you can also enter the following command, which unbinds the interface to the radio:

```
unset interface wireless0/1 wlan
```

You can also disable 802.1X on a wireless interface by changing the authentication method of an SSID that does not require use of an authentication server. Disabling the wireless interface with the **set interface** *wireless_interface* **shutdown** command also disables 802.1X.

Configuring 802.1X Settings

For Ethernet interfaces, you can optionally configure 802.1X settings. For wireless interfaces, these settings cannot be modified, and the default values are used, as listed in Table 117 on page 1664.

Table 117: 802.1X Settings

Option	Default Value	Alternative Values
Port control	auto	force-unauthorized
Control mode	virtual	interface
Maximum user	16 (Ethernet); 128 (wireless)	1-256
Reauthentication period	3600	0 -86400
Retransmission	enable	disable
Retransmission count	3	1-16
Retransmission period	3	1-120
Silent period	5	0-3600

Configuring 802.1X Port Control

You can configure how an Ethernet interface deals with 802.1X authentication attempts. By default, the port state is **auto**, which allows 802.1X authentication to proceed normally. You can configure the Ethernet interface so that it blocks all traffic and ignores all attempts by clients to authenticate by using the **force-unauthorized** option. You can also configure the interface to successfully authenticate all attempts by clients (also known as force-authorized state) by disabling 802.1X for the interface.

In the following examples, you set the port-control state to force-unauthorized for the ethernet1 interface, which specifies that the interface blocks all traffic and ignores client authentication attempts.

WebUI

Network > Interfaces > List > Edit (for Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Port Control: Select **Force-unauthorized**.

CLI

```
set interface ethernet1 dot1x port-control force-unauthorized
```

Configuring 802.1X Control Mode

You can specify whether MAC address-based authentication is performed on devices connected to the interface by specifying one of the following modes:

- Interface: MAC addresses of devices connected to the interface are not authenticated. Use this option only if one trusted device is connected to the interface.
- Virtual: MAC addresses of devices connected to the interface are authenticated. Packets from devices with unauthorized MAC addresses are dropped. This mode is the default for an interface. Wireless interfaces use only virtual mode.

In the following examples, you set the control mode to interface for the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Control Mode: Interface.

CLI

```
set interface ethernet1 dot1x control-mode interface
```

Setting the Maximum Number of Simultaneous Users

When an interface is in virtual control mode, the security device allows up to the configured number of simultaneous users. By default, the security device accepts 16 simultaneous users for Ethernet interfaces or 128 users for wireless interfaces. The valid value range is 1 through 256. If you have configured the control mode to interface mode, you cannot configure the maximum number of simultaneous users.

In the following examples, you set the maximum number of simultaneous users for the ethernet1 interface to 24.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Maximum User : 24.

CLI

```
set interface ethernet1 dot1x max-user 24
```

Configuring the Reauthentication Period

By default, reauthentication of 802.1X supplicants (clients) is enabled on the security device. The security device attempts to reauthenticate clients after 3600 seconds (1 hour).

For Ethernet interfaces, you can configure the reauthentication period from 0 through 86400 seconds (24 hours). To disable the reauthentication period, set the period to 0. For wireless interfaces, you cannot change the reauthentication period from its default value. If a RADIUS server provides a reauthentication period other than the default value, the security device can use the RADIUS-assigned value.

In the following examples, you set the reauthentication period to 7200 seconds (2 hours) for the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Re-Authentication Period: 7200

CLI

```
set interface ethernet1 dot1x reauth-period 7200
```

To set the reauthentication period to its default value, use the **unset interface** *interface_name* **dot1x reauth-period** command.

Enabling EAP Retransmissions

You can enable the retransmission of EAP requests to a client if it does not respond. By default, retransmission is enabled. Optionally, you can also configure the maximum number of EAP requests that are retransmitted and the time that elapses between retransmissions. If the maximum number of retransmissions is reached, the client's authenticated session is terminated, and authentication fails.

In the following examples, you enable the retransmission of EAP requests for the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Select the 8021.X Enable check box, then click **Apply**.

CLI

```
set interface ethernet1 dot1x retry
```

Configuring EAP Retransmission Count

By default, the security device sends up to three EAP requests. You can configure the number of EAP requests from 1 through 16.

In the following examples, you set the number of EAP request transmissions to 8.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Re-Transmission Count: 8

CLI

To configure the number of retransmit packets sent, use the following command:

```
set interface ethernet1 dot1x retry count 8
```

Configuring EAP Retransmission Period

By default, period between EAP retransmissions is 3 seconds. You can configure a period from 1 through 120 seconds.

In the following examples, you set the period between EAP retransmissions to 5.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Re-Transmission Period: 5

CLI

```
set interface ethernet1 dot1x retry period 5
```

Configuring the Silent (Quiet) Period

The silent (quiet) period is the amount of time the security device remains silent after authentication has failed. During the silent period, the security device does not initiate or respond to any client authentication requests.

By default, when authentication fails, the security device is silent for 5 seconds, and the authentication retry count resets to zero (0).

The silent period is a value from 0 through 3600 seconds (1 hour). The 802.1X authentication state remains unauthorized after the retry fails if you specify a silent period of zero (0).

In the following examples, you set the silent period to 30 seconds for the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Enter the following, then click **Apply**.

Silent Period: 30

CLI

```
set interface ethernet1 dot1x silent-period 30
```

Configuring Authentication Server Options

If you have configured authentication servers in your network and defined them, you can specify one of these servers as the authentication server for an interface. You can also set the following authentication server options:

- Account type (802.1X clients)
- Zone verification

Specifying an Authentication Server

You can use a predefined server as the authentication server for a specific interface.

Ethernet Interfaces

For Ethernet interfaces, you specify an authentication server by modifying the interface configuration. In the following examples, you specify the existing **radius1** server as the authentication server for the ethernet1 interface.

WebUI

Network > Interfaces > List > Edit (Ethernet1) > 802.1X: Select the authentication server from the Server Name list, then click **Apply**.

CLI

```
set interface ethernet1 dot1x auth-server radius1
```

Wireless Interfaces

For wireless interfaces, you specify an authentication server by modifying the SSID configuration. In the following examples, you modify the SSID named **hr** and specify the existing **radius1** server as the authentication server for the wireless0/1 interface.

Wireless > SSID > Edit (for hr SSID): Click **New**. Depending on the security device, select one of the following, then click **OK**.

WEP Based Authentication and Encryption Methods: WEP Encryption: Select **Open**; **WEP Encryption**; and **radius1** from the Auth Server list.
802.1X Based Authentication and Encryption Methods: Select **802.1X** and then select **radius1** from the Auth Server list.

CLI

```
set ssid hr authentication wpa encryption auto auth-server radius1
```

To use the WebUI to configure authentication server information, navigate to the Auth Servers page:

Configuration > Auth > Auth Servers: Enter or select the applicable option value, then click **Apply**.

Setting the Account Type

When defining an authentication server or modifying it, you can specify the authentication server to accept 802.1X clients.

WebUI

To specify that the authentication server accept 802.1X clients, use the following procedure:

Configuration > Auth > Auth Servers > New (or Edit for existing server): Enter all relevant information; in the Account Type area, select the 802.1X check box; click **Apply**.

CLI

To specify that the existing server named **radius1** accept 802.1X clients, enter the following command:

```
set auth-server radius1 account_type 802.1x
```

Enabling Zone Verification

If your RADIUS server supports vendor-specific attribute (VSA) enhancement, you can enable zone verification, which verifies the zones the user is a member of and the zone configured on the port. Authentication is allowed only if the zone configured on the port is a zone that a user is a member of.

In your dictionary file, add an attribute name of Zone_Verification as a string attribute type. The vendor ID is 3224, and the attribute number is 10.

WebUI

To enable zone verification, use the following procedure:

Configuration > Auth > Auth Servers > New (or Edit for existing server): Enter all relevant information; select RADIUS; select the Enabled check box for Zone Verification; click **Apply**.

CLI

To enable zone verification for the RADIUS server named **radius1**, enter the following command:

```
set auth-server radius1 radius zone-verification
```

Viewing 802.1X Information

You can view detailed information about 802.1X configuration in the CLI. Not all 802.1X information can be viewed using the WebUI.

Viewing 802.1X Global Configuration Information

Enter the following command to view the global 802.1X configuration information:

```
get dot1x
```

The command shows the following information for each Ethernet and wireless interface:

- 802.1X status: enabled or disabled
- Mode: virtual or interface
- Number of users out of the maximum number of users allowed
- Number of seconds until reauthentication is required
- Port-control mode status

The following is sample output for the **get dot1x** command:

```
------------------------------------------------------------------------------
Name     IEEE802.1x   Mode        User    re-auth      Status
  Ethernet1      Enabled       virtual      1/64     3600s       Auto
  Ethernet2      Disabled     virtual       0/64     3600s        Auto
  Ethernet3      Enabled       interface     --      1200s        F-U
  Ethernet3.1     Enabled       virtual      0/64     3600s       Auto
  Ethernet4      Enabled       virtual     0/16     3600s       Auto
------------------------------------------------------------------------------
```

Viewing 802.1X Information for an Interface

Enter the following command to view 802.1X configuration and user information for a specific interface:

```
get interface interface_name dot1x
```

The following is sample output:

```
IEEE 802.1x enabled
port-control: auto, mode: virtual
user 1/max 64  auth-server: test-radius
reauth enable   period 1200s
silent enable   period 300s
to-supplicant retry enable   count 3   period 10s
------------------------------------------------------------------
User 0003e40220b1, session id 1, authorized
Total 1 user shown
  ------------------------------------------------------------------
```

Viewing 802.1X Statistics

Use the WebUI or CLI to get 802.1X statistics for a specific interface.

WebUI

Network > 802.1X > Statistic: Select the interface from the list at the top of the page.

CLI

Enter the following command to view 802.1X statistics for the ethernet0/2 interface:

```
get interface ethernet0/2 dot1x statistics
```

The following is sample output:

```
Interface Ethernet0/2:
------------------------------------------------------------------------------
Interface ethernet1 802.1x statistics:
in eapol               0 | out eapol              0 | in start           0
in logoff              0 | in resp/id             0 | in resp            0
out req/id             0 | out req                0 | in invalid
   0
in len error           0 |
Interface ethernet1 802.1x diagnostics:
while connecting:
enters                 0 | eap logoffs            0 |
while authenticating:
enters                 0 | auth success           0 | auth timeouts      0
auth fail              0 | auth reauth            0 | auth start          0
auth logoff            0 |
```

Viewing 802.1X Session Statistics

Enter the following command to view 802.1X session statistics:

```
get dot1x session
```

The following is sample output:

```
Alloc 2/max 1024, alloc failed 0
Id 1/ vsys 0, flag 00000000, re-auth 3105s, ethernet1, 0003e40220c2, authorized
```

```
Id 2/ vsys 0, flag 00000000, re-auth 430s, ethernet3.1, 0003e40220b1, fail-silent
Total 2 session shown
```

Viewing 802.1X Session Details

Enter the following command to view detailed information for a specific 802.1X session:

```
get dot1x session id
```

The following is sample output:

```
Id 1, flag 00000000, vsys id 0(Root)
Interface ethernet1(vsd 0), supp-mac 0003e40220c2, status authorized
Re-auth timeout 3105s, type eap-md5
  As radius_test, zone-verification on
  Retry 0, as retry 0
  ------------------------------------------------------------------------
  statistics:
in octets      0 | out octets              0 | in frames         0
out frames      0
  ------------------------------------------------------------------------
```

Configuration Examples

This section contains the following three examples:

- “Configuring the Security Device with a Directly Connected Client and RADIUS Server” on page 1672
- “Configuring a Security Device with a Hub Between a Client and the Security Device” on page 1673
- “Configuring the Authentication Server with a Wireless Interface” on page 1674

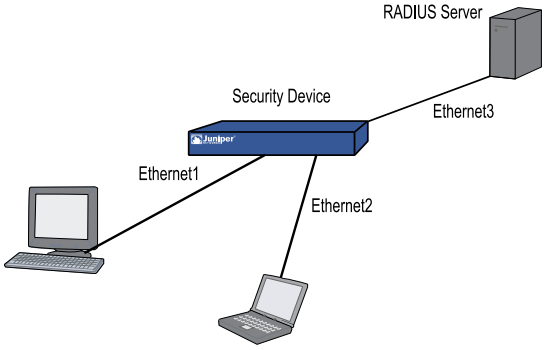
Configuring the Security Device with a Directly Connected Client and RADIUS Server

This network scenario, as shown in Figure 417 on page 1673, has two clients directly connected to the security device with the following parameters:

- Client directly connected to Ethernet1 interface
- Client directly connected to Ethernet2 interface
- Ethernet3 interface bound to Trust zone with an IP address of 10.1.40.3/24
- RADIUS server named radius1 (10.1.1.200) connected to Ethernet3 interface to authenticate users with 802.1X, using port 1812 as the authentication port and secret of mysecret

Because the two clients directly connected are the only devices connected to the Ethernet1 and Ethernet2 interfaces, the control-mode is configured to **interface**.

Figure 417: Security Device with a Directly Connected Client and RADIUS Server




```
set interface ethernet1 dot1x
set interface ethernet2 dot1x
set interface ethernet1 dot1x control-mode interface
set interface ethernet2 dot1x control-mode interface

set interface ethernet3 zone trust
set interface ethernet3 ip 10.1.1.10/24

set auth-server radius1 account-type 802.1x
set auth-server radius1 type radius
set auth-server radius1 radius port 1812
set auth-server radius1 radius secret mysecret
set auth-server radius1 server-name 10.1.1.200

set interface ethernet1 dot1x auth-server radius1
set interface ethernet2 dot1x auth-server radius1
```

Configuring a Security Device with a Hub Between a Client and the Security Device

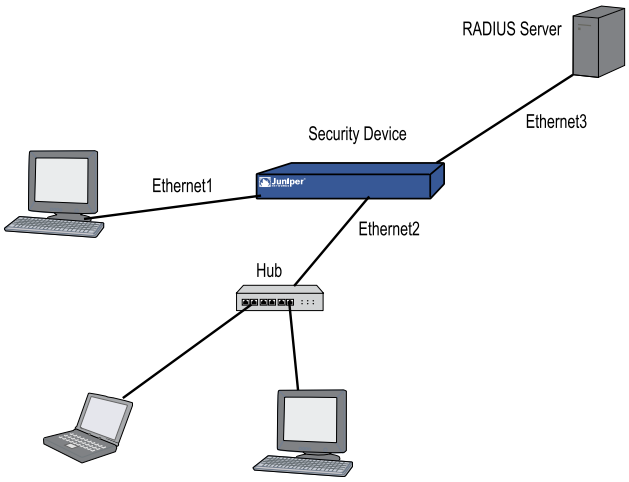
The following scenario, as shown in Figure 418 on page 1674, uses a hub with attached clients connected to the security device and a client directly connected to the security device.

NOTE: 802.1X functionality is not supported for a switch between the security device and clients. If you have a switch connected to the security device, we recommend disabling 802.1X on the interface to which the switch is connected.

This scenario uses the following parameters:

- Hub connected to Ethernet2 interface (control-mode of **virtual**)
- Client directly connected to Ethernet1 interface (control-mode of **interface**)
- Ethernet3 interface bound to Trust zone with an IP address of 10.1.40.3/24
- RADIUS server named radius1 (10.1.1.200) connected to Ethernet3 interface to authenticate users with 802.1X, using port 1812 as the authentication port and secret of mysecret

Figure 418: Security Device with a Hub Between a Client and the Security Device




```
set interface ethernet1 dot1x
set interface ethernet2 dot1x
set interface ethernet1 dot1x control-mode interface
set interface ethernet2 dot1x control-mode virtual

set interface ethernet3 zone trust
set interface ethernet3 ip 10.1.1.10/24

set auth-server radius1 account-type 802.1x
set auth-server radius1 type radius
set auth-server radius1 radius port 1812
set auth-server radius1 radius secret mysecret
set auth-server radius1 server-name 10.1.1.200

set interface ethernet1 dot1x auth-server radius1
set interface ethernet2 dot1x auth-server radius1
```

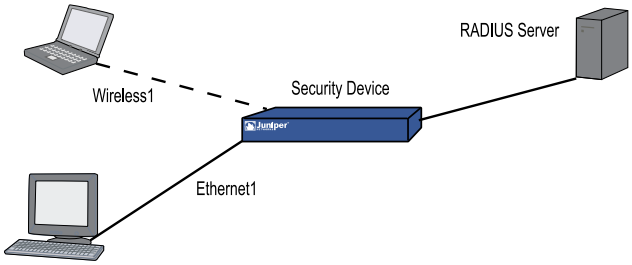
Configuring the Authentication Server with a Wireless Interface

The following scenario, as shown in Figure 419 on page 1675, has a security device with a wireless interface serving wireless clients and a client directly connected to the security device with the following parameters:

- Wireless clients connected to the wireless interface
- Client directly connected to Ethernet1 interface (control-mode of **interface**)
- Ethernet3 interface bound to Trust zone with an IP address of 10.1.40.3/24
- RADIUS server named radius1 (10.1.1.200) connected to Ethernet3 interface to authenticate users with 802.1X, using port 1812 as the authentication port and secret of mysecret

- SSID named engineering, using WPA authentication, either AES or TKIP encryption, specifying radius1 as the authentication server, and bound to wireless interface 1

Figure 419: Configuring an Authentication Server with a Wireless Interface




```
set interface ethernet1 dot1x
set interface ethernet1 dot1x control-mode interface

set interface ethernet3 zone trust
set interface ethernet3 ip 10.1.1.10/24
set auth-server radius1 account-type 802.1x
set auth-server radius1 type radius
set auth-server radius1 radius port 1812
set auth-server radius1 radius secret mysecret
set auth-server radius1 server-name 10.1.1.200

set interface ethernet1 dot1x auth-server radius1

set ssid name engineering
set ssid engineering authentication wpa encryption auto auth-server radius1
set ssid engineering interface wireless0/1
```

Part 10
Virtual Systems

Virtual Systems describes virtual systems, dedicated and shared interfaces, and VLAN-based and IP-based traffic classification.

This guide contains the following chapters:

Chapter 52
Virtual Systems

This chapter discusses virtual systems (vsys), objects, and administrative tasks. It contains the following sections:

Overview

You can logically partition a single Juniper Networks security device into multiple virtual systems (vsys) to provide multi-tenant services. Each vsys is a unique security domain and can have its own administrators (called *virtual system administrators* or *vsys admins*) who can individualize their security domain by setting their own address books, user lists, custom services, virtual private networks (VPNs), and policies. Only a root-level admin, however, can set firewall security options, create vsys admins, and define interfaces and subinterfaces.

NOTE: Refer to the Juniper Networks marketing literature to see which platforms support the virtual system feature.

For more information about the various levels of administration that ScreenOS supports, see *"Levels of Administration" on page 345*.

Juniper Networks virtual systems support two kinds of traffic classifications: virtual local area network (VLAN)-based and Internet Protocol(IP)-based, both of which can function separately or concurrently.

Table 118 on page 1680 shows the interfaces a vsys can support for the Untrust and Trust security zones:

Table 118: Virtual System Support

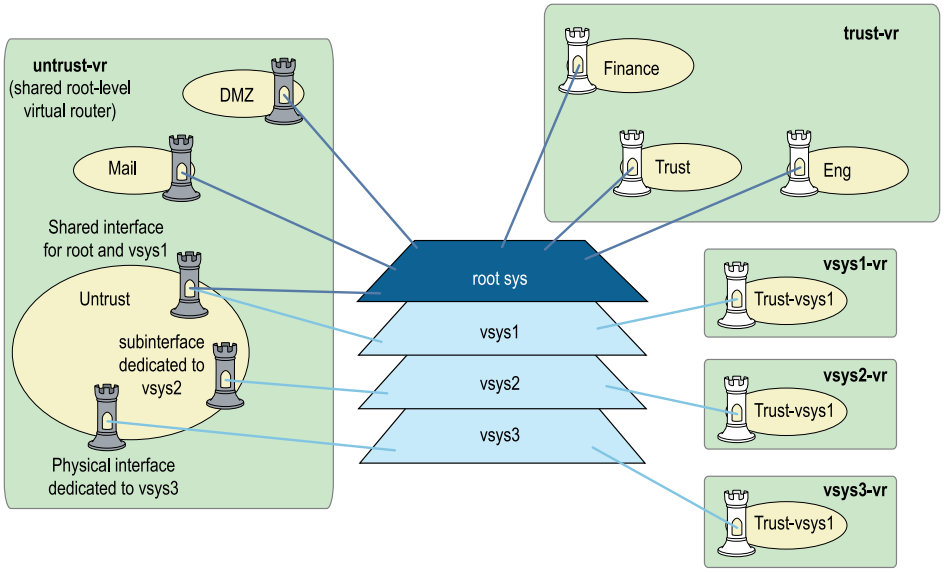
Untrust Zone Interface Types	Trust Zone Interface Types
Dedicated physical interface	Dedicated physical interface
Subinterface (with VLAN tagging as a means for trunking inbound and outbound traffic)	Subinterface (with VLAN tagging)
Shared interface (physical, subinterface, redundant interface, aggregate interface) with root system	Shared physical interface with root system (and IP-based traffic classification)

NOTE: For information about VLAN tagging and trunking concepts, see "VLAN-Based Traffic Classification" on page 1723.

For information about IP-based traffic classification, see "IP-Based Traffic Classification" on page 1757.

Figure 420 on page 1680 shows how you can bind one, two, or all three of the above interface types to a security zone concurrently. You can also bind multiple interfaces of each type to a zone.

Figure 420: Interface and Zone Bindings with Vsys



Vsys Objects

The root admin or root-level read-write admin must complete the following tasks to create a vsys object:

- Define a vsys.
- (Optional) Define one or more vsys admins.

NOTE: A root-level admin can define, per vsys, one vsys admin with read-write privileges and one vsys admin with read-only privileges.

- Select the virtual router (VR) that you want the vsys to use for its Trust-vsysname zone, Untrust-Tun-vsysname zone, and Global-vsysname zone.

After creating a vsys object, as the root-level admin, you need to perform other configurations to make it functional. You must configure subinterfaces or interfaces for the vsys, and possibly shared VRs and shared security zones. Subsequent configurations depend on whether the vsys is intended to support VLAN-based or IP-based traffic classifications, or a combination of both. After completing these configurations, you can then exit the vsys and allow a vsys admin, if one is defined, to log in and begin configuring addresses, users, services, VPNs, routes, and policies.

Creating a Virtual System Object and Admin

In this example, as a root-level admin, you create three vsys objects: vsys1, vsys2, and vsys3. For vsys1, you create vsys admin Alice with password wIEaS1v1. For vsys2, you create vsys admin Bob with password pjF56Ms2. For vsys3, you do not define a vsys admin. Instead, you accept the admin definition that the security device automatically generates. In the case of vsys3, the security device creates the admin "vsys_vsys3" with password "vsys_vsys3."

NOTE: Only a root-level admin can create a vsys admin's profile (username and password). Because the security device uses usernames to determine the vsys to which a user belongs, vsys admins cannot change their usernames. However, vsys admins can (and should) change their passwords.
Virtual System names, admin names, and passwords are case-sensitive. For Example, "Vsys abc" is different from "vsys ABC."

For vsys1 and vsys2, you use the default VR. For vsys3, you choose the sharable root-level untrust-vr.

After you create a vsys through the WebUI, you remain at the root level. Entering the newly created vsys requires a separate step:

Vsys > Configure > Click **Enter** (for the vsys you want to enter).

The WebUI pages of the vsys you have entered appear, with the name of the vsys above the central display area—Vsys:Name.

When you create a vsys through the CLI, you immediately enter the system that you have just created. (To enter an existing vsys from the root level, use the **enter vsys** *name_str* command.) When you enter a vsys, the CLI command prompt changes to include the name of the system in which you are now issuing commands.

WebUI

1. **Vsys1**

   Vsys > Configure > New: Enter the following, then click **OK**:

   Vsys Name: vsys1
   Vsys Admin Name: Alice
   Vsys Admin New Password: wlEaS1v1
   Confirm New Password: wlEaS1v1
   Virtual Router:
   Create a default virtual router: (select)

2. **Vsys2**

   Vsys > Configure > New: Enter the following, then click **OK**:

   Vsys Name: vsys2
   Vsys Admin Name: Bob
   Vsys Admin New Password: pjF56Ms2
   Confirm New Password: pjF56Ms2
   Virtual Router:
   Create a default virtual router: (select)

3. **Vsys3**

   Vsys > Configure > New: Enter the following, then click **OK**:

   Vsys Name: vsys3
   Virtual Router:
   Select an existing virtual router: (select) untrust-vr

CLI

1. **Vsys1**

```
device-> set vsys vsys1
device(vsys1)-> set admin name Alice
device(vsys1)-> set admin password wlEaS1v1
device(vsys1)-> save
device(vsys1)-> exit
```

NOTE: After issuing any commands, you must issue a save command before you issue an exit command in order for the security device to save your changes.

2. **Vsys2**

```
device-> set vsys vsys2
device(vsys2)-> set admin name Bob
device(vsys2)-> set admin password pjF56Ms2
device(vsys2)-> save
```

```
device(vsys2)-> exit
```

3. **Vsys3**

```
device-> set vsys vsys3 vrouter share untrust-vr
device(vsys3)-> save
```

Setting a Default Virtual Router for a Virtual System

When a root-level admin creates a vsys object, the vsys automatically has the following VRs available for its use:

- All shared root-level VRs, such as the untrust-vr

  In the same way that a vsys and the root system share the Untrust zone, they also share the untrust-vr, and any other VRs defined at the root level as sharable.
- Its own VR
- By default, a vsys-level VR is named *vsysname-vr*. You can also customize the name to make it more meaningful. This is a vsys-specific VR that, by default, maintains the routing table for the Trust-*vsysname* zone. All vsys-level VRs are nonsharable.

You can select any shared VR or the vsys-level virtual router as the default VR router for a vsys. To change the default VR, enter a vsys and use the following CLI command:
set vrouter *name* **default-vrouter**

As a root-level admin, if you want all of the vsys zones to be in the untrust-vr routing domain—for example, if all the interfaces bound to the Trust *-vsysname* zone are in route mode—you can dispense with the *vsysname*-VR by changing the vsys-level security zone bindings from the *vsysname*-vr to the untrust-vr. For more information about virtual routers, see *"Routing" on page 1235*.

NOTE: This release of ScreenOS supports user-defined VR within a vsys.

Binding Zones to a Shared Virtual Router

Each (vsys) is a unique security domain and can share security zones with the root system and have its own security zones. When a root-level admin creates a vsys object, the following zones are automatically inherited or created:

- All shared zones (inherited from the root system)
- Shared Null zone (inherited from the root system)
- Trust-*vsys -name* zone
- Untrust-Tun-*vsys-name* zone
- Global-*vsys-name* zone

NOTE: For information about each of these zone types, see *"Zones" on page 43*.

Each vsys can also support extra user-defined security zones. You can bind these zones to any shared VR defined at the root level or to the VR dedicated to that vsys. To create a security zone for a vsys named vsys1:

WebUI

Vsys > Enter (for vsys1)

Network > Zones > New: Enter the following, then click **OK**:

Zone Name: (type a name for the zone)
Virtual Router Name: (select a VR from the drop-down list)
Zone Type: Layer 3

CLI

```
device-> enter vsys vsys1
device(vsys1)-> set zone name name_str
device(vsys1)-> set zone vrouter vrouter
device(vsys1)-> save
```

The maximum number of security zones that a vsys or the root system can contain is limited only by the number of security zones available at the device level. It is possible for a single vsys to consume all available security zones if the root admin or a root-level read-write admin assigns all of the zones to that particular vsys. Conversely, if all vsys share root-level security zones and do not make use of any user-defined vsys-level zones, then all security zones are available for root-level use.

NOTE: The total number of user-definable (or custom) security zones available at the device level is the sum of the number of root-level custom zones—as defined by one or more zone license keys—and the number of custom zones permitted by the vsys license key.

Defining Identical Names for Zones Across Vsys

In previous releases of ScreenOS, names you defined for the zones in a vsys had to be unique regardless of whether the zones resided in the same or in different virtual systems. With this release of ScreenOS, you can name zones within a vsys without regard to zone names used in other vsys. In other words, the security device allows you to create zones with identical names, provided these zones are defined in a different vsys and not within the same vsys

NOTE: The name of a shared zone inherited from the root should always remain unique from that of the other zones created in the vsys.

When you create a zone, the security device checks to make sure no zone with the same name is already present within the current vsys. If the device finds a preexisting zone with that name, it returns an error message and fails to create the zone.

The **get zone *zone*** command displays the details of the zones with identical zone names across all vsys. Only in root vsys, you can view the list of zones that share the specified zone name.

For example, you can use the get zone internal command to view the list of zones across all vsys that share the name internal

The following is the sample output for get zone internal:

device> **get zone internal**

```
Zone name: internal, id: 1002, type: Security(L3), vsys: Root,
vrouter:trust-vr
Intra-zone block: Off, attrib: Non-shared, flag:0x6208
TCP non SYN send reset: On
IP/TCP reassembly for ALG on traffic from/to this zone: No
Asymmetric vpn: Disabled
Policy Configurable: Yes
PBR policy: None
Interfaces bound:2. Designated ifp is ethernet0/0
interface ethernet0/1(0x904f3b4)
interface ethernet0/0(0x8f31b34)
DHCP relay enabled

Zone name: internal, id: 1004, type: Security(L3), vsys: vsys1,
vrouter:vsys1-vr
Intra-zone block: Off, attrib: Non-shared, flag:0x6208
TCP non SYN send reset: On
IP/TCP reassembly for ALG on traffic from/to this zone: No
Asymmetric vpn: Disabled
Policy Configurable: Yes
PBR policy: None
Interfaces bound:2. Designated ifp is ethernet0/3
interface ethernet0/3(0x904f3a4)
interface ethernet0/4(0x8f31b24)
DHCP relay enabled
```

Logging In as a Virtual System Admin

Vsys admins enter their vsys directly, unlike root-level admin, who enter their vsys from the root level. When a vsys admin exits a vsys, the connection is immediately severed; however, when root-level admin exit a vsys, they exit to the root system.

The following example shows how you log into a vsys as a vsys admin, change your password, and log out.

In this example, you, as a vsys admin, log into vsys1 by entering your assigned login name **jsmith** and password **Pd50iH10**. You change your password to **I6Dls13guh** and then log out.

NOTE: Vsys admins cannot change their login names (usernames) because the security device uses those names, which must be unique among all vsys admins, to route the login connection to the appropriate vsys.

WebUI

1. **Logging In**

   In the URL field in your browser, enter the Untrust zone interface IP address for vsys1.

   When the Network Password dialog box appears, enter the following, then click **OK**:

   ```
   User Name: jsmith
   Password: Pd50iH10
   ```

2. **Changing Your Password**

   Configuration > Admin > Administrators: Enter the following, then click **OK**:

   ```
   Vsys Admin Old Password: Pd50iH10
   Vsys Admin New Password: I6DIs13guh
   Confirm New Password: I6DIs13guh
   ```

3. **Logging Out**

   Click **Logout**, located at the bottom of the menu column.

CLI

1. **Logging In**

   From a Secure Command Shell (SCS), Telnet, or HyperTerminal session command-line prompt, enter the Untrust zone interface IP address for vsys1.

   Log in with the following username and password:

   - User name: jsmith
   - Password: Pd50iH10

2. **Changing Your Password**

   ```
   set admin password I6DIs13guh
   save
   ```

3. **Logging Out**

   ```
   exit
   ```

Virtual System Profiles

A root-level user (the admin for the security device) can enable or disable session and resource limits on a per-vsys basis. If you configure a session limit for a particular vsys and the vsys reaches or exceeds its session limits, the security device enforces the session limit and begins dropping packets for that vsys. In the case of oversubscription, where the total number of vsys sessions is greater than the overall number of system sessions, you can reserve a specified number of sessions for a particular vsys. The security device tracks packets that are dropped as a result of a session limit.

NOTE: To use virtual systems, you must install a vsys key and then enable this feature. It is disabled by default.

ScreenOS provides two ways to configure resource limits for a vsys:

- Profile assignment
- Command overrides

Per-vsys resources for which you can define maximum and reserve limits include the following items:

- Dynamic IP (DIP) addresses
- Mapped IP (MIP) addresses
- User-defined services and groups
- Policies and multicast policies
- Sessions
- Zone address-book entries and groups, which are per-zone per-vsys limits
- User-defined security zones

WARNING: Check with your administrator before you assign a DIP ID to a vsys. Duplicate IDs used on the same device can cause dropped or misrouted traffic. The device will not check for or prevent duplicate DIP IDs, nor will it send a notification if such duplicates exist.

NOTE: ScreenOS enforces zone address-book and zone address-group limits for the shared zone, a zone that contains address and address groups from all vsys . When viewing addresses or address groups of a shared zone from a vsys, only those addresses and address groups configured in that vsys are listed. The resources used for addresses and address groups in a shared zone are charged against the root system in which the shared zone was created.

You cannot reserve addresses or address groups in shared zones.

Vsys profiles can also contain CPU weights, which allow you to allocate a certain percentage of CPU processing time for a particular vsys. See "Configuring CPU Weight" on page 1697 for more information.

Virtual System Session Counters

When the security device creates a new session for a particular vsys, the session counter for that vsys increments. When the session ends, the counter decrements.

The security device counts all sessions, active and inactive, held by the vsys at any time.

Virtual System Session Information

The security device records session statistics for each vsys. The security device admin (root admin) can view all of the collected statistics and session information for all virtual systems. A vsys admin can view-only the sessions and statistics pertaining to that admin's vsys domain.

In previous ScreenOS releases, the root admin could clear vsys-specific sessions from the security device only by clearing all sessions at the root. Beginning with the 6.2.0 release of ScreenOS, the root admin can use the **clear session** command to clear only the sessions for a specific vsys.

As root admin, you can clear vsys-specific sessions in the root only when you specify the vsys name or vsys ID as options in the **clear session** command. If you do not specify vsys as the option, then only sessions pertaining to the root are deleted. If you include the option **all**, all root and vsys sessions are deleted.

Use the following CLI commands to clear vsys-specific sessions:

CLI

1. **Clearing Session from Root**

```
clear session vsys-name vsys-name vsys-id id_num
save
```

2. **Clearing Session from Local Vsys**

```
clear session [ src-ip ip_addr ] [ dst-ip ip_addr ] [ src-mac mac_addr ] [ dst-mac
  mac_addr ] [ src-port port_num ] [ dst-port port_num ] [ protocol number ] [
  vsd-id id_num ]
save
```

NOTE: Users can use the **clear session** command to delete sessions only from their own vsys and not different vsys.

Behavior in High-Availability Pairs

When two devices configured with NetScreen Redundancy Protocol (NSRP) are in Active/Active mode and two sessions are simultaneously created, the result could mean that a vsys might have one session more than the configured limit.

For more information about NSRP or Active/Active mode, see *"High Availability" on page 1763*.

Creating a Vsys Profile

A *vsys profile* is a holder for the maximum limits and, in case of overload, specific limits and session-only alarm thresholds that you want ScreenOS to impose on a particular vsys or group of vsys. You can design tiered limits for services that fit the needs of your vsys clients. For example, you can set up different classes of service, such as gold, silver, and bronze, and assign each one different resource maximums.

Two default profiles exist:

- VsysDefaultProfile

  By default, when you create a new vsys, it uses the VsysDefaultProfile. By definition, the VsysDefaultProfile allows access to all resources but does not guarantee them. You can then re-assign a different vsys profile to the new vsys to control resource access. You cannot edit this vsys profile.

- RootProfile

  By default, the root vsys uses the RootProfile. You can configure limits in the Root Profile to reserve certain static resources for the exclusive use of the root vsys.

You can create 18 vsys profiles in addition to the default profiles. After creating profiles, you can assign one or more vsys to a vsys profile.

Setting Resource Limits

The global maximum value for any vsys resource depend on the security device. Vsys uses the default values for the device if you do not explicitly set maximum and reserved limits. To see the vsys limit values, use the **get vsys** *vsysname* command after you create the vsys.

When setting maximum and reserved limits for resources, keep the following in mind:

- You cannot set the maximum value higher than the device-dependent global maximum value. You can view the global maximum values by using the **get vsys-profile global** command.
- For all resources except sessions, you cannot set the maximum value lower than the resources currently being used (actual-use value). To view the actual-use value, use the **get vsys-profile global** command.

  For sessions, you can set the maximum value of sessions lower than the number of sessions used. If you do so, no current sessions are dropped. The maximum value is enforced when the session actual-use value falls below the maximum value, but in the meantime, no new sessions can be created. If you use the **get vsys session-limit** command, the number of available sessions shown is a negative number.
- You cannot set the reserved value higher than the configured maximum value.
- The total allocated usage, which is the sum of reserved values or actual-use values (whichever is higher) for all vsys, cannot exceed the global maximum value.

The following table lists how allocated usage is calculated for MIPs for three vsys (vs1, vs2, and vs3):

	vs1	vs2	vs3	Global
Reserved value (configured value)	20	2	40	
Actual-use	40	15	37	
Allocated usage	40	15	40	95

Although the actual-use value for vs3 is lower than the configured reserved value, the reserved value is used when calculating allocated usage. The global maximum value is 95.

In the following example, you create a new vsys profile with the following settings:

- Name: gold
- CPU weight: 30 (default = 50)
- DIPs: maximum: 25, reserve: 5
- MIPs: maximum: 25
- Mpolicies: maximum: 5
- Policies: maximum: 50
- Sessions: maximum: 1200

WebUI

Vsys > Profile: Select **New**, enter the name and desired settings, then click **OK**.

CLI

```
set vsys-profile name gold cpu-weight 30
set vsys-profile gold dips max 25 reserve 5
set vsys-profile gold mips max 25
set vsys-profile gold mpolicies max 5
set vsys-profile gold policies max 50
set vsys-profile gold sessions max 1200
save
```

Adding Session Limits Through Virtual-System Profile Assignment

You can assign a session limit to a vsys profile in the WebUI or the CLI. To set session limits, you need to configure one or more of the following parameters:

- session max

  The session maximum is a number between 100 and the maximum session number for the overall security system. The default value is the maximum session number for the overall security system (as if no session limitation is in force).

- reserve

  In case of over-subscription, the reserve number is the number of sessions you guarantee or reserve for the specified vsys. The reserve value is a number between zero (0) and the maximum number of sessions you allocate for the specified vsys.

- alarm

  The alarm threshold is a percentage of the maximum limit that triggers the alarm. The default value is 100 percent of the session limit for a configured vsys.

In the following example, you configure a session limitation in a vsys profile named **gold**. The desired limits are as follows:

- Session max: 2500
- Reserve: 2000
- Alarm: 90 (indicates the alarm is triggered when 90 percent of the session maximum is achieved)

A vsys that you assign to this profile can hold up to 2500 sessions at a time. When the overall security device becomes over-subscribed only 2000 sessions are guaranteed to the assigned vsys. At any time, if the assigned vsys consumes 90 percent of the session maximum value an alarm is triggered.

WebUI

Vsys > Profile > Edit

CLI

```
set vsys-profile gold session max 2500 reserve 2000 alarm 90
```

To assign the newly created vsys profile to a vsys named *vsys1*:

WebUI

Vsys > Configure > Edit

CLI

```
set vsys vsys1 vsys-profile name gold
```

Setting a Session Override

For each vsys, you can set an override for a session limit or reserve value defined in an existing vsys profile; you can also override the alarm threshold. To do this, you first enter the vsys and set the override. By default, no overrides exist in virtual systems.

NOTE: ScreenOS associates session overrides with a vsys and not with a vsys profile.

In the following example, you set an override to allow the session maximum to be 3500 instead of 2500.

WebUI

Vsys > Configure > Edit (*vsys*)

CLI

```
enter vsys vsys1
  (vsys1) set override session-limit max 3500
  (vsys1) save
```

Overriding a Session Limit Reached Alarm

You can configure a session limit reached (SLR) alarm. The alarm is triggered when the SLR level is reached or exceeded. The security device removes the alarm if the number of sessions of the vsys drops below the alarm trigger level for 10 consecutive seconds. The security device logs the alarm messages.

You can configure Simple Network Management Protocol (SNMP) traps for vsys SLR alarms. For more information about SNMP, see *"Fundamentals" on page 15*.

In the following example, you configure an alarm to be triggered when the number of vsys sessions is 80 percent of the session limit. The original gold profile indicates that the alarm is triggered at 90 percent of the session limit.

WebUI

Vsys > Configure > Edit (*vsys*)

CLI

```
enter vsys vsys1
  (vsys1) set override session-limit alarm 80
  (vsys1) save
```

Deleting a Vsys Profile

You can delete a vsys profile in the WebUI or the CLI. Before you delete a vsys profile, make sure that the profile is not used by any vsys. ScreenOS does not allow you to delete a profile that is in use.

If you receive a message that a profile you want to delete is in use, change the vsys profile of the vsys to use another profile and try to delete the profile again.

In the following example, you delete the vsys profile **gold**.

WebUI

Vsys > Profile: To the right of the vsys profile that you want to delete, click **Remove**.

CLI

```
unset vsys-profile gold
```

Viewing Vsys Settings

The admin for the security device can view the session statistics for all vsys. Within a vsys context, however, you can view only the statistics for that particular vsys.

Viewing Overrides

To view the configured overrides for a particular vsys in the CLI, you can enter the **get vsys** *vsysname* command or the **get vsys override** command. You can also enter the vsys context and then enter the **get override session-limit** command.

The following is sample output for **get vsys vsys2**:

```
device-> get vsys vsys2
```

```
Total number of vsys: 2

Name            Id Profile  Interface           IP Address         Vlan vsd
vsys2            2 VsysDef~ N/A                 N/A                N/A
Vsys-limit        Maximum   Reserved  Actual-use
dips                  254          0           0
mips                  384          0           0
mpolicies             200          0           0
policies              512          0           0
sessions           250064          0           0
user-serv-grps        128          0           0
user-servs            512          0           0
user-zones            215          0           1
zone-addr-grps        512          0           0(Untrust)
zone-addrs          20000          4           4(Untrust)
cpu-weight             50          -           0
(* - The marked setting has been overridden.)
```

You can also view the overrides in the WebUI.

In the following example, while in a vsys context, you view the reserve for a vsys named branch1.

WebUI

Vsys > Profile > Edit

CLI

enter vsys branch1
(branch1) get override session-limit
(branch1) exit

Viewing a Profile

As root admin, you can view each vsys profile with the WebUI or the CLI. From the WebUI, you cannot view all profiles or a summary of current usage. From the CLI, as root admin, you can view all vsys profiles and a global usage summary that includes actual use statistics. As vsys admin, using the CLI, you can enter a vsys and view the vsys-profile used for the vsys.

WebUI

Vsys > Profile: Select a profile to view.

CLI 1

device-> **get vsys-profile red**

```
 vsys-profile-name  ref-cnt  vsys-limit       maximum   reserved   peak-use
-------------------------------------------------------------------------------
 red                     0   dips                 254       0       0
                             mips                 384       0       0
```

```
                            mpolicies            200       0       0
                            policies             512       0       0
                            sessions            3000     100       0
                            user-serv-grps       128       0       0
                            user-servs           512       0       0
                            user-zones           215       0       0
                            zone-addr-grps       512       0       0
                            zone-addrs         20000       4       0
                            cpu-weight = 44, 29% of total cpu-weight 150
                            session alarm level = 100%
------------------------------------------------------------------------------
```

CLI 2

device-> **get vsys-profile**

```
* indicates default vsys profile.
 vsys-profile-name  ref-cnt  vsys-limit       maximum   reserved   peak-use
------------------------------------------------------------------------------
*VsysDefaultProfile      2   dips                254       0       0
                             mips                384       0       0
                             mpolicies           200       0       0
                             policies            512       0       0
                             sessions         250064       0       0
                             user-serv-grps       128       0       0
                             user-servs           512       0       0
                             user-zones           215       0       1(vsys2)
                             zone-addr-grps       512       0       0
                             zone-addrs         20000       4
(vsys2/Unt~)
                             cpu-weight = 50, 33% of total cpu-weight 150
                             session alarm level = 100%
------------------------------------------------------------------------------
 RootProfile              1   dips                254       0       0
                             mips                6144       0       0
                             mpolicies           200       0       0
                             policies           20000       0       0
                             sessions         250064       0       0
                             user-serv-grps       128       0       0
                             user-servs          2048       0       0
                             user-zones           215       0       0
                             zone-addr-grps       512       0       2(Root/Tru~)

                             zone-addrs         20000       0       7(Root/Tru~)

                             cpu-weight = 50, 33% of total cpu-weight 150
                             session alarm level = 100%
------------------------------------------------------------------------------
 red                      0   dips                254       0       0
                             mips                384       0       0
                             mpolicies           200       0       0
                             policies            512       0       0
                             sessions           3000     100       0
                             user-serv-grps       128       0       0
                             user-servs           512       0       0
                             user-zones           215       0       0
                             zone-addr-grps       512       0       0
                             zone-addrs         20000       4       0
                             cpu-weight = 44, 29% of total cpu-weight 150
```

```
                              session alarm level = 100%
------------------------------------------------------------------------------
global usage summary:         global-limit     maximum   allocated  actual
                                                           use      use
------------------------------------------------------------------------------
                              dips                65535       0       0
                              mips                 6145       0       0
                              mpolicies             200       0       0
                              policies            20000       0       0
                              sessions           250064       0       0
                              user-serv-grps        128       0       0
                              user-servs           2048       0       0
                              user-zones            215       1       1
                              zone-addr-grps        512       2       2
                              zone-addrs          20000      95      75
                              total cpu-weight = 150
```

NOTE: The peak-use value is the highest value among all vsys using a vsys profile.

Viewing Session Statistics

To view session statistics, enter the vsys context, then enter the **get session** command.

WebUI

Not available.

CLI

```
(vsys1)-> get session
vsys1: sw alloc 0/max 3500, alloc failed 0, mcast alloc 0
Total 0 sessions shown
(vsys1)->
```

Sharing and Partitioning CPU Resources

By default, all vsys within a single security system share the same CPU resources. It is possible for one virtual system (vsys) to consume excess CPU resources at the expense of other vsys.

For example, if one vsys , within a security system that houses 20 vsys , experiences a denial of service(DOS) attack that consumes all of the CPU resources, the CPU is unable to process traffic for any of the other 19 vsys. In essence, all 20 vsys experience the DOS attack. CPU overutilization protection, also known as the CPU limit feature, is intended to protect against this.

Overutilization protection allows you to configure the security device for fair use, or Fair mode, as opposed to shared use, or Shared mode. To enable a fairer distribution of processing resources, you can assign a flow CPU utilization threshold to trigger a transition to Fair mode, and you can choose a method for transition back to Shared mode. By default, the security device operates in Shared mode.

To enforce fair use, you assign a CPU weight to each vsys that you configure. ScreenOS uses these weights, relative to the weights of all vsys in the security device, to assign time quotas proportional to those weights. ScreenOS then enforces the time quotas over one-second intervals. This means that as long as a vsys does not exceed its time quota over that one-second period and the firewall is not too heavily loaded, no packets for that vsys should be dropped.

NOTE: The CPU overutilization protection feature is independent of the session limits imposed by a vsys profile.

As system admin you determine how much traffic passes through a given vsys in Fair mode by setting its CPU weight in relation to that of other vsys.

You must identify any anticipated burstiness (service curve) while the security system is in Fair mode, and then choose the CPU weight for each vsys appropriately so that bursts pass through the security system. We recommend that, before you deploy the vsys, you verify that adverse packet dropping does not occur with the chosen weights.

With this feature, you can also ensure a fixed CPU weight for the root vsys.

Configuring CPU Weight

CPU weight is a dimensionless quantity used to calculate the CPU time quota for each vsys. The CPU weight for a vsys is used in combination with the CPU weight for all the other vsys in a security device when calculating the time quota.

For example, you have vsys with the following CPU weights:

- vsys1: 10
- vsys2: 20
- vsys3: 30
- vsys4: 40

The sum of the CPU weights is 100. The time quota is calculated as the ratio of CPU weight to the sum of CPU weights multiplied by the CPU resources and expressed as a percentage of available CPU resources available to a vsys over one-second intervals. The time quotas for the vsys are as follows:

- 10/100: 10 percent
- 20/100: 20 percent
- 30/100: 30 percent
- 40/100: 40 percent

NOTE: CPU weight is not a static resource. ScreenOS recalculates CPU weight when you delete or add a vsys.

When you create a vsys, unless you specify another vsys profile, the default vsys profile (VsysDefaultProfile) is automatically applied. The default vsys profile has a configured CPU weight of 50. You can change the CPU weight for the vsys profile, which applies to the virtual systems that use that vsys profile, or you can override the CPU weight for a vsys by entering the vsys and using the **set override cpu-weight** command.

In the following example, you change the CPU weight to 40 for the **corp-profile** vsys profile.

WebUI

Vsys > CPU Limit: Click **Edit** for the corp-profile vsys profile, type **40** in the CPU Weight field, and click **OK**.

CLI

```
set vsys-profile corp-profile cpu-weight 40
```

Fair Mode Packet Flow

If you enable overutilization protection and the security device becomes heavily loaded, ScreenOS transitions the device to Fair mode.

While in Fair mode, ScreenOS processes a packet as follows:

1. The system allocates resources for the packet and timestamps it.
2. The flow CPU processes the packet.
3. The system determines the vsys against which the packet should be charged and the time-quota balance of that vsys. If the vsys is over its time quota, the system drops the packet. See Table 119 on page 1698 to see how ScreenOS determines which vsys to charge.
4. After the system processes the packet, the system computes the CPU processing time for the packet from the current time and timestamp from step 1. The system then charges the amount against the remaining time quota for the vsys.

When the time quota of a vsys is exhausted, the ScreenOS drops all subsequent packets for that vsys.

Table 119: Determining Charged Vsys

Source Vsys	Destination Vsys	Charged Vsys
Root	Root	Root
Root	Destination vsys	Destination vsys
Source vsys	Root	Source vsys
Source vsys	Destination vsys	Source vsys

NOTE: This packet dropping (enforcement) is done only in Fair mode.

The ScreenOS refreshes time quotas every 125 milliseconds.

NOTE: CPU overutilization protection is performed solely by the flow CPU with no hardware support. This feature provides a best effort to process packets of vsys that are not over their time quotas. There is no guarantee, however, that each vsys cannot use more than its assigned time quota, as it takes time to determine the appropriate vsys against which packets are charged.

The time required to drop packets for a vsys that is over its time quota is also charged to that vsys. If a vsys is receiving heavy traffic and is consistently over its time quota, no packets can pass through the system for that vsys.

However, on Juniper Networks security devices that support blacklisting of DoS attack traffic, the device drops the packet, based on the blacklist that you configure. In addition, such platforms support prioritizing the traffic in high-CPU utilization situations such as a DoS attack to ensure that critical traffic is not affected even though noncritical traffic may be dropped. On such devices, these features are implemented on the entire device, not on a virtual system basis. For more information about device-based traffic blacklisting, see *"CPU Protection with Blacklisting DoS Attack Traffic" on page 468* and *"Prioritizing Critical Traffic" on page 470*.

Returning from Fair Mode to Shared Mode

Depending on how a root admin configures the security device, ScreenOS takes one of the following actions:

- Remains in Fair mode until an admin explicitly configures the security device to Shared mode.
- Returns to Shared mode after a specific time limit.

Return to Shared mode automatically after the projected flow CPU utilization falls below a configured threshold.

Enabling the CPU Limit Feature

Before you can use many of the CPU limit commands in the CLI, you must first initialize and allocate resources for the feature. After configuring the CPU limit parameters using the CLI, you then must enable the feature.

WebUI

Vsys > CPU Limit: Select the CPU Limit Enable check box, then click **OK**.

CLI

To initialize and allocate resources for the CPU limit feature:

```
set cpu-limit
```

After configuring the CPU limit parameters, to enable the feature:

```
set cpu-limit enable
```

To disable the feature:

```
unset cpu-limit enable
```

To disable the feature and deallocate resources:

```
unset cpu-limit
```

Measuring CPU Usage

Each security device measures how many CPU cycles have passed. Using the CPU weights for each vsys within a security device, you can assign a resource quota to each vsys.

To determine the current CPU usage for a security system, log in as the root admin and use the **get performance cpu-limit** or **get vsys cpu-limit** command. These commands return a per-vsys breakdown of the percentage of CPU usage in terms of the percentage of CPU time quota assigned to each vsys.

NOTE: Before you can use these commands, you must enable the CPU limit feature by using the **set cpu-limit enable** command. For more information about this command, refer to the *ScreenOS CLI Reference Guide: IPv4 Command Descriptions*.

The following output for a security device with Fair mode enabled shows a total of six configured vsys (five vsys plus the root vsys):

```
                                CPU Quota %
Vsys Name     Wgt Cfg %   1 min  5 min  15 min
Root           50   16.6      0      0       0
corp           50   16.6     99     99      99
v1             50   16.6      8     18      10
v2             50   16.6      8     18      10
v3             50   16.6      7     17       9
v4             50   16.6      7     17      13
```

The output lists the following details:

- Vsys Name
- Wgt—configured CPU weight for this vsys
- Cfg %—configured percentage of CPU resources for this vsys

- 1 min—percentage of CPU quota used by this vsys in the last minute
- 5 min—percentage of CPU quota used by this vsys in the last 5 minutes
- 15 min—percentage of CPU quota used by this vsys in the last 15 minutes

In the previous example, of the configured vsys, vsys corp used almost all of its CPU time quota in the last minute, last 5 minutes, and last 15 minutes. Except for the root vsys, which used no CPU resources, the other vsys used 7 to 8 percent of their CPU time quotas in the last minute and 17 to 18 percent of their CPU time quotas in the last 5 minutes.

To look at detailed packet data for a vsys, use the **get performance cpu-limit detail vsys all** *vsysname* command. This command returns statistics for the specified vsys over the last 60 seconds and last 60 minutes.

The following output shows the following information:

- Number of packets successfully passed
- Number of dropped packets
- CPU quota in percentages

```
device-> get performance cpu-limit detail vsys corp
vsys corp:
Last 60 seconds (paks passed,paks dropped by cpu limit/cpu quota %):
59:   916, 10550/78  58:  1206, 13796/99  57:  1252, 13751/99
56:  1255, 13747/99  55:  1302, 13700/99  54:  1308, 13694/99
53:  1337, 13666/99  52:  1232, 13770/99  51:  1222, 13780/99
50:  1263, 13740/99  49:  1322, 13680/99  48:  1311, 13691/99
47:  1334, 13668/99  46:  1317, 13686/99  45:  1319, 13683/99
44:  1322, 13680/99  43:  1333, 13670/99  42:  1323, 13679/99
41:  1337, 13665/99  40:  1333, 13670/99  39:  1331, 13671/99
38:  1325, 13678/99  37:  1318, 13685/99  36:  1319, 13683/99
35:  1318, 13685/99  34:  1333, 13668/99  33:  1355, 13647/99
32:  1346, 13656/99  31:  1360, 13642/99  30:  1360, 13643/99
29:  1351, 13651/99  28:  1346, 13656/99  27:  1357, 13646/99
26:  1339, 13663/99  25:  1337, 13665/99  24:  1356, 13646/99
23:  1329, 13674/99  22:  7190,  6961/99  21: 13164,     0/ -
20: 13219,     0/ -  19: 13765,     0/ -  18: 15136,     0/ -
17:  7730,     0/ -  16:   200,     0/ -  15:   200,     0/ -
14:   200,     0/ -  13:   200,     0/ -  12:   200,     0/ -
11:   200,     0/ -  10:   200,     0/ -   9:   200,     0/ -
 8:   200,     0/ -   7:   200,     0/ -   6:   200,     0/ -
 5:   200,     0/ -   4:   200,     0/ -   3:   200,     0/ 7
 2:   648,  5566/47   1:  1317, 13685/99   0:  1333, 13670/99

Last 60 minutes (paks passed,paks dropped by cpu limit):
59:    77968,  471526 58:    85666,  537590 57:    33921,  523433
56:    21110,  564548 55:    80572,  748114 54:    91814,  538566
53:    83932,  544342 52:    72268,  624337 51:     1339,  708070
50:    87790,  970630 49:    96317, 1084226 48:    68805,  267087
47:        0,       0 46:        0,       0 45:        0,       0
44:        0,       0 43:        0,       0 42:        0,       0
41:        1,       0 40:        0,       0 39:        0,       0
38:        0,       0 37:        0,       0 36:        0,       0
35:        0,       0 34:        0,       0 33:        0,       0
32:        0,       0 31:        0,       0 30:        0,       0
29:        0,       0 28:        0,       0 27:        0,       0
```

```
26:        0,       0 25:        0,       0 24:        0,       0
23:        0,       0 22:        0,       0 21:        1,       0
20:        0,       0 19:        0,       0 18:        0,       0
17:        0,       0 16:        0,       0 15:        0,       0
14:        0,       0 13:        0,       0 12:        0,       0
11:        0,       0 10:    90714,  679865  9:    86549, 1478569
 8:    88999, 1429512  7:   238258,  566208  6:   316219,  479793
 5:   477711,       0  4:   376981,       0  3:   439035,       0
 2:   395397,  735399  1:    87908,  743423  0:        0,       0
```

This output shows that in the last 60 seconds, the corp vsys exceeded its assigned CPU quota from second 0 until second 2 and from second 22 to second 59, with an approximate average packet drop rate of over 10,000 packets per second.

For instance, at second 1,1,317 packets were passed, but 13,685 packets were dropped, because the corp vsys went over its assigned CPU quota. From second 3 until second 16, the corp vsys passed 200 packets per second, and the security device returned to Shared mode (ScreenOS outputs "-" in the % CPU quota column when in Shared mode). At second 22, the system reentered Fair mode.

As root admin, you have several options for the level of detail when viewing CPU utilization statistics. See Table 120 on page 1702.

Table 120: Get Command Options for CPU Utilization Protection

Command	Purpose
get performance cpu-limit	Returns CPU weights and corresponding CPU time quota percentages and CPU quota percentages for all vsys.
get performance cpu-limit detail vsys *vsysname*	Returns detailed statistics collected over the last 60 seconds and 60 minutes for the specified vsys.
get cpu-limit utilization	Returns the flow CPU utilization or the projected flow CPU utilization over the last 60 seconds: ■ When the device is in Shared mode, the number displayed is the flow CPU utilization. ■ When the device is in Fair mode, the number displayed is the projected flow CPU utilization. Use to determine the shared-to-fair, fair, and fair-to-shared automatic thresholds. Asterisk to the right of the number indicates that the device was in Fair mode at that time. Utilization shown using this command is 8 to 12 percent lower than the output shown using the **get performance cpu** command, because the **get cpu-limit utilization** command does not include some overhead values.
get vsys cpu-limit	Shows the same output as the **get performance cpu-limit** command.

Detailed Session Scan Debugging

You can get detailed statistics about a task using the task debug command. To do this, you set the debug option on a task, and then get the task details.

Command	Purpose
set task task-name | task-id debug	Sets the debug option on the specified task.
get task task-id	Returns the subtask details of the specified task. ■ Runtime: Subtask's CPU time consumption (in seconds) ■ Name: Subtask name ■ RunCnt: Number of times the subtask has been run ■ Schedule: Number of times the subtask was paused and resumed ■ LockLatency: Time spent by the CPU in resolving reservation of exclusive access for resources for a CPU in multi-CPU platforms

Setting the Shared-to-Fair Mode CPU Utilization Threshold

Perform the following steps to set a security system to transition from Shared mode (the default) to Fair mode in order to protect CPU resource availability for other vsys. You might have to repeat portions of this procedure until you are satisfied with the settings and have verified their effectiveness.

You can set the CPU utilization threshold in the WebUI or the CLI. The WebUI example is a summary of the command choices. The steps in the CLI example are complete.

NOTE: The flow CPU utilization, as configured by the CPU limit feature, is calculated differently from the output of the **get performance cpu** command. To set the shared-to-fair threshold, use one of the following procedures.

WebUI

Vsys > CPU Limit: Select the CPU Limit Enable check box, then click **OK**.

Fair to Shared: Select how or if you want the security device to return to Shared mode. If you select Automatic, enter a threshold. If you select Fair Time, which is an explicit number of seconds for the device to use Fair mode, enter the desired number of seconds.
Shared to Fair: Enter a threshold, then enter a hold-down time (optional).

CLI

1. Verify that the device is not processing traffic.
2. To initialize the CPU limit feature:

   set cpu-limit

3. The **shared-to-fair-threshold** setting indicates the threshold above which the security device transitions from Shared mode to Fair mode.

   You can also optionally set a hold-down time, which is the minimum amount of time the CPU usage exceeds the specified shared-to-fair threshold at which the security mode enters Fair mode.

   To set the **shared-to-fair-threshold** and **hold-down time** (optional), enter a threshold value (a number from 1 to 100) and a hold-down time (a value from 0 through 1800 seconds):

   set cpu-limit shared-to-fair threshold *threshold* [hold-down-time *number*]

4. To enable the CPU limit, enter the following command:

   set cpu-limit enable

5. To verify the current mode and other CPU limit parameters:

   get cpu-limit

```
device-> get cpu-limit
Current mode: shared
Shared->fair: threshold 80%, hold down time 1
Fair->shared: automatic, threshold 70%
CPU limit: enabled
```

6. Send traffic at a level that should keep the device in Shared mode.
7. To verify that the security system stays in Shared mode:

   get cpu-limit utilization

   Sample output:

```
device-> get cpu-limit utilization
Last 60 seconds:
59:  14   58:  14   57:  14   56:  14   55:  14   54:  14
53:  14   52:  15   51:  14   50:  14   49:  14   48:  15
47:  14   46:  15   45:  14   44:  15   43:  14   42:  15
.... [output continues]
```

If asterisks appear in the output after the traffic is started, the device is in Fair mode, and the shared-to-fair threshold is too low. Perform the following steps:

a. Stop traffic.

b. Raise the shared-to-fair threshold with the **set cpu-limit shared-to-fair threshold** *flow_threshold* command.

c. To force the security system to return to Shared mode:

```
exec cpu-limit mode shared
```

d. Restart traffic and repeat step 7 as necessary.

8. Increase the traffic to a level that should force the device to transition to Fair mode.

9. To verify that the security system transitioned to Fair mode:

```
get cpu-limit utilization
```

```
device-> get cpu-limit utilization
Last 60 seconds:
59:  96*  58:  94*  57:  96*  56:  96*  55:  96*  54:  82*
53:  20   52:  14   51:  15   50:  14   49:  15   48:  14
47:  14   46:  14   45:  14   44:  14   43:  14   42:  14
41:  15   40:  14   39:  15   38:  15   37:  15   36:  14
35:  15   34:  14   33:  14   32:  14   31:  15   30:  14
29:  14   28:  14   27:  15   26:  15   25:  15   24:  15
23:  91*  22:  96*  21:  97*  20:  96*  19:  97*  18:  96*
17:  97*  16:  96*  15:  98*  14:  96*  13:  98*  12:  96*
11:  98*  10:  96*   9:  97*   8:  96*   7:  97*   6:  96*
 5:  97*   4:  96*   3:  97*   2:  96*   1:  97*   0:  96*
(* - In Fair mode; projected CPU utilization displayed.)
```

If the system is not in Fair mode after the traffic is increased, the shared-to-fair threshold is set too high. Follow these steps:

a. Stop the traffic.

b. Lower the shared-to-fair threshold by entering the **set cpu-limit shared-to-fair threshold** *flow_threshold* command.

NOTE: You can use the **get cpu-limit utilization** command as a guide.

c. Restart the traffic and repeat step 9 until the threshold is correct.

Configuring a Method for Returning to Shared Mode

After setting the shared-to-fair CPU utilization threshold, you can configure the device to transition to Shared mode automatically, transition to Shared mode after an explicit period, or stay in Fair mode.

- To configure automatic transition to Shared mode, you configure a threshold value for the security device. The threshold is the projected flow CPU utilization value below which the security device transitions from Fair-to-Shared mode. You can also configure a hold-down time, which is the minimum amount of time

that the flow CPU utilization percentage must exceed the flow CPU utilization percentage threshold.

- To configure an explicit period, set the security device to use Fair mode with a fair time setting. The fair time can be between zero (0) and 7200 seconds.
- To maintain Fair mode, select **never**.

In the following example, logged in as the root admin, you configure the security device to automatically revert back to Shared mode after the projected flow CPU utilization falls below a specific threshold.

In this example, assume that you are setting the fair-to-shared CPU utilization threshold for the first time, so you might have to repeat the steps to try different settings before the device behaves as you expect it to. Verification steps are included. The example shows the CLI commands.

NOTE: This procedure is necessary only if you want to enable Fair Automatic mode.

WebUI

Vsys > CPU Limit: Select the CPU Limit Enable check box, then click **OK**.

Fair to Shared: Select how or if you want the security device to return to Shared mode. If you select **Automatic**, enter a threshold.

CLI

1. Verify that traffic is arriving at the security device at a level that keeps the device in Fair mode.
2. To set the Fair Automatic threshold:

   set cpu-limit fair-to-shared automatic threshold *number*

3. Lower the traffic rate to a level that should trigger the device to return to Shared mode.
4. To Verify that the device returns to Shared mode enter the **get cpu-limit utilization** command.
5. If the device is still in Fair mode, repeat all steps in this procedure using a lower value for the fair automatic threshold.

Setting a Fixed Root Vsys CPU Weight

To specify an explicit CPU percentage for the root vsys, you can calculate the root vsys CPU weight. However, when you add or delete a vsys, you have to recalculate the root vsys CPU weight.

To ensure that the calculated root vsys CPU weight is correct, you must configure the CPU weights of all other vsys. Then use the following formula to compute the required root vsys CPU weight:

R = PW/(1–P)

where:

- R is the root vsys CPU weight
- P is the proportion of the CPU desired for root vsys, where:
- W is the sum of the weights of all the other vsys.

In the following example, you want to assign 30 percent of the CPU resources to the root vsys when you have four vsys distributed among three vsys profiles, as follows:

- Gold profile (CPU weight = 40): 1 vsys
- Silver profile (CPU weight = 30): 2 vsys
- RootProfile: 1 root vsys

The sum of the CPU weights of all vsys excluding the root vsys (W) is 100. The percentage (P) you want to assign to the root vsys is 30 percent or .3.

Using the previous equation: R = P*W/(1-P) = .3*100/.7 = 43

To check, the root vsys percentage is 43/(100 + 43), which yields approximately 30 percent.

If you add or delete a vsys in the future, you must redetermine *W* and recalculate the root vsys CPU weight *R*.

Virtual Systems and Virtual Private Networks

The root vsys admin can view the following virtual private network (VPN) information:

- All configured or only active security associations (SAs)
- Internet Key Exchange (IKE) cookies

Read-write and read-only vsys admins can see the information that pertains only to their vsys.

The next sections explain more about this information and how to view it.

Viewing Security Associations

If you are the root system admin for the security device, you can view the SAs for all vsys by entering the **get sa** command. When you issue this command, you retrieve the total number of IPsec SAs stored in the security device, which is the root system plus all configured vsys.

WebUI

VPNs > Monitor Status

CLI

```
get sa
```

If you are a vsys admin and are using the CLI, you can view the SAs that are applicable to your particular vsys by entering a vsys context and then entering the **get sa** command.

To view only the active SAs, enter the **get sa active** command.

In the following example, as vsys admin for clothing_store, you can view only the active SAs for your vsys.

WebUI

VPNs > Monitor Status

CLI

```
enter vsys clothing_store
(clothing_store) get sa active
(clothing_store) exit
```

Viewing IKE Cookies

You can view IKE cookies from the CLI only.

As system admin for the security device, you can view all of the IKE cookies for the system, which is the root plus the vsys IKE cookies. You can view them from the CLI by entering the **get ike cookie** command.

In the following example, as system admin, you view the IKE cookies.

WebUI

Not available.

CLI

```
get ike cookie
```

As a vsys admin, you can view the IKE cookies for the vsys you manage by entering the vsys context and then entering the **get ike cookie** command.

In the following example, you view the IKE cookies for the vsys you manage, *card_shop*.

WebUI

Not available.

CLI

```
enter vsys card_shop
(card_shop) get ike cookie
(card_shop) exit
```

Policy Scheduler

Within a vsys context, a vsys admin can schedule a single or recurrent timeslot within which a policy is active.

When a new policy is created, a vsys admin can create a scheduler and then bind it to one or more existing policies. The session ages out when the scheduler times out.

This section explains the following tasks:

- "Creating a Policy Scheduler" on page 1709
- "Binding a Policy Schedule to a Policy" on page 1710
- "Viewing Policy Schedules" on page 1710
- "Deleting a Policy Schedule" on page 1711

Creating a Policy Scheduler

As a vsys admin, you can schedule a policy to be active for one time only or on a recurrent basis. You can configure a meaningful name and a start and stop time for the scheduler. You can also attach comments.

In the following example, you configure a scheduler for a recurring service restriction from Monday to Friday from 8:00 to 11:30 AM and from 1:00 to 5:00 PM. The scheduler sets the time restrictions; the policy sets the service restriction. You enter a vsys context and perform the configuration in the vsys context.

WebUI

Vsys > Configure > Enter

Objects > Schedules > Click **New** and fill in the schedule form, then click **OK**.

CLI

```
device(hr)-> set scheduler restrictionM recurrent monday start 8:00 stop 11:30 start
 13:00 stop 17:00
device(hr)-> set scheduler restrictionTu recurrent tuesday start 8:00 stop 11:30 start
 13:00 stop 17:00
```

```
device(hr)-> set scheduler restrictionW recurrent wednesday start 8:00 stop 11:30
start 13:00 stop 17:00
device(hr)-> set scheduler restrictionTh recurrent thursday start 8:00 stop 11:30
start 13:00 stop 17:00
device(hr)-> set scheduler restrictionF recurrent friday start 8:00 stop 11:30 start
13:00 stop 17:00
device(hr)-> save
```

Binding a Policy Schedule to a Policy

You can attach a scheduler to a policy as you create the policy, or you can bind the scheduler later in the WebUI.

In this example, you configure a new policy from Trust to Untrust and set the source and destination address-book entries.

WebUI

Vsys > Configure > Enter

Policies: Select the zones, then click **New**. After configuring the policy, click **Advanced**: At the bottom of the page, select the schedule from the drop-down list, then click **OK**.

CLI

```
device(hr)-> set policy id 1 from trust to untrust company any http deny schedule
restrictionM
device(hr)-> set policy id 1 from trust to untrust company any http deny schedule
restrictionTu
device(hr)-> set policy id 1 from trust to untrust company any http deny schedule
restrictionW
device(hr)-> set policy id 1 from trust to untrust company any http deny schedule
restrictionTh
device(hr)-> set policy id 1 from trust to untrust company any http deny schedule
restrictionF
device(hr)-> save
```

Viewing Policy Schedules

To view configured schedules:

WebUI

Vsys > Configure > Enter

Objects > Schedules

CLI

```
device(hr)-> get scheduler
```

Deleting a Policy Schedule

In the following example, as a vsys admin, you delete the schedule named **restrictionW**.

WebUI

Vsys > Configure > Enter

Objects > Schedules: Click **Remove**.

CLI

```
device(hr)-> unset scheduler restrictionW
```

Chapter 53
Traffic Sorting

This chapter explains how ScreenOS sorts traffic. It contains the following sections:

Overview

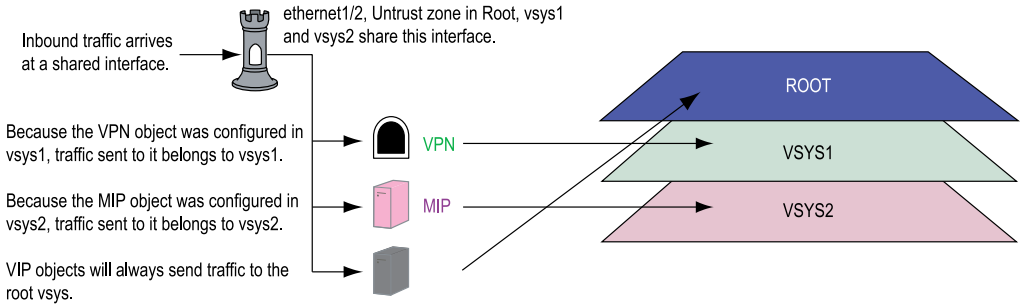
ScreenOS sorts every packet that it receives for delivery to the proper virtual system (vsys). A security device receives two kinds of user traffic, which it sorts in two different ways:

- Traffic destined for an IP address within the system itself, such as encrypted VPN traffic and traffic destined for a MIP
- Traffic destined for an IP address beyond the device

Sorting Traffic

For traffic destined for an object (VPN or MIP) on the security system, the system determines the vsys to which the traffic belongs through the association of the object with the vsys in which it was configured. Figure 421 on page 1713 displays how traffic is sorted.

Figure 421: VPN and MIP VIP Association



Inbound traffic can also reach a vsys through VPN tunnels; however, if the outgoing interface is a shared interface, you cannot create an AutoKey IKE VPN tunnel for a vsys and the root system to the same remote site.

Sorting Through Traffic

For traffic destined for an IP address beyond the security device (also known as "through traffic"), the device uses techniques made possible by VLAN-based and IP-based traffic classifications. VLAN-based traffic classification uses VLAN tags in frame headers to identify the system to which inbound traffic belongs. IP-based traffic classification uses the source and destination IP address in IP packet headers to identify the system to which traffic belongs. The process that the security device uses to determine the system to which a packet belongs progresses through the following three steps:

NOTE: VLAN tagging requires the use of subinterfaces. A subinterface must be dedicated to a system, in contrast to a shared interface, which is shared by all systems.

1. **Ingress Interface/Source IP Traffic Classification**

   The security device checks if the ingress interface is a dedicated interface or a shared interface.

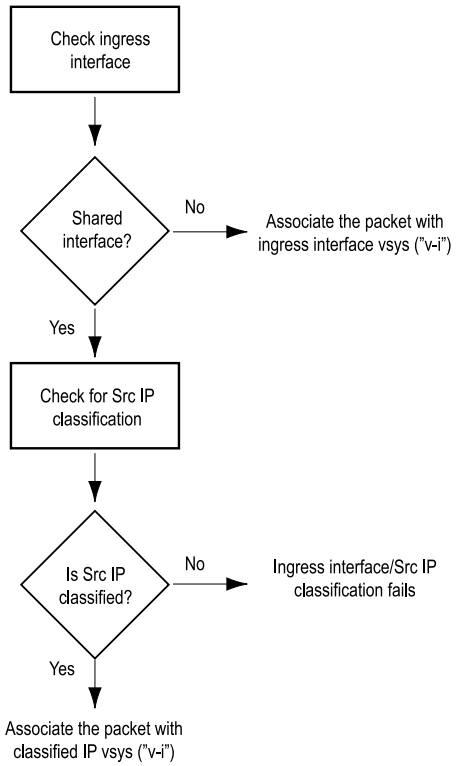
NOTE: For more information about shared and dedicated interfaces, see "Dedicated and Shared Interfaces" on page 1718.

   a. If the ingress interface is dedicated to a vsys ("v-i", for example), the security device associates the traffic with the system to which the interface is dedicated.

   b. If the ingress interface is a shared interface, the security device uses IP classification to check if the source IP address is associated with a particular vsys. See Figure 422 on page 1715.

      - If the source IP address is not associated with a particular vsys, ingress IP classification fails.
      - If the source IP address is associated with a particular vsys, ingress IP classification succeeds.

Figure 422: Step 1—Ingress Interface and Source IP Traffic Classification



*Extensible Authentication Protocol over LAN (EAPOL) is a protocol described in IEEE 802.1X. It was created to encapsulate EAP messages for transport across a local area network.

2. **Egress Interface/Destination IP Traffic Classification**

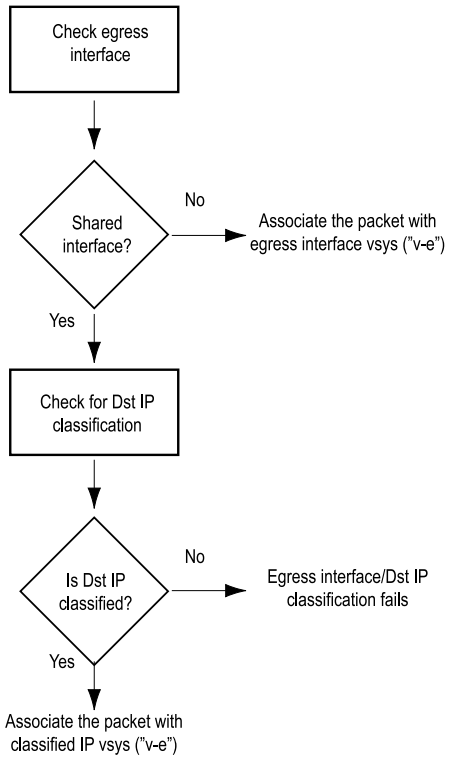
   The security device checks if the egress interface is shared or dedicated.

   a. If the egress interface is dedicated to a vsys ("v-e", for example), the security device associates the traffic with the system to which the interface is dedicated. Note that VIP traffic on an untrust interface will always be routed to the root vsys.

   b. If the egress interface is a shared interface, the security device uses IP classification to check if the destination IP address is associated with a particular vsys. See Figure 423 on page 1716.

      - If the destination IP address is not associated with a particular vsys, egress IP classification fails.
      - If the destination IP address is associated with a particular vsys, egress IP classification succeeds.

Figure 423: Step 2—Egress Interface/Destination IP Traffic Classification



*Extensible Authentication Protocol over LAN (EAPOL) is a protocol described in IEEE 802.1X. It was created to encapsulate EAP messages for transport across a local area network.

3. **Vsys Traffic Assignment**

   Based on the outcome of the ingress interface/source IP (I/S) and egress interface/destination IP (E/D) traffic classifications, the security device determines the vsys to which traffic belongs. See Figure 424 on page 1718.

   a. If I/S traffic classification succeeds, but E/D traffic classification fails, the security device uses the policy set and route table for the vsys associated with the ingress interface or source IP address (a vsys named “v-i”, for example).

      I/S traffic classification is particularly useful when permitting outbound traffic from a vsys to a public network such as the Internet.

   b. If E/D traffic classification succeeds, but I/S traffic classification fails, the security device uses the policy set and route table for the vsys associated with the egress interface or destination IP address (a vsys named “v-e”, for example).

      E/D traffic classification is particularly useful when permitting inbound traffic to one or more servers in a vsys from a public network such as the Internet. Note that VIP traffic on an untrust interface will always be routed to the root vsys.

c. If both classification attempts succeed and the associated virtual systems are the same, the security device uses the policy set and route table for that vsys.

   You can use both I/S and E/D IP traffic classification to permit traffic from specific addresses in one zone to specific addresses in another zone of the same vsys.

d. If both classification attempts succeed, the associated virtual systems are different, and the interfaces are bound to the same shared security zone, the security device first uses the policy set and route table for the I/S vsys, and then uses the policy set and route table for the E/D vsys.

   ScreenOS supports intrazone intervsys traffic when the traffic occurs in the same shared zone. The security device first applies the "v-i" policy set and route table, loops the traffic back on the Untrust interface, and then applies the "v-e" policy set and route table. Such intrazone traffic might be common if a single company uses one shared internal zone with different virtual systems for different internal departments and wants to allow traffic between the different departments.

e. If both classification attempts succeed, the associated virtual systems are different, and the interfaces are bound to different shared security zones, then the security device drops the packet.

NOTE: ScreenOS does not support interzone intervsys traffic between shared security zones. You cannot use a custom zone instead of the Untrust zone.

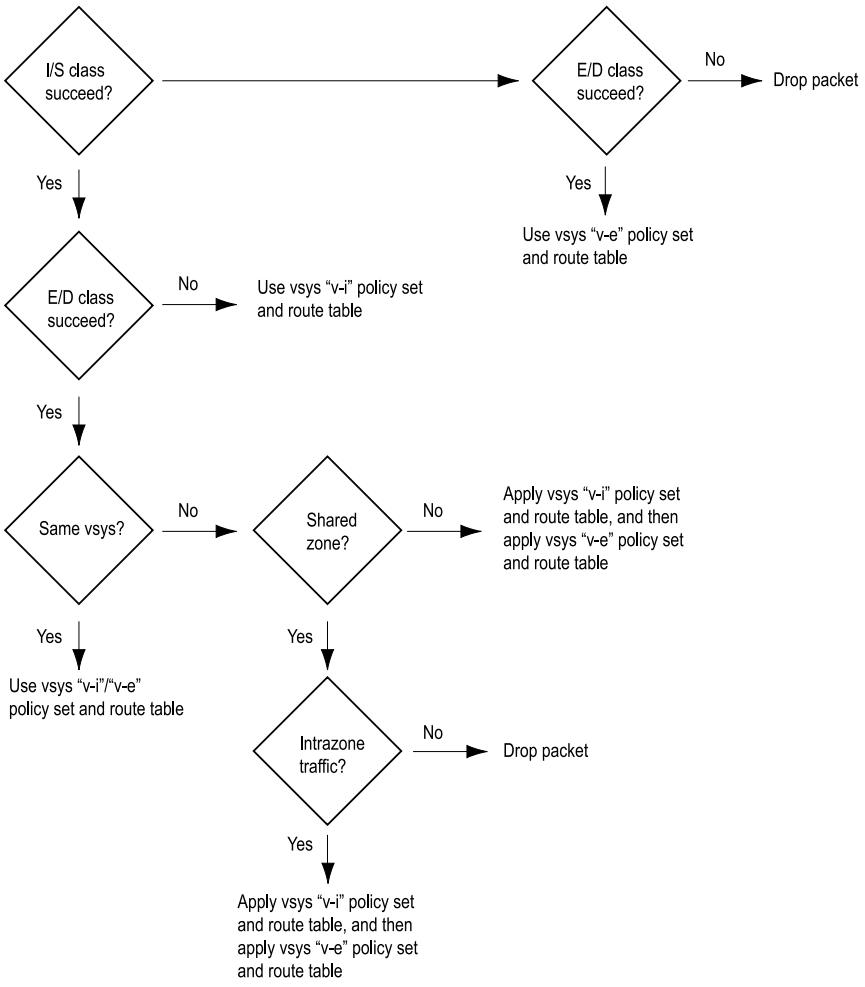
---

f. If both classification attempts succeed, the associated virtual systems are different, and the ingress and egress interfaces are bound to zones dedicated to different virtual systems, the security device first applies the "v-i" policy set and route table. It then loops the traffic back on the Untrust interface and applies the "v-e"policy set and route table. (See "Communicating Between Virtual Systems" on page 1748.)

   ScreenOS supports interzone intervsys traffic between dedicated security zones.

g. If both classification attempts fail, the security device drops the packet.

Figure 424: Step 3—Vsys Traffic Assignment



Dedicated and Shared Interfaces

Inbound traffic to dedicated and shared interfaces is sorted differently.

Dedicated Interfaces

A system—virtual and root—can have multiple interfaces or subinterfaces dedicated exclusively to its own use. Such interfaces are not sharable by other systems.

You can dedicate an interface to a system as follows:

- When you configure a physical interface, subinterface, redundant interface, or aggregate interface in the root system and bind it to a nonsharable zone, that interface remains dedicated to the root system.
- When you import a physical or aggregate interface into a vsys and bind it to either the shared Untrust zone or the Trust-vsys_name zone, that interface becomes a dedicated interface for that vsys.

- When you configure a subinterface in a vsys, it belongs to that vsys.

NOTE: When a system has a dedicated subinterface, the security device must employ VLAN-based traffic classification to properly sort inbound traffic.

Shared Interfaces

A system—virtual and root—can share an interface with another system. For an interface to be sharable, you must configure it at the root level and bind it to a shared zone in a shared virtual router. By default, the predefined untrust-vr is a shared virtual router, and the predefined Untrust zone is a shared zone. Consequently, a vsys can share any root-level physical interface, subinterface, redundant interface, or aggregate interface that you bind to the Untrust zone.

To create a shared interface in a zone other than the Untrust zone, you must define the zone as a shared zone at the root level. To do that, the zone must be in a shared virtual router, such as the untrust-vr or any other root-level virtual router that you define as sharable. Then, when you bind a root-level interface to the shared zone, it automatically becomes a shared interface.

NOTE: For the shared zone option to be available, the security device must be operating at Layer 3 (route mode), which means that you must previously assign an IP address to at least one root-level interface.
To create a virtual router, you need to obtain a vsys license key, which provides you with the ability to define virtual systems, virtual routers, and security zones for use either in a vsys or in the root system.

A shared virtual router can support both shared and nonsharable root-level security zones. You can define a root-level zone bound to a shared virtual router as sharable or not. Any root-level zone that you bind to a shared virtual router and define as sharable becomes a shared zone, available for use by other virtual systems, too.

Any root-level zone that you bind to a shared virtual router and define as nonsharable remains a dedicated zone for use by the root system alone. If you bind a vsys-level zone to either the virtual router dedicated to that vsys or to a shared virtual router created in the root system, the zone remains a dedicated zone, available for use only by the vsys for which you created it.

A shared zone can support both shared and dedicated interfaces. Any root-level interface that you bind to a shared zone becomes a shared interface, available for use by virtual systems also. Any vsys-level interface that you bind to a shared zone remains a dedicated interface, available for use only by the vsys for which you created it.

A nonsharable zone can only be used by the system in which you created it and can only support dedicated interfaces for that system. All vsys-level zones are nonsharable.

To create a shared interface, you must create a shared virtual router (or use the predefined untrust-vr), create a shared security zone (or use the predefined Untrust zone), and then bind the interface to the shared zone. You must do all three steps in the root system.

The options in the WebUI and CLI are as follows:

1. **To create a shared virtual router:**

   WebUI

   Network > Routing > Virtual Routers > New: Select the **Shared and accessible by other vsys** option, then click **Apply**.

   CLI

   ```
   set vrouter name name_str
   set vrouter name_str shared
   ```

   (You cannot modify an existing shared virtual router to make it unshared unless you first delete all virtual systems. However, you can modify a virtual router from unshared to shared at any time.)

2. **To create a shared zone, do the following at the root level:**

   WebUI

NOTE: At the time of this release, you can only define a shared zone through the CLI.

   CLI

   ```
   set zone name name_str
   set zone zone vrouter sharable_vr_name_str
   set zone zone shared
   ```

3. **To create a shared interface, do the following at the root level:**

   WebUI

   Network > Interfaces > New (or Edit for an existing interface): Configure the interface and bind it to a shared zone, then click **OK**.

   CLI

   ```
   set interface interface zone shared_zone_name_str
   ```

   When two or more virtual systems share an interface, the security device must employ IP-based traffic classification to properly sort inbound traffic. (For more information about IP-based traffic classification, including an example showing how to configure it for several vsys, see “IP-Based Traffic Classification” on page 1757.)

Importing and Exporting Physical Interfaces

You can dedicate one or more physical interfaces to a vsys. In effect, you import a physical interface from the root system to a virtual system. After importing a physical interface to a vsys, the vsys has exclusive use of it.

NOTE: Before you can import an interface to a virtual system, it must be in the Null zone at the root level.

Importing a Physical Interface to a Virtual System

In this example, you—as the root admin—import the physical interface ethernet4/1 to vsys1. You bind it to the Untrust zone and assign it the IP address 1.1.1.1/24.

WebUI

1. **Entering Vsys1**

   Vsys > Configure > Click **Enter** (for vsys1).

2. **Importing and Defining the Interface**

   Network > Interfaces: Click **Import** (for ethernet4/1).

   Network > Interfaces > Edit (for ethernet4/1): Enter the following, then click **OK**:

   Zone Name: Untrust
   IP Address/Netmask: 1.1.1.1/24

3. **Exiting Vsys1**

   Click the **Exit Vsys** button (at the bottom of the menu column) to return to the root level.

CLI

1. **Entering Vsys1**

   ```
   device-> enter vsys vsys1
   ```

2. **Importing and Defining the Interface**

   ```
   device(vsys1)-> set interface ethernet4/1 import
   device(vsys1)-> set interface ethernet4/1 zone untrust
   device(vsys1)-> set interface ethernet4/1 ip 1.1.1.1/24
   device(vsys1)-> save
   ```

3. **Exiting Vsys1**

```
device(vsys1)-> exit
```

Exporting a Physical Interface from a Virtual System

In this example, you bind the physical interface ethernet4/1 to the Null zone in vsys1 and assign it the IP address 0.0.0.0/0. Then you export interface ethernet4/1 to the root system.

WebUI

1. **Entering Vsys1**

   Vsys > Configure > Click **Enter** (for vsys1).

2. **Exporting the Interface**

   Network > Interfaces > Edit (for ethernet4/1): Enter the following, then click **OK**:

   ```
   Zone Name: Null
   IP Address/Netmask: 0.0.0.0/0
   ```

   Network > Interfaces: Click **Export** (for ethernet4/1).

   (Interface ethernet4/1 is now available for use in the root system or in another vsys.)

3. **Exiting Vsys1**

   Click the **Exit Vsys** button (at the bottom of the menu column) to return to the root level.

CLI

1. **Entering Vsys1**

   ```
   device-> enter vsys vsys1
   ```

2. **Exporting the Interface**

   ```
   device(vsys1)-> unset interface ethernet4/1 ip
   device(vsys1)-> unset interface ethernet4/1 zone
   device(vsys1)-> unset interface ethernet4/1 import
   This command will remove all objects associated with interface, continue? y/[n]
   device(vsys1)-> save
   ```

   (Interface ethernet4/1 is now available for use in the root system or in another vsys.)

3. **Exiting Vsys1**

   ```
   device(vsys1)-> exit
   ```

Chapter 54

VLAN-Based Traffic Classification

This chapter explains VLAN-based traffic classification for virtual systems, and VLAN retagging. It includes the following sections:

Overview

With VLAN-based traffic classification, a security device uses VLAN tagging to direct traffic to various subinterfaces bound to different systems.

By default, a vsys has two security zones—a shared Untrust zone and its own Trust zone. Each vsys can share the Untrust zone interface with the root system and with other virtual systems. A vsys can also have its own subinterface or a dedicated physical interface (imported from the root system) bound to the Untrust zone.

NOTE: ScreenOS supports VLANs compliant with the IEEE 802.1Q VLAN standard.

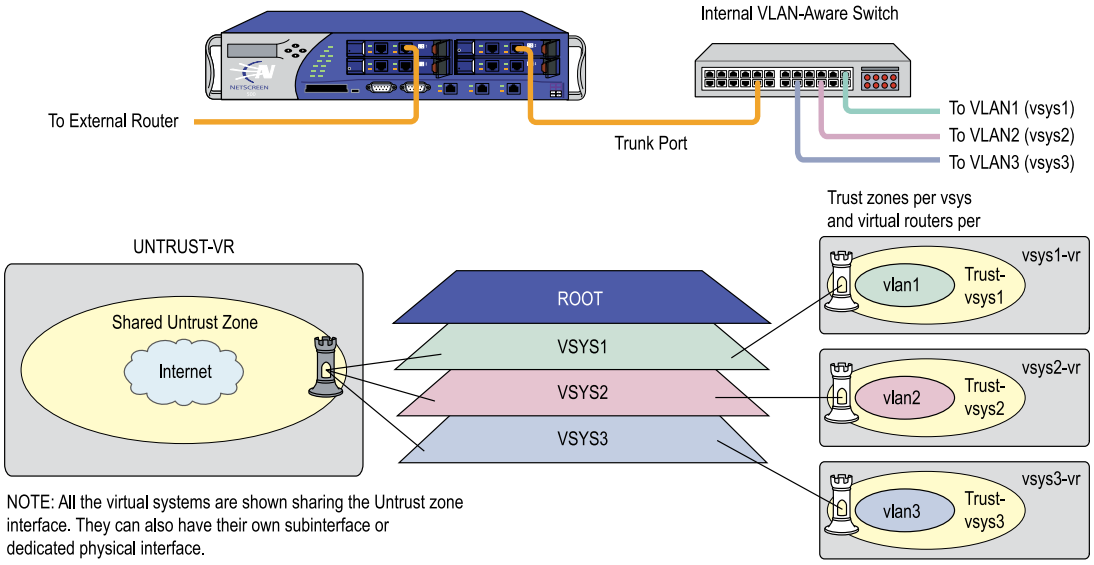
You can dedicate a physical interface to a virtual system by importing it from the root system to the virtual system. (See “Importing and Exporting Physical Interfaces” on page 1721.) When using physical interfaces, VLAN tagging is unnecessary for traffic on that interface.

VLANs

Figure 425 on page 1724 shows VLAN traffic classes. Each VLAN is bound to a system through a subinterface. Use the **set interface** *interface.subid* **tag** *vlanid* **zone** *zone_name* CLI command to assign a VLAN tag to a subinterface.

If a vsys shares the Untrust zone interface with the root system and has a subinterface bound to its Trust-*vsys_name* zone, the vsys must be associated with a VLAN in the Trust-*vsys_name* zone. If the vsys also has its own subinterface bound to the Untrust zone, the vsys must also be associated with another VLAN in the Untrust zone.

Figure 425: VLAN Traffic Classes



A subinterface stems from a physical interface, which then acts as a trunk port. A trunk port allows a Layer 2 network device to bundle traffic from several VLANs through a single physical port, sorting the various packets by the VLAN identifier (VID) in their frame headers. VLAN trunking allows one physical interface to support multiple logical subinterfaces, each of which must be identified by a unique VLAN tag. The VLAN identifier (tag) on an incoming ethernet frame indicates its intended subinterface—and hence the system—to which it is destined. When you associate a VLAN with an interface or subinterface, the security device automatically defines the physical port as a trunk port. When using VLANs at the root level in transparent mode, you must manually define all physical ports as trunk ports with the following CLI command: **set interface vlan1 vlan trunk**.

VLANs with Vsys

When a vsys uses a subinterface (not a dedicated physical interface) bound to the Trust-*vsys_name* zone, the internal switch and internal router in the Trust-*vsys_name* zone must have VLAN-support capabilities. If you create more than one subinterface on a physical interface, you must define the connecting switch port as a trunk port and make it a member of all VLANs that use it.

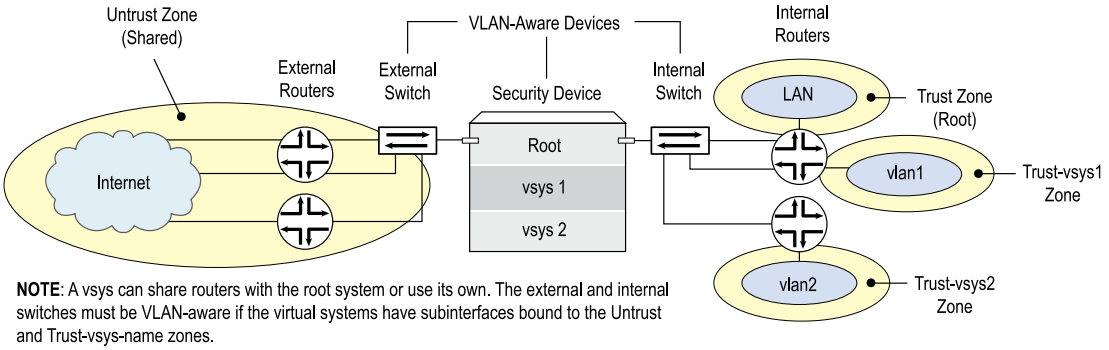
When a vsys uses a subinterface (not a shared interface or a dedicated physical interface) bound to the shared Untrust zone, the external switch and external router that receives its inbound and outbound traffic must have VLAN-support capabilities. The router tags the incoming frames so that when they reach the security device, it can direct them to the correct subinterface.

Although a vsys cannot be in transparent mode, because it requires unique interface or subinterface IP addresses, the root system can be in transparent mode. For the root system to support VLANs while operating in transparent mode, use the following CLI command to enable the physical interfaces bound to Layer 2 security zones to act as trunk ports: **set interface vlan1 vlan trunk**. See Figure 426 on page 1725 for an example of a VLAN using vsys.

There are three tasks that a root-level administrator must perform to create a VLAN for a vsys:

1. Enter a vsys.
2. Define a subinterface.
3. Associate the vsys with a VLAN.

Figure 426: VLAN with Vsys Example



NOTE: When the root system is in transparent mode, it cannot support virtual systems. It can, however, support root-level VLANs while in transparent mode.

VLANs with VSDs

When a VSD group is in Active/Active transparent mode, you can assign one or more VLAN groups to a VSD group member. In this way, VLAN traffic is shared among the VSD group members. A VLAN group consists of one or more VLAN interfaces that are assigned to specific VSD group member. (For more information about VSD groups, see *"Virtual Security Device Groups" on page 1788*.)

NOTE: A VLAN group assigned to a VSD group cannot be assigned to another VSD group. For example, if you have assigned VLAN group 1 to VSD group 1, you cannot assign VLAN group 1 to VSD group 2 again. By default, VLANs not assigned to any VSD group belong to VSD 0 (the default).

Example: Binding VLAN Group with VSD

In this example, you create a VLAN group called v100 and assign the VLAN interfaces 100 and 199 to VSD 0.

WebUI

1. To view the list of VLAN groups assigned to VSD group members, go to:

Network > VLAN > VSD Binding > List

Vsys Name: music

2. To create a VLAN group and assign it to a VSD group member, go to:

Network > VLAN > VSD Binding > New: Enter the following, then click **OK:**

VLAN Group Name: v100
VSD Group ID: 0

CLI

Use the following commands to create a VLAN group and assign VLAN interfaces to a VSD group member:

```
set vlan group name v100
set vlan group v100 100 199
set vlan group v100 vsd id 0
save
```

To view the VLAN groups assigned to VSD group members, use the **get vlan group** command. This command also displays VLANs assigned to a vsys.

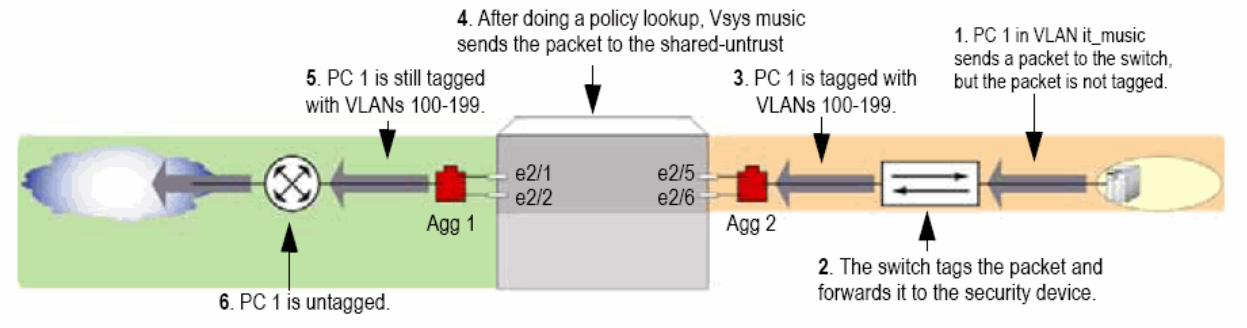
Configuring Layer 2 Virtual Systems

When you configure virtual systems in transparent mode, the security device functions much like a Layer 2 switch or bridge. Packets that traverse the security device are grouped with a unique virtual system (vsys) based on the virtual local area network (VLAN) tag in the packet header. Once the packet is grouped, it performs a policy lookup, then sends the packet through the security device without packet modification.

On the security device, you can logically partition a security system into multiple virtual systems, to provide multi-tenant services. Each vsys is a unique security domain within the device. Each vsys can have its own administrators (called “virtual system administrators” or “vsys admins”) who can individualize their security domains by setting their own objects, such as address books, user lists, custom services, and policies. Administrators then use these objects when defining policies for traffic within a vsys, or between one vsys and other security domains.

Figure 427 on page 1727 shows how the security device transfers data to trusted VLANs using vsys set policies. The numbers in the figure represent the order of data transfer.

Figure 427: How Security Device Uses Vsys set Policies to Transfer Data



Virtual systems in transparent mode are classified by VLAN tags. On the system, a range of VLAN tags is assigned to a VLAN group object, which is then assigned to a security zone that is applied to a port assigned to a vsys. Traffic entering the security system is then classified to the vsys based on the VLAN tag. Once inside the vsys, the traffic is enforced using the configured security zones and policies. The security device can support up to 500 virtual systems in transparent mode. For more information about VLANs and vsys, see “Virtual Systems” on page 1679.

ScreenOS also provides a management interface that manages a vsys when the interface is bound to the zone vlan. The security device creates the management zone vlan automatically when you create a vsys. You can bind more than one interface to the management zone for a single vsys. A VLAN management interface is created within a vsys so that the vsys administrator can manage the virtual systems using a unique IP address and VLAN ID. This management interface allows you to manage your virtual systems remotely or locally.

Only the root administrator can create a vsys and assign resources to it. The root administrator or vsys administrator can then use the CLI or WebUI to create and maintain a vsys configuration.

The security device supports a maximum of 4094 VLANs, which are classified in a vsys by way of VLAN tagging. Each vsys can be assigned from 2 to 4094; however, once a VLAN is assigned to one vsys it cannot be used in another. The root system is identified as vlan1. With a single 8G2 Secure Port Module (SPM), you can configure a maximum of two 4-port aggregate interfaces, four trusted and four untrusted. Assigning the VLANs to an aggregate interface provides a traffic bandwidth of 2 Gps in each direction, with a maximum of 4 Gps for bi-directional traffic per Application-Specific Integrated Circuit (ASIC).

The 8G SPM contains two ASICs. Ports ethernet2/1 through ethernet2/4 use one ASIC, ports ethernet2/5 through ethernet2/8 use the other. Aggregate interfaces must be configured in pairs, starting with port ethernet2/1. The following table shows assigned aggregate ports.

Table 121: 8G SPM

aggregate1	ethernet 2/1 and ethernet 2/2
aggregate 2	ethernet 2/3 and ethernet 2/4

Table 121: 8G SPM *(continued)*

aggregate 3	ethernet 2/5 and ethernet 2/6
aggregate 4	ethernet 2/7 and ethernet 2/8

If you are using the 8G2 SPM and the 5000M2 Management Module, you must use the configuration shown in the following table.

Table 122: 8G2 SPM

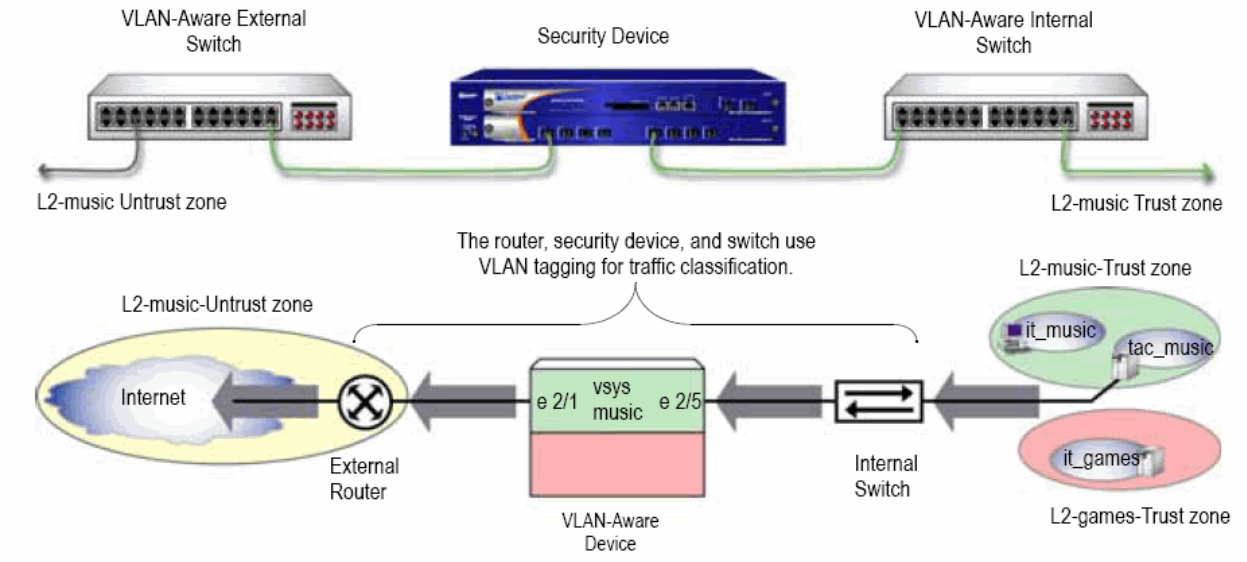
aggregate1	ethernet 2/1 and ethernet 2/2 ethernet 2/3 and ethernet 2/4
aggregate 2	ethernet 2/5 and ethernet 2/6 ethernet 2/7 and ethernet 2/8

If you are using the 8G2 SPM and the 5000M2 Management Module, you must use the configuration shown in the following table.

Example 1: Configuring a Single Port

The figure is created with two tifs on top of each other. Should be an eps done with a single picture. Redo for Qian.In this example, the security device is configured to support the vsys music in transparent mode. This vsys shares the L2-music-Untrust zone with the root system. Figure 428 on page 1728 shows how the security device secures the Trust and Untrust zones. You must import the VLANs to a vsys before they can be tagged.

Figure 428: Single Port



Configuring one transparent mode virtual system (vsys) involves the following steps:

1. Create the vsys named music with a vsys admin name and password.
2. Import (assign) VLAN tags from the root system to classify traffic.
3. Create a VLAN group that contains the VLAN tags to be supported on each port.
4. Create two Layer 2 zones, one for the Trust port and one for the Untrust port.
5. Bind the VLAN group to the ports.

NOTE: Only the root administrator can configure virtual systems, VLAN tags, and Layer 2 zones; however, both the root administrator and vsys administrator can set the management interface and policies. The root administrator can access any vsys in the security device. The vsys administrator can only access the vsys that is assigned to the vsys in which the policies were created.

6. Configure the policies for the vsys music. The policies configured in this example do the following:
   a. Permit HTTP traffic from the L2-music-Untrust zone to the L2-music-Trust zone
   b. Deny all other traffic from the L2-music-Untrust zone to the L2-music-Trust zone
   c. Permit all traffic from then L2-music-Trust zone to the L2-music-Untrust zone
7. Create the management interface. A VLAN management interface is created within a vsys so that the vsys administrator can manage the vsys using a unique IP address and VLAN ID

NOTE: ScreenOS 5.0-L2V with the security device can support up to 500 virtual systems.

WebUI

1. **Create Vsys Music**

   Vsys > Configure > New: Enter the following, then click **OK**:

   Vsys Name: music
   Vsys Admin Name: vsys_music
   Vsys Admin New Password: xyz
   Confirm New Password: xyz

2. **Import VLANs**

   Vsys > Configure > Click Enter (for vsys_music)

   Network > Vlan > Import: Enter the following, then click **Assign**:

Import Vlan ID:
Start: 100
End: 199

3. **Bind Ports**

Network > Vlan > Group > Edit (for it_music) > Port: Enter the following, then click **Add**:

port: (select) ethernet2/5
zone: (select) L2-music-Trust
port: (select) ethernet2/1
zone: (select) L2-music-Untrust

4. **Policies**

Policies (From: L2-music-Untrust, To: L2-music-Trust) > New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select) Any
Destination Address:
Address Book Entry: (select) Any
Service: (select) HTTP
Action: (select) Permit

Policies (From: L2-music-Untrust, To: L2-music-Trust) > New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select) Any
Destination Address:
Address Book Entry: (select) Any
Service: (select) ANY
Action: (select) Deny

Policies (From: L2-music-Trust, To: L2-music-Untrust) > New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select) Any
Destination Address:
Address Book Entry: (select) Any
Service: (select) ANY
Action: (select) Permit

5. **Create Management Interface**

Network > Interfaces > (select **VLAN** in the drop-down list) > New: Enter the following, then click **OK**:

Interface Name Vlan: 199
IP Address/Netmask: 1.0.1.199/24
Management: (deselect)
WebUI: (select)
Telnet: (select)

Ping: (select)

CLI

1. **Create Vsys Music**

```
device-> set vsys music
device(music)-> set admin name vsys_music
device(music)-> set admin password xyz
device(music)-> save
```

2. **Import VLAN Tag**

```
device(music)-> set vlan import 100 199
```

3. **Create VLAN Groups**

```
device(music)-> set vlan group name it_music
device(music)-> set vlan group it_music 100 199
```

4. **Create Layer 2 Zone**

```
device(music)-> set zone name L2-music-Trust L2
device(music)-> set zone name L2-music-Untrust L2
```

5. **Bind Ports**

```
device(music)-> set vlan port ethernet2/5 group it_music zone L2-music-Trust
device(music)-> set vlan port ethernet2/1 group it_music zone L2-music-Untrust
```

6. **Configure Policies**

```
device(music)-> set policy from L2-music-Untrust to L2-music-Trust any any http
 permit
device(music)-> set policy from L2-music-Untrust to L2-music-Trust any any any
 deny
device(music)-> set policy from L2-music-Trust to L2-music-Untrust any any any
 permit
```

7. **Create Management Interface**

```
device(music)-> set interface vlan199 zone vlan
device(music)-> set interface vlan199 ip 1.0.1.199/24
device(music)-> unset interface vlan199 manage
device(music)-> set interface vlan199 manage web
device(music)-> set interface vlan199 manage telnet
device(music)-> set interface vlan199 manage ping
device(music)-> save
device(music)-> exit
```

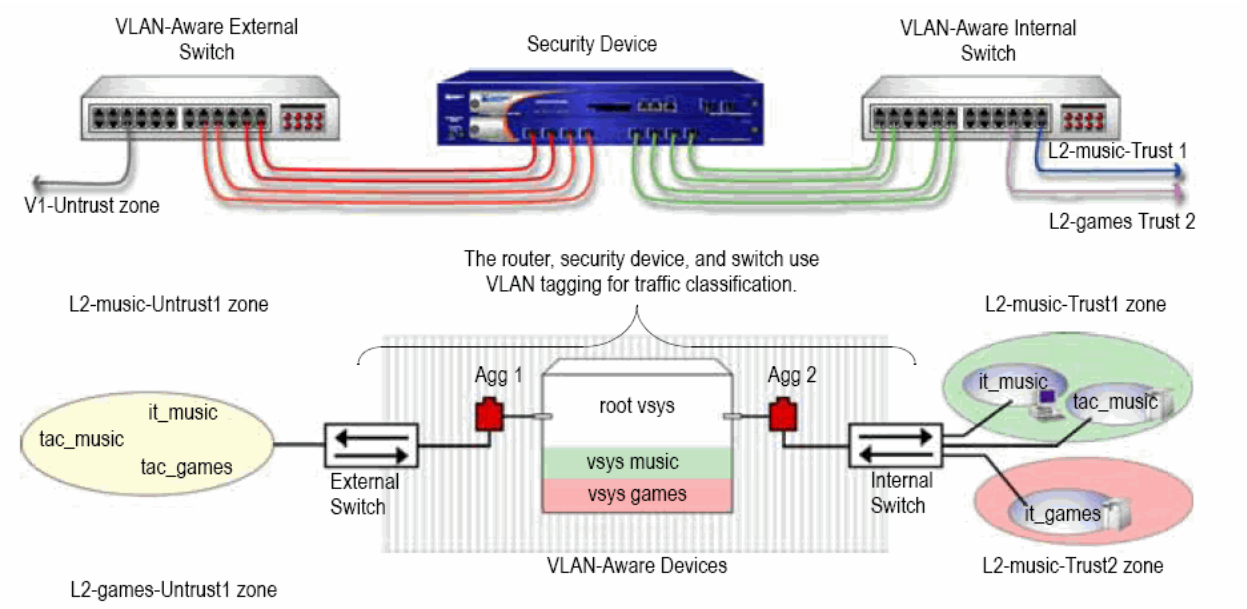
(Optional) Get VLAN Groups

```
device-> get vlan group it_music
```

Example 2: Configuring Two 4-Port Aggregates with Separate Untrust Zones

In this example, the security device is configured to support two virtual systems (vsys music and vsys games) in transparent mode. The two virtual systems have separate security zones. Vsys music consists of VLANs it_music and tac_music. Vsys games consists of VLAN it_games. Figure 429 on page 1732 shows how the security device secures the Trust and Untrust zones. You must import the VLANs to a vsys before they can be tagged.

Figure 429: Two 4-Port Aggregates with Separate Untrust Zones



Configuring two transparent mode virtual systems with two 4-port aggregates involves the following steps:

1. Set the aggregate ports at the root administration level.
2. Bind the interfaces to the aggregate ports.
3. Create the vsys named *music* with a vsys admin name and password.
4. Import (assign) VLAN tags from the root system to classify traffic for the vsys music.
5. Create the VLAN groups that contain the vsys tags for each port supported in the vsys music.
6. Create one Layer 2 zone for the Trust and Untrust interfaces in vsys music.
7. Bind aggregate ports to VLAN groups in the vsys music.

NOTE: Only the root administrator can configure virtual systems, VLAN tags, and Layer 2 zones or bind aggregate ports; however, both the root administrator and vsys administrator can set policies and management modules. The root administrator can access any vsys in the security device. The vsys administrator can only access the vsys that is assigned to the vsys in which the policies were created.

8. Set the IP address for the L2-music-Trust zone in the vsys music.
9. Configure the policies for vsys music. The policies configured in this example do the following:
   a. Permit HTTP traffic from the L2-music-Untrust zone to 10.0.1.200
   b. Deny all other traffic from the L2-music-Untrust zone to the L2-music-Trust zone
   c. Permit all traffic from the L2-music-Trust zone to the L2-music-Untrust zone
10. Create the management interface. A VLAN management interface is created within a vsys so that the vsys administrator can manage the vsys using a unique IP address and VLAN ID.
11. Create the vsys named games with a vsys admin name and password.
12. Import (assign) VLAN tags from the root system to classify traffic for the vsys games.
13. Create the VLAN groups that contain the vsys tags for each port supported in the vsys games.
14. Create one Layer 2 zone for the Trust and Untrust interfaces in the vsys games.
15. Bind aggregate ports to VLAN groups in the vsys games.
16. Configure the policies for the vsys games. The policies configured in this example do the following:
   a. Permit ftp traffic from the L2-games-Untrust zone to the L2-games-Trust zone
   b. Deny all other traffic from the L2-games-Untrust zone to the L2-games-Trust zone
   c. Permit all traffic from the L2-games-Trust zone to the L2-games-Untrust zone
17. Create the management interface.

NOTE: ScreenOS supports up to 500 virtual systems.

NOTE: In this example, each WebUI section lists only navigational paths, which lead to the pages necessary to configure the device. To see the specific parameters and values you need to set for any WebUI section, see the CLI section that follows it.

WebUI

1. **Set Aggregate Ports in Root**

   Network > Interfaces > (select **Aggregate** IF in the right-hand drop-down list) > New

2. **Bind Interfaces to Aggregate Ports**

   Network > Interfaces > Edit (for ethernet2/1)

   Network > Interfaces > Edit (for ethernet2/2)

   Network > Interfaces > Edit (for ethernet2/3)

   Network > Interfaces > Edit (for ethernet2/4)

   Network > Interfaces > Edit (for ethernet2/5)

   Network > Interfaces > Edit (for ethernet2/6)

   Network > Interfaces > Edit (for ethernet2/7)

   Network > Interfaces > Edit (for ethernet2/8)

3. **Create Vsys Music**

   Vsys > Configure > New

4. **Import VLANs**

   Vsys > Configure > Click Enter (for vsys_music)

   Network > Vlan > Import

5. **Create Group**

   Network > Vlan > Group > New

6. **Create Layer 2 Zones**

   Network > Zones > New

7. **Bind Aggregate Ports**

   Network > Vlan > Group > Edit (for it_music) > Port

   Network > Vlan > Group > Edit (for tac_music) > Port

8. **Set IP Address**

   Policy > Policy Elements > Addresses > List > V1-Untrust (select in the drop-down list) > New

   Policy > Policy Elements > Addresses > List > From: L2-music-Untrust To: L2-music-Trust > New

Policy > Policy Elements > Addresses > List > From: L2-music-Untrust To: L2-music-Trust > New

Policy > Policy Elements > Addresses > List > From: L2-music-Untrust To: L2-music-Trust > New

Policy > Policy Elements > Addresses > List > From: L2-music-Trust To: L2-music-Untrust > New

9. **Policies**

   Policies (From: L2-music-Untrust, To: L2-music-Trust) > New

   Policies (From: L2-music-Untrust, To: L2-music-Trust) > New

   Policies (From: L2-music-Untrust, To: L2-music-Trust) > New

   Policies (From: L2-music-Trust, To: L2-music-Untrust) > New

10. **Create Management Interface**

    Network > Interfaces > (select VLAN in the right side drop-down list) > New

11. **Create Vsys Games**

    Vsys > Configure > New

12. **Import VLAN**

    Vsys > Configure > Click Enter (for vsys_games)

    Network > Vlan > Import

13. **Create Groups**

    Network > Vlan > Group > New

14. **Create Layer 2 Zones**

    Network > Zones > New

15. **Bind Aggregate Ports**

    Network > Vlan > Group > Edit (for games) > Port

16. **Policies**

    Policies (From: L2-games-Untrust, To: L2-games-Trust) > New

    Policies (From: L2-games-Untrust, To: L2-games-Trust) > New

    Policies (From: L2-games-Trust, To: L2-games-Untrust) > New

17. **Create Management Interface**

    Network > Interfaces > (select VLAN in the right side drop-down list) > New

CLI

1. **Set Aggregate Ports in Root**

```
device-> set interface aggregate1 zone null
device-> set interface aggregate2 zone null
```

2. **Bind Interfaces to Aggregate Ports**

```
device-> set interface ethernet2/1 aggregate aggregate1
device-> set interface ethernet2/2 aggregate aggregate1
device-> set interface ethernet2/3 aggregate aggregate1
device-> set interface ethernet2/4 aggregate aggregate1
device-> set interface ethernet2/5 aggregate aggregate2
device-> set interface ethernet2/6 aggregate aggregate2
device-> set interface ethernet2/7 aggregate aggregate2
device-> set interface ethernet2/8 aggregate aggregate2
```

3. **Create Vsys Music**

```
device-> set vsys music
device(music)-> set admin name vsys_music
device(music)-> set admin password xyz
device(music)-> save
```

4. **Import VLAN Tag**

```
device(music)-> set vlan import 100 199
device(music)-> set vlan import 1033
device(music)-> set vlan import 1133
```

5. **Create VLAN Groups**

```
device(music)-> set vlan group name it_music
device(music)-> set vlan group it_music 100 199
device(music)-> set vlan group name tac_music
device(music)-> set vlan group tac_music 1033
device(music)-> set vlan group tac_music 1133
```

6. **Create Layer 2 Zone**

```
device(music)-> set zone name L2-music-Trust L2
device(music)-> set zone name L2-music-Untrust L2
```

7. **Bind Aggregate Ports**

```
device(music)-> set vlan port aggregate2 group it_music zone L2-music-Trust
device(music)-> set vlan port aggregate1 group it_music zone L2-music-Untrust
device(music)-> set vlan port aggregate2 group tac_music zone L2-music-Trust
device(music)-> set vlan port aggregate1 group tac_music zone L2-music-Untrust
```

8. **Set IP Addresses**

```
device(music)-> set address L2-music-Trust 10.0.1.200 10.0.1.200
255.255.255.0
```

9. **Configure Policies**

```
device(music)-> set policy from L2-music-Untrust to L2-music-Trust any
10.0.1.200 http permit
device(music)-> set policy from L2-music-Untrust to L2-music-Trust any any http
 permit
device(music)-> set policy from L2-music-Untrust to L2-music-Trust any any any
 deny
device(music)-> set policy from L2-music-Trust to L2-music-Untrust any any any
 permit
```

10. **Create Management Interface**

```
device(music)-> set interface vlan1033 zone vlan
device(music)-> set interface vlan1033 ip 1.0.0.33/24
device(music)-> set interface vlan199 zone vlan
device(music)-> set interface vlan199 ip 1.0.1.199/24
device(music)-> unset interface vlan199 manage
device(music)-> set interface vlan199 manage ping
device(music)-> set interface vlan199 manage https
device(music)-> set interface vlan199 manage telnet
device(music)-> save
device(music)-> exit
```

(Optional) Get VLAN Groups

```
device-> get vlan group it_music
device-> get vlan group tac_music
```

11. **Create Vsys Games**

```
device-> set vsys games
device(games)-> set admin name vsys_games
device(games)-> set admin password abc
device(games)-> save
```

12. **Import VLAN Tag**

```
device(games)-> set vlan import 200 250
```

13. **Create VLAN Groups**

```
device(games)-> set vlan group name games
device(games)-> set vlan group games 200 250
```

14. **Create Layer 2 Zone**

```
device(games)-> set zone name L2-games-Trust L2
device(games)-> set zone name L2-games-Untrust L2
```

15. **Bind Aggregate Ports**

```
device(games)-> set vlan port aggregate2 group games zone L2-games-trust
device(games)-> set vlan port aggregate1 group games zone L2-games-Untrust
```

16. **Configure Policies**

```
device(games)-> set policy from L2-games-Untrust to L2-games-Trust any any ftp
 permit
device(games)-> set policy from L2-games-Untrust to L2-games-Trust any any
any deny
device(games)-> set policy from L2-games-Trust to L2-games-Untrust any any
any permit
```

17. **Create Management Interface**

```
device(games)-> set interface vlan300 zone vlan
device(games)-> set interface vlan300 ip 1.0.0.20/24
device(games)-> unset interface vlan300 manage
device(games)-> set interface vlan300 manage web
device(games)-> set interface vlan300 manage telnet
device(games)-> set interface vlan300 manage ping
device(games)-> save
device(games)-> exit
```

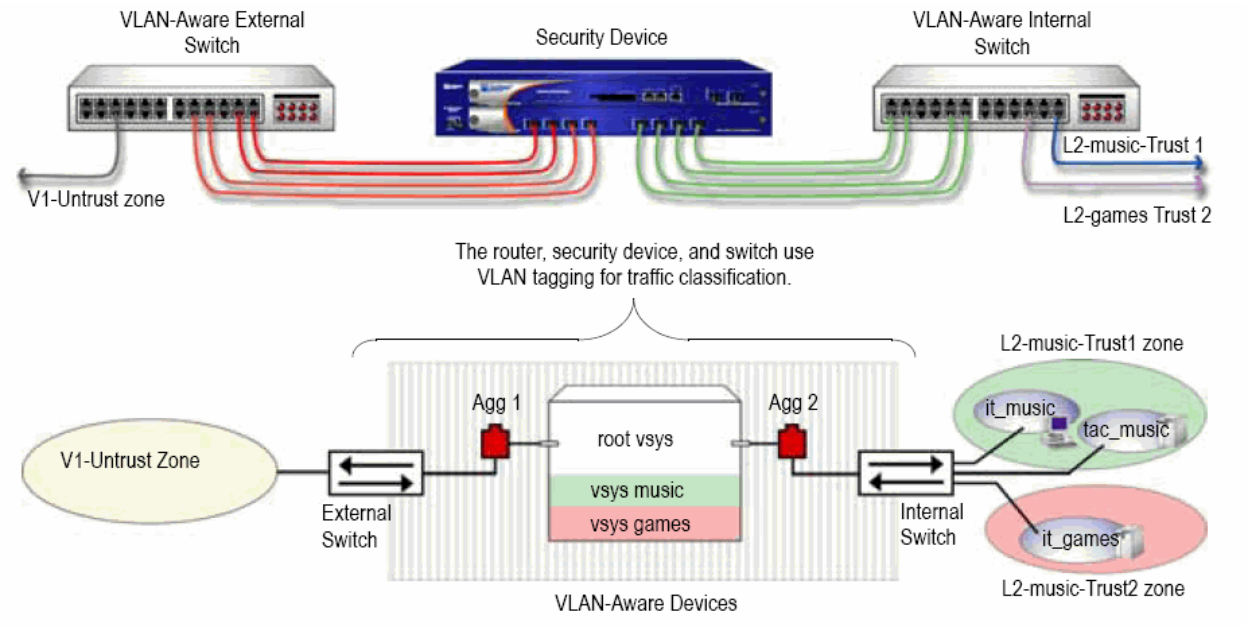
(Optional) Get VLAN Groups

```
device-> get vlan group games
```

Example 3: Configuring Two 4-Port Aggregates that Share One Untrusted Zone

In this example, the security device is configured to support two virtual systems (vsys music and vsys games) in transparent mode. The two virtual systems share the Untrust zone with the root system. Vsys music consists of VLANs it_music and tac_music. Vsys games consists of VLAN it_games. Figure 430 on page 1739 shows how the security device secures the Trust and Untrust zones. You must import the VLANs to a vsys before they can be tagged.

Figure 430: Two 4-Port Aggregates that Share One Untrusted Zone



Configuring two transparent mode virtual systems with two aggregate ports and a shared Untrust zone involves the following steps:

1. Set the aggregate ports at the root administration level.
2. Bind the interfaces to the aggregate ports.
3. Create the vsys named music with a vsys admin name and password.
4. Import VLAN tags from the root system to classify traffic for the vsys music.
5. Create the VLAN groups that contain the vsys tags for each port supported in the vsys music.
6. Create Layer 2 zones for the Trust interface for the vsys music.
7. Bind aggregate ports to VLAN groups in the vsys music Trust and Untrust zones.

NOTE: Only the root administrator can configure virtual systems, VLAN tags, and Layer 2 zones or bind aggregate ports; however, both the root administrator and vsys administrator can set policies and management modules. The root administrator can access any vsys in the security device. The vsys administrator can only access the vsys that is assigned to the vsys in which the policies were created.

8. Set the IP address for each zone in the vsys music.
9. Configure the policies for the vsys music. The policies configured in this example do the following:
   a. Permit all traffic from 10.0.1.200 to 10.0.1.100
   b. Permit all traffic from 10.0.1.201 to 10.0.1.101

c. Deny all traffic from the V1-Untrust zone to the L2-music-Trust zone

10. Create the management interface. A VLAN management interface is created within a vsys so that the vsys administrator can manage the vsys using a unique IP address and VLAN ID.
11. Create the vsys named games with a vsys admin name and password.
12. Import VLAN tags from the root system to classify traffic for the vsys games.
13. Create the VLAN groups that contain the vsys tags for each port supported in the vsys games.
14. Create Layer 2 zones for the Trust interface for the vsys games.
15. Bind aggregate ports to VLAN groups in the vsys games.
16. Set the IP address for each zone in the vsys games.
17. Configure the policies for the vsys games. The policies configured in this example do the following:
    a. Permit all traffic from 20.0.1.200 to 20.0.1.100
    b. Permit all traffic from 20.0.1.201 to 20.0.1.101
    c. Deny all traffic from the V1-Untrust zone to the L2-games-Trust1 zone
18. Create the management interface.

NOTE: ScreenOS 5 supports up to 500 virtual systems.

NOTE: In this example, each WebUI section lists only navigational paths, which lead to the pages necessary to configure the device. To see the specific parameters and values you need to set for any WebUI section, see the CLI section that follows it.

WebUI

1. **Set Aggregate Ports in Root**

   Network > Interfaces > (select **Aggregate IF** in the drop-down list four times)

2. **Bind Interfaces to Aggregate Ports**

   Network > Interfaces > Edit (for ethernet2/1 to aggregate1)

   Network > Interfaces > Edit (for ethernet2/2 to aggregate1)

   Network > Interfaces > Edit (for ethernet2/3 to aggregate1)

   Network > Interfaces > Edit (for ethernet2/4 to aggregate1)

   Network > Interfaces > Edit (for ethernet2/5 to aggregate2)

Network > Interfaces > Edit (for ethernet2/6 to aggregate2)

Network > Interfaces > Edit (for ethernet2/7 to aggregate2)

Network > Interfaces > Edit (for ethernet2/8 to aggregate2)

3. **Create Vsys Music**

   Vsys > Configure > New

4. **Import VLANs**

   Vsys > Configure > Click Enter (for vsys_music)

   Network > Vlan > Import

5. **Create Group**

   Network > Vlan > Group > New: 100-199

   Network > Vlan > Group > New: 1033-1033

   Network > Vlan > Group > New: 1133-1133

6. **Create Layer 2 Zones**

   Network > Zones > New: L2-music-Trust1

   Network > Zones > New: L2-music-Trust2

7. **Bind Aggregate Ports**

   Network > Vlan > Group > Edit (for music) > Port

8. **Set IP Address**

   Policy > Policy Elements > Addresses > List > (select V1-Untrust in the drop-down list) > New

   Policy > Policy Elements > Addresses > List > (select L2-music-Trust1 in the drop-down list)

   Policy > Policy Elements > Addresses > List > (select L2-music-Trust2 in the drop-down list

9. **Policies**

   Policies (From: L2-music-Trust1, To: V1-Untrust) > New

   Policies (From: L2-music-Trust2, To: V1-Untrust) > New

   Policies (From: V1-Untrust, To: L2-music-Trust2) > New

   Policies (From: V1-Untrust, To: L2-music-Trust1) > New

10. **Create Management Interface**

    Network > Interfaces > (select VLAN in the right side drop-down list) > New

Network > Interfaces > (select VLAN in the right side drop-down list) > New

11. **Create Vsys Games**

    Vsys > Configure > New

12. **Import VLANs**

    Vsys > Configure > Click Enter (for vsys_games)

    Network > Vlan > Import

13. **Create Group**

    Network > Vlan > Group > New: 200-299

    Network > Vlan > Group > New: 50-50

14. **Create Layer 2 Zones**

    Network > Zones > New: L2-games-Trust1

    Network > Zones > New: L2-games-Trust2

15. **Bind Aggregate Ports**

    Network > Vlan > Group > Edit (for games) > Port

16. **Set IP Addresses**

    Policy > Policy Elements > Addresses > List > (select V1-Untrust in the right-hand drop-down list) > New

17. **Policies**

    Policies (From: L2-games-Trust1, To: V1-Untrust) > New

18. **Create Management Interface**

    Network > Interfaces > (select VLAN in the drop-down list) > New

CLI

1. **Set Aggregate Ports in Root**

```
device-> set interface aggregate1 zone null
device-> set interface aggregate2 zone null
```

2. **Bind Interfaces to Aggregate Ports**

```
device-> set interface ethernet2/1 aggregate aggregate1
device-> set interface ethernet2/2 aggregate aggregate1
device-> set interface ethernet2/3 aggregate aggregate1
device-> set interface ethernet2/4 aggregate aggregate1
device-> set interface ethernet2/5 aggregate aggregate2
device-> set interface ethernet2/6 aggregate aggregate2
device-> set interface ethernet2/7 aggregate aggregate2
device-> set interface ethernet2/8 aggregate aggregate2
```

3. **Create Vsys Music**

```
device-> set vsys music
device(music)-> set admin name vsys_music
device(music)-> set admin password xyz
device(music)-> save
```

4. **Import VLAN Tag**

```
device(music)-> set vlan import 100 199
device(music)-> set vlan import 1033
```

5. **Create VLAN Groups**

```
device(music)-> set vlan group name music
device(music)-> set vlan group music 100 199
device(music)-> set vlan group music 1033
```

6. **Create Layer 2 Zone**

```
device(music)-> set zone name L2-music-Trust1 L2
device(music)-> set zone name L2-music-Trust2 L2
```

7. **Bind Aggregate Ports**

```
device(music)-> set vlan port aggregate2 group music zone L2-music-Trust1
device(music)-> set vlan port aggregate2 group music zone L2-music-Trust2
device(music)-> set vlan port aggregate1 group music zone V1-Untrust
```

8. **Set IP Address**

```
device(music)-> set address V1-Untrust 10.0.1.100 10.0.1.100
255.255.255.255
device(music)-> set address V1-Untrust 10.0.1.101 10.0.1.101
255.255.255.255
device(music)-> set address L2-music-Trust1 10.0.1.200 10.0.1.200
255.255.255.255
device(music)-> set address L2-music-Trust2 10.0.1.201 10.0.1.201
255.255.255.255
```

9. **Configure Policies**

```
device(music)-> set policy id 1 from L2-music-Trust1 to V1-Untrust 10.0.1.200
10.0.1.100 any permit
device(music)-> set policy id 2 from L2-music-Trust2 to V1-Untrust 10.0.1.201
10.0.1.101 any permit
device(music)-> set policy id 3 from V1-Untrust to L2-music-Trust2 any any any
deny
device(music)-> set policy id 4 from V1-Untrust to L2-music-Trust1 any any any
deny
```

10. **Create Management Interface**

```
device(music)-> set interface vlan1033 zone vlan
device(music)-> set interface vlan1033 ip 1.0.0.33/24
device(music)-> set interface vlan199 zone vlan
device(music)-> set interface vlan199 ip 1.0.1.199/24
```

```
device(music)-> unset interface vlan199 manage
device(music)-> set interface vlan199 manage web
device(music)-> set interface vlan199 manage telnet
device(music)-> set interface vlan199 manage ping
device(music)-> save
device(music)-> exit
```

(Optional) Get VLAN Groups

```
device-> get vlan group music
```

11. **Create Vsys Games**

```
device-> set vsys games
device(games)-> set admin name vsys_games
device(games)-> set admin password abc
device(games)-> save
```

12. **Import VLAN Tag**

```
device(games)-> set vlan import 200 299
device(games)-> set vlan import 50
```

13. **Create VLAN Groups**

```
device(games)-> set vlan group name games
device(games)-> set vlan group games 200 299
device(games)-> set vlan group games 50
```

14. **Create Layer 2 Zone**

```
device(games)-> set zone name L2-games-Trust1 L2
device(games)-> set zone name L2-games-Trust2 L2
```

15. **Bind Aggregate Ports**

```
device(games)-> set vlan port aggregate2 group games zone L2-games-Trust1
device(games)-> set vlan port aggregate2 group games zone L2-games-Trust2
device(games)-> set vlan port aggregate1 group games zone V1-Untrust
```

16. **Set IP Address**

```
device(games)-> set address V1-Untrust 20.0.1.100 20.0.1.100
255.255.255.255
device(games)-> set address V1-Untrust 20.0.1.101 20.0.1.101
255.255.255.255
device(games)-> set address L2-games-Trust1 20.0.1.200 20.0.1.200
255.255.255.255
device(games)-> set address L2-games-Trust2 20.0.1.201 20.0.1.201
255.255.255.255
```

17. **Configure Policies**

```
device(games)-> set policy id 1 from L2-games-Trust1 to V1-Untrust 20.0.1.200
 20.0.1.100 any permit
device(games)-> set policy id 2 from L2-games-Trust1 to V1-Untrust 20.0.1.201
 20.0.1.101 any permit
```

```
device(games)-> set policy id 3 from V1-Untrust to L2-games-Trust1 any any any
 deny
```

18. **Create Management Interface**

```
device(games)-> set interface vlan300 zone vlan
device(games)-> set interface vlan300 ip 1.0.0.20/24
device(games)-> unset interface vlan300 manage
device(games)-> set interface vlan300 manage web
device(games)-> set interface vlan300 manage telnet
device(games)-> set interface vlan300 manage ping
device(games)-> save
device(games)-> exit
```

(Optional) Get VLAN Groups

```
device-> get vlan group games
```

Defining Subinterfaces and VLAN Tags

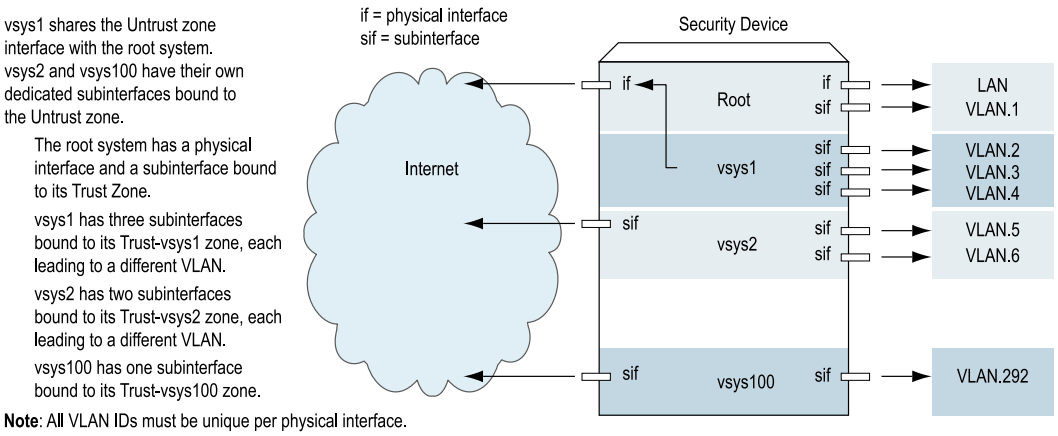
The Trust-*vsys_name* zone subinterface links a vsys to its internal VLAN. The Untrust zone subinterface links a vsys to the public WAN, usually the Internet. A subinterface has the following attributes:

- A unique VLAN ID (from 1 to 4095)
- A public or private IP address (the IP address is private by default)
- A netmask for a class A, B, or C subnet
- An associated VLAN

NOTE: For information about public and private IP addresses, see *"Public IP Addresses" on page 63* and *"Private IP Addresses" on page 64*.

A vsys can have a single Untrust zone subinterface and multiple Trust-*vsys_name* zone subinterfaces. If a virtual system does not have its own Untrust zone subinterface, it shares the root level Untrust zone interface. Security devices also support subinterfaces, VLANs at the root level, and IEEE 802.1Q-compliant VLAN tags.

Figure 431: VLAN Subinterfaces



A VLAN tag is an added bit in the Ethernet frame header that indicates membership in a particular VLAN. By binding a VLAN to a vsys, the tag also determines to which vsys a frame belongs, and consequently, which policy is applied to that frame. If a VLAN is not bound to a vsys, policies set in the root system of the security device are applied to the frame.

A root-level administrator can create a VLAN, assign members to it, and bind it to a vsys. (The assigning of members to a VLAN can be done by several methods—protocol type, MAC address, port number—and is beyond the scope of this document.) The vsys admin, if there is one, then manages the vsys through the creation of addresses, users, services, VPNs, and policies. If there is no vsys admin, then a root-level administrator performs these tasks.

NOTE: If the root-level admin does not associate a VLAN to a vsys, the VLAN operates within the root system of the security device.
All subnets in a vsys must be disjointed; that is, there must be no overlapping IP addresses among the subnets in the same vsys. For example: Subinterface1 – 10.2.2.1 255.255.255.0 and Subinterface2 – 10.2.3.1 255.255.255.0 are disjointed and link to acceptable subnets.
However, subnets with the following subinterfaces overlap and are unacceptable within the same vsys: subinterface1 – 10.2.2.1 255.255.0.0 and subinterface2 – 10.2.3.1 255.255.0.0.
The address ranges of subnets in different vsys can overlap.

In this example, you define subinterfaces and VLAN tags for the three virtual systems that you created in “Creating a Virtual System Object and Admin” on page 1681—vsys1, vsys2, and vsys3. The first two subinterfaces are for two private virtual systems operating in NAT mode, and the third subinterface is for a public virtual system operating in route mode. The subinterfaces are 10.1.1.1/24, 10.2.2.1/24, and 1.3.3.1/24. You create all three subinterfaces on ethernet3/2.

All three virtual systems share the Untrust zone and its interface (ethernet1/1; 1.1.1.1/24) with the root system. The Untrust zone is in the untrust-vr routing domain.

WebUI

1. **Vsys1 Subinterface and VLAN Tag**

   Vsys > Configure > Click **Enter** (for vsys1).

   Network > Interfaces > New Sub-IF (for ethernet3/2): Enter the following, then click **OK**:

   Interface Name: ethernet3/2.1
   Zone Name: Trust-vsys1
   IP Address / Netmask: 10.1.1.1/24
   VLAN Tag: 1

NOTE: You can define virtual systems to operate in route mode or NAT mode. The default is NAT mode, and it is unnecessary to specify NAT when creating the first two subinterfaces in this example.

2. **Vsys2 Subinterface and VLAN Tag**

   Vsys > Configure > Click **Enter** (for vsys2).

   Network > Interfaces > New Sub-IF (for ethernet3/2): Enter the following, then click **OK**:

   Interface Name: ethernet3/2.2
   Zone Name: Trust-vsys2
   IP Address / Netmask: 10.2.2.1/24
   VLAN Tag: 2

3. **Vsys3 Subinterface and VLAN Tag**

   Vsys > Configure > Click **Enter** (for vsys3).

   Network > Interfaces > New Sub-IF (for ethernet3/2): Enter the following, then click **Apply**:

   Interface Name: ethernet3/2.3
   Zone Name: Trust-vsys3
   IP Address / Netmask: 1.3.3.1/24
   VLAN Tag: 3

   Select **Interface Mode: Route**, then click **OK**.

   Click **Exit Vsys** to return to the root level.

CLI

1. **Vsys1 Subinterface and VLAN Tag**

```
device-> enter vsys vsys1
device(vsys1)-> set interface ethernet3/2.1 zone trust-vsys1
device(vsys1)-> set interface ethernet3/2.1 ip 10.1.1.1/24 tag 1
device(vsys1)-> save
device(vsys1)-> exit
```

NOTE: You can define virtual systems to operate in route mode or NAT mode. The default is NAT mode, and it is unnecessary to specify NAT when creating the first two subinterfaces in this example.

2. **Vsys2 Subinterface and VLAN Tag**

```
device-> enter vsys vsys2
device(vsys2)-> set interface ethernet3/2.2 zone trust-vsys2
device(vsys2)-> set interface ethernet3/2.2 ip 10.2.2.1/24 tag 2
device(vsys2)-> save
device(vsys2)-> exit
```

3. **Vsys3 Subinterface and VLAN Tag**

```
device-> enter vsys vsys3
device(vsys3)-> set interface ethernet3/2.3 zone trust-vsys3
device(vsys3)-> set interface ethernet3/2.3 ip 1.3.3.1/24 tag 3
device(vsys3)-> set interface ethernet3/2.3 route
device(vsys3)-> save
device(vsys3)-> exit
```

Communicating Between Virtual Systems

The members of a VLAN within a vsys have unrestricted communication access with each other. The VLAN members of different virtual systems cannot communicate with one another unless the participating vsys administrators specifically configure policies allowing the members of their respective systems to do so.

Traffic between root-level VLANs operates within the parameters set by root-level policies. Traffic between virtual system VLANs operates within the parameters set by the participating virtual system policies. The security device passes only traffic allowed to leave the originating virtual system and allowed to enter the destination virtual system. In other words, the vsys admins of both virtual systems must set policies allowing the traffic to flow in the appropriate direction—outgoing and incoming.

NOTE: Policies set in the root system and in virtual systems do not affect each other.

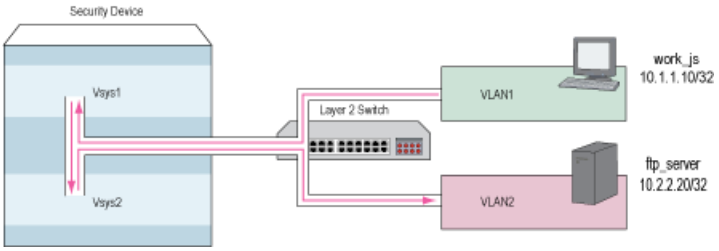
In this example configuration shown in Figure 432 on page 1749, the admins for vsys1 and vsys2—see "Defining Subinterfaces and VLAN Tags" on page 1745—set up policies to enable traffic between a workstation (work_js with the IP address 10.1.1.10/32) in VLAN1 and a server (ftp_server with the IP address 10.2.2.20/32) in VLAN2. The connection is possible if the following two conditions are met:

- The vsys admin for vsys1 has set a policy permitting traffic from the workstation in Trust-vsys1 to the server in its Untrust zone.
- The vsys admin for vsys2 has set a policy permitting traffic from the workstation in its Untrust zone to the server in Trust-vsys2.

The network device in front of the internal interface on the security device is a Layer 2 switch. This forces traffic from VLAN1 going to VLAN2 to go through the switch to the security device for Layer 3 routing. If the network device were a Layer 3 router, traffic between VLAN1 and VLAN2 could pass through the router, bypassing all policies set on the security device.

The vsys1 and vsys2 admins also set up the appropriate routes. The shared Untrust zone is in the untrust-vr and the Trust zones in vsys1 and vsys2.

Figure 432: InterVsys Communication



WebUI

1. **Vsys1**

   Addresses

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: work_js
   IP Address/Domain Name:
   IP/Netmask: (select), 10.1.1.10/32
   Zone: Trust-vsys1

   Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

   Address Name: ftp_server
   IP Address/Domain Name:
   IP/Netmask: (select), 10.2.2.20/32
   Zone: Untrust

   Routes

   Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.1.1.0/24
Next Hop Virtual Router Name: (select); vsys1-vr

Policy

Network > Routing > Routing Entries > vsys1-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Next Hop Virtual Router Name: (select); untrust-vr

Policies > (From: Trust-vsys1, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), work_js
Destination Address:
  Address Book Entry: (select), ftp_server
Service: FTP-Get
Action: Permit

2. **Vsys2**

Addresses

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: ftp_server
IP Address/Domain Name:
  IP/Netmask: (select), 10.2.2.2/32
Zone: Trust-vsys2

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: work_js
IP Address/Domain Name:
  IP/Netmask: (select), 10.1.1.10/32
Zone: Untrust

Routes

Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Next Hop Virtual Router Name: (select); vsys2-vr

Network > Routing > Routing Entries > vsys2-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Next Hop Virtual Router Name: (select); untrust-vr

Policy

Policies > (From: Untrust, To: Trust-vsys2) New: Enter the following, then click **OK**:

Source Address:
  Address Book Entry: (select), work_js
Destination Address:
  Address Book Entry: (select), ftp_server
Service: FTP-Get
Action: Permit

CLI

1. **Vsys1**

   Addresses

   ```
   set address trust-vsys1 work_js 10.1.1.10/32
   set address untrust ftp_server 10.2.2.20/32
   ```

   Routes

   ```
   set vrouter untrust-vr route 10.1.1.0/24 vrouter vsys1-vr
   set vrouter vsys1-vr route 0.0.0.0/0 vrouter untrust-vr
   ```

   Policy

   ```
   set policy from trust-vsys1 to untrust work_js ftp_server ftp-get permit
   save
   ```

2. **Vsys2**

   Addresses

   ```
   set address trust-vsys2 ftp_server 10.2.2.20/32
   set address untrust work_js 10.1.1.10/32
   ```

   Routes

   ```
   set vrouter untrust-vr route 10.2.2.0/24 vrouter vsys2-vr
   set vrouter vsys2-vr route 0.0.0.0/0 vrouter untrust-vr
   ```

   Policy

   ```
   set policy from untrust to trust-vsys2 work_js ftp_server ftp-get permit
   save
   ```

   Network > Zones > Edit (for Internal): Select the IP Classification check box, then click **OK**.

VLAN Retagging

VLAN retagging provides a way to selectively screen VLAN traffic. You place a security device in parallel with your Layer 2 switch (see Figure 433 on page 1753) and configure the switch to direct to the security device only traffic from VLANs you want screened. Traffic to and from your other VLANs continues to pass directly through the switch, thus avoiding any impact to throughput that might be caused by passing all VLAN traffic through the security device.

NOTE: The current release of ScreenOS supports VLAN retagging on ISG platforms.

VLAN retagging requires that retagged traffic be from VLANs with different IDs, that is, you cannot retag VLAN traffic from VLAN 10 to another VLAN with the same ID.

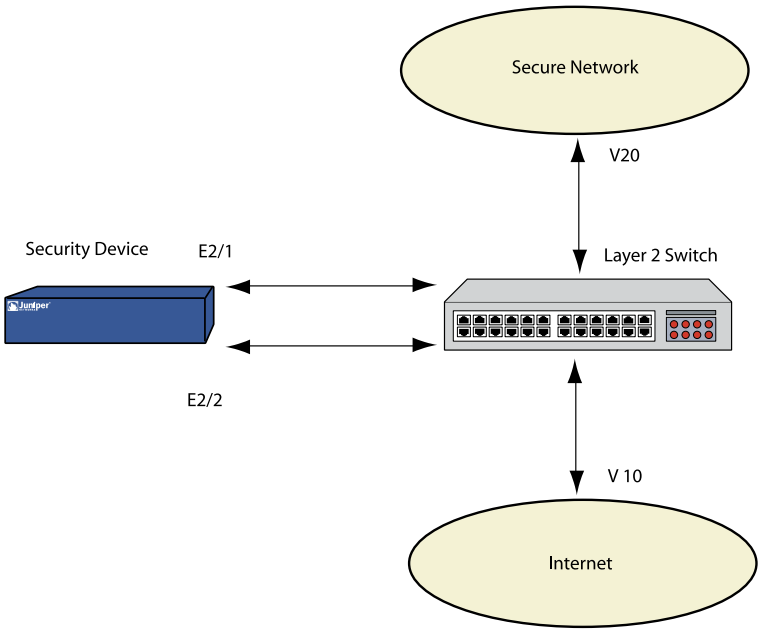
You configure VLAN retagging on the security device by creating a bidirectional VLAN retagging object and specifying the two VLANs for which you want it to screen traffic. (You must also bind it to an interface and create a policy.) For example, the following command creates a VLAN retagging object called secure_vlan that retags traffic from VLAN 10 to 20 and from VLAN 20 to 10:

```
set vlan retag name secure_vlan 10 20
```

The security device stores this retagging pair in a hash table and references it when it receives traffic from either of those VLANs. You must also assign the appropriate VLAN tags to the ports on your Layer 2 switch that connect to the security device.

Figure 433 on page 1753 illustrates this scenario.

Figure 433: VLAN Retagging Operation



Configuring VLAN Retagging

To configure a VLAN retagging pair from the WebUI,

Network > VLAN > Retagging

Click New, and enter a name for the pair, for example, **From_10_to_20.** In the From VLAN box, enter **10.** In the To VLAN box, enter **20,** then click OK. Then create a corresponding VLAN pair from VLAN 20 to VLAN 10, for example, **From_20_to_10.**

To bind this VLAN retagging pair to an interface,

Network > VLAN > Retagging Binding

Click New. In the Interface box, enter the name of the interface to which you want to bind the VLAN retagging pair. In the Binding box, select the VLAN pair name (in this case, **From_10_to_20),** then click the Ingress check box in the **Direction** field then click **OK.** Repeat for the egress pair (in this case, **From_20_to_10**)

Use the following CLI command to create a VLAN retagging pair:

set vlan retagging name *retagging-pair-name from-vlan to-vlan*

Use the following command to bind a VLAN retagging pair to an interface:

set vlan port *interface_name retag retagging-pair-name*

Example

In this example, you create a VLAN group called v10 and assign the incoming interface to the zone V1-Untrust and the outgoing interface to zone V1-Trust. You then create a security policy permitting all traffic to and from those zones. Finally, you create a bidirectional VLAN retagging object called secure_vlan to screen traffic between VLAN 10 and VLAN 20, and bind it to the V1-Trust interface.

WebUI

This example shows a VLAN retagging configuration at the root level. To configure an existing vsys, you must first enter the vsys. To enter the vsys, go to: Vsys > Configure > Enter (vsys name), then configuring VLAN retagging as follows:

Network > VLAN > Group > New: Enter the following, then click **Add:**

VLAN Group Name: v10
Start: 10
End: 10

Network > VLAN > Group > Edit (for group v10) > Port: Select the following, then click Add:

Port: (select ethernet2/1)
Zone: (select V1-Trust)

Network > VLAN > Group > Edit (for group v10) > Port: Select the following, then click **Add:**

Port: (select ethernet2/2)
Zone: (select V1-Untrust)

Network > VLAN > Retagging > New: Enter the following, then click **OK:**

Name: secure_vlan
From Vlan: 10
To Vlan: 20

Network > VLAN > Retagging Bind > New: Enter the following, then click **OK:**

Interface Name: (select) ethernet2/1
Binding: (select) secure_vlan

NOTE: Although the WebUI indicates a **From Vlan** and a **To Vlan,** the above configuration provides bidirectional retagging, that is, traffic from VLAN 10 is retagged with ID 20, and traffic from VLAN 20 is retagged with ID 10.

CLI

This example shows a VLAN retagging configuration from the root level. Use the following command to enter an existing vsys from the root level: **enter vsys** *name_str*, then configure VLAN retagging:

```
set vlan group name v10
set vlan group v10 10 10
set vlan port eth2/1 group v10 zone v1-trust
set vlan port eth2/2 group v10 zone v1-untrust
set policy from v1-trust to v1-untrust any any any permit
set vlan retag name secure_vlan 10 20
set vlan retag name secure_vlan 10 20 untag
set vlan port eth2/1 retag secure_vlan
```

NOTE: Using the untag option, you can remove the VLAN ID from a packet frame. This option sets the VLAN ID to zero in the output. This option is supported only on ISG platforms.

To view VLAN retagging information, use the **get vlan retag name [name** | **all]** command.

NOTE: To use a VLAN in an existing vsys, you must import the VLAN ID from the root vsys before you create a VLAN group. To do this in the above example, use the command **set vlan import 10 10.** In the WebUI, enter the vsys and then go to Network > Vlan > Import.

Chapter 55

IP-Based Traffic Classification

This chapter explains IP-based traffic classification for virtual systems. It contains the following sections:

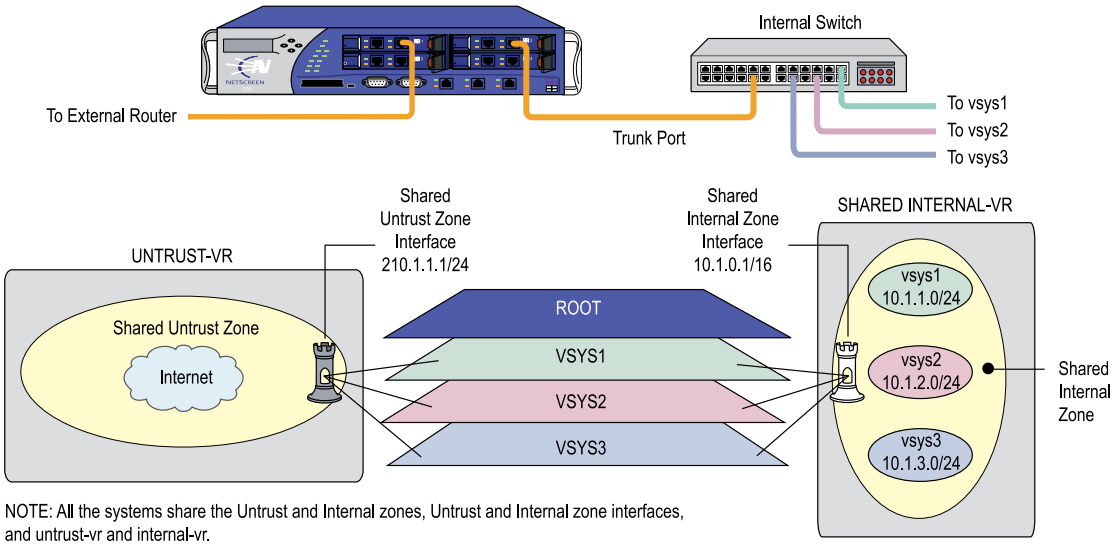
Overview

Figure 434 on page 1758 shows how IP-based traffic classification allows you to use virtual systems without VLANs. Instead of VLAN tags, the security device uses IP addresses to sort traffic, associating a subnet or range of IP addresses with a particular system—root or vsys. Using IP-based traffic classification exclusively to sort traffic, all systems share the following:

- The untrust-vr and a user-defined internal-vr
- The Untrust zone and a user-defined internal zone
- An Untrust zone interface and a user-defined internal zone interface

NOTE: Even when using VLAN-based traffic classification for internal traffic, for external traffic all systems use the shared Untrust zone—and, unless a system has a dedicated interface, a shared Untrust zone interface. Using a shared interface on one side and a dedicated interface (with VLAN tagging) on the other constitutes a hybrid approach. VLAN-based and IP-based traffic classification can coexist within the same system or among different systems simultaneously.

Figure 434: IP-Based Traffic Classification



Managing Inter-Vsys Traffic with a Shared DMZ Zone

Virtual systems across different zones generally use a shared Untrust zone for communication. However, inter-vsys traffic through a shared Untrust zone is often interrupted by external traffic and overlapping IP addresses. To overcome such traffic interference in the shared Untrust zone, you can use a shared DMZ zone created at the root level. Each shared DMZ zone that the root admin creates is automatically assigned to a nonsharable virtual router (VR). The root admin also determines to which shared DMZ zone a particular vsys should be subscribed. A shared DMZ zone is shared only with the virtual systems that are subscribed to it. However, each vsys can be subscribed to only one shared DMZ zone.

WebUI

Network > Zones > New: Enter the following, then click **OK:**

Zone name: smdz
Virtual Router Name: sdmz_vr
Zone Type: Shared-DMZ-Zone (select)

CLI

1. **Creating a Shared-DMZ Zone**

```
set zone name zone shared-dmz
save
```

2. **Subscribing a Vsys to a Shared-DMZ Zone**

```
set vsys name_str shared-dmz zone
save
```

NOTE: A shared DMZ zone works only on a security device running in NAT/route mode and cannot be bound to any interface other than the loopback interface. However, the default interface for the shared DMZ zone is Null.

Designating an IP Range to the Root System

To designate a subnet or range of IP addresses to the root system or to a previously created virtual system, you must do either of the following at the root level:

WebUI

Network > Zones > Edit (for zone) > IP Classification: Enter the following, then click **OK**:

System: (select **root** or **vsys_name_str**)
Address Type: (select **Subnet** and enter **ip_addr/mask**, or select **Range** and enter **ip_addr1** – **ip_addr2**)

CLI

set zone zone ip-classification net *ip_addr/mask* { root | vsys *name_str* }
set zone zone ip-classification range *ip_addr1-ip_addr2* { root | vsys *name_str* }

Because IP-based traffic classification requires the use of a shared security zone, virtual systems cannot use overlapping internal IP addresses, as is possible with VLAN-based traffic classification. Also, because all the systems share the same internal interface, the operational mode for that interface must be either NAT or route mode; you cannot mix NAT and route modes for different systems. In this regard, the addressing scheme of an IP-based approach is not as flexible as that allowed by the more commonly used VLAN-based approach.

Sharing virtual routers, security zones, and interfaces is inherently less secure than dedicating an internal virtual router, internal security zone, and internal and external interfaces to each vsys. When all virtual systems share the same interfaces, it is possible for a vsys admin in one vsys to use the **snoop** command to gather information about the traffic activities of another vsys. Also, because IP-spoofing is possible on the internal side, we recommend that you disable the IP-spoofing SCREEN option on the shared internal interface. When deciding which traffic classification scheme to use, you must weigh the ease of management offered by the IP-based approach against the increased security and greater addressing flexibility offered by the VLAN-based approach.

Configuring IP-Based Traffic Classification

In this example, you set up IP-based traffic classification for the three virtual systems created in “Creating a Virtual System Object and Admin” on page 1681. You define the trust-vr as sharable. You create a new zone, name it Internal, and bind it to the

trust-vr. You then make the Internal zone sharable. You bind ethernet3/2 to the shared Internal zone, assign it IP address 10.1.0.1/16, and select NAT mode.

You bind ethernet1/2 to the shared Untrust zone and assign it IP address 210.1.1.1/24. The IP address of the default gateway in the Untrust zone is 210.1.1.250. Both the Internal and Untrust zones are in the shared trust-vr routing domain.

The subnets and their respective vsys associations are as follows:

- 10.1.1.0/24 – vsys1
- 10.1.2.0/24 – vsys2
- 10.1.3.0/24 – vsys3

WebUI

1. **Virtual Routers, Security Zones, and Interfaces**

   Network > Routing > Virtual Routers > Edit (for trust-vr): Select the Shared and accessible by other vsys check box, then click **OK**.

   Network > Zones > New: Enter the following, then click **OK**:

   Zone Name: Internal
   Virtual Router Name: trust-vr
   Zone Type: Layer 3

   Network > Zones > Edit (for Internal): Select the Share Zone check box, then click **OK**.

   Network > Interfaces > Edit (for ethernet3/2): Enter the following, then click **OK**:

   Zone Name: Internal
   IP Address/Netmask: 10.1.0.1/16

   Network > Interfaces > Edit (for ethernet1/2): Enter the following, then click **OK**:

   Zone Name: Untrust
   IP Address/Netmask: 210.1.1.1/24

2. **Route**

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 0.0.0.0/0
   Gateway: (select)
   Interface: ethernet1/2
   Gateway IP Address: 210.1.1.250

3. **IP Classification of the Trust Zone**

Network > Zones > Edit (for Internal) > IP Classification: Enter the following, then click **OK**:

System: vsys1
Address Type:
Subnet: (select); 10.1.1.0/24

Network > Zones > Edit (for Internal) > IP Classification: Enter the following, then click **OK**:

System: vsys2
Address Type:
Subnet: (select); 10.1.2.0/24

Network > Zones > Edit (for Internal) > IP Classification: Enter the following, then click **OK**:

System: vsys3
Address Type:
Subnet: (select); 10.1.3.0/24

Network > Zones > Edit (for Internal): Select the IP Classification check box, then click **OK**.

CLI

1. **Virtual Routers, Security Zones, and Interfaces**

```
set vrouter trust-vr shared
set zone name Internal
set zone Internal shared
set interface ethernet3/2 zone Internal
set interface ethernet3/2 ip 10.1.0.1/16
set interface ethernet3/2 nat
set interface ethernet1/2 zone untrust
set interface ethernet1/2 ip 210.1.1.1/24
```

2. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet1/2 gateway 210.1.1.250
```

3. **IP Classification of the Trust Zone**

```
set zone Internal ip-classification net 10.1.1.0/24 vsys1
set zone Internal ip-classification net 10.1.2.0/24 vsys2
set zone Internal ip-classification net 10.1.3.0/24 vsys3
set zone Internal ip-classification
save
```

Part 11
High Availability

High Availability presents an overview of the NetScreen Redundancy Protocol (NSRP) and describes how to cable, configure, and manage Juniper Networks security devices in a redundant group to provide high availability (HA) services using NSRP. It also covers NSRP-Lite, which is a lightweight version of NSRP that does not support the synchronization of Run-Time Objects (RTOs).

This guide contains the following chapters:

- "NetScreen Redundancy Protocol" on page 1765 explains how to cable, configure, and manage Juniper Networks security devices in a redundant group to provide high availability (HA) using the NetScreen Redundancy Protocol (NSRP).
- "Interface Redundancy and Failover" on page 1817 describes the various ways in which Juniper Networks security devices provide interface redundancy.

Chapter 56

NetScreen Redundancy Protocol

This guide describes the components of NetScreen Redundancy Protocol (NSRP) and details how to use NSRP to support high availability (HA). This chapter contains the following sections:

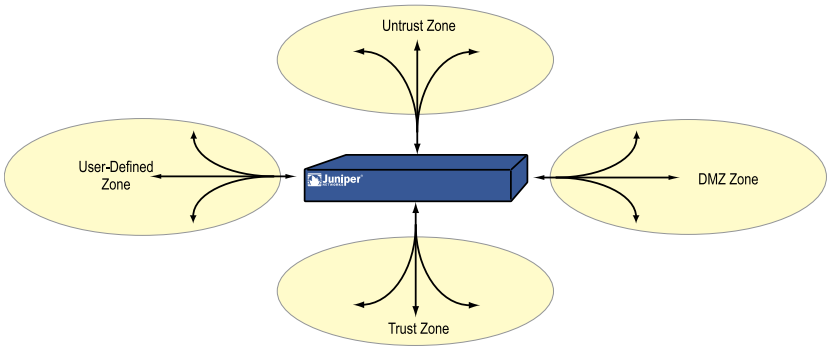
High Availability Overview

High availability (HA) provides a way to minimize the potential for device failure within a network. Because all of your network traffic passes through a Juniper Networks security device, you need to remove as many points of failure as possible from your network by ensuring that the device has a backup.

Setting up your security devices in HA pairs removes one potential point of failure from your network design. You can remove other potential points of failure by setting up redundant switches on either side of the HA pair of security devices.

For HA to function properly as a network firewall, a security device must be placed at the single point through which all inter-zone traffic must pass as shown in Figure 435 on page 1765.

Figure 435: All Inter-Zone Traffic Flowing Through the Firewall



With the security device in route or NAT mode, you can configure both devices in a redundant cluster to be active, sharing the traffic distributed between them by routers with load-balancing capabilities running a protocol such as the Virtual Router Redundancy Protocol (VRRP). This is accomplished using NSRP to create two virtual security device (VSD) groups, each with its own virtual security interfaces (VSIs).

Table 123 on page 1766 shows the HA features available with NSRP enabled on two security devices:

Table 123: High Availability Features

HA Feature	NSRP-Lite	NSRP
Fall-back to dialup		
Active/Passive	Yes	Yes
Active/Active	No	Yes
Active/Active Full-Mesh	No	Yes

NSRP Overview

When a security device is operating at Layer 3 (NAT or route mode) or in Layer 2 (transparent mode), it can be in an Active/Active or Active/Passive NetScreen Redundancy Protocol (NSRP) configuration. To manage a backup device for either mode, you must use the Manage IP address that you set per security zone interface. When you set a security device in an NSRP cluster, the device automatically creates VSD group 0 and transforms physical interfaces into VSIs for VSD group 0. The basic principle of NSRP is that there can be no single point of failure. All NSRP information passes between cluster members through two HA interfaces.

NOTE: You cannot set a Manage IP address on a VSI for any VSD group except VSD group 0.

Before NSRP can function, you must first cable two security devices together as explained in your device hardware installation and configuration guide. Also, if you want to maintain network connectivity for administrative traffic to one or more physical interfaces on a security device in an NSRP cluster, first set the Manage IP address for those interfaces as explained in"Setting Manage IPs for Multiple Interfaces" on page 343 before you enable NSRP.

WebUI

Network > NSRP > Synchronization: Select **NSRP RTO Synchronization**, then click **Apply**.

CLI

1. **Bind NSRP Cluster and VSD Group**

   set nsrp cluster id *number*

2. **Enable Automatic RTO Synchronization**

   set nsrp rto-mirror sync

3. **Configure Ports**

   set interface *interface* zone ha
   set interface *interface* zone ha

NSRP Default Settings

NSRP Default Settings	Value
VSD Group Information	■ VSD group ID: 0 ■ Device priority in the VSD group: 100 ■ Preempt option: disabled ■ Preempt hold-down time: 0 seconds ■ Initial state hold-down time: 5 seconds ■ Heartbeat interval: 1000 milliseconds ■ Lost heartbeat threshold: 3 ■ Master (Primary) always exist: no
RTO Mirror Information	■ RTO synchronization: disabled ■ Heartbeat interval: 4 seconds ■ Lost heartbeat threshold: 16

NSRP Link Information	■ Number of gratuitous ARPs: 4
	■ NSRP encryption: disabled
	■ NSRP authentication: disabled
	■ Track IP: none
	■ Interfaces monitored: none
	■ Secondary path: none
	■ HA link probe: none
	  ■ Interval: 15
	  ■ Threshold: 5

NOTE: The convention for indicating a VSI is *interface_name* **:** *VSD_group_ID*. For example, the following indicates that the redundant interface **red1** is a VSI for VSD group 1: **red1:1**. However, if the VSD group ID is 0, no VSD group ID is specified. For example, if the redundant interface **red2** is a VSI for VSD group 0, it appears simply as **red2**.

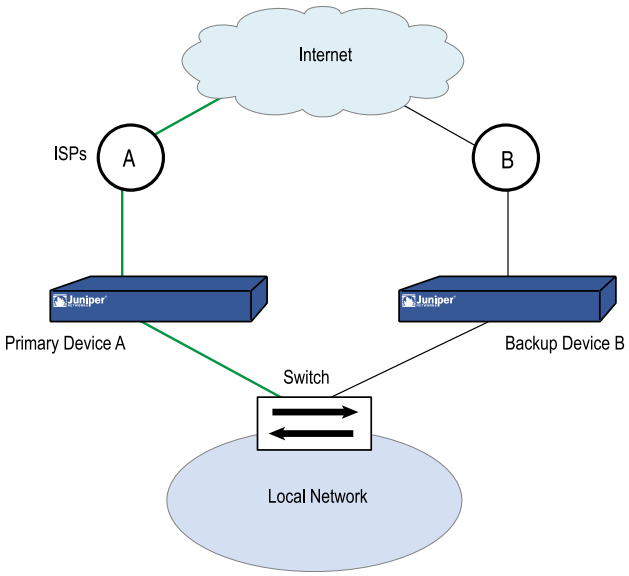
NSRP-Lite

NSRP-Lite supports selected NSRP features and is supported only on some Juniper Networks security platforms running ScreenOS in route or NAT mode. NSRP-Lite allows the following:

- Only "Active/Passive Configuration" on page 1777
- "Configuration Synchronization" on page 1784 synchronization, although not by default
- User session and VPN connection disruption when failover occurs because no RTO synchronization happens.

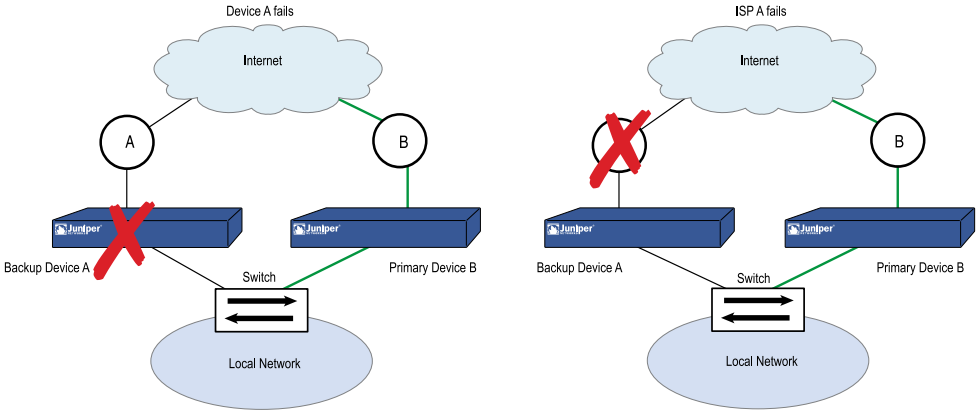
NOTE: VPN tunnels must be reestablished. We recommend that you enable VPN monitoring with the rekey option on VPN tunnels so that they automatically reestablish themselves.

Figure 436: NSRP-Lite Setup



If either device A or ISP A fails, device B becomes primary device and device A becomes backup (or it becomes inoperable if it has internal system problems). See Figure 437 on page 1769.

Figure 437: NSRP-Lite Failover



NSRP-Lite Default Settings

The basic NSRP-Lite configuration uses the following default settings shown in Table 124 on page 1769.

Table 124: Default NSRP-Lite Settings

Setting	Default
Configure Sync	disabled
RTO Sync	
VSD Group Information	■ VSD group ID: 0 ■ Device priority in the VSD group: 100 ■ Preempt option: disabled ■ Preempt hold-down time: 0 seconds ■ Initial state hold-down time: 5 seconds ■ Heartbeat interval: 1000 milliseconds ■ Lost heartbeat threshold: 3
NSRP Link Information	■ Number of gratuitous ARPs: 4 ■ NSRP encryption: disabled ■ NSRP authentication: disabled ■ Interfaces monitored: none ■ Secondary path: none

Basic NSRP Settings

If you bind two interfaces (gigabit or 100-megabit) to the HA zone, the interface with the lower number becomes the control link, and the interface with the higher number becomes the data link. For example, if you bind only ethernet 8 to the HA zone, it becomes the control link. If you then bind ethernet7 to the HA zone, it becomes the

control link (because it has a lower number than ethernet8), and ethernet8 changes to the data link. An exception is when you bind a Gigabit Ethernet interface and a Fast Ethernet interface to the HA zone. In this case, the Gigabit Ethernet interface becomes the control link.

On security devices that do not have dedicated HA interfaces, you must bind one or two physical Ethernet interfaces to the HA zone. If you bind a single gigabit interface to the HA zone, the HA link supports both control and data messages. If you bind one 100-megabit interface to the HA zone, the HA link supports control messages only.

NOTE: More than three interfaces can be bound to the HA zone; however, only the first three entries can be configured as an HA link.

Control Link Messages

There are two kinds of control messages: heartbeats and HA messages.

Heartbeats: Heartbeats are sent periodically to establish and sustain communications among the NSRP cluster members, VSD group members, and RTO mirrors. The heartbeats continually advertise the sender's member status, and the health of its system and network connectivity. Table 125 on page 1770 describes the three kinds of heartbeat messages.

Table 125: Heartbeat Message Descriptions

Heartbeat Messages	Description
HA physical link	Broadcast messages from the HA1 and HA2 interfaces of each member of an NSRP cluster to the other member. The purpose of these messages is to monitor the health of the HA interfaces. If, for example, one member does not receive three consecutive heartbeats from HA1, both devices transfer transmission of the control messages to HA2. Available for all NSRP configurations
VSD	Broadcast messages from the HA1 interface of each member of a VSD group. The VSD group uses these messages to monitor the membership status of all its members. If, for example, the primary device advertises that it has become inoperable, the primary backup immediately becomes the VSD group primary device. Available for all NSRP configurations.
RTO	Broadcast messages that are sent from each HA1 interface. The purpose of these messages is to locate an active peer and then maintain the mirror relationship by sending group active messages. If, for example, a device does not receive 16 consecutive RTO heartbeats from its peer, it changes its state from active to set. Available for Active/Passive and Active/Passive full-mesh NSRP configurations only.

NOTE: If you remove a device from a mirror group, it enters the undefined state and transmits a "group detach" message to its peer. The peer immediately changes its state from active to set without waiting for the missing heartbeats to exceed the threshold.

HA Messages: The two kinds of HA messages are as follows:

- Configuration messages—The network and configuration settings that the primary device sends to the other VSD group member
- RTO messages—The RTOs that the primary device sends to the other RTO mirror

The HA messages contain the information that enables the backup to become the primary device without causing a service interruption.

Data Link Messages

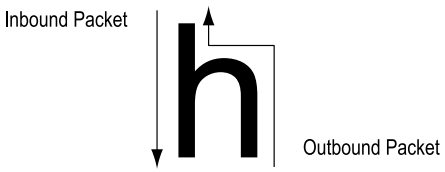
Data messages are IP packets traversing the firewall that the backup in a VSD group must forward to the device acting as primary device. When a packet arrives at the interface of a security device in an Active/Active configuration, the device first identifies which VSD group must process the packet. If the device that receives the packet is the primary device of the identified VSD group, it processes the packet itself. If the device is not the primary device, it forwards the packet over the HA data link to the primary device.

For example, a load-balancing router might send the first packet in a session to device A (primary device of VSD group 1), which creates an entry in its session table. If the router performs load balancing by sending packets round-robin (that is, the router sends each packet to a security device in turn), the router might send the next packet to device B (backup of VSD group 1). Because a session entry exists in device A, device B forwards the packet across the data link to device A, which processes it.

NOTE: If there is no data link, the security device that receives the packet drops it immediately.

Inbound packet forwarding across the data link occurs only when the security devices are in an Active/Active configuration in route mode. When in NAT mode, the router always sends the incoming packets to the MIP, VIP, or VPN tunnel gateway, although the security device that receives the returning outbound packet might forward it across the data link to the device that has the session entry to which the packet belongs. This kind of packet forwarding produces an h-shaped path. Like the down stroke in the letter h, the inbound packet goes straight through one device, but the outbound packet might be sent halfway through the other device and then forwarded across the data link to the first device. See Figure 438 on page 1772.

Figure 438: Packet Forwarding Across the Data Link



Dynamic Routing Advisory

If an NSRP cluster is in a dynamic routing environment and you disable packet forwarding, traffic arriving at an inactive interface can be lost. Because the security device cannot forward traffic across the data link to the security device on which the interface is active, it drops the traffic. To avoid this when you disable packet forwarding, the security device indicates the status of interfaces belonging to a backup VSD on a device as down instead of merely inactive. This status signals routers not to send traffic to these interfaces.

When packet forwarding is disabled, the VSI status on the backup VSD is down irrespective of the state of the physical base interface of the VSI. However, the VSI state on the backup VSD is inactive when packet forwarding is enabled. Table 126 on page 1772 displays the VSI link state table.

Table 126: VSI Link State Table

VSD	VSI State	VSI Physical Interface State
Primary	Up	Up
Primary	Down	Down
Backup	Inactive (packet forwarding enabled)	Up
Backup	Inactive (packet forwarding enabled)	Down
Backup	Down (packet forwarding disabled)	Up
Backup	Down (packet forwarding disabled)	Down

To disable packet forwarding in an NSRP cluster, use the **unset nsrp data-forwarding** CLI command. In addition, nonavailability of an HA zone or a primary VSD disables packet forwarding.

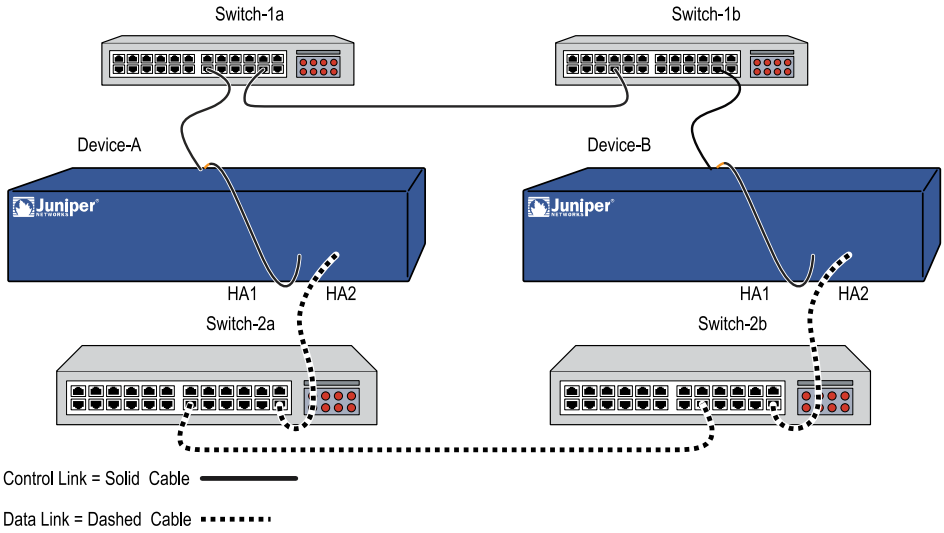
WebUI

NOTE: You must use the CLI to disable packet forwarding.

Dual Link Probes

You can connect the redundant HA interfaces by directly cabling HA ports on one device to the HA ports on another device. Or you can connect the HA ports on two devices through one or more switched networks. In the configuration shown in Figure 439 on page 1773, the HA1 port on Device A is connected to the HA1 port on Device B through two switches, Switch-1a and Switch-1b. To provide a redundant HA interface, the HA2 port on Device A is connected to the HA2 port on Device B through Switch-2a and Switch-2b. In this configuration, the link between the HA1 ports on Device A and Device B handles NSRP control messages, while the HA2 link handles network data messages. If the link between the HA1 port on Device A and Switch-1a goes down, Device A transfers transmission of the control messages to its HA2 port. However, Device B does not recognize the failure of the HA1 link as its HA1 port is still active and rejects the NSRP control messages sent by Device A on the HA2 link.

Figure 439: HA Links Connecting Through Switches



To prevent this situation, you can configure a security device to monitor the status of a HA zone by sending NSRP probe requests on the HA link to its peer. If a reply is received from the peer on the HA link, the request is considered successful and the HA link is assumed to be up. If no reply is received from the peer within the constraints specified, the HA link is considered to be down. This enables security devices to switch transmission of control messages to an available HA link when necessary, even if there is no physical failure on the HA ports on either device.

There are two ways by which probe requests can be sent on an HA link:

- **Manually by the administrator**—Probes are sent on a specific HA link once every second for a specified number of times. If no reply is received from the peer after the specified number of probes is sent, the HA link is considered to be down. Probes are sent out immediately after you execute the command.
- **Automatically by ScreenOS**—Probes are sent on all HA links once every second. (You can optionally specify the HA zone interface and the interval at which probes

are sent.) By default, if five consecutive probes are sent without receiving a reply from the peer, the link is considered to be down; you can specify a different threshold value for determining when the link is down. Note that even when a primary HA link is down, the security device continues to send probes on that link. If the primary HA link connection is restored and peer responses are once again received on the link, the security devices can switch transmission of control messages back to the primary HA link.

NSRP Clusters

An NSRP cluster consists of a group of security devices that enforce the same overall security policy and share the same configuration settings. When you assign a device to an NSRP cluster, any changes made to the configuration on one member of the cluster propagate to the others.

An NSRP cluster can contain two security devices. You can assign cluster IDs in the range 1–63 and VSD group IDs in the range 0–63. However, you cannot have 64 cluster IDs and VSD group IDs concurrently; for example, if you configure 32 NSRP cluster IDs, the number of VSD group IDs possible is 16.

NOTE: In a network with more than one NSRP cluster, do not assign the same cluster IDs or VSD group IDs for clusters in the same broadcast domain, because doing so may cause both clusters to become unreachable via network interface. NSRP clusters derive virtual MAC addresses (VMACs) based on the cluster ID, interface ID, and VSD group ID, and using the same cluster ID or VSD group ID for two clusters may cause them to be assigned the same VMAC.

You set the **nsrp-max-cluster** environment variable to enable the security device to support a maximum of 64 cluster IDs in the domain. To set the range for NSRP cluster IDs, use the **set envar nsrp-max-cluster** CLI command. You must reboot the security device in order for the changes to take effect.

NOTE: The combined value of NSRP clusters and VSD groups must be a power of two except 0. If you set a value greater than the maximum allowed range, ScreenOS rounds down the setting to 64. The combined value of the cluster IDs and the number of VSD group IDs cannot exceed 512.

Example

In the following example, you configure an NSRP cluster and VSD groups on two security devices – Device A and Device B. For each security device with a cluster id to 15, you create two VSD groups—VSD group ID 20 and VSD group ID 30. The device with lower priority (higher number) acts as the backup device, while the device with higher priority (lower number) acts as the primary.

NOTE: You use the preempt option to select the backup device that becomes the next immediate primary device in the VSD group based on the priority and the unit IDs.

Using this configuration, Device A acts as the backup and Device B as the primary for VSD group 20. Similarly for VSD group 30, Device B acts as the backup and Device A as the primary. You switch the device roles by changing the priority for each of the VSD groups.

WebUI

1. **Configure an NSRP cluster:**

   Network > NSRP > Cluster: Enter the following, then click Apply:

   NSRP Protocol Version: 2.0 (read-only)
   Client ID: (select)
   Local Unit: 12359360
   Active Unit Discovered: 12359360
   Number of Gratuitous ARP to Resend: 4

2. **Configure a VSD group:**

   Network > NSRP > VSD Group > Configuration: Enter the following, then click **OK:**

   Group ID: 20
   Priority: 100
   Enable Preempt: (select)
   Preempt Hold-Down Time (sec): 3

NOTE: You cannot configure environment variables using the WebUI.

CLI

1. **Configure an NSRP Cluster**

   ```
   set envar nsrp-max-cluster 16
   ```

2. **Configure a VSD Group**

   ```
   set envar nsrp-max-vsd 32
   ```

3. **Set VSD Group Priority (Device A)**

   ```
   set nsrp cluster id 15
   set nsrp vsd-group id 20 priority 100
   set nsrp vsd-group id 20 preempt
   set nsrp vsd-group id 30 priority 50
   ```

```
set nsrp vsd-group id 30 preempt
set nsrp rto-mirror sync
```

4. **Set VSD Group Priority (Device B)**

```
set nsrp cluster id 15
set nsrp vsd-group id 20 priority 50
set nsrp vsd-group id 20 preempt
set nsrp vsd-group id 30 priority 100
set nsrp vsd-group id 30 preempt
set nsrp rto-mirror sync
save
```

Members of the same NSRP cluster maintain the following identical settings:

- Policies and policy objects (such as addresses, services, VPNs, users, and schedules)
- System parameters (such as settings for authentication servers, DNS, SNMP, syslog, Web filtering, firewall detection options, and so on)

Table 127 on page 1776 displays the list of non-propagating commands. Members of an NSRP cluster do not propagate to these configuration settings.

Table 127: Non-Propagating Commands

NSRP	■ set/unset nsrp cluster id *number* ■ set/unset nsrp auth password *pswd_str* ■ set/unset nsrp encrypt password *pswd_str* ■ set/unset nsrp monitor interface *interface* ■ set/unset nsrp vsd-group id *id_num* { mode *string* | preempt | priority *number* }
Interface	■ set/unset interface *interface* manage-ip *ip_addr* ■ set/unset interface *interface* phy … ■ set/unset interface *interface* bandwidth *number* ■ set/unset interface redundant *number* phy primary *interface* ■ All commands pertaining to local interfaces
Monitored Objects	All IP tracking, zone monitoring, and interface monitoring commands
Console Settings	All console commands (set/unset console …)
Hostname	set/unset hostname *name_str*
SNMP	set/unset snmp name *name_str*
Virtual Router	set/unset vrouter *name_str* router-id *ip_addr*
Clear	All clear commands (clear admin, clear dhcp, …)
Debug	All debug commands (debug alarm, debug arp, …)

Table 127: Non-Propagating Commands *(continued)*

Gateway tracking	set/unset vrouter *name_str* route *ip_addr* gateway *ip_addr*

Cluster Names

Because NSRP cluster members can have different hostnames, a failover can disrupt SNMP communication and the validity of digital certificates because SNMP communication and certificates rely on the hostname of a device to function properly.

To define a single name for all cluster members, use the **set nsrp cluster name** *name_str* CLI command.

NOTE: You must use the CLI to set the NSRP cluster name.

On devices that do not have a dedicated HA port, you must bind the interface to the HA zone before configuring the NSRP cluster.

Use the cluster name when configuring the SNMP hostname for the security device (**set snmp name** *name_str*) and when defining the common name in a PKCS10 certificate request file.

The use of a single name for all cluster members allows SNMP communication and digital certificate use to continue without interruption after a device failover.

ScreenOS allows you to configure the following HA cluster configurations:

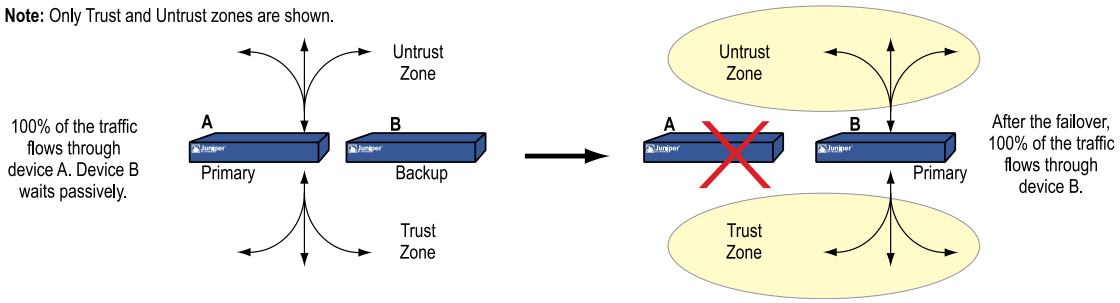
- “Active/Passive Configuration” on page 1777
- “Active/Active Configuration” on page 1778
- “Active/Active Full-Mesh Configuration” on page 1779

Active/Passive Configuration

To ensure a continuous traffic flow, you can cable and configure two security devices in a redundant cluster, with one device acting as a primary device and the other as its backup. The primary device propagates all its network and configuration settings and the current session information to the backup device. If the primary device fails, the backup device is promoted to primary and takes over the traffic processing.

In Figure 440 on page 1778, the two devices are in an Active/Passive configuration; that is, the primary device is active, handling all firewall and VPN activities, and the backup device is passive, waiting to take over when the primary device fails.

Figure 440: Active/Passive

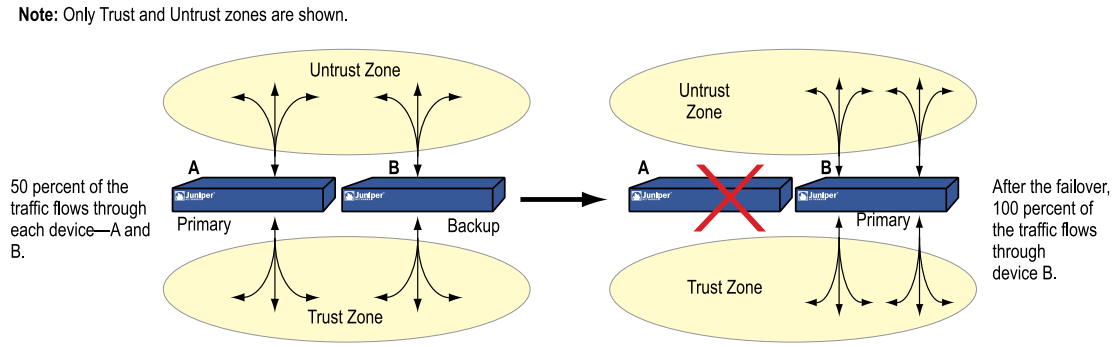


NOTE: Although the backup device is passive in the sense that it is not processing traffic, it is maintaining its synchronization with the configuration settings and session information it continuously receives from the primary device.

Active/Active Configuration

Devices A and B each receive 50 percent of the network and VPN traffic. Should device A fail, device B becomes the primary device of VSD group 1, as well as continuing to be the primary device of VSD group 2, and handles 100 percent of the traffic. Traffic redirection resulting from a failover in an Active/Active configuration is shown in Figure 441 on page 1778.

Figure 441: Active/Active

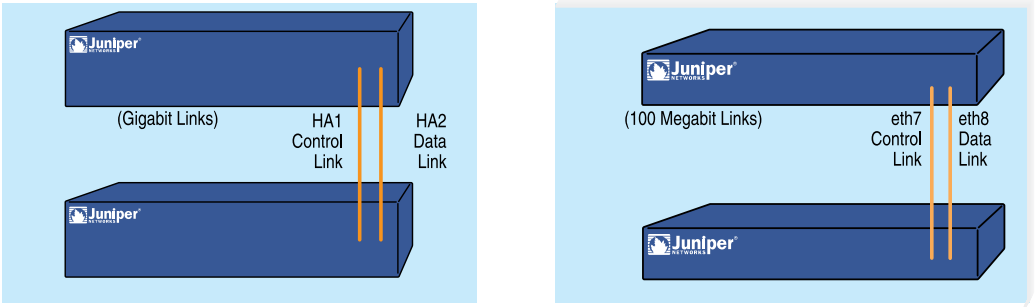


Although the total number of sessions divided between the two devices in an Active/Active configuration cannot exceed the capacity of a single security device (otherwise, in the case of a failover, the excess sessions might be lost), the addition of a second device doubles the available bandwidth potential. A second active device also guarantees that both devices have functioning network connections.

NOTE: Each device in an Active/Active configuration can tolerate traffic bursts exceeding 50 percent of the capacity of a single device for short periods of time; however, should a failover occur during that period, the excess traffic might be lost.

To better distribute the out-of-band bandwidth, HA1 handles the NSRP control messages while HA2 handles the network data messages. If either port fails on a security device with gigabit HA1 and HA2 interfaces, the remaining active port assumes both kinds of traffic. For security devices that must use a 100-megabit interface for the data link, a failure of the data link results in one active HA link for control messages only. If the control link fails on such devices, then the data link becomes the control link and sends and receives control messages only. See Figure 442 on page 1779.

Figure 442: Dedicated HA Links and User-Assigned HA Links



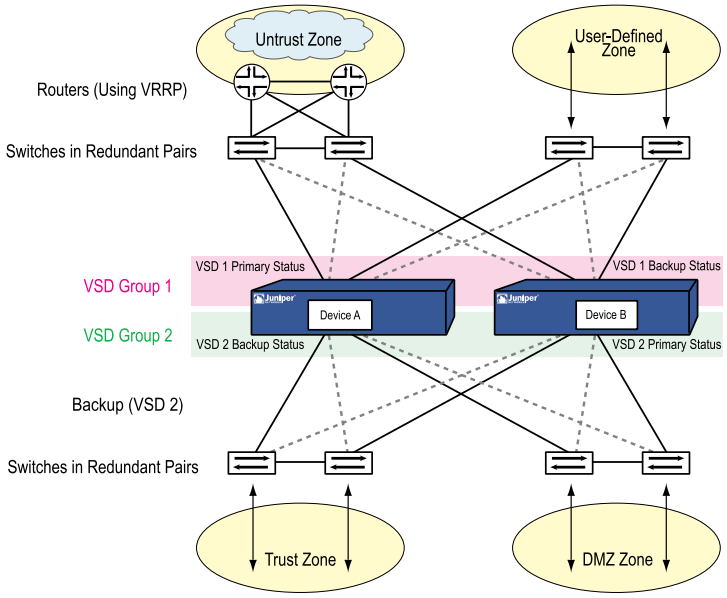
If either HA1 or HA2 fails, then the control and data messages are both sent over the remaining HA link.

If either ethernet7 or ethernet8 fails, then only control messages are sent over the remaining HA link.

NOTE: If you use a switch between HA ports, use port-based VLANs, which do not conflict with the VLAN tags on the forwarded packets.

Active/Active Full-Mesh Configuration

Figure 443: Introducing Fault-Tolerance into the Network



In addition to NSRP clusters, which are primarily responsible for propagating configurations among group members and advertising each member's current VSD group states, you can configure devices A and B as members in an RTO mirror group, which is responsible for maintaining the synchronicity of run-time objects (RTOs) between a pair of devices. When the primary device fails, the backup can immediately assume the primary device's role with minimal service downtime by maintaining all current sessions.

NOTE: RTOs are objects created dynamically in the security device memory during the normal operation of the device. RTOs allow the device to understand the network around it and enforce its policies. Examples of RTOs are TCP/UDP sessions, IPsec Phase 2 security associations (SAs), DHCP allocations, RSA and DSS key pairs, ARP tables, and DNS caches.

You can secure all NSRP traffic with encryption and authentication. NSRP supports the DES and MD5 algorithms. (For more information about these algorithms, see "Protocols" on page 711.)

NOTE: If the HA cables run directly from one security device to another (that is, not through a switch forwarding other kinds of network traffic), it is unnecessary to use encryption and authentication.

If you want to use Simple Network Management Protocol (SNMP) to monitor the security device, private NSRP MIBs are available for download at www.juniper.net/customers/support. (For more information about SNMP, see "Simple Network Management Protocol" on page 397.)

NSRP Cluster Authentication and Encryption

Because of the sensitive nature of NSRP communications, you can secure all NSRP traffic through encryption and authentication. For encryption and authentication, NSRP supports the DES and MD5 algorithms respectively.

NOTE: When the devices are cabled directly to one another, there is no need to use authentication and encryption. However, if the devices are cabled through a switch to which other devices connect, you might consider implementing these additional security measures.

To enable authentication or encryption, you must provide passwords on each device in the cluster.

WebUI

Network > NSRP > Cluster: Enter the following, then click **Apply**:

NSRP Authentication Password: (select), pswd_str
NSRP Encryption Password: (select), pswd_str

CLI

```
set nsrp auth password pswd_str
set nsrp encrypt password pswd_str
```

Run-Time Objects

Run-Time Objects (RTOs) are code objects created dynamically in memory during normal operation. Some examples of RTOs are session table entries, ARP cache entries, DHCP leases, and IPsec security associations (SAs). In the event of a failover, it is critical that the current RTOs be maintained by the new primary device to avoid service interruption.

NOTE: Using policies, you can specify which sessions to backup and which not to backup. For traffic whose sessions you do not want backed up, apply a policy with the HA session backup option disabled. In the WebUI, clear the HA Session Backup check box. In the CLI, use the **no-session-backup** argument in the **set policy** command. By default, the backing up of sessions is enabled.

IPv6 sessions also support Netscreen Redundancy Protocol (NSRP).

To accomplish this, RTOs are backed up by the members of an NSRP cluster.

Each member backs up the RTOs from the other, which allows RTOs to be maintained if the primary device of either VSD group in an Active/Active HA scheme fails.

In the current ScreenOS release, you do not have to configure one or more RTO mirror groups to synchronize RTOs among members of an NSRP cluster. Defining a security device as a member of a cluster and specifying RTO synchronization automatically enables the local device to send and receive RTOs.

By default, NSRP cluster members do not synchronize RTOs. Before enabling RTO synchronization, you must first synchronize the configurations between the cluster members. Unless the configurations on both members in the cluster are identical, RTO synchronization might fail.

To enable RTO synchronization:

WebUI

Network > NSRP > Synchronization: Select **NSRP RTO Synchronization**, then click **Apply**.

CLI

```
set nsrp rto-mirror sync
save
```

NOTE: In the event of a failover, the device that mirrors the primary device is the most desirable replacement—even if another VSD group member has a higher priority. In transparent mode, the physical interface of the security device goes down only when all the VSDs, both in Active/Active and in Active/Passive configurations, become passive.

To disable RTO session synchronization on the device acting as sender in an NSRP cluster:

WebUI

Network > NSRP > Synchronization: Deselect **NSRP Session Synchronization**, then click **Apply**.

CLI

```
set nsrp rto-mirror session off
save
```

Issuing this command on a device only disables session synchronization from that device to others in the cluster.

RTO Mirror Operational States

The procedure for two NSRP cluster members to initiate their RTO mirror relationship develops through two operational states—set and active. The devices progress through these states as follows:

1. After you add the first device to a group, its state is set. In the set state, the device waits for its peer to join the group. As the receiver of RTOs, it periodically transmits an r-ready message (receiver-ready), announcing its own availability. As the sender of RTOs, it waits until it gets an r-ready message from a device with the same cluster ID.
2. After you add the peer and the two devices are correctly cabled for HA, then the following occurs:
   a. The receiver sends an r-ready message.
   b. The sender gets the r-ready message, and immediately sends a group-active message to inform its peer that its state is now active.
   c. The receiver then changes its state to active as well.

In addition to passing RTOs from sender to receiver, both active mirrors send RTO heartbeats at user-defined intervals to communicate their operational status.

To define the heartbeat interval:

WebUI

Network > NSRP > Link: Enter a value in the **Interval** field, then click **Apply**.

CLI

```
set nsrp rto-mirror hb-interval number
save
```

If a device does not receive a specified number of consecutive heartbeats from its peer, it changes its state from active to set.

To define the lost heartbeat threshold required to impel a state changeover:

WebUI

Network > NSRP > Link: Enter a value in the **Threshold** field, then click **Apply**.

CLI

```
set nsrp rto-mirror hb-threshold number
save
```

NOTE: To maintain identical RTO heartbeat settings, the **set nsrp rto-mirror hb-interval** *number* and **set nsrp rto-mirror hb-threshold** *number* CLI commands are propagated.

If you want to clear a session for an inactive VSI:

WebUI

NOTE: You must use the CLI to clear rto-mirror sessions.

CLI

```
set nsrp rto-mirror session clear-on-inactive
save
```

NSRP Cluster Synchronization

When you add a new device to an active NSRP cluster, you must synchronize the configuration and files (such as PKI public/private key files) from the primary device of the VSD group or groups to the new device. After the configurations and files are synchronized, you must then synchronize the run-time objects (RTOs). You must also synchronize configurations, files, and RTOs after a member of a cluster becomes unsynchronized for any reason.

NSRP allows you to synchronize the following information:

- Files
- Configurations
- Routes
- Run-Time Objects
- System Clocks

File Synchronization

To synchronize all files:

WebUI

NOTE: You must use the CLI to synchronize files.

CLI

1. **Sync Files**

   exec nsrp sync file from peer

   or synchronize a single file

   exec nsrp sync file name *name_str* from peer

2. **Reapply All Licenses**

   exec license-key update

NOTE: The license-key update only applies if you synchronize all files.

To synchronize PKI objects such as local and CA certificates, key pairs, and CRLs, use the **exec nsrp sync global-config save** command.

Configuration Synchronization

If you make any configuration changes on one device while another in the cluster reboots (or if all HA links fail), it is possible that the configuration settings can become unsynchronized.

By default, devices placed in an NSRP cluster do not synchronize configurations and files. This setting is useful, for example, if you want all configuration changes to originate from Network and Security Manager (NSM).

To enable the automatic synchronization of configurations, use the **set nsrp config sync** command on all members in the cluster (the WebUI does not support this option).

NOTE: ScreenOS 6.2.0 supports configuration synchronization for the devices acting as DHCP clients or servers in an NRSP cluster.

Before enabling automatic configuration synchronization, we recommend that you first manually synchronize files—such as PKI objects—between the cluster members. If you synchronize the configuration, but one cluster member is missing a file that is referenced in the configuration, the configuration becomes invalid for that member. To avoid that, first synchronize files and then the configuration.

To discover if the configuration of one device is out of sync with that of another, use the **exec nsrp sync global-config check-sum** command. The output states whether the configurations of the two devices are in or out of sync and provides the checksums of the local and remote devices.

NOTE: In previous releases, when you used Telnet or Secure Shell (SSH) for remote administration of *exec nsrp* commands, the output did not appear until you subsequently used the *get log sys* command. In the current release, the output appears instantly for all *exec nsrp* commands except the following three:

- **exec nsrp probe...**
- **exec nsrp vsd-group...**
- **exec nsrp sync pki...**

If the devices in the NSRP cluster run two different versions of ScreenOS, the output will not appear instantly.

If the configurations are out of sync, use the **exec nsrp sync global-config save** command to resynchronize them. After you resynchronize the configurations, you must restart the device.

NOTE: Configurations on active devices in a cluster rarely become unsynchronized because NetScreen Reliable Transport Protocol (NRTP) is a low-overhead, TCP-like protocol.

Route Synchronization

When a failover occurs on devices in an NSRP Active/Passive cluster configured with Dynamic Routing Protocol (DRP) routes, traffic is disrupted while the backup device becomes the primary and converges its route table with its peers. If the primary device is configured to synchronize routes, however, the backup device is able to continue sending traffic even as it becomes the primary and converges with its peers.

When you configure route synchronization on the primary NSRP device, you need to set the routes in a VSD interface that does not have a local virtual route. DRP routes are synchronized similarly to RTO mirror synchronization, except the synchronization request only applies to routes.

NOTE: Route synchronization for IPv6 supports Netscreen Redundancy Protocol (NSRP).

To configure route sync on an NSRP Active/Passive cluster:

WebUI

Network > NSRP > Synchronization: Select **Route Synchronization**, enter the **Threshold** value, then click **Apply**.

CLI

```
set nsrp rto-mirror route
save
exec nsrp sync rto route from peer
```

Run-Time Object Synchronization

If you have enabled RTO mirror synchronization on a device in a cluster (see “Synchronizing RTOs Manually” on page 1809), when the device reboots, the RTOs automatically resynchronize. However, if you disable RTO mirror synchronization—perhaps to perform some debugging or maintenance on the device—when you again enable RTO synchronization, you must manually resync all the RTOs.

Table 128 on page 1786 provides the CLI commands you use to synchronize RTOs.

WebUI

NOTE: You must use the CLI to synchronize RTOs.

ScreenOS 6.2.0 supports RTO synchronization for devices acting as DHCP clients or servers in an NRSP cluster.

Table 128: CLI Commands for RTO Synchronization

Synchronization Method	CLI Command
Manual	exec nsrp sync rto all
Selected RTOs such as ARP, DNS, sessions, or VPNs	exec nsrp sync rto {...}

Table 128: CLI Commands for RTO Synchronization *(continued)*

Synchronization Method	CLI Command
Automatic	set nsrp rto-mirror sync

System Clock Synchronization

NSRP contains a mechanism for synchronizing the system clocks of NSRP cluster members. When you set the system clock manually, the NSRP time synchronization mechanism keeps the members' clocks properly synchronized. However, when you use Network Time Protocol (NTP) to set the system clocks on all the cluster members, and then use NSRP to synchronize the time among them, the time can become unsynchronized. Although the resolution for NSRP synchronization is in seconds, NTP has sub-second resolution. Because processing delays can cause the time on each cluster member to differ by a few seconds, we recommend that you disable NSRP time synchronization when NTP is enabled on all cluster members so that each member can update its system clock from an NTP server. To disable the NSRP time synchronization function, use the **set ntp no-ha-sync** CLI command.

WebUI

NOTE: You must use the CLI to synchronize the system clock.

Coldstart Synchronization

In normal session synchronization for an active/passive HA pair, only newly added or updated sessions and run-time objects (RTOs) on the primary device are added to the session table of the backup device. A coldstart synchronization, however, is a complete synchronization of all sessions and RTOs—from the primary to the backup device—in the session table in order to populate the backup device's session table with all active sessions. A coldstart synchronization starts when the primary and backup peers find each other after a device has been added to a cluster. This is the case whether the device is newly added or had previously failed out of the cluster for some reason and was later reenabled.

If you have enabled the **preempt** option, device failover occurs only after the synchronization concludes. When the coldstart synchronization is complete, the security device displays a **coldstart sync done** message.

NOTE: For ScreenOS 5.3 and later versions, the **coldstart sync done** message appears 2000 seconds (approximately 30 minutes) after the backup device has been restarted and the coldstart synchronization is complete.

Virtual Security Device Groups

A virtual security device (VSD) group is a pair of physical security devices that together make up a single VSD. One physical device acts as the primary device of the VSD group. The virtual security interface (VSI) of the VSD is bound to the physical interface of the primary device. The other physical device acts as the backup. If the primary device fails, the VSD fails over to the backup and the VSI binding is transferred to the physical interface on the backup, which is instantly promoted to primary device.

You can assign VSD group IDs in the range 0–63. However, you cannot have 64 cluster IDs and VSD group IDs concurrently; for example, if you configure 32 NSRP cluster IDs, the number of VSD group IDs possible is 16.

To assign a range to a VSD group ID, use the set envar nsrp-max-vsd CLI command.

NOTE: The current ScreenOS release only supports two members in a VSD group. If, in later releases, there is support for more than two members in a VSD group, one device would act as the primary device, another as the primary backup, and the remaining VSD group members as secondary backups.

By grouping two security devices into two VSD groups, with each physical device being the primary device in one group and the backup in the other, both devices can actively process traffic as primary devices while backing up each other in the event of a failover.

Upon initial NSRP configuration, the VSD group member with the priority number closest to 0 becomes the primary device. (The default is 100.) If two devices have the same priority value, the device with the higher MAC address becomes the primary device.

To view how many transitions have occurred between the master and the backup state, use get nsrp. The following example displays the output of the **get nsrp** command.

```
Device-> get nsrp counter protocol
Nsrp arp counts: 4, interval: 1
Nsrp VSD group counts:
```

group	st_chg	to ms	pb	bk	inop	ineli	init	dup_ms	dup_pb	hb_tx	hb_rx
0	2	1	1	0	0	0	0	0	0	379	524

The **st_chg** column displays the number of transitions. To clear the nsrp counter command, use **clear nsrp counter** command.

To view the state duration of the security devices' VSD groups, use the **get nsrp vsd-group** command. The following example displays the output of this command:

```
Device-> get nsrp vsd-group
Nsrp arp counts: 4, interval: 1
Nsrp VSD group counts:
```

group	priority	preempt	holddown	inelig	master	PB	other members	myself uptime
0	120	yes	3	no	myself	14854976		1d;05:18:42

```
total number of vsd groups: 1
Total iteration=1826,time=5062786,max=15806,min=493,average=2772

vsd group id: 0, member count: 2, master: 14810368
member information:
```

group	unit_id	state	prio	flag	rto_peer	hb	miss	holddown	uptime
0	14854976	primary backup	200	2	0		0	3	00:40:01
0	14810368	master	120	2	0	0	0	3	1d;05:18:42

The myself uptime column displays the VSD state duration of the current security device. The uptime column displays the VSD groups' state duration.

NOTE: To view the VSD state duration of the current security device alone, use the **get nsrp** command.

Preempt Option

You can determine whether a better priority number (closer to zero) can initiate a failover by setting the device that you want to be primary in preempt mode. If you enable the preempt option on that device, it becomes the primary device of the VSD group if the current primary device has a lesser priority number (farther from zero). If you disable this option, a primary device with a lesser priority than a backup can keep its position (unless some other factor, such as an internal problem or faulty network connectivity, causes a failover).

Using the hold-down time to delay a failover can prevent a flurry of rapid failovers in the event of port-flickering on an adjacent switch and also ensure that surrounding network devices have sufficient time to negotiate new links before the new primary device becomes available.

To set a VSD group:

WebUI

Network > NSRP > VSD Group > Configuration:

Group ID: **2**
Priority: **1**
Enable Preempt: (Selected)
Preempt Hold-down Time: **0**
Status: (Read Only)

CLI

```
set nsrp vsd-group id 2 preempt hold-down 0
```

Member States

Table 129 on page 1790 provides the status descriptions for members of a VSD group.

WebUI

NOTE: You must use the CLI to set VSD group members.

Table 129: VSD Group Status

Status	Description
Master	The state of a VSD group member that processes traffic sent to the VSI. It is the primary device.
Primary Backup	The state of a VSD group member that becomes the primary device if the primary device fails. The election process uses device priorities to determine which member to promote. Note that when electing a new primary device, an RTO peer has precedence over any other VSD group member, even if that member has a better priority rating.
Backup	The state of a VSD group member that monitors the status of the primary backup and elects one of the backup devices to primary backup if the current one steps down.
Initial	The transient state of a VSD group member while it joins a VSD group, either when the device starts or when it is added with the **set nsrp vsd-group id** *id_num* CLI command. To specify how long a VSD group member stays in the initial state, use the **set nsrp vsd-group init-hold** *number* CLI command.
Ineligible	The state that an administrator purposefully assigns to a VSD group member so that it cannot participate in the election process. To set the ineligible state to a VSD group member, use the **set nsrp vsd-group id id_num mode ineligible** CLI command.
Inoperable	The state of a VSD group member after a system check determines that the device has an internal problem (such as no processing boards) or a network connection problem (such as when an interface link fails).

To determine the initial state hold-down time, multiply init-hold value by the VSD heartbeat-interval (init-hold x hb-interval = initial state hold-down time). For example, if the init-hold is 5 and the hb-interval is 1000 milliseconds, then the initial state hold-down time is 5,000 milliseconds, or 5 seconds (5 x 1000 = 5000).

NOTE: If you reduce the VSD heartbeat interval, you should increase the init-hold value. For information about configuring the heartbeat interval, see "Heartbeat Message" on page 1791.

When the device returns from either the ineligible state (when you issue the **exec nsrp vsd-group id** *id_num* **mode backup** CLI command) or inoperable state (when the system or network problem has been corrected), it must first pass through the initial state.

Heartbeat Message

Every VSD group member—even if it is in the initial, ineligible, or inoperable state—communicates with its group members by sending a heartbeat message at configured intervals. These messages allow every member to know the current state of every other member. The heartbeat message includes the following information:

- Unit ID of the device
- VSD group ID
- VSD group member status (master, primary backup, or backup)
- Device priority
- RTO peer information

NOTE: If a device is in the inoperable state with all HA links down, it can neither send nor receive VSD heartbeat messages unless you have configured a secondary path for these messages.

The interval for sending VSD heartbeats is configurable (200, 600, 800, or 1000 milliseconds; 1000 ms is the default). ScreenOS also allows you to configure the lost heartbeat threshold that is used to determine when a VSD group member is considered as missing.

To configure the VSD group heartbeat values:

WebUI

NOTE: You must use the CLI to set VSD group heartbeat values.

CLI

```
set nsrp vsd-group hb-interval number
set nsrp vsd-group hb-threshold number
```

The heartbeat messages are sent over the HA1 link. For more information about the HA1 and HA2 interfaces and the kinds of messages communicated over each, see "Dual Link Probes" on page 1773.

Virtual Security Interfaces and Static Routes

After you create a VSD group, you must bind VSIs to the VSD. When you put a security device in an NSRP cluster, all the security zone interfaces become VSIs of VSD group 0. You must manually assign VSIs to VSDs with other IDs for each security zone configured on the security device.

By default, the security device adds an entry to its routing table for the immediate subnet of a VSI. For static routes to addresses beyond the immediate subnet, you must manually make route table entries for each VSI through which you want the security device to forward traffic to those addresses. For example, if you have two VSDs and you want to configure a default route to a router in the Untrust zone, you must make a routing table entry for the Untrust zone VSI of both VSDs. If you set the default route on only one VSD (for example, VSD 0), the security device acting as the primary device of the other VSD (for example, VSD 1) must pass all outbound traffic sent to it across the HA data link to the device acting as the primary device of VSD 0. See Figure 444 on page 1793.

ScreenOS 6.2.0 supports DHCP clients on a VSI interface and DHCP servers on a VSI sub-interface. An NSRP cluster can have many VSI interfaces configured as DHCP clients, so it is essential to have a unique client ID for each VSI interface. You set the client ID for a VSI interface through the WebUI or the CLI.

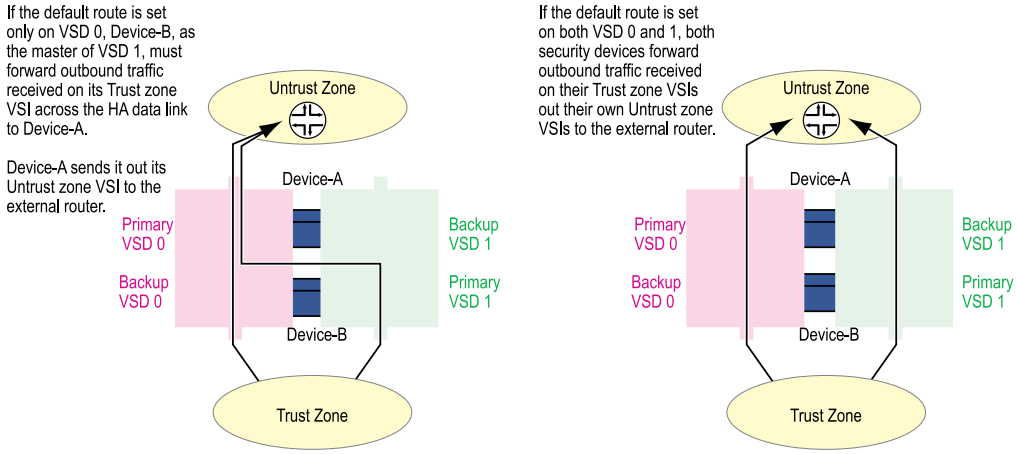
WebUI

Network > DHCP > Edit (for ethernet0/1) > DHCP Client: Enter the client ID, then click **OK:**

CLI

```
set interface interface dhcp client settings client-id string
save
```

Figure 444: Forwarding Traffic Through VSIs Using Static Routes



Configuration Examples

To configure two security devices for high availability (HA), you must cable them to the network and to each other and then configure them for HA using NSRP.

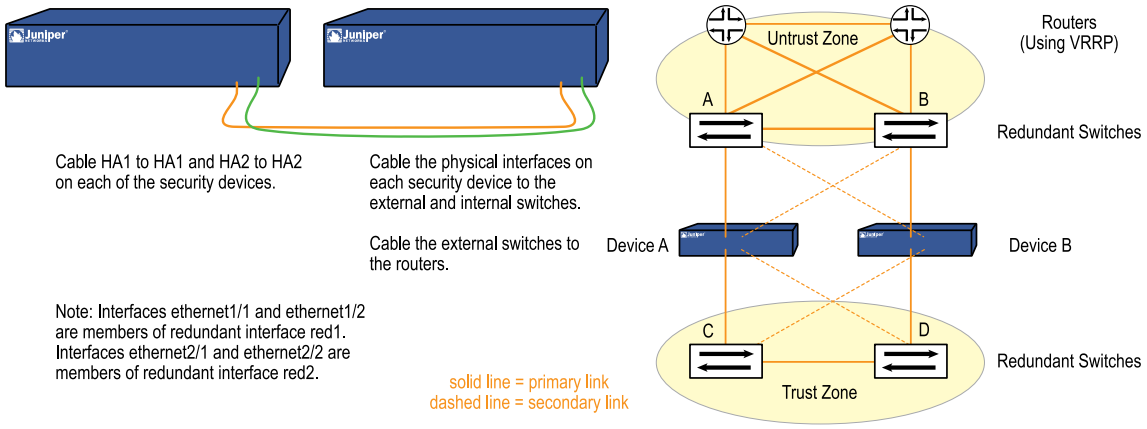
This section provides the following NSRP configuration examples:

Cabling Devices for Active/Active Full-Mesh NSRP

An active/active full-mesh configuration provides the highest level of availability, because it ensures that there is no single point of failure, whether it be a switch, router, or security device. Each device is wired twice to a connecting switch or router. Though not required, dual HA links are also connected between each security device. This allows each device to continue communication in case one of the HA links is severed. Figure 445 on page 1794 and Figure 446 on page 1795 illustrate the cabling of two security devices to each other and to redundant pairs of internal switches and external switches. The external switches are then cabled to a pair of redundant routers running VRRP, completing the full-mesh configuration. Figure 445 on page 1794 shows two security devices with dedicated HA interfaces. Figure 446 on page 1795 shows two security devices using network interfaces for HA traffic.

NOTE: Depending on the topology in which you are deploying the security devices and the kinds of switches and routers you use, the cabling presented in Figure 445 on page 1794 might differ from what your network requires.

Figure 445: Cabling Security Devices with Dedicated HA Interfaces



Cable two security devices (device A and device B) for NSRP in a full-mesh configuration as follows:

Device A and Device B: HA Links

1. Cable together the HA1 interfaces on each security device.
2. Cable together the HA2 interfaces on each security device.

Device A: Redundant1 (eth1/1 and eth1/2), Untrust Zone

3. Cable ethernet1/1 to external switch A. (ethernet1/1 is one of the two physical interfaces bound to the redundant interface red1 in the Untrust zone.)
4. Cable ethernet1/2 to external switch B. (ethernet1/2 is the other physical interface bound to red1 in the Untrust zone.)

Device A: Redundant2 (eth2/1 and eth2/2), Trust Zone

5. Cable ethernet2/1 to internal switch C. (ethernet2/1 is one of the two physical interfaces bound to the redundant interface red2 in the Trust zone.)
6. Cable ethernet2/2 to internal switch D. (ethernet2/2 is the other physical interface bound to red2 in the Trust zone.)

Device B: Redundant1 (eth1/1 and eth1/2), Untrust Zone

7. Cable ethernet1/1 to external switch B. (ethernet1/1 is one of the two physical interfaces bound to the redundant interface red1 in the Untrust zone.)
8. Cable ethernet1/2 to external switch A. (ethernet1/2 is the other physical interface bound to red1 in the Untrust zone.)

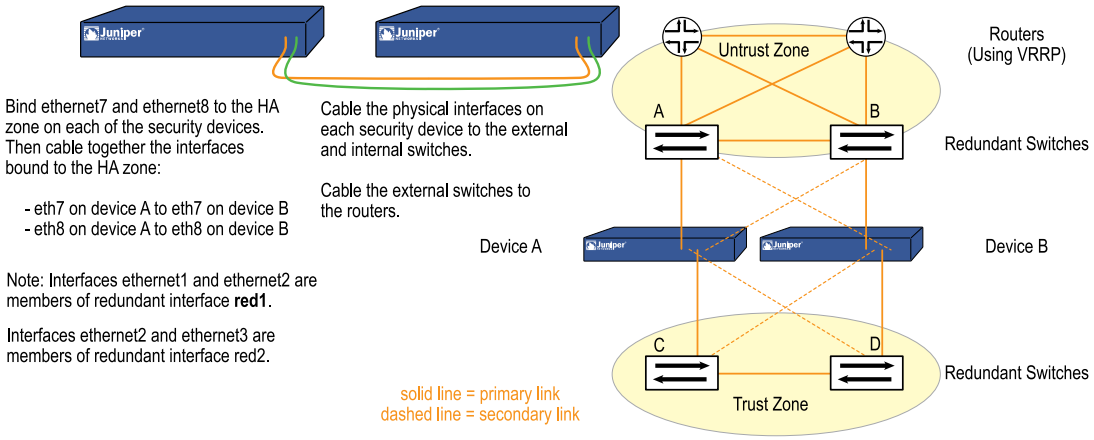
Device B: Redundant2 (eth2/1 and eth2/2), Trust Zone

9. Cable ethernet2/1 to internal switch D. (ethernet2/1 is one of the two physical interfaces bound to the redundant interface red2 in the Trust zone.)
10. Cable ethernet2/2 to internal switch C. (ethernet2/2 is the other physical interface bound to red2 in the Trust zone.)

Switches and Routers

11. Cable the external redundant switches together.
12. Cable the external switches to the redundant routers in the same configuration that you used to cable the security devices to the switches.
13. Cable the internal redundant switches together.

Figure 446: Security Devices Using Network Interfaces for HA Links



After binding ethernet7 and ethernet8 to the HA zone on both security devices (device A and device B), cable the devices for NSRP in a full-mesh configuration as follows:

Device A and Device B: HA Links

1. Cable together the ethernet7 interfaces on each security device.
2. Cable together the ethernet8 interfaces on each security device.

Device A: Redundant1 (ethernet1 and ethernet2), Untrust Zone

3. Cable ethernet1 to external switch A. (ethernet1 is one of the two physical interfaces bound to the redundant interface red1 in the Untrust zone.)
4. Cable ethernet2 to external switch B. (ethernet2 is the other physical interface bound to red1 in the Untrust zone.)

Device A: Redundant2 (ethernet3 and ethernet4), Trust Zone

5. Cable ethernet3 to internal switch C. (ethernet3 is one of the two physical interfaces bound to the redundant interface red2 in the Trust zone.)

6. Cable ethernet4 to internal switch D. (ethernet4 is the other physical interface bound to red2 in the Trust zone.)

Device B: Redundant1 (ethernet1 and ethernet2), Untrust Zone

7. Cable ethernet1 to external switch B. (ethernet1 is one of the two physical interfaces bound to the redundant interface red1 in the Untrust zone.)
8. Cable ethernet2 to external switch A. (ethernet2 is the other physical interface bound to red1 in the Untrust zone.)

Device B: Redundant2 (ethernet3 and ethernet4), Trust Zone

9. Cable ethernet3 to internal switch D. (ethernet3 is one of the two physical interfaces bound to the redundant interface red2 in the Trust zone.)
10. Cable ethernet4 to internal switch C. (ethernet4 is the other physical interface bound to red2 in the Trust zone.)

Switches and Routers

11. Cable the external redundant switches together.
12. Cable the external switches to the redundant routers in the same configuration that you used to cable the security devices to the switches.
13. Cable the internal redundant switches together.

Creating an NSRP Cluster

The reliability of your network is among the most important necessities of any organization. To ensure that your network is always available, your network devices must be able to failover to a redundant device. The most basic setup for this is to have an extra device ready to takeover in case the first device fails (Active/Passive). In ScreenOS, this redundancy is ensured through NSRP clusters. Because NSRP cluster members can have different hostnames, a failover can disrupt SNMP communication and the validity of digital certificates because SNMP communication and certificates rely on the hostname of a device to function properly.

To define a single name for all cluster members, type the following CLI command:

```
set nsrp cluster name name_str
```

NOTE: On devices that do not have a dedicated HA port, you must bind the interface to the HA zone before configuring the NSRP cluster.

Use the cluster name when configuring the SNMP hostname for the security device (**set snmp name** *name_str*) and when defining the common name in a PKCS10 certificate request file.

The use of a single name for all cluster members allows SNMP communication and digital certificate use to continue without interruption after a device failover.

In the example shown in Figure 447 on page 1797, you group devices A and B within NSRP cluster ID 1 with cluster name “cluster1.” You also specify the following settings on each device:

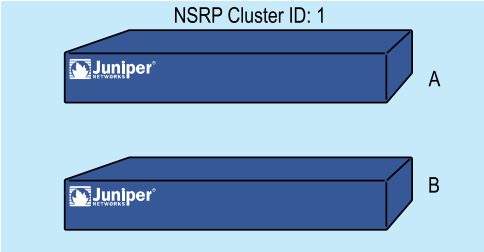
NSRP communication security: Assign passwords—725dCaIgDL and WiJoaw4177—for creating authentication and encryption keys to secure NSRP communications.

After you have grouped both devices in the same cluster and given them the same authentication and encryption passwords, you can enter the following settings on either device A or B. (Most settings entered on one device in a cluster propagate to the other device. For a list on non-propagating commands, see Table 127 on page 1776.)

- **Interface monitoring:** Select the ethernet1 (bound to the Untrust zone) and ethernet2 (bound to the Trust zone) for monitoring Layer 2 network connectivity.
- **Secondary link:** Specify that the ethernet2 interface carry VSD heartbeats should both HA1 and HA2 links go down. The purpose of this feature is to prevent multiple VSD group primary devices when both HA links fail.
- **Gratuitous ARP broadcasting:** Specify the number of ARP broadcasts as 5 (the default is 4). ARP broadcasts notify surrounding network devices of the MAC address of a new primary device after a failover has occurred.

(All the interfaces on these devices become VSIs for VSD group 0. For information about creating a second VSD group for these devices, see “Virtual Security Device Groups” on page 1788.)

Figure 447: NSRP Cluster



WebUI (Device A)

1. **NSRP Cluster and Communication Security**

   Network > NSRP > Cluster: Enter the following, then click **Apply**:

   Cluster ID: 1
   NSRP Authentication Password: (select) 725dCaIgDL
   NSRP Encryption Password: (select) WiJoaw4177

NOTE: You can only set a cluster name through the CLI.

WebUI (Device B)

2. **NSRP Cluster and Communication Security**

   Network > NSRP > Cluster: Enter the following, then click **Apply**:

   Cluster ID: 1
   Number of Gratuitous ARPs to Resend: 5
   NSRP Authentication Password: (select) 725dCaIgDL
   NSRP Encryption Password: (select) WiJoaw4177

3. **NSRP Settings**

   Network > NSRP > Monitor > Interface > VSD ID: Device Edit Interface: Select **ethernet1** and **ethernet2**, then click **Apply**.

   Network > NSRP > Link: Select **ethernet2** from the Secondary Link drop-down list, then click **Apply**.

CLI (Device A)

1. **NSRP Cluster and Communication Security**

```
set nsrp cluster id 1
set nsrp auth password 725dCaIgDL
set nsrp encrypt password WiJoaw4177
save
```

CLI (Device B)

2. **NSRP Cluster and Communication Security**

```
set nsrp cluster id 1
set nsrp auth password 725dCaIgDL
set nsrp encrypt password WiJoaw4177
save
```

3. **NSRP Settings**

```
set nsrp cluster name cluster1
set nsrp monitor interface ethernet1
set nsrp monitor interface ethernet2
set nsrp secondary-path ethernet2
set nsrp arp 5
save
```

Configuring an Active/Passive NSRP Cluster

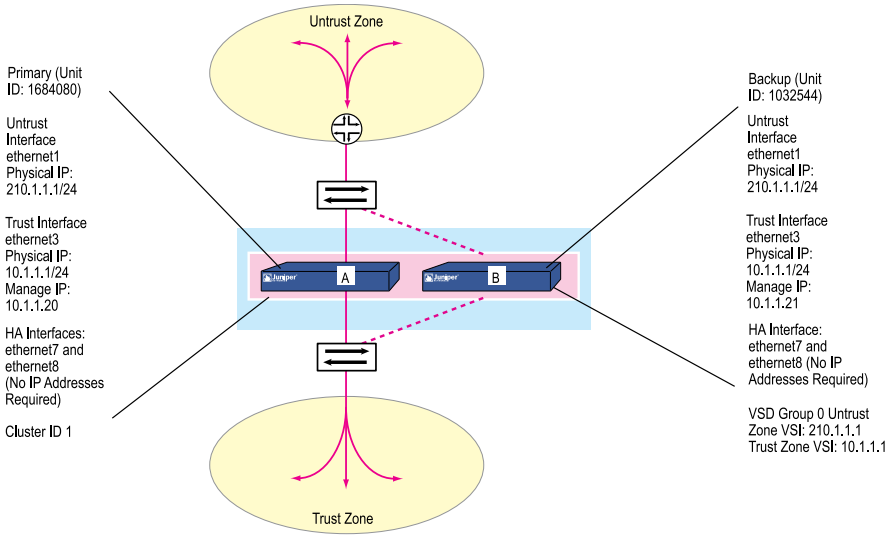
In the example shown in Figure 448 on page 1799, you cable ethernet7 on Device A to ethernet7 on Device B. You cable the ethernet8 interfaces likewise. Then you bind

ethernet7 and ethernet8 to the HA zone. You set Manage IP addresses for the Trust zone interfaces on both devices—10.1.1.20 for Device A and 10.1.1.21 for Device B. You then assign each device to NSRP cluster ID 1. When the devices become members of the NSRP cluster, the IP addresses of their physical interfaces automatically become the IP addresses of the VSIs for VSD group ID 0. Each VSD member has a default priority of 100, the device with the higher unit ID becomes the VSD group's primary device.

NOTE: By default, ethernet8 is bound to the HA zone. Binding it to the HA zone is only necessary if you have previously bound it to a different zone. Also, on devices that do not have a dedicated HA port, you must bind the interface to the HA zone before configuring the NSRP cluster.

You configure the devices to monitor ports ethernet1 and ethernet3, so that loss of network connectivity on either of those ports triggers a device failover. You also enable the automatic synchronization of RTOs.

Figure 448: Basic Active/Passive Configuration



WebUI (Device A)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet7): Enter the following, then click **OK**:

   Zone Name: HA

   Network > Interfaces > Edit (for ethernet8): Enter the following, then click **OK**:

   Zone Name: HA

   Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 210.1.1.1/24

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24
Manage IP: 10.1.1.20

Enter the following, then click **OK**:

Interface Mode: NAT

2. **NSRP**

Network > NSRP > Monitor > Interface > VSD ID: Device Edit Interface: Enter the following, then click **Apply**:

ethernet1: (select); Weight: 255
ethernet3: (select); Weight: 255

NOTE: The default setting for an NSRP failover threshold is 255. Therefore, if ethernet1 or ethernet3 fails with a weight of 255, its failure triggers a device failover.

Network > NSRP > Synchronization: Select **NSRP RTO Synchronization**, then click **Apply**.

NOTE: If you do not enable the automatic RTO synchronization option, you cannot manually synchronize RTOs with the CLI command **exec nsrp sync rto all**. The RTO will be dropped of you do not enable synchronization.

Network > NSRP > Cluster: In the Cluster ID field enter **1**, then click **Apply**.

WebUI (Device B)

3. **Interfaces**

Network > Interfaces > Edit (for ethernet7): Enter the following, then click **OK**:

Zone Name: HA

Network > Interfaces > Edit (for ethernet8): Enter the following, then click **OK**:

Zone Name: HA

Network > Interfaces > Edit (for ethernet1): Enter the following, then click **OK**:

Zone Name: Untrust
Static IP: (select this option when present)
IP Address/Netmask: 210.1.1.1/24

Network > Interfaces > Edit (for ethernet3): Enter the following, then click **Apply**:

Zone Name: Trust
Static IP: (select this option when present)
IP Address/Netmask: 10.1.1.1/24
Manage IP: 10.1.1.21

Enter the following, then click **OK**:

Interface Mode: NAT

4. **NSRP**

Network > NSRP > Monitor > Interface > VSD ID: Device Edit Interface: Enter the following, then click **Apply**:

ethernet1: (select); Weight: 255
ethernet3: (select); Weight: 255

Network > NSRP > Synchronization: Select **NSRP RTO Synchronization**, then click **Apply**.

Network > NSRP > Cluster: In the Cluster ID field enter **1**, then click **Apply**.

CLI (Device A)

1. **Interfaces**

```
set interface ethernet7 zone ha
set interface ethernet8 zone ha
set interface ethernet1 zone untrust
set interface ethernet1 ip 210.1.1.1/24
set interface ethernet3 zone trust
set interface ethernet3 ip 10.1.1.1/24
set interface ethernet3 manage-ip 10.1.1.20
set interface ethernet3 nat
```

2. **NSRP**

```
set nsrp rto-mirror sync
set nsrp monitor interface ethernet1
set nsrp monitor interface ethernet3
set nsrp cluster id 1
save
```

NOTE: If you do not enable the automatic RTO synchronization option, you cannot manually synchronize RTOs with the CLI command **exec nsrp sync rto all**.

The default weight for a monitored interface is 255 and the default NSRP failover threshold is 255. Therefore, if ethernet1 or ethernet3 fails with a weight of 255, its failure triggers a device failover. In the CLI, if you do not specify a weight for a monitored interface, the security device uses the default (255).

CLI (Device B)

3. **Interfaces**

```
set interface ethernet7 zone ha
set interface ethernet8 zone ha
set interface ethernet1 zone untrust
set interface ethernet1 ip 210.1.1.1/24
set interface ethernet3 zone trust
set interface ethernet3 ip 10.1.1.1/24
set interface ethernet3 manage-ip 10.1.1.21
set interface ethernet3 nat
```

4. **NSRP**

```
set nsrp rto-mirror sync
set nsrp monitor interface ethernet1
set nsrp monitor interface ethernet3
set nsrp cluster id 1
save
```

NOTE: After performing this configuration, type the **get nsrp** command to check the default NSRP settings that the device automatically creates, which are noted on "Basic NSRP Settings" on page 1769.

Configuring an Active/Active NSRP Cluster

After cabling the security devices together and to the surrounding network devices, you can then configure them for HA. A complete Active/Active configuration involves the following steps:

1. Creating an NSRP cluster, which automatically includes the creation of VSD group 0
2. Creating a second VSD group within the cluster
3. Enabling device failure tracking methods—such as interface monitoring and path monitoring

In the example shown in Figure 449 on page 1804, builds upon the interfaces configured in “Example” on page 1838, you create an NSRP cluster with ID 1 and name “cluster1” for two security devices—device A and device B—which do not have any other user-defined settings configured.

NOTE: To enable command propagation, you must first define the cluster ID number on each device. The following settings are not propagated and must be configured on each device in the cluster: VSD group, VSD priority, authentication and encryption passwords, Manage IP addresses, and IP tracking settings. All other commands are propagated among devices within the cluster.

When you create the NSRP cluster, the security device automatically creates VSD group 0. You define VSD group 1. You assign device A priority 1 in VSD group 0 and priority 100 (the default) in VSD group 1. You assign device B priority 1 in VSD group 1 and leave its priority at the default (100) in VSD group 0.

NOTE: The VSD group ID “0” does not appear in the names of VSIs in VSD 0. Instead of *redundant1:0,* the VSI is identified simply as *redundant1*.

You set the interface monitoring option to monitor the two redundant interfaces—redundant1 and redundant2—for Layer 2 network connectivity. If the primary physical interface for either of the monitored interfaces fails, the device immediately fails over to the secondary interface. If both physical interfaces comprising the members of a monitored redundant interface fail, the device fails over to the other device.

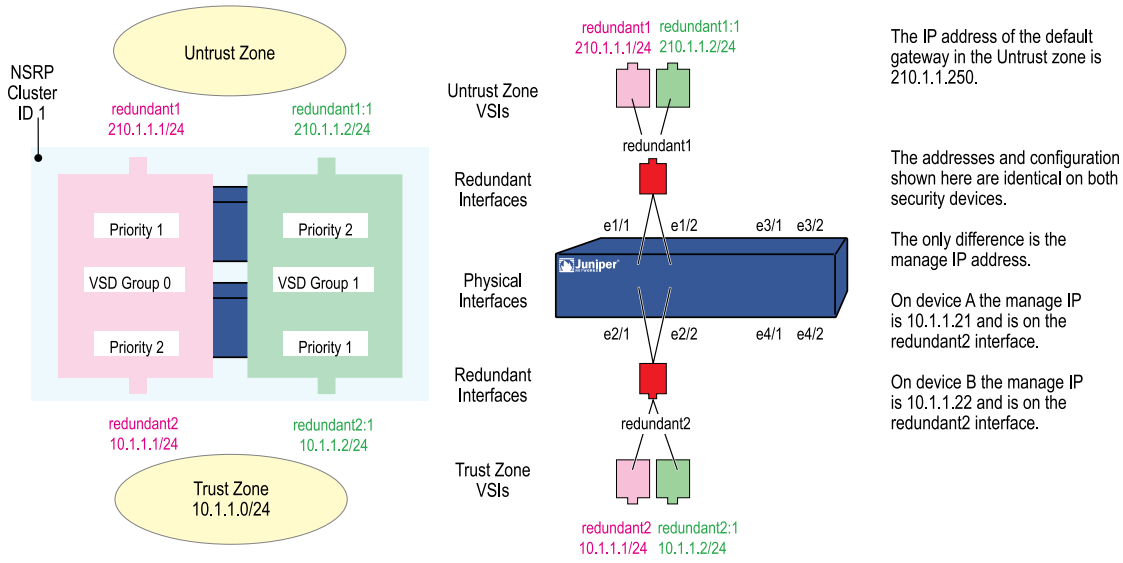
NOTE: Redundant interfaces are optional and not required for Active/Active configurations.

You define the ethernet2/1 interface as a secondary link for VSD heartbeat messages and the number of gratuitous ARPs after a device failover has occurred to 5. Because HA cables run directly between the two security devices, communication between members of the NSRP cluster is neither authenticated nor encrypted.

You also set a route to the default gateway (210.1.1.250) for each Untrust zone VSI, and a route to the internal network for each Trust zone VSI. All security zones are in the trust-vr routing domain.

Finally, after the configurations for both devices are synchronized, you enable RTO synchronization.

Figure 449: Active/Passive NSRP Configuration



WebUI (Device A)

1. **Cluster and VSD Groups**

   Network > NSRP > Cluster: Type **1** in the Cluster ID field, then click **Apply**.

   Network > NSRP > VSD Group > Edit (for Group ID 0): Enter the following, then click **OK**:

   Priority: 1
   Enable Preempt: (select)
   Preempt Hold-Down Time (sec): 10

NOTE: The hold-down time can be any length from 0 to 255 seconds, effectively delaying the failover to prevent a flurry of rapid failovers.

   Network > NSRP > VSD Group > New: Enter the following, then click **OK**:

   Group ID: 1
   Priority: 100
   Enable Preempt: (clear)
   Preempt Hold-Down Time (sec): 0

WebUI (Device B)

2. **Cluster and VSD Groups**

   Network > NSRP > Cluster: Enter the following, then click **Apply**:

Cluster ID: 1
Number of Gratuitous ARPs to Resend: 5

NOTE: You can only set the cluster name through the CLI.

This setting specifies that after a device failover, the new VSD group primary device sends five gratuitous ARP packets announcing the association of the VSI and virtual MAC address to the new primary device.

Network > NSRP > Link: Select **ethernet2/1** from the Secondary Link drop-down list, then click **Apply**.

NOTE: If both HA1 and HA2 links are lost, the VSD heartbeat messages pass via the ethernet2/1 in the Trust zone.

Network > NSRP > Synchronization: Select **NSRP RTO Synchronization**, then click **Apply**.

Network > NSRP > VSD Group > New: Enter the following, then click **OK**:

Group ID: 1
Priority: 1
Enable Preempt: (select)
Preempt Hold-Down Time (sec): 10

3. **Redundant Interfaces and Manage IP**

Network > Interfaces > New Redundant IF: Enter the following, then click **OK**:

Interface Name: redundant1
Zone Name: Untrust
IP Address / Netmask: 210.1.1.1/24

Network > Interfaces > Edit (for ethernet1/1): Select **redundant1** in the "As member of" drop-down list, then click **OK**.

Network > Interfaces > Edit (for ethernet1/2): Select **redundant1** in the "As member of" drop-down list, then click **OK**.

Network > Interfaces > New Redundant IF: Enter the following, then click **Apply**:

Interface Name: redundant2
Zone Name: Trust
IP Address / Netmask: 10.1.1.1/24
> Enter **10.1.1.22** in the Manage IP field, then click **OK**.

Network > Interfaces > Edit (for ethernet2/1): Select **redundant2** in the "As member of" drop-down list, then click **OK**.

Network > Interfaces > Edit (for ethernet2/2): Select **redundant2** in the "As member of" drop-down list, then click **OK**.

Network > NSRP > Monitor > Interface > VSD ID: Device Edit Interface: Select **redundant1** and **redundant2**, then click **Apply**.

4. **Virtual Security Interfaces**

   Network > Interfaces > New VSI IF: Enter the following, then click **OK**:

   Interface Name: VSI Base: redundant1
   VSD Group: 1
   IP Address / Netmask: 210.1.1.2/24

   Network > Interfaces > New VSI IF: Enter the following, then click **OK**:

   Interface Name: VSI Base: redundant2
   VSD Group: 1
   IP Address / Netmask: 10.1.1.2/24

5. **Routes**

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   Network Address/Netmask: 0.0.0.0/0
   Gateway: (select)
   Interface: redundant1
   Gateway IP Address: 210.1.1.250

   Network > Routing > Routing Entries > trust-vr New: Enter the following, then click **OK**:

   Network Address: 0.0.0.0/0
   Gateway: (select)
   Interface: redundant1:1
   Gateway IP Address: 210.1.1.250

WebUI (Device A)

6. **Manage IP Address**

   Network > Interfaces > Edit (for redundant2): Enter **10.1.1.21** in the Manage IP field, then click **OK**.

7. **RTO Synchronization**

   Network > NSRP > Synchronization: Select **NSRP RTO Mirror Synchronization**, then click **Apply**.

CLI (Device A)

1. **Cluster and VSD Groups**

```
set nsrp cluster id 1
set nsrp vsd-group id 0 preempt hold-down 10
set nsrp vsd-group id 0 preempt
set nsrp vsd-group id 0 priority 1
set nsrp vsd-group id 1
set nsrp rto-mirror sync
save
```

NOTE: The hold-down time can range from 0 to 255 seconds, effectively delaying the failover to prevent a flurry of rapid failovers.

CLI (Device B)

2. **Cluster and VSD Groups**

```
set nsrp cluster id 1
set nsrp cluster name cluster1
set nsrp rto-mirror sync
set nsrp vsd-group id 1 priority 1
set nsrp vsd-group id 1 preempt hold-down 10
set nsrp vsd-group id 1 preempt
set nsrp secondary-path ethernet2/1
set nsrp arp 5
set arp always-on-dest
```

NOTE: Because devices A and B are both members of the same NSRP cluster, all subsequent commands (except those otherwise noted) that you enter on device B propagate to device A.

In the example, the commands **set nsrp vsd-group id 1 priority 1** and **set nsrp vsd-group id 1 preempt hold-down 10** are not propagated.

If both HA1 and HA2 links are lost, the VSD heartbeat messages pass via the ethernet2/1 in the Trust zone.

The **set nsrp arp 5** setting specifies that, after a device failover, the new VSD group primary device sends five gratuitous ARP packets announcing the association of the VSI and virtual MAC address to the new primary device.

After you enter the **set arp always-on-dest** command, the security device always does an ARP lookup to learn a destination MAC address instead of learning it from the source MAC in the originating ethernet frame. The external routers in this example are grouped as a virtual router running VRRP. Frames coming from this router use the virtual IP address as the source IP but the physical MAC address as the source MAC. If the router fails over and the security device has learned the MAC from the source MAC in the incoming frame, it would then direct return traffic to the wrong location. By doing an ARP lookup for the destination MAC, the security device can properly send traffic to the location of the new physical MAC address.

3. **Redundant Interfaces and Manage IP**

```
set interface redundant1 zone untrust
set interface redundant1 ip 210.1.1.1/24
set interface ethernet1/1 group redundant1
set interface ethernet1/2 group redundant1
set interface redundant2 zone trust
set interface redundant2 ip 10.1.1.1/24
set interface redundant2 manage-ip 10.1.1.22
set interface ethernet2/1 group redundant2
set interface ethernet2/2 group redundant2
set nsrp monitor interface redundant1
set nsrp monitor interface redundant2
```

4. **Virtual Security Interfaces**

```
set interface redundant1:1 ip 210.1.1.2/24
set interface redundant2:1 ip 10.1.1.2/24
```

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface redundant1 gateway 210.1.1.250
set vrouter trust-vr route 0.0.0.0/0 interface redundant1:1 gateway 210.1.1.250
save
```

CLI (Device A)

6. **Manage IP Address**

```
set interface redundant2 manage-ip 10.1.1.21
```

7. **RTO Synchronization**

```
set nsrp rto-mirror sync
save
```

Synchronizing RTOs Manually

There are situations in which RTOs on your devices will become out of sync and you do not have RTO synchronization enabled. In cases like this, you must manually synchronize RTOs in order to ensure that all dynamic information is mirrored across your devices.

In this example, devices A and B are in NSRP cluster 1 and VSD groups 1 and 2. Device A is the primary device of VSD group 1 and the backup in VSD group 2. Device B is the primary device of VSD group 2 and the backup in VSD group 1.

You want to do some troubleshooting on device B, and you do not want to disconnect it from the network. You force device B to become the backup in VSD group 2, and then you disable RTO synchronization. Device A becomes the primary device of both VSD groups. After you finish troubleshooting device B, you again enable RTO mirror synchronization and then manually resync the RTOs from device A to device B. Finally, you reassign device B as the primary device of VSD group 2.

WebUI

NOTE: The manual synchronization of RTOs is only available through the CLI.

CLI

Device B

```
exec nsrp vsd-group id 2 mode backup
unset nsrp rto-mirror sync
```

Device B is no longer processing traffic nor synchronizing RTOs with device A. At this point, you can troubleshoot device B without affecting the traffic-processing performance of device A.

```
set nsrp rto-mirror sync
exec nsrp sync rto all from peer
exec nsrp vsd-group id 2 mode master
```

Configuring Manual Link Probes

In this example, the ethernet7 and ethernet8 interfaces on the security device are bound to the HA zone. You configure 5 link probes to be sent on the ethernet8 interface to the peer's MAC address 00e02000080. (Note that if you do not specify a MAC address, the default NSRP MAC address is used.)

WebUI

NOTE: You must use the CLI to send probes manually on an HA link.

CLI

```
exec nsrp probe ethernet8 00e02000080 count 5
```

Configuring Automatic Link Probes

There are cases in which the HA link may actually be the weakest link in an NSRP configurations. That is because the link which may be down seems to be active for a Juniper device. In cases like this, it is important to consistently check the state of not only each device but also the HA links connecting them. In this example, the ethernet7 and ethernet8 interfaces on the security device are bound to the HA zone. You configure link probes to be automatically sent to both interfaces at three-second intervals. You also set the threshold value so that if there is no reply from the peer after sending four consecutive requests, the HA link is considered to be down.

WebUI

Network > NSRP > Link: Enter the following, then click **Apply**:

```
Enable HA Link Probe: (select)
Interval: 3
Threshold: 5
```

CLI

```
set nsrp ha-link probe interval 3 threshold 5
```

Configuring NSRP in an IPv6 Environment

In this example, you configure an NSRP cluster on an IPv6 interface. First, you configure an Active/Active NSRP cluster, then you configure the IPv6 interface. Finally, the IPv6 related configuration is synchronized to its NSRP peer.

NOTE: ScreenOS does not support NSRP IPv6 related RTO synchronization. This example explains only about the configuration synchronization.

Configuring an Active/Active NSRP Cluster

Create an NSRP cluster with cluster ID 15 and a VSD group with VSD ID 0, 5. You can create the NSRP cluster and VSD group using the WebUI or the CLI.

WebUI

Network > NSRP > Cluster: Enter the following, then click **Apply:**

Cluster ID: 15

Network > NSRP > VSD Group > Edit (for Group ID 0): Enter the following, then click **OK:**

Priority: 50
Enable Preempt: (select)
Preempt Hold-Down Time (sec): 10

Network > NSRP > VSD Group > New: Enter the following, then click **OK:**

Group ID: 5
Priority: 100
Enable Preempt: (select)
Preempt Hold-Down Time (sec): 10

CLI

```
set nsrp cluster id 15
set nsrp vsd-group id 0 preempt hold-down 10
set nsrp vsd-group id 0 preempt
set nsrp vsd-group id 0 priority 50
set nsrp vsd-group id 5 preempt hold-down 10
set nsrp vsd-group id 5 preempt
set nsrp vsd-group id 5 priority 100
save
```

Configuring the IPv6 Environment

The next step is to configure an IPv6 interface and configure the Neighbor Discovery (ND) and Router Advertisement (RA) settings for the interface. After this step, NSRP will synchronize the IPv6 related configuration.

WebUI

Network > Interfaces: Select Edit to change the IPv6 mode for an existing interface entry or click New to configure a new interface entry, then click **Apply.**

Enable IPv6: (select)
Mode: Router
Interface ID (64-bit): 0000000000000001
Unicast Address1/Prefix: 1::1/64
Path MTU (IPv6): (select)
NUD (Neighbor Unreachability Detection): (select)

ND/RA Configuration

Base Reachable Time:
Probe Time: 20
Retransmission Time: 9
Allow RA Transmission: (select)

CLI

```
set interface ethernet1/1 ipv6 mode router
set interface ethernet1/1 ipv6 enable
set interface ethernet1/1 ipv6 interface-id 0000000000000001
set interface ethernet1/1 ipv6 ip 1::1/64
set interface ethernet1/1:5 ip 1.1.1.1/24
set interface ethernet1/1:5 ipv6 mode router
set interface ethernet1/1:5 ipv6 enable
set interface ethernet1/1:5 ipv6 ip 1::2/64
set interface loopback.1 zone trust
set interface loopback.1 ipv6 mode host
set interface loopback.1 ipv6 enable
set interface loopback.1 ipv6 ip 2::2/128
set interface loopback.1 ipv6 ip share-prefix interface e1/1
```

ND/RA Configuration

```
set interface ethernet1/1 ipv6 nd base-reachable-time 20
set interface ethernet1/1 ipv6 nd nud
set interface ethernet1/1 ipv6 nd probe-time 20
set interface ethernet1/1 ipv6 nd retransmit-time 9
set interface ethernet1/1 ipv6 ra transmit
set interface ethernet1/1 dhcp6 server/client
set interface ethernet1/1 pmtu ipv6
save
```

Resetting the Configuration

You can reset the configuration using the following CLI.

CLI

1. **Deleting the cluster ID**

   ```
   unset nsrp cluster id
   ```

2. **Deleting the VSD groups**

```
unset nsrp vsd id 0
unset nsrp vsd id 5
```

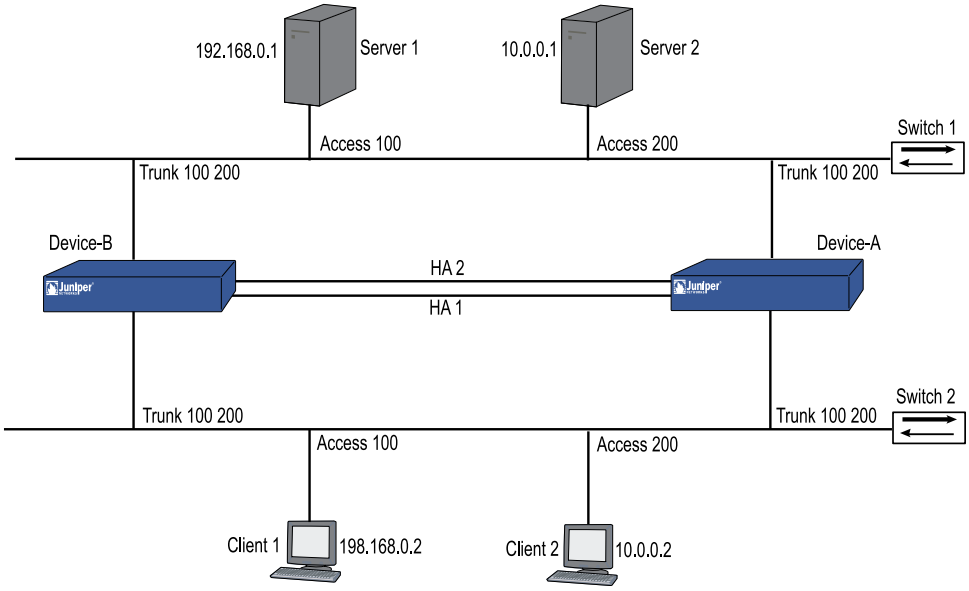
3. **Deleting the ethernet1/1 and loopback interfaces**

```
unset interface ethernet1/1:5 ipv6 mode
unset interface ethernet1/1:5 ipv6 en
unset interface ethernet1/1:5
unset interface ethernet1/1 ipv6 mode
unset interface ethernet1/1 ipv6 en
unset interface loopback.1 ipv6 mode
unset interface loopback.1 ipv6 en
unset interface loopback.1
save
```

Configuring Active/Active NSRP in Transparent Mode

In the example shown in Figure 450 on page 1813, you configure an Active/Active NSRP on two security devices – Device A and Device B – in transparent mode. You create two VSD groups—VSD 5 and VSD 7—and verify that Device A is the master and Device B the backup in VSD 5 and that Device B is the master and Device A the backup in VSD 7. Verify that both devices are members of the NSRP and check if the VSI Media Access Control (MAC) address of the NSRP devices is correct. The configuration can be synchronized to its NSRP Peer and the traffic belonging to different VLANs can be passed from each VSD.

Figure 450: Active/Passive NSRP Configuration in Transparent Mode



First, you configure the Active/Active NSRP on Device A and Device B. Second, you configure transparent mode on both devices. You can use the WebUI or the CLI to configure the Active/Active NSRP.

WebUI (Device A)

1. **Cluster and VSD Groups**

   Network > NSRP > Cluster: Enter the following, then click **Apply:**

   Cluster ID: 1

   Network > NSRP > VSD Group > Edit (for Group ID 0): Enter the following, then click **OK:**

   Priority: 1
   Enable Preempt: (select)
   Preempt Hold-Down Time (sec): 10

   Network > NSRP > VSD Group > New: Enter the following, then click **OK:**

   Group ID: 5
   Priority: 50
   Enable Preempt: (clear)
   Preempt Hold-Down Time (sec): 0

2. **VLAN and VLAN Group**

   Network > Vlan > Group > Edit: Enter the following, then click **Add:**

   Vlan Group Name: v100
   Assign Vlan ID Start: 100
   End: 100

3. **Bind VLAN Group and VSD Group**

   Network > Vlan > VSD Binding > Edit: Enter the following, then click **OK:**

   Vlan Group Name: v100
   VSD Group ID: 5

WebUI (Device B)

4. **Cluster and VSD Groups**

   Network > NSRP > Cluster: Enter the following, then click **Apply:**

   Cluster ID: 1

   Network > NSRP > VSD Group > New: Enter the following, then click **OK:**

   Group ID: 7
   Priority: 100
   Enable Preempt: (select)
   Preempt Hold-Down Time (sec): 0

5. **VLAN and VLAN Group**

Network > Vlan > Group > Edit: Enter the following, then click **Add:**

Vlan Group Name: v200
Assign Vlan ID Start: 200
End: 200

6. **Bind VLAN Group and VSD Group**

Network > Vlan > VSD Binding > Edit: Enter the following, then click **OK:**

Vlan Group Name: v200
VSD Group ID: 7

7. **VLAN Group and Zone**

Network > Vlan > Group > Edit (Port): Enter the following, then click **Add:**

Port: ethernet2/1
Zone: l2-aa-trust

Network > Vlan > Group > Edit (Port): Enter the following, then click **Add:**

Port: ethernet2/2
Zone: l2-aa-untrust

CLI

1. **Cluster and VSD Groups for Device A**

```
set interface ethernet2/7 zone ha
set interface ethernet2/8 zone ha
set nsrp cluster id 7
unset nsrp vsd id 0
set nsrp vsd id 5 priority 50
set nsrp vsd id 5 preempt
set nsrp vsd id 7 priority 100
set nsrp vsd id 7 preempt
set nsrp rto-mirror sync
```

2. **Cluster and VSD Groups for Device B**

```
set interface ethernet2/7 zone ha
set interface ethernet2/8 zone ha
set nsrp cluster id 7
unset nsrp vsd id 0
set nsrp vsd id 7 priority 100
set nsrp vsd id 7 preempt
set nsrp vsd id 7 priority 50
set nsrp vsd id 7 preempt
set nsrp rto-mirror sync
```

3. **Create VLAN Group v100 and Assign It to VSD 5:**

```
set vlan group name v100
set vlan group v100 100
set vlan group v100 vsd id 5
```

4. **Create VLAN Group v200 and Assign It to VSD 7:**

```
set vlan group name v200
set vlan group v200 200
set vlan group v200 vsd id 7
```

5. **Create L2 zone and Assign the VLAN Group to Different Zones:**

```
set zone name l2-aa-trust l2
set zone name l2-aa-untrust l2
set vlan port ethernet2/1 group v100 zone L2-aa-trust
set vlan port ethernet2/2 group v100 zone L2-aa-untrust
set vlan port ethernet2/1 group v200 zone L2-aa-trust
set vlan port ethernet2/2 group v200 zone L2-aa-untrust
```

6. **Create Policy on Both Directions:**

```
set policy from l2-aa-untrust to l2-aa-trust any any any permit
set policy from l2-aa-trust to l2-aa-untrust any any any permit
save
```

Chapter 57

Interface Redundancy and Failover

This chapter describes the various ways in which security devices provide interface redundancy and failover. It contains the following sections:

Redundant Interfaces and Zones

For HA interface redundancy, Juniper Networks security devices either provide dedicated physical redundant HA interfaces or allow you to bind two generic interfaces to the HA zone. You can also create redundant security zone interfaces, as described in this section.

Applying a similar kind of virtualization that allows a VSI to shift its binding from the physical interface on one device to the physical interface on another device, the redundant interface can shift its binding from one physical interface to another physical interface on the same device. For example, if the link from the primary interface to the switch becomes disconnected, the link fails over to the secondary interface, which prevents device failover from the VSD primary device to the backup device.

NOTE: Using aggregate or redundant interfaces in an active-active HA pair of ISG 2000 systems does not allow traffic to pass through the device in the packet forwarding case. In the case of NS5000 8G and 8G2, the ports you want to combine into an aggregate or a redundant interface should be from the same ASIC, either 1–4 or 5–8. For more information on aggregate port details for 8G and 8G2 SPMs, see "Configuring Layer 2 Virtual Systems" on page 1726.

You can create a redundant interface with the WebUI or the CLI.

WebUI

Network > Interfaces > New Redundant IF: Enter the following, then click **OK**:

Interface Name: redundant1
Zone Name: Untrust
IP Address / Netmask: 210.1.1.1/24

CLI

```
set interface redundant1 zone untrust
```

Holddown Time Settings

You can also set a holddown time for a physical interface to wait before becoming the primary interface after an interface failover occurs. To set a holddown time for a member of a redundant interface, use the following command, in which the interface name is that of a physical interface: **set interface** *interface* **phy holddown** *number*. See Figure 451 on page 1819

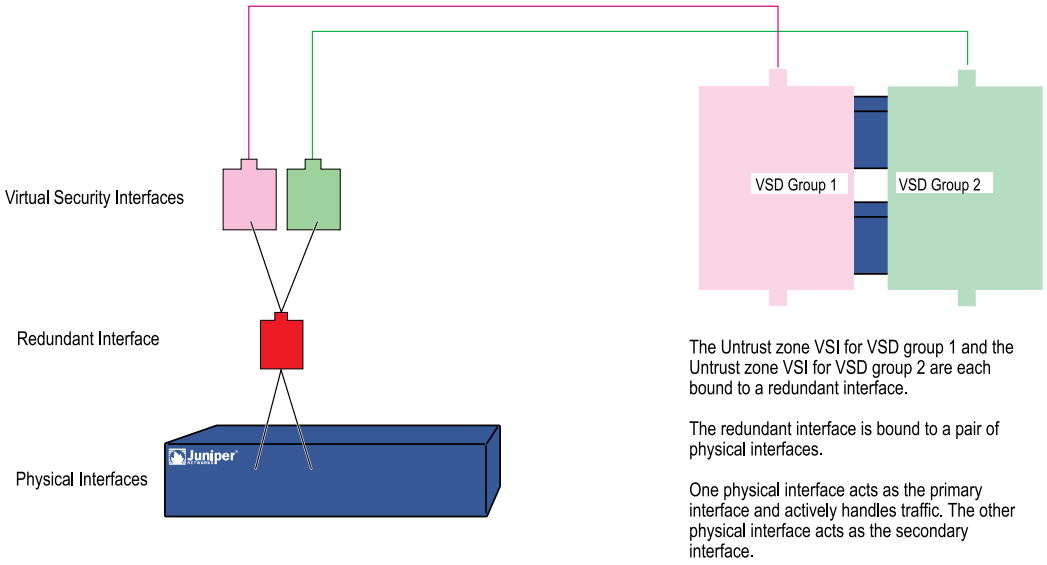
NOTE: You must enter this command before making the interface a member of a redundant group.

You can bind a VSI to any of the following interface types:

- A subinterface
- A physical interface
- A redundant interface, which in turn is bound to two physical interfaces
- A loopback interface

NOTE: You cannot group subinterfaces or a loopback interface to a redundant interface. However, you can define a VLAN on a redundant interface in the same way that you can define a VLAN on a subinterface.

Figure 451: Relationship of Physical, Redundant, and Virtual Security Interfaces



Aggregate Interfaces

Some system platforms, such as the Integrated Security Gateway (ISG) systems and the NetScreen-5000 systems, allow you to combine the throughput of one or more pairs of physical ports into a single virtual port. This virtual port is known as an *aggregate interface*. Only Secure Port Modules (SPMs) support this feature, and you can only aggregate side-by-side ports that reside on the same module.

NOTE: Aggregation is not allowed across I/O modules.

You can aggregate two 2 Gigabit ports to make a single full-duplex 4 Gigabit pipe, or you can aggregate eight Fast Ethernet ports into a single full-duplex 1.6 Gigabit pipe.

You must assign one of the following names to the aggregate interface: **aggregate1**, **aggregate2**, **aggregate3**, or **aggregate4**.

NOTE: As with most other ports and interfaces, you must assign the aggregate interface an IP address so that other hosts on the network can reach it.

In the following example, you combine two Gigabit Ethernet mini-GBIC ports, each running at 1-Gbps, into an aggregate interface aggregate1 running at 2-Gbps. The aggregate interface consists of Ethernet ports 1 and 2 on a 5000-8G SPM (residing in Slot 2) and is bound to the Trust zone.

NOTE: To see the physical ports that are available on the system, go to the Network > Interfaces screen in the WebUI or enter the CLI command **get interface**.

WebUI

Network > Interfaces > Aggregate IF > New: Enter the following, then click **Apply**:

Interface Name: aggregate1
Zone Name: Trust (select)
IP Address / Netmask: 10.1.1.0/24

Enter the following, then click **OK**:

Interface Mode: NAT

Network > Interfaces > Edit (for ethernet2/1): Enter the following, then click **OK**:

As member of: aggregate1 (select)

Network > Interfaces > Edit (for ethernet2/2): Enter the following, then click **OK**:

As member of: aggregate1 (select)

CLI

```
set interface aggregate1 zone trust
set interface aggregate1 ip 10.1.1.0/24
set interface aggregate1 nat
set interface ethernet2/1 aggregate aggregate1
set interface ethernet2/2 aggregate aggregate1
save
```

ScreenOS supports round-robin mode for distribution of traffic on an aggregate interface. In round-robin (RR) mode, the cross ASIC traffic in the security device takes a different path on each flow, resulting in a nonsequential outflow of packets from the security device. To overcome the drawback in RR mode, the current release of ScreenOS uses the hashing mode technology on Netscreen 5000-8G2 systems to direct all traffic from the same session to flow through a single path.

In hash mode, all packets from the same session flow through the same channel and same member port of the aggregate interface. Similarly, you can see packets from different sessions directed to different member ports of an aggregate interface for maximum throughput. The hashing mode of traffic distribution not only ensures the sequential forwarding of packets from a security device, but also provides load balancing for traffic from multiple sessions.

Interface Failover

When there are both primary and backup interfaces bound to the Untrust zone, you can manually force traffic from the primary interface to the backup interface through the WebUI or the CLI. You can also configure the security device to automatically

forward traffic to the backup interface if ScreenOS detects a failure on the primary interface connection.

Backup Interface Traffic

In this example, you manually force traffic from the primary interface to the backup interface.

WebUI

Network > Untrust Failover: Select **Failover**, then click **Apply**. Then click **Force to Failover**.

CLI

```
set failover enable
save
exec failover force
```

When the primary interface is again available, you need to use the WebUI or the CLI to switch traffic from the backup to the primary interface.

Primary Interface Traffic

In the previous example, you forced a failover from the primary to the backup interface. In this example, you manually force traffic from the backup interface to revert to the primary interface.

WebUI

Network > Untrust Failover: Click **Force to Revert**.

CLI

exec failover revert

Automatic Traffic Failover

In this example, you configure the security device to fail over traffic automatically to the backup interface if the security device detects an IP tracking failure on the primary interface. When IP tracking on the primary interface again succeeds, the security device automatically reverts traffic from the backup to the primary interface.

NOTE: For information about setting IP tracking to trigger an interface failover, see “Interface Failover with IP Tracking” on page 1825.

By default, there is a 30-second interval (holddown time) between the time that the IP tracking failure threshold occurs and the interface failover occurs. The purpose of the holddown time is to avoid unnecessary failovers that intermittent latency or

interference in the network might cause. In this example, you shorten the holddown time to 20 seconds.

WebUI

Network > Untrust Failover: Select the following, then click **Apply**:

Track IP: (select)
Automatic Failover: (select)
Failover: (select)
Failover Holddown Time: 20

CLI

```
set failover type track-ip
set failover auto
set failover enable
set failover holddown 20
save
```

Serial Interfaces

By default, you must use the WebUI or CLI to force ScreenOS to switch over to the serial interface when the primary interface (Untrust or ethernet3 interface) connection fails and to switch back to the primary interface when the primary is again available. You can configure the interface failover to be automatic. You can also configure IP tracking to monitor failure on the Untrust or ethernet3 interfaces.

By default, policies that are enabled for traffic from the Trust zone to the Untrust zone or from the Untrust zone to the Trust zone are still active after a failover to the serial interface. But traffic through the primary interface could be so heavy that it cannot be handled by the dialup link. When you define a policy, you can specify whether or not the policy should be active if ScreenOS switches to the serial interface. See “Policy Deactivation” on page 1823 for information on how to configure this in the WebUI and the CLI.

The serial interface is bound by default to the Null zone and you need to explicitly bind it to the Untrust zone to use it as a backup interface. If you bind the serial interface to the Untrust zone using the WebUI, ScreenOS automatically adds a default route for the serial interface. If you bind the serial interface to the Untrust zone using the CLI, ScreenOS does not add a default route to the serial interface and you must explicitly add a default route for the serial interface if traffic is to be routed through the serial interface. See “Default Route Deletion” on page 1822 for information on how to configure this in the WebUI and the CLI.

Default Route Deletion

If you bind the serial interface to the Untrust zone using the WebUI, ScreenOS automatically adds a default route for the serial interface. In this example, you use the WebUI to bind the serial interface to the Untrust zone. You then delete the default route that has been automatically created for the serial interface.

WebUI

Network > Interfaces > Edit (for serial): Enter the following, then click **OK**:

Zone Name: (select) Untrust

Network > Routing > Routing Entries: In the Configure column, click **Remove** for the default route to 0.0.0.0/0 through the serial interface.

Default Route Addition

If you bind the serial interface to the Untrust zone using the CLI, ScreenOS does not add a default route to the serial interface and you must explicitly add a default route for the serial interface if you want the security device to route traffic through the serial interface. In this example, you use the CLI to bind the serial interface to the Untrust zone. You then add a default route for the serial interface, which is bound to the Untrust zone.

CLI

```
set interface serial zone untrust
set vrouter trust-vr route 0.0.0.0/0 interface serial
save
```

Policy Deactivation

In this example, normal traffic through the primary interface (ethernet3) to the Untrust zone includes large files transferred via FTP from host22 in the Trust zone to ftp_srv in the Untrust zone. If a failover to the serial interface occurs, the dialup link might drop such large FTP traffic. Whenever there is a failover to the serial interface, any policy that is configured to be inactive for the serial interface becomes invalid and the policy lookup procedure continues to the next policy.

WebUI

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), host22
Destination Address:
Address Book Entry: (select), ftp_srv
Service: FTP
Action: Permit

> Advanced: Clear **Valid for Serial**, then click **Return** to set the advanced options and return to the basic configuration page.

CLI

```
set policy from trust to untrust host22 ftp_srv ftp permit no-session-backup
save
```

Monitoring Failover

An interface failover can occur when ScreenOS detects a physical link problem on the primary interface connection, such as an unplugged cable. You can also define the following types of interface failover:

- When certain IP addresses become unreachable through a given interface using IP tracking
- When certain VPN tunnels on the primary Untrust interface become unreachable using VPN tunnel monitoring

The interface failover sequence occurs as follows:

1. The security device determines that interface monitoring on the primary interface has failed. The interface might be physically disconnected or there might be a failure with IP tracking or VPN monitoring.
2. The security device waits until the failover holddown time elapses.
3. When the failover holddown time has expired, the state of the primary interface changes from up to down, the state of the backup interface changes from down to up, and the security device reroutes traffic using the primary interface to the backup interface.
4. The security device connects to its ISP using DHCP or PPPoE on the now-activated backup interface.

NOTE: The security device can initiate a new PPPoE connection after it receives new outbound traffic or immediately after the failover occurs (**set pppoe name** name **auto-connect**).

The recovery sequence is essentially in reverse order from the failover sequence:

1. The security device determines that interface monitoring on the primary interface has succeeded. The interface might be physically reconnected, or IP tracking or VPN monitoring might have succeeded again.
2. The security device waits until the failover holddown time elapses.
3. When the failover holddown time has expired, the state of the backup interface changes from up to down, the state of the primary interface changes from down to up, and the security device reroutes traffic using the backup interface to the primary interface.
4. The security device connects to its ISP using DHCP or PPPoE on the now-reactivated primary interface.

NOTE: The ISP to which the security device connects on the primary interface can be the same one as or a different one from the ISP it connects to on the backup interface.

Interface Failover with IP Tracking

You can specify that when certain IP addresses become unreachable through the primary Untrust zone interface, the security device fails over to the backup Untrust zone interface even if the physical link is still active. ScreenOS uses Layer 3 path monitoring, or *IP tracking,* similar to that used for NSRP, to monitor IP addresses through the primary interface. If the IP addresses become unreachable through the primary Untrust zone interface, the security device considers the interface to be down, and all routes associated with that interface are deactivated. When the primary Untrust zone interface changes to the down state, failover to the backup Untrust zone interface occurs. You can configure IP tracking without configuring automatic interface failover.

Active-to-Backup Tunnel Failover

You can configure a redundant pair of bidirectional VPN tunnels on the security device to a remote IKE peer. Only one tunnel is active at any given time. The VPN tunnel from the primary interface is active initially (vpn1 in this example). If the primary tunnel fails, then the security device fails over VPN traffic destined for the remote peer to the backup tunnel (vpn2 in this example).

Interface Failover with VPN Tunnel Monitoring

You can specify an interface failover when certain VPN tunnels on the primary interface are determined to be "down." For each VPN tunnel, you specify a failover weight, in percent. The assigned weights only come into play when the status of one or more monitored tunnels is "down". If the cumulative weight of the down VPN tunnels reaches or exceeds 100 percent, ScreenOS fails over to the backup interface.

By applying a *weight,* or a value, to a VPN tunnel, you can adjust the importance of the tunnel status in relation to other tunnels. You can assign comparatively greater weight to relatively more important tunnels, and less weight to relatively less important tunnels. The accumulated weights of *all* monitored VPN tunnels determine when interface failover occurs. For example, failure of a VPN tunnel with a weight of 50 brings the primary interface closer to a failover than would the failure of a VPN tunnel with a weight of 10. Tunnels that are in inactive, ready, or undetermined states are counted as 50 percent of the assigned weight. That is, if you assign a weight of 50 to a tunnel that is in inactive state, the tunnel's weight that is counted toward interface failover is 25.

If failover to the backup interface occurs, ScreenOS can still try to establish new VPN tunnel(s) on the primary interface if the VPN monitor rekey feature is enabled. If one or more VPN tunnels on the primary interface returns to "up" status so that the accumulated failover weight is less than 100 percent, ScreenOS can revert traffic back to the primary interface. Enable the VPN monitor rekey feature to allow ScreenOS to switch traffic from the backup interface to the primary.

NSRP Object Monitoring to Trigger Failover

With NSRP, you can monitor certain objects to determine failover of the security device or of a VSD group. NSRP monitored objects can include:

Table 130: NSRP Monitored Objects

Object Type	Description
Physical interfaces	Ensures that the physical ports are active and connected to other devices.
Security Module	Ensures that the security modules on your device are active.
Zones	Ensures that all physical ports in a zone are active.
Specific target IP addresses	The device sends ping or ARP requests to up to 16 specified IP addresses per monitored object at specified intervals and then monitors responses from the targets. All the IP addresses configured on the device or for a specified VSD group constitute a single monitored object. A device can have one monitored object and each VSD group can have its own monitored object.

NOTE: A security device supports up to 32 monitored objects for use by NSRP and interface-based monitoring and up to 64 tracked IP addresses.

Configuring device or VSD group failover with monitored objects involves setting the following:

- **Device or VSD failover threshold**—The device or VSD group failover threshold is the total weight of failed monitored objects that is required to cause either a VSD group on a device or a device in an NSRP cluster to initiate a failover to the backup device. If the cumulative weight of the failures of all monitored objects exceeds the threshold, then the VSD group or the device fails over to the backup VSD group or device. You can set the device or VSD failover threshold at any value between 1 and 255. The default threshold is 255.
- **Failure weight of each object being monitored**—Each monitored object has a configurable failure weight, which is the weight that the failure of the monitored object contributes toward the device or VSD failover threshold. You can set the object failure weight at any value between 1 and 255.

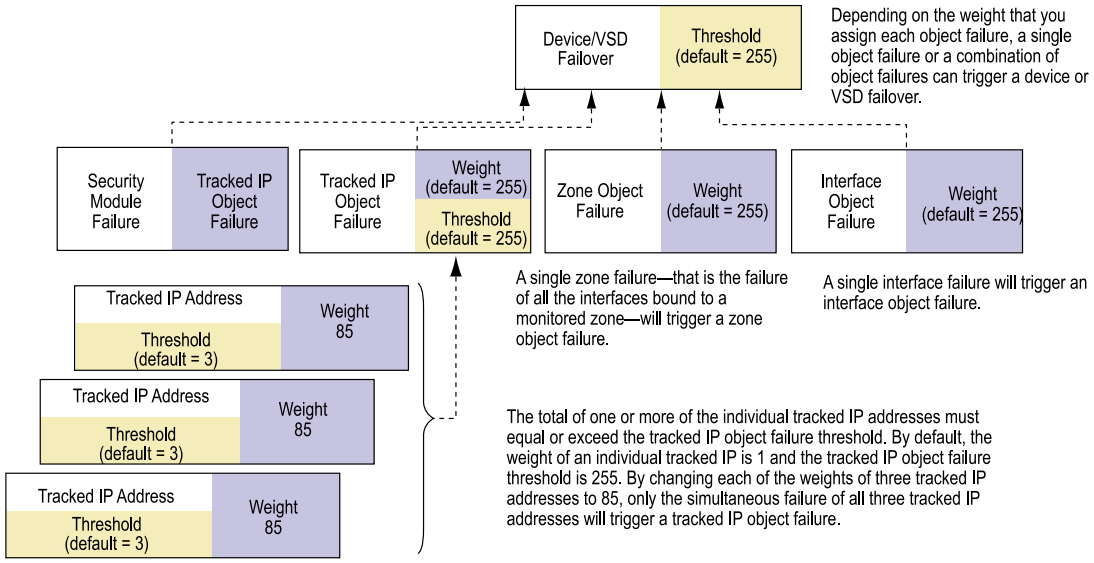
For tracked IP addresses, you need to specify individual IP addresses and how they are to be monitored. You also need to define what constitutes the failure of each tracked IP address (the threshold) and the weight that the failed IP address carries. For the tracked IP object, you also specify a failure threshold. This threshold is the

sum of the weights of all failed tracked IP addresses required for the tracked IP object to be considered failed.

Objects that are monitored for a VSD group are independent from the objects monitored for the device. That is, you can configure a specific set of objects, weights, and thresholds for a VSD group and a different set for a device. You can also configure independent sets of monitored objects for different VSD groups. For example, you can configure the same monitored objects for two VSD groups with different weights and thresholds specified for each VSD group for the object.

Figure 452 on page 1827 shows the relationship of various monitored objects to the device or VSD group failover. The weights of all failed monitored objects contribute toward the device or VSD failover threshold. If you do not change the default weight of a monitored object or the device or VSD failover threshold, failure of any monitored object can cause the device or VSD to fail over. For tracked IP addresses, the weights of all failed tracked IP addresses contribute toward the tracked IP object failure threshold. If the tracked IP object failure threshold is reached, the tracked IP object failure weight is compared to the device or VSD failover threshold.

Figure 452: Object Monitoring Weights and Failover Thresholds



Security Module

If your device contains a security module and the device fails, you can set a weight for each failure. This gives you the flexibility to decide whether an entire device should fail if a security module on that device fails. In this example, you set the failure weight for security module 2 to 100.

CLI

```
set nsrp monitor sm 2 100
save
```

Physical Interface

Layer 2 path monitoring functions by checking that the physical ports are active and connected to other network devices. Failure of a physical interface object occurs when the port is no longer active.

In this example, you enable the monitoring of ethernet2/1 for a possible device failover. You set a failure weight of 100 for the interface.

WebUI

Network > NSRP > Monitor > Interface > VSD ID: Device Edit Interface: Enter the following, then click **Apply**:

Interface Name: ethernet2/1 (select)
Weight: 100

CLI

```
set nsrp monitor interface ethernet2/1 weight 100
save
```

Zone Objects

Failure of a zone object occurs only when *all* interfaces in a monitored zone are down. There is no zone failure as long as there is still an active port in the zone. If a monitored zone has no interfaces bound to it, the zone object cannot not fail. The security device always perceives its state as up. If a down interface is the only interface bound to a monitored zone, the zone object fails; if you unbind the interface from the zone, the zone object is no longer failed. If you unbind an active interface from a monitored zone where the remaining interfaces are down, the zone fail.

In this example, you enable the monitoring of the Trust zone for a possible device failover. You set a failure weight of 100 for the zone.

WebUI

Network > NSRP > Monitor > Zone > VSD ID: Device Edit Zone: Enter the following, then click **Apply**:

Zone Name: Trust (select)
Weight: 100

CLI

```
set nsrp monitor zone trust weight 100
save
```

Tracked IP Objects

IP tracking functions by sending ping or ARP requests to up to 16 specified IP addresses at user-determined intervals and then monitoring the targets for responses. The ping or ARP request will be counted as a failure if the response time of the request exceeds the specified time-out value. When you configure IP tracking, the device sends either ping or ARP requests from a Manage IP address that is bound to a physical interface, redundant interface, or subinterface. (The Manage IP address must be a different IP address from the IP address of the interface.) You cannot use a virtual security interface (VSI) for IP tracking because that address can shift its bindings among multiple devices.

For IPv6 addresses, the device sends only ping requests. Because the Manage IP address does not apply to IPv6 addressing, the only way to configure the **track-ip** option for IPv6 is to unset VSD group 0.

NOTE: When routers are grouped in a redundant cluster using Virtual Router Redundancy Protocol (VRRP), the router functioning as the primary device cannot respond to ping requests to the virtual IP (VIP) address if it is not the IP address owner (which might be the case after a failover). However, the primary device virtual router must respond to ARP requests with the virtual MAC address regardless of IP address ownership. (Refer to RFC 2338 for details.) To use ARP when IP tracking, the polled device must be on the same physical subnet as the Manage IP address.

For each tracked IP address, you specify the following:

- **Tracked IP Failure Threshold**—The number of consecutive failures to elicit a ping or an ARP response from a specific IP address that constitutes a failed attempt. Not exceeding the threshold indicates an acceptable level of connectivity with the address; exceeding it indicates an unacceptable level. You can set the threshold to any value between 1-200. The default is 3.
- **Tracked IP Failure Weight—**The weight that failure to elicit a response from the tracked IP address contributes to the tracked IP object failure weight. By applying a weight to a tracked IP address, you can adjust the importance of connectivity to that address in relation to reaching other tracked IP addresses. You can assign greater weights to relatively more important addresses and lesser weights to relatively less important addresses. The assigned weights come into play when a tracked IP failure threshold is reached. For example, exceeding the tracked IP failure threshold for an address weighted 10 adds more to the tracked IP object failure weight than would a tracked IP failure for an address weighted 1. You can assign weights from 1 to 255. The default is 1.

You also configure a failure threshold for the tracked IP object that contributes to the device or VSD failover threshold. If one or more tracked IP addresses exceed their failure thresholds, then the weights for the individual failed addresses are totaled. If the sum reaches or exceeds the failure threshold for the tracked IP object, then the tracked IP object failure weight is applied to the device or VSD failover threshold. Only the failure weight of the tracked IP object is applied to the device or VSD failover

threshold; failure weights of individual tracked IP addresses are never applied to the device or VSD failover threshold. Consider the following example:

Tracked IP Addresses	Failure Weights			
10.10.10.250	100	Tracked IP Object Failure Threshold	Tracked IP Object Failure Weight	Device Failover Threshold
1.1.1.30	75	125	255	255
2.2.2.40	75			

If the tracked IP address 10.10.10.250 fails, then the tracked IP failure weight (100) is compared to the tracked IP object failure threshold (125). Since the tracked IP failure weight is less than the tracked IP object failure threshold, the tracked IP object is not considered failed. If both tracked IP addresses 1.1.1.30 and 2.2.2.40 fail, then the combined failure weight (150) is compared to the tracked IP object failure threshold (125). Since the combined failure weight exceeds the tracked IP object failure weight, the tracked IP object is considered failed. The tracked IP object failure weight (255) is compared to the device failover threshold (255). Since the tracked IP object failure weight equals the device failover threshold, the device fails over.

To set a failure weight of 100 for the tracked IP address 10.10.10.250:

WebUI

Network > NSRP > Track IP > New: Enter the following, then click **OK**:

Track IP: 10.10.10.250
Weight: 100

CLI

```
set nsrp track-ip ip 10.10.10.250 weight 100
save
```

To set a failure threshold of 125 for the tracked IP object for a possible device failover, enter the following:

WebUI

Network > NSRP > Monitor > Track IP > VSD ID: Device Edit: Enter the following, then click **Apply**:

Enable Track IP: (select)
Failover Threshold: 125

CLI

```
set nsrp monitor track-ip threshold 125
save
```

To set a time-out value of 10 for tracked IP address 1.1.1.250:

WebUI

Network > NSRP > Monitor > Track IP > New: Enter the following, then click **Apply**:

```
Track IP: 1.1.1.250
Interval: 20
Time Out: 10
```

CLI

```
set nsrp monitor track-ip ip 1.1.1.250 interval 20
set nsrp monitor track-ip ip 1.1.1.250 time-out 10
save
```

NOTE: The time-out value should not be greater than the interval value.

Track IP for Device Failover

Two security devices are in an Active/Active configuration. Every 10 seconds, both devices send ARP requests to the physical IP addresses (addresses dedicated to the physical routers that comprise the VRRP cluster) of two external routers running VRRP in a redundant cluster in the Untrust zone and ping requests to two Web servers in the Trust zone. The tracked IP object failure threshold is 51. The tracked IP object weight and the device failover threshold are the default values (255). The weights and failure thresholds of the tracked IP addresses are as follows:

- Redundant routers in the Untrust zone
  - 210.1.1.250—Weight: 16, threshold 5
  - 210.1.1.251—Weight: 16, threshold 5
- Webservers in the Trust zone
  - 10.1.1.30—Weight: 10, threshold 3
  - 10.1.1.40—Weight: 10, threshold 3

Not receiving an ARP response after 5 consecutive attempts to one of the routers is considered a failed attempt and contributes a weighted value of 16 toward the total failover threshold. Not receiving a ping response after 3 consecutive attempts to one of the Web servers is considered a failed attempt and contributes a weighted value of 10 toward the total failover threshold.

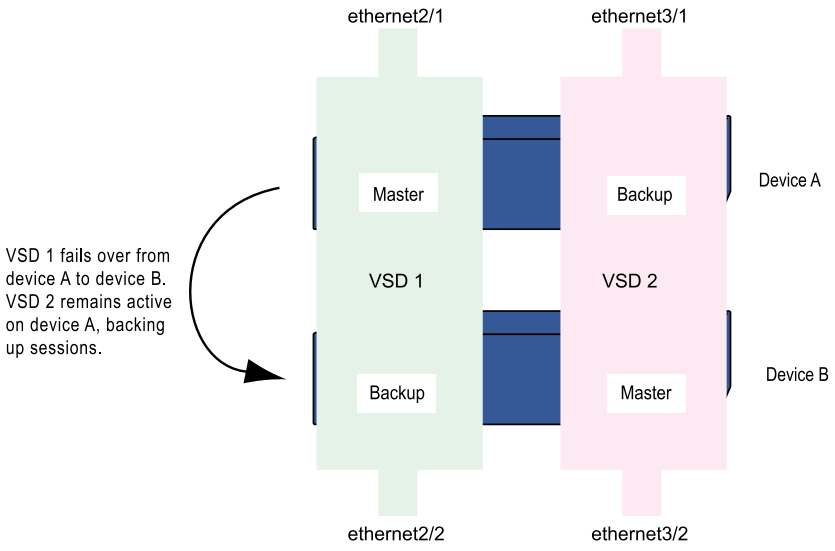
Because the device failover threshold is 51, all four tracked IP addresses must fail before a device failover occurs. If you are not willing to tolerate that amount of failure, you can lower the threshold to a more acceptable level.

Virtual Security Device Group Failover

In addition to device failover, you can configure NSRP for VSD group failover. Like device failover, failure of one or more monitored objects can cause the primary device in a VSD group to fail over to the backup device for the group. See “NSRP Object Monitoring to Trigger Failover” on page 1826 for information about the objects and how to configure them. For VSD failover, you can configure the same objects to be monitored as you can for device failover.

In the example shown in Figure 453 on page 1832, if a port on a primary device in a VSD group fails, the entire device does not necessarily fail over to the backup device. In the following configuration, if ethernet 2/1 fails, VSD 1 fails over from the primary VSD group on device A to the backup VSD group on device B. VSD 2 remains active, backing up sessions on device A.

Figure 453: VSD Group 1 Failover



Virtual System Failover

For a virtual system to fail over, it must be in a VSD group. For a VSD group to support virtual systems, you must create VSIs for each virtual system. A virtual system has its own Trust zone VSI, and it can have its own Untrust zone VSI. A virtual system can also share the Untrust zone VSI with the root level. When virtual systems have their own Untrust zone VSIs, they must be in different subnets from each other and from the Untrust zone VSI at the root level. All Trust zone virtual system VSIs must also be in different subnets from one another.

Table 131 on page 1833 lists the two security devices (device A and device B) that are in an Active/Active full-mesh configuration. You have already configured the root

system of device A as the primary device of VSD 0 and that of device B as the primary device of VSD group 1. The Trust and Untrust zone VSIs for VSDs 0 and 1 in the root system are as follows:

Table 131: VSD IP Address

VSIs for VSD Group 0		VSIs for VSD Group 1	
redundant1	210.1.1.1/24	redundant1:1	210.1.1.2/24
redundant2	10.1.1.1/24	redundant2:1	10.1.1.2/24

Device Failover

When you configure two security devices in an NSRP cluster, the primary device device synchronizes all configuration and state information with the backup device so that the backup device can become the primary device when necessary. For example, if the primary device in a cluster fails, the backup device is promoted to primary device and takes over traffic processing. If the original primary device is restored to its pre-failure status, it can resume traffic processing.

Many different conditions exist that can cause a primary device in an NSRP cluster to fail over to the backup, such as the following physical problems or administrator-introduced thresholds and weights:

- **Physical problems:** system crash, loss of power, down link, or removal of processor or memory boards from the device
- **Administrator-introduced failover:** loss of connection to certain gateways or servers causing a primary device to fail over to the backup

You can configure NSRP to monitor different objects so that the failure of one or more of the monitored objects causes a failover of the primary device. For more information about these objects and how to configure them, see “Interface Failover with VPN Tunnel Monitoring” on page 1825.

In the event of multiple failover instances within a cluster, at least one device must remain as the primary device. If a device is the last device in a cluster that has not failed or become ineligible to become the primary device, that device continues to act as primary device. Under certain conditions, the failure of monitored objects can cause both devices in a cluster to become ineligible, which results in a traffic *black hole*. To ensure that one device is still elected as primary device and can forward traffic, issue the CLI command **set nsrp vsd-group master-always-exist**. This allows a device in the NSRP cluster to continue to forward traffic even if all units in the cluster are deemed to have failed due to NSRP object monitoring. If all devices in a cluster simultaneously transition to a failed state, a new primary device is elected based on the unit ID, the failure weight, and the preempt and priority values that you previously configured for the devices. Depending on whether the monitored objects are used to determine failover of the security device or of a VSD group, the type of values used to elect a new primary device differ.

Example 1

In this example, you monitor an object to determine failover of the security device. If device failover occurs in a cluster, the device with the lowest failure weight sum becomes the primary device. If the failure weight sums for two or more devices are equal, the device with the highest unit ID becomes the primary device. In this case, the preempt and priority values are not considered.

CLI

```
set nsrp cluster id 1
set nsrp vsd-group master-always-exist
set nsrp vsd-group id 0 priority 100
set nsrp monitor track-ip ip
set nsrp monitor track-ip threshold 1
set nsrp monitor track-ip ip 43.3.3.3 interface ethernet1/2
set nsrp monitor track-ip ip 43.3.3.3 interval 3
set nsrp monitor track-ip ip 43.3.3.3 threshold 2
```

Example 2

In this example, you monitor an object to determine failover of a VSD group. If VSD failover occurs, the VSD group member with the lowest failure weight sum becomes the primary device. If the failure weight sums for two or more devices are equal, the VSD group member with the priority number closest to 0 becomes the primary device. If the VSD priority also equals, the VSD group member with the highest unit ID is elected as the primary device.

CLI

```
set nsrp cluster id 1
set nsrp vsd-group master-always-exist
set nsrp vsd-group id 0 priority 100
set nsrp vsd id 0 monitor track-ip ip
set nsrp vsd id 0 monitor track-ip threshold 1
set nsrp vsd id 0 monitor track-ip ip 43.3.3.3 interface ethernet1/2
set nsrp vsd id 0 monitor track-ip ip 43.3.3.3 interval 3
set nsrp vsd id 0 monitor track-ip ip 43.3.3.3 threshold 2
set nsrp vsd id 0 monitor track-ip ip 43.3.3.3 method arp
```

VRRP Support

If your Juniper Networks security device is deployed in a high availability (HA) configuration that includes devices from other vendors, you must enable the Virtual Router Redundancy Protocol (VRRP) on the device. VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. VRRP eliminates the single point of failure inherent in a static default routed environment.

VRRP enables hosts on a LAN to use redundant routers on that LAN with the default route on the hosts. VRRP routers share the IP address corresponding to the default

route configured on the hosts. In a VRRP configuration, one of the VRRP routers acts as the primary (active) and the others as backups. If the primary router fails, one of the backup routers becomes the new primary, thus always providing a virtual default router and allowing traffic on the LAN to be routed without relying on a single router.

You can configure VRRP support on an interface only on security devices with Ethernet or gigabit Ethernet interfaces. WAN and serial interfaces cannot support VRRP. By default, the security device uses the VRRP group ID as the NSRP virtual security device (VSD) group ID and uses the physical interface IP as the Manage IP. Similarly, the virtual security interface (VSI) uses the VRRP virtual IP as the IP address of the interface.

Note that NSRP and VRRP cannot exist on the same device. While VRRP provides the redundancy configuration in a cluster, it differs from NSRP in the following ways:

- NSRP can be configured to force a virtual router to become the primary when all virtual routers in a cluster are ineligible to become the primary for any reason, thus preventing a *black hole*. For more information, see "Device Failover" on page 1833.
- VRRP does not support two interfaces within the same VRRP group on the same router. On the other hand, NSRP supports two interfaces by separate VSD and VSI.
- VRRP failover is based on IP address or interface. The VRRP interface becomes logically down either due to physical link failures or interface monitoring. NSRP failover can be based on the device.
- In VRRP, one interface can hold multiple virtual IP addresses. In NSRP, one VSI can hold only one virtual IP address. However, with VSD groups, NSRP can support multiple virtual IP addresses per interface.
- VRRP does not support synchronization sessions. NSRP supports synchronization sessions.

Configuration Examples

This section provides procedures for the following configuration examples:

Configuring Track IP for Device Failover

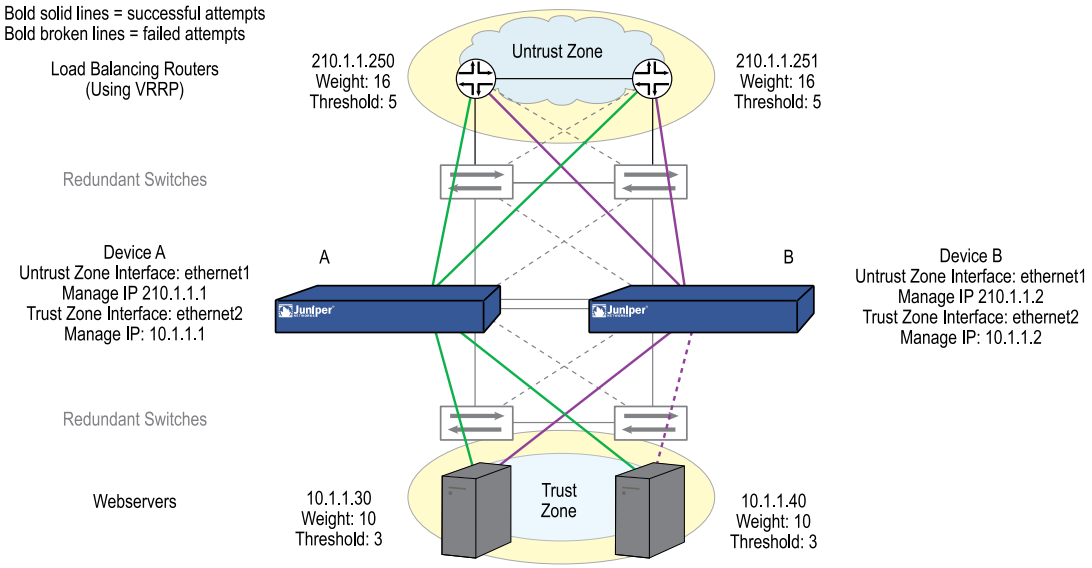
Using IP tracking is an option, along with interface monitoring and zone monitoring, for tracking device failure. Figure 454 on page 1836 shows device A has a 100 percent

success rate, while device B has failed to receive three consecutive responses from 10.1.1.40, contributing a value of 10 toward the total failover threshold of 51.

NOTE: All NSRP monitoring settings apply to the local unit only. The IP tracking settings do not propagate to other devices in a VSD group. You must enter the same settings on all devices in the group if necessary.

The Untrust zone interface is ethernet1 and the Trust zone interface is ethernet2 on both devices. The ethernet1 Manage IP address is 210.1.1.1 on device A, and 210.1.1.2 on device B. The ethernet2 Manage IP address is 10.1.1.1 on device A, and 10.1.1.2 on device B. All the security zones are in the trust-vr routing domain.

Figure 454: Track IP for Device Failover



WebUI

1. **Track IP Addresses**

   Network > NSRP > Monitor > Track IP > New: Enter the following, then click **OK**:

   Track IP: 210.1.1.250
   Method: ARP
   Weight: 16
   Interval (sec): 10
   Threshold: 5
   Interface: ethernet1
   VSD Group ID: Device

   Network > NSRP > Monitor > Track IP > New: Enter the following, then click **OK**:

Track IP: 210.1.1.251
Method: ARP
Weight: 16
Interval (sec): 10
Threshold: 5
Interface: ethernet1
VSD Group ID: Device

Network > NSRP > Monitor > Track IP > New: Enter the following, then click **OK**:

Track IP: 10.1.1.30
Method: Ping
Weight: 10
Interval (sec): 10
Threshold: 3
Interface: ethernet2
VSD Group ID: Device

Network > NSRP > Monitor > Track IP > New: Enter the following, then click **OK**:

Track IP: 10.1.1.40
Method: Ping
Weight: 10
Interval (sec): 10
Threshold: 3
Interface: ethernet2
VSD Group ID: Device

2. **Track IP Object Failure Threshold**

Network > NSRP > Monitor > Track IP > Edit (for VSD: Device): Enter the following, then click **Apply**:

Enable Track IP: (select)
Failover Threshold: 51

CLI

1. **Track IP Addresses**

```
set nsrp track-ip ip 210.1.1.250 interface ethernet1
set nsrp track-ip ip 210.1.1.250 interval 10
set nsrp track-ip ip 210.1.1.250 method arp
set nsrp track-ip ip 210.1.1.250 threshold 5
set nsrp track-ip ip 210.1.1.250 weight 16
set nsrp track-ip ip 210.1.1.251 interface ethernet1
set nsrp track-ip ip 210.1.1.251 interval 10
set nsrp track-ip ip 210.1.1.251 method arp
set nsrp track-ip ip 210.1.1.251 threshold 5
set nsrp track-ip ip 210.1.1.251 weight 16
set nsrp track-ip ip 10.1.1.30 interface ethernet2
set nsrp track-ip ip 10.1.1.30 interval 10
```

```
set nsrp track-ip ip 10.1.1.30 method ping
set nsrp track-ip ip 10.1.1.30 threshold 3
set nsrp track-ip ip 10.1.1.30 weight 10
set nsrp track-ip ip 10.1.1.40 interface ethernet2
set nsrp track-ip ip 10.1.1.40 interval 10
set nsrp track-ip ip 10.1.1.40 method ping
set nsrp track-ip ip 10.1.1.40 threshold 3
set nsrp track-ip ip 10.1.1.40 weight 10
set nsrp track-ip
```

NOTE: By default, pinging is the method for IP tracking and a tracked IP failure threshold value is 3; therefore, you do not need to specify them. The commands **set nsrp track-ip ip 10.1.1.30** and **set nsrp track-ip ip 10.1.1.40** are sufficient.

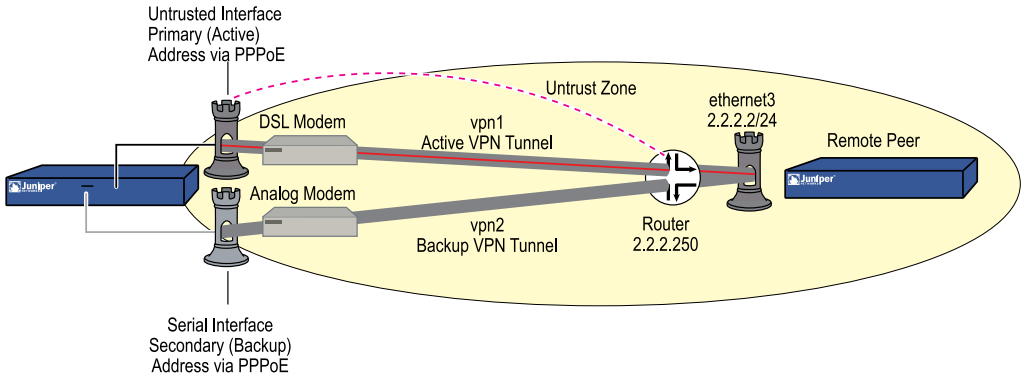
2. **Track IP Object Failure Threshold**

```
set nsrp track-ip threshold 51
save
```

Configuring a Redundant VPN Tunnel

Figure 455 on page 1838 shows you how to configure a redundant pair of bidirectional VPN tunnels on the security device to a remote IKE peer. Only one tunnel is active at any given time. The VPN tunnel from the primary interface is active initially (vpn1 in this example). If the primary tunnel fails, then the security device fails over VPN traffic destined for the remote peer to the backup tunnel (vpn2 in this example).

Figure 455: Tunnel Failover from the Untrusted Interface to the Serial Interface



You configure only one VPN tunnel at the remote peer site because—from the remote peer's point of view—there is only one VPN tunnel from the device, resulting in a Y-shaped VPN configuration.

NOTE: When setting up a Y-shaped VPN configuration and the backup interface is an ethernet interface (Dual-Untrust mode for example), do not enable the VPN monitoring rekey option for any VPN tunnel using the backup interface. If you do, the security device continually tries to bring up that tunnel even though you want it to stay down. If the backup interface is a serial interface (as in this example), it does not matter if you enable VPN monitoring with the rekey option for a VPN tunnel on the backup interface.

You use IP tracking to determine if a failover from the Untrusted interface to the serial interface ever becomes necessary. You configure IP tracking to ping the remote peer's external router at 2.2.2.250. You track that address instead of the address of the remote peer's Untrust zone interface (2.2.2.2) because the security device at the remote site is not configured to respond to ICMP echo requests arriving at its Untrust zone interface. You set the following IP tracking values:

- Track IP: 2.2.2.250
  - Weight: 255
  - Interval: 4
  - Threshold: 3
- Track IP failure threshold: 255
- Monitor failure threshold: 255
- Failover holddown: 16

Using the above settings, a failover from vpn1 to vpn2 takes about 30 seconds after IP tracking begins losing connectivity with the tracked IP address (2.2.2.250): 3 failed ICMP echo requests at 4-second intervals = 12 seconds + the 16-second holddown time. During the holddown time, the NetScreen-5GT continues to send ICMP echo requests every 4 seconds; so, in sum, a failover requires a total of 7 consecutive failed attempts to elicit replies from ICMP echo requests (the first 3 + 4 more during the holddown period).

NOTE: Because this particular example is long, only the CLI configuration is included in its entirety. The WebUI section simply lists the navigational paths to the pages where you can set the various elements of the configuration. You can see what you need to set by referring to the CLI commands.

WebUI

1. **Login and Interfaces**

   Log back into the security device. Then continue with the following configuration:

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for serial)

   Network > Interfaces > New Tunnel IF

2. **Address**

   Policy > Policy Elements > Addresses > List > New

3. **PPPoE**

   Network > PPPoE > New

4. **VPN Tunnels**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey IKE > New

5. **Asymmetric VPN**

   Network > Zones > Edit (for Trust)

6. **IP Tracking**

   Network > Interfaces > Edit (for untrust) > Monitor

   Network > Interfaces > Edit (for untrust) > Monitor > Monitor Track IP ADD

7. **Tunnel Failover**

   Network > Untrust Failover

   Network > Untrust Failover > Edit Weight

8. **Routes**

   Network > Routing > Routing Entries > trust-vr New

9. **Policies**

   Policies > (From: Trust, To: Untrust) New

   Policies > (From: Untrust, To: Trust) New

WebUI (Remote Peer)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet3)

   Network > Interfaces > Edit (for ethernet1)

   Network > Interfaces > New Tunnel IF

2. **Address**

   Policy > Policy Elements > Addresses > List > New

3. **VPN Tunnel**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey IKE > New

4. **Routes**

   Network > Routing > Routing Entries > trust-vr New

5. **Policies**

   Policies > (From: Trust, To: Untrust) New

   Policies > (From: Untrust, To: Trust) New

CLI

1. **Login and Interfaces**

   Log back into the security device. Then continue with the following configuration:

```
set interface trust ip 10.1.1.1/24
set interface trust nat
set interface serial zone untrust
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface untrust
set interface tunnel.2 zone untrust
set interface tunnel.2 ip unnumbered interface untrust
```

2. **Address**

```
set address untrust peer1 10.2.2.0/24
```

3. **PPPoE**

```
set pppoe name isp1a
set pppoe name isp1a username ns5gt password juniper
set pppoe name isp1a idle 0
set pppoe name isp1a interface untrust
exec pppoe name isp1a connect
```

4. **VPN Tunnels**

```
set ike gateway gw1 address 2.2.2.2 aggressive local-id ns5gt outgoing-interface
 untrust preshare netscreen1 sec-level compatible
set ike gateway gw2 address 2.2.2.2 aggressive local-id ns5gt outgoing-interface
 serial preshare netscreen1 sec-level compatible
set vpn vpn1 gateway gw1 sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
set vpn vpn2 gateway gw2 sec-level compatible
set vpn vpn2 bind interface tunnel.2
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

5. **Asymmetric VPN**

```
set zone trust asymmetric-vpn
```

6. **IP Tracking**

```
set interface untrust monitor track-ip ip
set interface untrust monitor track-ip ip 2.2.2.250 interval 4
set interface untrust monitor track-ip ip 2.2.2.250 threshold 3
set interface untrust monitor track-ip ip 2.2.2.250 weight 255
```

7. **Tunnel Failover**

```
set failover enable
set failover auto
set failover holddown 16
set failover type track-ip
set interface untrust track-ip threshold 255
```

8. **Routes**

```
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.1
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.2
set vrouter trust-vr route 10.2.2.0/24 interface null metric 100
```

9. **Policies**

```
set policy from trust to untrust any any any permit
set policy from untrust to trust peer1 any any permit
save
```

CLI (Remote Peer)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.1 zone trust
set interface tunnel.1 ip unnumbered interface ethernet1
```

2. **Address**

   ```
   set address untrust ns5gt 10.1.1.0/24
   ```

3. **VPN Tunnel**

   ```
   set ike gateway ns5gt dynamic ns5gt aggressive outgoing-interface ethernet3
   preshare netscreen1 sec-level compatible
   set vpn vpn1 gateway ns5gt sec-level compatible
   set vpn vpn1 bind interface tunnel.1
   set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
   ```

4. **Routes**

   ```
   set vrouter trust-vr route 10.1.1.0/24 interface tunnel.1
   set vrouter trust-vr route 10.1.1.0/24 interface null metric 100
   set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
   ```

5. **Policy**

   ```
   set policy from untrust to trust ns5gt any any permit
   set policy from trust to untrust any ns5gt any permit
   save
   ```

Configuring Virtual Security Interfaces

In the example shown in Figure 456 on page 1845, devices A and B are members of two VSD groups—VSD group 0 and VSD group 1—in an Active/Active configuration. Device A is the primary device of VSD group 0 and the backup in VSD group 1. Device B is the primary device of VSD group 1 and the backup in VSD group 0. The security devices are linked to two pairs of redundant switches—switches A and B in the Untrust zone and switches C and D in the Trust zone.

NOTE: This example only presents the creation of redundant interfaces on device A. Because devices A and B are members of the same NSRP cluster, device A propagates all interface configurations to device B except the Manage IP address, which you enter on the redundant2 interface on both devices: device A 10.1.1.21, device B 10.1.1.22.

You put ethernet1/1 and ethernet1/2 in redundant1, and ethernet2/1 and ethernet2/2 in redundant2. On the redundant2 interface, you define a Manage IP of 10.1.1.21 for device A and a Manage IP of 10.1.1.22 for device B on this interface.

The physical interfaces that are bound to the same redundant interface connect to different switches:

- Physical interfaces bound to a redundant interface in the Untrust zone: ethernet1/1 to switch A, ethernet1/2 to switch B
- Physical interfaces bound to a redundant interface in the Trust zone: ethernet2/1 to switch C, ethernet2/2 to switch D

NOTE: The physical interfaces do not have to be in the same security zone as the redundant interface to which you bind them.

By putting ethernet1/1 and ethernet2/1 in their respective redundant interfaces first, you designate them as primary interfaces. (You can change the primary status assignments via the CLI command **set interface** *redundant1* **primary** *interface1/1*.) If the link to a primary interface becomes disconnected, the security device reroutes traffic through the secondary interface to the other switch without requiring the VSD primary device to fail over.

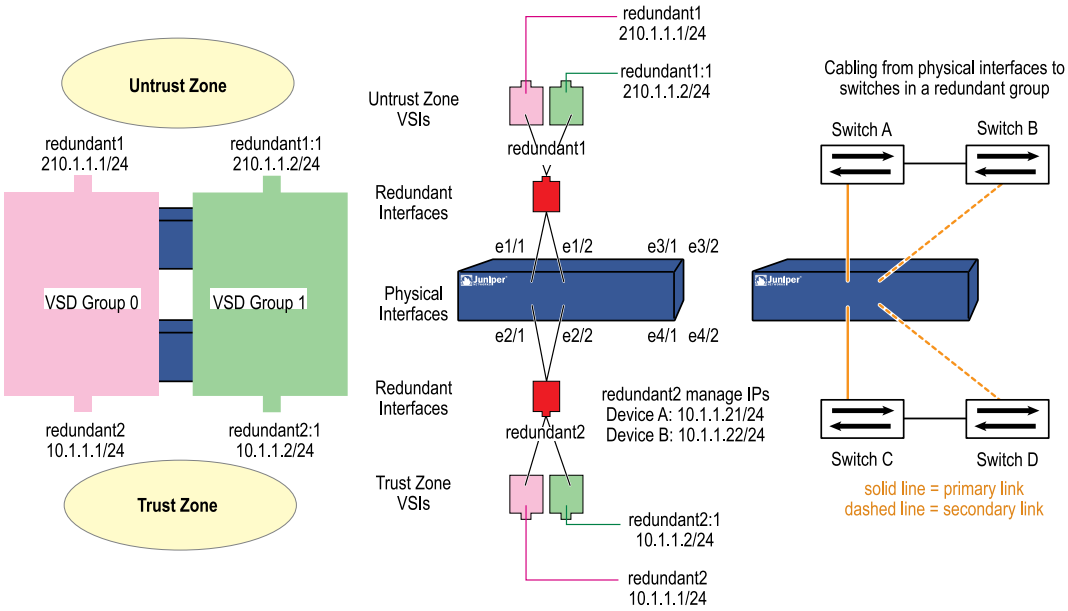
In this example, the cable from ethernet1/1 becomes disconnected, causing a port failover to ethernet1/2. Consequently, all the traffic to and from devices A and B passes through switch B. Reconnecting the cable from ethernet1/1 on device A to switch A automatically causes that interface to regain its former priority.

The IP addresses for the VSIs:

VSIs for VSD Group 0		VSIs for VSD Group 1	
redundant	210.1.1.1/24	redundant1:1	210.1.1.2/24
redundant2	10.1.1.1/24	redundant2:1	10.1.1.2/24

NOTE: IP addresses for multiple VSIs can be in the same subnet or in different subnets if the VSIs are on the same redundant interface, physical interface, or subinterface. If the VSIs are on different interfaces, they must be in different subnets.

Figure 456: Redundant Interfaces for VSIs



WebUI (Device A)

1. **Redundant Interfaces**

   Network > Interfaces > New Redundant IF: Enter the following, then click **OK**:

   Interface Name: redundant1
   Zone Name: Untrust
   IP Address / Netmask: 210.1.1.1/24

   Network > Interfaces > Edit (for ethernet1/1): Select **redundant1** in the "As member of" drop-down list, then click **OK**.

   Network > Interfaces > Edit (for ethernet1/2): Select **redundant1** in the "As member of" drop-down list, then click **OK**.

   Network > Interfaces > New Redundant IF: Enter the following, then click **Apply**:

   Interface Name: redundant2
   Zone Name: Trust
   IP Address / Netmask: 10.1.1.1/24

   Enter **10.1.1.21** in the Manage IP field, then click **OK**.

   Network > Interfaces > Edit (for ethernet2/1): Select **redundant2** in the "As member of" drop-down list, then click **OK**.

   Network > Interfaces > Edit (for ethernet2/2): Select **redundant2** in the "As member of" drop-down list, then click **OK**.

2. **Virtual Security Interfaces**

Network > Interfaces > New VSI IF: Enter the following, then click **OK**:

Interface Name: VSI Base: redundant1
VSD Group: 1
IP Address / Netmask: 210.1.1.2/24

Network > Interfaces > New VSI IF: Enter the following, then click **OK**:

Interface Name: VSI Base: redundant2
VSD Group: 1
IP Address / Netmask: 10.1.1.2/24

WebUI (Device B)

Network > Interfaces > Edit (for redundant2): Type **10.1.1.22** in the Manage IP field, then click **OK**.

NOTE: You must enter static routes to addresses beyond the immediate subnet of a VSI for each VSI in each VSD.

CLI (Device A)

1. **Redundant Interfaces**

```
set interface redundant1 zone untrust
set interface redundant1 ip 210.1.1.1/24
set interface ethernet1/1 group redundant1
set interface ethernet1/2 group redundant1
set interface redundant2 zone trust
set interface redundant2 ip 10.1.1.1/24
set interface redundant2 manage-ip 10.1.1.21
set interface redundant2 nat
set interface ethernet2/1 group redundant2
set interface ethernet2/2 group redundant2
set interface redundant1 primary ethernet1/1
set interface redundant2 primary ethernet2/1
```

2. **Virtual Security Interfaces**

```
set interface redundant1:1 ip 210.1.1.2/24
set interface redundant2:1 ip 10.1.1.2/24
save
```

CLI (Device B)

```
set interface redundant2 manage-ip 10.1.1.22
save
```

NOTE: You must enter static routes to addresses beyond the immediate subnet of a VSI for each VSI in each VSD.

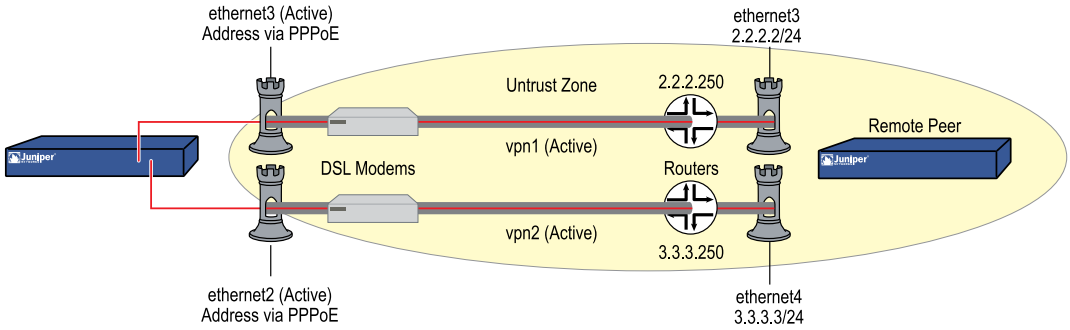
Configuring Dual Active Tunnels

The purpose of this configuration is to support VPN traffic failover between two active VPN tunnels.

You configure a redundant pair of bidirectional VPN tunnels (vpn1 and vpn2) from the NetScreen-5GT to a remote IKE peer. Both tunnels are active at the same time, and the NetScreen-5GT performs a basic form of load-balancing, alternating sessions between the two tunnels. (This is not true load-balancing because the amount of traffic can vary greatly from one session to another, resulting in uneven “loads.”) If either tunnel fails, then the NetScreen-5GT directs all VPN traffic destined for the remote peer through the other tunnel.

The NetScreen-5GT is in Dual-Untrust mode. Both ethernet3 and ethernet2 connect to DSL modems. They both become active interfaces when you disable the failover option. See Figure 457 on page 1847.

Figure 457: Failover Between Two Active Tunnels



You enable the asymmetric VPN option for the Trust zone at each site so that if an existing session established on one VPN tunnel transfers to another, the security device at the other end of the tunnel does not reject it.

NOTE: Because this particular example is long, only the CLI configuration is included in its entirety. The WebUI section simply lists the navigational paths to the pages where you can set the various elements of the configuration. You can see what you need to set by referring to the CLI commands.

WebUI

1. **Login and Interfaces**

   Log back into the security device. Then continue with the following configuration:

   Network > Interfaces > Edit (for ethernet1)

   Network > Interfaces > New Tunnel IF

2. **Address**

   Policy > Policy Elements > Addresses > List > New

3. **PPPoE**

   Network > PPPoE > New

4. **VPN Tunnels**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey IKE > New

5. **Dual Tunnels**

   Network > Untrust Failover

6. **Asymmetric VPN**

   Network > Zones > Edit (for Trust)

7. **Routes**

   Network > Routing > Routing Entries > trust-vr New

8. **Policies**

   Policies > (From: Trust, To: Untrust) New

   Policies > (From: Untrust, To: Trust) New

WebUI (Remote Peer)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet1)

   Network > Interfaces > Edit (for ethernet3)

   Network > Interfaces > Edit (for ethernet4)

   Network > Interfaces > New Tunnel IF

2. **Address**

Policy > Policy Elements > Addresses > List > New

3. **VPN Tunnels**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey IKE > New

4. **Asymmetric VPN**

   Network > Zones > Edit (for Trust)

5. **Routes**

   Network > Routing > Routing Entries > trust-vr New

6. **Policies**

   Policies > (From: Trust, To: Untrust) New

   Policies > (From: Untrust, To: Trust) New

CLI

1. **Login and Interfaces**

   Log back into the security device. Then continue with the following configuration:

```
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface tunnel.1 zone trust
set interface tunnel.1 ip unnumbered interface ethernet1
set interface tunnel.2 zone trust
set interface tunnel.2 ip unnumbered interface ethernet1
```

2. **Address**

```
set address untrust peer1 10.2.2.0/24
```

3. **PPPoE**

```
set pppoe name isp1a
set pppoe name isp1a username ns5gt1a password juniper1a
set pppoe name isp1a idle 0
set pppoe name isp1a interface ethernet3
exec pppoe name isp1a connect
set pppoe name isp1b
set pppoe name isp1b username ns5gt1b password juniper1b
set pppoe name isp1b idle 0
set pppoe name isp1b interface ethernet2
exec pppoe name isp1b connect
```

4. **VPN Tunnels**

```
set ike gateway gw1 address 2.2.2.2 aggressive local-id 5gt-e3 outgoing-interface
 ethernet3 preshare netscreen1 sec-level compatible
```

```
set ike gateway gw2 address 3.3.3.3 aggressive local-id 5gt-e2 outgoing-interface
 ethernet2 preshare netscreen2 sec-level compatible
set vpn vpn1 gateway gw1 sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
set vpn vpn1 monitor source-interface ethernet1 destination-ip 2.2.2.2 rekey
set vpn vpn2 gateway gw2 sec-level compatible
set vpn vpn2 bind interface tunnel.2
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
set vpn vpn2 monitor source-interface ethernet1 destination-ip 3.3.3.3 rekey
```

5. **Dual Tunnels**

```
unset failover enable
```

6. **Asymmetric VPN**

```
set zone trust asymmetric-vpn
```

7. **Routes**

```
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.1
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.2
set vrouter trust-vr route 10.2.2.0/24 interface null metric 100
```

8. **Policies**

```
set policy from trust to untrust any any any permit
set policy from untrust to trust peer1 any any permit
save
```

CLI (Remote Peer)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface ethernet4 zone untrust
set interface ethernet4 ip 3.3.3.3/24
set interface tunnel.1 zone trust
set interface tunnel.1 ip unnumbered interface ethernet1
set interface tunnel.2 zone trust
set interface tunnel.2 ip unnumbered interface ethernet1
```

2. **Address**

```
set address untrust ns5gt 10.1.1.0/24
```

3. **VPN Tunnels**

```
set ike gateway gw1 dynamic ns5gt-e3 aggressive outgoing-interface ethernet3
 preshare netscreen1 sec-level compatible
```

```
set ike gateway branch2 dynamic ns5gt-e2 aggressive outgoing-interface
ethernet4 preshare netscreen2 sec-level compatible
set vpn vpn1 gateway gw1 sec-level compatible
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
set vpn vpn2 gateway gw2 sec-level compatible
set vpn vpn2 bind interface tunnel.2
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any
```

4. **Asymmetric VPN**

```
set zone trust asymmetric-vpn
```

5. **Routes**

```
set vrouter trust-vr route 10.1.1.0/24 interface tunnel.1
set vrouter trust-vr route 10.1.1.0/24 interface tunnel.2
set vrouter trust-vr route 10.1.1.0/24 interface null metric 100
```

6. **Policies**

```
set policy from trust to untrust any any any permit
set policy from untrust to trust ns5gt any any permit
save
```

Configuring Interface Failover Using Track IP

In this example, you first configure the NetScreen-5GT for Dual Untrust mode. You then configure the device for automatic failover. If automatic failover from the primary interface to the backup interface occurs, the backup interface carries all traffic to and from the Untrust zone until the primary interface is restored.

For the primary interface, the security device monitors three IP addresses to determine when failover occurs. Each tracked IP address has the following weight:

- 2.2.2.2— Weight = 6
- 3.3.3.3—Weight = 4
- 4.4.4.4— Weight = 4

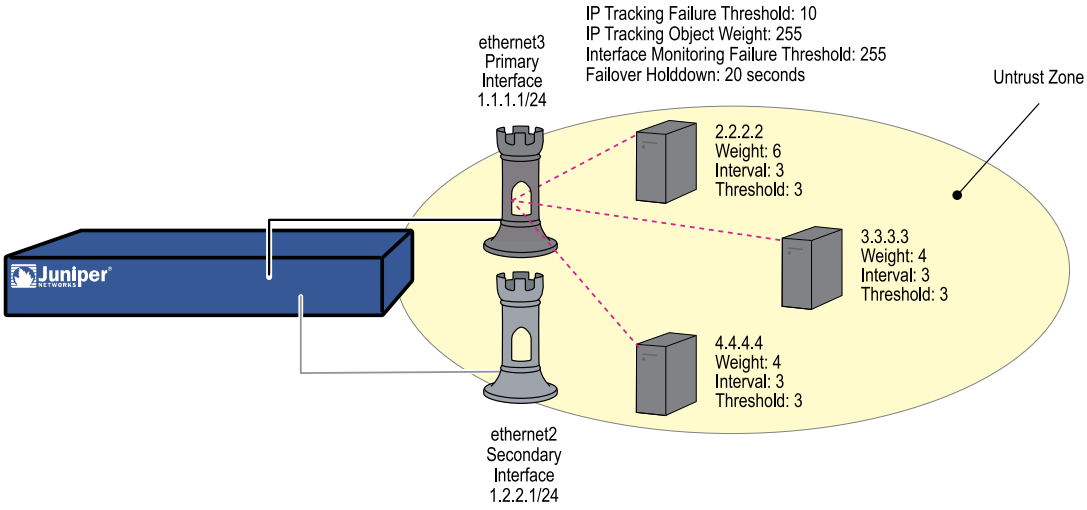
For each of the above tracked IP addresses, the failure threshold is the default value 3 and you set the interval between ICMP echo requests as 3 seconds. If the security device is unable to obtain responses from 3 consecutive ICMP requests to a tracked IP address—each request being three seconds apart—it considers the IP address unreachable through the primary interface.

When an IP tracking failure occurs, the security device adds the weight of the failed address toward a total weight for all IP tracking failures. If the total weight reaches or exceeds the IP tracking object threshold, which in this example you set at 10, then IP tracking adds its weight toward the interface monitoring failure threshold. The IP tracking object weight in this example uses the default value 255 and the interface monitoring failure threshold is also the default value 255.

Therefore, an interface failover occurs if the total weight of IP tracking failures reaches 10. For that to happen, both IP addresses 2.2.2.2 and 3.3.3.3—or 2.2.2.2 and 4.4.4.4—must become unreachable through the primary interface at the same time. If IP addresses 3.3.3.3 and 4.4.4.4 both become unreachable through the primary interface, the cumulative weight of their failures equals 8, which causes no failover to occur.

In the example shown in Figure 458 on page 1852, the interface monitoring failure threshold can be reached in as quickly as 9 seconds (3 failed ICMP requests with 3-second intervals). However, you set a holddown time of 20 seconds so that if the IP tracking weight (255) reaches the interface monitoring failure threshold (255), the security device waits another 20 seconds before failing over the primary interface to the backup.

Figure 458: Interface Failover



WebUI

1. **Login and Interfaces**

   Log in again, and set the interface IP addresses. Then continue with the following configuration:

2. **Automatic Failover and IP Tracking**

   Network > Untrust Failover: Select the following, then click **Apply**:

   Track IP: (select)
   Automatic Failover: (select)
   Failover: (select)
   Failover Holddown Time: 20

   Network > Interfaces > Edit (for ethernet3) > Monitor > Monitor Track IP ADD: Enter the following, then click **Add**:

Static: (select)
Track IP: 2.2.2.2
Weight: 6
Interval: 3
Threshold: 3

Monitor Track IP ADD: Enter the following, then click **Add**:

Static: (select)
Track IP: 3.3.3.3
Weight: 4
Interval: 3
Threshold: 3

Monitor Track IP ADD: Enter the following, then click **Add**:

Static: (select)
Track IP: 4.4.4.4
Weight: 4
Interval: 3
Threshold: 3

Network > Interfaces > Edit (for ethernet3) > Track IP Options: Enter the following, then click **Apply**:

Monitor Option:
Enable Track IP: (select)
Monitor Threshold: 255
Track IP Option:
Threshold: 10
Weight: 255

CLI

1. **Login and Interfaces**

   Log in again, and set the interface IP addresses. Then continue with the following configuration:

2. **Automatic Failover and IP Tracking**

```
set failover enable
set failover auto
set failover holddown 12
set failover type track-ip
set interface ethernet3 track-ip threshold 10
set interface ethernet3 track-ip ip 2.2.2.2 weight 6
set interface ethernet3 track-ip ip 2.2.2.2 interval 3
set interface ethernet3 track-ip ip 2.2.2.2 threshold 3
set interface ethernet3 track-ip ip 3.3.3.3 weight 4
set interface ethernet3 track-ip ip 3.3.3.3 interval 3
set interface ethernet3 track-ip ip 3.3.3.3 threshold 3
set interface ethernet3 track-ip ip 4.4.4.4 weight 4
set interface ethernet3 track-ip ip 4.4.4.4 interval 3
set interface ethernet3 track-ip ip 4.4.4.4 threshold 3
```

```
set interface ethernet3 track-ip
save
```

Configuring Tunnel Failover Weights

In the example shown in Figure 459 on page 1855, you create three pairs of unidirectional VPN tunnels, each pair consisting of a primary tunnel and a backup tunnel. The tunnels connect hosts in the Trust zone at a branch site with DNS, SMTP, and HTTP servers in the Trust zone at the corporate site. All zones at each site are in the trust-vr routing domain.

You first configure the security device protecting the branch site for Dual Untrust mode. You then configure three VPN tunnels with the primary Untrust zone interface (ethernet3) as the outgoing interface and three backup VPN tunnels with the backup Untrust zone interfaces (ethernet2) as the outgoing interface. The security device monitors the primary VPN tunnels to determine when a failover occurs. Each VPN tunnel has the following failover weight:

- vpn1—Weight = 60
- vpn2—Weight = 40
- vpn3—Weight = 40

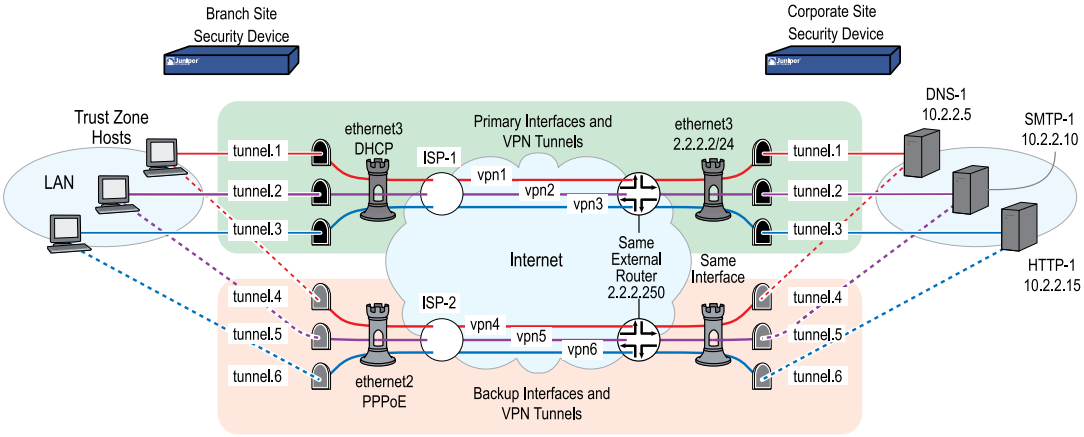
You configure the security device for automatic failover. If automatic failover from the primary interface to the backup interface occurs, the backup interface carries all traffic to and from the Untrust zone until the primary interface is restored. Primary interface failover occurs when the cumulative failover weight reaches or exceeds 100 percent. This means that if both vpn1 and vpn2 are down, the cumulative weight of the failures would be 100 percent, which would cause an automatic failover to the backup interface. If only vpn2 and vpn3 are down, the cumulative weight of the failures would be 80 percent, and no failover occurs.

You also enable the VPN monitor rekey feature. In the event of a failover, this feature allows the security device to revert traffic from the backup interface to the primary if the accumulated weight of the VPN tunnels on the primary interface becomes less than 100 percent.

Finally, you enable the asymmetric VPN option for the Trust zone at each site so that if an existing session established on one VPN tunnel fails over to another, the security device at the other end of the tunnel does not reject it.

The device receives its Untrust zone interfaces address, default gateway, and DNS server addresses dynamically from two different ISPs. Each ISP uses a different protocol. ISP-1 uses DHCP to assign an address to ethernet3, and ISP-2 uses PPPoE to assign an address to ethernet2. The security device at the corporate site has a static IP address (2.2.2.2). The IP address of its default gateway is 2.2.2.250.

Figure 459: Primary and Backup Interfaces and VPN Tunnels



The destination address for VPN monitoring is not the default—the remote gateway IP address (2.2.2.2)—but the addresses of the three servers (10.2.2.5, 10.2.2.10, 10.2.2.15). If you use the remote gateway IP address and it becomes unreachable, then all three primary tunnels always fail over to the backups together at the same time. This defeats the use of weights to cause the failover to occur only when two tunnels (vpn1 + vpn2, or vpn1 + vpn3) fail at the same time. On the other hand, if VPN monitoring targets a different destination address through each tunnel and it can no longer ping DNS-1 through vpn1, no failover occurs. If the NetScreen-5XT then cannot ping SMTP-1 through vpn2, the combined weights total 100 percent (60 + 40) and vpn1 fails over to vpn4 and vpn2 fails over to vpn5, while vpn3 remains active.

NOTE: Because this particular example is long, only the CLI configuration is included in its entirety. The WebUI section simply lists the navigational paths to the pages where you can set the various elements of the configuration. You can see what you need to set by referring to the CLI commands.

WebUI (Branch)

1. **Login and Interfaces**

   Log back into the security device. Then continue with the following configuration:

   Network > Interfaces > Edit (for ethernet1)

   Network > Interfaces > Edit (for ethernet3)

   Network > Interfaces > Edit (for ethernet2)

   Network > Interfaces > New Tunnel IF

2. **VPN Tunnels**

   VPNs > AutoKey Advanced > Gateway > New

VPNs > AutoKey IKE > New

3. **Tunnel Failover**

   Network > Untrust Failover

   Network > Untrust Failover > Edit Weight

4. **Asymmetric VPN**

   Network > Zones > Edit (for Trust)

5. **Routes**

   Network > Routing > Routing Entries > trust-vr New

6. **Policy**

   Policies > (From: Trust, To: Untrust) New

WebUI (Corp)

7. Interfaces

   Network > Interfaces > Edit (for ethernet1)

   Network > Interfaces > Edit (for ethernet3)

   Network > Interfaces > New Tunnel IF

8. **Addresses**

   Policy > Policy Elements > Addresses > List > New

9. **Service Group**

   Policy > Policy Elements > Services > Groups > New

10. **VPN Tunnels**

    VPNs > AutoKey Advanced > Gateway > New

    VPNs > Autokey IKE > New

11. **Asymmetric VPN**

    Network > Zones > Edit (for Trust)

12. **Route**

    Network > Routing > Routing Entries > trust-vr New

13. **Policy**

    Policies > (From: Trust, To: Untrust) New

CLI (Branch)

1. **Login and Interfaces**

   Log back into the security device. Then continue with the following configuration:

```
set interface ethernet1 ip 10.1.1.1/24
set interface ethernet1 nat
set interface ethernet3 dhcp client
exec dhcp client ethernet3 renew
set pppoe interface ethernet2
set pppoe username ns5gt password juniper
set interface tunnel.1 zone trust
set interface tunnel.1 ip unnumbered interface ethernet1
set interface tunnel.2 zone trust
set interface tunnel.2 ip unnumbered interface ethernet1
set interface tunnel.3 zone trust
set interface tunnel.3 ip unnumbered interface ethernet1
set interface tunnel.4 zone trust
set interface tunnel.4 ip unnumbered interface ethernet1
set interface tunnel.5 zone trust
set interface tunnel.5 ip unnumbered interface ethernet1
set interface tunnel.6 zone trust
set interface tunnel.6 ip unnumbered interface ethernet1
```

2. **VPN Tunnels**

```
set ike gateway corp1 address 2.2.2.2 aggressive local-id 5gt-e3
outgoing-interface ethernet3 preshare netscreen1 sec-level basic
set ike gateway corp2 address 2.2.2.2 aggressive local-id 5gt-e2
outgoing-interface ethernet2 preshare netscreen2 sec-level basic

set vpn vpn1 gateway corp1 sec-level basic
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 DNS
set vpn vpn1 monitor source-interface ethernet1 destination-ip 10.2.2.5 rekey
set vpn vpn2 gateway corp1 sec-level basic

set vpn vpn2 bind interface tunnel.2
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 SMTP
set vpn vpn2 monitor source-interface ethernet1 destination-ip 10.2.2.10 rekey
set vpn vpn3 gateway corp1 sec-level basic
set vpn vpn3 bind interface tunnel.3

set vpn vpn3 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 HTTP
set vpn vpn3 monitor source-interface ethernet1 destination-ip 10.2.2.15 rekey
set vpn vpn4 gateway corp2 sec-level basic
set vpn vpn4 bind interface tunnel.4
set vpn vpn4 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 DNS
set vpn vpn4 monitor source-interface ethernet1 destination-ip 10.2.2.5 rekey
set vpn vpn5 gateway corp2 sec-level basic
set vpn vpn5 bind interface tunnel.5
set vpn vpn5 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 SMTP
```

```
set vpn vpn5 monitor source-interface ethernet1 destination-ip 10.2.2.10 rekey
set vpn vpn6 gateway corp2 sec-level basic
set vpn vpn6 bind interface tunnel.6
set vpn vpn6 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 HTTP
set vpn vpn6 monitor source-interface ethernet1 destination-ip 10.2.2.15 rekey
```

NOTE: Usually, the proxy ID can be "local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 any". In the line **set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 SMTP** in the example above, however, the proxy ID for each tunnel must be different to distinguish one tunnel from another. If the service is the same for each proxy ID, a configuration conflict results and the security device rejects the proxy IDs for vpn2 and vpn3 (and vpn5 and vpn6).

3. **Tunnel Failover**

```
set failover type tunnel-if
set failover auto
set vpn vpn1 failover-weight 60
set vpn vpn2 failover-weight 40
set vpn vpn3 failover-weight 40
```

4. **Asymmetric VPN**

```
set zone trust asymmetric-vpn
```

5. **Routes**

```
set vrouter trust-vr route 10.2.2.5/32 interface tunnel.1
set vrouter trust-vr route 10.2.2.10/32 interface tunnel.2
set vrouter trust-vr route 10.2.2.15/32 interface tunnel.3
set vrouter trust-vr route 10.2.2.5/32 interface tunnel.4
set vrouter trust-vr route 10.2.2.10/32 interface tunnel.5
set vrouter trust-vr route 10.2.2.15/32 interface tunnel.6
set vrouter trust-vr route 10.2.2.0/24 interface null metric 100
```

6. **Policy**

```
set policy from trust to untrust any any any permit
save
```

CLI (Corp)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ip 10.2.2.1/24
set interface ethernet1 nat
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
set interface tunnel.1 zone trust

set interface tunnel.1 ip unnumbered interface ethernet1
```

```
set interface tunnel.2 zone trust
set interface tunnel.2 ip unnumbered interface ethernet1
set interface tunnel.3 zone trust
set interface tunnel.3 ip unnumbered interface ethernet1
set interface tunnel.4 zone trust
set interface tunnel.4 ip unnumbered interface ethernet1
set interface tunnel.5 zone trust
set interface tunnel.5 ip unnumbered interface ethernet1
set interface tunnel.6 zone trust
set interface tunnel.6 ip unnumbered interface ethernet1
```

NOTE: Instead of creating six tunnel interfaces—one for each VPN tunnel—you can also create one tunnel interface and bind multiple VPN tunnels to it. The security device uses the Next Hop Tunnel Binding (NHTB) table to differentiate each tunnel. For information about NHTB, see "Multiple Tunnels per Tunnel Interface" on page 983.

2. **Addresses**

```
set address untrust branch 10.1.1.0/24
set address trust DNS-1 10.2.2.5/32
set address trust SMTP-1 10.2.2.10/32
set address trust HTTP-1 10.2.2.15/32
set group address trust servers add DNS-1
set group address trust servers add SMTP-1
set group address trust servers add HTTP-1
```

3. **Service Group**

```
set group service vpn-srv add DNS
set group service vpn-srv add SMTP
set group service vpn-srv add HTTP
set group service vpn-srv add ICMP
```

4. **VPN Tunnels**

```
set ike gateway branch1 dynamic ns5gt-e3 aggressive outgoing-interface ethernet3
  preshare netscreen1 sec-level basic
set ike gateway branch2 dynamic ns5gt-e2 aggressive outgoing-interface ethernet3
  preshare netscreen2 sec-level basic
set vpn vpn1 gateway branch1 sec-level basic
set vpn vpn1 bind interface tunnel.1
set vpn vpn1 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 DNS
set vpn vpn2 gateway branch1 sec-level basic
set vpn vpn2 bind interface tunnel.2
set vpn vpn2 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 SMTP
set vpn vpn3 gateway branch1 sec-level basic
set vpn vpn3 bind interface tunnel.3
set vpn vpn3 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 HTTP
set vpn vpn4 gateway branch2 sec-level basic
set vpn vpn4 bind interface tunnel.4
set vpn vpn4 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 DNS
set vpn vpn5 gateway branch2 sec-level basic
set vpn vpn5 bind interface tunnel.5
```

```
set vpn vpn5 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 SMTP
set vpn vpn6 gateway branch2 sec-level basic
set vpn vpn6 bind interface tunnel.6
set vpn vpn6 proxy-id local-ip 0.0.0.0/0 remote-ip 0.0.0.0/0 HTTP
```

5. **Asymmetric VPN**

```
set zone trust asymmetric-vpn
```

6. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
```

7. **Policy**

```
set policy from untrust to trust branch servers vpn-srv permit
save
```

Configuring Virtual System Failover

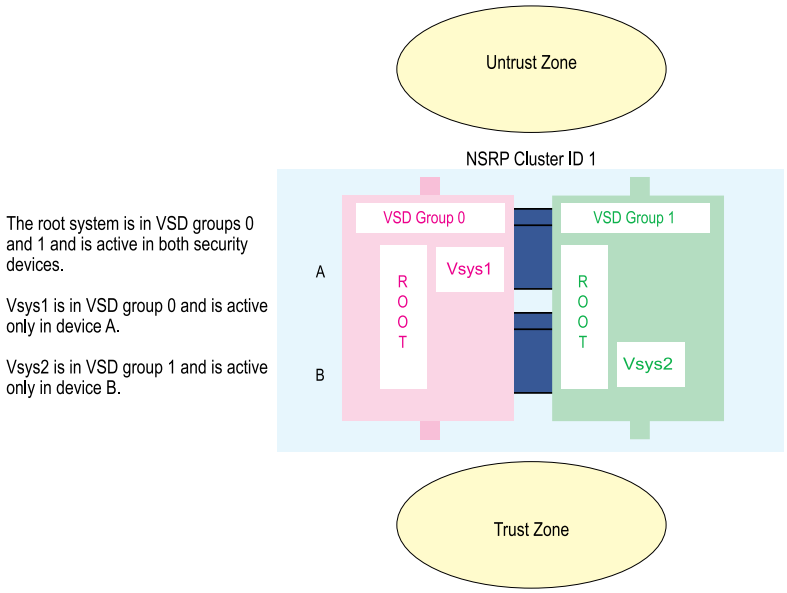
In the example shown in Figure 460 on page 1861, you configure two virtual systems (vsys1 and vsys2) for NSRP. To provide load sharing for incoming traffic to the virtual systems, VSD membership is apportioned as follows:

- Vsys1 is a member of VSD group 0.
- Vsys2 is a member of VSD group 1.

NOTE: In Figure 460 on page 1861, the load is not evenly distributed or load balanced. The two security devices share the load, with devices A and B receiving incoming traffic in dynamically shifting proportions (60/40 percent, 70/30 percent, and so on).

The security devices share the incoming traffic load by distributing the VSD groupings of the virtual systems. Because of the initial design of configuring vsys1 on device A and vsys2 on device B, incoming traffic to these virtual systems is directed to the device that contains it.

Figure 460: Virtual Systems in an NSRP Configuration

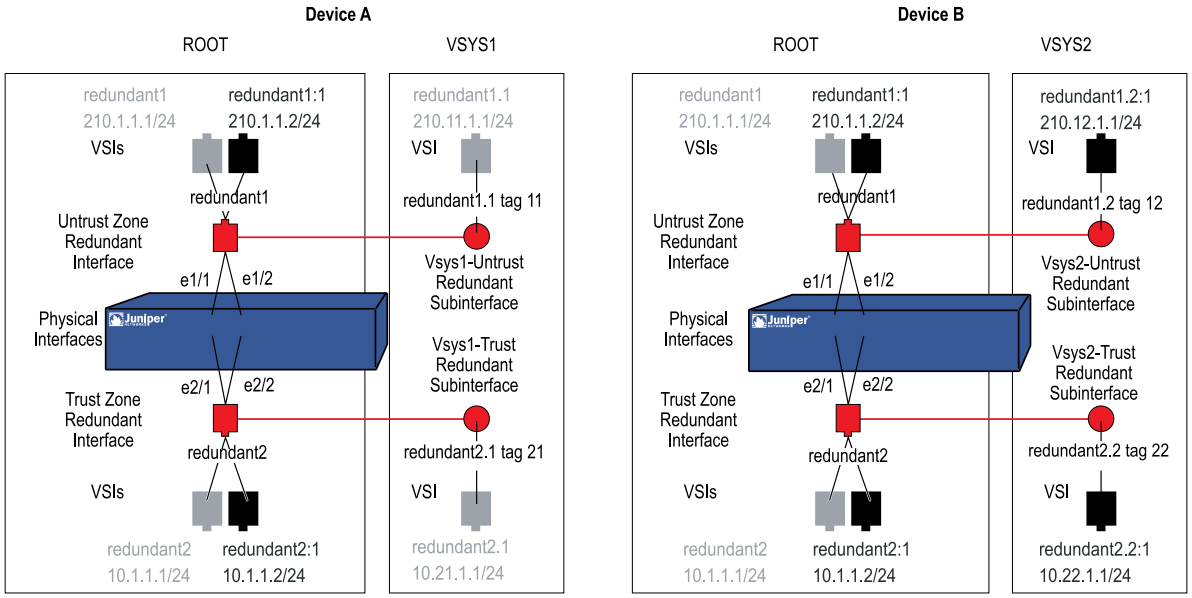


The default gateway for outbound traffic is different for the root system and each virtual system:

- Root: 210.1.1.250
- Vsys1: 210.11.1.250
- Vsys2: 210.12.1.250

Because Figure 461 on page 1862 builds on "Active/Active Configuration" on page 1778, in which you set up VSD groups 0 and 1 and set the devices in NSRP cluster ID 1, NSRP is already enabled. Therefore, the settings you configure on device A automatically propagate to device B.

Figure 461: Relationship of Physical Interfaces, Redundant Interfaces, Subinterfaces, and VSIs



VSD Group 0: VSIs for VSD 0 do not display their VSD ID number.
VSD Group 1: VSIs for VSD 1 indicate their VSD ID by colon+1.

WebUI

1. **Device A: Root**

NOTE: The NSRP configuration for the root system is identical to that in "Active/Active Configuration" on page 1778.

---

2. **Device A: Vsys1**

Vsys > New: Enter the following, then click **OK**:

VSYS Name: vsys1

NOTE: If you do not define a vsys admin, the security device automatically creates one by appending "vsys_" to the vsys name. In this example, the vsys admin for vsys1 is vsys_vsys1.

Vsys > Enter (vsys1) > Network > Interface > New Sub-IF: Enter the following, then click **OK**:

Interface Name: Redundant1.1
Zone Name: Untrust
VLAN Tag: 11

Network > Interfaces > New VSI IF: Enter the following, then click **OK**:

VSI Base: Redundant1.1
VSD Group: 0
IP Address / Netmask: 210.11.1.1/24

Network > Interfaces > New Sub-IF: Enter the following, then click **OK**:

Interface Name: Redundant2.1
Zone Name: Trust-vsys-vsys1
VLAN Tag: 21

Network > Interfaces > New VSI IF: Enter the following, then click **OK**:

VSD Group ID: 0
IP Address / Netmask: 10.21.1.1/24
Interface Mode: Route

NOTE: Virtual systems can be in either route or NAT mode, independent of the mode you set at the root level.

Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: Redundant1
Gateway IP Address: 210.11.1.250

Click **Exit Vsys** to return to the root level.

3. **Device A: Vsys2**

Vsys > New: Enter the following, then click **OK**:

VSYS Name: vsys2

Vsys > Enter (vsys2) > Network > Interface > New Sub-IF: Enter the following, then click **OK**:

Interface Name: Redundant1.2
Zone Name: Untrust
VLAN Tag: 12

Network > Interfaces > New VSI IF: Enter the following, then click **OK**:

VSI Base: Redundant1.2
VSD Group: 1
IP Address / Netmask: 210.12.1.1

Network > Interfaces > New Sub-IF: Enter the following, then click **OK**:

Interface Name: Redundant2.2
Zone Name: Trust-vsys-vsys2
VLAN Tag: 22

Network > Interfaces > New VSI IF: Enter the following, then click **OK**:

VSD Group ID: 1
IP Address / Netmask: 10.22.1.1/24
Interface Mode: Route

Network > Routing > Routing Entries > untrust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: Redundant1
Gateway IP Address: 210.12.1.250

Click **Exit Vsys** to return to the root level.

4. **Device B**

NOTE: Because device A propagates the other configuration settings to device B, you do not need to enter them again in device B.

CLI

1. **Device A: Root**

NOTE: The NSRP configuration for the root system is identical to that in "Active/Active Configuration" on page 1778.

2. **Device A: VSYS 1**

```
set vsys vsys1
ns(vsys1)-> set interface redundant1.1 tag 11 zone untrust
ns(vsys1)-> set interface redundant1.1 ip 210.11.1.1/24
ns(vsys1)-> set interface redundant2.1 tag 21 zone trust-vsys1
ns(vsys1)-> set interface redundant2.1 ip 10.21.1.1/24
ns(vsys1)-> set interface redundant2.1 route
ns(vsys1)-> set vrouter untrust-vr route 0.0.0.0/0 interface redundant1 gateway
 210.11.1.250
ns(vsys1)-> save
ns(vsys1)-> exit
```

NOTE: Virtual systems can be in either route or NAT mode, independent of the mode you set at the root level.

3. **Device A: VSYS 2**

```
set vsys vsys2
ns(vsys2)-> set interface redundant1.2 tag 12 zone untrust
ns(vsys2)-> set interface redundant1.2:1 ip 210.12.1.1/24
ns(vsys2)-> set interface redundant2.2 tag 22 zone trust-vsys2
ns(vsys2)-> set interface redundant2.2:1 ip 10.22.1.1/24
ns(vsys2)-> set interface redundant2.2:1 route
ns(vsys2)-> set vrouter untrust-vr route 0.0.0.0/0 interface redundant1 gateway
 210.12.1.250
ns(vsys2)-> save
ns(vsys2)-> exit
```

4. **Device B**

NOTE: Device A propagates the other configuration settings to device B, so you do not need to enter them again in device B.

Part 12
WAN, DSL, Dial, and Wireless

WAN, DSL, Dial, and Wireless describes the wide area network (WAN), digital subscriber line (DSL), dial, and wireless local area network (WLAN) interfaces. This guide contains the following sections:

- "Wide Area Networks" on page 1869 describes how to configure a wide area network (WAN).
- "Digital Subscriber Line" on page 1949 describes the asymmetric digital subscriber line (ADSL) and G.symmetrical digital subscriber line (G.SHDSL) interfaces.
- "ISP Failover and Dial Recovery" on page 1995 describes how to set priority and define conditions for ISP failover and how to configure a dialup recovery solution.
- "Wireless Local Area Network" on page 2001 describes the wireless interfaces on Juniper Networks wireless devices and provides example configurations.
- "Wireless Information" on page 2267 lists available channels, frequencies, and regulatory domains and lists the channels that are available on wireless devices for each country.

Chapter 58

Wide Area Networks

Some security devices allow you to use wide area network (WAN) data links to transmit and receive traffic across geographically dispersed networks. These networks can be privately owned but more typically include public or shared networks. Certain properties must be configured before WAN links can operate correctly, such as the clocking and signal-handling options (for the physical line) and the encapsulation method (for transferring data across the WAN).

This chapter contains the following sections:

WAN Overview

This section defines the following WAN interfaces:

Serial

Serial links provide bidirectional links that require very few control signals. In a basic serial setup, the data circuit-terminating equipment (DCE) is responsible for establishing, maintaining, and terminating a connection. A modem is a typical DCE device. A serial cable connects the DCE to a telephony network where, ultimately, a link is established with data terminal equipment (DTE). DTE is typically where a link terminates.

Some security devices support the following types of serial interfaces:

- TIA/EIA 530—The Telecommunications Industry Association/Electronics Industries Alliance (TIA/EIA) Standard 530, *High-Speed 25-Position Interface for Data Terminal Equipment and Data Circuit-Terminating Equipment*, describes the interconnection of DTE and DCE using serial binary data interchange with control information exchanged on separate control circuits.
- V.35—The Telecommunication Standardization Sector of the International Telecommunications Union (ITU-T) Recommendation V.35, *Data Transmission at 48 kbit/s Using 60-108 kHz Group Band Circuits*, describes a synchronous, Physical Layer protocol used for communications between a network access device and a packet network. V.35 is most commonly used in the United States and Europe.
- X.21—The ITU-T Recommendation X.21, *Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment for Synchronous Operation on Public Data Networks*, describes serial communications over synchronous digital lines. The X.21 protocol is used primarily in Europe and Japan.
- RS-232—TIA/EIA-232-F (the current revision), *Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange*, describes the physical interface and protocol for communication with modems and other serial devices.
- RS-449—The EIA standard *EIA-449 General Purpose 37-Position and 9-Position Interface for Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange*, specifies the interface between DTE and DCE.

When a serial connection is made, a serial line protocol—such as EIA-530, X.21, RS-422/449, RS-232, or V.35—begins controlling the transmission of signals across the line as follows:

1. The DCE transmits a DSR signal to the DTE, which responds with a DTR signal. After this transmission, the link is established and traffic can pass.
2. When the DTE device is ready to receive data, it sets its RTS signal to a marked state (all 1s) to indicate to the DCE that it can transmit data.
3. When the DCE device is ready to receive data, it sets its clear-to-send (CTS) signal to a marked state to indicate to the DTE that it can transmit data.
4. When the negotiation to send information has taken place, data is transmitted across the transmitted data (TD) and received data (RD) lines:
   - TD line—Line through which data from a DTE device is transmitted to a DCE device.
   - RD line—Line through which data from a DCE device is transmitted to a DTE device.

T1

T1, also known as data signal 1 (DS1), is a digital data transmission medium capable of handling 24 simultaneous connections running at a combined 1.544 Mbps. T1

combines these 24 separate connections, called *channels* or *timeslots*, onto a single link.

The T1 data stream is broken into frames. Each frame consists of a single framing bit and 24 8-bit channels, totalling 193 bits per T1 frame. Frames are transmitted 8,000 times per second, at a data transmission rate of 1.544 Mbps (8,000 x 193 = 1.544 Mbps). As each frame is received and processed, the data in each 8-bit channel is maintained with the channel data from previous frames, enabling T1 traffic to be separated into 24 separate flows across a single medium.

Supported T1 standards include:

- American National Standards Institute (ANSI) T1.107, *Digital Hierarchy - Formats Specifications,* describes digital-hierarchy formats and is used in conjunction with T1.102, *Digital Hierarchy - Electrical Interfaces.*
- Telcordia GR 499-CORE, *Transport Systems Generic Requirements (TSGR): Common Requirements,* describes basic generic requirements common to transport systems. Telcordia GR 253-CORE, *Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria,* describes generic SONET criteria.
- AT&T Technical Reference 54014, *ACCUNET T45 and T45R Service Description and Interface Specification,* describes the service description and interface specification for AT&T ACCUNET T45 and T45R services.
- International Telecommunications Union (ITU-T) Recommendations G.751 and G.703 describe physical and electrical characteristics of hierarchical digital interfaces.

E1

E1 is the European format for DS1 digital transmission. E1 links are similar to T1 links except that they carry signals at 2.048 Mbps. Each signal has 32 channels, and each channel transmits at 64 Kbps. E1 links have higher bandwidth than T1 links because they use all 8 bits of a channel.

The following standards apply to E1 interfaces:

- ITU-T Recommendation G.703, *Physical/Electrical Characteristics of Hierarchical Digital Interfaces*, describes data rates and multiplexing schemes.
- ITU-T Recommendation G.751, *General Aspects of Digital Transmission Systems: Terminal Equipment*, describes framing methods.
- ITU-T Recommendation G.775, *Loss of Signal (LOS) and Alarm Indication Signal (AIS) Defect Detection and Clearance Criteria*, describes alarm-reporting methods.

T3

T3, also known as data signal 3 (DS3), is a high-speed data-transmission medium formed by multiplexing 28 DS1signals into seven separate DS2 signals, and combining the DS2 signals into a single DS3 signal. T3 links operate at 43.736 Mbps.

Supported T3 standards include:

- American National Standards Institute (ANSI) T1.107, *Digital Hierarchy - Formats Specifications,* describes digital-hierarchy formats and is used in conjunction with T1.102, *Digital Hierarchy - Electrical Interfaces.*
- Telcordia GR 499-CORE, *Transport Systems Generic Requirements (TSGR): Common Requirements*, describes basic generic requirements common to transport systems. Telcordia GR 253-CORE, *Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria*, describes generic SONET criteria.
- Telcordia TR-TSY-000009, A*synchronous Digital Multiplexes, Requirements and Objectives*, describes generic technical requirements and objectives for asynchronous multiplexes that operate at DS1C (3.152 Mbps), DS2 (6.312 Mbps), and/or DS3 (44.736 Mbps) digital rates.
- AT&T Technical Reference 54014, *ACCUNET T45 and T45R Service Description and Interface Specification*, describes the service description and interface specification for AT&T ACCUNET T45 and T45R Services.
- ITU G.751, *Digital multiplex equipment operating at the third order bit rate of 34 368 kbit/s and the fourth order bit rate of 139 264 kbit/s and using positive justification*, G.703, *Physical/electrical characteristics of hierarchical digital interfaces*, and G.823, *The control of jitter and wander within digital networks which are based on the 2048 kbit/s hierarchy*, describe transmission systems and media, digital systems, and networks.

E3

E3 is the European equivalent of T3. It is formed by multiplexing 16 separate E1 signals together. It operates at 34.368 Mbps.

Supported E3 standards include:

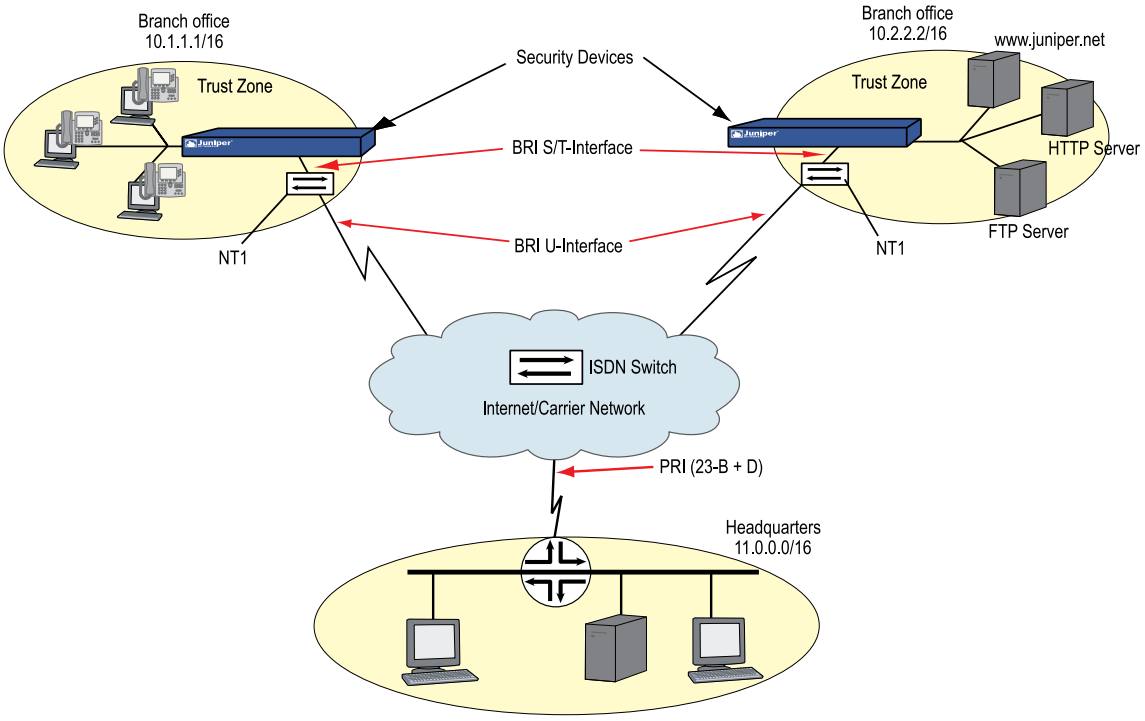
- ITU G.751, *Digital multiplex equipment operating at the third order bit rate of 34 368 kbit/s and the fourth order bit rate of 139 264 kbit/s and using positive justification*, G.703, *Physical/electrical characteristics of hierarchical digital interfaces*, and G.823, *The control of jitter and wander within digital networks which are based on the 2048 kbit/s hierarchy*, describe transmission systems and media, digital systems, and networks.
- Telcordia GR 499-CORE, *Transport Systems Generic Requirements (TSGR): Common Requirements*, describes basic generic requirements common to transport systems. Telcordia GR 253-CORE, *Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria*, describes generic SONET criteria.

ISDN

Integrated Services Digital Network (ISDN) is an international communications standard for sending voice, video, and data over digital telephone lines. As a dial-on-demand service, ISDN has fast call setup and low latency as well as the ability to carry high-quality voice, data, and video transmissions. ISDN is also a circuit-switched service that can be used on both multipoint and point-to-point connections.

Figure 462 on page 1873 illustrates a basic setup for ISDN connectivity. The Branch office is connected to the Corporate headquarters using ISDN. The connection is automatically established for any request to send a packet to the Internet and the connection is dropped after a set number of seconds elapses with no traffic. Because ISDN connections typically takes a few milliseconds to establish (almost instantaneously) the connection can be easily made and broken as demand warrants.

Figure 462: Basic ISDN Topology



In North America, most Carriers provide a U-interface for ISDN connectivity. To communicate with your security device, you must use additional equipment, Network Termination unit (NT1), to convert the U-interface to a S/T-interface. Juniper Networks security devices are provided with the S/T-interface only. The NT1 is located at the customer's site (see Branch Offices in Figure 462 on page 1873) and may be provided by the Carrier.

ISDN in ScreenOS supports the following features on your security device:

- Dial-on-Demand Routing (DDR)

  DDR allows the security device to automatically initiate and close a session as transmitting stations demand. The device spoofs keepalives so that end stations treat the session as active. DDR lets the user bring up WAN links only when necessary and thus reduce remote-site access costs.

- Basic Rate Interface (BRI)

  BRI is also called 2B + D, because it consists of two 64 Kbps B-channels and one 16 Kbps D-channel. The B-channels are used for user data, and the D-channel

is responsible for carrying signaling traffic to establish and terminate connections between sites.

Each ISDN BRI uses the naming convention bri*x*/0, where *x* = slot-id and *x*/0 represents **slot-id/port-id**. The two B-channels for bri0/0, for example, are identified as *bri0/0.1* and *bri0/0.2*.

- Bandwidth on Demand

  The two 64-Kbps B-channels can be combined to form a single 128-Kbps connection as needed. The device supports bandwidth-on-demand on the ISDN interface as follows:

  - Brings up more channel when traffic is beyond configured threshold
  - Disconnects channel when traffic is less than the configured threshold

  Bandwidth on demand is implemented on your security device using multilink PPP (MLPPP) encapsulation.

- Dialer interface and dialer pool

  The dialer interface and dialer pool allows the ISDN interface to dial out to multiple destinations when the number of destinations exceeds the number of available physical lines. The ISDN interface can belong to more than one pool, allowing a single line to be used to dial out to more than one destination.

  For more information on this feature, see "Dialing Out Using the Dialer Interface" on page 1927.

- Dial backup

  You can use the ISDN interface for dial backup, to activate a secondary WAN link when a primary synchronous line fails.

- Leased Line

  The ISDN leased line is supported for 128Kbps. In leased line mode, the ISDN interface operates as Layer 3 interface that can only deliver data, so the D-channel is not required.

- Monitor ISDN and Dialer interfaces

WAN Interface Options

This section explains the WAN interfaces options that are available on some of the security devices. Table 132 on page 1874 displays which physical attributes are available on the WAN interfaces.

Table 132: WAN Interface Physical Attributes

Physical Attributes	Serial	T1	E1	T3	E3	ISDN (BRI)
Hold time	X	X	X	X	X	X

Table 132: WAN Interface Physical Attributes *(continued)*

Physical Attributes	Serial	T1	E1	T3	E3	ISDN (BRI)
DTE options	X					
Frame checksum		X	X	X	X	
Idle-cycle Flag		X	X	X	X	X
Start/End Flag		X	X	X	X	
Signal Handling	X					
Clocking						
Clocking Mode	X					
Clocking source		X	X	X	X	
Internal Clock rate	X					
Transmit clock Inversion	X					
Time Slots						
Fractional T1 Time Slots		X				
Fractional E1 Time Slots			X			
Line Encoding						
AMI		X				
B8ZS		X				
HDB3			X			
Byte Encoding		X				
Data Inversion		X	X			
Framing						
Superframe		X				
Extended Frame		X				
G.704 Frame			X			
G.751 Frame					X	
C-Bit Parity Frame				X		
Loopback Signal						

Table 132: WAN Interface Physical Attributes *(continued)*

Physical Attributes	Serial	T1	E1	T3	E3	ISDN (BRI)
Loopback Mode		X			X	X
Bit Error Rate Test (BERT)		X	X	X	X	
CSU Compatibility Mode				X		
Remote loopback response		X		X	X	
FEAC Response				X		
ISDN Options						
Switch type						X
SPID1						
SPID2						X
TEI negotiation						X
Calling number						X
T310 value						X
Send complete						X
BRI Mode (leased line/dialer)						X
Dialer Options						
Primary/alternate numbers						X
Load Threshold						X
Idle time						X
Retry times						X
Interval						X
Dialer pool						X

Hold Time

Hold time specifies how much time can pass before the device considers the interface connection to be up or down. The hold time is useful in situations where an interface is connected to an add-drop multiplexer (ADM) or a wavelength-division multiplexer (WDM), or to protect against Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) framer holes because you might not want the interface to advertise that its connection status is up or down.

For example, if an interface goes from up to down, it is not advertised to the rest of the system as being down until it has remained down for the specified hold-time period. Similarly, an interface is not advertised to the rest of the system as being up until it has remained up for the specified hold-time period.

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Enter the following, then click **Apply**:

Hold time
Down: 500
Up: 500

CLI

```
set interface interface hold-time { down 500 | up 500 }
save
```

NOTE: A 0 hold-time indicates that the interface drops traffic when the device receives an message that the interface is down.

Frame Checksum

Frame checksum verifies that frames passing through a device are valid using a bit-encoding scheme. Some WAN interfaces use a 16-bit frame checksum, but can configure a 32-bit checksum to provide more reliable packet verification.

To configure the WAN interface to use a 32-bit checksum (x is t1, e1, or t3):

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: For the **Frame Checksum** option select **32-bits**, then click **Apply**.

CLI

```
set interface interface x-options fcs 32
save
```

Idle-cycle Flag

An idle cycle is the duration when the device has no data to transmit. Idle-cycle flags allow some WAN interfaces to transmit the value 0x7E in the idle cycles. To configure the WAN interface to transmit the value 0xFF (all ones) (x is t1, e1, t3, or bri):

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: For the **Idle-cycle Flags** option select **0xFF (All ones)**, then click **Apply**.

CLI

```
set interface interface x-options idle-cycle-flag ones
save
```

Start/End Flag

Start and end flags for T1 or E1 interfaces wait two idle cycles between sending a start and an end flag. To configure the interface to share the transmission of start and end flags (x is either t1 or e1):

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: For the **Start/End Flags on Transmission** option select **Shared**, then click **Apply**.

CLI

```
set interface interface x-options start-end-flag shared
```

To share the transmission of start and end flags on a T3 interface:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Select the Line encoding type, then click **Apply**.

CLI

```
set interface interface t3-options start-end-flag
save
```

Line Encoding

Following are common T1 and E1 encoding techniques:

- Alternate mark inversion (AMI)—T1 and E1
- Bipolar with 8-zero substitution (B8ZS)—T1 only
- High-density bipolar 3 code (HDB3)—E1 only

To change the encoding type (x is either t1 or e1):

WebUI

Network > Interfaces > List > Edit (*X Interface*) > WAN: Select the line encoding type, then click **Apply**.

CLI

```
set interface interface x-options line-encoding option
save
```

Alternate Mark Inversion Encoding

AMI encoding forces the 1s signals on a T1 or E1 line to alternate between positive and negative voltages for each successive 1 transmission. When Alternate Mark Inversion (AMI) encoding is used, a data transmission with a long sequence of 0s has no voltage transitions on the line. In this situation, devices have difficulty maintaining clock synchronization, because they rely on the voltage fluctuations to constantly synchronize with the transmitting clock. To counter this effect, the number of consecutive 0s in a data stream is restricted to 15. This restriction is called the 1s density requirement, because it requires a certain number of 1s for every 15 0s that are transmitted. On an AMI-encoded line, two consecutive pulses of the same polarity—either positive or negative—are called a bipolar violation (BPV), which is generally flagged as an error.

Data Inversion

When you enable data inversion, all data bits in the datastream are transmitted inverted; that is, zeroes are transmitted as ones, and ones are transmitted as zeroes. Data inversion is normally used only in AMI mode to provide the density in the transmitted stream. To enable data inversion:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Check the Invert Data check box, then click **Apply**.

CLI

```
set interface interface t1-options invert-data
save
```

B8ZS and HDB3 Line Encoding

Both bipolar with 8-zero substitution (B8ZS) and high-density bipolar 3 code (HDB3) encoding do not restrict the number of 0s that can be transmitted on a line. Instead, these encoding methods detect sequences of 0s and substitute bit patterns in their place to provide the signal oscillations required to maintain timing on the link. The B8ZS encoding method for T1 lines detects sequences of eight consecutive 0 transmissions and substitutes a pattern of two consecutive BPVs (11110000). Because the receiving end uses the same encoding, it detects the BPVs as 0s substitutions,

and no BPV error is flagged. A single BPV, which does not match the 11110000 substitution bit sequence, is likely to generate an error, depending on the configuration of the device.

The HDB3 encoding method for E1 lines detects sequences of four consecutive 0 transmissions and substitutes a single BPV (1100). Similar to B8ZS encoding, the receiving device detects the 0s substitutions and does not generate a BPV error.

Byte Encoding

A T1 interface uses byte encoding of 8 bits per byte (nx64). You can configure an alternative byte encoding of 7 bits per byte (nx56). To configure the interface byte encoding:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Select the byte encoding type, then click **Apply**.

CLI

```
set interface interface t1-options byte-encoding option
save
```

Line Buildout

Some WAN interfaces allow you to configure the line buildout, which is the programmable distance between the device and your main office. A T1 interface has five possible setting ranges for the line buildout:

- 0 to 132 feet (ft) (0 to 40 meters (m))
- 133 to 265 ft (40 to 81 m)
- 266 to 398 ft (81 to 121 m)
- 399 to 531 ft (121 to 162 m)
- 532 to 655 ft (162 to 200 m)

A T3 interface has two settings for the T3 line buildout: a short setting, which is less than 255 feet (about 68 meters), and a long setting, which is greater than 255 feet and less than 450 feet (about 137 meters).

To set the interface line range (X is either t1 or t3) or CLI:

WebUI

Network > Interfaces > List > Edit (X Interface) > WAN: Select the line buildout range, then click **Apply**.

CLI

```
set interface interface t1-options buildout range
save
or
set interface interface t3-options long-buildout
save
```

Framing Mode

T1 interface uses two types of framing: superframe (SF) and extended superframe (ESF). E1 interfaces use G.704 framing or G.704 with no CRC4 framing, or can be in unframed mode. E3 interfaces use G.751 framing or can be in unframed mode.

To configure framing for the WAN interface (x is t1, e1, t3, or e3):

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Select the framing mode, then click **Apply**.

CLI

```
set interface interface x-options framing options
```

Superframe for T1

A superframe (SF), also known as a D4 frame, consists of 192 data bits: 24 8-bit channels and a single framing bit. The single framing bit is part of a 12-bit framing sequence. The 193rd bit in each T1 frame is set to a value, and every 12 consecutive frames are examined to determine the framing bit pattern for the 12-bit SF. The receiving device detects these bits to synchronize with the incoming data stream and determine when the framing pattern begins and ends.

Extended Superframe for T1

Extended superframe (ESF) extends the D4 SF from 12 frames to 24 frames, which also increases the bits from 12 to 24. The extra bits are used for frame synchronization, error detection, and maintenance communications through the facilities data link (FDL). Only the framing bits from frames 4, 8, 12, 16, 20, and 24 in the SF sequence are used to create the synchronization pattern.

C-Bit Parity Framing for T3

C-bit parity mode controls the type of framing that is present on the transmitted T3 signal. When C-bit parity mode is enabled, the C-bit positions are used for the Far End Block Error (FEBE), Far-End Alarm and Control (FEAC), terminal data link, path parity, and mode indicator bits, as defined in ANSI T1.107a-1989. When C-bit parity mode is disabled, the basic T3 framing mode (M13) is used.

To disable C-bit parity mode and use M13 framing for your T3 interface:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Select the framing mode, then click **Apply**.

CLI

```
unset interface interface t3-options cbit-parity
save
```

Clocking

Clocking determines how networks sample transmitted data. As streams of information are received by a router in a network, a clock source specifies when to sample the data. Some WAN interfaces allow you to configure the following clocking information:

- Clocking Mode on page 1882
- Clocking Source on page 1883
- Internal Clock Rate on page 1884
- Transmit Clock Inversion on page 1885

Clocking Mode

There are three clocking modes:

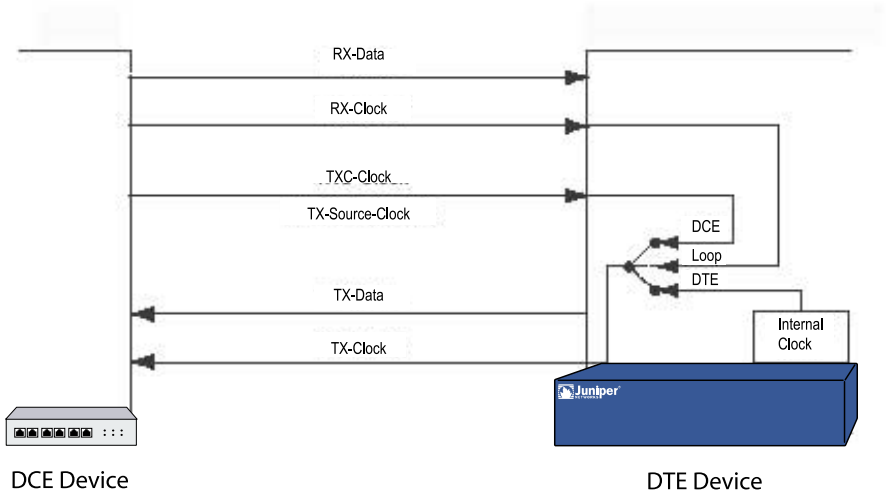
- **Loop clocking mode** uses the DCE Receive (RX) clock to clock data from the DCE to the DTE.
- **DCE clocking mode** uses the DTE Transmit (TX) clock, which the DCE generates for the DTE to use as the transmit clock for DTE.
- **Internal clocking mode**, also known as *line timing*, uses an internally generated clock.

NOTE: For TIA/EIA 530, V.35, RS0232, and RS-449 interfaces, you can configure each interface independently to use loop, DCE, or internal clocking mode. For X.21 interfaces, only loop clocking mode is supported.

DCE clocking mode and loop clocking mode use external clocks generated by the DCE.

Figure 463 on page 1883 shows the clock sources for loop, DCE, and internal clocking modes.

Figure 463: Serial Interface Clocking Mode



To configure the clocking mode of a serial interface:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN:

Hold Time
Clock Mode

Select the clocking mode, then click **Apply**.

CLI

set interface *interface* serial-options clocking-mode { dce | internal | loop }

Clocking Source

The clock source can be the internal stratum 3 clock, which resides on the control board, or an external clock that is received from the interface you are configuring.

By default, the interface clocking source is internal, which means that each interface uses the internal stratum 3 clock. For interfaces that can use different clock sources, the source can be internal (also called *line timing* or *normal timing*) or external (also called *loop timing*).

To set the clock source of an interface to use an external clock:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN:

Hold Time
Clocking

Select the clocking mode, then click **Apply**.

CLI

set interface *interface* clocking external

Internal Clock Rate

The internal clock rate is the speed of the internal clock which is typically used with the internal clocking mode.

NOTE: For RS-232 interfaces with internal clocking mode configured, the clock rate must be less than 20 KHz.

To configure the clock rate:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Select the following, then click **Apply**:

Hold Time
Clock Rate: Select the rate

CLI

set interface *interface* serial-options clock-rate *number*
save

You can configure the following interface rates:

1.2 KHz	56.0 KHz	250.0 KHz	1.3 Mhz
2.4 KHz	64.0 KHz	500.0 KHz	2.0 Mhz
9.6 KHz	72.0 KHz	800.0 KHz	4.0 Mhz
19.2 KHz	125.0 KHz	1.0 Mhz	8.0 Mhz
38.4 KHz	148.0 KHz		

Although the WAN interface is intended for use at the default rate of 8.0 MHz, you might need to use a slower rate under any of the following conditions:

- The interconnecting cable is too long for effective operation.
- The interconnecting cable is exposed to an extraneous noise source that might cause an unwanted voltage in excess of + 1 volt measured differentially between the signal conductor and circuit common at the load end of the cable, with a 50-ohm resistor substituted for the generator.

- You need to minimize interference with other signals.
- You need to invert one or more signals.

NOTE: For TIA/EIA 530, V.35, RS-232, and RS-449 interfaces with internal clocking mode enabled, you can configure the clock rate. For more information about internal clocking mode, see "Clocking Mode" on page 1882.

For detailed information about the relationship between signaling rate and interface-cable distance, refer to the following standards:

- EIA 422-A, *Electrical Characteristics of Balanced Voltage Digital Interface Circuits*
- EIA 423-A, *Electrical Characteristics of Unbalanced Voltage Digital Interface Circuits*

Transmit Clock Inversion

The transmit clock aligns each outgoing packet transmitted over the WAN interfaces. When the device uses externally timed clocking mode (DCE or loop), long cables might introduce a phase shift of the DTE-transmitted clock and data. At high speeds, this phase shift could cause errors. Inverting the transmit clock corrects the phase shift, thereby reducing error rates.

To set the transmit to be inverted:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Select the following, then click **Apply**:

Serial Options
Transmit clock invert: (select)

CLI

```
set interface interface serial-options transmit-clock invert
save
```

Signal Handling

Normal signal handling is defined by the following standards:

- TIA/EIA Standard 530
- ITU-T Recommendation V.35
- ITU-T Recommendation X.21

Table 133 on page 1886 shows the serial-interface modes that support each signal type.

Table 133: Signal Handling by Serial-Interface Type

Signal	Serial Interfaces
From-DCE signals:	
Clear-to-Send (CTS)	TIA/EIA 530, V.35, RS-232, RS-449
Data-Carrier-Detect (DCD)	TIA/EIA 530, V.35, RS-232, RS-449
Data-Set-Ready (DSR)	TIA/EIA 530, V.35, RS-232, RS-449
Test-Mode (TM)	TIA/EIA 530 only
To-DCE signals:	
Data-Transfer-Ready (DTR)	TIA/EIA 530, V.35, RS-232, RS-449
Request-to-Send (RTS)	TIA/EIA 530, V.35, RS-232, RS-449

To configure serial interface characteristics:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Select the following, then click **Apply**:

DTE Options
Select your options

CLI

```
set interface interface serial-options dte-options { ... }
save
```

NOTE: If **ignore-all** is specified, other signal-handling options cannot be configured.

Loopback Signal

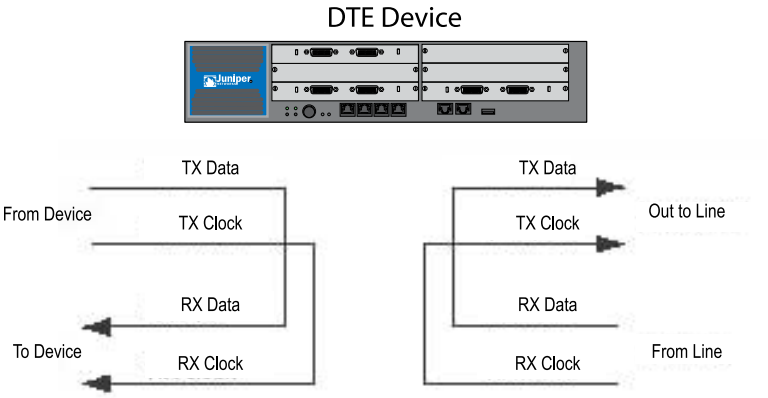
The control signal on a T1, E1, T3, or E3 link is the loopback signal. Using the loopback signal, the operators at the network control center can force the device at the remote end of a link to retransmit its received signals back onto the transmit path. The transmitting device can then verify that the received signals match the transmitted signals to perform end-to-end checking on the link.

Remote and Local Loopback

Remote line interface unit (LIU) loopback loops the transmit (TX) data and TX clock back to the device as receive (RX) data and RX clock. From the line, LIU loopback

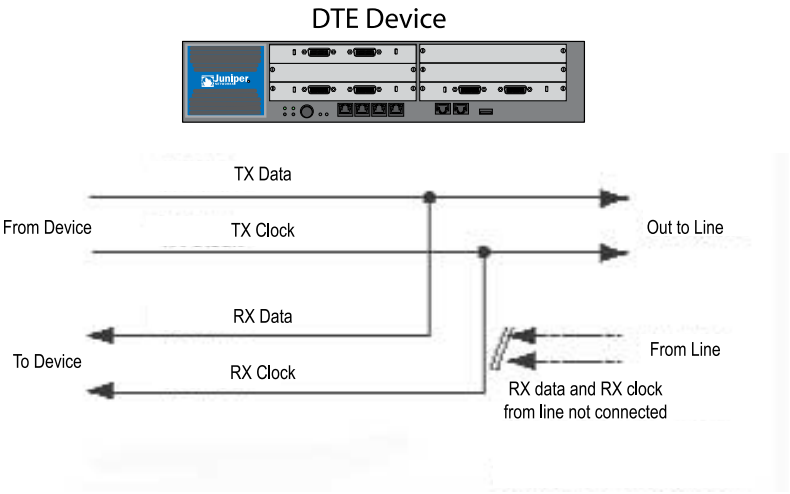
loops the RX data and RX clock back out the line as TX data and TX clock, as shown in Figure 464 on page 1887.

Figure 464: WAN Interface LIU Loopback



DCE local and DCE remote control the TIA/EIA 530 interface-specific signals for enabling local and remote loopback on the link-partner DCE. Figure 465 on page 1887 shows local loopback.

Figure 465: WAN Interface Local Loopback



Loopback Mode

You can configure loopback mode between the local T1, T3, E1, E3, or ISDN (bri) interface and the remote channel service unit (CSU), as shown in Figure 466 on page 1888. You can configure the loopback mode to be local or remote. With local loopback, the interface can transmit packets to the CSU but receives its own transmission back again and ignores data from the CSU. With remote loopback, packets sent from the CSU are received by the interface, forwarded if there is a valid route, and immediately retransmitted to the CSU. Local and remote loopback transmissions loop back both

data and clocking information. Packets can be looped on either the local routing platform or the remote CSU.

To configure loopback mode on a serial interface:

WebUI

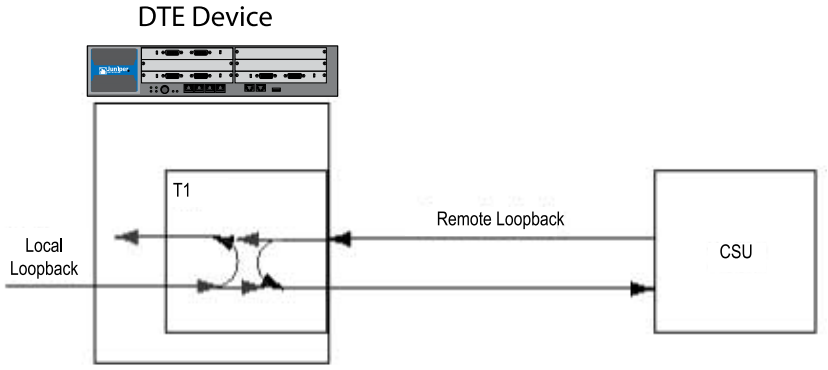
Network > Interfaces > List > Edit (*serial interface*) > WAN: Select the following, then click **Apply**:

Diagnosis Options
Loopback Mode:

CLI

set interface *interface* serial-options loopback { dce-local | local | remote }
save

Figure 466: Remote and Local WAN Interface Loopback Traffic



To configure the loopback mode on E1, E3, or ISDN (bri) interface:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Select the following, then click **Apply**:

Diagnosis Options
Loopback Mode: local or remote

CLI

set interface *interface* e1-options loopback { local | remote }
set interface *interface* e3-options loopback { local | remote }
set interface *interface bri*-options loopback { local | remote }
save

Some WAN interfaces allow you to specify the loopback payload option to loop back data without clocking information on the remote router.

To configure loopback payload on a T1 and T3 interfaces:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Select the following, then click **Apply**:

Diagnosis Options
Loopback Mode: payload

CLI

set interface *interface* t1-options loopback payload
set interface *interface* t3-options loopback payload
save

T3 HDLC payload scrambling provides better link stability. Both sides of a connection must either use or not use scrambling.

To configure scrambling on the T3 or E3 channels on the interface:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN:

T3 Options
Check the Payload Scrambling check box

CLI

set interface *interface* t3-options payload-scrambler
set interface *interface* e3-options payload-scrambler
save

CSU Compatibility Mode

Subrating a T3 or E3 interface reduces the maximum allowable peak rate by limiting the HDLC-encapsulated payload. Subrate modes configure the interface to connect with CSUs that use proprietary methods of multiplexing.

You can configure a T3 interface to be compatible with a Digital Link, Kentrox, Adtran, Verilink, or Larscom CSU. You can configure an E3 interface to be compatible with a Digital Link, or Kentrox CSU.

To configure a T3 or E3 interface to be compatible with the CSU at the remote end of the line:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN:

T3 Options

Select the CSU Compatibility Mode

CLI

```
set interface interface t3-options compatibility-mode { adtran | digital-link | kentrox |
  larscom | verilink } number
set interface interface e3-options compatibility-mode { digital-link | kentrox } number
save
```

The subrate of a T3 or E3 interface must exactly match that of the remote CSU.

Each CSU compatibility mode has different configuration parameters:

- Adtran: You must specify the subrate as a value from 1 through 588 that exactly matches the value configured on the CSU. A subrate value of 588 corresponds to 44.2 Mbps, or 100 percent of the HDLC-encapsulated payload. A subrate value of 1 corresponds to 44.2/588, which is 75.17 Kbps, or 0.17 percent of the HDLC-encapsulated payload.
- Digital Link: You must specify the subrate as the data rate you configured on the CSU in the format **xKb** or **x.xMb**. For Digital Link CSUs, you can specify the subrate value to match the data rate configured on the CSU in the format **xkb** or **x.xMb**. For a list of supported values, enter **?** after the **compatibility-mode digital-link subrate** option.
- Kentrox: You must specify the subrate as a value from 1 through 69 that exactly matches the value configured on the CSU. A subrate value of 69 corresponds to 34.995097 Mbps, or 79.17 percent of the HDLC-encapsulated payload (44.2 Mbps). A subrate value of 1 corresponds to 999.958 Kbps, which is 2.26 percent of the HDLC-encapsulated payload. Each increment of the subrate value corresponds to a rate increment of about 0.5 Mbps.
- Larscom: You must specify the subrate as a value from 1 through 14 that exactly matches the value configured on the CSU. A subrate value of 14 corresponds to 44.2 Mbps, or 100 percent of the HDLC-encapsulated payload. A subrate value of 1 corresponds to 44.2/14, which is 3.16 Mbps, 7.15 percent of the HDLC-encapsulated payload.
- Verilink: You must specify the subrate as a value from 1 through 28 that exactly matches the value configured on the CSU. To calculate the maximum allowable peak rate, multiply the configured subrate by 1.578 Mbps. For example, a subrate value of 28 corresponds to 28 multiplied by 1.578 Mbps, which is 44.2 Mbps, 100 percent of the HDLC-encapsulated payload. A subrate value of 1 corresponds to 1.578 Mbps, 3.57 percent of the HDLC-encapsulated payload. A subrate value of 20 corresponds to 20 multiplied by 1.578 Mbps, which is 31.56 Mbps, 71.42 percent of the HDLC-encapsulated payload.

Remote Loopback Response

The T1 facilities data-link loop-request signal is used to communicate various network information in the form of in-service monitoring and diagnostics. Extended superframe (ESF), through the facilities data link (FDL), supports nonintrusive signaling and control, thereby offering clear-channel communication. Remote loopback requests can be over the FDL or inband. To configure the interface to respond to remote-loopback requests:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Select the following, then click **Apply**:

T1 Options
Remote Loopback Respond: (select)

CLI

set interface *interface* t1-options remote-loopback-respond
save

FEAC Response

The T3 far-end alarm and control (FEAC) signal is used to send alarm or status information from the far-end terminal back to the near-end terminal and to initiate T3 loopbacks at the far-end terminal from the near-end terminal.

To allow the remote Channel Service Unit (CSU) to place the local routing platform into loopback, you must configure the routing platform to respond to the CSU's FEAC request:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Select the following, then click **Apply**:

T3 Options
Remote Loopback Respond: (select)

CLI

set interface *interface* t3-options feac-loop-respond
save

NOTE: If you configure remote or local loopback with the T3 **loopback** option, the routing platform does not respond to FEAC requests from the CSU even if you include the **feac-loop-respond** option in the configuration. In order for the routing platform to respond, you must delete the **loopback** option from the configuration.

Timeslots

Timeslots, also known as channels or connectors, are used with T1 and E1 links and allow users to fraction pin usage or use all of them to a create a single link.

Fractional T1

You can designate any combination of timeslots. For a T1 interface, the time-slot range is from 1 through 24.

NOTE: Use hyphens to configure ranges of timeslots. Use commas to configure discontinuous timeslots. Do not include spaces.

To allocate a specific set of timeslots to a fractional T1 interface:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Enter the following, then click **Apply**:

T1 Options
Timeslots: 1-5,10,24

CLI

```
set interface interface t1-options timeslots 1-5,10,24
save
```

Fractional E1

You can designate any combination of timeslots. ScreenOS reserves slot 1 for framing and cannot be used to configure a fractional E1 interface. For an E1 interface, the time-slot range is 2 through 32.

NOTE: Use hyphens to configure ranges of timeslots. Use commas to configure discontinuous timeslots. Do not include spaces.

To allocate a specific set of timeslots to a fractional E1 interface:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Enter the following, then click **Apply**:

E1 Options
Timeslots: 4-6,11,25

CLI

```
set interface interface e1-options timeslots 4-6,11,25
save
```

Bit Error Rate Testing

Bit error rate testing (BERT) checks the quality of links and allows you to troubleshoot interface errors. You can configure some of the WAN interfaces to execute a BERT when the interface receives a request to run this test.

A BERT requires a line loop to be in place on either the transmission device or the far-end router. The local router generates a known bit pattern then sends it out the transmit path. The received pattern is then verified against the sent pattern. The higher the bit error rate (BER) of the received pattern, the worse the noise is on the physical circuit. As you move the position of the line loop increasingly downstream toward the far-end router, you can isolate the troubled portion of the link.

Before you can start BERT, disable the interface with the **set interface** *interface* **disable** CLI command.

To configure the BERT parameters, perform the following steps:

1. Set the bit pattern or algorithm to send on the transmit path.
2. Set the error rate to monitor when receiving the inbound pattern. You specify this rate in the form of an integer from 0 (the default) through 7, which corresponds to a BER from 10^{-0}(1 error per bit) to 10^{-7}(1 error per 10 million bits).
3. Set the test duration.
4. Save your configuration.
5. Start the BERT.

In this example, you set the T3 interface BERT parameters to run the test for 60 seconds, with the algorithm of pseudo-2t35-o151 and error rate of 10^{-4}:

WebUI

Network > Interfaces > List > Edit (*WAN interface*) > WAN: Enter the following, then click **Apply**:

```
BERT Algorithm: pseudo-2e9-o153 (select)
BERT Error Rate: 4
BERT Test Length: 60
Loopback mode: None (select)
```

CLI

```
set interface serial1/0 t3-options bert-algorithm pseudo-2t35-o151
set interface serial1/0 t3-options bert-error-rate 4
set interface serial1/0 t3-options bert-period 60
save
exec interface serial1/0 bert-test start
```

NOTE: If you want to terminate the test sooner, use the **exec interface** *interface* **bert-test stop** CLI command.

To view the results of the BERT, use the **get counter statistics interface** *interface* **extensive** CLI command.

NOTE: To exchange BERT patterns between a local and a remote routing platform, include the **loopback remote** option in the interface configuration at the remote end of the link. From the local routing platform, issue the **exec interface** *interface* **bert-test start** CLI command.

You can determine whether there is an internal or an external problem by checking the error counters in the output with the **show interface** *interface* **extensive** CLI command.

ISDN Options

The minimum ISDN option you must configure is the ISDN switch type. The other options are determined by your Carrier.

Switch Type

The supported ISDN switch types are ATT5E, NT DMS-100, INS-NET, ETSI, and NI1.

WebUI

Network > Interfaces > List > Edit (*bri*) > **WAN**: Enter the following, then click **Apply**:

Switch Type After Reboot:

CLI

```
set interface bri0/0 isdn switch-type att5e
```

SPID

If you are using an ISP that requires a Service Profile Identifier (SPID), your ISDN device cannot place or receive calls until it sends a valid, assigned SPID to the ISP when it accesses the switch to initialize the connection.

A SPID is usually a seven-digit telephone number with some optional numbers. However, different ISPs may use different numbering schemes. For the DMS-100 switch type, two SPIDs are assigned, one for each B-channel. Your Carrier defines the SPID numbers.

Currently, only the DMS-100 and NI1 switch types require a SPID. The AT&T 5ESS switch type may support a SPID, but we recommend that you set up that ISDN service

without one. In addition, SPIDs have significance at the local access ISDN interface only. Remote routers never receive the SPID.

WebUI

Network > Interfaces > List > Edit (*bri*) > ISDN: Enter the following, then click **Apply**:

SPID1: 123456789
SPID2: 987654321

CLI

```
set interface bri0/0 isdn spid1 123456789
```

TEI Negotiation

Terminal Endpoint Identifier (TEI) negotiation is useful for switches that may deactivate Layer 1 or 2 when there are no active calls. Typically, this setting is used for ISDN service offerings in Europe and connections to DMS-100 switches that are designed to initiate TEI negotiation.

WebUI

Network > Interfaces > List > Edit (*bri*) > ISDN: Enter the following, then click **Apply**:

TEI negotiation: First Call

CLI

```
set interface bri0/0 isdn tei-negotiation first-call
```

You can have TEI negotiation occur when the first call is made (default) or at device power up.

Calling Number

A number for outgoing calls that the ISDN BRI supplies to the ISDN switch. Some networks offer better pricing on calls where the number is presented.

WebUI

Network > Interfaces > List > Edit (*bri*) > ISDN: Enter the following, then click **Apply**:

Calling Number: 1234567890

CLI

```
set interface bri0/0 isdn calling-number 1234567890
```

The calling number must be a string with fewer than 32 characters.

T310 Value

If the security device does not receive an ALERT, a CONNECT, a DISC, or a PROGRESS message after receiving a CALL PROC message, it sends a DISC message out to the network after the T310 timeout value expires.

WebUI

Network > Interfaces > List > Edit (*bri*) > ISDN: Enter the following, then click **Apply**:

T310 Value: 20

CLI

```
set interface bri0/0 isdn t310-value 20
```

You can enter a value between 5 and 100 seconds. The default T310 timeout value is 10 seconds.

Send Complete

In some geographic locations, such as Hong Kong and Taiwan, ISDN switches require that the Sending Complete Information Element be added in the outgoing call-setup message to indicate that the entire number is included. This IE is generally not required in other locations.

WebUI

Network > Interfaces > List > Edit (*bri*) > ISDN: Enter the following, then click **Apply**:

Send Complete: check

CLI

```
set interface bri0/0 isdn sending-complete
```

The default setting does not include the send complete information element.

BRI Mode

The ISDN interface (bri) can be configured for leased line mode or as a dialer.

Leased-Line Mode

The BRI can be configured to support leased-line mode, which eliminates signaling on the D-channel. If the BRI is configured for leased-line mode, it becomes a Layer 3 interface that can only deliver data. If you have the BRI in leased-line mode, you

must also provide the IP address of the security device and include PPP encapsulation in your BRI configuration.

The Q931 dialing is not required to set up a channel. For more information on the Q931 and Q921 protocols, refer to the *ScreenOS CLI Reference Guide: IPv4 Command Descriptions*. All other ISDN options do not function in the leased-line mode.

WebUI

Network > Interfaces > List > Edit (*bri*): Enter the following, then click **Apply**:

BRI Mode
  Leased Line (128kps): check

CLI

```
set interface bri0/0 isdn leased-line 128kbps
```

Dialer Enable

If an ISDN BRI is set to enable dialing, the BRI can act as a dialer interface. The BRI must be configured as a dialer interface before it can provide Dial-on-Demand Routing (DDR). By default, an ISDN BRI is not dialer-enabled.

WebUI

Network > Interfaces > List > Edit (*bri*): Enter the following, then click **Apply**:

BRI Mode
  Dial using BRI: check

CLI

```
set interface bri0/0 isdn dialer-enable
```

When you click apply, the dialer options appear. For more information on configuring the dialer options, see “Dialer Options” on page 1897

Dialer Options

The dialer interface name is formatted as **dialerx**, where x is a number from 0 to 9. If a dialer interface has not yet been created, the dialer interface with the number you specify is created automatically.

The following dialer options can be set using the **set interface dialerx** command:

- Primary/alternate number(s)

  The primary number provides a remote destination for the security device to call. If the primary number is not connected, the alternate number is used. The primary and alternate numbers can be any string length less than 32 characters.

- **Load threshold**

This option provides additional bandwidth on demand. If you set this option and the traffic exceeds the load threshold you specified for one B-channel, then the second B-channel is utilized. The range for the B-channel load threshold is 1 to 100 (in percent). The default is 80 percent.

- **Idle time**

  Use this option to set the amount of time (in seconds) you want the device to wait for traffic before it drops the connection. The idle time can be set for 0 to 60,000 seconds, where a setting of zero (0) means the connection cannot be idle. The default is 180 seconds.

- **Retry times**

  Use this option to set the number of attempts you want the security device to dial the phone number specified. If the call does not connect, the number is redialed (one to six times) the number of attempts specified. The default is three attempts.

- **Interval**

  Use this option to set the dial interval (in seconds) between redial attempts caused by no connection. You can specify 1 to 60 seconds; the default is 30 seconds.

- **Dialer pool**

  Use this option to identify the dialer pool that you want the dialer interface to use. The dialer pool identification can be any string length less than 32 characters.

Use the following command to create a dialer interface:

WebUI

Network > Interfaces > List > New > Dialer IF: Enter the following, then click **Apply**:

Interface Name: dialerx
Primary Number: 16900
Alternate Number: 44440
Load Threshold: 80
Idle Time: 100
Retry times: 3
Interval: 30
Dialer Pool:

CLI

```
set interface dialerx primary-number 16900
set interface dialerx alternative-number 44440
set interface dialerx load-threshold 80
set interface dialerx idle-time 100
set interface dialerx retry 30
```

Disabling a WAN Interface

You can disable a WAN interface using either the WebUI or the CLI.

WebUI

Network > Interfaces > Edit (*interface*) > WAN: Deselect the Enable check box, then click **Apply**.

CLI

```
set interface interface disable
save
```

WAN Interface Encapsulation

After the WAN interface is configured, interface encapsulation can be set. This section describes the following WAN interface encapsulation types:

- Point-to-Point Protocol on page 1899
- Frame Relay on page 1900
- Cisco–High-Level Data Link Control (Cisco–HDLC) on page 1901
- Basic Encapsulation Options on page 1901
- PPP Encapsulation Options on page 1904
- PPP Authentication Protocols on page 1907
- Frame Relay Encapsulation Options on page 1909

Single Interface Encapsulation Option	Encapsulation Type Supported
Unnumbered Interfaces	PPP, FR, and Cisco–HDLC
Protocol MTU	PPP and FR
Static IP	PPP and Cisco–HDLC
Keepalives	PPP, FR, and Cisco–HDLC
Keepalive LMI	FR

Point-to-Point Protocol

Point-to-Point Protocol (PPP) links provide full-duplex, simultaneous, bidirectional operation and are a common solution for easy connection of a wide variety of hosts, bridges, and routers.

PPP encapsulation allows different Network Layer protocols to be multiplexed simultaneously over commonly used physical links. PPP uses High-Level Data Link Control (HDLC) for packet encapsulation. HDLC is a bit-oriented, synchronous, Data Link Layer protocol that specifies a data-encapsulation method on synchronous serial links using frame characters and checksums. PPP encapsulation is defined in RFC 1661, *The Point-to-Point Protocol (PPP)*.

To establish a PPP connection, each end of a PPP link configures the link by exchanging Link Control Protocol (LCP) packets. LCP is used to establish, configure, and test data-link options. These options include encapsulation format options; authentication of the peer on the link; handling of varying limits on sizes of packets, detecting a looped-back link and other common configuration errors; determining when a link is functioning properly or failing; and terminating the link.

PPP allows for authentication during link-establishment to permit or deny connection to a device. This authentication can be performed using either Password Authentication Protocol (PAP) or Challenge-Handshake Authentication Protocol (CHAP), as documented in RFC 1334, *PPP Authentication Protocols*. These authentication protocols are intended for use primarily by hosts and routers that connect to a network server over switched circuits or dial-up lines, but they can also be used with dedicated lines.

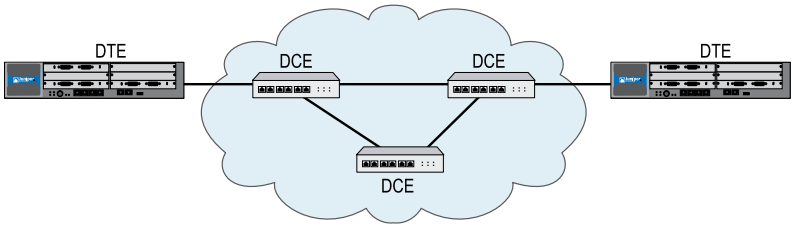
Frame Relay

Frame Relay is a WAN protocol that operates at the Data Link Layer of the Open Systems Interconnection (OSI) Reference Model. Frame Relay encapsulation is defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*, and in the Frame Relay Forum Implementation Agreement FRF3.1/3.2.

The Frame Relay protocol allows you to reduce costs by using shared data-transmission facilities that are managed by a Frame Relay service provider. You pay fixed charges for the local connections from each site in the Frame Relay network to the first point of presence (POP) in which the provider maintains a Frame Relay switch. The portion of the network between Frame Relay switches is shared by all customers of the service provider.

Figure 467 on page 1900 depicts the devices in a Frame Relay network.

Figure 467: Devices in a Frame Relay Network



A Frame Relay network can contain two types of device:

- **Data terminal equipment (DTE)** devices are generally the terminating equipment for a specific network and are typically located on the customer premises.

- **Data circuit-terminating equipment (DCE)** devices are generally carrier-owned devices that provide switching services in a network. DCEs are typically packet switches.

A Frame Relay permanent virtual circuit (PVC) provides a logical connection between two DTE devices across a Frame Relay network. A number of PVCs can be multiplexed into a single physical circuit for transmission across the network. Each PVC is assigned a unique data-link connection identifier (DLCI) to ensure that each customer receives only their own traffic.

You can create traffic shaping for WAN interfaces using Frame Relay protocol. (For more information about configuring traffic shaping for WAN interfaces, see *Fundamentals.)*

Cisco–High-Level Data Link Control (Cisco–HDLC)

The default protocol for serial interfaces on Cisco routers and bridges is Cisco High-Level Data Link control (Cisco–HDLC). Cisco–HDLC is used to encapsulate local area network (LAN) protocol packets for transfer over WAN links.

Cisco–HDLC is an extension to the standard HDLC protocol developed by the International Organization for Standardization (ISO). HDLC is a bit-oriented, synchronous, Data Link Layer protocol that specifies a data-encapsulation method on synchronous serial links using frame characters and checksums.

Cisco–HDLC monitors line status on a serial interface by exchanging keepalive messages with peer network devices. A keepalive message is a signal from one endpoint to the other that the first endpoint is still active. Keepalives are used to identify inactive or failed connections. Keepalives can also allow routers to discover IP addresses of neighbors by exchanging Serial Line Address Resolution Protocol (SLARP) address-request and address-response messages with peer network devices.

Basic Encapsulation Options

To configure encapsulation on a WAN interface:

WebUI

Network > Interfaces > Edit (*WAN interface*): Select the WAN encapsulation type and the security zone, then click **Apply**.

(Optional) Configure encapsulation options for the physical link. This step is required only if you need to change the default HDLC options for a link.

CLI

```
set interface interface encapsulation type
set interface interface zone zone
```

NOTE: You configure the physical link by configuring the interface that represents the link.

Unnumbered Interfaces

An unnumbered interface is not assigned its own IP address but instead borrows an IP address from other interfaces. In this way, address space is conserved. If an unnumbered interface is pointing to an interface that is not functioning (Interface status UP or Protocol UP is not displayed), then the unnumbered interface also does not function. We recommend that unnumbered interfaces point to a loopback interface since loopback interfaces do not fail.

To configure an IP unnumbered interface:

WebUI

Network > Interface > Edit (*WAN interface*): Select the unnumbered option, the source interface, and then click **Apply**.

CLI

```
set interface interface ip unnumbered interface src interface
save
```

Protocol Maximum Transmission Unit Configuration

You can configure the protocol Maximum Transmission Unit (MTU) on each physical interface with PPP or Frame Relay encapsulation. The default protocol MTU is 1500 bytes for serial, T1, T3, E1, E3, and multilink interfaces. You can specify a value from 1280 to 8192 bytes.

The media MTU is derived from the protocol MTU. If you increase the size of the protocol MTU, you must ensure that the size of the media MTU is equal to or greater than the sum of the protocol MTU and the encapsulation overhead.

NOTE: When IPv6 processing is enabled on interfaces that support Frame Relay or Multilink Frame Relay encapsulation, you should specify an MTU value from 1280 to 4080 bytes.

To configure the protocol MTU on a physical interface:

WebUI

Network > Interfaces > Edit (*WAN interface*): Enter a value between 1280 and 8192 in the Maximum Transfer Unit (MTU) field, then click **OK**.

CLI

```
set interface interface mtu number
save
```

Static IP Address Configuration

A WAN interface uses an IP address dynamically assigned by a server at the other end of the WAN data link. Alternatively, you can explicitly assign a static IP address to the interface.

To assign an IP address to the WAN interface:

WebUI

Network > Interfaces > Edit (*WAN interface*): Enter an IP address and netmask in the IP Address/Netmask fields, then click **Apply**.

CLI

```
set interface interface ip ip_addr/mask
save
```

Keepalives

A keepalive message is a signal from one endpoint to the other that the first endpoint is still active. Keepalives are used to identify inactive or failed connections. Physical interfaces configured with WAN encapsulation send keepalive packets at 10-second intervals.

To configure the interface to send keepalive packets at a different interval:

WebUI

Network > Interfaces > Edit (*WAN interface*) > WAN encapsulation type: Enter the number of seconds in **Keepalive Interval**, then click **Apply**.

CLI

```
set interface interface keepalives seconds
save
```

To disable the sending of keepalives on a physical interface:

WebUI

Network > Interfaces > Edit (*WAN interface*) > WAN Encapsulation type: Deselect the Keepalive check box, then click **Apply**.

CLI

```
unset interface interface keepalives
save
```

The receipt of keepalive packets by a destination determines whether the link is down or up. By default, if a destination fails to receive three successive keepalive packets, ScreenOS determines that the link is down. A down link returns to up when the destination receives a single keepalive packet.

To change the counts by which the destination determines a link to be down or up:

WebUI

Network > Interfaces > Edit (*wan interface*) > WAN Encapsulation type: Enter the number of counts for **Down Counter** or **Up Counter**, then click **Apply**.

CLI

```
set interface interface keepalives down-count number
set interface interface keepalives up-count number
save
```

PPP Encapsulation Options

This section explains the Point-to-Point Protocol (PPP) encapsulation options that are available on some WAN interfaces. To configure Multilink Point-to-Point Protocol (MLPPP), see "Multilink PPP Configuration Options" on page 1917.

To configure PPP on a single physical link on an device that supports a WAN interface:

1. Configure PPP encapsulation on the physical link, and assign the link to a security zone. Configure the IP address on the physical link.
2. (Optional) Configure PPP options for the physical link. This step is required only if you need to change the default PPP options for the link.
3. Configure a PPP access profile, and bind it to the interface. This step is required even if no authentication is used on the PPP data link.
4. (Optional) If CHAP or PAP authentication is used, configure the peer's username and password in the local database of the device.

PPP Access Profile

A PPP access profile includes the following information:

- Whether authentication is used to permit or deny connection to devices during Link Control Protocol (LCP) link setup. If authentication is specified, you can configure options for the selected authentication method.

NOTE: Even if no authentication is used on the PPP connection, you must configure an access profile that specifies no authentication method and bind it to the interface.

- Whether the interface uses a static IP address that you have already configured. If the interface uses an IP address dynamically assigned by a server, you can specify the netmask for the IP address.

During LCP link setup, authentication can be used to permit or deny connection to devices; if authentication fails, the PPP link is terminated. By default, authentication is disabled on interfaces that are configured for PPP encryption. If you do not explicitly enable authentication on the interface, the interface makes no authentication requests and denies all incoming authentication challenges.

You can configure interfaces to support one or both of the following authentication protocols:

- Password Authentication Protocol (PAP), as defined in RFC 1334, *PPP Authentication Protocols*
- Challenge Handshake Authentication Protocol (CHAP), as defined in RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*

To configure a WAN interface with PPP encapsulation:

WebUI

Network > PPP > PPP Profile > New: Enter *profile_name* in the PPP Profile field and enter other options, then click **OK**.

Network > Interfaces > Edit (*WAN interface*): Select the *profile_name* in the Binding a PPP Profile drop-down list, then click **Apply**.

CLI

```
set ppp profile profile_name ...
set interface interface ppp profile profile_name
save
```

PPP Authentication Method

During Link Control Protocol (LCP) link setup, you can setup authentication to permit or deny connection to devices; if authentication fails, the PPP link is terminated. To set the PPP authentication method:

WebUI

Network > PPP > PPP Profile > Edit (*profile_name*): Select **Any**, **CHAP**, **PAP**, or **none**, then click **OK**.

CLI

```
set ppp profile profile_name { chap | pap | any | none }
save
```

If you use a static IP address in an access profile, you can only bind the profile to an interface that has an explicitly configured IP address. Conversely, if an interface has a static IP address, the access profile you bind to the interface must specify the **static IP** option.

If the IP address for the interface is dynamically assigned by a server, the netmask for the interface is /32 (255.255.255.255). To specify a different netmask value for the interface:

WebUI

Network > PPP > PPP Profile > Edit (*profile_name*): Enter a new *mask* value in the Netmask field, then click **OK**.

CLI

```
set ppp profile profile_name netmask mask
save
```

Password

The password is used to authenticate the PPP client on the interface with its peer. To set the password:

WebUI

Network > PPP > PPP Profile > Edit (*profile_name*): Enter a string in **Password**, then click **OK**.

CLI

```
set ppp profile profile_name auth secret password
save
```

Network Control Protocol

Using Network Control Protocol (NCP), PPP can negotiate with IPCP, IPv6CP, or both. To set the NCP session:

WebUI

Network > PPP > PPP Profile > Edit (profile_name): Select the required NCP:

- **Try IPCP Only**—Select this option to enable PPP to negotiate with an IPCP session. This option is selected by default.
- **Try IPv6CP Only**—Select this option to enable PPP to negotiate with an IPv6CP session.
- **Try IPCP first, then IPv6CP**—Select this option to enable PPP to negotiate first with an IPCP session, then with an IPv6CP session.
- **Try IPv6CP first, then IPCP**—Select this option to enable PPP to negotiate first with an IPv6CP session, then with an IPCP session.

CLI

```
set ppp profile profile_name ncp ipcp
save
```

PPP Authentication Protocols

This section explains the PPP authentication protocols that are available on some of the WAN interfaces.

Challenge Handshake Authentication Protocol

When Challenge Handshake Authentication Protocol (CHAP) authentication is enabled on an interface, the interface uses the system hostname as the name sent in challenge and response packets. You can configure a different name for the interface to use in challenge and response packets. To change the CHAP local name:

WebUI

Network > PPP > PPP Profile > Edit (*profile_name*): Enter a name in the Local Name field, then click **OK**.

CLI

```
set ppp profile profile_name auth local-name name
save
```

By default, when PPP authentication is enabled on the interface, the interface always challenges its peer and responds to challenges from its peer. You can configure the interface not to challenge its peer and to respond only when challenged (this behavior is called *Passive mode*).

NOTE: Passive mode works only for Challenge Handshake Authentication Protocol (CHAP).

To enable Passive mode:

WebUI

Network > PPP > PPP Profile > Edit (*profile_name*): Select **Passive**, then click **OK**.

CLI

```
set ppp profile profile_name passive
save
```

Password Authentication Protocol

Password Authentication Protocol (PAP) uses a two-way handshake and transmits account names and passwords over the link in cleartext. Systems generally use PAP only when they have no other authentication protocols in common.

To set the authentication protocol to PAP:

WebUI

Network > PPP > Edit (*profile_name*): Select the authentication type, then click **Apply**.

CLI

```
set ppp profile profile_name auth type pap
save
```

Local Database User

If CHAP or PAP is used on the PPP link, the peer device sends its username and password to the device for authentication. The device compares the received username and password with WAN user-type entries configured in its local database. Only a peer whose username and password match an entry in the local database is allowed to connect to the device to send or receive data.

To configure a WAN user:

WebUI

Objects > Users > Local > New: Enter the following, then click **OK**:

```
WAN User: (select)
User Name: name_str
User Password: pswd_str
Confirm Password: pswd_str
```

CLI

```
set user name_str password pswd_str
set user name_str type wan
save
```

NOTE: WAN users can only be configured in a local database.

Frame Relay Encapsulation Options

This section describes how to configure Frame Relay encapsulation options that are available on some WAN interfaces. To configure MLFR, see "Multilink Frame Relay Configuration Options" on page 1918.

NOTE: Make sure that the **Main Link** option is selected in the Basic Properties page for the WAN interface.

Keepalive Messages

Frame Relay keepalive messages are implemented by the sending of Local Management Interface (LMI) packets. ScreenOS sends LMI keepalive messages by default on a Frame Relay interface.

For back-to-back Frame Relay connections, either disable the sending of keepalives on both sides of the connection or configure one side of the connection as DTE (the default ScreenOS configuration) and the other as DCE.

If keepalives are enabled, the number of possible DLCI configurations on a multipoint or multicast connection is limited by the MTU size selected for the interface. To calculate the available DLCIs, use the following formula:

(MTU – 12) / 5

To increase the number of possible DLCIs, disable keepalives. To disable the sending of keepalives on a physical interface, you use the WebUI or the CLI.

WebUI

Network > Interfaces > Edit (*interface*) > FR: Select **No-Keepalive**, then click **Apply**.

CLI

```
set interface interface frame-relay lmi no-keepalive
save
```

Frame Relay LMI Type

By default, ScreenOS sends LMIs specified by ANSI T1.617 Annex D. To change the LMI type to ITU Q933 Annex A:

WebUI

Network > Interfaces > Edit (*WAN interface*) > FR: Select **ITU**, then click **Apply**.

CLI

```
set interface interface frame-relay lmi type itu
save
```

You can configure the following Frame Relay LMI keepalive options:

- **DTE full status polling interval** (denoted by the n391-dte keyword in the CLI). The DTE sends a status inquiry to the DCE at the interval specified by the DTE polling timer. The polling interval specifies the frequency at which these inquiries receive a full status report; for example, a value of 10 would specify a full status report in response to every tenth inquiry. The intermediate inquiries request a keepalive exchange only.
- **DTE error threshold** (denoted by the n392-dte keyword in the CLI). The number of errors required to bring down the link, within the event-count specified by the DTE monitored event-count.
- **DTE monitored event-count** (denoted by the n393-dte keyword in the CLI). The range is from 1 through 10, with a default value of 4.
- **DTE keepalive timer** (denoted by the t391-dte keyword in the CLI). The period at which the DTE sends out a keepalive response request to the DCE and updates status depending on the DTE error-threshold value.

To configure Frame Relay LMI options:

WebUI

Network > Interfaces > Edit (*WAN interface*) > FR: Enter appropriate values for the LMI options, then click **Apply**.

CLI

```
set interface interface frame-relay lmi option
save
```

Creating and Configuring PVCs

Within a single Frame Relay physical interface you can create multiple point-to-point virtual interfaces, which are identified as subinterfaces. Each subinterface maps to a permanent virtual circuit (PVC), which is identified by a data-link connection identifier (DLCI). You can specify only one DLCI for each subinterface. The DLCI is

a value from 16 through 1022. (Numbers 1 through 15 are reserved.) You can choose to multiplex a number of PVCs onto a single physical link for transmission across a Frame Relay packet-switched network. The DLCI value you specify is the DLCI that your local provider has assigned to your PVC.

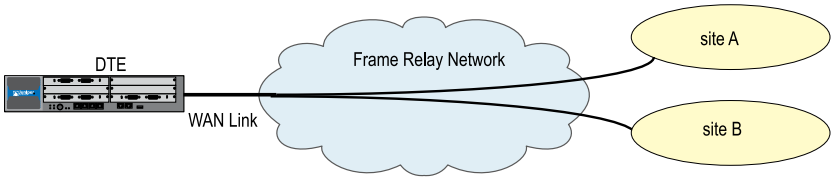
The subinterface name consists of the physical interface name and a subinterface number. For example, if the physical interface name is **serial1/1**, its subinterfaces can be **serial1/1.1** and **serial1/1.2**.

NOTE: In the WebUI, the subinterface number is automatically added when you select the interface name. In the CLI, you must enter both the interface name and the subinterface number.

You can also configure the subinterface for management functions such as a Manage IP address, service options, and other features. (For more information about configuring Manage IP and service options on an interface, see “Fundamentals” on page 15.)

Figure 468 on page 1911 illustrates two point-to-point PVCs configured for the physical interface serial1. You can associate each PVC with a different security zone; the security zone for each PVC can be different from the security zone assigned to the physical interface.

Figure 468: Point-to-Point Frame Relay Subinterfaces



To configure a point-to-point Frame Relay subinterface, create the subinterface and assign it to a security zone, and then assign a Frame Relay DLCI and an IP address to the subinterface:

NOTE: You can assign a subinterface to a different security zone from that assigned to the physical interface.

WebUI

Network > Interface > New > WAN Sub-IF: Enter the following, then click **OK**:

Interface Name: *interface* (select)
Zone Name: (select)
Frame Relay DLCI: (enter *id_num*)
IP Address/Netmask: (enter *ip_addr*)

CLI

set interface *subinterface* zone *zone*
set interface *subinterface* frame-relay dlci *id_num*
set interface *subinterface* ip *ip_addr*
save

Inverse Address Resolution Protocol

Frame Relay subinterfaces can support inverse Address Resolution Protocol (ARP), as described in RFC 2390, *Inverse Address Resolution Protocol*. When inverse ARP is enabled, the device responds to received inverse Frame Relay ARP requests by providing IP address information to the requesting device on the other end of the Frame Relay PVC. The device does not initiate inverse Frame Relay ARP requests.

By default, inverse Frame Relay ARP is disabled. To configure a device to respond to inverse Frame Relay ARP requests:

WebUI

Network > Interface > Edit (*subinterface*): Select **Frame Relay Inverse ARP**, then click **Apply**.

CLI

set interface *subinterface* frame-relay inverse-arp
save

Inverse Neighbor Discovery Protocol

To enable IPv6 support on a Frame Relay interface, you can enable the Inverse Neighbor Discovery Protocol (INDP) setting for the IPv6 interface. With this setting enabled, the interface accepts Inverse Neighbor Discovery (IND) solicitation messages from other IPv6 peer devices. An IND advertisement containing IP address information is sent back to the peer device.

By default, INDP for Frame Relay is disabled. To configure a host to respond to Frame Relay INDP requests:

WebUI

Network > Interfaces > Edit (*for IPv6 interface*): Enter the following, then click OK:

Inverse-nd: (select)
Inverse-nd MTU option: (select)

CLI

set interface *interface* ipv6 inverse-nd
set interface *interface* ipv6 inverse-nd link-mtu
save

Frame Relay interfaces can support INDP, as described in RFC 3122, Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification.

Multilink Encapsulation

This section provides the following information about multilink encapsulation on WAN interfaces:

Overview

WAN interfaces support Multilink Frame Relay (MLFR) and Multilink Point-to-Point Protocol (MLPPP) for User-to-Network Interface (UNI), based on the Frame Relay Forum FRF.16, *Multilink Frame Relay UNI/Network-to-Network Interface (NNI) Implementation Agreement*. Both multilink encapsulation types provide a cost-effective way to increase bandwidth for applications by enabling multiple physical links to be aggregated into a *bundle*. Each physical link in the bundle is referred to as a *bundle link*. For example, if an application requires more bandwidth than is available on a single T1 line, you have two choices to increase bandwidth for the application's use:

- Lease a T3 line
- Bundle multiple T1 links

MLFR and MLPPP also provides fault-tolerance. For example, when a single link in the bundle fails, the bundle continues to support Frame Relay or PPP services by transmitting across the remaining bundle links. MLFR and MLPPP also provide load balancing across the links within a bundle. If a bundle link chosen for transmission is busy transmitting a long packet, another link transmits the data.

You can create traffic shaping for WAN interfaces using MLFR protocol. (For more information about configuring traffic shaping for WAN interfaces, see *"Fundamentals" on page 15*.)

Multilink Encapsulation Options	Encapsulation Type Supported
Minimum Links	MLPPP and MLFR
Fragment Threshold	MLPPP and MLFR
MRRU	MLPPP
Drop Timeout	MLPPP and MLFR
Acknowledge Retries	MLFR
Acknowledge Time	MLFR

Multilink Encapsulation Options	Encapsulation Type Supported
Hello Timer	MLFR

Basic Multilink Bundle Configuration

A bundle is accessed by a multilink interface that you create. The name of the multilink interface must be ml *id_num*. For example, multilink interface names can be **ml1**, **ml2**, and so on.

NOTE: In the WebUI, *id_num* is automatically added when you create a new multilink interface. In the CLI, you must specify an *id_num* value.

By default, the encapsulation type for new multilink interfaces is MLPPP. You must explicitly configure MLFR encapsulation on multilink interfaces that support Frame Relay.

To create a multilink interface and configure it for MLFR encapsulation:

WebUI

Network > Interfaces > New > Multilink IF: Select **Multi-Link Frame Relay** for the WAN encapsulation, then click **Apply**.

CLI

```
set interface interface encapsulation mlfr-uni-nni
save
```

Bundle Identifier

The bundle ID, as specified by FRF.16, associates a local and a remote endpoint with a specific bundle. All bundle links in the ML bundle must use the same bundle ID, which can be up to 80 bytes. If you are configuring more than one bundle between two devices, each bundle ID should be unique. For example, you can use network node identifiers, system serial numbers, or network addresses for bundle IDs.

If you do not configure a specific bundle ID for the multilink interface, the multilink interface name (for example, **ml1** or **ml2**) is used. To configure a bundle ID:

WebUI

Network > Interfaces > Edit (*interface*) > Basic: Enter the ML type for the WAN encapsulation, then click **Apply**.

CLI

```
set interface interface bundle-id name_str
save
```

Drop Timeout

You can configure a drop-timeout value to provide a recovery mechanism if individual links in the multilink bundle drop one or more packets. Drop timeout is not a differential delay-tolerance setting and does not limit the overall latency. We recommend setting a drop-timeout value significantly larger than the expected differential delay across the links; this way, the timeout period elapses when there is actual packet loss and not under normal jitter conditions.

To configure the drop-timeout value:

WebUI

Network > Interfaces > Edit (*interface*) > ML Encapsulation Type: Enter the number of milliseconds in **Drop Timeout**, then click **Apply**.

CLI

```
set interface interface drop-timeout milliseconds
save
```

We do not recommend settings of less than 5 milliseconds; zero (0) disables the timeout.

NOTE: For multilink interfaces, a packet or fragment that encounters an error condition in the network while bound for a disabled bundle or link does not contribute to the dropped packet and framecount in the per-bundle statistics. ScreenOS counts the packet under the global error statistics but not in the global output bytes or output packet counts. This unusual accounting situation happens only if the error conditions are generated inside the multilink interface and not if the packet encounters errors elsewhere in the network.

The minimum-links value can be from 1 to 8. If you specify 8, all configured links of a bundle must be up in order for the bundle to be up.

Fragment Threshold

You can configure a fragmentation threshold to set a maximum size for packet payloads transmitted across the individual links within the multilink circuit. ScreenOS splits any incoming packet that exceeds the fragmentation threshold into smaller units suitable for the circuit size; it reassembles the fragments at the other end but does not affect the output traffic stream.

To configure the fragment threshold:

WebUI

Network > Interfaces > Edit (*interface*) > ML Encapsulation Type: Enter a value for **Bundle-link Fragmentation Threshold**, then click **Apply**.

CLI

```
set interface interface mlfr-uni-nni fragment-threshold number
save
```

By default, the fragment threshold is the MTU of the physical interface. (For serial, T1, and E1 bundle links, the default MTU size is 1500 bytes; for T3 bundle links, the default MTU size is 4470 bytes.) The maximum fragment size can be from 128 through 16,320 bytes. Any value you set must be a multiple of 64 bytes (*N*x64).

NOTE: To ensure proper load-balancing for MLPPP WAN interfaces, do not set both fragment-threshold and short-sequence options in the configuration.

For MLPPP interfaces, if the MTU of links in a bundle is less than the bundle MTU plus encapsulation overhead, then fragmentation is automatically enabled. You should avoid this situation for MLPPP WAN interfaces on which short-sequencing is enabled.

To configure a fragmentation threshold value:

WebUI

Network > Interfaces > Edit (*interface*) > ML Encapsulation Type: Enter a value for **Fragment Threshold**, then click **Apply**.

CLI

```
set interface interface fragment-threshold bytes
save
```

Minimum Links

You can set the minimum number of links that must be up in order for the entire bundle to be up. By default, only one link must be up for the bundle to be considered up.

To set the minimum number of links in a bundle:

WebUI

Network > Interfaces > Edit (*interface*) > ML Encapsulation Type: Enter a new number in **Minimum Links**, then click **Apply**.

CLI

set interface *interface* minimum-links *number*
save

Multilink PPP Configuration Options

This section explains the additional MLPPP configuration options available on some WAN interfaces.

Basic Configuration Steps

To configure MLPPP on an interface that supports MLPPP encapsulation:

1. Create a bundle, and configure it for MLPPP encapsulation. Assign the bundle to a security zone. You can also set other options such as a bundle identification or the MTU for the bundle.
2. (Optional) Configure MLPPP options for the bundle. This step is required only if you need to change the default MLPPP options for the bundle.
3. Configure a PPP access profile, and bind it to the interface. This step is required even if no authentication is used on the PPP data link.
4. (Optional) If CHAP or PAP authentication is used, configure the username and password of the peer in the local database of the security device.
5. Assign bundle links to the bundle.
6. (Optional) Configure PPP options for each bundle link in the bundle. this step is required only if you need to change the default PPP options for the link.

Maximum Received Reconstructed Unit

The maximum received reconstructed unit (MRRU) is similar to a maximum transmission unit (MTU) but applies only to multilink bundles; it is the maximum packet size that the multilink interface can process. By default, the MRRU is set to 1500 bytes; you can configure a different MRRU value if the peer equipment allows it. The MRRU includes the original payload plus the 2-byte PPP header, but it does not include the additional MLPPP header applied while the individual multilink packets are traversing separate links in the bundle.

To configure a different MRRU:

WebUI

Network > Interfaces > Edit (*interface*) > MLPPP: Enter the number of bytes in **Maximum Received Reconstructed Unit**, then click **Apply**.

CLI

set interface *interface* mrru *bytes*
save

Sequence-Header Format

The sequence-header format blank and can be set to 12 or 24 bits, but 24 bits is considered the more robust value for most networks.

To configure a 12-bit sequence header format:

WebUI

Network > Interfaces > Edit (*interface*) > MLPPP: Select **Short-Sequence MLPPP Number**, then click **Apply**.

CLI

```
set interface interface short-sequence
save
```

Multilink Frame Relay Configuration Options

This section explains the additional MLFR configuration options available on some WAN interfaces.

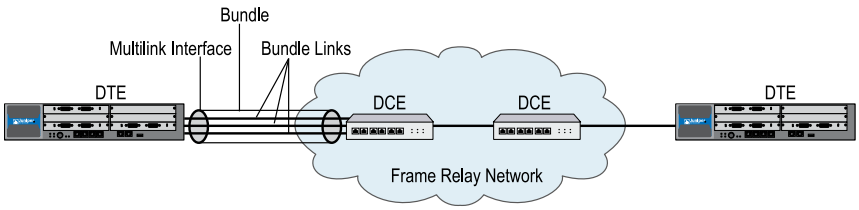
Basic Configuration Steps

To configure MLFR on an interface that supports MLFR encapsulation:

1. Create a bundle, and configure it for MLFR encapsulation. Assign the bundle to a security zone. You can also set other options, such as a bundle identification or the MTU, for the bundle.
2. (Optional) Configure Frame Relay options for the bundle. This step is required only if you need to change the default Frame Relay options for the bundle.
3. Assign bundle links to the bundle.
4. (Optional) Configure MLFR options for each link in the bundle. This step is required only if you need to change the default MLFR options for the link.
5. Create one or more PVCs for the bundle, and assign each PVC a Frame Relay DLCI and an IP address.

Figure 469 on page 1919 shows how MLFR allows multiple T1 bundle links to be aggregated into a single bundle.

Figure 469: Multilink Frame Relay Bundle



Frame Relay functions are configured on the multilink interface and not on each bundle link. (Although bundle links are visible to peer DTE and DCE devices, they are invisible to the Frame Relay Data Link Layer.) The local device and peer devices exchange Link Integrity Protocol (LIP) control messages to determine which bundle links are operational and to synchronize which bundle links are associated with each bundle.

For link management, each end of the bundle link follows the MLFR LIP and exchanges link control messages with its peer at the other end of the bundle link. To bring up a bundle link, both ends of the link must complete an exchange of ADD_LINK and ADD_LINK_ACK messages. To maintain the link, both ends periodically exchange HELLO and HELLO_ACK messages. The exchange of hello messages and acknowledgements serves as a keepalive mechanism for the link. If a device sends a hello message but does not receive an acknowledgement, it resends the hello message up to a configured maximum number of retries. If the device sends the maximum number of retries without receiving an acknowledgement, ScreenOS identifies the bundle link as down.

ScreenOS brings up the bundle link when the peer device acknowledges that it will use the link for the bundle. The link remains up when the peer device acknowledges the hello messages from the local device. When Local Management Interface (LMI) is enabled, the bundle link status is considered up when the Frame Relay Data Link Layer on the local device and on the peer device synchronize using LMI. The bundle link remains up as long as the LMI keepalives are successful.

Link Assignment for MLFR

An MLFR interface can be either DCE or DTE (the default ScreenOS configuration). The DTE acts as a master, requesting status from the DCE part of the link.

For physical links configured for MLFR encapsulation, each link endpoint in a bundle initiates a request for bundle operation with its peer by transmitting an add-link message. A hello message notifies the peer endpoint that the local endpoint is up. Both ends of a link generate a hello message periodically or as configured with the hello timer. A remove-link message notifies the peer that the local end management is removing the link from bundle operation. Endpoints respond to add-link, remove-link, and hello messages by sending acknowledgement messages.

Acknowledge Retries

For bundle links, you can configure the number of retransmission attempts to be made for consecutive hello or remove-link messages after the expiration of the acknowledgement timer. To configure the retransmission attempts:

WebUI

Network > Interfaces > Edit (*interface*) > MLFR: Enter a value for **Line Interface Protocol (LIP) Retransmission Count before Link-Down**, then click **Apply**.

CLI

```
set interface interface mlfr-uni-nni acknowledge-retries number
save
```

Acknowledge Timer

You can configure the maximum period to wait for an add-link acknowledgement, a hello acknowledgement, or a remove-link acknowledgement. To configure the acknowledge time:

WebUI

Network > Interfaces > Edit (*interface*) > MLFR: Enter a value for **Maximum Period to Wait for an Acknowledgement**, then click **Apply**.

CLI

```
set interface interface mlfr-uni-nni acknowledge-timer seconds
save
```

Hello Timer

To configure the rate at which hello messages are sent:

WebUI

Network > Interfaces > Edit (*interface*) > MLFR: Enter a value for **LIP Hello Keepalive Interval**, then click **Apply**.

CLI

```
set interface interface mlfr-uni-nni hello-timer seconds
save
```

A hello message is transmitted after the specified period (in milliseconds) has elapsed. When the hello timer expires, a link endpoint generates an add-link message.

WAN Interface Configuration Examples

This section provides the following WAN configuration examples:

- Configuring a Serial Interface on page 1921
- Configuring a T1 Interface on page 1921

Configuring a Serial Interface

This example configures the WAN properties of a serial interface. Once you have configured the WAN interface properties, see "Encapsulation Configuration Examples" on page 1932 to configure the WAN encapsulation.

WebUI

Network > Interfaces > Edit (serial6/0) > WAN: Select the following, then click **Apply**:

```
Hold Time
   Clock Mode: Internal (select)
   Clock Rate: 8.0 (select)
DTE Options
   Line Encoding: Non-Return-To-Zero (select)
```

CLI

1. **Set the Clocking Information**

```
set interface serial6/0 serial-options clocking-mode internal
set interface serial6/0 serial-options clock-rate 8.0
```

2. **Set the Line Encoding**

```
set interface serial6/0 serial-options encoding nrz
save
```

Configuring a T1 Interface

This example configures the WAN properties of a T1 interface. Once you have configured the WAN interface properties, see "Encapsulation Configuration Examples" on page 1932 to configure the WAN encapsulation.

WebUI

Network > Interfaces > Edit (serial3/0) > WAN: Select the following, then click **Apply**:

Hold Time
Clock Mode: External (select)
T1 Options
Line buildout: 0 - 132 (select)
Line Encoding: 8-bits Zero Suppression (select)
Byte Encoding: 8-bits per byte (select)
Frame Checksum: 16-bits (select)
Framing Mode: Extended Super Frame (select)
Transmitting Flag in Idle Cycles: 0x7E (select)
Start/End Flags on Transmission: Filler (select)
Invert Data: (deselect)

CLI

1. **Set the Clocking Source**

```
set interface serial3/0 clocking external
```

2. **Set the Line Buildout**

```
set interface serial3/0 t1-options buildout 0-132
```

3. **Set the Line Encoding**

```
set interface serial3/0 t1-option line-encoding b8zs
unset interface serial3/0 t1-option invert-data
```

4. **Set the Byte Encoding**

```
set interface serial3/0 t1-option byte-encoding nx56
```

5. **Set the Framing Options**

```
set interface serial3/0 t1-options fcs 16
set interface serial3/0 t1-options framing esf
```

6. **Set the Flag Options**

```
set interface serial3/0 t1-options idle-cycle-flag flags
set interface serial3/0 t1-options start-end-flag filler
save
```

Configuring an E1 Interface

This example configures the WAN properties of an E1 interface. Once you have configured the WAN interface properties, see "Encapsulation Configuration Examples" on page 1932 to configure the WAN encapsulation.

WebUI

Network > Interfaces > Edit (serial6/1) > WAN: Select the following, then click **Apply**:

Hold Time
Clock Mode: External (select)
E1 Options
Frame Checksum: 16-bits (select)
Framing Mode: with CRC4 (select)
Transmitting Flag in Idle Cycles: 0x7E (select)
Start/End Flags on Transmission: Filler (select)
Invert Data: (deselect)
(Optional) Time slots: 2-32

CLI

1. **Set the Clocking Source**

```
set interface serial6/1 clocking external
```

2. **Set the Framing Options**

```
set interface serial6/1 e1-options fcs 16
set interface serial6/1 e1-options framing g704
```

3. **Set the Flags**

```
set interface serial6/1 e1-options idle-cycle-flag flags
set interface serial6/1 e1-options start-end-flag filler
unset interface serial6/1 e1-options invert-data
```

4. **(Optional) Set the Timeslots**

```
set interface serial6/1 e1-options timeslots 2-32
save
```

Configuring a T3 Interface

This example configures the WAN properties of a T3 interface. Once you have configured the WAN interface properties, see "Encapsulation Configuration Examples" on page 1932 to configure the WAN encapsulation.

WebUI

Network > Interfaces > Edit (serial4/0) > WAN: Select the following, then click **Apply**:

Hold Time
Clock Mode: External (select)
T3 Options
Frame Checksum: 16-bits (select)
Transmitting Flag in Idle Cycles: 0x7E (select)
Start/End Flags on Transmission: Filler (select)

CLI

1. **Set the Clocking Source**

```
set interface serial4/0 clocking external
```

2. **Set the Framing Option**

```
set interface serial4/0 t3-options fcs 16
```

3. **Set the Flags**

```
set interface serial4/0 t3-options idle-cycle-flag flags
set interface serial4/0 t3-options start-end-flag filler
save
```

Configuring an E3 Interface

This example configures the WAN properties of an E3 interface. Once you have configured the WAN interface properties, see "Encapsulation Configuration Examples" on page 1932 to configure the WAN encapsulation.

WebUI

Network > Interfaces > Edit (serial4/0) > WAN: Select the following, then click **Apply**:

Hold Time
Clock Mode: External (select)
E3 Options
Frame Checksum: 16-bits (select)
Transmitting Flag in Idle Cycles: 0x7E (select)
Start/End Flags on Transmission: Filler (select)

CLI

1. **Set the Clocking Source**

```
set interface serial4/0 clocking external
```

2. **Set the Framing Option**

```
set interface serial4/0 e3-options fcs 16
```

3. **Set the Flags**

```
set interface serial4/0 e3-options idle-cycle-flag flags
set interface serial4/0 e3-options start-end-flag filler
save
```

Configuring a Device for ISDN Connectivity

The following steps summarize the minimum setup that is required to configure your device for ISDN using the default options:

Configuration Steps	See
1. Select the ISDN switch type.	"Step 1: Selecting the ISDN Switch Type" on page 1925
2. Configure a PPP profile.	"Step 2: Configuring a PPP Profile" on page 1925
3. Set up the ISDN interface (BRI).	"Step 3: Setting Up the ISDN BRI Interface" on page 1926
4. Route traffic through the ISDN BRI.	"Step 4: Routing Traffic to the Destination" on page 1931

Step 1: Selecting the ISDN Switch Type

The minimum ISDN options to configure is to select the ISDN switch type your device is connected to. For more information on other ISDN options, see "ISDN Options" on page 1894.

WebUI

Network > Interfaces > List > Edit (*bri*): Select **WAN** and select the applicable option value, then click Apply:

Switch type after Reboot: etsi

CLI

```
set in bri0/0 isdn switch-type etsi
```

Use the **get int bri2/0 isdn** command to display the ISDN stack configuration.

Step 2: Configuring a PPP Profile

Configure a PPP profile using a static or dynamic IP address.

WebUI

Network > PPP > PPP Profile > New: Enter the applicable option value, then click **OK**:

PPP Profile: isdn-ppp
Authentication: CHAP

Passive: Check
Local Name: 169
Password: 169

CLI

```
set ppp profile isdn-ppp
set ppp profile isdn-ppp auth local-name 169
set ppp profile isdn-ppp auth secret 169
set ppp profile isdn-ppp auth type chap
set ppp profile isdn-ppp passive
```

Step 3: Setting Up the ISDN BRI Interface

This section describes the three methods of configuring the ISDN BRI for ISDN support:

Configuring the ISDN BRI	See
Using the ISDN BRI as a dialer	"Dialing Out to a Single Destination Only" on page 1926
Using the Dialer interface to dial out	"Dialing Out Using the Dialer Interface" on page 1927
Using a leased line mode	"Using Leased-Line Mode" on page 1930

See the respective sections for examples of configuring your device for ISDN support.

Dialing Out to a Single Destination Only

In this example, you configure the ISDN interface (BRI) as the dialer to connect the endpoints at the Branch Office (see Figure 462 on page 1873) to the Corporate Headquarters. Set up this configuration if you are dialing out to a single destination only when you have intermittent traffic between the two sites. The connection drops when there is no traffic.

Branch Office A on the 10.1.1.1 network dials out to Branch Office B on the 10.2.2.2/16 network or to the Corporate Headquarters on the 11.0.0.0/16 network through the bri0/0 interface.

WebUI

Network > Interfaces > List > Edit (*bri*): Select **Basic** and select the applicable option value, then click Apply:

BRI Mode: Dial Using BRI
Dialer Enable Options
  Primary Number: 16900
WAN Encapsulation: PPP

Click **Apply.**

Binding a PPP Profile: isdn-ppp

CLI

```
set in bri0/0 dialer-enable
set in bri0/0 encap ppp
set in bri0/0 ppp profile isdn-ppp
set in bri0/0 primary-number 16900
```

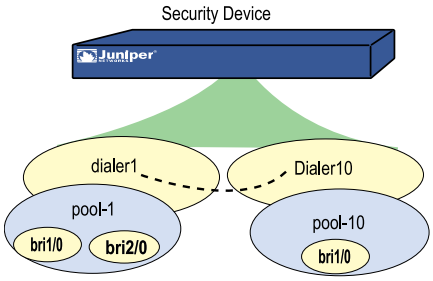
Dialing Out Using the Dialer Interface

In this example, you dial out using the dialer interface. Use this method to dial out to multiple destinations, when the number of destinations exceeds the number of available physical lines. This configuration supports dial-on-demand Routing (DDR) and bandwidth-on-demand.

The dialer pool utilizes ISDN BRI by using logical dial peers via the dialer interfaces. This separates the actual physical links from all the potential destinations. When the number of destinations exceeds the number of available physical lines, a physical interface can be configured as a member of a dialer pool. The physical interface can also belong to more than one pool, allowing the single line to be used to dial more than one destination.

Figure 470 on page 1927 illustrates the relationships among the dialer interface, the dialer pool and the ISDN BRI.

Figure 470: Dialing Out Using the Dialer Interface



By default, ISDN BRIs are not in any dialer pools. Also, each ISDN BRI can be added to several dialer pools.

The following section provides step-by-step instructions on configuring dialer1 and dialer10 as shown in Figure 470 on page 1927 using the WebUI and CLI. At the end of the configuration, two stations at Branch Office A (see Figure 462 on page 1873) can connect to Branch Office B using the dialer interface, dialer1. At the same time, another workstation at Branch Office A can dial out to the headquarters using dialer10.

1. Configure the dialer1 and dialer 10 interfaces.
   a. Create and configure the dialer interfaces.
   b. Bind the PPP profile (isdn-ppp) to the dialer interfaces.
2. Configure the dialer pools, pool-1 and pool-10.
   a. Create the two dialer pools.

b. Bind the dialer pools to the respective dialer interfaces. Bind pool-1 to dialer1 and pool-10 to dialer10.

3. Add the ISDN interface (BRI) to the dialer pool.

   Add bri1/0 and bri2/0 to dialer pool-1.
   Add bri1/0 to dialer pool-10.

To set other BRI options, see "BRI Mode" on page 1896.

WebUI

1. **Configuring the dialer1 Interface**

   Network > Interface > List > New > Dialer IF: Enter the following, then click **OK**:

   Interface Name: dialer1
   Primary Number: 16900
   WAN Encapsulation: Multi-link PPP
   Zone Name: Untrust
   MTU: 1500

   Click **Apply.**

   Binding a PPP Profile: isdn-ppp

   Network > Interfaces > List > Edit (dialer1) > Dialer Pool: Enter the applicable option value, then click Apply:

   Dialer pool: pool-1

   Click **Add**.

   Network > Interfaces > List > Edit (bri1/0):

   Select **Basic** and enter the applicable option value, then click Apply:

   BRI Mode:
   Leased Line Mode(128kbps): uncheck
   Dial Using BRI: uncheck

   Select **Dialer Pool** and enter the applicable option value, then click Apply.

   Set the priority for the pool-1 and check the Select as Member box.

   Priority: 1
   Select as Member: Check

   Network > Interfaces > List > Edit (bri2/0):

   Select **Basic** and enter the applicable option value, then click Apply:

BRI Mode:
Leased Line Mode(128kbps): uncheck
Dial Using BRI: uncheck

Select **Dialer Pool** and enter the applicable option value, then click Apply.

Set the priority for the pool-1 and check the Select as Member box.

Priority: 1
Select as Member: check

2. **Configuring the dialer10 Interface**

Network > Interface > List > New > Dialer IF: Enter the following, then click **OK**:

Interface Name: dialer10
Primary Number: 16900
WAN Encapsulation: Multi-link PPP
Zone Name: Untrust
MTU: 1500

Click **Apply**.

Binding a PPP Profile: isdn-ppp

Network > Interfaces > List > Edit (dialer10) > Dialer Pool: Enter the applicable option value, then click Apply.

Dialer pool: pool-10

Click **Add**.

Network > Interfaces > List > Edit (bri1/0):

Select **Basic** and enter the applicable option value, then click Apply:

BRI Mode:
Leased Line Mode(128kbps): uncheck
Dial Using BRI: uncheck

Select **Dialer Pool** and enter the applicable option value, then click Apply.

Set the priority for the **pool-10** and check the Select as Member box.

Priority: 1
Select as Member: check

CLI

1. **Configuring the dialer1 Interface**

```
set interface dialer1 zone Untrust
set interface dialer1 primary-number 16900
set interface dialer1 encap mlppp
```

```
set interface dialer1 mtu 1500
set interface dialer1 ppp profile isdn-ppp

set dialer pool name pool-1
set interface dialer1 dialer-pool pool-1

set dialer pool pool-1 member-interface bri2/0 priority 1
```

2. **Configuring the dialer10 Interface**

```
set interface dialer10 zone Untrust
set interface dialer10 primary-number 16900
set interface dialer10 encap mlppp
set interface dialer10 mtu 1500
set interface dialer10 ppp profile isdn-ppp

set dialer pool name pool-1
set interface dialer10 dialer-pool pool-10

set dialer pool pool-10 member-interface bri1/0 priority 1
```

As shown in Figure 462 on page 1873, two stations at Branch Office A can connect to Branch Office B using the dialer interface, dialer1. At the same time, another workstation at Branch Office A can dial out to the headquarters using dialer10.

Using Leased-Line Mode

In this example, you establish ISDN connectivity using a leased line. If the BRI is configured for leased-line mode, it becomes a Layer 3 interface that can only deliver data, so the D-channel is not required. Only one channel (B + B) with a total data rate of 128 Kbps is supported. The Q931 dialing is not needed to set up a channel. For more information on the Q931 and Q921 protocols, refer to the *ScreenOS CLI Reference Guide: IPv4 Command Descriptions*.

Use this configuration method to connect two sites with a cost-effective and reliable high-speed connection.

WebUI

Network > Interfaces > List > Edit (*bri*): Select **Basic** and select the applicable option value, then click **OK**:

```
BRI Mode: Leased Line
WAN Encapsulation: PPP
```

Click **Apply**.

```
Binding a PPP Profile: isdn-ppp
```

CLI

```
set in bri0/0 isdn leased-line 128kbps
set in bri0/0 encap ppp
set in bri0/0 ppp profile isdn-ppp
```

Step 4: Routing Traffic to the Destination

Configure the following at the Branch Office A security device to route the traffic through the ISDN interface (BRI) and the dialer interface.

WebUI

Network > Routing > Destination > New: Select the applicable option value, then click **OK**:

```
IP Address/Netmask: 10.2.2.2/16
Next Hop: Gateway
Interface: bri1/0
```

Network > Routing > Destination > New: Select the applicable option value, then click **OK**

```
IP Address/Netmask: 11.0.0.0/16
Next Hop: Gateway
Interface: bri1/0
```

Network > Routing > Destination > New: Select the applicable option value, then click **OK**:

```
IP Address/Netmask: 10.2.2.2/16
Next Hop: Gateway
Interface: bri2/0
```

Network > Routing > Destination > New: Select the applicable option value, then click **OK**:

```
IP Address/Netmask: 11.0.0.0/16
Next Hop: Gateway
Interface: bri2/0
```

CLI

```
set route 10.2.2.2/16 interface bri1/0
set route 10.2.2.2/16 interface bri2/0
set route 11.0.0.0/16 interface bri1/0
set route 11.0.0.0/16 interface bri2/0
```

WebUI

Network > Routing > Destination > New: Select the applicable option value, then click **OK**:

IP Address/Netmask: 10.2.2.2/16
Next Hop: Gateway
Interface: dialer1

Network > Routing > Destination > New: Select the applicable option value, then click **OK**:

IP Address/Netmask: 11.0.0.0/16
Next Hop: Gateway
Interface: dialer1

Network > Routing > Destination > New: Select the applicable option value, then click **OK**:

IP Address/Netmask: 10.2.2.2/16
Next Hop: Gateway
Interface: dialer10

Network > Routing > Destination > New: Select the applicable option value, then click **OK**:

IP Address/Netmask: 11.0.0.0/16
Next Hop: Gateway
Interface: dialer10

CLI

```
set route 10.2.2.2/16 interface dialer1
set route 11.0.0.0/16 interface dialer1

set route 10.2.2.2/16 interface dialer10
set route 11.0.0.0/16 interface dialer10
```

Two stations at Branch Office A can connect to Branch Office B using the dialer interface, dialer1. At the same time, another workstation at Branch Office A can dial out to the headquarters using dialer10.

Encapsulation Configuration Examples

This section provides the following WAN encapsulation examples:

- Configuring PPP Encapsulation on page 1933
- Configuring MLPPP Encapsulation on page 1934
- Configuring Frame Relay Encapsulation on page 1935
- Configuring MLFR Encapsulation on page 1936
- Configuring Cisco HDLC Encapsulation on page 1937
- Configuring IPv6 on WAN Interfaces on page 1939

NOTE: Configure the WAN interface properties before configuring the encapsulation information.

Configuring PPP Encapsulation

This example shows the basic PPP encapsulation configuration.

WebUI

1. **Set the PPP Access Profile**

   Network > PPP > Edit > New: Enter the following, then click **Apply**:

   PPP Profile: juniper1
   Authentication: CHAP (select)
   Static IP: (select)
   Local Name: local-firewall
   Password: abcd1234#B

2. **Set the User Information**

   Objects > User > Local > New: Enter the following, then click **Apply**:

   User name: router
   WAN User: (select)
   Authentication User: (select)
   User Password: abcd1234#C
   Confirm Password: abcd1234#C

3. **Assign the WAN Interface the juniper1 Access Profile**

   Network > Interfaces > List > Edit (serial2/0): Select the following, then click **Apply**:

   WAN Encapsulation: PPP (select)
   Binding a PPP Profile: juniper1 (select)
   Zone Name: untrust (select)
   Fixed IP: (select)
   IP Address/Netmask: 192.168.100.1/24
   Manageable: (select)

CLI

1. **Set the PPP Access Profile**

```
set ppp profile juniper1 auth type chap
set ppp profile juniper1 auth local-name local
set ppp profile juniper1 auth secret abcd1234#B
set ppp profile juniper1 static-ip
```

2. **Set the User Information**

```
set user router password abcd1234#C
set user router type wan
```

3. **Assign the WAN Interface the juniper1 Access Profile**

```
set interface serial2/0 untrust zone
set interface serial2/0 encap ppp
set interface serial2/0 ppp profile juniper1
set interface serial2/0 ip 192.168.100.1/24
set interface serial2/0 manage
save
```

Configuring MLPPP Encapsulation

This example shows the basic MLPPP encapsulation configuration.

WebUI

1. **Set the PPP Access Profile**

   Network > PPP > Edit > New: Enter the following, then click **Apply**:

   PPP Profile: juniper-mlppp
   Authentication: CHAP (select)
   Static IP: (select)
   Local Name: local-firewall
   Password: abcd1234

2. **Set the User Information**

   Objects > User > Local > New: Enter the following, then click **Apply**:

   User name: router
   WAN User: (select)
   Authentication User: (select)
   User Password: abcd1234
   Confirm Password: abcd1234

3. **Set the Multilink Interface**

   Network > Interfaces > List > New Multilink IF: Enter the following, then click **Apply**:

   Interface Name: ML.1
   WAN Encapsulation: Multi-Link PPP (select)
   Zone Name: Untrust (select)

   Edit (ml1 interface): Enter the following, then click **Apply**:

   Binding a PPP Profile: juniper-mlppp (select)
   Fixed IP: (select)
   IP Address/Netmask: 192.168.100.1/24
   Manageable: (deselect)
   Service Options
   Other Services: ping (select)

4. **Set the WAN Interfaces in the Multilink Bundle**

Network > Interfaces > List > Edit (serial1/0): Select Member link options, ml1 Multilink Interface option, then click **Apply**.

Network > Interfaces > List > Edit (serial2/0): Select Member link options, ml1 Multilink Interface option, then click **Apply**.

CLI

1. **Set the PPP Access Profile**

```
set ppp profile juniper-mlppp auth type chap
set ppp profile juniper-mlppp auth local-name local-firewall
set ppp profile juniper-mlppp auth secret abcd1234
set ppp profile juniper-mlppp static-ip
```

2. **Set the User Information**

```
set user router password abcd1234
set user router type wan
```

3. **Set the Multilink Interface**

```
set interface ml1 zone untrust
set interface ml1 encap mlppp
set interface ml1 ppp profile juniper-mlppp
```

4. **Set the WAN Interfaces in the Multilink Bundle**

```
set interface serial1/0 bundle ml1
set interface serial2/0 bundle ml1
set interface ml1 ip 192.168.100.1/24
set interface ml1 manage ping
save
```

Configuring Frame Relay Encapsulation

This example shows the basic Frame Relay encapsulation configuration.

WebUI

1. **Set the Frame Relay Encapsulation**

Network > Interfaces > List > Edit (serial2/0): For WAN encapsulation, select Frame Relay, then click **Apply**:

Edit (serial2/0) > FR: Select the ITU type, then click **Apply**.

2. **Set the PVC**

Network > Interfaces > List > New WAN Sub-IF: Enter the following, then click **Apply**:

Interface Name: serial2/0 (select) .1
Zone Name: Untrust (select)
Frame Relay DLCI: 200
Fixed IP: (select)
IP Address/Netmask: 192.168.100.1/24
Manageable: (deselect)
Service Options
Other Services: ping (select)

NOTE: The DLCI value you specify is the DLCI that your local provider has assigned to your PVC.

CLI

1. **Set the Frame Relay Encapsulation**

```
set interface serial2/0 encap frame-relay
set interface serial2/0 frame-relay lmi type itu
```

2. **Set the PVC**

```
set interface serial2/0.1 zone untrust
set interface serial2/0.1 frame-relay dlci 200
set interface serial2/0.1 ip 192.168.100.1/24
set interface serial2/0.1 manage ping
save
```

Configuring MLFR Encapsulation

This example shows the basic Multilink Frame Relay (MLFR) encapsulation configuration.

WebUI

1. **Create the Multilink Interface**

   Network > Interfaces > List > New Multilink IF: Enter the following, then click **Apply**:

   Interface Name: ML 1
   WAN Encapsulation: Multi-Link FR (select)
   Zone Name: Untrust (select)

   Network > Interfaces > List > Edit (serial2/0): For WAN encapsulation, select Frame Relay, then click **Apply**:

   Edit (serial2/0) > FR: Select the ITU type, then click **Apply**.

2. **Set the WAN Interfaces in a Bundle**

Network > Interfaces > List > Edit (serial1/0): Select Member link options, ml1 Multilink Interface option, then click **Apply**.

Network > Interfaces > List > Edit (serial2/0): Select Member link options, ml1 Multilink Interface option, then click **Apply**.

3. **Set the Bundle PVC**

   Network > Interfaces > List > New WAN Sub-IF: Enter the following, then click **Apply**:

   Interface Name: ml (select).1
   Zone Name: Untrust (select)
   Frame Relay DLCI: 200
   Frame Relay Inverse ARP: (select)
   Fixed IP: (select)
   IP Address/Netmask: 192.168.100.1/24
   Manageable: (deselect)
   Service Options
   Other Services: ping (select)

NOTE: The DLCI value you specify is the DLCI that your local provider has assigned to your PVC.

CLI

1. **Create the Multilink Interface**

   ```
   set interface ml1 zone untrust
   set interface ml1 encap mlfr-uni-nni
   set interface ml1 frame-relay lmi type itu
   ```

2. **Set the WAN Interfaces in a Bundle**

   ```
   set interface serial1/0 bundle ml1
   set interface serial2/0 bundle ml1
   ```

3. **Set the Bundle PVC**

   ```
   set interface ml1.1 zone untrust
   set interface ml1.1 frame-relay dlci 200
   set interface ml1.1 frame-relay inverse-arp
   set interface ml1.1 ip 192.168.100.1/24
   set interface ml1.1 manage ping
   save
   ```

Configuring Cisco HDLC Encapsulation

This example shows the basic Cisco HDLC encapsulation configuration.

WebUI

Device A

Network > Interfaces > List > Edit (serial2/0): Enter the following, then click **Apply**:

WAN Encapsulation: Cisco HDLC (select)
Zone Name: Trust (select)
Fixed IP: (select)
IP Address/Netmask: 192.168.3.1/24

Device B

Network > Interfaces > List > Edit (serial2/0): Enter the following, then click **Apply**:

WAN Encapsulation: Cisco HDLC (select)
Zone Name: Trust (select)
Fixed IP: (select)
IP Address/Netmask: 192.168.3.2/24

CLI

Device A

1. **Bind the WAN Interface to a Security Zone**

   ```
   set interface serial2/0 zone trust
   ```

2. **Set the Encapsulation Type**

   ```
   set interface serial2/0 encap cisco-hdlc
   ```

3. **Set the Interface IP Address**

   ```
   set interface serial2/0 ip 192.168.3.1/24
   save
   ```

Device A

1. **Bind the WAN Interface to a Security Zone**

   ```
   set interface serial2/0 zone trust
   ```

2. **Set the Encapsulation Type**

   ```
   set interface serial2/0 encap cisco-hdlc
   ```

3. **Set the Interface IP Address**

   ```
   set interface serial2/0 ip 192.168.3.2/24
   save
   ```

Configuring IPv6 on WAN Interfaces

The examples provided in this section describe how to configure IPv6 on the following types of WAN interfaces:

- Configuring IPv6 on Point-to-Point Protocol Interface on page 1939
- Configuring IPv6 on a Multilink Point-to-Point Protocol Interface on page 1941
- Configuring IPv6 on a Frame Relay Interface on page 1944
- Configuring IPv6 on a Multilink Frame Relay Interface on page 1945

Configuring IPv6 on Point-to-Point Protocol Interface

This example shows an IPv6 configuration on a Point-to-Point Protocol (PPP) interface.

Figure 471: Enabling IPv6 on a PPP Interface

WebUI (Device A)

1. **Create a PPP Access Profile**

   Network > PPP > PPP Profile > New: Enter the following, then click **Apply**:

   PPP Profile: ppp1
   Authentication: (No Selection)
   Static IP: (select)
   Passive: Don't Challenge Peer: (select)
   Local name: none
   Password: none
   NCP
   Try IPv6CP only: (select)

2. **Set the PPP Encapsulation**

   Network > Interfaces > List > Edit (serial5/0): Enter the following, then click **Apply**:

   WAN Encapsulation
   PPP: (select)

3. **Enable IPv6 on a Serial Interface**

   Network > Interfaces > List > Edit (serial5/0) > IPv6: Select the following, then click **Apply**:

   Enable IPv6: (select)
   Mode: Host (select)
   Interface ID (64-bit HEX): 0000000000000012

> **ND/RA Settings**: Select the following, then click **OK**:

Accept Incoming Router Advertisements: (select)

Network > Interfaces > List > Edit (serial5/0) > Basic: Select the following, then click **Apply**:

Binding a PPP Profile: ppp1 (select)
Fixed IP: (select)
IP Address/Netmask: 0.0.0.0/0

WebUI (Device B)

1. **Create a PPP Access Profile**

   Network > PPP > PPP Profile > New: Enter the following, then click **Apply**:

   PPP Profile: ppp1
   Authentication: (No Selection)
   Static IP: (select)
   Passive: Don't Challenge Peer: (select)
   Local name: none
   Password: none
   NCP
   Try IPv6CP only: (select)

2. **Set the PPP Encapsulation**

   Network > Interfaces > List > Edit (serial3/0): Enter the following, then click **Apply**:

   WAN Encapsulation
     PPP: (select)

3. **Enable IPv6 on a Serial Interface**

   Network > Interfaces > List > Edit (serial3/0) > IPv6: Select the following, then click **Apply**:

   Enable IPv6: (select)
   Mode: Host (select)
   Interface ID (64-bit HEX): 0000000000000016

   > **ND/RA Settings**: Select the following, then click **OK**:

   Accept Incoming Router Advertisements: (select)

   Network > Interfaces > List > Edit (serial3/0) > Basic: Select the following, then click **Apply**:

   Binding a PPP Profile: ppp1 (select)
   Fixed IP: (select)
   IP Address/Netmask: 0.0.0.0/0

CLI

1. **Configuration for Device A**

```
set ppp profile ppp1
set ppp profile ppp1 ncp ipv6cp
set interface serial5/0 zone Untrust
set interface serial5/0 encap ppp
set interface serial5/0 ipv6 mode host
set interface serial5/0 ipv6 enable
set interface serial5/0 ipv6 interface-id 0000000000000012
set interface serial5/0 route
set interface serial5/0 ipv6 ra accept
set interface serial5/0 ipv6 nd nud
set interface serial5/0 ppp profile ppp1
save
```

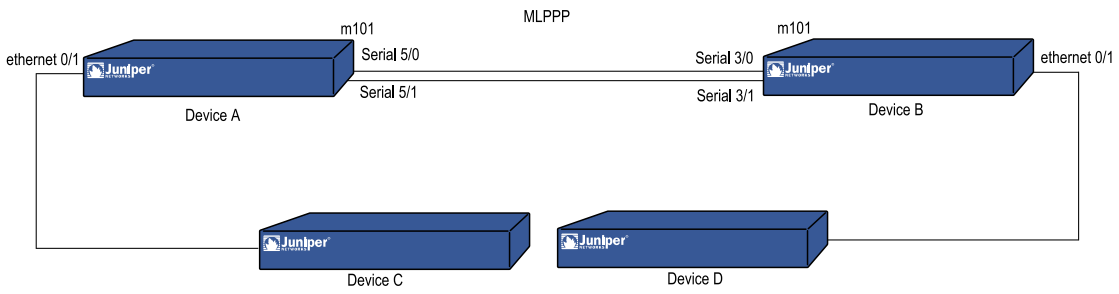
2. **Configuration for Device B**

```
set ppp profile ppp1
set ppp profile ppp1 ncp ipv6cp
set interface serial3/0 zone Untrust
set interface serial3/0 encap ppp
set interface serial3/0 ipv6 mode router
set interface serial3/0 ipv6 ip 2201::24/64
set interface serial3/0 ipv6 enable
set interface serial3/0 ipv6 interface-id 0000000000000016
set interface serial3/0 route
set interface serial3/0 ipv6 ra link-address
set interface serial3/0 ipv6 ra transmit
set interface serial3/0 ipv6 nd nud
set interface serial3/0 ppp profile ppp1
save
```

Configuring IPv6 on a Multilink Point-to-Point Protocol Interface

This example shows an IPv6 configuration on a Multilink Point-to-Point Protocol (MLPPP) interface.

Figure 472: Enabling IPv6 on a MLPPP Interface



WebUI (Device A)

1. **Set the PPP Access Profile**

   Network > Routing > Virtual Router > Edit (trust-vr) > Dynamic Routing Protocol Support > RIP > **Edit RIP Instance**: Enter the following, then click **Apply**:

   Protocol RIP: enable (select)

2. **Create a PPP Access Profile**

   Network > PPP > PPP Profile > New: Enter the following, then click **Apply**:

   PPP Profile: ppp1
   Authentication: (No selection)
   Static IP: (select)
   Passive: Don't Challenge Peer: (select)
   Local name: none
   Password: none
   NCP
   Try IPv6CP only: (select)

3. **Create an MLPPP Interface**

   Network > Interfaces > List > New (Multilink IF): Enter the following, then click **Apply**:

   Interface Name: ml101
   WAN Encapsulation: Multi-Link PPP (select)
   Zone Name: Untrust (select)

   Network > Interfaces > List > Edit (ml101) > IPv6: Enter the following, then click **Apply**:

   Enable IPv6: (select)
   Mode: Host (select)
   Interface ID (64-bit HEX): 0000000000000012

   > **ND/RA Settings**: Select the following, then click **OK**:

   Accept Incoming Router Advertisements: (select)

   Network > Interfaces > List > Edit (ml101) > Basic: Select the following, then click **Apply**:

   Binding a PPP Profile: ppp1 (select)
   Fixed IP: (select)
   IP Address/Netmask: 0.0.0.0/0

   Network > Interfaces > List > Edit (ml101) > RIP: Select the following, then click **Apply**:

   Protocol RIP: Enable (select)

4. **Set the WAN Interfaces in the Multilink Bundle**

   Network > Interfaces > List > Edit (serial5/0): Enter the following, then click **Apply**.

   WAN Configure
   Member Link: (select)
   Multilink Interface: ml101 (select)

   Network > Interfaces > List > Edit (serial5/1): Enter the following, then click **Apply**.

   WAN Configure
   Member Link: (select)
   Multilink Interface: ml101 (select)

Follow the above WebUI procedure to configure MLPPP on the second device Device B.

CLI

1. **Configuration for Device A**

```
set vr trust protocol ripng enable
set ppp profile "ppp1"
set ppp profile "ppp1" ncp ipv6cp
set interface "ml101" zone "Untrust"
set interface "ml101" encap mlppp
set interface "ml101" ipv6 mode "host"
set interface "ml101" ipv6 interface-id 0000000000000012
set interface "ml101" ipv6 enable
set interface ml101 route
set interface ml101 ipv6 ra accept
set interface ml101 ipv6 nd nud
set interface "ml101" ppp profile ppp1
set interface serial5/0 bundle ml101
set interface serial5/1 bundle ml101
set interface ml101 protocol ripng
set interface ml101 protocol ripng enable
save
```

2. **Configuration for Device B**

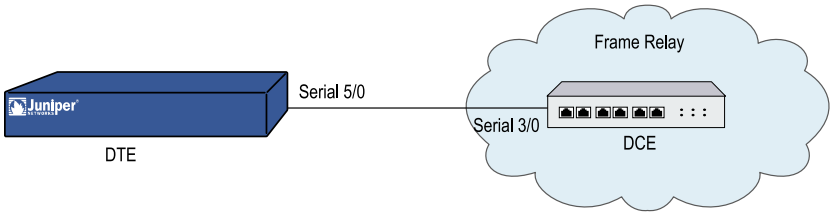
```
set vr trust protocol ripng enable
set ppp profile "ppp1"
set ppp profile "ppp1" ncp ipv6cp
set interface "ml101" zone "Untrust"
set interface "ml101" encap mlppp
set interface "ml101" ipv6 mode "router"
set interface "ml101" ipv6 ip 2501::24/64
set interface "ml101" ipv6 enable
set interface ml101 route
set interface ml101 ipv6 ra link-address
set interface ml101 ipv6 ra transmit
```

```
set interface ml101 ipv6 nd nud
set interface "ml101" ppp profile ppp1
set interface serial3/0 bundle ml101
set interface serial3/1 bundle ml101
set interface ml101 protocol ripng
set interface ml101 protocol ripng enable
save
```

Configuring IPv6 on a Frame Relay Interface

This example shows an IPv6 configuration on a Frame Relay interface. While the first device is a data terminal equipment (DTE), the second device is treated as a FrameRelay data circuit-terminating equipment (DCE).

Figure 473: Enabling IPv6 on a FrameRelay interface



WebUI

1. **Set the Frame Relay Encapsulation**

   Network > Interfaces > List > Edit (serial5/0): Enter the following, then click **Apply**:

   ```
   WAN Encapsulation
     FrameRelay: (select)
   ```

2. **Set a New WAN Subinterface**

   Network > Interfaces > List > New (WAN Sub-IF): Enter the following, then click **Apply**:

   ```
   Interface Name: serial5/0 (select) .1
   Zone Name: Untrust (select)
   Frame Relay DLCI: 51
   Fixed IP: (select)
   IP Address/Netmask: 0.0.0.0/0
   ```

3. **Enable IPv6 on a Serial Interface**

   Network > Interfaces > List > Edit (serial5/0.1) > IPv6: Enter the following, then click **Apply**:

   ```
   Interface Name: serial5/0.1 (read-only)
   Enable IPv6: (select)
   Mode: Router (select)
   ```

Link Local Address (read-only)
Unicast Address 1 / Prefix: 2001:22/64

> **ND/RA Settings**: Select the following, then click **OK**:

RA (Router Advertisement) Configuration
Allow RA Transmission: (select)

CLI

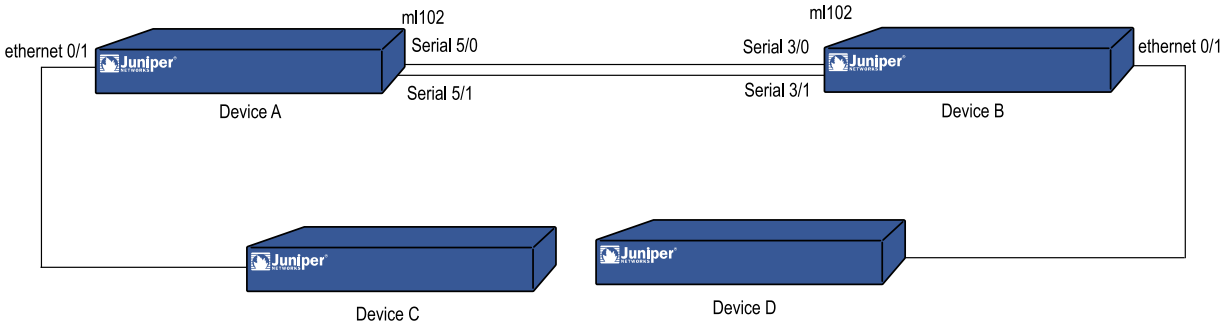
Configuration for DTE

```
set interface "serial5/0" zone "Untrust"
set interface "serial5/0.1" zone "Untrust"
set interface "serial5/0" encap frame-relay
set interface "serial5/0.1" ipv6 mode "router"
set interface "serial5/0.1" ipv6 ip 2001::22/64
set interface "serial5/0.1" ipv6 enable
set interface serial5/0.1 route
set interface serial5/0.1 ipv6 ra link-address
set interface serial5/0.1 ipv6 ra transmit
set interface serial5/0.1 ipv6 nd nud
set interface "serial5/0.1" frame-relay dlci 51
save
```

Configuring IPv6 on a Multilink Frame Relay Interface

This example shows an IPv6 configuration on a Multilink Frame Relay (MLFR) interface.

Figure 474: Enabling IPv6 on an MLFR Interface



WebUI (Device A)

1. **Create a Multilink Interface**

   Network > Interfaces > List > New (Multilink IF): Enter the following, then click **Apply**:

Interface Name: ml102
WAN Encapsulation: Multi-Link Frame Relay (select)
Zone Name: Untrust (select)

2. **Create a Subinterface**

   Network > Interfaces > List > New (WAN Sub-IF): For WAN encapsulation, select Frame Relay, then click **Apply**:

   Interface Name: ml102 (select) .1
   Zone Name: Untrust (select)
   Frame Relay DLCI: 102
   Fixed IP: (select)
   IP Address/Netmask: 0.0.0.0/0

3. **Enable IPv6 on the Subinterface**

   Network > Interfaces > List > Edit (ml102.1) > IPv6: Enter the following, then click **Apply**:

   Interface Name: ml102.1 (read-only)
   Enable IPv6: (select)
   Mode: Host (select)
   Interface ID(64-bit HEX): 0000000000000012
   Link Local Address (read-only)
   Unicast Address 1 / Prefix: 2002::22/64

   > **ND/RA Settings**: Select the following, then click **OK**:

   Accept Incoming Router Advertisements: (select)

4. **Set the WAN Interfaces in a Multilink Bundle**

   Network > Interfaces > List > Edit (Serial5/0): Enter the following, then click **Apply**.

   WAN Configure
   Member Link: (select)
   Multilink Interface: ml102 (select)

   Network > Interfaces > List > Edit (Serial5/1): Enter the following, then click **Apply**.

   WAN Configure
   Member Link: (select)
   Multilink Interface: ml102 (select)

Follow the above WebUI procedure to configure MLPPP on the second device Device B.

CLI

1. **Configuration for Device A**

```
set interface "ml102" zone "Untrust"
set interface "ml102.1" zone "Untrust"
set interface "ml102" encap mlfr-uni-nni
set interface "ml102.1" ipv6 mode "host"
set interface "ml102.1" ipv6 enable
set interface ml102.1 route
set interface ml102.1 ipv6 ra accept
set interface ml102.1 ipv6 nd nud
set interface serial5/0 bundle ml102
set interface serial5/1 bundle ml102
set interface "ml102.1" frame-relay dlci 102
save
```

2. **Configuration for Device B**

```
set interface "ml102" zone "Untrust"
set interface "ml102.1" zone "Untrust"
set interface "ml102" encap mlfr-uni-nni
set interface "ml102.1" ipv6 mode "router"
set interface "ml102.1" ipv6 ip 2002:102::24/64
set interface "ml102.1" ipv6 enable
set interface ml102.1 route
set interface ml102.1 ipv6 ra link-address
set interface ml102.1 ipv6 ra transmit
set interface ml102.1 ipv6 nd nud
set interface serial3/0 bundle ml102
set interface serial3/1 bundle ml102
set interface "ml102" frame-relay dce
set interface "ml102.1" frame-relay dlci 102
save
```

Chapter 59

Digital Subscriber Line

ScreenOS allows you to configure asymmetric digital subscriber line (ADSL) and G.symmetric high-speed digital subscriber line (G.SHDSL) connections with integrated Internet Protocol security virtual private network (IPsec VPN) and firewall services for a broadband telecommuter, a branch office, or a retail outlet. This section describes the available DSL interfaces and provides example configurations.

The data transmission rates available to you depend upon the type of DSL service you obtain from your service provider. Most service providers offer several rate levels, with higher-speed transmissions being more costly than lower-rate transmissions.

NOTE: For information about configuring IPsec VPN and firewall features on the security device, see *"Virtual Private Networks" on page 705*.

This chapter contains the following sections:

Digital Subscriber Line Overview

Traditional telephone lines use analog signals to carry voice service through twisted-pair copper wires. However, analog transmission uses only a small portion of the available bandwidth. Digital transmission allows the service provider to use a wider bandwidth on the same media. The service provider can separate the analog and digital transmissions, using only a small portion of the available bandwidth to transmit voice. This separation allows a telephone and computer to be used simultaneously on the same line. At the central office of the service provider, the DSL access multiplexer (DSLAM) connects many DSL lines to a high-speed network such as an Asynchronous Transfer Mode (ATM) network.

The information that you configure for a DSL interface must match the DSLAM configuration for your DSL connection, so you must obtain the following information from your service provider before configuring the interface:

- Virtual Path Identifier and Virtual Channel Identifier (VPI/VCI), which identifies the VC on the DSLAM.
- ATM encapsulation method. ScreenOS supports the following ATM encapsulations for the DSL interfaces:
  - Virtual circuit (VC)-based multiplexing, in which each protocol is carried over a separate ATM VC.
  - Logical Link Control (LLC), which allows several protocols to be carried on the same ATM VC. This is the default encapsulation method.

NOTE: Check with your service provider for the type of multiplexing used on the DSL connection.

- Point-to-Point Protocol (PPP) is a standard protocol for transmitting IP packets over serial point-to-point links, such as an ATM permanent virtual circuit (PVC). ScreenOS supports the following methods of transporting PPP packets:
  - PPP over Ethernet (PPPoE). RFC 2516 describes the encapsulation of PPP packets over Ethernet. For more information about PPPoE, see"System Parameters" on page 263.
  - PPP over ATM (PPPoA). RFC 1483 describes the encapsulation of network traffic over ATM. For more information about PPPoA, see "Point-to-Point Protocol over ATM" on page 1952 .

NOTE: If the network of the service provider uses PPPoE or PPPoA, the service provider needs to provide the username and password for the connection, the authentication method used, and any other protocol-specific parameters.

- The service provider might give the network a static IP address or a range of IP addresses. The service provider should also provide the address of the Domain Name System (DNS) server to use for DNS name and address resolution.

Asynchronous Transfer Mode

The DSL interfaces use ATM as their Transport Layer. There are two types of ATM virtual circuits (VCs): switched virtual circuits (SVCs) are temporary logical network connections that are created and maintained for individual data transfer sessions, while permanent virtual circuits (PVCs) are continuously available logical connections to the network. The DSL interfaces support multiple PVCs on a single physical line.

To set PVC on your physical line:

WebUI

Network > Interfaces > Edit (adsl or shdsl interface): Enter the following, then click **OK**:

Zone Name: Untrust (selected)
VPI/VCI: 8/35
Multiplexing Method: LLC (selected)
RFC1483 Protocol Mode: Bridged (selected)

CLI

```
set interface interface pvc 8 35 mux llc protocol bridge zone untrust
save
```

ATM Quality of Service

The ATM Quality of Service (QoS) shapes ATM traffic that the user transmits, limiting the rate of transmission. ATM QoS has many benefits:

- Ensures that traffic from one VC does not consume the entire bandwidth of an interface, thus adversely affecting other VCs and leading to data loss.
- Controls bandwidth access when policy dictates that the rate of a given VC on average not exceed a certain rate.
- Match the transmission rate of the local interface to the speed of a remote target interface. For example, suppose one end of a link transmits at 256 Kbps and the other end transmits at 128 Kbps. Without an even, end-to-end pipe, an intermediate switch may have to drop some packets at the lower-speed end, disrupting applications using the link.

Juniper Networks supports three ATM QoS services on the ADSL mini-PIM:

- **Constant Bit Rate (CBR):** A service that is often used when transmitting fixed-rate uncompressed video.
- **Unspecified Bit Rate (UBR):** A service that is often used when transmitting data that can tolerate delays.
- **Variable Bit Rate Non-Real-Time (VBR-NRT):** A service that is often used when transmitting compressed packetized voice and video data, such as video-conferencing.

To set the ATM QoS:

WebUI

Network > Interfaces > Edit (adsl interface): Enter the following, then click **OK**:

Zone Name: Untrust (selected)
VPI/VCI: 8/35
QoS: UBR (selected)
RFC1483 Protocol Method: Bridged (selected)

CLI

```
set interface adsl1/0 qos ubr
save
```

Point-to-Point Protocol over ATM

Point-to-Point Protocol over ATM (PPPoA) is usually used for PPP sessions that are to be terminated on a security device with an DSL interface. PPPoA is primarily used for business class services as it does not require a desktop client.

The following are configuration parameters for a PPPoA client instance:

- Username and password for the PPPoA connection.
- Interface to which the PPPoA instance is bound (the DSL interface or subinterface) and the netmask for the interface (the default is 255.255.255.255).
- Authentication method: Challenge Handshake Authentication Protocol (CHAP), Password Authentication Protocol (PAP), or any authentication protocol (any is the default).
- Auto connect: The number of seconds before a previously closed connection is automatically reinitiated. The default (0) disables this function.
- Clear on disconnect: Specifies that IP information is cleared when a connection is closed. This is disabled by default.
- Idle interval: Specifies the number of minutes that the connection is idle before the security device terminates the connection. The default is 30 minutes.
- PPP Link Control Protocol (LCP) parameters for sending LCP-Echo requests.

When a PPPoA connection is initiated, the PPPoA server automatically provides the IP addresses for the Untrust zone interface and for the DNS servers. When DNS server addresses are received from PPPoA, the device updates the DHCP server on the device with these DNS server addresses. If you do not want the DNS server addresses updated on the DHCP server, you can disable the automatic updating of DNS parameters received through the PPPoA connection.

To display the state of the PPPoA instance, use the WebUI (Network > PPPoA) or the use the **get pppoa all** CLI command. The **get pppoa all** command also shows the state of the physical interface.

The default timeout value for a PPP session on a security device is 1800 seconds (30 minutes). This value is based on the default number of times that an LCP-Echo request is retried (10) multiplied by the interval between each request (180 seconds). You can configure the number of times an LCP-Echo request is retried and the interval between requests.

To set the number of times an LCP-Echo request is retried to 12 and the interval between requests to 190:

WebUI

Network > PPP > PPPoA Profile > Edit (for PPPoA instance): Enter the following, then click **OK**:

PPP Lcp Echo Retries: 12
PPP Lcp Echo Timeout: 190

CLI

```
set pppoa name poa1 ppp lcp-echo-retries 12
set pppoa name poa1 ppp lcp-echo-timeout 190
save
```

Multilink Point-to-Point Protocol

ScreenOS allows you to configure Multilink Point to Point Protocol (MLPPP) over ADSL, which is used to bundle two or more channels into one high-speed connection. This bundle doubles the downstream and upstream bandwidth. When two interfaces are bundled to a multilink (ML) interface while the interfaces are up, Link Control Protocol (LCP) starts. The ML interface does not change its status to up until the LCP negotiation successfully finishes. If the ML interface does not use a static IP address, the ML interface gets a dynamic IP address after the LCP negotiation is finished.

The following restrictions apply for MLPPP to work with ADSL interfaces:

- Two PIMs must connect to the same BRAS
- Only two interfaces can be bundled for MLPPP
- The interface must be in the same security zone as the ML interface

Discrete Multitone for DSL Interfaces

Discrete multitone (DMT) is a method for encoding digital data in an analog signal. By default, the ADSL interface automatically negotiates the DMT operating mode with the DSLAM of the service provider. The mode on the ADSL interface can be changed to cause the interface to use only one of the following DMT standards:

- American National Standards Institute (ANSI) TI.413 Issue 2, which supports data rates up to 8 Mbps downstream and 1 Mbps upstream.
- International Telecommunications Union (ITU) G.992.1 (also known as G.dmt), which supports data rates of 6.144 Mbps downstream and 640 kbps upstream.
- ITU 992.2 (also known as *G.lite*), which supports data rates up to 1.536 Mbps downstream and 512 kbps upstream. This standard is also called *splitterless DSL*, because you do not have to install a signal splitter on your ADSL line; the service provider's equipment remotely splits the signal.
- ITU 992.3 (also known as ADSL2), which supports data rates up to 1.2 Mbps upstream and 12 Mbps downstream.
- ITU 992.5 (also known as ADSL2+), which supports data rates up to 1.2 Mbps upstream and 24 Mbps downstream.

To set the ADSL DMT operating mode to ADSL2+:

WebUI

Network > Interfaces > Edit (adsl interface): Enter the following, then click **OK**:

Operating Mode: ADSL2+ (selected)

CLI

```
set interface adsl1/0 phy operating-mode adsl2plus
save
```

The DMT for the G.SHDSL interface is defined as ITU G.991.2, single-pair High-speed Digital Subscriber Line (SHDSL) Transceiver.

NOTE: Contact your ISP for operational mode compatibility. We recommend using auto mode to determine which operational mode is supported with your connection.

Annex Mode

The annex mode defines the System Reference Model for connecting DSL networks to the plain old telephone service (POTS).

ScreenOS supports the following Annex modes:

- Annex A: Used in North American network implementations.
- Annex B: Used in European network implementations

To configure the ADSL interface to use the annex B mode:

WebUI

Network > Interfaces > Edit (adsl1/0 interface): Enter the following, then click **OK**:

Operating Mode: NON-UR2 (selected)

CLI

```
set interface adsl1/0 phy operating-mode non-ur2
save
```

To configure the G.SHDSL interface to use the annex A mode:

WebUI

Network > Interfaces > Edit (shdsl1/0 interface): Enter the following, then click **OK**:

Operating Mode
Annex: Annex-A (selected)

CLI

```
set interface shdsl1/0 operating-mode annex-a
save
```

Virtual Circuits

To add virtual circuits, you create subinterfaces to the ADSL or G.SHDSL interface. You can create up to 10 subinterfaces. For example, to create a new subinterface named **adsl1/0.1** bound to the predefined zone named **Untrust**:

WebUI

Network > Interfaces > List > New ADSL Sub-IF: Enter the following, then click **Apply**:

```
Interface Name: adsl1/0.1
VPI/VCI: 0/35
Zone Name: Untrust (select)
```

CLI

```
set interface adsl 1/0.1 pvc 0 35 zone Untrust
save
```

You need to configure a subinterface in the same way as the main interface, including setting the VPI/VCI values, as described in "VPI/VCI and Multiplexing Method" on page 1955. You configure a subinterface independently of the main interface; that is, you can configure a different multiplexing method, VPI/VCI, and PPP client on the subinterface than on the main interface. You can also configure a static IP address on a subinterface, even if the main interface does not have a static IP address.

VPI/VCI and Multiplexing Method

Your service provider assigns a VPI/VCI pair for each virtual-circuit connection. For example, you may receive the VPI/VCI pair 1/32, which means a VPI value of 1 and a VCI value of 32. These values must match the values that the service provider has configured on the subscriber's side of the Digital Subscriber Line Access Multiplexer (DSLAM).

To configure the VPI/VCI pair 1/32 on the adsl1/0 interface:

WebUI

Network > Interfaces > List > Edit (for the adsl1/0 interface): Enter **1/32** in the VPI/VCI field, then click **Apply**.

CLI

```
set interface adsl1/0 pvc 1 32
save
```

By default, the device uses Logical Link Control (LLC)-based multiplexing for each virtual circuit.

To configure the VPI/VCI 1/32 on the adslx/0 interface and use LLC encapsulation on the virtual circuit:

WebUI

Network > Interfaces > List > Edit (for the adsl1/0 interface): Enter the following, then click **Apply**:

VPI/VCI: 1 / 32
Multiplexing Method: LLC (selected)

CLI

```
set interface adsl1/0 pvc 1 32 mux llc
save
```

PPPoE or PPPoA

PPPoE is the most common form of ADSL and G.SHDSL encapsulation and is intended for termination on each host on your network. PPPoA is used primarily for business-class service, as PPP sessions can be terminated on the device. To allow the device to connect to the network of the service provider, you need to configure the username and password assigned by the service provider. The configuration for PPPoA is similar to the configuration for PPPoE. PPPoE is described in *"Setting Up PPPoE" on page 290*.

NOTE: The device supports only one PPPoE session on each virtual circuit.

To configure the username **roswell** and password **area51** for PPPoE and bind the PPPoE configuration to the adsl1/0 interface:

WebUI

Network > PPP > PPPoE Profile > New: Enter the following, then click **OK**:

PPPoE Instance: poe1
Bound to Interface: adsl1/0 (select)
Username: roswell
Password: area51

CLI

```
set pppoe name poe1 username roswell password area51
set pppoe name poe1 interface adsl1/0
save
```

There are other PPPoE or PPPoA parameters that you can configure on the device, including method of authentication (by default, the device supports either Challenge Handshake Authentication Protocol or Password Authentication Protocol), idle timeout (default is 30 minutes), and so on. Ask your service provider if there are additional

PPPoE or PPPoA parameters that you need to configure to enable proper communications with the service provider's server.

Static IP Address and Netmask

If your service gave you a specific, fixed IP address and netmask for your network, then configure the IP address and netmask for the network and the IP address of the router port connected to the device. You need to also specify that the device is to use the static IP address. (Typically, the device acts as a PPPoE or PPPoA client and receives an IP address for the ADSL interface through negotiations with the PPPoE or PPPoA server.)

You need to configure a PPPoE or PPPoA instance and bind it to the adsl1/0 interface, as described in "PPPoE or PPPoA" on page 1956. Make sure that you select **Obtain IP using PPPoE** or **Obtain IP using PPPoA** and the name of the PPPoE or PPPoA instance.

To configure the static IP address 1.1.1.1/24 for the network:

WebUI

Network > Interfaces > List > Edit (for the adsl1/0 interface): Enter the following, then click **Apply**:

IP Address/Netmask: 1.1.1.1/24
Static IP: (select)

CLI

```
set interface adsl1/0 ip 1.1.1.1/24
set pppoe name poe1 static-ip
save
```

or

```
set interface adsl1/0 ip 1.1.1.1/24
set pppoa name poa1 static-ip
save
```

ADSL Interface

Asymmetric digital subscriber line (ADSL) is a digital subscriber line (DSL) technology that allows existing telephone lines to carry both voice telephone service and high-speed digital transmission. A growing number of service providers offer ADSL service to home and business customers.

The transmission is *asymmetric* because the rate at which you can send data (the *upstream* rate) is considerably less than the rate at which you can receive data (the *downstream* rate). This is ideal for Internet access because most messages that you send to the Internet are small and do not require much upstream bandwidth, while most data that you receive from the Internet — such as graphic, video, or audio content — requires greater downstream bandwidth.

The ADSL cable provided with some security devices is used to connect the ADSL port on the device to the telephone outlet. No ADSL modem is needed. Signal splitters and microfilters can also be installed once they are obtained from the service provider.

Your network uses the ADSL2/2+ interface **adslx/0**, with x representing the PIM slot, on the device to connect to the service provider's network through an Asynchronous Transfer Mode (ATM) virtual circuit. You can configure additional virtual circuits by creating ADSL2/2+ subinterfaces. For more information, see "Virtual Circuits" on page 1955.

In the WebUI, navigate to the Network > Interfaces > List page to see a list of the current interfaces on the device. If you are using a Telnet or Console session, enter the **get interface** CLI command. You should see that the adslx/0 interface is bound to the Untrust zone. For information about connecting your device to a network using signal splitters and microfilters, refer to the *PIM and Mini-PIM Installation and Configuration Guide*.

If you are using the ADSL2/2+ interface to connect to the service network of the provider, you must configure the adsl(x/0) interface. To do this, you must obtain the following information from your service provider:

- Virtual Path Identifier and Virtual Channel Identifier (VPI/VCI) values
- ATM Adaptation Layer 5 (AAL5) multiplexing method, which can be one of the following:
  - Virtual circuit-based multiplexing, in which each protocol is carried over a separate ATM virtual circuit
  - Logical Link Control (LLC) encapsulation, which allows several protocols to be carried on the same ATM virtual circuit (the default multiplexing method)
- Username and password assigned by the service provider for connection to the service provider's network using either Point-to-Point Protocol over Ethernet (PPPoE) or Point-to-Point Protocol over ATM (PPPoA)
- Authentication method, if any, provided for the PPPoE or PPPoA connection
- Optionally, a static IP address and netmask value for your network

G.SHDSL Interface

The G.symmetric high-speed digital subscriber line (SHDSL) interface supports multi-rate, high-speed, symmetrical digital subscriber line technology for data transfer between a single customer premises equipment (CPE) subscriber and a central office (CO). Unlike ADSL, which was designed for delivering more bandwidth downstream than upstream, SHDSL is symmetrical and delivers a bandwidth of 2.3 Mbps in both directions. The G.SHDSL interface supports ATM-over-SHDSL mode only.

G.SHDSL interfaces can use an ATM interface to send network traffic through a point-to-point connection to a DSLAM. You can configure Point-to-Point Protocol over Ethernet (PPPoE) or PPP over ATM (PPPoE) to connect through DSL connections. PPPoE is described in *"Setting Up PPPoE" on page 290*.

Line-Rate

ScreenOS allows you to specify the available line rates, in kilobytes per second. There are two PIC modes that can be configured on the G.SHDSL interface. You can configure only one mode on each interface:

- 2-port two-wire mode — Supports auto detection of the line rate or fixed line rates and provides network speeds from 192 Kbps to 2.3 Mbps in 64-Kbps increments. Two-wire mode provides two separate, slower SHDSL interfaces.
- 1-port four-wire mode — Supports fixed line rates only and provides network speeds from 384 Kbps to 4.6 Mbps in 128-Kbps increments, doubling the bandwidth. Four-wire mode provides a single, faster SHDSL interface.

To configure the G.SHDSL line-rate:

WebUI

Network > Interfaces > Edit (shdsl): Select the following, then click **OK**:

Operating Mode
Line Rate: auto (selected)

CLI

```
set interface shdsl1/0 phy operating-mode line-rate auto
save
```

Loopback Mode

Loopback testing is a diagnostic procedure in which a signal is transmitted and returned to the sending device after passing through all or a portion of a network or circuit. The returned signal is compared with the transmitted signal in order to evaluate the integrity of the equipment or transmission path.

ScreenOS allows you to specify the type of loopback testing for the G.SHDSL interface. To configure the G.SHDSL interface loopback mode:

WebUI

Network > Interfaces > Edit (shdsl): Select the following, then click **OK**:

Operating Mode
Loopback: local (selected)

CLI

```
set interface shdsl1/0 phy operating mode loopback local
save
```

Operation, Administration, and Maintenance

ScreenOS allows you to set the operation, administration, and maintenance (OAM) F5 loopback cell thresholds, also known as liveness, on ATM virtual circuits. OAM loopback is used to confirm the connectivity of a specific PVC. You can also set the OAM period, which is the interval, in seconds, at which OAM cells are transmitted on ATM virtual circuits.

To set the OAM liveness:

WebUI

Network > Interfaces > Edit (shdsl): Select the following, then click **OK**:

Operating Mode
Oam-Liveness: Down **5** Up **5**

CLI

```
set interface shdsl1/0 phy operating-mode oam-liveness 5
save
```

To set the OAM period:

WebUI

Network > Interfaces > Edit (shdsl): Select the following, then click **OK**:

Operating Mode
Oam-Period: Period **5**

CLI

```
set interface shdsl1/0 phy operating-mode oam-period 5
save
```

Signal-to-Noise Ratio

Signal-to-noise ratio (SNR) measures the quality of a transmission channel or an audio signal over a network channel. Setting the SNR creates a more stable G.SHDSL connection by making the line train at a SNR margin higher than the threshold. If any external noise below the threshold is applied to the line, the line remains stable. You can also disable the SNR margin thresholds. An SNR of zero indicates that the signal is indistinguishable from the unwanted noise.

ScreenOS allows you to configure either or both of the following thresholds:

- **current:** The line trains at higher than current noise margin plus SNR threshold. The range is 0 to 10 db and the default value is **0**.
- **snext:** The line trains at higher than self-near-end crosstalk (SNEXT) threshold. The default is **disabled**.

To configure the SNR threshold:

WebUI

Network > Interfaces > Edit (shdsl): Select the following, then click **OK**:

Operating Mode
Snr-margin: Current **0** Snext **11**

CLI

```
set interface shdsl1/0 phy operating-mode snr-margin current 0
set interface shdsl1/0 phy operating-mode snr-margin snext 11
save
```

ADSL Configuration Examples

This section contains the following configuration examples:

- "Example 1: (Small Business/Home) PPPoA on ADSL Interface" on page 1962. Configure the security device as a firewall with an Internet connection through the ADSL interface with PPPoA (or PPPoE).
- "Example 2: (Small Business/Home) 1483 Bridging on ADSL Interface" on page 1965. Configure the security device as a firewall with an Internet connection through the ADSL interface with 1483 Bridging.
- "Example 3: (Small Business) 1483 Routing on ADSL Interface" on page 1967. Configure the security device as a firewall with an Internet connection through the ADSL interface, using RFC 1483 Routing.
- "Example 4: (Small Business/Home) Dialup Backup" on page 1969. Configure the security device as a firewall with the primary Internet connection through the ADSL interface using PPPoE with dialup as the backup connection.
- "Example 5: (Small Business/Home) Ethernet Backup" on page 1972. Configure the security device as a firewall with the primary Internet connection through the ADSL interface using PPPoE with Ethernet as the backup connection.
- "Example 6: (Small Business/Home) ADSL Backup" on page 1975. Configure the security device as a firewall with the primary Internet connection through the ADSL interface using PPPoE with another ADSL interface as the backup connection.
- "Example 7: (Small Business) MLPPP ADSL" on page 1978. Configure the security device as a firewall with an Internet connection through a MultiLink PPP ADSL connection.
- "Example 8: (Small Business) Allow Access to Local Servers" on page 1981. Configure the security device as a firewall with an Internet connection through the ADSL interface. Allow Internet access to local Web servers while protecting other internal hosts from being directly accessible from the Internet.
- "Example 9: (Branch Office) VPN Tunnel Through ADSL" on page 1983. Configure the security device as a firewall with a VPN tunnel to corporate headquarters

through the ADSL interface. Allow Internet access to local Web servers while protecting other internal hosts from being directly accessible from the Internet.

- "L2TP and L2TP-over-IPsec" on page 939. Configure the security device as a firewall with both an Internet connection and a connection to corporate headquarters through the ADSL interface. Configure a VPN tunnel through the Internet to corporate headquarters as a secondary connection.

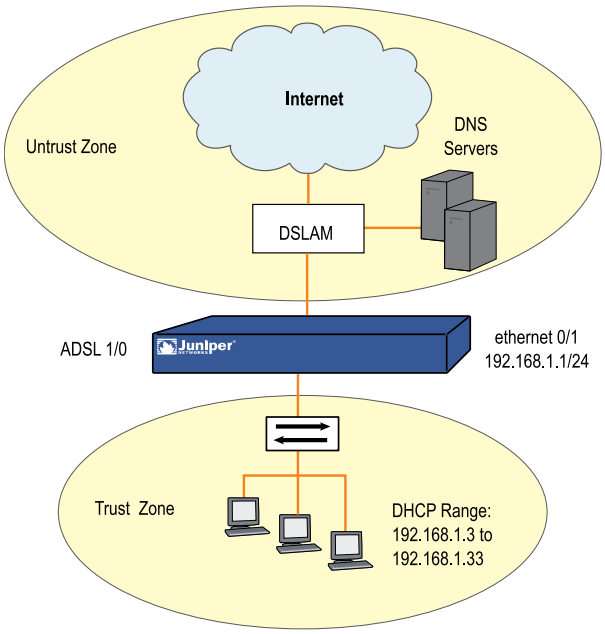
Example 1: (Small Business/Home) PPPoA on ADSL Interface

This example, as shown in Figure 475 on page 1963, explains how to configure a security device as a firewall with an Internet connection through the ADSL interface using PPPoA. Some security devices act as both a PPPoA client and a DHCP server.

PPPoA configuration on an ADSL interface includes the following:

1. Assign the ehternet0/1 interface to the Trust zone and set it as the DHCP Server. When the security device assigns IP addresses to the hosts in the Trust zone, it also provides to the hosts the DNS server address obtained from the service provider.
2. Configure a PVC on the ADSL interface with the VPI/VCI pair value 0/35 that uses LLC encapsulation, and a PPPoA instance named "poa1," which is bound to the ADSL interface. When the security device receives the IP address for the ADSL interface, it also receives one or more IP addresses for DNS servers.
3. Activate PPPoA on the security device. The security device receives a dynamically assigned IP address for its ADSL interface (adsl1/0) from the service provider through PPPoA, and the security device also dynamically assigns IP addresses for the hosts in its Trust zone.
4. Activate DHCP on the internal network.

Figure 475: ADSL Interface Using PPPoA



WebUI

1. **Ethernet Interface and DHCP Server**

   Network > Interfaces > ethernet0/1 > Edit: Enter the following, then click **OK**:

   Zone: Trust
   Static IP: (select)
   IP Address/Netmask: 192.168.1.1/24

   Network > DHCP > Edit (for ethernet2/1) > DHCP Server: Select **Apply**.

   > Addresses > New: Enter the following, then click **OK**:

   Dynamic: (select)
   IP Address Start: 192.168.1.3
   IP Address End: 192.168.1.33

2. **ADSL Interface and PPPoA**

   Network > Interfaces > Edit (for adsl 1/0): Enter the following, then click **OK**:

   VPI/VCI: 0/35
   Encapsulation: LLC (select)
   Zone Name: Untrust

   Network > PPP > PPPoA Profile > New: Enter the following, then click **OK**:

   PPPoA Instance: poa1
   Bound to Interface: adsl1/0 (selected)

Username: alex
Password: tSOCbme4NW5iYPshGxCy67Ww48ngtHC0Bw==
Automatic Update of DHCP Server's DNS Parameters: (select)

3. **Activating PPPoA on the Security Device**

   Turn off the power to the security device and the workstations in the Trust zone.

   Turn on the security device.

   The security device makes a PPPoA connection to the DSLAM, and obtains the IP address for the ADSL interface and the IP addresses for the DNS servers.

4. **Activating DHCP on the Internal Network**

   Turn on the workstations.

   The workstations automatically receive the IP address for the DNS server and obtain an IP address for themselves when they attempt a TCP/IP connection.

CLI

1. **Trust Interface and DHCP Server**

```
set interface ethernet0/1 zone trust
set interface ethernet0/1 ip 192.168.1.1/24
set interface ethernet0/1 dhcp server service
set interface ethernet0/1 dhcp server ip 192.168.1.3 192.168.1.33
```

2. **ADSL Interface and PPPoA**

```
set interface adsl2/1 pvc 0 35 mux llc zone untrust
set pppoa name poa1 username alex password
tSOCbme4NW5iYPshGxCy67Ww48ngtHC0Bw==
set pppoa name poa1 interface adsl1/0
set pppoa name poa1 update-dhcpserver
save
```

3. **Activating PPPoA on the Security Device**

   Turn off the power to the security device and the workstations in the Trust zone.

   Turn on the security device.

   The security device makes a PPPoA connection to the DSLAM, and obtains the IP address for the ADSL interface and the IP addresses for the DNS servers.

4. **Activating DHCP on the Internal Network**

   Turn on the workstations.

   The workstations automatically receive the IP address for the DNS server and obtain an IP address for themselves when they attempt a TCP/IP connection.

Example 2: (Small Business/Home) 1483 Bridging on ADSL Interface

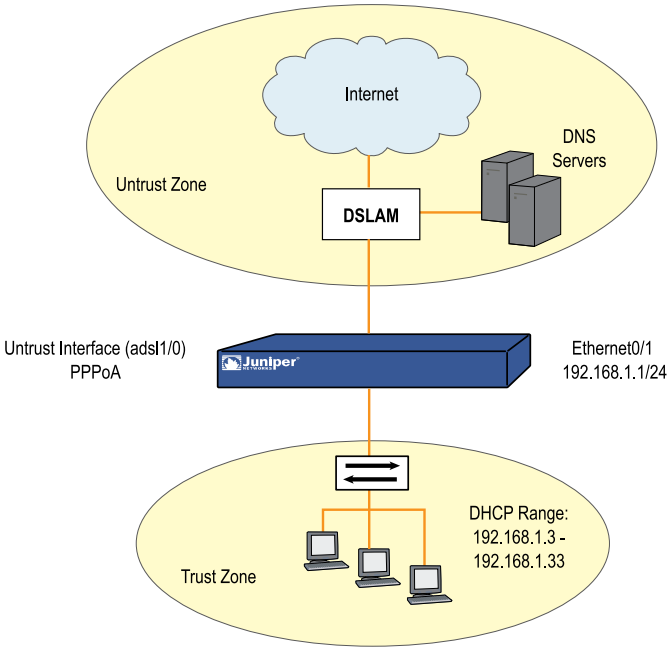
RFC 1483 describes methods of transporting bridged protocol data units (PDUs) over AAL5 links. The bridged PDUs do not require the overhead of IPsec processing, thus allowing more usable bandwidth to be available for data traffic. Such traffic is not secured at the IP Packet l Layer and should only be used where you have a private VC (the service provider assigns you a static IP address for your ADSL interface).

This example, as shown in Figure 476 on page 1965, explains how to configure the security device as a firewall with an Internet connection through the ADSL interface using 1483 bridging. A service provider assigns the static IP address 1.1.1.1/32 for your network, as well as an IP address for the DNS server.

To configure 1483 Bridging on an ADSL interface, do the following:

1. Configure the trust interface and set it as the DHCP server.
2. Configure a PVC on the ADSL interface with the VPI/VCI pair 0/35 and the static IP address 1.1.1.1/32, which is assigned by the service provider.
3. Activate DHCP on the internal network. The security device also dynamically assigns IP addresses for the hosts in its Trust zone. When the security device assigns IP addresses to the hosts in the Trust zone, it also provides the DNS server address from the service provider.

Figure 476: ADSL Interface Using RFC 1483 Bridging



WebUI

1. **Ethernet Interface and DHCP Server**

Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

```
Zone: Trust
Static IP: (select)
IP Address/Netmask: 192.168.1.1/24
Interface Mode: NAT
```

Network > DHCP > Edit (for trust interface) > DHCP Server: Enter the following, then click **Apply**.

```
Gateway: 1.1.1.1
Netmask: 255.255.255.0
DNS#1: 1.1.1.221
```

> Addresses > New: Enter the following, then click **OK**:

```
Dynamic: (select)
IP Address Start: 192.168.1.3
IP Address End: 192.168.1.33
```

2. **ADSL Interface**

   Network > Interfaces > Edit (for adsl1/0 interface): Enter the following, then click **Apply**:

   ```
   VPI/VCI: 0/35
   Zone Name: Untrust
   Static IP: (select)
   IP Address/Netmask: 1.1.1.1/32
   ```

3. **Activating DHCP on the Internal Network**

   Turn off the workstations.

   The workstations automatically receive the IP address for the DNS server and obtain an IP address for themselves when they attempt a TCP/IP connection.

CLI

1. **Trust Interface and DHCP Server**

   ```
   set interface ethernet0/1 zone trust
   set interface ethernet0/1 ip 192.168.1.1/24
   set interface ethernet0/1 dhcp server service
   set interface ethernet0/1 dhcp server ip 192.168.1.3 192.168.1.33
   ```

2. **ADSL Interface**

   ```
   set interface adsl1/0 pvc 0 35 mux llc zone untrust
   set interface adsl1/0 ip 1.1.1.1/32
   save
   ```

3. **Activating DHCP on the Internal Network**

   Turn off the workstations.

The workstations automatically receive the IP address for the DNS server and obtain an IP address for themselves when they attempt a TCP/IP connection.

Example 3: (Small Business) 1483 Routing on ADSL Interface

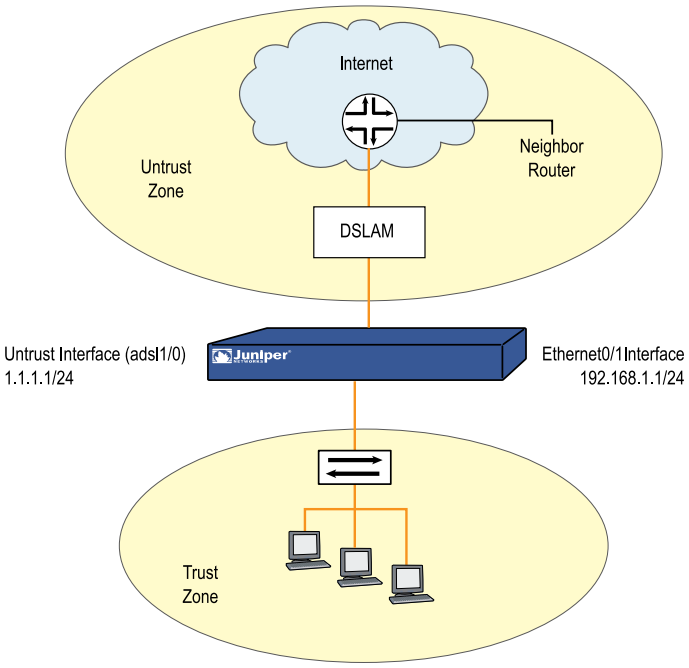
RFC 1483 describes methods of transporting routed protocol data units (PDUs) over AAL5 links. Use this configuration to enable the device to exchange routing information with another router through the ADSL interface.

This example, as shown in Figure 477 on page 1967, explains how to configure the security device as a firewall with an Internet connection through the ADSL interface using 1483 routing and LLC encapsulation.

To configure 1483 routing on an ADSL interface, do the following:

1. Configure the ADSL interface. Set a PVC on the ADSL interface with the VPI/VCI pair 0/35 and the static IP address 1.1.1.1/24. You can also configure the ADSL interface to be a DHCP client which receives its IP address from a DHCP server running on the neighbor router.
2. Configure the Ethernet interface. Set the IP address to 192.168.1.1/24 and the interface mode to route.
3. Enable the dynamic routing protocol—which can be either RIP, OSPF, or BGP—in the trust-vr virtual router and on the ADSL and trust interfaces; in the example, the dynamic routing protocol is RIP. The interface on the neighbor router is also configured for LLC encapsulation and 1483 routing.

Figure 477: 1483 Routing on an ADSL Interface



WebUI

1. **ADSL Interface**

   Network > Interfaces > Edit (for adsl1/0 interface): Enter the following, then click **OK**:

   VPI/VCI: 0/35
   Multiplexing Method: LLC (select)
   RFC1483 Protocol Mode: Routed (select)
   Zone: Untrust
   Static IP: (select)
   IP Address/Netmask: 1.1.1.1/24

2. **Ethernet Interface**

   Network > Interfaces > Edit (for Ethernet0.1): Enter the following, then click **OK**:

   Zone: Trust
   Static IP: (select)
   IP Address/Netmask: 192.168.1.1/24
   Interface Mode: Route

3. **Dynamic Routing Protocol**

   Network > Routing > Virtual Router (trust-vr) > Edit: Select **Create RIP Instance**.

   Select **Enable RIP**, then click **OK**.

   Network > Interface > Edit (for adsl1 interface) > RIP: Select **Protocol RIP Enable**, then click **Apply**.

   Network > Interface > Edit (for trust interface) > RIP: Select **Protocol RIP Enable**, then click **Apply**.

CLI

1. **ADSL Interface**

   ```
   set int adsl1/0 pvc 0 35 mux llc protocol routed zone untrust
   set int adsl1/0 ip 1.1.1.1/24
   ```

2. **Trust Interface**

   ```
   set interface ethernet0/1 zone trust
   set interface ethernet0/1 ip 192.168.1.1/24
   set interface ethernet0/1 route
   ```

3. **Dynamic Routing Protocol**

   ```
   set vr trust-vr protocol rip
   set vr trust-vr protocol rip enable
   set interface adsl1/0 protocol rip
   ```

```
set interface adsl1/0 protocol rip enable
set interface ethernet0/1 protocol rip
set interface ethernet0/1 protocol rip enable
save
```

Example 4: (Small Business/Home) Dialup Backup

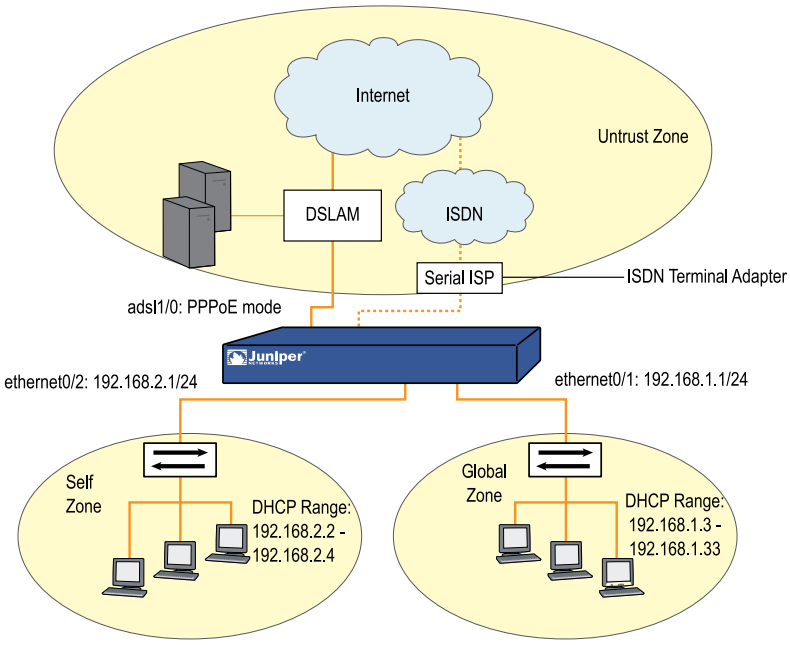
This example, as shown in Figure 478 on page 1970, explains how to configure the security device as a firewall with the primary Internet connection through the ADSL interface using PPPoE, and a backup Internet connection through a dialup connection.

NOTE: Some devices do not support the backup feature.

To configure the primary connection through the ADSL interface and the backup connection through dialup, do the following:

1. Configure the ADSL interface and PPPoE. Configure a PVC on the ADSL interface with the VPI/VCI pair value 0/35 that uses LLC encapsulation, and a PPPoE instance named “poe1,” which is bound to the ADSL interface.
2. Configure a backup connection to the Internet using the serial interface on the Modem port. When the ADSL and serial interface are both bound to the Untrust zone, interface failover is automatically configured. This means that if the ADSL interface becomes unavailable, the security device automatically sends outgoing traffic to the serial interface, dialing through the Integrated Services Digital Network (ISDN) terminal adapter or modem to your ISP account. When the ADSL interface is again available, the security device automatically sends outgoing traffic to the adsl1 interface. For information about interface failover, see *“Interface Failover” on page 1820*. For information about ISDN configuration, see “ISP Failover and Dial Recovery” on page 1995.
3. Configure the Global zone. Set the static IP of the Global zone as 192.168.1.1/24 and set the interface mode to NAT.
4. Configure the Self zone. Set the static IP of the Self zone as 192.168.2.1/24 and set the interface mode to NAT.
5. Activate DHCP on the Self and Global zones.

Figure 478: ADSL with Dialup Backup



To configure the serial interface, you need the following information:

- Login and password for your account to the dialup service provider
- Primary phone connection for dialing into the account
- Modem initialization string

WebUI

1. **ADSL Interface and PPPoE**

   Network > Interfaces > Edit (for adsl1/0 interface): Enter the following, then click **OK**:

   VPI/VCI: 0/35
   Encapsulation: LLC (selected)
   Zone Name: Untrust

   Network > PPP > PPPoE Profile > New: Enter the following, then click **OK**:

   PPPoE Instance: poe1
   Bound to Interface: adsl1/0 (select)
   Username: alex
   Password: tSOCbme4NW5iYPshGxCy67Ww48ngtHC0Bw==
   Automatic Update of DHCP Server's DNS Parameters: (select)

2. **Backup Dialup Interface**

   Network > Interfaces > Backup: Enter the following, then click **OK**:

Primary Interface: adsl1/0
Backup Interface: serial0/0
Type: track ip

Network > Interfaces > Edit > Monitor (for adsl1/0 interface): Enter the following, then click **OK**:

Enable Track IP: (select)
Threshold: 1
Weight: 255

Network > Interfaces > Edit > Monitor > Track IP > Click **ADD**: Enter the following, then click **OK**:

Dynamic: (select)

3. **LAN Interface**

Network > Interfaces > Edit (for ethernet1 interface): Enter the following, then click **OK**:

Zone: Work (already selected)
Static IP: (select)
IP Address/Netmask: 192.168.1.1/24
Interface Mode: NAT

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 192.168.1.3
IP Address End: 192.168.1.33

4. **Self Interface**

Network > Interfaces > Edit (for ethernet0/2 interface): Enter the following, then click **OK**:

Zone: Home (already selected)
Static IP: (select)
IP Address/Netmask: 192.168.2.1/24
Interface Mode: NAT

Network > DHCP > Edit (for ethernet2 interface) > DHCP Server: Select **Apply**.

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 192.168.2.2
IP Address End: 192.168.2.5

5. **Activating DHCP on the Home and Work Zones**

Turn off the workstations.

The workstations automatically receive the IP address for the DNS server and obtain an IP address for themselves when they attempt a TCP/IP connection.

CLI

1. **ADSL Interface and PPPoE**

```
set interface adsl1/0 pvc 0 35 mux llc zone untrust
set pppoe name poe1 username alex password
tSOCbme4NW5iYPshGxCy67Ww48ngtHC0Bw==
set pppoe name poe1 interface adsl1/0
```

2. **Backup Dialup Interface**

```
set interface adsl1/0 backup interface serial0/0 type track-ip
set interface adsl1/0 monitor track ip
set interface adsl1/0 monitor track-ip dynamic
```

3. **LAN Interface**

```
set interface ethernet0/1 ip 192.168.1.1/24
set interface ethernet0/1 dhcp server service
set interface ethernet0/1 dhcp server ip 192.168.1.3 192.168.1.33
```

4. **Self Interface**

```
set interface ethernet0/2 ip 192.168.2.1/24
set interface ethernet0/2 dhcp server service
set interface ethernet0/2 dhcp server ip 192.168.2.2 192.168.2.5
save
```

5. **Activating DHCP on the Home and Work Zones**

   Turn off the workstations.

   The workstations automatically receive the IP address for the DNS server and obtain an IP address for themselves when they attempt a TCP/IP connection.

Example 5: (Small Business/Home) Ethernet Backup

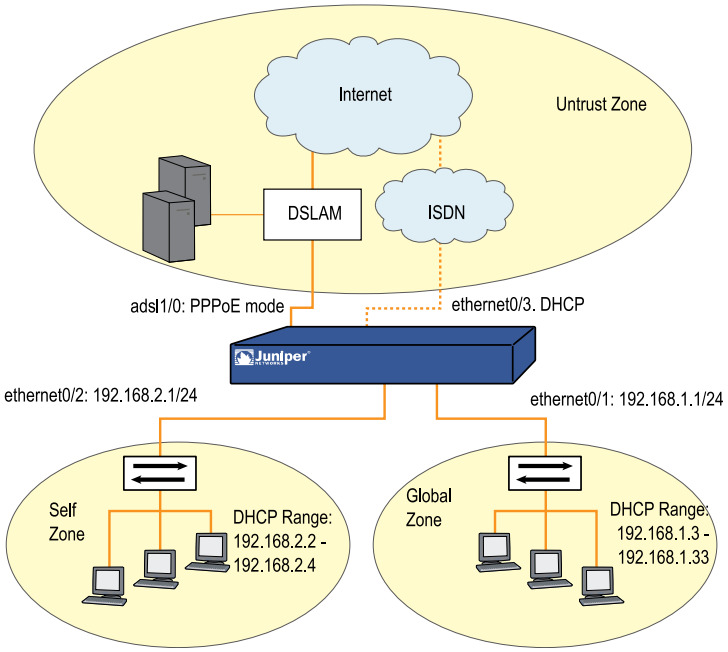
This example, as shown in Figure 479 on page 1973, explains how to configure the security device as a firewall with the primary Internet connection through the ADSL interface using PPPoE, and a backup Internet connection through an Ethernet connection.

NOTE: This example is similar to the configuration shown in "Example 4: (Small Business/Home) Dialup Backup" on page 1969, except that the backup connection to the Internet is through the Untrusted Ethernet port.

To configure the primary connection through the ADSL interface and the backup connection through an Ethernet connection, do the following:

1. Configure the ADSL interface with PPPoE. Configure a PVC on the ADSL interface with the VPI/VCI pair value 0/35 that uses LLC encapsulation, and a PPPoE instance named “poe1,” which is bound to the ADSL interface.
2. Configure the backup interface as ethernet3.
3. Configure the Global zone.
4. Configure the Self zone.

Figure 479: ADSL with Ethernet Backup



WebUI

1. **ADSL Interface and PPPoE**

   Network > Interfaces > Edit (for adsl1/0): Enter the following, then click **OK**:

   VPI/VCI: 0/35
   Encapsulation: LLC (selected)
   Zone: Untrust

   Network > PPP > PPPoE Profile > New: Enter the following, then click **OK**:

   PPPoE Instance: poe1
   Bound to Interface: adsl1/0 (select)
   Username: alex
   Password: tSOCbme4NW5iYPshGxCy67Ww48ngtHC0Bw==
   Automatic Update of DHCP Server’s DNS Parameters: (select)

2. **Backup Ethernet Interface**

Network > Interfaces > Edit (for ethernet0/3): Enter the following, then click **OK**:

Zone Name: Untrust (select)
Obtain IP using DHCP: (select)
Automatic update DHCP server parameters: (select)

3. **Global Zone**

Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

Zone: Global
Static IP: (select)
IP Address/Netmask: 192.168.1.1/24
Interface Mode: NAT

Network > DHCP > Edit (for ethernet0/1) > DHCP Server: Select **Apply**.

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 192.168.1.3
IP Address End: 192.168.1.33

4. **Self Zone**

Network > Interfaces > Edit (for ethernet0/2): Enter the following, then click **OK**:

Zone: Self
Static IP: (select)
IP Address/Netmask: 192.168.2.1/24
Interface Mode: NAT

Network > DHCP > Edit (for ethernet 0/2) > DHCP Server: Select **Apply**.

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 192.168.2.2
IP Address End: 192.168.2.5

CLI

1. **ADSL Interface and PPPoE**

```
set interface adsl1/0 pvc 0 35 mux llc zone untrust
set pppoe name poe1 username alex password
tSOCbme4NW5iYPshGxCy67Ww48ngtHC0Bw==
set pppoe name poe1 interface adsl1/0
```

2. **Backup Ethernet Interface**

```
set interface ethernet0/3 zone untrust
set interface ethernet0/3 dhcp client
set interface ethernet0/3 update-dhcpserver
```

3. **Global Zone**

```
set interface ethernet0/1 ip 192.168.1.1/24
set interface ethernet0/1 dhcp server service
set interface ethernet0/1 dhcp server ip 192.168.1.3 192.168.1.33
```

4. **Self Zone**

```
set interface ethernet0/2 ip 192.168.2.1/24
set interface ethernet0/2 dhcp server service
set interface ethernet0/2 dhcp server ip 192.168.2.2 192.168.2.5
save
```

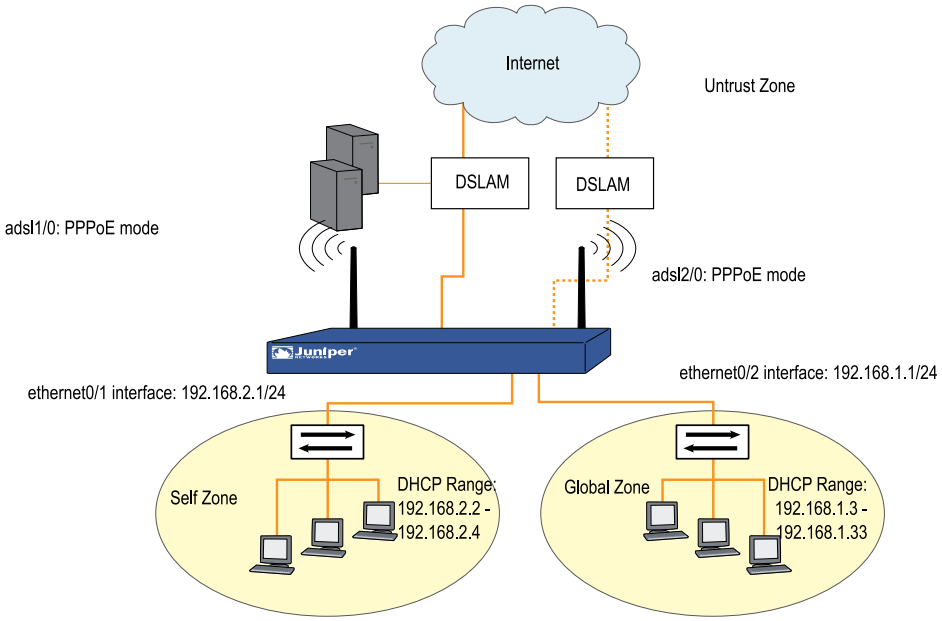
Example 6: (Small Business/Home) ADSL Backup

This example, as shown in Figure 480 on page 1976, explains how to configure the security device as a firewall with the primary Internet connection through the ADSL interface using PPPoE, and a backup Internet connection through a second ADSL connection.

To configure the primary connection through the ADSL interface and the backup connection through an Ethernet connection, do the following:

1. Configure the ADSL interfaces with PPPoE. Configure a PVC on the ADSL interface with the VPI/VCI pair value 0/35 that uses LLC encapsulation, and a PPPoE instance named “poe1,” which is bound to the ADSL interface.
2. Configure the backup ADSL interface as adsl2/0.
3. Configure the Global zone.
4. Configure the Self zone.

Figure 480: ADSL with ADSL Backup



WebUI

1. **ADSL Interface and PPPoE**

   Network > Interfaces > Edit (for adsl1/0): Enter the following, then click **OK**:

   VPI/VCI: 0/35
   Encapsulation: LLC (selected)
   Zone: Untrust

   Network > PPP > PPPoE Profile > New: Enter the following, then click **OK**:

   PPPoE Instance: poe1
   Bound to Interface: adsl1/0 (select)
   Username: alex
   Password: tSOCbme4NW5iYPshGxCy67Ww48ngtHC0Bw==
   Automatic Update of DHCP Server's DNS Parameters: (select)

2. **Backup ADSL Interface**

   Network > Interfaces > Edit (for adsl2/0): Enter the following, then click **OK**:

   VPI/VCI: 8/35
   Encapsulation: LLC (selected)
   Zone: Untrust

   Network > PPP > PPPoE Profile > New: Enter the following, then click **OK**:

PPPoE Instance: poe2
Bound to Interface: adsl2/0 (select)
Username: alex
Password: tSOCbme4NW5iYPshGxCy67Ww48ngtHC0Bw==
Automatic Update of DHCP Server's DNS Parameters: (select)

Network > Interfaces > Backup: Enter the following, then click **OK**:

Primary Interface: adsl1/0
Backup Interface: adsl2/0
Type: track ip

Network > Interfaces > Edit > Monitor (for adsl1/0 interface): Enter the following, then click **OK**:

Enable Track IP: (select)
Threshold: 1
Weight: 255

3. **Global Zone**

Network > Interfaces > Edit (for ethernet0/2): Enter the following, then click **OK**:

Zone: Global
Static IP: (select)
IP Address/Netmask: 192.168.1.1/24
Interface Mode: NAT

Network > DHCP > Edit (for ethernet0/2) > DHCP Server: Select **Apply**.

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 192.168.1.3
IP Address End: 192.168.1.33

4. **Self Zone**

Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

Zone: Self
Static IP: (select)
IP Address/Netmask: 192.168.2.1/24
Interface Mode: NAT

Network > DHCP > Edit (for ethernet0/1) > DHCP Server: Select **Apply**.

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 192.168.2.2
IP Address End: 192.168.2.5

CLI

1. **ADSL Interface and PPPoE**

```
set interface adsl1/0 pvc 0 35 mux llc zone untrust
set pppoe name poe1 username alex password
tSOCbme4NW5iYPshGxCy67Ww48ngtHC0Bw==
set pppoe name poe1 interface adsl1/0
```

2. **Backup ADSL Interface**

```
set interface adsl2/0 pvc 8 35 mux llc zone untrust
set pppoe name poe1 username alex password
tSOCbme4NW5iYPshGxCy67Ww48ngtHC0Bw==
set pppoe name poe2 interface adsl2/0
set interface adsl1/0 backup interface adsl2/0 type track-ip
set interface adsl1/0 monitor track ip
set interface adsl1/0 monitor track-ip dynamic
```

3. **Global Zone**

```
set interface ethernet0/2 ip 192.168.1.1/24
set interface ethernet0/2 dhcp server service
set interface ethernet0/2 dhcp server ip 192.168.1.3 192.168.1.33
```

4. **Self Zone**

```
set interface ethernet0/1 ip 192.168.2.1/24
set interface ethernet0/1 dhcp server service
set interface ethernet0/1 dhcp server ip 192.168.2.2 192.168.2.5
save
```

Example 7: (Small Business) MLPPP ADSL

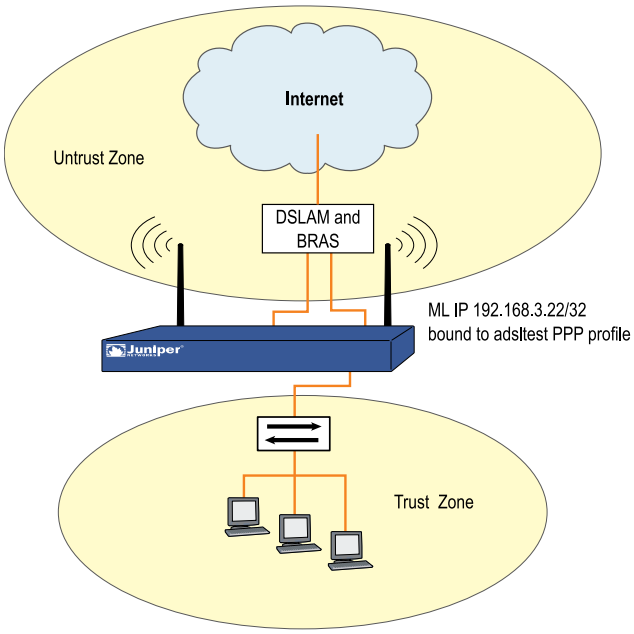
This example, as shown in Figure 478 on page 1970, explains how to configure a PVC on the ADSL interface with the VPI/VCI pair value 8/35 that uses MLPPP encapsulation, and a PPP profile named “adsltest,” which is bound to the ML interface.

NOTE: Some devices do not support MLPP encapsulation.

To configure MLPPP over ADSL, do the following:

1. Configure the multilink interface.
2. Configure a PPP profile with a dynamic IP address.
3. Bind the PPP profile to the multilink interface.
4. Bundle the ADSL interface to the multilink interface.

Figure 481: MLPPP over ADSL



WebUI

1. **Multilink Interface**

   Network > Interfaces > New (multilink interface): Enter the following, then click **OK**:

   Interface Name: 1
   WAN Encapsulation: Multi-Link PPP (select)
   Zone Name: Untrust (select)

2. **PPP Profile with Dynamic IP Address**

   Network > PPP Profile > Edit: Enter the following, then click **OK**:

   PPP Profile: adsltest
   Authentication: CHAP (select) and PAP (select)
   Static IP: (deselect)
   Netmask: 255.255.255.255
   Passive: Don't challenge peer (deselect)
   Local Name: root
   Password: 123456 (does not display)

   PPP Profile with Static IP Address (Optional)

   Static IP: (select)

3. **Bind PPP Profile to ML Interface**

Network > Interfaces > Edit (ml1 interface) > Basic properties: Enter the following, then click **OK**:

WAN Encapsulation: Multi-Link PPP (select)
Binding a PPP Profile: adsltest (select)
Fixed IP: (select)
IP Address/Netmask: 192.168.3.22/24
Manage IP: 0.0.0.0
Interface Mode: Route (select)
Maximum Transfer Unit (MTU): 1500

4. **Bundle ADSL Interface to ML Interface**

Network > Interfaces > Edit (for adsl1/0 interface): Enter the following, then click **OK**:

VPI/VCI: 8/35
QoS Options: UBR (select)
Multiplexing Method: LLC (select)
RFC1483 Protocol Mode: Bridged (select)
Operating Mode: Auto (select)
Bind to a multilink interface: ml1 (select)
DNS Proxy: (select)

CLI

1. **Multilink Interface**

```
set interface ml1 zone untrust
set interface ml1 encap mlppp
```

2. **PPP Profile with Dynamic IP Address**

```
set ppp profile adsltest
set ppp profile adsltest auth type any
set ppp profile adsltest auth local-name root
set ppp profile adsltest auth secret 123456
```

PPP Profile with Static IP Address (Optional)

```
set ppp profile adsltest static-ip
set interface ml1 ip 192.168.3.22/32
```

3. **Bind PPP Profile to ML Interface**

```
set interface ml1 ppp profile adsltest
```

4. **Bundle ADSL Interface to ML Interface**

```
set interface adsl1/0 pvc 8 35 zone untrust
set interface adsl2/0 pvc 8 35 zone untrust
set interface adsl1/0 bundle ml1
set interface adsl2/0 bundle ml1
```

5. **Remove a Member Link from the Bundle**

```
unset interface adsl1/0 bundle
unset interface adsl2/0 bundle
```

Example 8: (Small Business) Allow Access to Local Servers

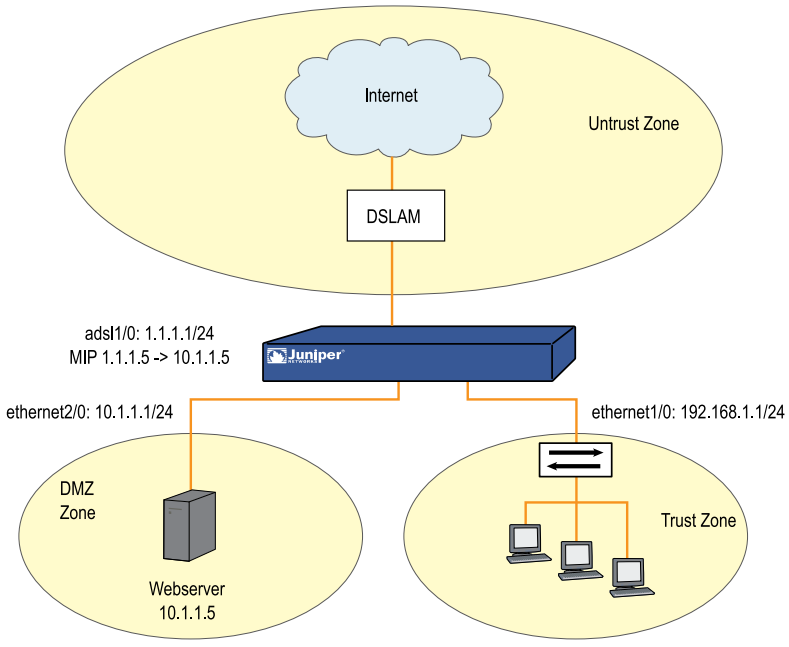
This example, as shown in Figure 482 on page 1981, explains how to configure the security device to allow internal hosts to access the Internet through the ADSL interface and allow Internet users to access a local Web server while protecting other internal hosts.

To segregate traffic flow to the Web server from the rest of your internal network, do the following:

1. Configure the trust and dmz interfaces.
2. Configure the ADSL interface and mapped IP (MIP). Set a PVC on the ADSL interface with the VPI/VCI pair 0/35 and the static IP address 1.1.1.1/24 which is assigned by the service provider. You bind the Trust interface to the Trust zone and assign it IP address 192.168.1.1/24. You configure a MIP to direct incoming HTTP traffic destined for the host 1.1.1.5 to the Web server at 10.1.1.5 in the DMZ zone.
3. Create a policy to allow only HTTP traffic to the zone in which the Web server resides.

   (Default policies allow all traffic from the Trust zone to the Untrust zone and block all traffic from the Untrust zone to the Trust zone.)

Figure 482: ADSL Interface Allowing Access to Local Servers



WebUI

1. **Ethernet Interfaces**

   Network > Interfaces > Edit (for ethernet1/0): Enter the following, then click **OK**:

   Static IP: (select)
   IP Address/Netmask: 192.168.1.1/24
   Interface Mode: NAT

   Network > DHCP > Edit (for ethernet1/0) > DHCP Server: Select **Apply**.

   > Addresses > New: Enter the following, then click **OK**:

   Dynamic: (select)
   IP Address Start: 192.168.1.3
   IP Address End: 192.168.1.33

   Network > Interfaces > Edit (for ethernet2/0): Enter the following, then click **OK**:

   Static IP: (select)
   IP Address/Netmask: 10.1.1.1/24
   Interface Mode: NAT

2. **ADSL Interface and MIP**

   Network > Interfaces > Edit (for adsl1/0): Enter the following, then click **Apply**:

   VPI/VCI: 0/35
   Zone Name: Untrust
   Static IP: (select)
   IP Address/Netmask: 1.1.1.1/24

   > MIP > New: Enter the following, then click **OK**:

   Mapped IP: 1.1.1.5
   Netmask: 255.255.255.255
   Host IP Address: 10.1.1.5
   Host Virtual Router Name: trust-vr

3. **Policy**

   Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

   Source Address:
   Address Book Entry: (select) Any
   Destination Address:
   Address Book Entry: (select), MIP(1.1.1.5)
   Service: HTTP
   Action: Permit

CLI

1. **Trust and DMZ Interfaces**

```
set interface ethernet1/0 ip 192.168.1.1/24
set interface ethernet1/0 nat
set interface ethernet1/0 dhcp server service
set interface ethernet1/0 dhcp server ip 192.168.1.3 192.168.1.33
set interface ethernet2/0 ip 10.1.1.1/24
set interface ethernet2/0 nat
```

2. **ADSL Interface and MIP**

```
set interface adsl1 pvc 0 35 zone untrust
set interface adsl1/0 ip 1.1.1.1/24
set interface adsl1/0 mip 1.1.1.5 host 10.1.1.5 netmask 255.255.255.255
vrouter trust-vr
```

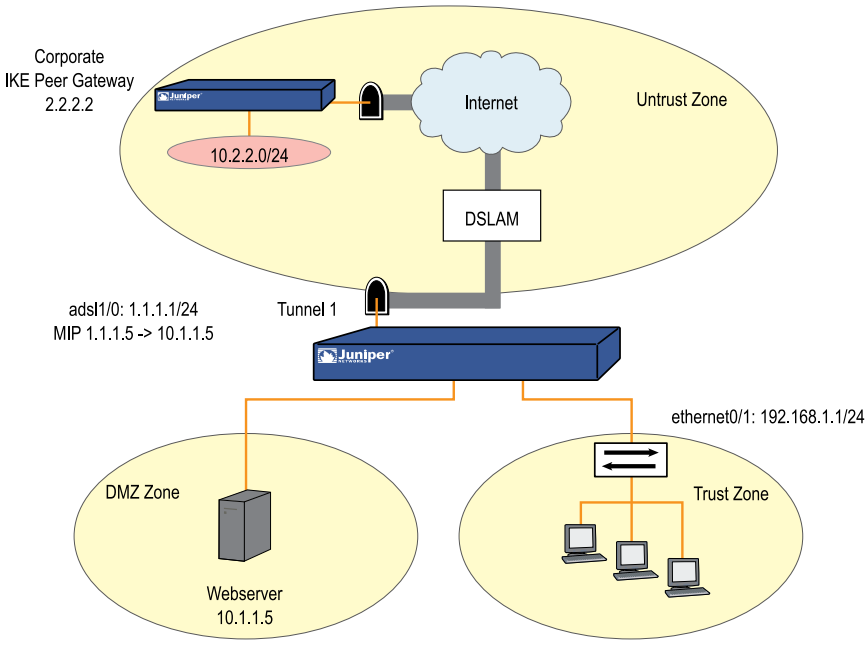
3. **Policy**

```
set policy from untrust to dmz any mip(1.1.1.5) http permit
save
```

Example 9: (Branch Office) VPN Tunnel Through ADSL

This example, as shown in Figure 483 on page 1984, explains how to configure a VPN tunnel to corporate headquarters through the ADSL interface on the security device and how to allow Internet access to local Web servers while protecting other internal hosts from being directly accessible from the Internet, as described in “Example 7: (Small Business) MLPPP ADSL” on page 1978.

Figure 483: VPN Tunnel Through ADSL Interface



This example also explains how to configure a route-based AutoKey IKE tunnel using a preshared secret. For the Phase 1 and 2 security levels, configure pre-g2-3des-sha for the Phase 1 proposal and the predefined "Compatible" set of proposals for Phase 2. To configure a VPN tunnel through the ADSL interface, do the following:

1. Configure the trust and dmz interfaces.
2. Configure the ADSL interface and mapped IP (MIP). Set a PVC on the ADSL interface with the VPI/VCI pair 0/35 and the static IP address 1.1.1.1/24 which is assigned by the service provider. You bind the Trust interface to the Trust zone and assign it IP address 192.168.1.1/24. You configure a MIP to direct incoming HTTP traffic destined for the host 1.1.1.5 to the Web server at 10.1.1.5 in the DMZ zone.
3. Create a tunnel interface and bind it to the Untrust security zone. To create a tunnel, do the following:
   a. Configure the tunnel interface to borrow the IP address from the adsl1 interface, which is also bound to the Untrust security zone (this is known as an "unnumbered" interface).
   b. Configure the VPN tunnel, designate the adsl1 interface as its outgoing interface in the Untrust zone, bind it to the tunnel interface, and configure its proxy-ID.
   c. Enter a route to the Corporate LAN through the tunnel interface.
   d. Set up policies for VPN traffic to pass between the branch office and corporate headquarters.
4. Create a policy to allow only HTTP traffic to the zone in which the Web server resides.

(Default policies allow all traffic from the Trust zone to the Untrust zone and block all traffic from the Untrust zone to the Trust zone.)

WebUI

1. **Trust and DMZ Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

   Static IP: (select)
   IP Address/Netmask: 192.168.1.1/24
   Interface Mode: NAT

   Network > DHCP > Edit (for ethernet0/1) > DHCP Server: Select **Apply**.

   > Addresses > New: Enter the following, then click **OK**:

   Dynamic: (select)
   IP Address Start: 192.168.1.3
   IP Address End: 192.168.1.33

   Network > Interfaces > Edit (for ethernet0/2): Enter the following, then click **OK**:

   Static IP: (select)
   IP Address/Netmask: 10.1.1.1/24
   Interface Mode: NAT

2. **ADSL Interface and MIP**

   Network > Interfaces > Edit (for adsl1/0): Enter the following, then click **Apply**:

   VPI/VCI: 0/35
   Zone Name: Untrust
   Static IP: (select)
   IP Address/Netmask: 1.1.1.1/24

   > MIP > New: Enter the following, then click **OK**:

   Mapped IP: 1.1.1.5
   Netmask: 255.255.255.255
   Host IP Address: 10.1.1.5
   Host Virtual Router Name: trust-vr

3. **VPN Tunnel**

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Unnumbered: (select)
   Interface: adsl1 (trust-vr)

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_Corp
Security Level: Custom
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 2.2.2.2
Preshared Key: h1p8A24nG5

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (for Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: Branch1_Corp
Security Level: Compatible
Remote Gateway:
Predefined: (select), To_Corp

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Security Level: Compatible
Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP/Netmask: 192.168.1.1/24
Remote IP/Netmask: 10.2.2.0/24
Service: ANY

Network > Routing > Destination > trust vr > New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
Interface: Tunnel.1

4. **Policies**

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select) Any
Destination Address:
Address Book Entry: (select), MIP(1.1.1.5)
Service: HTTP
Action: Permit

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: To_Corp
Source Address: 192.168.1.1/24
Destination Address: 10.2.2.0/24
Service: ANY
Action: Permit
Position at Top: (select)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Name: From_Corp
Source Address: 10.2.2.0/24
Destination Address: 192.168.1.1/24
Service: ANY
Action: Permit
Position at Top: (select)

CLI

1. **Trust and DMZ Interfaces**

```
set interface ethernet0/1 ip 192.168.1.1/24
set interface ethernet0/1 nat
set interface ethernet0/1 dhcp server service
set interface ethernet0/1 dhcp server ip 192.168.1.3 192.168.1.33
set interface ethernet0/2 ip 10.1.1.1/24
set interface ethernet0/2 nat
```

2. **ADSL Interface and MIP**

```
set interface adsl1/0 pvc 0 35 zone untrust
set interface adsl1/0 ip 1.1.1.1/24
set interface adsl1/0 mip 1.1.1.5 host 10.1.1.5 netmask 255.255.255.255
vrouter trust-vr
```

3. **VPN Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface adsl1
set ike gateway To_Corp address 2.2.2.2 main outgoing-interface adsl1/0
preshare hIp8A24nG5 proposal pre-g2-3des-sha
set vpn Branch1_Corp gateway To_Corp sec-level compatible
set vpn Branch1_Corp bind interface tunnel.1
set vpn Branch1_Corp proxy-id local-ip 192.168.1.1/24 remote-ip 10.2.2.0/24
 any
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.1
```

4. **Policies**

```
set policy from untrust to dmz any mip(1.1.1.5) http permit
set policy top name “To Corp” from trust to untrust 192.168.1.1/24
10.2.2.0/24 any permit
set policy top name “From Corp” from untrust to trust 10.2.2.0/24
192.168.1.1/24 any permit
save
```

Example 10: (Branch Office) Secondary VPN Tunnel

This example, as shown in Figure 484 on page 1989, explains how to configure the security device as a firewall with both an Internet connection and a connection to corporate headquarters through the ADSL interface. This example is similar to the

configuration shown in "Example 9: (Branch Office) VPN Tunnel Through ADSL" on page 1983, but you create two PVCs: one to the Internet and another to corporate headquarters. You also configure a VPN tunnel through the Internet to corporate headquarters as a secondary connection.

To configure and primary and secondary VPN tunnel through ADSL, do the following:

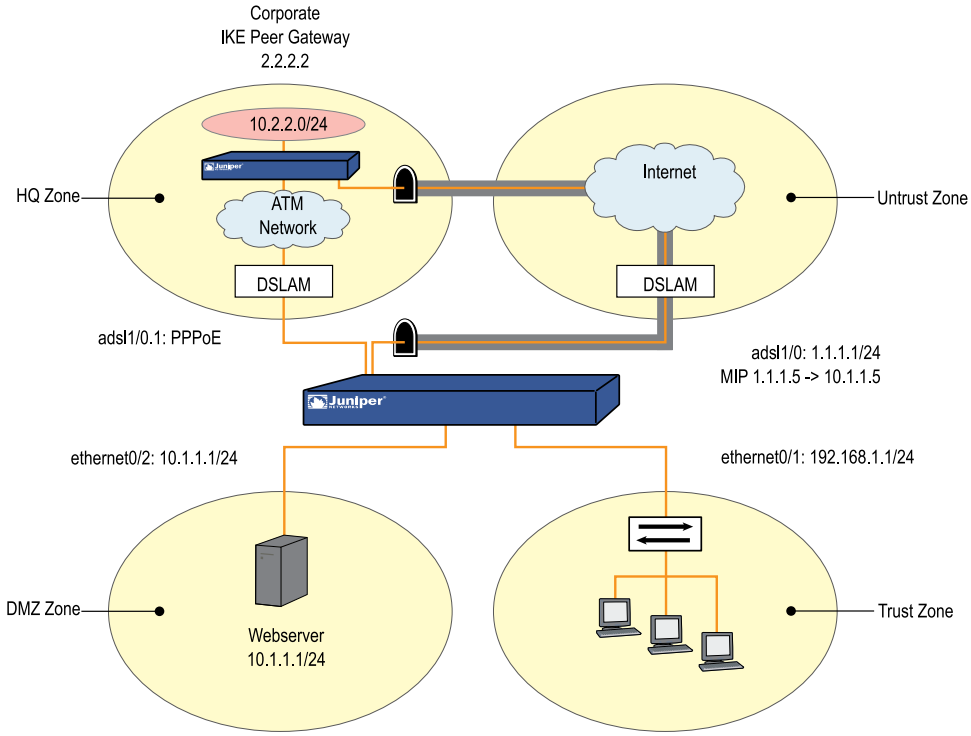
1. Configure the trust and dmz interfaces.
2. Configure the ADSL interface and mapped IP (MIP). Set a PVC on the ADSL interface with the VPI/VCI pair 0/35 and the static IP address 1.1.1.1/24 which is assigned by the service provider. You bind the Trust interface to the Trust zone and assign it IP address 192.168.1.1/24. You configure a MIP to direct incoming HTTP traffic destined for the host 1.1.1.5 to the Web server at 10.1.1.5 in the DMZ zone.
3. Configure the Headquarter (HQ) custom zone.
4. Configure an ADSL subinterface. Set an additional PVC on the security device by creating the ADSL subinterface adsl1.1. The adsl1.1 subinterface with the VPI/VCI pair value 1/35 that uses LLC encapsulation and a PPPoE instance named poe1, which is bound to the subinterface. You then need to define policies to allow the flow of traffic to and from the HQ zone.

NOTE: You can bind the ADSL interface and each of its subinterfaces to different security zones; you bind the ADSL subinterface to the custom zone "HQ" (the main ADSL interface is bound to the Untrust zone by default).

5. Create a tunnel interface and bind it to the Untrust security zone. To create a tunnel, do the following:
   a. Configure the tunnel interface to borrow the IP address from the adsl1 interface, which is also bound to the Untrust security zone (this is known as an "unnumbered" interface).
   b. Configure the IKE gateway.
   c. Configure the VPN tunnel, designate the adsl1 interface as its outgoing interface in the Untrust zone, bind it to the tunnel interface, and configure its proxy-ID.
6. Create a virtual route (trust-vr).
7. Set up policies for VPN traffic to pass between the branch office and corporate headquarters.

   (Default policies allow all traffic from the Trust zone to the Untrust zone and block all traffic from the Untrust zone to the Trust zone.)

Figure 484: ADSL Interface with a Secondary Tunnel



Because you have two different routes between workstations in the Trust zone and corporate headquarters—one using the adsl1.1 interface and another using the VPN tunnel interface—you need to specify which route is "preferred." This is done by setting the metric for the route through the VPN tunnel higher than the route through the adsl1.1 interface.

WebUI

1. **Trust and DMZ Interfaces**

   Network > Interfaces > Edit (for ethernet0/1): Enter the following, then click **OK**:

   Static IP: (select)
   IP Address/Netmask: 192.168.1.1/24
   Interface Mode: NAT

   Network > DHCP > Edit (for ethernet0/1) > DHCP Server: Select **Apply**.

   > Addresses > New: Enter the following, then click **OK**:

   Dynamic: (select)
   IP Address Start: 192.168.1.3
   IP Address End: 192.168.1.33

   Network > Interfaces > Edit (for ethernet0/2): Enter the following, then click **OK**:

Static IP: (select)
IP Address/Netmask: 10.1.1.1/24
Interface Mode: NAT

2. **ADSL Interface and MIP**

   Network > Interfaces > Edit (for adsl1/0): Enter the following, then click **Apply**:

   VPI/VCI: 0/35
   Zone Name: Untrust
   Static IP: (select)
   IP Address/Netmask: 1.1.1.1/24

   > MIP > New: Enter the following, then click **OK**:

   Mapped IP: 1.1.1.5
   Netmask: 255.255.255.255
   Host IP Address: 10.1.1.5
   Host Virtual Router Name: trust-vr

3. **HQ Zone**

   Network > Zones > New: Enter the following, then click **OK**:

   Zone Name: HQ
   Block Intra-Zone Traffic: (select)

4. **ADSL Subinterface**

   Network > Interfaces > New ADSL Sub-IF: Enter the following, then click **OK**:

   Interface Name: adsl1/0.1
   VPI/VCI: 1/35
   Encapsulation: LLC (selected)
   Zone: HQ (select)

   Network > PPPoE > New: Enter the following, then click **OK**:

   PPPoE Instance: poe1
   Bound to Interface: adsl1/0.1 (select)
   Username: felix
   Password: ioP936QNlwab48Rc

5. **VPN Tunnel**

   Network > Interfaces > New Tunnel IF: Enter the following, then click **OK**:

   Tunnel Interface Name: tunnel.1
   Zone (VR): Untrust (trust-vr)
   Unnumbered: (select)
   Interface: adsl1 (trust-vr)

   VPNs > AutoKey Advanced > Gateway > New: Enter the following, then click **OK**:

Gateway Name: To_Corp
Security Level: Custom
Remote Gateway Type:
Static IP Address: (select), IP Address/Hostname: 2.2.2.2
Preshared Key: h1p8A24nG5

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic Gateway configuration page:

Security Level: Custom
Phase 1 Proposal (for Custom Security Level): pre-g2-3des-sha
Mode (Initiator): Main (ID Protection)

VPNs > AutoKey IKE > New: Enter the following, then click **OK**:

VPN Name: Branch1_Corp
Security Level: Compatible
Remote Gateway:
Predefined: (select), To_Corp

> Advanced: Enter the following advanced settings, then click **Return** to return to the basic AutoKey IKE configuration page:

Security Level: Compatible
Bind to: Tunnel Interface, tunnel.1
Proxy-ID: (select)
Local IP/Netmask: 192.168.1.1/24
Remote IP/Netmask: 10.2.2.0/24
Service: ANY

6. **Routes**

Network > Routing > Destination > trust vr > New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
Interface: adsl1.1
Metric: 1

Network > Routing > Destination > trust vr > New: Enter the following, then click **OK**:

Network Address/Netmask: 10.2.2.0/24
Gateway: (select)
Interface: Tunnel.1
Metric: 5

7. **Policies**

Policies (From: Trust, To: HQ) > New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select) Any
Destination Address:
Address Book Entry: (select) Any

Service: ANY
Action: Permit

Policies (From: HQ, To: Trust) > New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select) Any
Destination Address:
Address Book Entry: (select) Any
Service: ANY
Action: Permit

Policies (From: DMZ, To: HQ) > New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select) Any
Destination Address:
Address Book Entry: (select) Any
Service: ANY
Action: Permit

Policies (From: HQ, To: DMZ) > New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select) Any
Destination Address:
Address Book Entry: (select) Any
Service: ANY
Action: Permit

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select) Any
Destination Address:
Address Book Entry: (select), MIP(1.1.1.5)
Service: HTTP
Action: Permit

Policies > (From: Trust, To: Untrust) New: Enter the following, then click **OK**:

Name: To_Corp
Source Address: 192.168.1.1/24
Destination Address: 10.2.2.0/24
Service: ANY
Action: Permit
Position at Top: (select)

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Name: From_Corp
Source Address: 10.2.2.0/24
Destination Address: 192.168.1.1/24
Service: ANY
Action: Permit
Position at Top: (select)

CLI

1. **Trust and DMZ Interfaces**

```
set interface ethernet0/1 ip 192.168.1.1/24
set interface ethernet0/1 nat
set interface ethernet0/1 dhcp server service
set interface ethernet0/1 dhcp server ip 192.168.1.3 192.168.1.33
set interface ethernet0/2 ip 10.1.1.1/24
set interface ethernet0/2 nat
```

2. **ADSL Interface and MIP**

```
set interface adsl1/0 pvc 0 35 zone untrust
set interface adsl1/0 ip 1.1.1.1/24
set interface adsl1/0 mip 1.1.1.5 host 10.1.1.5 netmask 255.255.255.255
vrouter trust-vr
```

3. **HQ Zone**

```
set zone name HQ
set zone HQ block
set zone HQ vrouter trust-vr
```

4. **ADSL Subinterface**

```
set interface adsl1/0.1 pvc 1 35 mux llc zone HQ
set pppoe name poe1 username felix password ioP936QNlwab48Rc
set pppoe name poe1 interface adsl1/0.1
```

5. **VPN Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface adsl1/0
set ike gateway To_Corp address 2.2.2.2 main outgoing-interface adsl1/0
preshare hIp8A24nG5 proposal pre-g2-3des-sha
set vpn Branch1_Corp gateway To_Corp sec-level compatible
set vpn Branch1_Corp bind interface tunnel.1
set vpn Branch1_Corp proxy-id local-ip 192.168.1.1/24 remote-ip 10.2.2.0/24

any
```

6. **Routes**

```
set vrouter trust-vr route 10.2.2.0/24 interface adsl1/0.1 metric 1
set vrouter trust-vr route 10.2.2.0/24 interface tunnel.1 metric 5
```

7. **Policies**

```
set policy from trust to HQ any any any permit
set policy from HQ to trust any any any permit
set policy from dmz to HQ any any any permit
set policy from HQ to dmz any any any permit
set policy from untrust to dmz any mip(1.1.1.5) http permit
set policy top name “To Corp”  from trust to untrust 192.168.1.1/24
```

```
10.2.2.0/24 any permit
set policy top name “From Corp”  from untrust to trust 10.2.2.0/24
192.168.1.1/24 any permit
save
```

Chapter 60

ISP Failover and Dial Recovery

Devices that support ISP failover and dial recovery have either an Aux serial port (for an external V.92 modem) or an ISDN BRI port or built-in V.92 analog modem port. On some models these features are available as mini-PIM options. These ports are commonly used as backup ports when the primary link (the Ethernet port) fails. The primary link is typically an ADSL modem to the ISP with Ethernet connectivity to the security device. VPN monitoring or physical port failure are typical triggers for failover to the backup interface.

This chapter contains the following sections:

Setting ISP Priority for Failover

You, as a root administrator (not a trustee admin), can configure a total of four ISPs. The priority of each ISP must be a unique number. You can also configure one or more ISP entries with a priority of zero for testing (monitoring); however, any ISP entry assigned a zero will not be used for failover.

When using a modem connection, a trustee administrator can manually change an ISP priority. If a failover situation occurs, the priority assigned to an ISP indicates in what order relative to other ISPs that a particular ISP will be contacted. The lower the value, the higher the priority of the ISP. Trustee admins can also check the availability of an ISP with a priority setting of zero (0).

WebUI

Home > Interface link status: Click **Edit**. Then click **ISP** for the serial interface.

CLI

```
set interface serial0/0 modem isp pac-bell-1 priority 1
set interface serial0/0 modem isp pac-bell-1 primary-number 555-55-55
alternative-number 666-66-66
set interface serial0/0 modem isp pac-bell-1 account login rbrockie password !2007ah
set interface serial0/0 modem isp pac-bell-2 priority 2
set interface serial0/0 modem isp pac-bell-2 primary-number 777-77-77
```

```
alternative-number 888-88-88
set interface serial0/0 modem isp pac-bell-2 account login rbrockie password !2007ah
save
```

Defining Conditions for ISP Failover

After setting up the ISP information and priorities, you can selectively monitor a route in the untrust-vr that can trigger a failover to the next highest priority ISP (next lowest priority number) when the monitored route disappears from the untrust-vr routing table. When the failover to ISP 2 occurs, it does not necessarily mean that ISP 1 went down or failed. It means that a particular route that you want to access that is beyond ISP 1 became unavailable and you want the device to wait a specified number of seconds before giving up and calling the backup ISP.

To use this feature you specify a particular route with an IP address that currently appears in the untrust-vr. Optionally, you can specify the number of seconds for the device to wait before it calls the backup ISP. The default holddown time is 30 seconds.

In the previous example you, as root administrator, set up a priority 1 ISP and a backup ISP. In this example, you set the security device to monitor the route 1.1.1.1/24, which you have identified as an interesting or important route. This route currently exists in the untrust-vr. You set the holddown timer to be 100 seconds. You must use the CLI to set which route in the Untrust-vr you want to monitor. If the specified route becomes unavailable, failover to the next ISP (priority 2) occurs.

WebUI

Home > Interface link status: Click Edit. Then click ISP Failover for the serial interface.

CLI

```
set interface serial0/0 modem isp-failover holddown 100
set interface serial0/0 modem isp-failover type route vrouter untrust-vr 1.1.1.1/24
save
```

Configuring a Dialup Recovery Solution

You can set up a dialup disaster-recovery solution for your network by configuring the following items:

- Security device
- Modem or ISDN terminal adapter (TA)
- Interface-failover trigger mechanism
- Method for the far-end device (another router or firewall device) to identify a return path to the sending device

You can choose one of three different failover mechanisms to trigger an interface failover:

- **Track IP** monitors the availability of a specified IPv4 address to determine failover. To use IP tracking, enter the **set *interface* interface backup interface serial0/0 type track-ip** command (where ***interface*** is the main interface).
- **Tunnel tracking** monitors VPN tunnel status to determine failover. To track by tunnel interface, enter the **set *interface* interface backup interface serial0/0 type tunnel-if** command (where ***interface*** is the main interface).
- **Route tracking** monitors a known route's status. The route entry can be propagated by a dynamic routing protocol, such as BGP or OSPF. If a BGP adjacency is lost, the security device removes all routes learned from that BGP peer. If the route entry is not active for a period that exceeds the hold-down time, the security device triggers an interface failover to a backup interface. This feature requires an exact address match to an active route in the routing table of the specified vrouter to avoid failover. To achieve failover by route, enter the **set *interface* interface backup interface serial0/0 type route vroute untrust 1.1.1.1/24** command (where interface is main interface and 1.1.1.1 is monitor route's IP address).

In the following example (see Figure 485 on page 1998), a computer located in downtown San Jose, California is networked to a server in a remote office in another part of San Jose. The security device always uses tunnel.1, a VPN tunnel bound to ethernet3 (e3), to send traffic from the computer (PC) across the Internet to the firewall to access a remote server. The traffic uses static routes and VPN monitoring (default settings). Tunnel.1 has a metric of 1 that ensures it will always be the preferred route to the firewall; tunnel.2 has a higher metric, 180, so that does not become a preferred route.

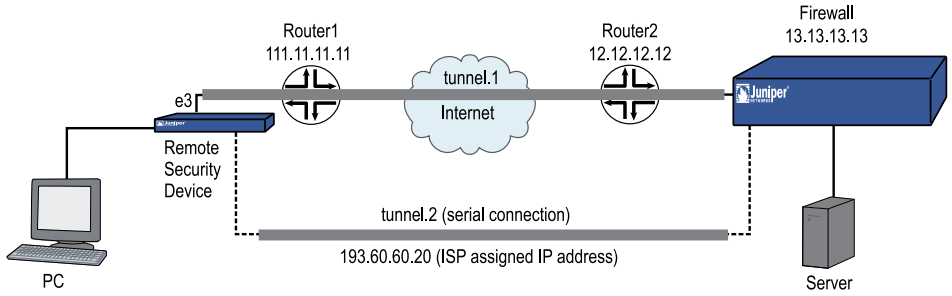
If tunnel.1 goes down, the security device brings up tunnel.2, bound to the serial interface, to contact the firewall. In this example, tunnel tracking is preferred over IP tracking to achieve the interface failover. With IP tracking, if the link goes down, failover occurs; but it will be unknown if failover occurred because the VPN tunnel is actually down or not. The tunnel-tracking feature fails over only when the tunnel interface bound to the primary Untrust interface goes down. The primary Untrust interface can fail if a cable is unplugged or if the VPN monitor is triggered, which also means that the tunnel failed.

The security device tunnel.2 (serial) interface uses an ISP-assigned IP address to connect over ISDN to the firewall.

The failover is set to "auto" so that when the interface becomes usable again, failback to tunnel.1 occurs.

VPN monitoring becomes active when you enter the **set vpn** *name* **monitor rekey** command. If necessary, you can choose to change the default interval and threshold settings by entering the **set vpnmonitor { interval | threshold }** command. You can view VPN monitoring settings by entering the **get vpnmonitor** command.

Figure 485: Dial Recovery Configuration



Following is the configuration for the remote security device as shown in Figure 485 on page 1998.

WebUI

Configure the serial and tunnel interfaces

Network > Interfaces: Select **Edit** for the serial interface, and bind the interface to the Untrust zone. Click **Apply**.

Network > Interfaces: Select **New** for a tunnel interface (tunnel.1) and bind the interface to the Untrust zone. Configure settings as applicable, then click **Apply**.

Network > Interfaces: Select **New** for a tunnel interface (tunnel.2) and bind the interface to the Untrust zone. Configure settings as applicable, then click **Apply**.

Set static routes and metrics

Network > Routing > Destination: Select **New** route for the untrust-vr; set the IP address, gateway information, and metric; then click **OK**.

Network > Routing > Destinations: Select **New** route for the untrust-vr, set the IP address from that the ISP assigned for the serial connection, gateway information (if applicable), metric (higher number than previous entry), then click **OK**.

Set the IKE gateway

VPNs > AutoKey IKE > Edit: Enter IKE requirements, then click **OK**.

Bind the VPNs to the IKE gateways

VPNs > AutoKey IKE > Edit: Select **Advanced** and bind the VPN to the IKE gateway for tunnel.1. Click **OK**.

VPNs > AutoKey IKE > Edit: Select **Advanced** and bind the VPN to the IKE gateway for tunnel.2. Click **OK**.

Configure interface failover

Configure from CLI.

Configure the inband modem port settings

Network > Interface (Modem)

Set the primary ISP account

Network > Interface (ISP)

CLI

Configure the serial and tunnel interfaces

```
set interface serial0/0 zone untrust
set interface ethernet0/3 zone untrust
set interface ethernet0/3 ip 111.11.11.10/24
set interface tunnel.1 ip unnumbered interface ethernet0/3
set interface serial0/0 ip 193.60.60.19/24
set interface tunnel.2 ip unnumbered interface serial0/0
```

Set static routes and metrics

```
set route 0.0.0.0/0 interface ethernet0/3 metric 1
set route 0.0.0.0/0 interface serial0/0 metric 180
```

Set the IKE gateway

```
set ike gateway eth address 13.13.13.13 Main outgoing-interface ethernet0/3
preshare 123qwe! sec-level standard

set ike gateway serial address 13.13.13.13 Main outgoing-interface serial0/0
preshare 123qwe! sec-level standard
```

Bind the VPNs to the IKE gateways

```
set vpn eth gateway eth no-replay tunnel idletime 0 sec-level standard
set vpn eth monitor rekey
set vpn eth id 1 bind interface tunnel.1

set vpn serial gateway serial no-replay tunnel idletime 0 sec-level standard
set vpn serial monitor rekey
set vpn serial id 2 bind interface tunnel.2
```

Configure interface failover

```
set interface ethernet0/3 backup interface serial0/0 type tunnel-if
set interface ethernet0/3 backup deactivation-delay 5
```

```
set interface ethernet0/3 backup activation-delay 5
set interface ethernet0/3 backup auto
set vpn eth backup-weight 100
```

Configure the inband modem port settings

```
set interface serial0/0 modem settings port-1 init-strings AT&F
set interface serial0/0 modem settings port-1 active
```

Set the primary ISP account

```
set interface serial0/0 modem isp isp-1 priority 1
set interface serial0/0 modem isp isp-1 primary-number 555-55-55 alternative-number
 666-66-66
set interface serial0/0 modem isp isp-1 account login rbrockie pass !2007ah
```

Chapter 61

Wireless Local Area Network

Juniper Networks wireless devices and systems provide wireless local area network (WLAN) connections with integrated Internet Protocol security virtual private network (IPsec VPN) and firewall services for wireless clients, such as telecommuters, branch offices, or retail outlets.

This chapter explains how to configure wireless interfaces and provides sample configurations. It contains the following sections:

Overview

Wireless security devices and systems connect wireless users or other wireless devices to wired or wireless networks. A device that enables a wireless device to access a local area network is a wireless access point (AP). The features listed in this chapter require security devices or systems with built-in wireless interfaces.

NOTE: All wireless platforms can support up to four active wireless interfaces at one time.

ScreenOS runs on wireless security devices to provide routing and firewall services to interface with your existing or planned wired networks.

As with wired interfaces, you can configure the following for a wireless interface:

- IP address/netmask and Manage IP address
- Management options, such as WebUI, SNMP, Telnet, SSH, or SSL
- Address translation
- Domain Name Services (DNS) Proxy
- WebAuth
- Dynamic Host Configuration Protocol (DHCP) server functionality (DHCP client or relay functionality is not supported)

NOTE: See *"Fundamentals" on page 15* and *"Address Translation" on page 1467*.

The following wireless ScreenOS features enable you to manage and secure a WLAN:

- Up to 4 WLANs per system
- Up to 16 service set identifier (SSIDs)
- Authentication
  - Open
  - Wired Equivalent Privacy (WEP) (shared-key)
  - WEP (802.1X)
  - Wi-Fi Protected Access (WPA) (preshared key)
  - WPA (802.1X)
  - WPA2 (preshared key)
  - WPA2 (802.1X)
- Encryption
  - Advanced Encryption Standard (AES)
  - Temporal Key Integrity Protocol (TKIP)
  - WEP
- Wi-Fi™ Multimedia (WMM) Quality of Service feature
- Turbo mode with nearly double the performance per radio band

NOTE: Refer to the datasheet that accompanied the product for capacity statements.

Wireless Product Interface Naming Differences

Wireless products support up to four WLANs. Entering commands in the WebUI and CLI differ by product.

Some wireless products have two radio transceivers:

- 2.4 GHz (WLAN 0)
- 5GHz (WLAN 1)

Transceiver-specific parameters automatically appear in the WebUI or CLI.

Basic Wireless Network Feature Configuration

Certain wireless features must be configured, but other features are optional. Each time you make changes to a wireless interface, however, you must reactivate the WLAN (for more information, see “Reactivating a WLAN Configuration” on page 2007).

This section contains the following:

Creating a Service Set Identifier

A wireless network is identified by a service set identifier (SSID). SSIDs allow you to maintain multiple WLANs using one wireless security device. You must bind an SSID to a wireless interface, which can be bound to a security zone. The SSID is a unique name that can be up to 32 text characters in length. To use spaces in the name, you must enclose the name in double quotation marks.

NOTE: You are not constrained by the number of wireless interfaces in the security device when creating SSIDs. You can have more SSIDs than the number of wireless interfaces. You can bind a maximum of four SSIDs to wireless interfaces. You can activate site- or time-specific WLANs by binding and unbinding SSIDs to wireless interfaces as your network needs change.

In the following example, you configure an SSID with the name “My Home Network.” For increased security, you can make the name difficult to guess and not include the location of the device in the SSID name.

WebUI

Wireless > SSID > New: Enter the name in the SSID field, then click **OK**.

CLI

```
set ssid name “My Home Network”
```

Suppressing SSID Broadcast

After creating and SSID, you can disable the broadcasting of SSIDs in beacons that are advertised by the security device. If SSID broadcasting is disabled, only wireless clients that know of the SSID are able to associate. By default, SSIDs are broadcast in beacons.

To suppress an SSID broadcast, use one of the WebUI as follows:

WebUI

Wireless > SSID > Edit (for *name_str*): Select **Disable SSID Broadcast**, then click **OK**.

CLI

```
set ssid name_str ssid-suppression
```

Isolating a Client

By isolating the client, you prohibit wireless clients in the same subnet from communicating directly with each other. This forces each client to communicate through the firewall. By default, this option is disabled.

To prohibit wireless clients in the same subnet from communicating directly with each other, use one of the following procedures.

WebUI

Wireless > SSID > Edit (for *name_str*): Select **SSID Client Isolation**, then click **OK**.

CLI

```
set ssid name_str client-isolation enable
```

Setting the Operation Mode for a 2.4 GHz Radio Transceiver

You can configure WLAN 0 to operate in one of the following modes:

- 802.11b mode (11b)
- 11g mode with 802.11b compatibility (11g)
- 11g mode without 802.11b compatibility (11g-only)
- Turbo static 11g-only mode

The **11g** mode without 802.11b compatibility (**11g-only**) option prevents the security device to associate with 802.11b clients. When the **11g** option is selected, the device allows association with 802.11b and 802.11g clients. The **11b** option sets the device to only allow association with 802.11b clients. **Turbo** mode is a high performance option.

NOTE: Only 11g-turbo-supported clients can connect when Turbo mode is enabled for the security device.

The default mode is set to 802.11g; this mode allows 802.11g and 802.11b clients.

To set the operational mode to 802.11g, use one of the following procedures:

WebUI

Wireless > WLAN > General Settings: Select **802.11g** from the Operation Mode list, then click **Apply**. (If the security device has more than one radio, make the selection for the 2.4 GHz radio.)

CLI

To set the operational mode to 802.11g for a security device with one radio, enter the following command:

set wlan mode 11g

To set the operational mode to 802.11g for a security device with two radios, enter the following command:

set wlan 0 mode 11g

Setting the Operation Mode for a 5GHz Radio Transceiver

You can configure WLAN 1, the 5GHz transceiver, to operate in 802.11a mode or in Turbo mode. 802.11a mode operates within a 5GHz frequency band and is the default operational mode.

NOTE: Only 11a-Turbo-supported clients can associate when Turbo mode is enabled on the security device.

By enabling Turbo mode, you can increase the performance of downloads.

In the following example, you enable Turbo mode for WLAN 1.

WebUI

Wireless > WLAN > General Settings: Select **Turbo** from the Operation Mode list, then click **Apply**.

CLI

To enable Turbo mode for WLAN 1, enter the following command:

```
set wlan 1 mode turbo
```

Configuring Minimum Data Transmit Rate

You can set the minimum data transmit rate in megabits per second (Mbps) for sending frames. The data transmit rate depends on the radio type and can be one of the following.

- 802.11a: 6, 9, 12, 18, 24, 36, 48, 54
- 802.11a with XR enabled: 0.25, 0.5, 3, 6, 9, 12, 18, 24, 36, 48, 54
- 802.11b: 1, 2, 5.5, 11
- 802.11g: 1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54
- 802.11g with XR enabled: 0.25, 0.5, 1, 2, 3, 5.5, 11, 6, 9, 12, 18, 24, 36, 48, 54
- If Turbo mode is enabled: 12, 18, 24, 36, 48, 72, 96, 108

The **auto** rate, which is the default value, uses the best rate first and then automatically falls back to the next rate if transmission fails. Juniper wireless devices now support FCC3 channels

To configure the data transmit rate, use one of the following procedures:

WebUI

Wireless > General Settings: Select the rate from the Transmit Data Rate list (if the security device has more than one radio, make the selection for the radio you want).

CLI

To set the data transmit rate to 11 Mbps on a security device with one radio, enter the following command:

```
set wlan transmit rate 11
```

To set the data transmit rate to 54 Mbps for the 5GHz radio on a security device with two radios, enter the following command:

```
set wlan 1 transmit rate 54
```

Configuring Transmit Power

You can set the power transmission and adjust the radio range for the security device. You can set the power level to an eighth, full, half, minimum, or quarter of the maximum transmit power, which is the maximum power allowed in the country the security device is operating in. The default is full power.

To configure the transmit power, use one of the following procedures:

WebUI

Wireless > General Settings: Select the power level from the Transmit Power list (if the security device has more than one radio, make the selection for the radio you want).

CLI

To set the transmit power to half on a security device with one radio, enter the following command:

```
set wlan transmit power half
```

To set the transmit per to half for the 5GHz radio on a security device with two radios, enter the following command:

```
set wlan 1 transmit power half
```

Reactivating a WLAN Configuration

After making any changes to a WLAN configuration, you must reactivate the WLAN subsystem within the device, which reboots the wireless interfaces. Any WLAN-related configuration changes take effect only after you reactivate this subsystem.

Depending on the network, the reactivation process can take 60 seconds or more to complete. Wireless traffic is disrupted, and all wireless client sessions are terminated. Wireless clients must reconnect to the wireless network to reestablish connectivity.

To reactivate the system, use one of the following procedures:

WebUI

For security devices with one radio:

Wireless > Activate Changes: Click the Activate Changes button.

For security devices with two radios:

Click the Activate Changes button at the top of any wireless page.

CLI

To reactivate the WLAN, enter the following command:

```
exec wlan reactivate
```

Configuring Authentication and Encryption for SSIDs

The settings for authentication and encryption are specific to each SSID. You can configure different authentication and encryption preferences for each SSID.

ScreenOS supports the following authentication and encryption mechanisms for WLANs:

- Authentication
  - Open
  - WEP (shared-key)
  - WEP (802.1X)
  - WPA (preshared key)
  - WPA (802.1X)
  - WPA2 (preshared key)
  - WPA2 (802.1X)
- Encryption
  - Advanced Encryption Standard (AES)
  - Temporal Key Integrity Protocol (TKIP)
  - WEP

The following sections describe WEP, WPA, and WPA2 and explain how to configure them in an SSID. For more information about EAP and 802.1X, see *"Extensible Authentication for Wireless and Ethernet Interfaces" on page 1661*.

Configuring Wired Equivalent Privacy

Wired Equivalent Privacy (WEP) provides confidentiality for wireless communication. It uses the Rivest Cipher 4 (RC4) stream cipher algorithm to encrypt and decrypt data as it travels over the wireless link. You can store the WEP key locally or negotiate a key dynamically with an external authentication server. Wireless clients in turn store this key on their systems.

ScreenOS supports two WEP key lengths: 40 and 104 bits. The keys are concatenated with a 24-bit initialization vector (IV) and result in 64 and 128 bit lengths.

NOTE: Some third-party wireless clients include the 24 bits from the IV when specifying their WEP key lengths. To avoid connectivity issues, the same WEP key length described as 40 or 104 bits on the wireless device might actually be the same length as a key described as 64 or 128 bits on a client.

Multiple WEP Keys

You can create up to four WEP keys per SSID.

If you create only one WEP key, the wireless device uses that key to authenticate wireless clients in that SSID and to encrypt and decrypt traffic sent between itself and the clients.

You can also define multiple WEP keys on the wireless device—up to four keys for a single SSID. Using multiple keys allows you to adjust the level of security for different wireless clients within the same SSID. You can use longer keys to provide greater security for some traffic and shorter keys to reduce processing overhead for other, less critical, traffic. The wireless devices use the WEP key specified as **default** for encryption, and another key (or the default key again) for authenticating and decrypting. If you do not specify a key as the default, the first key you define becomes the default.

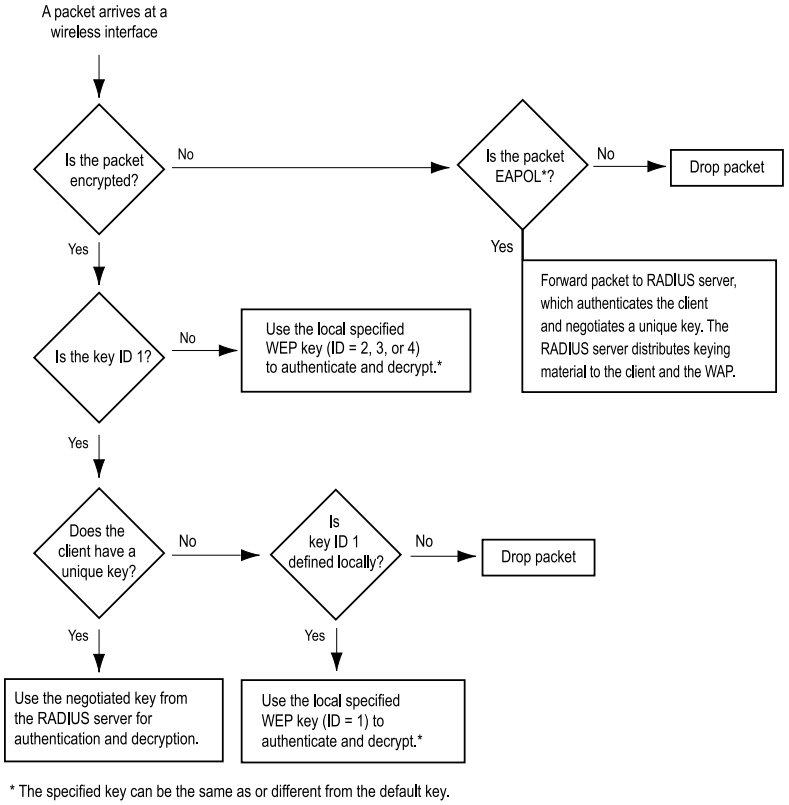
Keep the following in mind about WEP key storage and key ID numbers:

- When clients use a unique, dynamically created WEP key from an external RADIUS server, the wireless device uses this unique specific key—which it also receives from the RADIUS server—for bidirectional communication.
- When wireless clients use statically defined WEP keys stored locally on the wireless device, the device uses the default key to encrypt all wireless traffic that it transmits. The clients must also have this key loaded to be able to decrypt traffic from the wireless device.
- If you store multiple WEP keys on the wireless device, the default key ID can be 1, 2, 3, or 4.
- If you store some WEP keys on the wireless device and use dynamically created WEP keys from an external RADIUS server, the ID for the default WEP key on the wireless device cannot be 1, because the RADIUS server uses 1 as the ID for all of its keys. The wireless device can use a default WEP key with key ID 2, 3, or 4 for encryption, and it can use a statically defined WEP key with ID 1, 2, 3, or 4 for authentication and decryption.
- If you exclusively use WEP keys from a RADIUS server, the server uses a key ID of 1 for all its keys. RADIUS creates and distributes a different key per session for each client.
- You can specify a different locally stored key for the wireless device to use when authenticating and decrypting traffic it receives from wireless clients. The clients must have this key and its ID number loaded to be able to authenticate themselves and encrypt traffic sent to the device. (If a client does not supply a key ID, the device tries to use the default WEP key to authenticate the client and decrypt its traffic.)

NOTE: If a client uses only one key for encrypting, decrypting, and authenticating, then it must use the default WEP key.

Figure 486 on page 2010 shows how the wireless device processes a wireless connection request when WEP keys are stored locally and when they come from a RADIUS server.

Figure 486: Connectivity Process with WEP on RADIUS Server



Configuring Open Authentication

You can configure open authentication, which specifies that no authentication is performed. The wireless client provides the SSID and is connected to the wireless network. When using open authentication, you can specify the following encryption key options:

- No encryption
- WEP encryption
  - Local key source: The WEP key is stored on the security device. You must specify a default key.
  - Server: The WEP key is a dynamic key negotiated from a RADIUS server.

- Both: Only available for security devices with one radio, the WEP key is stored on the device and a RADIUS server. You must specify a default key.

You can specify up to four WEP keys per SSID.

Configuring Open Authentication with WEP Keys from RADIUS Server

The following examples use the following parameters for the SSID named hr:

- Open authentication
- WEP encryption
- Dynamically generated WEP key obtained from RADIUS server named rs1

WebUI

Use the following procedure if you have a security device with one radio:

Wireless > SSID > Edit: Enter the following, then click **OK**:

WEP Based Authentication and Encryption Methods: Open, WEP Encryption
Key Source: Server
Auth Server: rs1 (click **Create new Auth Server** to define RADIUS server if it does not already exist)

Use the following procedure if you have a security device with two radios:

Wireless > SSID > Edit: Enter the following, then click **OK**.

802.1X Based Authentication and Encryption Methods: 802.1X
Auth Server: rs1 (click **Create new Auth Server** to define RADIUS server if it does not already exist)

CLI

Use the following command if you have a security device with one radio:

```
set ssid hr authentication open encryption wep key-source server rs1
```

Use the following command if you have a security device with two radios:

```
set ssid hr authentication 802.1x auth-server rs1
```

Configuring Open Authentication with Local WEP Keys

The following examples use the following parameters for the SSID named hr:

- Open authentication
- WEP encryption
- WEP key stored locally on security device
- Key ID: 1

- Key length: 40-bit
- ASCII text: 1a2i3
- Key with ID 1 is default key

WebUI

Use the following procedure if you have a security device with two radios:

Wireless > SSID > Edit: Enter the following, then click **OK**.

WEP Based Authentication and Encryption Methods: Open, WEP Encryption

Click **WEP Key**, enter the following, then click **Add**:

Key ID: 1
Key Length: 40
Key String: Select ASCII and enter 1a2i3 (provide again in Confirm field)
Default Key (select)

CLI

Use the following command if you have a security device with two radios:

```
set ssid hr key-id 1 length 40 method asciitext 1a2i3 default
set ssid hr authentication open encryption wep
```

Configuring WEP Shared-Key Authentication

You can configure a static WEP key that is stored on the security device that is used to authenticate clients, who also have the static WEP key configured on their wireless devices. You can create up to four WEP keys per SSID.

You can specify a 40-bit encryption by providing a 5-digit hexadecimal number or a string consisting of 5 ASCII characters. Specify 104-bit encryption by providing a 26-digit hexadecimal number or a string consisting of 13 ASCII characters.

The following examples use the following parameters for the SSID named hr:

- WEP shared-key
- Key ID: 1
- Key length: 40-bit
- ASCII text: 1a2i3
- Key with ID 1 is default key

WebUI

Wireless > SSID > Edit: Enter the following, then click **OK**.

WEP Based Authentication and Encryption Methods: WEP Shared Key

Click **WEP Key**, enter the following, then click **Add**:

Key ID: 1
Key Length: 40
Key String: Select **ASCII** and enter **1a2i3** (provide again in Confirm field)
Default Key (select)

CLI

```
set ssid hr authentication shared-key
set ssid hr key-id 1 length 40 method asciitext 1a2i3 default
```

ScreenOS provides a mechanism for automatically negotiating with a wireless client whether it authenticates itself with a WEP shared key. Using this option can improve compatibility if you want to allow access to wireless clients using various operating systems that support different implementations of WEP.

To enable automatic negotiation, do one of the following:

WebUI

Wireless > SSID > Edit (for *name_str)*: Select **Auto**.

CLI

```
set ssid name_str authentication auto
```

NOTE: Although you can configure WEP for all of the SSIDs, the device intentionally restricts its use to only one interface at a time. For this reason, we recommend using WPA or WPA2.

Configuring Wi-Fi Protected Access

Wi-Fi Protected Access (WPA) is a more secure solution for WLAN authentication and encryption and was designed in response to many of the weaknesses in WEP. ScreenOS supports WPA and WPA2.

WPA and WPA2 support 802.1X authentication, which use an Extensible Authentication Protocol (EAP) method for authentication through a RADIUS server. EAP is an encapsulation protocol used for authentication and operates at the Data Link Layer (Layer 2). For more information, refer to RFC 2284, *PPP Extensible Authentication Protocol (EAP)*.

ScreenOS interoperates with 802.1X-compliant RADIUS servers, such as the Juniper Networks Steel-Belted RADIUS server and the Microsoft Internet Authentication Service (IAS) RADIUS server.

When using WPA or WPA2 with a RADIUS server, the security device forwards authentication requests and replies between the wireless clients and the RADIUS server. After successfully authenticating a client, the RADIUS server sends an encryption key to the client and the security device. From that point, the security

device manages the encryption process, including the encryption type—Temporal Key Integrity Protocol (TKIP) or Advanced Encryption Standard (AES)—and the rekey interval. For information about TKIP, see the IEEE Standard 802.11. For information about AES, see RFC 3268, *Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)*.

You can also use WPA or WPA2 with a preshared key, which is a static key that is configured on the security device and the client's device. Both devices use the key to generate a unique key (group key) for the session. You can specify the preshared key by using an ASCII passphrase (password) or in hexadecimal format. You also use the same encryption types as with 802.1X authentication: TKIP or AES.

If you want to allow WPA and WPA2 as the authentication type, you can specify the wpa-auto keyword (or WebUI option) if you are using 802.1X as the authentication method or the wpa-auto-psk keyword (or WebUI option) if you are using a preshared key as the authentication method.

Configuring 802.1X Authentication for WPA and WPA2

To configure 802.1X authentication for WPA and WPA2, you specify the following:

- RADIUS server
- Encryption type: In addition to TKIP or AES, you can specify auto, which specifies TKIP and AES as the encryption type.
- Rekey interval: Time that elapses before the group key for clients is updated. By default, the rekey interval is 1800 seconds (30 minutes). The value range is 30 through 4294967295 seconds. Use the **disable** CLI keyword or specify zero (0) in the WebUI to disable the rekey interval.

In addition to specifying WPA or WPA2 as the authentication type, you can also specify the auto option, which allows WPA and WPA2 as the authentication type.

The following examples use the following parameters for an SSID named hr:

- WPA (auto option)
- RADIUS server named rs1
- Rekey interval of 3600 seconds
- Encryption type of AES

WebUI

Wireless > SSID > (select **hr SSID**): Enter the following information, then click **OK**:

WPA Based Authentication and Encryption Methods: WPA Auto Pre-shared Key
Auth Server: rs1 (click **Create new Auth Server** to define RADIUS server if it does not already exist)
Rekey Interval: 3600
Encryption Type: AES

CLI

```
set ssid hr authentication wpa-auto rekey-interval 3600 encryption aes auth-server
rs1
```

Configuring Preshared Key Authentication for WPA and WPA2

To configure preshared key authentication for WPA and WPA2, you specify the following:

- Preshared key
  - Hexadecimal format: Specifies the key in raw format, which is a 256-bit (64 characters) hexadecimal value.
  - ASCII passphrase: Specifies a passphrase to access the SSID and consists of 8 to 63 ASCII characters.
- Encryption type: In addition to TKIP or AES, you can specify auto, which specifies TKIP and AES as the encryption type.
- Rekey interval: Time that elapses before the group key for clients is updated. By default, the rekey interval is 1800 seconds (30 minutes). The value range is 30 through 4294967295 seconds. Use the **disable** CLI keyword or specify zero (0) in the WebUI to disable the rekey interval.

In addition to specifying WPA or WPA2 as the authentication type, you can also specify the auto option, which allows WPA and WPA2 as the authentication type.

The following examples use the following parameters for an SSID named hr:

- WPA2
- Preshared key using ASCII passphrase of $FKwinnisJamesTown8fg4
- Rekey interval of 3600 seconds
- Encryption type of TKIP

WebUI

Wireless > SSID > (select **hr SSID**): Enter the following information, then click **OK**:

```
WPA Based Authentication and Encryption Methods: WPA2 Pre-shared Key
Key by Password: $FKwinnisJamesTown8fg4 (provide passphrase again in Confirm
Key by Password field)
Rekey Interval: 3600
Encryption Type: TKIP
```

CLI

```
set ssid hr authentication wpa2-psk passphrase $FKwinnisJamesTown8fg4 encryption
 tkip rekey-interval 3600
```

Specifying Antenna Use

The wireless security device allows you to choose a specific antenna or enable antenna diversity. Antenna A or antenna B, whichever has the stronger signal, is used when diversity is selected. The default setting is diversity. The diversity setting adapts in most situations. To use an external unidirectional antenna, you can specify antenna A or antenna B. For some security devices, antenna A is antenna closest to the power inlet. Antennae A and B are labeled on some security devices. See the installation and configuration guide for your device for more information.

To change the antenna setting, use one of the following procedures:

WebUI

Wireless > General Settings: Select the antenna setting from the Antenna Diversity list, then click **OK**.

CLI

To select antenna A on a security device with one radio, enter the following command:

```
set wlan antenna a
```

To select antenna A for the 5GHz radio on a security device with two radios, enter the following command:

```
set wlan 1 antenna a
```

Setting the Country Code, Channel, and Frequency

The regulatory domains used for channel assignments come preset as FCC (US), TELEC (Japan), ETSI (Europe), or WORLD (all countries). The ETSI regulatory domain is available only for security devices with two radios. You cannot change a preset regulatory domain. If the regulatory domain is preset to FCC or TELEC, you cannot select a country. If the regulatory domain is WORLD or ETSI, you must select a country. If you do not set a country for a device preset to WORLD or ETSI, a warning message appears and wireless capabilities will not function.

The wireless security device uses the same channel and frequency for all service set identifiers (SSIDs) in one radio transceiver. The device automatically selects the appropriate channel based on the country code that you enter (unless you manually selected a specific channel). The channel in use appears in the channel list in the WebUI. The device can select the channel if you leave the setting at **auto**.

For the list of available country codes, channels, and frequencies, see *"Wireless Information" on page 2267*.

To configure the country code and channel:

WebUI

Wireless > General Settings: Select the country, channel, and frequency from the drop-down lists, then click **Apply** .

CLI

```
set wlan country-code country_abbreviation
set wlan { 0 | 1 } channel { auto | 1 | 2 | 3 | ... }
```

Using Extended Channels

If the security device is located in a regulatory domain that allows the use of channels 12 and 13, you can enable the 2.4 GHz radio transceiver to use them.

WebUI

Wireless > General Settings: Select the Extended Channel Mode check box.

CLI

For a security device with one radio, enter the following command:

```
set wlan extended-channel
```

For a security device with two radios, enter the following command:

```
set wlan 0 extended-channel
```

Performing a Site Survey

You can scan the broadcast vicinity to see if there are any other access points broadcasting nearby. Running a site survey allows you to see if there are any rogue access points in the area. A site survey detects any access points emitting a beacon in the area and records the following details about each detected access point:

- Service Set Identifier (SSID)
- MAC address
- Received signal strength indicator (RSSI)

  The RSSI numbers are measured in decibels (dBs), which indicate the signal-to-noise ratio (SNR). The SNR is the signal level divided by the noise level, which results in a value representing signal strength.
- Broadcast channel

In addition to performing an initial site survey, you might want to perform surveys occasionally to ensure that no rogue access points are in the area. To perform a site survey:

WebUI

Wireless > Statistics > Site Survey

CLI

```
exec wlan site-survey
```

NOTE: Depending on your network, a site survey can take up to 60 seconds to complete and disrupts wireless network traffic.

Locating Available Channels

Using the CLI, you can find the best radio channel for the device to use for transmission. Use this command if you do not want to use the default setting that automatically select channels and want to find the channel with the least interference.

To find the best channel available, use the following command:

```
exec wlan find-channel
```

NOTE: This feature is not available from the WebUI.

Setting an Access Control List Entry

You can control which wireless clients have access to the network through an access control list (ACL). The ACL identifies clients by their MAC addresses and specifies whether the wireless device allows or denies access for each address. The ACL can operate in one of three access modes:

- Disabled: The wireless device does not filter any MAC addresses. This is the default mode.
- Enabled: The wireless device allows access to all clients except those marked with a Deny action.
- Strict: The wireless device denies access to all clients except those marked with an Allow action.

NOTE: The ACL settings apply globally to all SSIDs.

You can define up to 64 denied clients and 64 allowed clients.

To add a MAC address to the ACL, use one of the following procedures:

WebUI

Wireless > MAC Access List: Enter the following, then click **Add**:

Access Mode: (select one of the three modes from the list)
Input a new MAC address: (type the MAC address of a wireless client)
Control Status: (select either **Allow** or **Deny**)

In the WebUI, you can also select a MAC address from the Select a learned MAC address list. Entries appear in this list when a wireless client makes an association with the wireless device. The list is a dynamic display of all currently associated wireless clients, regardless of the SSID to which they belong.

NOTE: You can also set the access mode through the MAC Address Access Control list on the Wireless > General Settings page.

CLI

```
set wlan acl mode { disable | enable | strict }
set wlan acl mac_addr { deny | allow }
```

Configuring Super G

In wireless devices that have an Atheros Communications chipset with Super G® feature, you can enable Super G, which can increase user data throughput rate up to 4 Mbps for 802.11a and 802.11g clients by using the following methods:

- Bursting: Allows the device to transmit multiple frames in a burst rather than pausing after each frame.
- Fast frames: Allows for more information per frame to be transmitted by allowing a larger-than-standard frame size.
- Compression: Link-level hardware compression is performed by a built-in data compression engine.

By default, this feature is disabled.

If wireless clients do not support Super G and the security device has Super G enabled, they can still connect to the wireless network, but the Super G feature is not available.

NOTE: You can read more about Atheros Communications Super G chipset at www.atheros.com.

To enable Super G, use one of the following procedures:

WebUI

Wireless > General Settings: Select the Super-G check box (if the security device has more than one radio, make the selection for the radio you want).

CLI

To enable Super G on a security device with one radio, enter the following command:

```
set wlan super-g
```

To enable Super G for the 5GHz radio on a security device with two radios, enter the following command:

```
set wlan 1 super-g
```

Configuring Atheros XR (Extended Range)

You can enable Atheros Communications eXtended Range (XR) technology. XR processes 802.11 signals, defined by IEEE 802.11a and 802.11g standards, so that wireless networks to have fewer “dead spots” and greater range than usual. XR processes weaker signals more effectively and allows greater coverage. XR provides increased coverage at a lower data transmission rate.

Only the first active SSID per radio can support XR. When XR is enabled, the first active SSID per radio uses the XR feature.

To enable XR, use one of the following procedures:

WebUI

Wireless > General Settings: Select the XR Support check box (if the security device has more than one radio, make the selection for the radio you want).

CLI

To enable XR on a security device with one radio, enter the following command:

```
set wlan xr
```

To enable XR for the 5GHz radio on a security device with two radios, enter the following command:

```
set wlan 1 xr
```

Configuring Wi-Fi Multimedia Quality of Service

Wi-Fi™ Multimedia (WMM) quality of service (QoS) feature enables you to enhance the performance of your wireless network by adjusting the transmission priorities

of audio, video, and voice applications to accommodate the different latency and throughput requirements of each application. By default, WMM is disabled.

WMM is based on Enhanced Distributed Channel Access (EDCA) as defined in 802.11e. For more information about WMM, see http://www.wi-fi.org.

This feature is not available for all security devices.

Enabling WMM

You can enable WMM on the 2.4 GHz radio (WLAN 0) or 5GHz radio (WLAN 1).

To enable WMM, use one of the following procedures:

WebUI

Wireless > WMM Settings: Select the Enable radio button for the radio, then click **Apply**.

CLI

```
set wlan [ 0 | 1 ] wmm enable
```

Configuring WMM Quality of Service

After you enable WMM, you can configure WMM parameters accommodate your network requirements. You configure WMM to operate from each end of the connection: access point (ap) and station (sta).

- *ap* is the WMM configuration for the security device.
- *sta* is the WMM configuration for the client. Clients internally queue traffic according to the four ACs and then send packets as they detect transmit opportunities based on the parameters you set.

The WMM settings are used only when the security device or clients (stations) send a packet.

Access Categories

Based on Internet Engineering Task Force (IETF) Differentiated Services Code Point (DSCP) headers, the security device and client sort traffic into one of four access categories (AC):

- Best effort (0)—traffic that cannot process QoS levels and traffic that is less sensitive to latency but that can be affected by long delays.
- Background priority (1)—low-priority traffic
- Video (2)—video traffic gets a higher priority than other data traffic
- Voice (3)—voice traffic gets the highest priority

Table 134 on page 2022 lists the mappings between access categories and Type of Service (TOS).

Table 134: Access Category and TOS Mappings

Access Category	TOS Value
Voice	0xC0, 0xB8, 0xE0
Video	0x80, 0xA0, 0x88
Best effort	0x00, 0x60, or other
Background	0x40, 0x20

NOTE: 802.1d tags are not supported.

Although specific priorities and settings are associated with each AC, you can override these settings through the WebUI or the CLI.

WMM Default Settings

The following terms describe the configurable WMM parameters and appear as column headings in Table 135 on page 2023 and Table 136 on page 2024, which list the default settings for access point (security device) and station (client) configuration:

- *aifs*

  Arbitrary Inter-Frame Space Number (AIFSN) specifies the number of slots, after a SIFS duration, that the security device or client for an AC will check the medium-idle before transmitting or executing a backoff.

- *logcwmin* and *logcwmax*

  WMM defines a Contention Window (CW), which is equivalent to a random backoff period.

  The CWmin parameter specifies the minimum number of slots of the contention window used by the security device or client for a particular AC to generate a random number for the backoff. If logcwmin is x, then CWmin is 2x-1.

  The CWmax parameter specifies the maximum number of slots of the window used by the security device or client for a particular AC to generate a random number for the backoff. If logcwmax is x, then CWmax is 2x-1.

  ScreenOS does not support contention-free or scheduled access.

- *txoplimit*

Transmit Opportunity specifies the maximum amount of time the security device or client can initiate transmissions. If you set txoplimit to x, the maximum time is 32*x microseconds.

- *ackpolicy*

  You can enable or disable an acknowledgement policy for the access point. This parameter does not apply to clients.

Table 135 on page 2023 lists the default values for all supported wireless modes for a security device in application type WMM. By default, *ackpolicy* is disabled for all wireless modes.

Table 135: Access Point WMM Default Values Organized by AC

AC	Wireless Mode	aifs	logcwmin	logcwmax	txoplimit
Best Effort (0)	802.11a	3	4	6	0
	802.11a Turbo	2	3	5	0
	802.11b	3	5	7	0
	802.11g	3	4	6	0
	802.11g Turbo	2	3	5	0
	XR	0	3	3	0
Background (1)	802.11a	7	4	10	0
	802.11a Turbo	7	3	10	0
	802.11b	7	5	10	0
	802.11g	7	4	10	0
	802.11g Turbo	7	4	10	0
	XR	0	3	3	0
Video (2)	802.11a	1	3	4	94
	802.11a Turbo	1	2	3	94
	802.11b	1	4	5	188
	802.11g	1	3	4	94
	802.11g Turbo	1	2	3	94
	XR	0	3	3	0
Voice (3)	802.11a	1	2	3	47

Table 135: Access Point WMM Default Values Organized by AC *(continued)*

AC	Wireless Mode	aifs	logcwmin	logcwmax	txoplimit
	802.11a Turbo	1	2	2	47
	802.11b	1	3	4	102
	802.11g	1	2	3	47
	802.11g Turbo	1	2	2	47
	XR	0	3	3	0

Table 136 on page 2024 lists the default values for all supported wireless modes for the WMM configuration for a client (sta).

Table 136: Station WMM Default Values Organized by AC

AC	Wireless Mode	aifs	logcwmin	logcwmax	txoplimit
Best Effort	802.11a	3	4	10	0
	802.11a Turbo	2	3	10	0
	802.11b	3	5	10	0
	802.11g	3	4	10	0
	802.11g Turbo	2	3	10	0
	XR	0	3	3	0
Background	802.11a	7	4	10	0
	802.11a Turbo	7	3	10	0
	802.11b	7	5	10	0
	802.11g	7	4	10	0
	802.11g Turbo	7	4	10	0
	XR	0	3	3	0
Video	802.11a	2	3	4	94
	802.11a Turbo	2	2	3	94
	802.11b	2	4	5	188
	802.11g	2	3	4	94
	802.11g Turbo	2	2	3	94

Table 136: Station WMM Default Values Organized by AC *(continued)*

AC	Wireless Mode	aifs	logcwmin	logcwmax	txoplimit
	XR	0	3	3	0
Voice	802.11a	2	2	3	47
	802.11a Turbo	1	2	2	47
	802.11b	2	3	4	102
	802.11g	2	2	3	47
	802.11g Turbo	1	2	2	47
	XR	0	3	3	0

Example

In the following example, you use the station configuration of WMM on the 5GHz transceiver and change the settings for voice traffic (A = 0) as follows:

- logcwmin: zero (0)
- logcwmax: 15
- aifs: 4
- txoplimit: 10

To configure WMM with these settings:

WebUI

Wireless > WMM Settings: Enter the desired settings, then click **Apply**.

CLI

```
set wlan 1 wmm sta 0 logcwmin 0
set wlan 1 wmm sta logcwmax 15
set wlan 1 aifs 4
set wlan 1 txoplimit 10
save
```

Configuring Advanced Wireless Parameters

This section contains information about advanced wireless parameters. You might need to make small changes to increase performance in certain type of wireless deployments.

The following advanced wireless features are discussed in this section:

Configuring Aging Interval

You can specify the amount of time that elapses before a wireless client is disconnected if there is no traffic to or from the client. This value can be between 60 seconds and 1,000,000 seconds. The default value is 300 seconds. To disable aging, use the **set wlan advanced aging-interval disable** command.

After the aging-interval elapses and a client is disconnected, its MAC information is deleted from a MAC table on the security device. The MAC table for each radio can contain up to 60 client MAC addresses. Because new clients are denied connectivity when the MAC table is full, set the aging-interval so that existing clients whose connections are not being used are disconnected and their MAC addresses are removed from the MAC table in a timely manner.

To set the aging interval to 500 seconds for the 2.4 GHz radio, use one of the following procedures:

WebUI

Wireless > General Settings > Advanced: Enter the following, then click **Return**:

Aging Interval: 500 (for devices with two radios, specify the aging interval for WLAN0)

CLI

To change the aging interval for a security device with one radio, enter the following command:

```
set wlan advanced aging-interval 500
```

To change the aging interval to 500 seconds for the 2.4 GHz radio on a security device with two radios, enter the following command:

```
set wlan 0 advanced aging-interval 500
```

Configuring Beacon Interval

You can configure the interval at which beacons are sent. The value range is 20 to 1,000 time units (1 time unit equals 1024 µs) The default value is 100 time units.

To set the beacon interval to 200 time units (2048 µs) for the 2.4 GHz transceiver, use one of the following procedures:

WebUI

Wireless > General Settings > Advanced: Enter the following, then click **Return**:

Beacon Interval: 200 (for devices with two radios, specify the beacon interval for WLAN0)

CLI

To change the beacon interval for a security device with one radio, enter the following command:

```
set wlan advanced beacon-interval 200
```

To change the beacon interval for the 2.4 GHz radio on a security device with two radios, enter the following command:

```
set wlan 0 advanced beacon-interval 200
```

Configuring Delivery Traffic Indication Message Period

You can set the number of beacons that are sent before the delivery traffic indication map (DTIM) is sent. Increasing the DTIM period decreases the number of broadcasts sent to clients. The value range is 1 to 255. The default value is 1 beacon interval.

To set the DTIM period to 2, use one of the following procedures:

WebUI

Wireless > General Settings > Advanced: Enter the following, then click **Return**:

DTIM Period: 2 (for devices with two radios, specify the DTIM period for WLAN0)

CLI

To change the DTIM period for a security device with one radio, enter the following command:

```
set wlan advanced dtim-period 2
```

To change the DTIM period for the 2.4 GHz radio on a security device with two radios, enter the following command:

```
set wlan 0 advanced dtim-period 2
```

Configuring Burst Threshold

You can set a maximum number of frames in a burst. The valid range is between 2 and 255. The default value is 3. This feature is not available on all security devices.

To change the burst threshold to 5, use one of the following procedures:

WebUI

Wireless > General Settings > Advanced: Enter the following, then click **Return**:

Burst Threshold: 5

CLI

```
set wlan advanced burst-threshold 5
```

Configuring Fragment Threshold

You can set the maximum length of a frame before it is fragmented into multiple frames before transmission. Value range is even numbers between 256 and 2346. The default value is 2346.

To set the fragment threshold to 500 for the 2.4 GHz radio, use one of the following procedures:

WebUI

Wireless > General Settings > Advanced: Enter the following, then click **Return**:

Fragment Threshold: 500 (for devices with two radios, specify the fragment threshold for WLAN0)

CLI

To change the fragment threshold for a security device with one radio, enter the following command:

```
set wlan advanced fragment-threshold 500
```

To set the fragment threshold for the 2.4 GHz radio on a security device with two radios, enter the following command:

```
set wlan 0 advanced fragment-threshold 500
```

Configuring Request to Send Threshold

You can set the maximum length a frame is before using the Request to Send (RTS) method to send the frame. The value range is 256 to 2346. The default value is 2346.

To set the RTS threshold to 500 for the 2.4 GHz radio, use one of the following procedures:

WebUI

Wireless > General Settings > Advanced: Enter the following, then click **Return**:

RTS Threshold: 500 (for devices with two radios, specify the RTS threshold for WLAN0)

CLI

To change the RTS threshold for a security device with one radio, enter the following command:

```
set wlan advanced rts-threshold 500
```

To set the RTS threshold of the 2.4 GHz radio on a security device with two radios, enter the following command:

```
set wlan 0 advanced rts-threshold 500
```

Configuring Clear to Send Mode

Clear to Send (CTS) protection blocks acknowledgement (ACK) packets to reduce some of the overhead required to run 802.11. The default setting is *auto*. By default the security device detects the CTS setting of clients. You can also select **on** to always use CTS or **off** to never use CTS.

NOTE: This feature does not work in 802.11b wireless mode and is not available on all security devices.

When modifying default behavior of the security device, you might also have to modify the CTS rate and type, as described in “Configuring Clear to Send Rate” on page 2030 and “Configuring Clear to Send Type” on page 2030

To turn off CTS protection, use one of the following procedures:

WebUI

Wireless > General Settings > Advanced: Select the following, then click **Return**:

CTS Mode: Off

CLI

```
set wlan advanced cts-mode off
```

Configuring Clear to Send Rate

You can set the rate (in Mbps) at which CTS frames are sent. This feature does not work in 802.11b wireless mode and is not available on all security devices. Valid values are 1, 2, 5.5, and 11 Mbps. The default is 11 Mbps.

To set the CTS rate to 5.5, use one of the following procedures:

WebUI

Wireless > General Settings > Advanced: Select the following, then click **Return**:

CTS Rate: 5.5

CLI

```
set wlan advanced cts-rate 5.5
```

Configuring Clear to Send Type

ScreenOS provides two Clear to Send (CTS) protection types: CTS-only and CTS-RTS. The purpose of CTS is to decrease collisions between two wireless clients. The CTS-only option (default) forces the security device to wait for a CTS frame before forwarding any data. The CTS-RTS (Request to Send) option forces the security device to complete a RTS -CTS handshake before forwarding data.

This feature is not available on all security devices.

To set the CTS type to CTS-only, use one of the following procedures:

WebUI

Wireless > General Settings > Advanced: Select the following, then click **Return**:

CTS Type: CTS Only

CLI

```
set wlan advanced cts-type cts-only
```

Configuring Slot Time

When the slot time is set to **long**, the security device uses only long slot time. By default, the security devices uses short slot time. This feature is used only in 802.11g mode.

To enable long slot time for the 2.4 GHz radio, use one of the following procedures:

WebUI

Wireless > General Settings > Advanced: Select the Long Slot Time check box, then click **Return**.

CLI

To enable long slot time for a security device with one radio, enter the following command:

```
set wlan advanced slot-time long
```

To enable long slot time for the 2.4 GHz radio on a security device with two radios, enter the following command:

```
set wlan 0 advanced slot-time long
```

Configuring Preamble Length

You can modify the transmit preamble from short to long. When set to long, only long preambles are used. When short is enabled, both short and long preambles are used. The default is short. This command only applies when setting the 2.4 GHz transceiver for 802.11b and 802.11g modes.

WebUI

Wireless > General Settings > Advanced: Select the Long Transmit Preamble check box, then click **Return**.

CLI

To enable long preambles for a security device with one radio, enter the following command:

```
set wlan advanced long-preamble
```

To enable long preambles for the 2.4 GHz radio on a security device with two radios, enter the following command:

```
set wlan 0 advanced long-preamble
```

Working with Wireless Interfaces

This section describes the following tasks you can perform with wireless interfaces.

Binding an SSID to a Wireless Interface

When the security device initially boots, several wireless interfaces exist, but they are not associated with SSIDs. After creating an SSID, you need to bind the interface to an SSID to activate the interface.

For security devices with one radio, after you create an SSID, you must bind it to a wireless interface that is bound to a specific security zone. For security devices with two radios, wireless interfaces can be bound to a zone or placed in a bridge group (bgroup). For more information about bridge groups, see "Creating Wireless Bridge Groups" on page 2033.

To bind an SSID to a wireless interface, use one of the following procedures:

WebUI

Wireless > SSID > Edit (for the SSID that you want to bind to an interface): Select an interface from the Wireless Interface Binding list, then click **OK**.

or

For security devices with one radio, do the following:

Network > Interfaces > Edit: Select the SSID from the Bind to SSID list, then click **OK**.

For security devices with two radios, do the following:

Network > Interfaces > List > Edit: Select the SSID from the Bind to SSID list, then click **OK**.

CLI

set ssid *name_str* interface *interface*

Binding a Wireless Interface to a Radio

On some security devices, you can specify the radio a wireless interface uses. By default, wireless interfaces are bound to two radios and can run in 802.11a, 802.11b, or 802.11g modes.

You can enable the following options:

- 0, which enables only the 2.4 GHz transceiver for 802.11b and 802.11g.
- 1, which enables only the 5GHz transceiver for 802.11a.
- both, which enables both transceivers (2.4 GHz and 5GHz) on the interface (802.11a/b/g).

For example, if you specify **802.11b** as the operation mode with the **set wlan mode** command and select the **0** option, only 802.11b is available for the wireless interface.

To specify that a wireless0/0 interface use the 2.4 GHz radio, use one of the following procedures:

WebUI

Network > Interface > List > Edit (for the wireless interface): Select **2.4G(802.11b/g)** from the Wlan list, then click **OK**.

CLI

```
set interface wireless0/0 wlan 0
```

Creating Wireless Bridge Groups

Some security devices support bridge groups (bgroups). A bgroup allows network users to switch between wired/wireless traffic without having to reconfigure or reboot their computer. You can configure multiple SSIDs to operate one bgroup or configure an SSID to operate in the same subnet as the wired subnet.

On wireless security devices that support bridge groups, the bridge groups are identified as bgroup0 through bgroup3.

To set an Ethernet and wireless interface to the same bgroup, use one of the following procedures:

WebUI

Network > Interfaces > List > Edit (for bgroup) > Bind Port: Select **Bind to Current Bgroup** for the interface, then click **Apply**.

CLI

To set an Ethernet and wireless interface to the same bridge group interface, do the following:

```
set interface bgroup_name port wireless_interface
set interface bgroup_name port ethernet_interface
```

NOTE: *bgroup_name* can be bgroup0—bgroup3.
ethernet_interface can be ethernet0/0—ethernet0/4.
wireless_interface can be wireless0/0—wireless0/3.

Disabling a Wireless Interface

You can disable a particular wireless interface from the WebUI or the CLI. By default, when you bind an interface to an SSID, it is activated. This feature is not available on all platforms.

To disable a wireless interface, use one of the following procedures:

WebUI

Network > Interfaces > List > Deactivate (for wireless interface to be disabled)

CLI

set interface *wlan_if_name* shutdown

Viewing Wireless Configuration Information

You can view details of wireless configurations and statistics. These command are available from the CLI and not the WebUI.

To view a WLAN configuration, enter the following command:

```
device-> get wlan
AP software version: 5.0
AP bootrom version: 1..1
Regulatory Domain is World, Country Code is China
ACL mode is disabled


WLAN   Mode  Antenna  Channel  Rate Power  SuperG
0   b/g  Diversity AUTO   11  full  Enabled
1   a   a    116 (5580) 54  full  Disabled
```

To view wireless interface association information, enter the following command:

get interface *wlan_if_name* association [*mac_addr*]

To view the access control list (ACL) for a WLAN, enter the following command:

get wlan acl

To view interface details for a WLAN interface, enter the following command:

get interface *wlan_if_name*

Configuration Examples

This section contains configurations for the following examples:

- Example 1: Open Authentication and WEP Encryption on page 2034
- Example 2: WPA-PSK Authentication with Passphrase and Automatic Encryption on page 2035
- Example 3: WLAN in Transparent Mode on page 2036
- Example 4: Multiple and Differentiated Profiles on page 2040

Example 1: Open Authentication and WEP Encryption

In this example for a security device with one radio, you create a BSS with the SSID named **openwep,** which you then bind to the wireless2 interface. This configuration sets the WEP key-id to 1 with an input ASCII string of 40-bits. This configuration allows anyone to authenticate but encrypts communication using the WEP key.

WebUI

Wireless > SSID > New: Enter the following, then click **OK**:

SSID: openwep

> WEP Key: Enter the following, then click **Back to SSID Edit**:

Key ID: 1
Key Length: 40
Key String
ASCII: (select), abcde
Add: (select)

> WEP Based Authentication and Encryption Methods

Open: (select)
WEP Encryption: (select); Key Source: Local
Wireless Interface Binding: wireless2

Wireless > Activate Changes: Click the Activate Changes button.

CLI

```
set ssid name openwep
set ssid openwep key-id 1 length 40 method ascii abcde
set ssid openwep authentication open encryption wep
set ssid openwep interface wireless2
exec wlan reactivate
```

Example 2: WPA-PSK Authentication with Passphrase and Automatic Encryption

In this example for a security device with one radio, you create an SSID named **wpapsk**, which you then bind to the wireless2 interface. The configuration sets Wi-Fi Protected Access (WPA) authentication with a preshared key and automatic encryption. Wireless clients who want to connect to the wireless2 interface to access the network must use the WPA passphrase i7BB92-5o23iJ when establishing a wireless connection.

WebUI

Wireless > SSID > New: Enter the following, then click **OK**:

> WPA Based Authentication Methods

WPA Pre-shared Key: (select)
Key by Password: (select), i7BB92-5o23iJ
Confirm key by Password: i7BB92-5o23iJ
Encryption Type: Auto
Wireless Interface Binding: wireless2

Wireless > Activate Changes: Click the Activate Changes button.

CLI

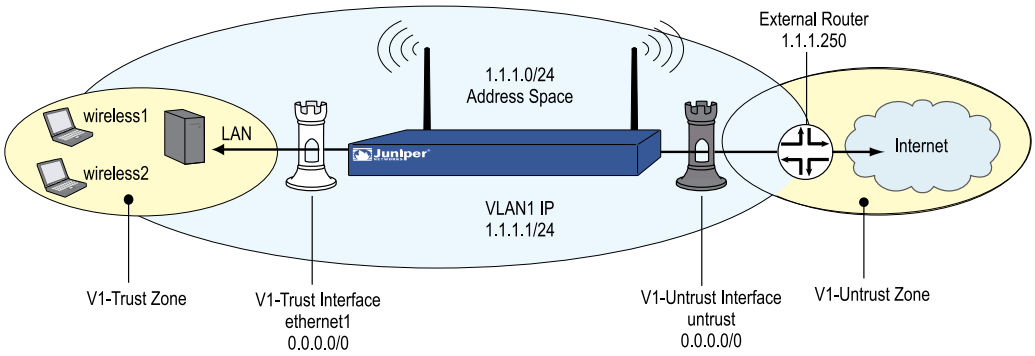
```
set ssid name wpapsk
set ssid wpapsk authentication wpa-psk passphrase i7BB92-5o23iJ encryption auto
set ssid wpapsk interface wireless2
exec wlan reactivate
```

Example 3: WLAN in Transparent Mode

In this example, a single WLAN is protected by a security device with one radio in transparent mode. To increase the security of management traffic, do the following

1. Change the HTTP port number for WebUI management from 80 to 5555, and the Telnet port number for CLI management from 23 to 4646.
2. Use the VLAN1 IP address—1.1.1.1/24—to manage the security device from the V1-Trust security zone.
3. Configure a default route to the external router at 1.1.1.250, so that the security device can send outbound VPN traffic to it. (The default gateway on all hosts in the V1-Trust zone is also 1.1.1.250.)

Figure 487: WLAN Device in Transparent Mode



WebUI

1. **VLAN1 Interface**

   Network > Interfaces > Edit (for the VLAN1 interface): Enter the following, then click **OK**:

   ```
   IP Address/Netmask: 1.1.1.1/24
   Management Services: WebUI, Telnet (select)
   Other Services: Ping (select)
   ```

2. **Port Mode**

   Configuration > Port Mode: Select **Trust-Untrust** in the Port Mode drop-down list.

3. **HTTP Port**

   Configuration > Admin > Management: In the HTTP Port field, type **5555**, then click **Apply**.

NOTE: The default port number is 80. Changing this to any number between 1024 and 32,767 is advised for discouraging unauthorized access to the configuration. When logging in to manage the device later, enter the following in the URL field of your browser: http://1.1.1.1:5555.

4. **Interfaces**

   Network > Interfaces > Edit (for trust): Enter the following, then click **OK**:

   Zone Name: V1-Trust
   IP Address/Netmask: 0.0.0.0/0

   Network > Interfaces > Edit (for untrust): Enter the following, then click **OK**:

   Zone Name: V1-Untrust
   IP Address/Netmask: 0.0.0.0/0

   Network > Interfaces > Edit (for wireless1): Enter the following, then click **OK**:

   Zone Name: V1-Trust
   IP Address/Netmask: 0.0.0.0/0

   Network > Interfaces > Edit (for wireless2): Enter the following, then click **OK**:

   Zone Name: V1-Trust
   IP Address/Netmask: 0.0.0.0/0

5. **Zones**

   The default virtual router for the V1-Trust, V1-Untrust, and VLAN zones is trust-vr.

6. **SSIDs**

   Wireless > SSID > New: Enter the following, then click **OK**:

   SSID: xparent-wpa

   > WPA Based Authentication Methods

   WPA Pre-shared Key: (select)
   Key by Password: (select), 12345678
   Confirm key by Password: 12345678
   Encryption Type: TKIP
   Wireless Interface Binding: wireless2

   Wireless > SSID > New: Enter the following, then click **OK**:

   SSID: xparent-share

> WEP Based Authentication and Encryption Methods

> WEP Key: Enter the following, then click **Back to SSID Edit**:

Key ID: 1
Key Length: 40
Key String
ASCII: (select), abcde
Default Key: (select)
Add: (select)

> WEP Based Authentication and Encryption Methods

WEP Shared Key: (select)
Wireless Interface Binding: wireless1

7. **Route**

Network > Routing > Destination > trust-vr New: Enter the following, then click **OK**:

Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: vlan1(trust-vr)
Gateway IP Address: 1.1.1.250
Metric: 1

8. **Policies**

Policies > (From: V1-Trust, To: V1-Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: Any
Action: Permit

9. **WLAN Configuration Activation**

Wireless > Activate Changes: Click the Activate Changes button.

CLI

1. **VLAN1**

```
set interface vlan1 ip 1.1.1.1/24
set interface vlan1 manage web
set interface vlan1 manage telnet
set interface vlan1 manage ping
```

2. **Port Modes**

```
exec port-mode trust-untrust
```

3. **HTTP Port**

```
set admin telnet port 4646
```

NOTE: The default port number for Telnet is 23. Changing this to any number between 1024 and 32,767 is advised for discouraging unauthorized access to the configuration. When logging in to manage the device later through Telnet, enter the following address: 1.1.1.1 4646.

4. **Interfaces**

```
set interface trust ip 0.0.0.0/0
set interface trust zone v1-trust
set interface untrust ip 0.0.0.0/0
set interface untrust zone v1-untrust
set interface wireless1 ip 0.0.0.0/0
set interface wireless1 zone v1-trust
set interface wireless2 ip 0.0.0.0/0
set interface wireless2 zone v1-trust
```

5. **Zones**

```
set zone V1-Trust vrouter trust-vr
set zone V1-Untrust vrouter trust-vr
set zone VLAN vrouter trust-vr
```

6. **SSID**

```
set ssid name xparent-wpa
set ssid xparent-wpa authentication wpa-psk passphrase 12345678 encryption
 tkip
set ssid xparent-wpa interface wireless2
set ssid name xparent-share
set ssid xparent-share key-id 1 length 40 method asciitext abcde default
set ssid xparent-share authentication shared-key
set ssid xparent-share interface wireless1
exec wlan reactivate
```

7. **Route**

```
set vrouter trust-vr route 0.0.0.0/0 interface vlan1 gateway 1.1.1.250 metric 1
```

8. **Policies**

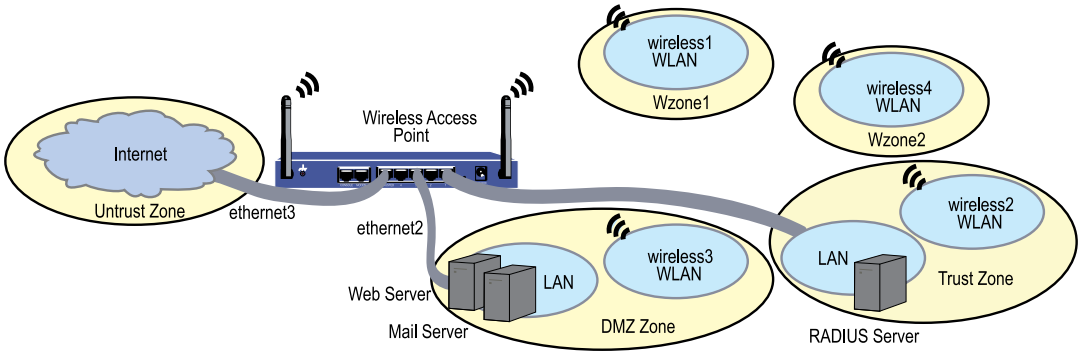
```
set policy from v1-trust to v1-untrust any any any permit
save
```

Example 4: Multiple and Differentiated Profiles

In this example, you create four SSIDs, each with a different name and authentication and encryption scheme for a wireless security device with one radio in Extended port mode. This mode provides the following port, interface, and zone bindings:

Interface	Security Zones	Basic Service Sets Names
ethernet1 (ports 1 and 2)	Trust	NA
ethernet2 (ports 3 and 4)	DMZ	NA
ethernet3 (Untrust port)	Untrust	NA
wireless1	Wzone1	SSID: wzone1-wpa with WPA preshared-key
wireless2	Trust	SSID: trust-wpa with WPA using RADIUS server
wireless3	DMZ	SSID: dmz-share with WEP shared key
wireless4	Wzone2	SSID: wzone2-open with WEP open/no encryption

Figure 488: Wireless with Multiple and Differentiated Profiles



In this example, you do the following:

1. Set your Basic Service Sets (BSS) by assigning SSID names, setting the encryption and authentication methods, and binding the SSID to a wireless interface.
2. Set each wireless interface to act as a DHCP server to assign addresses dynamically to the wireless clients for each SSID.
3. Enable wireless device management on the wireless1 interface.
4. Configure the wireless device to use a RADIUS server for WPA encryption.
5. Create policies for each wireless interface.
6. Reactivate the WLAN.

You can configure this example with either the WebUI or CLI:

WebUI

1. **Setting the Basic Service Sets**

   Wireless > SSID > New: Enter the following, then click **OK**:

   SSID: wzone1-wpa

   > WPA Based Authentication Methods

   WPA Pre-shared Key: (select)
   Key by Password: (select), 12345678
   Confirm key by Password: 12345678
   Encryption Type: Auto
   Wireless Interface Binding: wireless1

   Wireless > SSID > New: Enter the following, then click **OK**:

   SSID: trust-wpa

   > WPA Based Authentication Methods

   WPA: (select)
   Encryption Type: Auto

   Wireless > SSID > New: Enter the following, then click **OK**:

   SSID: dmz-share

   > WEP Based Authentication and Encryption Methods

   WEP Key: Enter the following, then click **Back to SSID Edit**:
   Key ID: 1
   Key Length: 40
   Key String
   ASCII: (select), abcde
   Add: (select)
   WEP Shared Key: (select)
   Wireless Interface Binding: wireless3

   Wireless > SSID > New: Enter the following, then click **OK**:

   > WEP Based Authentication and Encryption Methods

   Open: (select)
   No Encryption: (select)
   Wireless Interface Binding: wireless4

2. **Interfaces**

   Network > Interfaces > Edit (for ethernet3): Enter the following, then click **OK**:

Obtain IP using DHCP: (select)
Automatic update DHCP server parameters: (select)

Network > Interfaces > Edit (for wireless1): Enter the following, then click **OK**:

IP Address/Netmask: 192.168.5.1/24

> Management Options

Management Services: WebUI, Telent, SSH, SNMP, SSL
Other Services: Ping

Network > DHCP > Edit (for wireless1) > DHCP Server: Enter the following, then click **OK**:

DHCP Server: (select)
DHCP Server Mode: Enable
Lease: Unlimited
DNS#1: 192.168.5.30

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 192.168.5.2
IP Address End: 192.168.5.22

NOTE: By default, device management is enabled for wireless2 with the default IP address of 192.168.2.1/24.

Network > DHCP > Edit (for wireless2) > DHCP Server: Enter the following, then click **OK**:

DHCP Server: (select)
DHCP Server Mode: Enable
Lease: Unlimited

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 192.168.2.2
IP Address End: 192.168.2.22

Network > Interfaces > Edit (for wireless3): Enter **192.168.3.1/24** in the IP Address/Netmask fields, then click **OK**.

Network > DHCP > Edit (for wireless3) > DHCP Server: Enter the following, then click **OK**:

DHCP Server: (select)
DHCP Server Mode: Enable

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 192.168.3.2
IP Address End: 192.168.3.22

Network > Interfaces > Edit (for wireless4): Enter **192.168.4.1/24** in the IP Address/Netmask fields, then click **OK**.

Network > DHCP > Edit (for wireless4) > DHCP Server: Enter the following, then click **OK**:

DHCP Server: (select)
DHCP Server Mode: Enable
Lease: Unlimited

> Addresses > New: Enter the following, then click **OK**:

Dynamic: (select)
IP Address Start: 192.168.4.2
IP Address End: 192.168.4.22

3. **RADIUS Auth Server**

Configuration > Auth > Servers > New: Enter the following, then click **OK**:

Name: radius1
IP/Domain Name: 192.168.1.50
Backup1: 192.168.1.60
Backup2: 192.168.1.61
Timeout: 30
Account Type: 802.1X
RADIUS: (select)
Shared Secret: 456htYY97kl

4. **Policies**

Policies > (From: Wzone1, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: ANY
Action: Permit

Policies > (From: Wzone1, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: ANY
Action: Permit

Policies > (From: Wzone2, To: Untrust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: ANY
Action: Permit

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: HTTP
Action: Permit

Policies > (From: Untrust, To: DMZ) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: MAIL
Action: Permit

5. **WLAN Configuration Activation**

Wireless > Activate Changes: Click the Activate Changes button.

CLI

1. **Basic Service Sets**

```
set ssid name wzone1-wpa
set ssid wzone1-wpa authentication wpa-psk passphrase 12345678 encryption
 auto
set ssid wzone1-wpa interface wireless1

set ssid name trust-wpa
set ssid trust-wpa authentication wpa encryption auto
set ssid trust-wpa interface wireless2

set ssid name dmz-share
set ssid dmz-share key-id 1 length 40 method ascii abcde
set ssid dmz-share authentication shared-key
set ssid dmz-share interface wireless3

set ssid name wzone2-open
set ssid wzone2-open authentication open encryption none
set ssid wzone2-open interface wireless4
```

2. **Interfaces**

```
set interface ethernet3 dhcp client settings update-dhcp server
set interface ethernet3 dhcp client
```

```
set interface wireless1 ip 192.168.5.1/24
set interface wireless1 route
set interface wireless1 ip manageable
set interface wireless1 dhcp server service
set interface wireless1 dhcp server enable
set interface wireless1 dhcp server option gateway 192.168.5.1
set interface wireless1 dhcp server option netmask 255.255.255.0
set interface wireless1 dhcp server option dns1 192.168.5.30

set interface wireless1 dhcp server ip 192.168.5.2 to 192.168.5.22

set interface wireless2 dhcp server ip 192.168.2.2 to 192.168.2.22

set interface wireless3 ip 192.168.3.1/24
set interface wireless3 dhcp server ip 192.168.3.2 to 192.168.3.22

set interface wireless4 ip 192.168.4.1/24
set interface wireless4 dhcp server ip 192.168.4.2 to 192.168.4.22
```

3. **RADIUS Auth Server**

```
set auth-server radius1 server-name 192.168.1.50
set auth-server radius1 type radius
set auth-server radius1 account-type 802.1X
set auth-server radius1 backup1 192.168.1.60
set auth-server radius1 backup2 192.168.1.61
set auth-server radius1 timeout 30
set auth-server radius1 radius secret A56htYY97kl
```

4. **Policies**

```
set policy from wzone1 to untrust any any any permit
set policy from wzone1 to dmz any any any permit
set policy from wzone2 to untrust any any any permit
set policy from untrust to dmz any any http permit
set policy from untrust to dmz any any mail permit
```

5. **WLAN Configuration Activation**

```
exec wlan reactivate
```

Part 13
General Packet Radio Service

General Packet Radio Service is for GPRS network operators who possess advanced knowledge of GPRS technology.

This guide describes the GTP features in ScreenOS and demonstrates how to configure GTP functionality on a Juniper Networks security device. It contains the following chapter:

- "GPRS" on page 2049 describes the GPRS Tunneling Protocol (GTP) features in ScreenOS and demonstrates how to configure GTP functionality on a Juniper Networks security device.

Chapter 62
GPRS

General Packet Radio Service (GPRS) networks connect to several external networks including those of roaming partners, corporate customers, GPRS Roaming Exchange (GRX) providers, and the public Internet. GPRS network operators face the challenge of protecting their network while providing and controlling access to and from these external networks. Juniper Networks provides solutions to many of the security problems plaguing GPRS network operators.

In the GPRS architecture, the fundamental cause of security threats to an operator's network is the inherent lack of security in GPRS Tunneling Protocol (GTP). GTP is the protocol used between GPRS Support Nodes (GSNs). Communication between different GPRS networks is not secure because GTP does not provide any authentication, data integrity, or confidentiality protection. Implementing Internet Protocol security (IPsec) for connections between roaming partners, setting traffic rate limits, and using stateful inspection can eliminate a majority of the GTP's security risks. Juniper Networks security devices mitigate a wide variety of attacks on the Gp, Gn, and Gi interfaces. The GTP firewall features in ScreenOS address key security issues in mobile operators' networks.

NOTE: Only ISG 2000 and 1000 devices (with 2 GB memory) support GTP functionality.

This chapter describes the GTP features that ScreenOS supports and explains how you can configure them on a Juniper Networks security device. This chapter contains the following sections:

The Security Device as a GPRS Tunneling Protocol Firewall

The GPRS Tunneling Protocol (GTP) is used to establish a GTP tunnel, for individual mobile stations (MS), between a Serving GPRS Support Node (SGSN) and a Gateway GPRS Support Node (GGSN). A GTP tunnel is a channel between GSNs through which two hosts exchange data. The SGSN receives packets from the MS and encapsulates them within a GTP header before forwarding them to the GGSN through the GTP tunnel. When the GGSN receives the packets, it decapsulates them and forwards them to the external host.

A Juniper Networks GTP-licensed security device provides firewall protection for the following types of GPRS interfaces:

- Gn—The Gn interface is the connection between an SGSN and a GGSN within the same Public Land Mobile Network (PLMN).
- Gp—The Gp interface is the connection between two Public Land Mobile Network PLMNs.
- Gi—The Gi interface is the connection between a GGSN and the Internet or destination networks connected to a PLMN.

NOTE: The term *interface* has different meanings in ScreenOS and in GPRS technology. In ScreenOS, an interface is like a doorway to a security zone and allows traffic to enter and exit the zone. In GPRS, an interface is a connection, or a reference point, between two components of a GPRS infrastructure, for example, an SGSN and a GGSN.

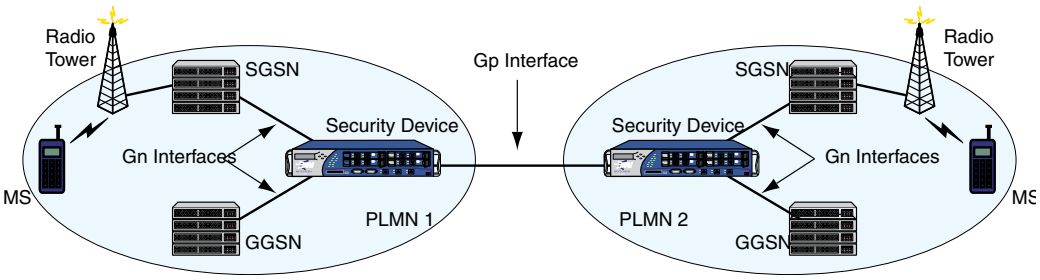
Gp and Gn Interfaces

You implement a security device on the Gn interface to protect core network assets such as the SGSN and GGSN. To secure GTP tunnels on the Gn interface, you place the security device between SGSNs and GGSNs within a common PLMN.

When you implement a security device to the Gp interface, you protect a PLMN against another PLMN. To secure GTP tunnels on the Gp interface, you place the SGSNs and GGSNs of a PLMN behind the security device so that all traffic, incoming and outgoing, goes through the firewall.

Figure 489 on page 2051 illustrates the placement of Juniper Networks security devices to protect PLMNs on the Gp and Gn interfaces.

Figure 489: Gp and Gn Interfaces



Gi Interface

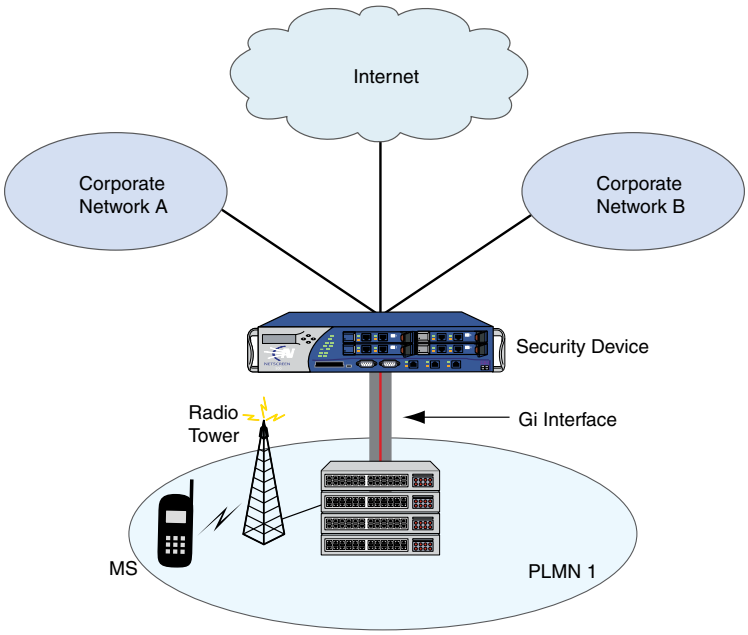
When you implement a security device on the Gi interface, you can simultaneously control traffic for multiple networks, protect a PLMN against the Internet and external networks, and protect mobile users from the Internet and other networks. ScreenOS provides a great number of virtual routers, making it possible for you to use one virtual router per customer network and thereby allow the separation of traffic for each customer network.

The security device can securely forward packets to the Internet or destination networks using the Layer 2 Tunneling Protocol (L2TP) for IPsec virtual private network (VPN) tunnels. (Note, however, that Juniper Networks security devices do not support full L2TP.)

For more information about the features and capabilities of virtual routers, see *"Routing" on page 1235*.

Figure 490 on page 2052 illustrates the implementation of a security device to protect a PLMN on the Gi interface.

Figure 490: Gi Interface



Operational Modes

ScreenOS supports two interface operational modes with GTP: transparent mode and route mode. If you want the security device to participate in the routing infrastructure of your network, you can run it in route mode. This requires a certain amount of network redesign. Alternatively, you can implement the security device into your existing network in transparent mode without having to reconfigure the entire network. In transparent mode, the security device functions as a Layer 2 switch or bridge, and the IP addresses of interfaces are set at 0.0.0.0, making the presence of the security device invisible, or *transparent*, to users.

ScreenOS supports Network Address Translation (NAT) on interfaces and policies that do not have GTP inspection enabled.

Currently in ScreenOS, both transparent and route mode support Active/Passive high availability (HA) and Active/Active HA.

For more information about operational modes and high availability, see *"Fundamentals" on page 15* and *"High Availability" on page 1763*, respectively.

Virtual System Support

Juniper Networks security devices fully support GTP functionality in virtual systems (vsys). To conserve resources, however, we recommend that you use no more than 10 vsys.

Policy-Based GPRS Tunneling Protocol

By default, the PLMN that the security device protects is in the Trust zone. The security device protects the PLMN in the Trust zone against other PLMNs in other zones. You can place all the PLMNs against which you are protecting your PLMN in the Untrust zone, or you can create user-defined zones for each PLMN. A PLMN can occupy one security zone or multiple security zones.

You must create policies to enable traffic to flow between zones and PLMNs. Policies contain rules that permit, deny, or tunnel traffic. A security device performs GTP policy filtering by checking every GTP packet against policies that regulate GTP traffic and by then forwarding, dropping, or tunneling the packet based on these policies.

By selecting the GTP service in a policy, you enable the security device to permit, deny, or tunnel GTP traffic. However, this does not enable the device to inspect GTP traffic. In order for the security device to inspect GTP traffic, you must apply a GTP configuration, also referred to as a *GTP inspection object*, to a policy.

Before you can apply a GTP configuration to a policy, you must first create a GTP inspection object (see "GPRS Tunneling Protocol Inspection Object" on page 2055). You can apply only one GTP inspection object per policy, but you can apply a GTP inspection object to multiple policies. Using policies, you can permit or deny the establishment of GTP tunnels from certain peers such as an SGSN.

You can configure policies that specify "Any" as the source or destination zone (thereby including all hosts in the zone), and you can configure policies that specify multiple source and destination addresses.

In policies, you can enable features such as traffic logging and traffic counting. For more information about policies, see *"Fundamentals" on page 15*.

Example: Configuring Policies to Enable GTP Inspection

In this example, you configure interfaces and create addresses and two policies to allow bidirectional traffic between two networks within the same PLMN. You also apply a GTP inspection object to the policies.

WebUI

1. **GTP Inspection Object**

   Objects > GTP > New: Enter the following, then click **Apply**.

   GTP Name: GPRS1

2. **Interfaces**

   Network > Interfaces > Edit (for ethernet1/1): Enter the following, then click **Apply**:

   Zone Name: Trust

IP Address/Netmask: 10.1.1.1/24

Network > Interfaces > Edit (for ethernet1/2): Enter the following, then click **OK**:

Zone Name: Untrust
IP Address/Netmask: 1.1.1.1/24

3. **Addresses**

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: local-GGSN
IP Address/Domain Name:
IP/Netmask: (select), 10.1.1.0/24
Zone: Trust

Policy > Policy Elements > Addresses > List > New: Enter the following, then click **OK**:

Address Name: remote-SGSN
IP Address/Domain Name:
IP/Netmask: (select), 1.2.2.5/32
Zone: Untrust

4. **Policies**

Policies > (From: Trust, To: Untrust) > New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), local-GGSN
Destination Address:
Address Book Entry: (select), remote-SGSN
Service: GTP
GTP Inspection Object: GPRS1 (select)
Action: Permit

Policies > (From: UnTrust, To: Trust) > New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), remote-SGSN
Destination Address:
Address Book Entry: (select), local-GGSN
Service: GTP
GTP Inspection Object: GPRS1 (select)
Action: Permit

CLI

1. **GTP Inspection Object**

```
set gtp configuration gprs1
(gtp:gprs1)-> exit
```

save

2. **Interfaces**
   set interface ethernet1/1 zone trust
   set interface ethernet1/1 ip 10.1.1.1/24
   set interface ethernet1/2 zone untrust
   set interface ethernet1/2 ip 1.1.1.1/24

3. **Addresses**
   set address trust local-ggsn 10.1.1.0/32
   set address untrust remote-sgsn 2.2.2.5/32

4. **Policies**
   set policy from trust to untrust local-ggsn remote-sgsn gtp permit

   The system returns a policy ID, for example: policy id = 4.

   set policy id 4 gtp gprs1
   set policy from untrust to trust remote-sgsn local-ggsn gtp permit

   The system returns a policy ID, for example: policy id = 5.

   set policy id 5 gtp gprs1
   save

GPRS Tunneling Protocol Inspection Object

To enable the security device to perform the inspection of GPRS Tunneling Protocol (GTP) traffic, you must create a GTP inspection object and then apply it to a policy. GTP inspection objects provide more flexibility in that they allow you to configure multiple policies that enforce different GTP configurations. You can configure the security device to control GTP traffic differently based on source and destination zones and addresses, action, and so on.

To configure GTP features, you must enter the context of a GTP configuration. To save your settings in the CLI, you must first exit the GTP configuration, then enter the **save** command.

Example: Creating a GTP Inspection Object

In this example, you create a GTP inspection object named LA-NY. You preserve most of the default values, but you enable the Sequence Number Validation and GTP-in-GTP Denied features.

WebUI

Objects > GTP > New: Enter the following, then click **Apply**.

GTP Name: LA-NY
Sequence Number Validation: (select)
GTP-in-GTP Denied: (select)

CLI

```
set gtp configuration la-ny
(gtp:la-ny)-> set seq-number-validated
(gtp:la-ny)-> set gtp-in-gtp-denied
(gtp:la-ny)-> exit
save
```

GTP Message Filtering

When a security device receives a GTP packet, it checks the packet against policies configured on the device. If the packet matches a policy, the device inspects the packet according to the GTP configuration applied to the policy. If the packet fails to meet any of the GTP configuration parameters, the device drops the packet.

This section describes features that constitute a GTP configuration, which the security device uses to perform GTP traffic inspection. It includes the following sections:

Packet Sanity Check

The security device performs a GTP sanity check on each packet to determine if it is a valid UDP and GTP packet. The sanity check protects GPRS Support Node (GSN) resources by preventing them from trying to process malformed GTP packets.

When performing the GTP packet sanity check, the security device examines the header of each GTP packet for the following:

- GTP release version number—ScreenOS supports versions 0 and 1 (including GTP').
- Appropriate setting of predefined bits—Which predefined bits are examined depends on the GTP release version number.
- Protocol type—For version 1 (including GTP').
- UDP/TCP packet length.

If the packet does not conform to UDP and GTP standards, the security device drops the packet, thus preventing the security device from forwarding malformed or forged

traffic. The security device performs GTP packet sanity checking automatically; there is no need to configure this feature.

NOTE: Juniper Networks complies with GTP standards established by the 3rd Generation Partnership Project (3GPP). For more information about these standards, refer to the following technical specification documents:

- 3GPP TS 09.60 v6.9.0 (2000-09)
- 3GPP TS 29.060 v3.8.0 (2001-03)
- 3GPP TS 32.015 v3.9.0 (2002-03)

Message-Length Filtering

You can configure the security device to drop packets that do not meet your specified minimum or maximum message lengths. In the GTP header, the message length field indicates the length, in octets, of the GTP payload. It does not include the length of the GTP header itself, the UDP header, or the IP header. The default minimum and maximum GTP message lengths are 0 and 1452, respectively.

Example: Setting GTP Message Lengths

In this example, you configure the minimum GTP message length to be 8 octets and the maximum GTP message length to be 1200 octets for the GPRS GTP inspection object.

WebUI

Objects > GTP > Edit (GPRS1): Enter the following, then click **Apply**:

```
Minimum Message Length: 8
Maximum Message Length: 1200
```

CLI

```
set gtp configuration gprs1
(gtp:gprs1)-> set min-message-length 8
(gtp:gprs1)-> set max-message-length 1200
(gtp:gprs1)-> exit
save
```

Message-Type Filtering

You can configure the security device to filter GTP packets and permit or deny them based on their message type. By default, the security device permits all GTP message types.

A GTP message type includes one or many messages. When you permit or deny a message type, you automatically permit or deny all messages of the specified type.

For example, if you select to drop the **sgsn-context** message type, you thereby drop **sgsn context request**, **sgsn context response**, and **sgsn context acknowledge** messages. For more information about message types, see "Supported Message Types" on page 2058.

You permit and deny message types based on the GTP version number. For example, you can deny message types for one version while you permit them for the other version.

Example: Permitting and Denying Message Types

In this example, for the GPRS1 GTP configuration, you configure the security device to drop the error-indication and failure-report message types for version 1.

WebUI

Objects > GTP > Edit (GPRS1) > Message Drop: Select the following in the Version 1 column, then click **Apply**:

```
Tunnel Management:
Error Indication: (select)
Location Management:
Failure Report Request/Response: (select)
```

CLI

```
set gtp configuration gprs1
(gtp:gprs1)-> set drop error-indication
(gtp:gprs1)-> set drop failure-report
(gtp:gprs1)-> exit
save
```

Supported Message Types

Table 137 on page 2058 lists the GPRS Tunneling Protocol (GTP) messages supported in GTP Releases 1997 and 1999 (including charging messages for GTP') and the message types that you can use to configure GTP message-type filtering.

Table 137: GPRS Tunneling Protocol (GTP) Messages

Message	Message Type	Version 0	Version 1
create AA pdp context request	create-aa-pdp	b	
create AA pdp context response	create-aa-pdp	b	
create pdp context request	create-pdp	b	b
create pdp context response	create-pdp	b	b
Data record request	data-record	b	b
Data record response	data-record	b	b

Table 137: GPRS Tunneling Protocol (GTP) Messages *(continued)*

Message	Message Type	Version 0	Version 1
delete AA pdp context request	delete-aa-pdp	b	
delete AA pdp context response	delete-aa-pdp	b	
delete pdp context request	delete-pdp	b	b
delete pdp context response	delete-pdp	b	b
echo request	echo	b	b
echo response	echo	b	b
error indication	error-indication	b	b
failure report request	failure-report	b	b
failure report response	failure-report	b	b
forward relocation request	fwd-relocation	b	b
forward relocation response	fwd-relocation	b	b
forward relocation complete	fwd-relocation	b	b
forward relocation complete acknowledge	fwd-relocation	b	b
forward SRNS context	fwd-srns-context	b	b
forward SRNS context acknowledge	fwd-srns-context	b	b
identification request	identification	b	b
identification response	identification	b	b
node alive request	node-alive	b	b
node alive response	node-alive	b	b
note MS GPRS present request	note-ms-present	b	b
note MS GPRS present response	note-ms-present	b	b
pdu notification request	pdu-notification	b	b
pdu notification response	pdu-notification	b	b
pdu notification reject request	pdu-notification	b	b
pdu notification reject response	pdu-notification	b	b
RAN info relay	ran-info	b	b
redirection request	redirection	b	b

Table 137: GPRS Tunneling Protocol (GTP) Messages *(continued)*

Message	Message Type	Version 0	Version 1
redirection response	redirection	b	b
relocation cancel request	relocation-cancel	b	b
relocation cancel response	relocation-cancel	b	b
send route info request	send-route	b	b
send route info response	send-route	b	b
sgsn context request	sgsn-context	b	b
sgsn context response	sgsn-context	b	b
sgsn context acknowledge	sgsn-context	b	b
supported extension headers notification	supported-extension	b	b
g-pdu	gtp-pdu	b	b
update pdp context request	update-pdp	b	b
updated pdp context response	update-pdp	b	b
version not supported	version-not-supported	b	b

Message-Rate Limiting

You can configure the security device to limit the rate of network traffic going to a GSN. You can set separate thresholds, in packets per second, for GTP-Control (GTP-C) messages. Because GTP-C messages require processing and replies, they can potentially overwhelm a GSN. By setting a rate limit on GTP-C messages, you can protect your GSNs from possible Denial of Service (DoS) attacks such as the following:

- **Border gateway bandwidth saturation:** A malicious operator connected to the same GRX as your PLMN can direct so much network traffic at your Border Gateway that legitimate traffic is starved for bandwidth in or out of your PLMN, thus denying roaming access to or from your network.
- **GTP flood:** GTP traffic can flood a GSN, forcing it to spend its CPU cycles processing illegitimate data. This can prevent subscribers from roaming, forwarding data to external networks, and can prevent a GPRS from attaching to the network.

This feature limits the rate of traffic sent to each GSN from the Juniper Networks security device. The default rate is unlimited.

Example: Setting a Rate Limit

In the following example, you limit the rate of incoming GTP-C messages to 300 packets per second.

WebUI

Objects > GTP > Edit (GPRS1): Enter the following, then click **Apply**:

Control Plane Traffic Rate Limit: 300

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set limit rate 300
(gtp:gprs1)-> exit
save
```

Sequence Number Validation

You can configure a security device to perform sequence-number validation.

The header of a GTP packet contains a Sequence Number field. This number indicates to the GGSN receiving the GTP packets the order of the packets. During the Packet Data Protocol (PDP) context-activation stage, a sending GGSN uses zero (0) as the sequence-number value for the first G-PDU it sends through a tunnel to another GGSN. The sending GGSN increments the sequence number value for each following G-PDU it sends. The value resets to zero when it reaches 65535.

During the PDP context-activation stage, the receiving GGSN sets its counter to zero. Subsequently, whenever the receiving GGSN receives a valid G-PDU, the GGSN increments its counter by one. The counter resets to zero when it reaches 65535.

Normally, the receiving GGSN compares the sequence number in the packets it received with the sequence number from its counter. If the numbers correspond, the GGSN forwards the packet. If they differ, the GGSN drops the packet. By implementing a security device between the GGSNs, the device can perform this validation for the GGSN and drop packets that arrive out of sequence. This feature helps conserve GGSN resources by preventing the unnecessary processing of invalid packets.

Example: Enabling Sequence Number Validation

In this example, you enable the Sequence Number Validation feature.

WebUI

Objects > GTP > Edit (GPRS1): Select **Sequence Number Validation**, then click **Apply**.

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set seq-number-validated
(gtp:gprs1)-> exit
save
```

IP Fragmentation

A GTP packet consists of the message body and three headers: GTP, UDP and IP. If the resulting IP packet is larger than the message transmission unit (MTU) on the transferring link, the sending SGSN or GGSN performs an IP fragmentation.

By default, a security device buffers IP fragments until it receives a complete GTP message, and then inspects the GTP message.

GTP-in-GTP Packet Filtering

You can configure a security device to detect and drop a GTP packet that contains another GTP packet in its message body.

Example: Enabling GTP-in-GTP Packet Filtering

In this example, you enable the security device to detect and drop GTP packets that contain a GTP packet in the message body.

WebUI

Objects > GTP > Edit (GPRS1): Select **GTP-in-GTP Denied**, then click **Apply**.

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set gtp-in-gtp-denied
(gtp:gprs1)-> exit
save
```

Deep Inspection

You can configure the security device to perform deep inspection (DI) on the tunnel endpoint ID (TEID) in G-PDU data messages.

Example: Enabling Deep Inspection on the TEID

In this example, you enable the security device to perform DI of G-PDU data messages on the TEID. You can configure DI only from the CLI.

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set teid-di
(gtp:gprs1)-> exit
save
```

GTP Information Elements

Information Elements (IEs) are included in all GTP control message packets. IEs provide information about GTP tunnels, such as creation, modification, deletion, and status. ScreenOS supports IEs consistent with 3GPP Release 6. If you are running an earlier release, or have contractual agreements with operators running earlier releases of 3GPP, you can reduce network overhead by restricting control messages containing unsupported IEs.

This section describes IEs contained in control messages you can configure the security device to screen based on IEs. It includes the following sections:

Access Point Name Filtering

An Access Point Name (APN) is an IE included in the header of a GTP packet that provides information about how to reach a network. An APN comprises two elements:

- **Network ID**—Identifies the name of an external network such as “mobiphone.com”
- **Operator ID**—Which uniquely identifies the operators’ PLMN such as “mnc123.mcc456”

By default, the security device permits all APNs. However, you can configure the security device to perform APN filtering to restrict roaming subscribers’ access to external networks.

To enable APN filtering, you must specify one or more APNs. To specify an APN, you need to know the domain name of the network (for example, mobiphone.com)

and, optionally, the operator ID. Because the domain name (network ID) portion of an APN can potentially be very long and contain many characters, you can use the wildcard "*" as the first character of the APN. The wildcard indicates that the APN is not limited only to mobiphone.com but also includes all the characters that might precede it.

You must also set a *selection mode* for the APN. The selection mode indicates the origin of the APN and whether or not the Home Location Register (HLR) has verified the user subscription. You set the selection mode according to the security needs of your network. Possible selection modes include the following:

- **Mobile Station**—Mobile station-provided APN, subscription not verified

This selection mode indicates that the mobile station (MS) provided the APN and that the HLR did not verify the user's subscription to the network.

- **Network**—Network-provided APN, subscription not verified

This selection mode indicates that the network provided a default APN because the MS did not specify one, and that the HLR did not verify the user's subscription to the network.

- **Verified**—MS or network-provided APN, subscription verified

This selection mode indicates that the MS or the network provided the APN and that the HLR verified the user's subscription to the network.

APN filtering applies only to **create pdp request** messages. When performing APN filtering, the security device inspects GTP packets to—look for APNs that match APNs that you set. If the APN of a GTP packet matches an APN that you specified, the security device then verifies the selection mode and only forwards the GTP packet if both the APN and the selection mode match the APN and the selection mode that you specified. Because APN filtering is based on perfect matches, using the wildcard "*" when setting an APN suffix can prevent the inadvertent exclusion of APNs that you would otherwise authorize. The security device automatically denies all other APNs that do not match.

Additionally, a security device can filter GTP packets based on the combination of an International Mobile Subscriber Identity (IMSI) prefix and an APN.

Example: Setting an APN and a Selection Mode

In this example, you set **mobiphone.com.mnc123.mcc456.gprs** as an APN and use the wildcard "*". You also set **Network** as the selection mode.

WebUI

Objects > GTP > Edit (GPRS1) > APN + IMSI > New: Enter the following, then click **OK**:

```
Access Point Name: *mobiphone.com.mnc123.mcc456.gprs
Selection Mode: Network (select)
```

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set apn *mobiphone.com.mnc123.mcc456.gprs selection net
(gtp:gprs1)-> exit
```

IMSI Prefix Filtering

A GPRS Support Node (GSN) identifies a mobile station (MS) by its International Mobile Station Identity (IMSI). An IMSI comprises three elements: the Mobile Country Code (MCC), the Mobile Network Code (MNC), and the Mobile Subscriber Identification Number (MSIN). The MCC and MNC combined constitute the IMSI prefix and identify the mobile subscriber's home network, or Public Land Mobile Network (PLMN).

By setting IMSI prefixes, you can configure the security device to deny GTP traffic coming from nonroaming partners. By default, a security device does not perform IMSI prefix filtering on GTP packets. By setting IMSI prefixes, you configure the security device to filter **create pdp request** messages and only permit GTP packets with IMSI prefixes that match the ones you set. The security device allows GTP packets with IMSI prefixes that do not match any of the IMSI prefixes that you set. To block GTP packets with IMSI prefixes that do not match any of the IMSI prefixes set, use an explicit wildcard for the IMSI filter, and the action: **drop** should be the last IMSI prefix filtering policy.

When you filter GTP packets based on an IMSI prefix, you must also specify an APN. See "Example: Setting a Combined IMSI Prefix and APN Filter" on page 2065

Example: Setting a Combined IMSI Prefix and APN Filter

In this example, you set **mobiphone.com.mnc123.mcc456.gprs** as an APN and use the wildcard "*". You permit all selection modes for this APN. You also set the IMSI prefix for a known PLMN, which is 246565. The MCC-MNC pair can be five or six digits.

WebUI

Objects > GTP > Edit (GPRS1) > APN + IMSI: Enter the following, then click **OK**:

```
Access Point Name: *mobiphone.com.mnc123.mcc456.gprs
Mobile Country-Network Code: 246565
Selection Mode: Mobile Station, Network, Verified (select)
```

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set mcc-mnc 246565 apn *mobiphone.com.mnc123.mcc456.gprs pass
(gtp:gprs1)-> exit
save
```

NOTE: Selecting the variable **pass** in the CLI is equal to selecting all three selection modes in the WebUI. Using this variable permits traffic from all selection modes for the specified APN.

Radio Access Technology

The Radio Access Technology (RAT) information element provides ways to stimulate Wideband Code Division Multiple Access (WCDMA), and to perform reporting via billing information systems.

Previously, the SGSN IP address was used to distinguish between 3rd Generation Wireless Mobile Communication Technology (3G) systems and 2nd Generation Wireless Mobile Communication Technology (2G) systems. With the introduction of combined 2G/3G SGSNs, however, you must configure the RAT Information Element to enable the security device to make this distinction. When you set a RAT IE, you must also specify an APN. See “Example: Setting an RAT and APN Filter” on page 2066

Example: Setting an RAT and APN Filter

In this example, you set **mobiphone.com.mnc123.mcc456.gprs** as an APN and use the wildcard “*”. You permit all selection modes for this APN. You configure the security device to drop the GTP message if the value of the RAT IE matches the string value 123.

WebUI

Currently, you can set an RAT and APN combination only from the command line interface (CLI).

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set rat 123 apn *mobiphone.com drop
(gtp:gprs1)-> exit
save
```

Routing Area Identity and User Location Information

Some countries restrict subscriber access to certain types of network content. To comply with these regulatory demands, network operators need to be able to police subscriber’s requested content before allowing a content download. ScreenOS gives network operators the ability to screen content based on the Routing Area Identity (RAI) and User Location Information (ULI) IEs. Because the current 3GPP Call Detail Record (CDR) formats and realtime charging interfaces lack these attributes, billing and charging systems are required to look up SGSN IP addresses to determine roaming partners for settlement and end user charging. ScreenOS gives network operators the ability to screen control messages based on RAI and ULI. When you set a RAI or ULI IE, you must also specify an APN. See “Example: Setting an RAI and APN Filter” on page 2067 and “Example: Setting a ULI and APN Filter” on page 2067 and

Example: Setting an RAI and APN Filter

In this example, you set **mobiphone.com** as an APN and use the wildcard "*". You permit all selection modes for this APN. And you configure the security device to drop GTP messages if the RAI IE matches the string value: 12345*.

WebUI

Currently, you can set an RAI and APN combination only from the command line interface (CLI).

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set rai 12345* *mobiphone.com drop
(gtp:gprs1)-> exit
save
```

Example: Setting a ULI and APN Filter

In this example, you set **mobiphone.com** as an APN and use the wildcard "*". You permit all selection modes for this APN. And you configure the security device to drop GTP messages if the ULI IE matches the string value 123456.

WebUI

Currently, you can set a ULI and an APN combination only from the command line interface (CLI).

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set uli 123456 apn mobiphone.com drop
(gtp:gprs1)-> exit
save
```

APN Restriction

Multiple concurrent primary Packet Data Protocol (PDP) contexts, and an MS/UE capable of routing between these two access points, can put IP security at risk for corporate users who have both private and a public APNs. The APN Restriction IE, added to the GTP **create PDP context response** message, ensures the mutual exclusivity of a PDP context if requested by a GGSN (or rejected if this condition cannot be met), and thus avoids the security threat.

IMEI-SV

The International Mobile Equipment Identity-Software Version (IMEI-SV) IE provides ways to adapt content to the terminal type and client application whenever a proxy server for this purpose is not present. This IE is also useful in reports generated from

the GGSN, AAA and/or Wireless Application Protocol Gateway (WAP GW). The GTP-aware security device supports the RAT, RAI, ULI, APN Restriction and IMEI-SV in GTP attributes to avoid treatment or categorization as unambiguous traffic, which can be harmful to GPRS traffic or GPRS roaming traffic. These attributes are included in the set of useful filter attributes used to block specific GPRS traffic and or GPRS roaming traffic. When you set an IMEI-SV IE, you must also specify an APN. See "Example: Setting an IMEI-SV and APN Filter" on page 2068

Example: Setting an IMEI-SV and APN Filter

In this example, you set **mobiphone.com** as an APN and use the wildcard "*". You permit all selection modes for this APN. And you configure the security device to pass the GTP message if the IMEI-SV IE matches the string: 87652.

WebUI

Currently, you can set an RAI and APN combination only from the command line interface (CLI).

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set imei-sv 87652* apn mobiphone.com pass
(gtp:gprs1)-> exit
save
```

Protocol and Signaling Requirements

The security device supports the following attributes in the GTP **Create PDP Context Request** message:

- RAT
- RAI
- ULI
- APN Restriction
- IMEI-SV

The security device supports the following attributes in the GTP **Update PDP Context Request** message:

- RAT
- RAI
- ULI

The security device supports the APN Restriction attribute in the GTP **Update PDP Context Response** message.

You can configure the above GTP signaling messages on the security device as follows:

- Transparently pass
- Block based on (individually)
  - RAT
  - RAI (with ranges such as 123*)
  - ULI (with ranges)
  - IMEI-SV (with ranges)

Combination Support for IE Filtering

For concurrent support of R6 filtering on Information Elements (IEs), the following rules apply:

- By default, the security device does not perform IE filtering on GTP packets.
- In each command line, attributes are anded in the following order of precedence:
  1. RAT
  2. RAI
  3. ULI
  4. IMEI
  5. MCC-MNC
- Whenever you set an attribute restriction, you must also specify an APN.

  For example, if you want the security device to pass GTP messages containing RAT 1 *and* RAI 567* *and* MCC-MNC 56789, or to pass messages with RAI 123*, but to default to drop packets with any APN value, the following configuration will accomplish this:

  ```
  set rat 1 rai 567* mcc-mnc 56789 apn * pass
  set rai 123* apn * pass
  set apn * drop
  ```

  The first line of the configuration causes the security device to pass GTP messages containing RAT 1, RAI 567*, MCC-MNC 56789, *and* any APNs. The second line of the configuration causes the device to pass messages containing RAI 123* *and* any APNS. The third line causes the device to drop any APNs.

Supported R6 Information Elements

ScreenOS supports all 3GPP R6 IEs for GTP, as listed in Table 138 on page 2069.

Table 138: Supported Information Elements

IE Type Value	Information Element
1	Cause

Table 138: Supported Information Elements *(continued)*

IE Type Value	Information Element
2	International Mobile Subscriber Identity (IMSI)
3	Routing Area Identity (REI)
4	Temporary Logical Link Identity (TLLI)
5	Packet TMSI (P-TMSI)
8	Reordering Required
9	Authentication Triplet
11	MAP Cause
12	P-TMSI Signature
13	MS Validated
14	Recovery
15	Selection Mode
16	Tunnel Endpoint Identifier Data I
17	Tunnel Endpoint Identifier Control Plane
18	Tunnel Endpoint Identifier Data II
19	Teardown ID
20	NSAPI
21	RANAP Cause
22	RAB Context
23	Radio Priority SMS
24	Radio Priority
25	Packet Flow ID
26	Charging Characteristics
27	Trace Reference
28	Trace Type
29	MS Not Reachable Reason
127	Charging ID
128	End User Address

Table 138: Supported Information Elements *(continued)*

IE Type Value	Information Element
129	MM Context
130	PDP Context
131	Access Point Name
132	Protocol Configuration Options
133	GSN Address
134	MS International PSTN/ISDN Number (MSISDN)
135	Quality of Service Profile
136	Authentication Quintuplet
137	Traffic Flow Template
138	Target Identification
139	UTRAN Transparent Container
140	RAB Setup Information
141	Extension Header Type List
142	Trigger Id
143	OMC Identity
144	RAN Transparent Container
145	PDP Context Prioritization
146	Additional RAB Setup Information
147	SGSN Number
148	Common Flags
149	APN Restriction
150	Radio Priority LCS
151	RAT Type
152	User Location Information
153	MS Time Zone
154	IMEI-SV
155	CAMEL Charging Information Container

Table 138: Supported Information Elements *(continued)*

IE Type Value	Information Element
156	MBMS UE Context
157	Temporary Mobile Group Identity (TMGI)
158	RIM Routing Address
159	MBMS Protocol Configuration Options
160	MBMS Service Area
161	Source TNC PDCP context Information
162	Additional Trace Information
163	Hop Counter
164	Selected PLMN ID
165	MBMS Session Identifier
166	MBMS2G/3G Indicator
167	Enhanced NSAPI
168	MBMS Session Duration
169	Additional MBMS Trace Information
251	Charging Gateway Address
255	Private Extension

3GPP R6 IE Removal

The 3GPP R6 IE Removal feature allows you to retain interoperability in roaming between 2GPP and 3GPP networks. You can configure the GTP-aware security device, residing on the border of a PLMN and a GRX and acting as a Gp firewall, to remove 3GPP-specific attributes from the GTP packet header when the packet passes into a 2GPP network. You can configure the security device to remove the RAT, RAI, ULI, IMEI-SV, and APN Restriction IEs from GTP messages prior to forwarding these messages to the GGSN.

Example: R6 Removal

In this example, you configure the Gp interface of the security device to remove newly added R6 IEs (RAT, ULI, IMEI-SV and APN Restriction) from the GTP message.

WebUI

Objects > GTP > New: Select the following, then click **Apply**:

Remove R6 EI: (Select)

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set remove-r6
(gtp:gprs1)-> exit
save
```

GTP Tunnels

A GTP tunnel enables the transmission of GTP traffic between GSNs using the GPRS Tunneling Protocol (GTP). There are two types of tunnels: one for GTP-U (user data) messages and one for GTP-C (signaling and control) messages.

GTP Tunnel Limiting

You can configure the security device to limit the number of GTP tunnels. The GSNs to which this limit applies is specified in the policy to which you append the GTP inspection object. This ensures that the capacity of the GSNs is not exceeded.

Example: Setting GTP Tunnel Limits

In the following example, you limit the number of roaming GTP tunnels to 800 for the GPRS1 GTP inspection object.

WebUI

Objects > GTP > Edit (GPRS1): Enter the following, then click **Apply**:

Maximum Number of Tunnels
Limited to tunnels: (select), 800

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set limit tunnel 800
(gtp:gprs1)-> exit
save
```

Stateful Inspection

Following a series of GTP packet verifications (see Figure 162 on page 638), the security device verifies the GTP packet against the current GTP tunnel state. The security device bases its action of forwarding or dropping a GTP packet on previous GTP packets it received. For example, a request message precedes a response message, so if the security device receives a **create pdp context response** message when it did not previously receive a **create pdp context request** message, the security device drops the response message.

Basically, if the security device receives a GTP packet that does not belong in the current GTP state model, it drops the packet. The following are simplified examples of GTP state models.

GTP Tunnel Establishment and Teardown

A mobile station (MS) wants to reach an external network (www.buygadgets.com) and performs a GPRS attach with an SGSN to initiate a GTP tunnel establishment. The SGSN sends a **create pdp context request** message to a GGSN. If the GGSN is able to successfully accept the connection (authentication if any, resource allocation, Quality of Service (QoS) guarantees), it replies with a **create pdp context response** message. This exchange of messages between the SGSN and the GGSN establishes a GTP tunnel through which the MS can send GTP-U messages to the external network.

To terminate the communication, the MS performs a GPRS detach with the SGSN to initiate the GTP tunnel teardown. The SGSN sends a **delete pdp context request** message to the GGSN. The GGSN replies with a **delete pdp context response** message and deletes the GTP tunnel from its records. When the SGSN receives the response, it also removes the GTP tunnel from its records.

A security device can receive multiple requests to establish GTP tunnels for different GSNs simultaneously. To help keep track of all tunnels (tunnel status and log messages for the different tunnels), a security device assigns a unique index to each tunnel upon its creation. That tunnel index appears for each logged GTP tunnel message.

Inter SGSN Routing Area Update

When an MS moves out of the range of the current SGSN and enters a new SGSN area, the new SGSN sends a **sgsn context request** to the old SGSN asking it to transfer all information it has on the MS. The old SGSN responds with a **sgsn context response** message and sends the new SGSN all the information it has on the MS. Upon receiving the response and information, the new SGSN confirms reception by sending a **sgsn context acknowledge** message to the old SGSN.

From this point on, the old SGSN forwards to the new SGSN any new T-PDUs it receives for the MS. To complete this **hand over** procedure, the new SGSN must send an **update pdp context request** message to the GGSN to which the GGSN replies with a **update pdp context response** message.

In the case where the SGSNs are located in different PLMNs, all the GTP messages go through the security device. In the case where the two SGSNs are in the same PLMN and the GGSN is in a different PLMN, only the **update pdp context request/response** messages go through the security device.

Tunnel Failover for High Availability

ScreenOS supports Active/Active and Active/Passive high availability in Route or transparent mode. In essence, two security devices in an HA configuration act as master and backup devices. The backup device mirrors the master's configuration, including existing GTP tunnels, and is ready to take over the duties of the master device if the master fails. The failover between master and backup is rapid and invisible to the user.

During failover, established GTP tunnels remain active and intact, but GTP tunnels in the process of establishment are lost. For these, you have to re-initiate GTP tunnel establishment after a failover. It is also possible that GTP tunnels in the process of teardown (or termination) miss the confirmation message and are left hanging on the security device. Hanging GTP tunnels can occur for various reasons. With regards to HA, a hanging GTP tunnel occurs when the GSN at one end of a tunnel sends the GSN at the other end of the tunnel a **delete pdp context request** message, and while it is waiting for the response, a failure occurs disrupting the communication and preventing the GSN from receiving the **delete pdp context response** message (confirming the deletion) from the other GSN. The GSN that sent the confirmation message simultaneously deleted its pdp context while the GSN at the other end of the GTP tunnel is left hanging, still waiting for the deletion confirmation.

You can configure the security device to remove hanging GTP tunnels. For more information, see "Hanging GTP Tunnel Cleanup" on page 2075.

For more information about HA and to learn how to configure security devices for high availability, see *"High Availability" on page 1763*.

Hanging GTP Tunnel Cleanup

This feature removes hanging GTP tunnels on the security device. GTP tunnels may hang for a number of reasons, for instance, **delete pdp context response** messages might get lost on a network or a GSN might not get properly shut down. You can configure the security device to detect and remove hanging GTP tunnels automatically.

When you set a GTP tunnel timeout value, the security device automatically identifies as "hanging" any GTP tunnel that is idle for the period of time specified by the timeout value and removes it. The default GTP tunnel timeout value is 24 hours.

Example: Setting the Timeout for GTP Tunnels

In this example, you set the GTP tunnel timeout for the "GPRS1" GTP inspection object to 12 hours.

WebUI

Objects > GTP > Edit (GPRS1): Enter the following, then click **Apply**:

Tunnel Inactivity Timeout: 12

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set timeout 12
(gtp:gprs1)-> exit
save
```

SGSN and GGSN Redirection

Juniper Networks security devices support GTP traffic redirection between SGSNs and GGSNs.

- **SGSN Redirection**—An SGSN (A) can send create-pdp-context requests in which it can specify different SGSN IP addresses (SGSN B and SGSN C) for subsequent GTP-C and GTP-U messages. Consequently, the GGSN sends the subsequent GTP-C and GTP-U messages to SGSNs B and C, instead of A.
- **GGSN Redirection**—A GGSN (X) can send create-pdp-context responses in which it can specify different GGSN IP addresses (GGSN Y and GGSN Z) for subsequent GTP-C and GTP-U messages. Consequently, the SGSN sends the subsequent GTP-C and GTP-U messages to GGSNs Y and Z, instead of X.

Overbilling-Attack Prevention

You can configure security devices to prevent GPRS Overbilling attacks. The following section describes the Overbilling attack and then explains the solution.

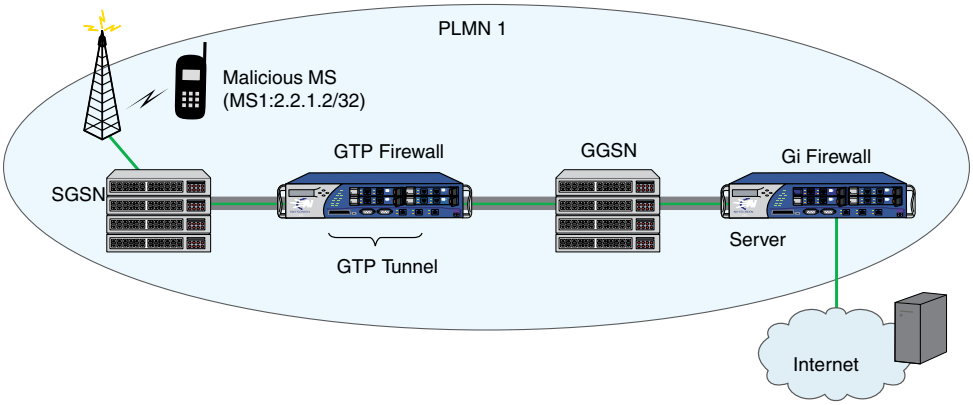
Overbilling-Attack Description

In order to understand an Overbilling attack, it is important to know that a mobile station (MS) gets its IP address from an IP pool. This said, an Overbilling attack can occur in various ways. Namely, it can occur when a legitimate subscriber returns his IP address to the IP pool, at which point an attacker can hijack the IP address, which is vulnerable because the session is still open. When the attacker takes control of the IP address without being detected and reported, the attacker can download data for free (or, more accurately, at the expense of the legitimate subscriber) or send data to other subscribers.

An Overbilling attack can also occur when an IP address becomes available and gets reassigned to another MS. Traffic initiated by the previous MS might be forwarded to the new MS, causing the new MS to be billed for unsolicited traffic. Figure 491 on page 2077, Figure 492 on page 2077, and Figure 493 on page 2078 illustrate this scenario in detail.

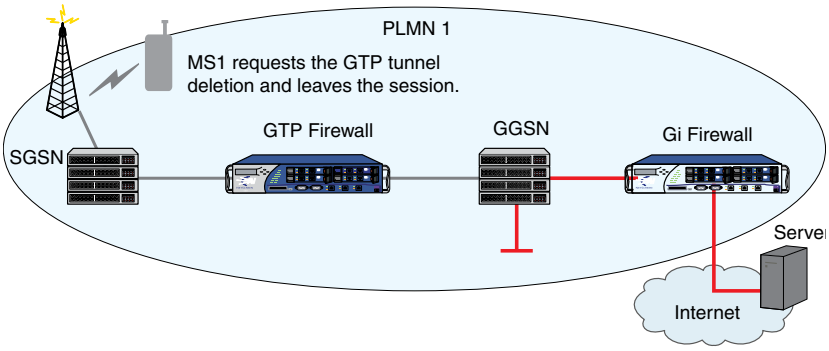
In Figure 491 on page 2077, the MS1 gets an IP address and requests a GTP tunnel to the GGSN. The SGSN builds a GTP tunnel per MS1's request. MS1 initiates a session with the server.

Figure 491: Starting a Session



In Figure 492 on page 2077, as the server begins to send packets to MS1, MS1 simultaneously sends a request to the SGSN to delete the GTP tunnel but leaves open the session to the server.

Figure 492: Deleting a GTP Tunnel

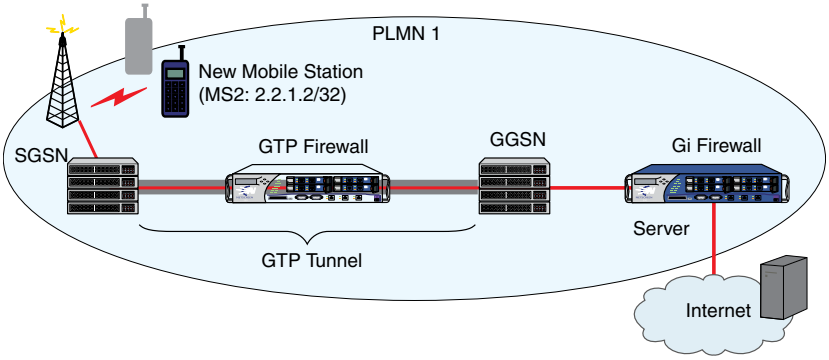


The server continues to send packets to the GGSN. The Gi firewall, not aware that the GTP tunnel was deleted, forwards the packets to the GGSN. The GGSN drops the packets because the GTP tunnel no longer exists.

In Figure 493 on page 2078, a new mobile station, MS2 (the victim), sends a request to the SGSN for a GTP tunnel to the GGSN and receives the IP address of 2.2.1.2/32 (the same IP address used by MS1). The SGSN creates a new GTP tunnel to the GGSN.

Upon detecting the new GTP tunnel for destination IP address 2.2.1.2, the GGSN, which kept receiving packets for the old session with the same destination IP address but different MS (MS1), now forwards these packets to MS2. Although MS2 did not solicit this traffic intended for MS1, MS2 gets billed for it.

Figure 493: Receiving Unsolicited Data



Overbilling-Attack Solution

To protect subscribers of a PLMN from Overbilling attacks requires two security devices and involves NetScreen Gatekeeper Protocol (NSGP) and the NSGP module.

The NSGP module includes two components: the client and the server. The client connects to the server and sends requests, which the server processes. Both client and server support multiple connections to each other and to others simultaneously.

NSGP uses Transmission Control Protocol (TCP) and monitors connectivity between a client and a server by sending Hello messages at set intervals. NSGP currently only supports the session context, which is a space that holds user-session information, is bound to a security zone, and is identified by a unique number (context ID).

When configuring NSGP on the client and server devices, you must use the same context ID on each device. When the client sends a clear session request to the server, the request must include the context ID and IP address of the server. Upon receiving the clear session request, the server matches the context ID and then clears the session from its table.

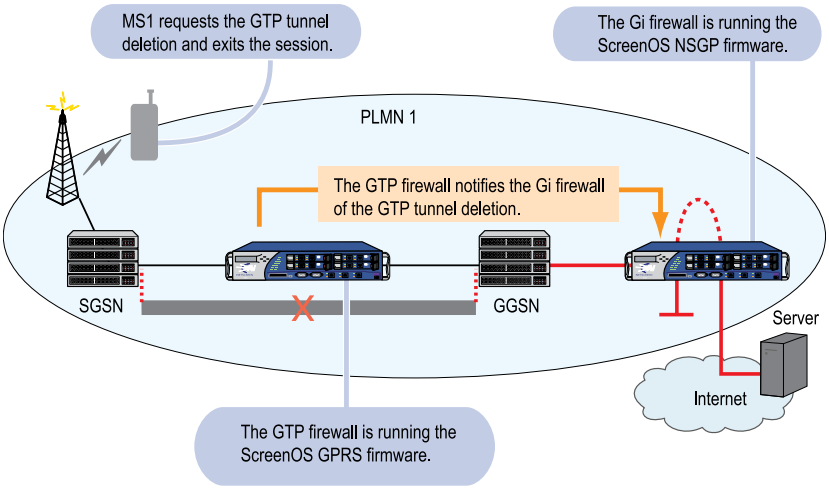
The security device acting as the Gi firewall (the server) and the other device acting as the GTP firewall (the client) must run ScreenOS 5.0.0 or a later release. You configure NSGP on the GTP firewall to enable it to notify the Gi firewall when a GTP tunnel is deleted and you configure NSGP on the Gi firewall to enable it to automatically clear sessions whenever the Gi firewall gets a notification from the GTP firewall that a GTP tunnel was deleted. By clearing the sessions, the Gi firewall stops the unsolicited traffic.

NOTE: We strongly recommend that you upgrade to the latest ScreenOS release.

After MS1 initiates a session with the server and as the server begins to send packets to MS1, MS1 sends a request to the SGSN to delete the GTP tunnel and exits the session.

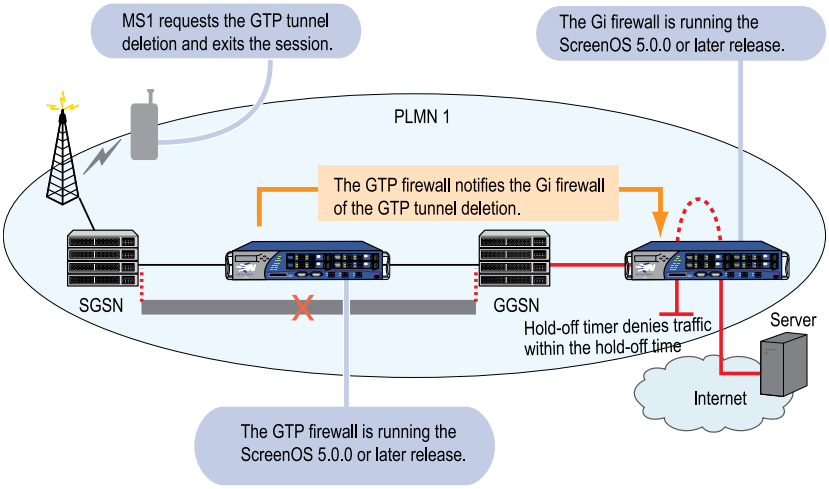
When the SGSN deletes the tunnel, the GTP firewall immediately notifies the Gi firewall, and the Gi firewall removes the session from its table. Subsequently, when the server attempts to send packets to the GGSN, the Gi firewall intercepts and drops them. As a result, a new MS cannot receive and be charged for traffic it did not initiate itself, even if it uses the same IP address as a previous MS. However, sometimes traffic intended for MS1 can still traverse the Gi firewall because of policy permissions. To avoid this, ScreenOS provides a *hold-off timer,* see Figure 8. By setting this timer, you direct the Gi firewall to deny unintended traffic from the server that arrives within the hold-off time. Additionally, the IP address used by MS1 will be assigned to a new MS only after the hold-off timer expires.

Figure 494: GTP Tunnel Deletion Notification



By notifying the Gi firewall of the GTP tunnel deletion, denying traffic for the duration of the hold-off time, and retaining the IP address for a specific period, ScreenOS saves the new MS from GPRS Overbilling attacks.

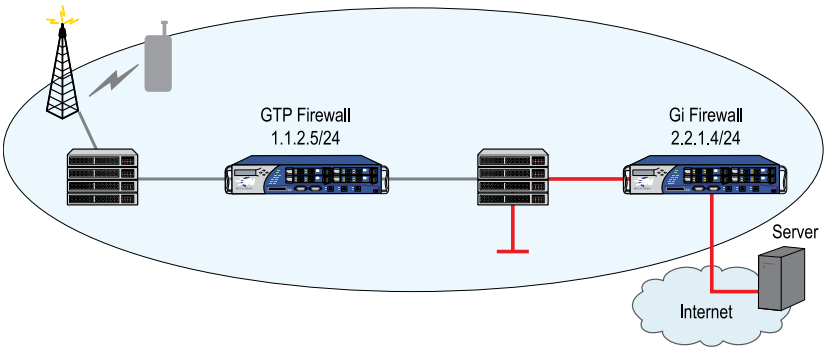
Figure 495: Denying Traffic using Hold-off Timer



Example: Configuring the Overbilling Attack Prevention Feature

In this example, you configure NSGP on both the GTP firewall (client) and the Gi firewall (server). This example assumes that you configured the GPRS1 GTP inspection object on both the GTP and Gi firewalls.

Figure 496: GTP and Gi Firewall Setup



GTP Firewall (Client)

WebUI

Network > Interface > Edit (ethernet1/2): Enter the following, then click **Apply**:

Zone Name: Untrust (select)
IP Address/Netmask: 1.1.2.5/24
Management Services: Telnet (select)

Objects > GTP > Edit (GPRS1) > Overbilling: Enter the following, then click **Apply**:

Overbilling Notify: (select)

Destination IP: 2.2.1.4
Source Interface: ethernet1/2
Destination Context: 2

Policies > (From: Untrust, To: Trust) New: Enter the following, then click **OK**:

Source Address:
Address Book Entry: (select), Any
Destination Address:
Address Book Entry: (select), Any
Service: Any
GTP Inspection Object: GPRS1
Action: Permit

CLI

```
set interface ethernet1/2 zone Untrust
set interface ethernet1/2 ip 1.1.2.5/24
set interface ethernet1/2 manage telnet
set gtp config gprs1
(gtp:gprs1)-> set notify 2.2.1.4 src-interface ethernet1/2 context 2
(gtp:gprs1)-> exit
save
set policy from untrust to trust any any any permit
```

The system returns a policy ID, for example: policy id = 2

```
set policy id 2 gtp gprs1
save
```

Gi Firewall (Server)

WebUI

Network > Interface > Edit (ethernet1/2): Enter the following, then click **Apply**:

Zone Name: Untrust (select)
IP Address/Netmask: 2.2.1.4/24
Management Services: Telnet (select)
Other Services: Overbilling (select)

NSGP: Enter the following, click **Add**, then click **OK**:

Context ID: 2
Zone: Untrust

CLI

```
set interface ethernet1/2 zone Untrust
set interface ethernet1/2 ip 2.2.1.4/24
set interface ethernet1/2 manage telnet
set interface ethernet1/2 nsgp
set nsgp context 2 type session zone untrust
save
```

GTP Traffic Monitoring

Juniper Networks security devices provide comprehensive tools for monitoring traffic flow in real-time. For GTP traffic, you can monitor traffic using the GTP traffic logging and the GTP traffic counting features.

Traffic Logging

With the GTP traffic logging feature, you can configure the security device to log GTP packets based on their status. You can also specify how much information, basic or extended, you want about each packet. You can use the console, syslog, and the WebUI to view traffic logs.

The status of a GTP packet can be any of the following:

- Forwarded—A packet that the security device transmits because the GTP policy allows it.
- Prohibited—A packet that the security device drops because the GTP policy denies it.
- Rate-limited—A packet that the security device drops because it exceeds the maximum rate limit of the destination GSN.
- State-invalid—A packet that the security device drops because it failed stateful inspection.
- Tunnel-limited—A packet that the security device drops because the maximum limit of GTP tunnels for the destination GSN is reached.

NOTE: By default, traffic logging is disabled on a Juniper Networks security device.

Each log entry in its basic form contains the following information:

- Timestamp
- Source IP address
- Destination IP address
- Tunnel Identifier (TID) or Tunnel Endpoint Identifier (TEID)
- Message type
- Packet status: forwarded, prohibited, state-invalid, rate-limited, or tunnel-limited
- Interface, vsys, or vrouter name (if applicable)
- Public Land Mobile Network (PLMN) or zone name

Each log entry in its extended form contains the following information in addition to the “basic” information:

- IMSI
- MSISDN
- APN
- Selection mode
- SGSN address for signaling
- SGSN address for user data
- GGSN address for signaling
- GGSN address for user data

NOTE: For more information about monitoring features, see *"High Availability" on page 1763*.

When enabling the logging of GTP packets with a Packet Rate-Limited status, you can also specify a logging frequency to control the interval at which the security device logs these messages. For example, if you set the frequency value to 10, the security device only logs every tenth message above the set rate limit.

By setting a logging frequency, you help conserve resources on the syslog server and on the security device and can avoid a logging overflow of messages.

Example: Enabling GTP Packet Logging

In this example, for the "GPRS1" GTP Object Inspection, you configure the security device to log prohibited, rate-limited and state-invalid GTP packets. You opt for basic logging of prohibited and rate-limited packets, with a frequency value of 10 for the rate-limited packets, and extended logging for state-invalid packets.

WebUI

Objects > GTP > Edit (GPRS1) > Log: Enter the following, then click **Apply**:

```
Packet Prohibited: Basic (select)
Packet State-invalid: Extended (select)
Packet Rate-Limited: Basic (select)
When Packet Rate Limit is exceeded, log every other messages: 10
```

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set prohibited basic
(gtp:gprs1)-> set state-invalid extended
(gtp:gprs1)-> set rate-limited basic 10
(gtp:gprs1)-> exit
save
```

Traffic Counting

With the GTP traffic counting feature, you can configure the security device to tally the number of user data and control messages (or bytes of data), received from and forwarded to the GGSNs and SGSNs that it protects. The security device counts traffic for each GTP tunnel separately and differentiates GTP-User and GTP-Control messages. When a tunnel is deleted, the security device counts and logs the total number of messages or bytes of data that it received from and forwarded to the SGSN or GGSN.

The log entry for the deletion of a tunnel contains the following information:

- Timestamp
- Interface name (if applicable)
- SGSN IP address
- GGSN IP address
- TID
- Tunnel duration time in seconds
- Number of messages sent to the SGSN
- Number of messages sent to the GGSN

NOTE: By default, traffic logging is disabled on Juniper Networks security devices.

Example: Enabling GTP Traffic Counting

In this example, you enable GTP traffic counting by messages in the "GPRS1" GTP inspection object.

WebUI

Objects > GTP > Edit (GPRS1) > Log: Enter the following, then click **Apply**:

Traffic Counters: Count by Message (select)

CLI

```
set gtp config gprs1
(gtp:gprs1)-> log traffic-counters
(gtp:gprs1)-> exit
save
```

Lawful Interception

You can configure a security device to identify and log the contents of GTP-U or GTP-C messages based on IMSI prefixes or Mobile Station-Integrated Services Data Network (MS-ISDN) identification. You can identify subscribers by their IMSI or

MS-ISDN and log the content of user data and control messages going to and from the subscriber.

You can configure the number of subscribers that the security device can actively trace concurrently. The default number of simultaneous active traces is three. For GTP packets containing user data, you can specify the number of bytes of data to log. You can log partial or complete packets. The default value is zero, which means that the security device does not log any of the content from a GTP-U packet.

The security device sends the logged packets to an external server (such as Syslog) dedicated to Lawful Interception operations.

Example: Enabling Lawful Interception

In this example, you enable the security device to trace a subscriber with 345678 as an IMSI prefix in the "GPRS1" GTP inspection object. You also set the number of active traces to 2 and the number of bytes to log to 1064.

WebUI

Objects > GTP > Edit (GPRS1) > Subscriber Trace: Enter the following, then click **Apply**:

Maximum Simultaneous Active Trace: 2
Trace Message: 1064
Subscribers identified by: Select **IMSI**, enter **123456789012345**, then click **Add**.

CLI

```
set gtp config gprs1
(gtp:gprs1)-> set trace imsi 123456789012345
(gtp:gprs1)-> set trace max-active 2 save-length 1064
(gtp:gprs1)-> exit
save
```

Part 14
Dual-Stack Architecture with IPv6

Dual Stack Architecture with IPv6 describes ScreenOS support for Internet Protocol version 6 (IPv6) and how to secure IPv6 and IPv4/IPv6 transitional networks with tunneling and IPsec.

Dual-stack architecture allows an interface to operate simultaneously in IPv4 and IPv6 modes and facilitates network management, while a network contains both IPv4 and IPv6 devices that pass traffic between IPv4/IPv6 boundaries.

This guide contains the following sections:

- "Internet Protocol Version 6 Introduction" on page 2089 explains IPv6 headers, concepts, and tunneling guidelines.
- "IPv6 Configuration" on page 2097 explains how to configure an interface for operation as an IPv6 router or host.
- "Connection and Network Services" on page 2123 explains how to configure Dynamic Host Configuration protocol version 6 (DHCPv6), Domain Name Services (DNS), Point-to-Point Protocol over Ethernet (PPPoE), and fragmentation.
- "Static and Dynamic Routing" on page 2141 explains how to set up static and dynamic routing. This chapter explains ScreenOS support for Routing Information Protocol-Next Generation (RIPng).
- "Address Translation" on page 2173 explains how to use Network Address Translation (NAT) with dynamic IP (DIP) and mapped-IP (MIP) addresses to traverse IPv4/IPv6 boundaries.
- "IPv6 in an IPv4 Environment" on page 2189 explains manual and dynamic tunneling.
- "IPsec Tunneling" on page 2203 explains how to configure IPsec tunneling to connect dissimilar hosts.
- "IPv6 XAuth User Authentication" on page 2223 explains how to configure Remote Authentication Dial In User Service (RADIUS) and IPsec Access Session (IAS) management.
- "Switching" on page 2275 lists options for using the security device as a switch to pass IPv6 traffic.

Chapter 63

Internet Protocol Version 6 Introduction

ScreenOS supports Internet Protocol version 6 (IPv6), developed by the Internet Engineering Task Force (IETF).

NOTE: Some security devices support IPv6. Check the datasheet for your security platform to determine which features it supports.

This chapter contains the following sections:

Overview

By using addressing and schema that are different from IPv4, IPv6 allows a greater number of connected hosts than IPv4 can allow. In addition, IPv6 reduces packet processing overhead and increases network scalability. Together, these improvements allow a greater exchange of data traffic.

IPv6 provides for interoperability between IPv4 devices and IPv6 devices. It is usually possible to install IPv6 on security devices without losing IPv4 capability, so organizations can perform incremental upgrades and avoid service disruptions while migrating from IPv4 to IPv6.

NOTE: For more information about IPv6, refer to RFC 2460.

ScreenOS features dual-stack architecture, which allows an interface to operate simultaneously in IPv4 and IPv6 modes. Dual-stack architecture allows you to secure your network infrastructure while it contains both IPv4 and IPv6 devices and to secure traffic that passes across IPv4/IPv6 boundaries.

Each IPv6-enabled security device can operate as an IPv6 host or router.

IPv6 Addressing

IPv6 addresses differ from IPv4 addresses in several ways:

- Notation
- Prefixes
- Address Types

These differences give IPv6 addressing greater simplicity and scalability than IPv4 addressing.

Notation

IPv6 addresses are 128 bits long (expressed as 32 hexadecimal numbers) and consists of eight colon-delimited sections. Each section contains 2 bytes, and each byte is expressed as a hexadecimal number from 0 to FF.

An IPv6 address looks like this:

2080:0000:0000:0000:0008:0800:200c:417a

By omitting the leading zeroes from each section or substituting contiguous sections that contain zeroes with a double colon, you can write the example address as: 2080:0:0:0:8:800:200c:417a or 2080::8:800:200c:417a

For example, 0000:0000:0000:0000:0000:0000:93fc:9303 can be written as ::93fc:9303.

You can use the double-colon delimiter only once within a single IPv6 address. For example, you cannot express the IPv6 address 32af:0:0:0:ea34:0:71ff:fe01 as 32af::ea34::71ff:fe01.

Prefixes

Each IPv6 address contains bits that identify a network and a node or interface. An IPv6 prefix is the portion of an IPv6 address that identifies the network. The prefix length is a positive integer that denotes a number of consecutive bits, beginning with the most significant (left-most) bit. The prefix length follows a forward slash and, in most cases, identifies the portion of the address owned by an organization. All remaining bits (up to the right-most bit) represent individual nodes or interfaces.

For example, 32f1::250:af:34ff:fe26/64 has a prefix length of 64.

The first 64 bits of this address are the prefix (32f1:0000:0000:0000). The rest (250:af:34ff:fe26) identifies the interface.

Address Types

RFC 2373 describes three major categories of IPv6 addresses:

- Unicast
- Anycast
- Multicast

Unicast Addresses

A unicast address is an identifier for a single interface. When a network device sends a packet to a unicast address, the packet goes only to the specific interface identified by that address.

Devices use the following types of unicast addresses:

- A global unicast address is a unique IPv6 address assigned to a host interface. Global unicast addresses serve essentially the same purposes as IPv4 public addresses. Global unicast addresses are aggregatable for efficient and hierarchical addressing.
- A 6to4 address enables an IPv6 host interface for 6to4 tunneling. A 6to4 interface can serve as a border router between the host and IPv4 network space. In most cases, this method is not suitable for performing IPsec operations such as authentication and encryption.
- A link-local IPv6 address allows communication between neighboring hosts that reside on the same link. The device automatically generates a link-local address for each configured IPv6 interface.
- An IPv4-mapped address is a special IPv6 address that is the equivalent of an IPv4 address. A device uses IPv4-mapped addresses for address translation, when the device must send traffic from an IPv6 network to an IPv4 network.

Anycast Addresses

An anycast address is an identifier for a set of interfaces, which typically belongs to different nodes. When a network device sends a packet to an anycast address, the packet goes to one of the interfaces identified by that address. The routing protocol used in the network usually determines which interface is physically closest within the set of anycast addresses and routes the packet along the shortest path to its destination.

For more information about anycast addresses, refer to RFC 2526.

Multicast Addresses

A multicast address is an identifier for a set of interfaces, which typically belongs to different nodes. When a network device sends a packet to a multicast address, the device broadcasts the packet to all interfaces identified by that address.

Devices use the following types of multicast addresses:

- Solicited-node multicast addresses for Neighbor Solicitation (NS) messages.
- All-nodes multicast address for Router Advertisement (RA) messages.
- All-routers multicast address for Router Solicitation (RS) messages.

IPv6 Headers

The IETF designed IPv6 headers for low overhead and scalability. IPv6 headers allow optional extension headers, which contain extra information usable by network devices.

Basic Header

Every IPv6 packet has a basic IPv6 header. IPv6 headers occupy 40 bytes (320 bits). Figure 497 on page 2092 shows each field, arranged in order.

Figure 497: Header Structure

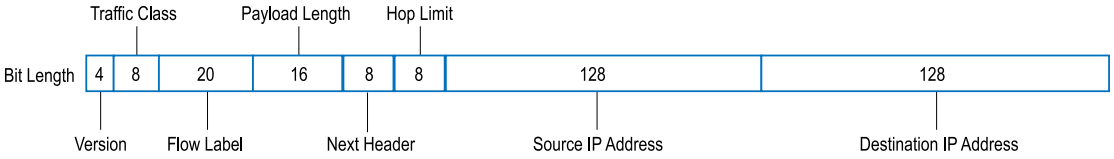


Table 139 on page 2092 lists the fields with bit lengths and their purposes.

Table 139: IPv6 Header Fields, Length, and Purpose

Field Name	Bit Length	Purpose
Version	4	Specifies the Internet Protocol used by the header and packet. This value tells destination internet devices which IP stack (IPv4 or IPv6) to use when processing the packet header and payload. IPv6 Version fields contain a value of 6. (IPv4 Version fields contain a value of 4.)
Traffic Class	8	Allows source nodes or routers to identify different classes (or priorities) of IPv6 packets. (This field replaces the IPv4 Type of Service field, which identified categories of packet transfer services.)
Flow Label	20	Identifies the flow to which the packet belongs. Packets in a flow share a common purpose, or belong to a common category, as interpreted by external devices such as routers or destination hosts. Typically, the source host inserts Flow Label values into outgoing packets to request special handling by the external devices. The external devices can uniquely identify each flow by evaluating the source address in combination with the Flow Label value. Traffic transmitted by a source host can contain packets in a single flow, multiple flows, no flow, or any combination. (Packets that do not belong to a flow carry a Flow Label of zero.)
Payload Length	6	Specifies the length of the of the IPv6 packet payload, expressed in octets.
Next Header	8	Identifies the type of IP protocol for the header that immediately follows the IPv6 header. This protocol can be one of two types: ■ An IPv6 extension header. For example, if the device performs IPsec security on exchanged packets, the Next Header value is probably 50 (ESP extension header) or 51 (AH extension header). Extension headers are optional. ■ An upper-layer Protocol Data Unit (PDU). For example, the Next Header value could be 6 (for TCP), 17 (for UDP), or 58 (for ICMPv6). The Next Header field replaces the IPv4 Protocol field. It is an optional field.

Table 139: IPv6 Header Fields, Length, and Purpose *(continued)*

Field Name	Bit Length	Purpose
Hop Limit	8	Specifies the maximum number of hops the packet can make after transmission from the host device. When the Hop Limit value is zero, the device drops the packet and generates an error message. (This field is similar the to Time to Live IPv4 field.)
Source IP Address	128	Identifies the host device that generated the IPv6 packet.
Destination IP Address	128	Identifies the intended recipient of the IPv6 packet.

Extension Headers

Extension headers contain supplementary information used by network devices (such as routers, switches, and endpoint hosts) to decide how to direct or process an IPv6 packet. The length of each extension header is an integer multiple of eight octets. This allows subsequent extension headers to use 8-octet structures.

Any header followed by an extension header contains a Next Header value that identifies the extension header type.

Extension headers always follow the basic IPv6 header in order as follows:

1. The Hop-by-Hop Options header specifies delivery parameters at each hop on the path to the destination host. When a packet uses this header, the Next Header value of the previous header (the basic IPv6 header) must be 0.
2. The Destination Options header specifies packet delivery parameters for either intermediate destination devices or the final destination host. When a packet uses this header, the Next Header value of the previous header must be 60.
3. The Routing header defines strict source routing and loose source routing for the packet. (With strict source routing, each intermediate destination device must be a single hop away. With loose source routing, intermediate destination devices can be one or more hops away.) When an packet uses this header, the Next Header value of the previous header must be 43.
4. The Fragment header specifies how to perform IPv6 fragmentation and reassembly services. When a packet uses this header, the Next Header value of the previous header must be 44.
5. The Authentication header provides authentication, data integrity, and anti-replay protection. When a packet uses this header, the Next Header value of the previous header must be 51.
6. The Encapsulating Security Payload header provides data confidentiality, data authentication, and anti-replay protection for encapsulated security payload (ESP) packets. When a packet uses this header, the Next Header value of the previous header must be 50.

IPv6 Packet Handling

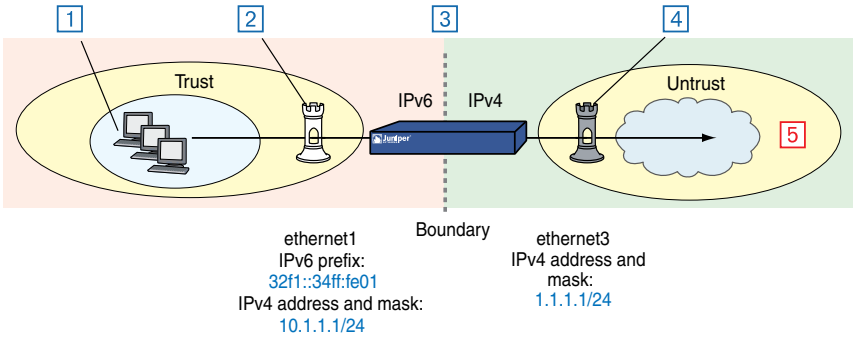
An interface configured for dual-stack operation provides both IPv4 and IPv6 capability. Such an interface can have an IPv4 address, at least one IPv6 address, or both.

If the interface resides at the boundary between an IPv4 network and an IPv6 network, the device can pass IP traffic over the boundary in one of two ways:

- Encapsulate (effectively hiding) any packet that passes across the boundary.
- Perform address translation on the packet source and destination addresses.

Figure 498 on page 2094 shows a packet-handling flow that might occur when an IPv6 host passes an outgoing service request packet across the IPv6/IPv4 boundary into an IPv4 network space.

Figure 498: Packet Flow Across IPv6/IPv4 Boundary



1. An IPv6 host transmits a service request packet. The source and destination addresses use IPv6 format.
2. IPv6 interface ethernet1 receives the packet.
3. The security device applies a policy to the packet. It either encapsulates the entire packet inside an IPv4 packet or translates the addresses to IPv4 format.
4. IPv4 interface ethernet3 transmits the packet (now using IPv4 address format).
5. A remote gateway node receives the packet.
   - If encapsulation occurred, the device might de-encapsulate the packet or continue to treat it as an IPv4 packet.
   - If IPv6/IPv4 address translation occurred, the device might translate the addresses back to IPv6 format or continue to treat the packet as an IPv4 packet.

IPv6 Router and Host Modes

You can configure each interface in a security device to function as an IPv6 host or router.

- In host mode, the interface functions as an IPv6 host and autoconfigures itself by requesting and accepting Router Advertisement (RA) messages from other devices.
- In router mode, the interface functions as an IPv6 router. An IPv6 router replies to Router Solicitation (RS) messages from IPv6 hosts by sending RAs. In addition, the interface can broadcast RAs periodically or in response to configuration changes to keep the on-link hosts updated.

IPv6 Tunneling Guidelines

Before deciding which kind of tunneling to use, ask your upstream ISP which IPv6 services they provide and how they provide them. We recommend the following guidelines:

- If your ISP provides only dual-stack IPv6, which is Internet Protocol Control Protocol (IPCP) and IPv6CP, you should configure run dual-stack, native IPv6. 6to4 addressing format is not appropriate in this case.
- If your ISP provides manual tunnel IPv6 (IPv6-over-IPv4 tunnel), you should use manual tunneling.
- If your ISP does not provide IPv6, go to www.hexago.com/ to find an upstream IPv6 provider, and follow their posted instructions.

For updates and service information about IPV6, visit one of the following websites:

- http://www.ipv6day.org/
- http://www.ipv6tf.org/
- http://www.ipv6forum.com/

NOTE: The above references are provided as effective resources as of the publication date of this document. However, we encourage administrators to seek out their own IPv6 references, which might be more current.

If you do not find an IPv6 provider, use 6to4 tunneling. This option, however, is only feasible if the next-hop router is configured for it.

Chapter 64

IPv6 Configuration

This chapter explains how to enable IPv6 features on the security device and how to configure the security device to act as an Internet Protocol version 6 (IPv6) router or IPv6 host.

This chapter contains the following sections:

Overview

ScreenOS allows you to configure a security device to be an IPv6 router or an IPv6 host.

This overview explains the following topics:

- Address autoconfiguration
- Neighbor discovery

The sections following the overview explain how to configure an IPv6 host or router.

Address Autoconfiguration

Address autoconfiguration allows local hosts to autoconfigure IPv6 addresses from their extended unique identifier (EUI) values. A security device configured for address autoconfiguration advertises an IPv6 prefix to local IPv6 hosts. The local hosts use this prefix to autoconfigure IPv6 addresses from their EUI values.

Address autoconfiguration reduces the need to manually assign addresses to individual hosts. Ideally, IPv6 hosts have address autoconfiguration enabled. An interface, configured to operate as an IPv6 router, can enable local on-link IPv6 hosts to perform

autoconfiguration. Autoconfiguration does not require a stateful configuration protocol, such as Dynamic Host Configuration Protocol version 6 (DHCPv6).

Extended Unique Identifier

An EUI address is a 64-bit hex interface identifier. If you do not specify an EUI value explicitly, the security device autogenerates it from the MAC address of the IPv6 interface. This usually happens immediately the first time you define an IPv6 interface.

A device configured for address autoconfiguration advertises an IPv6 prefix to local IPv6 hosts. The local hosts use this prefix to autoconfigure IPv6 addresses from their EUI-ID values.

NOTE: For more information about EUI, refer to *Guidelines for 64-Bit Global Identifier (EUI-64) Registration Authority* at http://standards.ieee.org/regauth/oui/tutorials/EUI64.html

Router Advertisement Messages

A Router Advertisement (RA) is a message sent by a router to on-link hosts periodically or in response to a Router Solicitation (RS) request from another host. The autoconfiguration information in an RA includes the following:

- IPv6 prefixes of the IPv6 router, which allow the on-link hosts to access the router
- Maximum Transmission Unit (MTU), which informs the on-link hosts the maximum size (in bytes) of exchanged packets
- Specific routes to the router, which allow the on-link hosts to send packets through the router
- Whether or not to perform IPv6 address autoconfiguration and, when appropriate, a prefix list
- Period that autoconfigured addresses remain valid and preferred

Router Solicitation Messages

A Router Solicitation (RS) is a message sent by hosts to discover the presence and properties of on-link routers. When an IPv6 router receives an RS request from a host, it responds by transmitting an RA message back to the host. An RA announces the existence of the router and provides the host with the information it needs to perform autoconfiguration tasks.

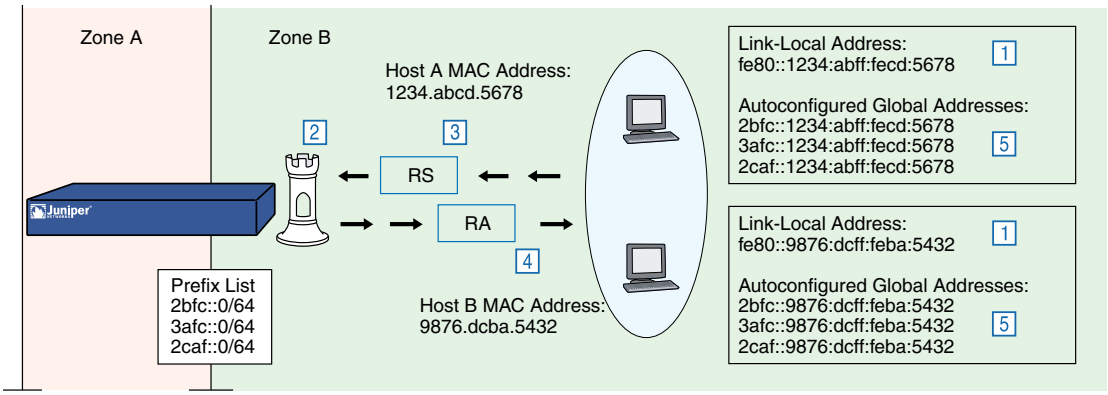
Each RS contains the link-local address of the source host. The host derives the link-local address from its MAC address. When the IPv6 router receives the RS, it uses the link-local address to transmit an RA back to the host.

Prefix Lists

A *prefix list* is a table containing IPv6 prefixes. When entries are present in the list, the router includes them in the RAs it sends to on-link hosts. Each time a host receives an RA, it can use the prefixes to perform address autoconfiguration. Figure 499 on

page 2099 shows Host A and Host B using three prefixes to generate unique global aggregatable addresses.

Figure 499: Address Autoconfiguration



1. On startup, IPv6 Hosts A and B generate link-local addresses from their MAC addresses.
2. Each host broadcasts RS messages. Each message uses the host link-local address as the source address for the RS packets.
3. The IPv6 router receives the RS message.
4. The IPv6 router transmits confirming RA messages to the hosts. These messages contain a prefix list.
5. The hosts use the prefixes to perform autoconfiguration.

Neighbor Discovery

Neighbor Discovery (ND) is the process of tracking the reachability status for neighbors in a local link. A device views a neighbor as reachable when the device receives recent confirmation that the neighbor received and processed IP traffic or Neighbor Solicitation (NS) requests. Otherwise, it considers the neighbor unreachable. Although not explicitly required, IPv6 host might have ND enabled. An IPv6 router with ND enabled can send ND information downstream.

Neighbor Cache Table

The Neighbor Cache table contains information about neighbors to which hosts have recently sent traffic. In addition, the table tracks the current reachability status of neighbors on the local link. Each entry contains the following information:

- IPv6 address of the neighbor
- MAC address of the neighbor
- Current neighbor reachability state
- Age of the neighbor entry
- Number of packets currently queued for transmission to the destination neighbor

Table entries are keyed on the IPv6 address. All entries are synced to the backup device in an NSRP cluster. When device failover occurs, the backup device becomes the primary and sends the NDP packet to the neighbor host to notify it of the change.

Neighbor Unreachability Detection

Neighbor Unreachability Detection (NUD) works by building and maintaining a Neighbor Cache table, which contains the address for each neighbor to which a host has recently sent traffic. The device uses these entries to record changes in the reachability status of the neighbors. NUD allows the device to track the changing reachability state of each neighbor and to make traffic-forwarding decisions.

Neighbor Entry Categories

A Neighbor Cache table entry can belong to any of four categories. The category of the entry determines how the device generates the entry initially and manages reachability states thereafter.

- **Endpoint host entries:** When an endpoint host makes an initial attempt to send traffic to a neighbor, the device automatically generates a corresponding entry in the Neighbor Cache table. The device uses this entry for further communication and to track the reachability state of the endpoint host.
- **Next-hop gateway router entries:** When you create a virtual routing table entry in a device for a gateway router, the device automatically generates a corresponding entry in the Neighbor Cache table. The device uses this entry for further communication and to track the reachability state of the gateway router.
- **Manual tunnel gateway interface entries**: When you set up a manual IPv6 over IPv4 tunnel interface, the device automatically generates an entry for the interface. This entry has an IPv6 link-local address. The device uses this entry to monitor the reachability state of the tunnel.
  - For information about IPv6 over IPv4 tunneling, see “IPsec 6in4 Tunneling” on page 2212.
  - For information about link-local addresses, see “Configuring Manual Tunneling” on page 2190.
  - For information about tunnel gateway transitions, see “Tunnel Gateway State Transitions” on page 2103.
- **Static entries**: When you create a Neighbor Cache entry statically, the device does not use it to perform ND or NUD operations. Instead, it assigns the entry a special reachability state called Static. This enables the entry in all circumstances. While the entry exists, the device forwards any traffic sent to the represented neighbor.

Neighbor Reachability States

No Neighbor Cache entry exists for a neighbor until the device sends the neighbor an initial NS request. Until this happens, the device does not recognize the existence of the neighbor. When the device sends the initial request, it creates a table entry and sets it to the Incomplete state.

The reachability states are as follows:

- **Incomplete:** A host attempted to send traffic to a neighbor currently unknown to the device, and initial address resolution is still in progress. The device broadcasts an NS request (using a solicited node multicast address) to find the neighbor, but has not yet received a confirming Neighbor Advertisement (NA).

  The Incomplete state has different characteristics when the neighbor is a next-hop gateway router. For more information, see "Next-Hop Gateway Router State Transitions" on page 2102.

- **Reachable:** The device currently considers the neighbor reachable because it received a confirming NA reply from the neighbor. While the entry state is Reachable, the device forwards any traffic sent to the neighbor. The entry state remains Reachable until the Reachable Time interval (expressed in seconds) elapses. Then the state changes to Stale.

- **Stale:** The device considers the neighbor unreachable because the Reachable Time interval has elapsed since the most recent NA from the neighbor. However, the device makes no attempt to verify reachability until a host attempts to send more traffic to the neighbor.

- **Delay:** A host attempted to send traffic to the neighbor while the state was Stale. The device makes no active attempt to verify neighbor reachability. Instead, it waits for upper-layer protocols to provide reachability confirmation. The device maintains the Delay state for five seconds. If the device receives confirmation during this delay period, the state changes to Reachable. Otherwise, the state changes to Probe.

- **Probe:** The Delay period elapsed, and the device received no confirmation from the upper-layer application. The device sends up to two unicast NS probes to verify reachability. If the device receives an NA message from the neighbor, the state changes to Reachable. Otherwise, the device deletes the reachability entry from the table. In effect, removal of a neighbor entry makes the device view the neighbor as nonexistent.

- **Probe Forever:** The device no longer considers the neighbor reachable, has made an attempt to forward traffic to the neighbor, and is sending unicast NS probes to verify reachability. The device continues to retransmit the probes indefinitely or until it receives a reachability confirmation from the neighbor.

  The device uses the Probe Forever reachability state *only* when the entry represents a next-hop gateway router.

NOTE: The Neighbor Cache entry might also exist in Active and Inactive states but only when the neighbor is a manual IPv6in4 tunnel gateway interface.

How Reachability State Transitions Occur

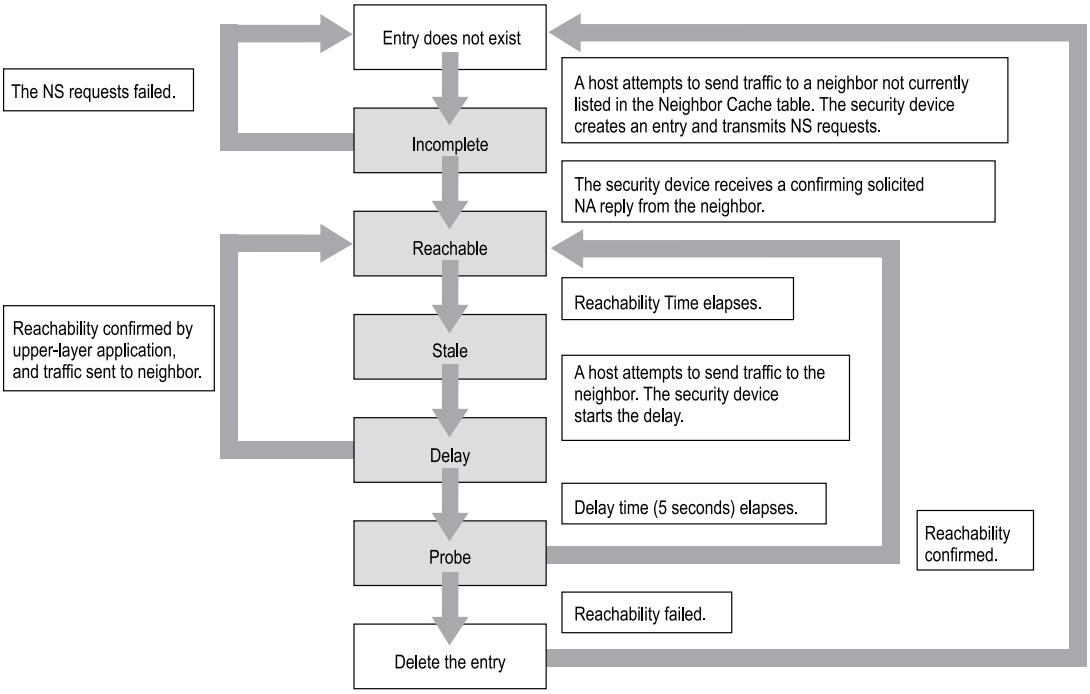
A device changes the reachability state of a Neighbor Cache entry depending on the neighbor category, the current state of the entry, and whether on-link hosts attempt to send traffic to the neighbor.

Endpoint Host State Transitions

When an on-link host attempts to send traffic to a neighbor, the device searches the Neighbor Cache table for a corresponding Neighbor Cache table entry. If no entry exists, the device broadcasts an NS message for the neighbor. It then creates a new table entry and assigns it an Incomplete state.

Figure 500 on page 2102 shows how the device handles reachability state transitions after it generates a Neighbor Cache entry.

Figure 500: Endpoint Host Reachability Transitions

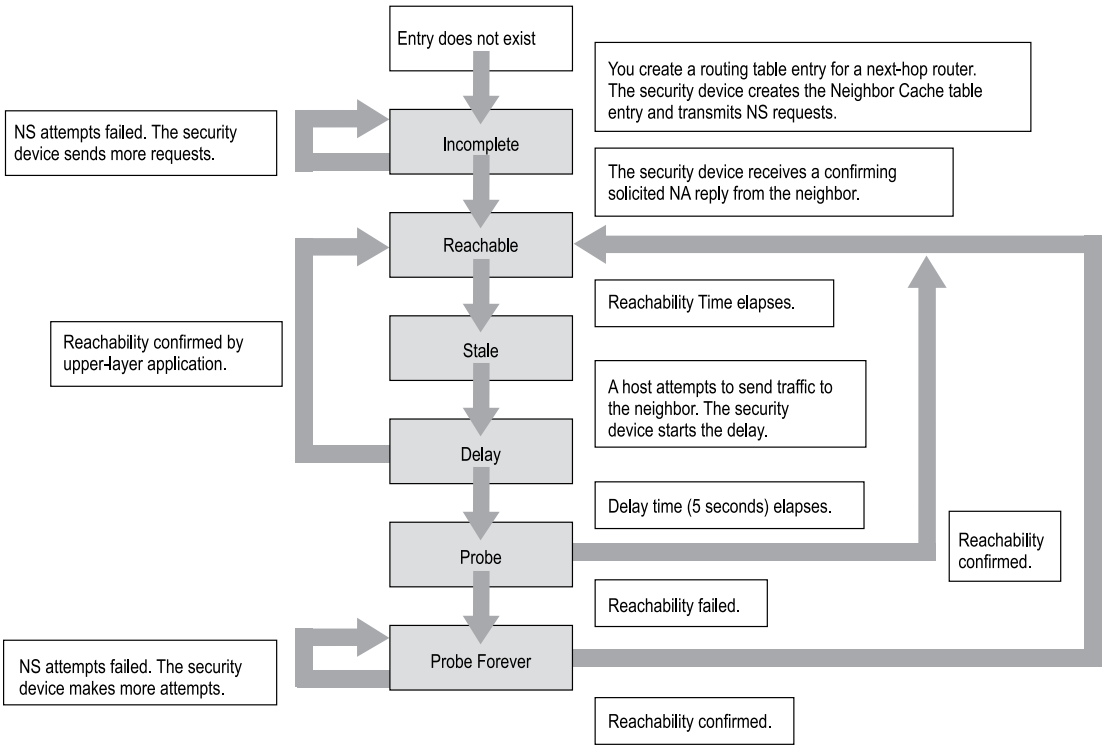


Next-Hop Gateway Router State Transitions

When you create a routing table entry to an IPv6 next-hop gateway router, the device automatically generates a corresponding Neighbor Cache table entry and assigns it an Incomplete state.

Figure 501 on page 2103 shows how the device handles reachability state transitions after it generates the Neighbor Cache entry for the first time.

Figure 501: Next-Hop Gateway Router Reachability Transitions



When you remove a router from which the device generated a Neighbor Cache table entry, the device deletes the entry automatically.

Tunnel Gateway State Transitions

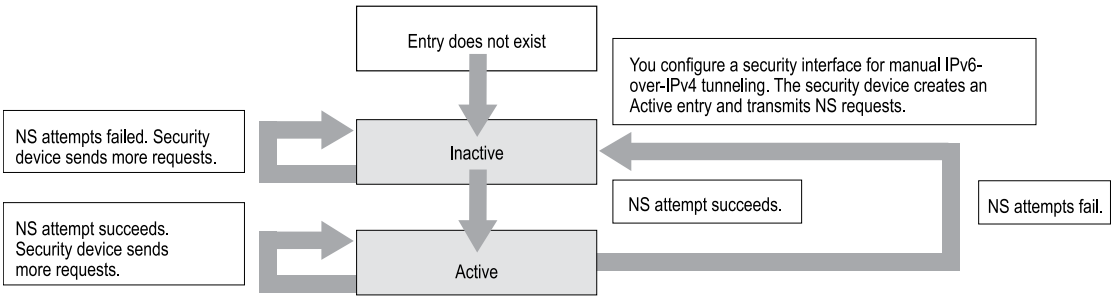
When you configure an interface for manual IPv6-in-IPv4 tunneling, the device automatically generates a corresponding entry in the Neighbor Cache table.

A device uses two entry states (also known as heartbeat states).

- **Inactive:** The device does not consider the neighbor reachable and has sent an NS message to test for reachability.
- **Active:** The device considers the neighbor reachable and periodically sends NS messages to confirm reachability.

When you create the IPv6-over-IPv4 tunnel, the device generates the Neighbor Cache entry and assigns it the Inactive state. Figure 502 on page 2104 shows how the device handles reachability state transitions after initial generation of the entry.

Figure 502: Tunnel Gateway State Transitions



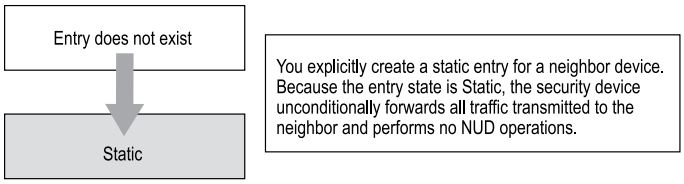
When you remove the manual tunnel from which the device generated the Neighbor Cache Table entry, the device deletes the entry automatically.

For information about manual IPv6 over IPv4 tunneling, see “IPsec Tunneling” on page 2203.

Static Entry Transitions

When you create a static entry, the device always forwards traffic transmitted to the neighbor because no NUD operations apply.

Figure 503: Static Entry State Transitions



Enabling an IPv6 Environment

To set up a security device for IPv6 operation, you must first enable an IPv6 environment variable on the device. You must complete this step otherwise you cannot view IPv6 features or options in the WebUI or the CLI.

Enabling IPv6 at the Device Level

To enable a device for IPv6, you must start a CLI session with the device. You can establish CLI sessions using software that emulates a VT100 terminal, such as Telnet or Secure Command Shell (SSH). If you make a direct connection through the console port, you can use HyperTerminal. (For more information about establishing CLI sessions, refer to the installation and configuration guide for the security device.

To check the IPv6 status of the security device, enter the following command:

```
get envar
```

If the device is currently IPv6-enabled, the following appears in the console output:

```
ipv6=yes
```

If this output does not appear, the device is not IPv6-enabled (default). To enable IPv6, enter the following commands:

```
set envar ipv6=yes
save
reset save-config yes
```

When the confirmation prompt appears, enter **y**.

Disabling IPv6 at the Device Level

You must start a CLI session with the device by establishing a console connection with HyperTerminal or another terminal emulation software.

To disable IPv6, enter the following commands:

```
unset envar ipv6
save
reset
```

When the confirmation prompt appears, enter **y**.

Configuring an IPv6 Host

After enabling the device for IPv6 operation, you can configure the device to be an IPv6 host by performing the following steps:

1. Bind the interface to a zone (such as Trust, Untrust, or a user-defined zone).
2. Enable the mode and interface.
3. Configure address autoconfiguration.
4. Configure neighbor discovery.

The following sections describe ScreenOS settings pertinent to IPv6 host configuration.

NOTE: Optionally, in addition to an IPv6 address, the security device can have an IPv4 IP address associated with the same interface. For information about IPv4 interface configuration, see *"Fundamentals" on page 15*.

Binding the IPv6 Interface to a Zone

You can bind an interface to a custom or preset security zone with the WebUI or the CLI. Interface naming varies by platform. To view the interfaces on your security device you can use the **get interface** command .

In the following example, you bind an interface named ethernet1/2 to the trust zone.

WebUI

Network > Interfaces: Select **Edit** to change the zone binding for an existing interface entry or click **New** to configure a new interface entry.

CLI

```
set interface ethernet1/2 zone trust
save
```

Enabling IPv6 Host Mode

You can enable IPv6 modes from the WebUI or the CLI. The mode options are: none (not using IPv6), host, or router.

WebUI

Network > Interfaces: Select **Edit** to change the IPv6 mode for an existing interface entry or click **New** to configure a new interface entry. Select **Host** mode.

Network > Interfaces > Edit (for ethernet1) > IPv6

CLI

```
set interface ethernet1/2 ipv6 mode host
set interface ethernet1/2 ipv6 enable
save
```

Setting an Interface Identifier

You can configure the Extended Unique Identifier (EUI) for the interface. The EUI is a 64-bit hexadecimal extension of the Ethernet Media Access Control (MAC) address. The device uses this value to autoconfigure an IPv6 link-local IP address for the interface.

WebUI

Network > Interfaces: Select **Edit** to change the zone binding for an existing interface entry, or click **New** to configure a new interface entry. Enter an **Interface ID**.

CLI

```
set interface ethernet1/2 ipv6 interface-id 0210dbfffe7ac108
save
```

Configuring Address Autoconfiguration

When you define a prefix list entry for address autoconfiguration, IPv6 on-link hosts can use the prefix to generate unique IPv6 addresses. In the following example, you define prefix list entry 2bfc::0/64.

WebUI

Network > Interfaces > Edit (for IPv6 interface) > Prefix lists: Enter the following, then click **OK**:

New IPV6 Prefix/Length: 2bfc::0/64
Prefix Flags
Autonomous: (select)
Onlink (select)

CLI

```
set interface ethernet3 ipv6 ra prefix 2bfc::0/64 autonomous onlink
```

After you make this setting, the device automatically includes the prefix in any Router Advertisement (RA) messages sent to on-link hosts.

Configuring Neighbor Discovery

To direct the interface to discover the existence and identity of other routers, you can enable the Accept Incoming RAs setting for the IPv6 interface. With this setting enabled, the interface accepts RA messages from other IPv6 peer devices.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Accept Incoming Router Advertisements: (select)

CLI

```
set interface ethernet3 ipv6 ra accept
```

After you enable this setting, the interface accepts any route advertisement it receives from another host in the link. When the interface receives such an advertisement, it stores the advertised IPv6 address and MAC address in the Neighbor Cache table.

To see if the interface received and stored any advertised routes, you can view the contents of the NDP table by executing the following command:

```
get ndp
```

Configuring an IPv6 Router

To configure an IPv6 router, perform the following steps:

1. Enable the IPv6 environment on the device.
2. Bind the interface to a zone (such as Trust, Untrust, or a user-defined zone).
3. Enable the mode and interface.
4. Configure address autoconfiguration.
5. Configure Router Advertisement (RA) parameters.
6. Configure Neighbor Discovery (ND) parameters.

The following sections describe ScreenOS settings pertinent to IPv6 router configuration.

NOTE: Optionally, in addition to an IPv6 address, the security device can have an IPv4 address associated with the same interface. For information about IPv4 interface configuration, see *"Fundamentals" on page 15*.

Binding the IPv6 Interface to a Zone

You can bind an interface to a custom or preset security zone with the WebUI or the CLI. Interface naming varies by platform.

In the following example, you bind an interface named ethernet1/2 to the trust zone.

WebUI

Network > Interfaces: Select **Edit** to change the zone binding for an existing interface entry or click **New** to configure a new interface entry.

CLI

```
interface ethernet1/2 zone trust
save
```

Enabling IPv6 Router Mode

You can enable IPv6 modes from the WebUI or the CLI. The mode options are: none (not using IPv6), host, or router.

WebUI

Network > Interfaces: Select **Edit** to change the IPv6 mode for an existing interface entry or click **New** to configure a new interface entry. Select **Router** mode.

Network > Interfaces > Edit (for ethernet1) > IPv6

CLI

```
set interface ethernet1/2 ipv6 mode router
set interface ethernet1/2 ipv6 enable
save
```

Setting an Interface Identifier

An IPv6 interface identifier sets the Extended Unique Identifier (EUI) for the interface. The EUI is a 64-bit hexadecimal extension of the Ethernet Media Access Control (MAC) address. The device uses this value to autoconfigure an IPv6 link-local IP address for the interface.

WebUI

Network > Interfaces: Select **Edit** to change the zone binding for an existing interface entry, or click **New** to configure a new interface entry. Enter an **Interface ID**.

CLI

```
set interface ethernet1/2 ipv6 interface-id 0210dbfffe7ac108
save
```

Setting Address Autoconfiguration

To set host autoconfiguration for an IPv6 router, you must do all of the following settings:

- Enable the Outgoing Router Advertisements.
- Disable the Managed Configuration Flag.
- Disable the Other Parameters Configuration Flag.

Outgoing Router Advertisements Flag

Enabling the Outgoing Router Advertisements flag allows the interface to send Router Advertisement (RA) messages to on-link hosts. After enabling this setting, the interface immediately broadcasts a route advertisement to hosts in the link. It also broadcasts an RA automatically when it receives a Router Solicitation (RS) from a host or when you change any RA setting on the interface.

In the following example, you enable the Allow RA Transmission setting.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Allow RA Transmission: (select)

CLI

```
set interface ethernet3 ipv6 ra transmit
```

Managed Configuration Flag

Enabling the Managed Configuration flag directs local hosts to use a stateful address autoconfiguration protocol, such as DHCPv6, to generate host addresses.

Local hosts cannot perform stateless address autoconfiguration while this setting is enabled.

In the following example, you disable the Managed Configuration flag to allow autoconfiguration.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Managed Configuration Flag: (deselect)

CLI

```
unset interface ethernet3 ipv6 ra managed
```

Other Parameters Configuration Flag

Enabling the Other Parameters Configuration flag directs local hosts to use a stateful address autoconfiguration protocol (DHCPv6) to configure parameters other than host addresses.

Local hosts cannot perform stateless address autoconfiguration while this setting is enabled.

In the following example, you disable the Other Parameters Configuration flag to allow autoconfiguration.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Other Parameters Configuration Flag: (deselect)

CLI

```
unset interface ethernet3 ipv6 ra other
```

Disabling Address Autoconfiguration

To disable host autoconfiguration for an IPv6 router, you must do the following:

- Disable the Outgoing Router Advertisements setting so that on-link host can't send router advertisements.
- Enable the Managed Configuration Flag to force the hosts to use a stateful addressing protocol, such as DHCPv6.
- Enable the Other Parameters Configuration Flag.

In the following example, you disable address autoconfiguration on an IPv6 router.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

```
Allow RA Transmission: (deselect)
Managed Configuration Flag: (select)
Other Parameters Configuration Flag: (select)
```

CLI

```
unset interface ethernet3 ipv6 ra transmit
set interface ethernet3 ipv6 ra managed
set interface ethernet3 ipv6 ra other
```

Setting Advertising Time Intervals

Address autoconfiguration uses several advertised time interval parameters for IPv6 routers. These intervals determine the frequency of events or the lifetime of identified objects.

Advertised Reachable Time Interval

Enabling the Reachable Time setting instructs the interface to include the Reachable Time interval in outgoing RA messages. This interval tells on-link hosts how long in seconds to consider the IPv6 interface reachable after they receive an RA from the interface.

The interface bases the Reachable Time interval on the current Base Reachable Time setting. For information about the Base Reachable Time, see "Base Reachable Time" on page 2116.

The interface uses this value while performing Neighbor Unreachability Detection (NUD). The security device builds and maintains a Neighbor Cache table, which contains the address for each neighbor to which a host has recently sent traffic. The device uses these entries to record changes in the reachability status of the neighbors. NUD allows the device to track the changing reachability state of each neighbor and to make traffic-forwarding decisions accordingly.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

```
Reachable Time: (select)
```

CLI

set interface ethernet3 ipv6 ra reachable-time

Advertised Retransmit Time Interval

Enabling the Retransmission Time instructs the interface to include the Retransmission Time interval in outgoing RA messages. This interval (expressed in seconds) is the time that elapses between retransmissions of NS messages.

For information about the Retransmission Time interval, see "Retransmission Time" on page 2117.

The interface uses this value while performing NUD.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Retransmission Time: (select)

CLI

set interface ethernet3 ipv6 ra retransmit-time

Maximum Advertisement Interval

The Maximum Advertisement interval specifies the maximum number of seconds allowed between transmission of unsolicited multicast RAs from the IPv6 interface.

In the following example, you set the interval to 500 seconds.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Maximum Advertisement Interval: 500

CLI

set interface ethernet3 ipv6 ra max-adv-int 500

Minimum Advertisement Interval

The Minimum Advertisement interval setting specifies the minimum number of seconds allowed between transmission of unsolicited multicast RAs from the IPv6 interface.

In the following example, you set the interval to 100 seconds.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Minimum Advertisement Interval: 100

CLI

```
set interface ethernet3 ipv6 ra min-adv-int 100
```

Advertised Default Router Lifetime

The Default Router Lifetime setting specifies the number of seconds that hosts can identify the interface to be the default router, after the hosts receive the last RA from the interface.

In the following example, you set the lifetime to 1500 seconds.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Minimum Advertisement Interval: 1500

CLI

```
set interface ethernet3 ipv6 ra default-life-time 1500
```

Advertising Packet Characteristics

An RA can provide on-link host devices with information about packets exchanged through the IPv6 interface, including the link MTU and the hop limit.

Link MTU Value

Enabling the Link MTU flag directs the IPv6 interface to include the Link MTU field in RA messages. The Link MTU is the maximum size (in bytes) of any IPv6 packet sent by a host over a link.

NOTE: The default Link MTU value for Ethernet is 1500 bytes. The default for PPPoE is 1490 bytes. This value must be from 1280 to 1500. You can change the MTU value for some platforms. Refer to the documentation for your security device to see if you can configure the MTU value.

In the following example, you enable advertisement of the link MTU.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Link MTU: (select)

CLI

set interface ethernet3 ipv6 ra link-mtu

Current Hop Limit

The Current Hop Limit setting specifies the hop limit for packets sent by any local IPv6 host that uses RAs from this interface for address autoconfiguration. Setting the Current Hop Limit value to zero denotes an unspecified number of hops.

In the following example, you set the Current Hop Limit value to 50 hops.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Current Hop Limit: 50

CLI

set interface ethernet3 ipv6 ra hop-limit 50

Advertising Router Characteristics

An IPv6 interface can use RAs to inform on-link hosts of router attributes, including the Link Layer (MAC) address of the interface and the router preference level.

Link Layer Address Setting

Enabling the Link Layer Address flag directs the IPv6 interface to include the Link Layer (MAC) address of the interface in outgoing RA messages.

In the following example, you enable the Link Layer Address flag.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Link Layer Address: (select)

CLI

set interface ethernet3 ipv6 ra link-address

Advertised Router Preference

The Advertised Router Preference setting specifies the preference level for the router.

When a host receives the RA from the IPv6 interface, and other IPv6 routers are present, the preference level determines whether the host views the interface as a primary router or a secondary router.

In the following example, you set the preference to **High**, which designates the interface as the primary router.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Advertised Router Preference: High (select)

CLI

```
set interface ethernet3 ipv6 ra preference high
```

Configuring Neighbor Discovery Parameters

When you configure an interface for IPv6 operation, the interface must locate and confirm the existence of neighbors on the same link as the interface. Neighbor Discovery (ND) allows a device to track the reachability status for neighbors in a local link.

Neighbor Unreachability Detection

Enabling Neighbor Unreachability Detection (NUD) directs the interface to perform NUD. Each entry in the table represents a neighbor and contains the current reachability status of the neighbor.

In the following example, you enable NUD for IPv6 interface ethernet3.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

NUD (Neighbor Unreachability Detection): (select)

CLI

```
set interface ethernet3 ipv6 nd nud
```

MAC Session-Caching

By default, the interface uses MAC address session-caching to increase the speed and efficiency of address resolution. Enabling the always-on-dest setting instructs

the interface to bypass this process. Before transmitting traffic to a neighbor, the interface always consults the Neighbor Cache table instead of caching the MAC address in the session.

To bypass MAC session caching, enter the following CLI command:

```
set ndp always-on-dest
```

Static Neighbor Cache Entries

To create a static entry in the Neighbor Cache table, set the NDP (Neighbor Discovery Parameter) using the CLI command **set ndp**.

```
set ndpip_addr mac_addr interface
```

where:

- *ip_addr* is the neighbor IPv6 address.
- *mac_addr* is the neighbor MAC address.
- *interface* is the device interface.

In the following example, you start a new entry in the Neighbor Cache table.

- Neighbor IP address 32f1::250:af:34ff:fe27
- MAC address 1234abcd1234
- Device interface ethernet3

```
set ndp 32f1::250:af:34ff:fe27 1234abcd1234 ethernet3
```

To delete the entry, enter the **unset ndp** *ip_addr interface* command.

Base Reachable Time

When an IPv6 interface transmits a Neighbor Solicitation (NS) message to a neighbor and receives a Neighbor Advertisement (NA) message in reply, the device sets the neighbor reachability status to Reachable. The Base Reachable Time setting specifies the approximate length of time (expressed in seconds) that the interface maintains the Reachable status.

After this time interval passes, the status goes to stale mode.

NOTE: The Base Reachable Time setting only specifies an approximation of the actual time that the status remains Reachable. The exact time interval is called Reachable Time. The interface determines Reachable Time randomly, using the Base Reachable Time as a baseline value. The resulting Reachable Time is usually within 50 to 150 percent of the Base Reachable Time setting.

In the following example, you set the Base Reachable Time (for IPv6 interface ethernet3) to 45 seconds.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Base Reachable Time: 45

CLI

```
set interface ethernet3 ipv6 nd base-reachable-time 45
```

Probe Time

While an entry status is Incomplete or Probe Forever (as when the neighbor is a next-hop gateway), the interface attempts to confirm the reachability of the neighbor. Each attempt is called a *probe*. During a probe, the interface transmits NS requests to the neighbor. The endpoint host state mode Probe Time setting specifies the interval of time (expressed in seconds) between probes.

In the following example, you set the Probe Time (for IPv6 interface ethernet3) to 3 seconds.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Retransmission Time: 3

CLI

```
set interface ethernet3 ipv6 nd probe-time 3
```

Retransmission Time

When an IPv6 interface begins a probe, it transmits NS requests to the neighbor. The Retransmission Time setting specifies the time interval (expressed in seconds) that elapses between NS requests.

The following example sets the Retransmission Time (for IPv6 interface ethernet3) to 2 seconds.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

Retransmission Time: 2

CLI

set interface ethernet3 ipv6 nd retransmit-time 2

Duplicate Address Detection Retry Count

Duplicate Address Detection (DAD) determines if more than one on-link device has the same unicast address. The DAD Retry Count setting specifies the number of consecutive NS messages to send while performing DAD for the IPv6 interface.

The following example sets the DAD Retry Count (for IPv6 interface ethernet3) to 2 retries.

WebUI

Network > Interfaces > Edit (for IPv6 interface): Enter the following, then click **OK**:

DAD (Duplicate Address Detection) Retry Count: 2

CLI

set interface ethernet3 ipv6 nd dad-count 2

Viewing IPv6 Interface Parameters

To view IPv6 configuration information:

WebUI

Network > Interfaces > IPv6

CLI

get interface

Viewing Neighbor Discovery Configurations

You can view the current Neighbor Discovery (ND) settings using the WebUI or the CLI. The following example displays the current ND settings for an IPv6 interface (ethernet3).

WebUI

Network > Interfaces > Edit (for IPv6 interface)

CLI

get interface ethernet3 ipv6 config

Viewing the Current RA Configuration

Before configuring an IPv6 interface to function as an IPv6 router, we recommend that you check the current RA configuration.

In the following example, you display the RA configuration settings for IPv6 interface ethernet3.

WebUI

Network > Interfaces > Edit (for IPv6 interface ethernet3): View the current settings.

CLI

get interface ethernet3 ipv6 ra

Multicast Listener Discovery Protocol

Multicasting is the process of sending information to a set of interested listeners on a specific multicast address. The set of hosts or clients that listen on a specific multicast address is called a multicast group. Multicast traffic is sent to a single multicast address, and host computers that belong to the multicast group receive the traffic that is sent to the group's address.

An IPv6 router uses the Multicast Listener Discovery (MLD) protocol to discover the presence of multicast listeners (clients) on its directly attached links and to specifically learn which multicast address they are interested in joining by sending MLD messages. MLD messages are used within the network segment to exchange membership status information between the router and the hosts. Table 2 describes the types of MLD messages.

Table 140: Multicast Listener Discovery (MLD) Messages

MLD message	Description
Multicast Listener Query	Sent by a multicast router to request or poll for group membership from the clients in a network segment.
Multicast Listener Report	Sent by a client in response to an MLD Multicast Listener Query sent by a multicast router.
Multicast Listener Done	Sent by a client when it is not interested in joining the multicast group.

To enable MLD on an interface.

WebUI

Network > Interfaces > Edit: Click **MLD**
MLD: (select)
MLD enable: (select)

CLI

set interface interface-name protocol MLD enable

By default, every MLD-enabled interface will support host mode. If the interface is in IPv6 router mode, the interface acts as an MLD router also. In this way the MLD interface acts in both host and router mode at the same time.

To support the MLD functionality, the ScreenOS security device will act as both the multicast client and multicast router. As a multicast router, the security device sends periodical MLD query packets on IPv6 interfaces to poll for group members. As a multicast client it sends multicast listener report to all MLD enabled interfaces, specifying the multicast address the nodes are interested in joining.

Table 3 describes the IPv6 multicast addresses currently supported by ScreenOS:

Table 141: Multicast Address

Address	Description
FFO2::2	Link local scope routers
FFO2::1	Link local scope hosts
FF02:0:0:0:0:1:FFXX:XXXX	Solicited node multicast address

The multicast client sends an MLD multicast listener report to the multicast router in the following three situations:

- When an interface changes its state from DOWN to UP it sends a report to the multicast router specifying all the multicast addresses it is interested in listening to on that particular segment.
- When a security device sends a Multicast Listener Query, the client sends an MLD multicast listener report to the router specifying all the multicast address it is listening to.
- When an unicast or anycast address of the interface is changed, the device sends an unsolicited report for a solicited node multicast address.

You can view MLD information related to an interface and to a multicast group by using the **get mld interface interace-name** command and **get mld group group address** command, respectively.

To clear information and statistics of groups in router mode interface:

```
clear mld interface interface-name [ group group-address | statistics ]
```

Configuration Examples

This section contains examples of IPv6 router and host configuration. Only CLI commands are shown.

IPv6 Router

This example shows a security device with two interfaces that operate in IPv6 router mode:

CLI

```
set interface ethernet2 zone trust
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 interface-id 1111111111111111
set interface ethernet2 ipv6 ip 2eee::1/64
set interface ethernet2 ipv6 ra transmit
set interface ethernet3 zone trust
set interface ethernet3 ipv6 mode router
set interface ethernet3 ipv6 enable
set interface ethernet3 ipv6 interface-id 2222222222222222
set interface ethernet3 ipv6 ip 3eee::1/64
set interface ethernet3 ipv6 ra transmit
```

IPv6 Host

This following commands configure an IPv6 host:

CLI (Device B)

```
set interface ethernet2 zone trust
set interface ethernet2 ip 20.1.1.2/24
set interface ethernet2 ipv6 mode host
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 interface-id 3333333333333333
set interface ethernet2 ipv6 ra accept
```

Chapter 65

Connection and Network Services

IPv6-enabled interfaces support Dynamic Host Configuration Protocol version 6 (DHCPv6) and Point-to-Point Protocol over Ethernet (PPPoE).

This chapter contains the following sections:

Overview

This chapter explains the following network and connection services:

- Dynamic Host Configuration Protocol version 6 (DHCPv6)
- Domain Name System (DNS)
- Point-to-Point over Ethernet (PPPoE)
- Fragmentation

The next sections provide an overview of each network or connection service.

Dynamic Host Configuration Protocol Version 6

An IPv6 router can only be a DHCPv6 server. An IPv6 host can only be a DHCP client.

As a DHCPv6 client, the interface can request (from a DHCPv6 server):

- Delegation of long-lived prefixes across an administrative boundary. The server does not have to know the topology of the targeted local network. For example, an ISP can use DHCPv6 to assign prefixes to downstream networks through downstream DHCP clients. To speed up the client/server interaction, the client can request rapid commit (if enabled). Rapid commit reduces the number of messages from four to two.

NOTE: For more information about the impact of using rapid commit options, refer to RFCs 3315 and 3633.

- IP addresses of available DNS servers. The interface can also request DNS search-list information. This list contains partial domain names, which assist DNS searches by concatenating entered usernames to the domain names.

As a DHCPv6 server, the interface can provide both of these services to a DHCPv6 client. To speed up prefix delegation, an IPv6 router configured to be a DHCPv6 server can support a rapid commit option. You can also set a server preference option.

NOTE: DHCPv6 is complementary to address autoconfiguration. The services are not interchangeable.

Device-Unique Identification

A Device-Unique Identification (DUID) identifies a network device such as a host or a router. You can use the DUID (or a name associated with the DUID) to determine how the device performs prefix delegation for a particular router or host that has a particular DUID.

Identity Association Prefix Delegation-Identification

An Identity Association Prefix Delegation-Identification (IAPD-ID) is a positive integer that identifies a certain prefix on the server. The client can use this value to request a specific prefix from the server. If you specify a nonzero IAPD-ID value, it maps statically to a single prefix that has the same IAPD-ID on the server. If you specify a zero IAPD-ID value or omit the value (which assigns it a zero value by default) the IAPD-ID maps it dynamically to a pool of prefixes on the server. The device treats all prefixes with a zero IAPD-ID as belonging to this pool.

A DHCPv6 server can contain up to 16 prefixes per client DUID.

Prefix Features

Prefixes delegated by a DHCPv6 server contain the following elements:

- The Top-Level Aggregator (TLA) identifies the highest level in the routing hierarchy (the left most portion of the IPv6 address). The TLA usually specifies owned address space for an organization, such as an ISP. You specify the length of the TLA using the prefix length.
- The Site-Local Aggregator (SLA) identifies a network or subnet used by the organization.

For example, a server might delegate the following prefix:

```
2001:908e:1::/48
```

where:

- TLA is 2001:908e
- SLA is 1

The same server might also delegate the following prefix:

```
2001:908e:2::/48
```

where:

- TLA is 2001:908e
- SLA is 2

The SLAs differ in each example, which means that each prefix maps to a different network but belongs to the same owned address space.

Server Preference

You can optionally send a server preference value in the DHCP advertise messages that the device sends to DHCPv6 clients. A DHCPv6 client uses this value to choose one server instead of another.

The default value is –1 and indicates that no preference appears in the advertise messages sent to clients. You can set the value from 0 (lowest priority) to 255 (highest priority).

In the following example, you set the server preference for interface ethernet1/2 to 255, the highest priority.

WebUI

Network > DHCPV6 > Edit (Server): Enter 255 in the Server Preference Number field, then click **Apply**.

CLI

```
set interface ethernet1/2 dhcp6 server preference 255
```

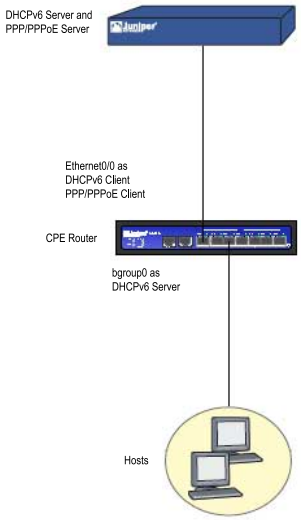
Dynamic IPv6 Prefix and DNS Information Update

In some cases, a CPE router acting as both a DHCPv6 client and a PPPoE client is configured to negotiate IPv6 prefixes and DNS information for the downstream DHCPv6 server on the other interface of the same CPE router.

In Figure 504 on page 2126, for example, the CPE router acts as a DHCPv6 and PPP/PPPoE client on ethernet0/0 and as a DHCPv6 server on bgroup0. It initiates PPPoE connection through ethernet0/0 to request IPv6 prefixes and DNS information from the upstream DHCPv6 server. The learned prefix and DNS information are

updated to the downstream DHCPv6 server on bgroup0 of the same CPE router. If the PPPoE connection is disconnected and then reestablished, the IPv6 prefix and the DNS information on the upstream DHCPv6 server will be changed. The CPE router cannot update this information on the downstream DHCPv6 server on bgroup0 unless the already delegated prefix expires. To avoid this, ScreenOS supports dynamic IPv6 prefix and DNS information updates so that the CPE router is not required to wait until the delegated prefix expires to renew the information on the downstream DHCPv6 server. This feature is also supported on Ethernet and PPP connections.

Figure 504: CPE Router Acting As Both DHCPv6 Client and PPPoE Client

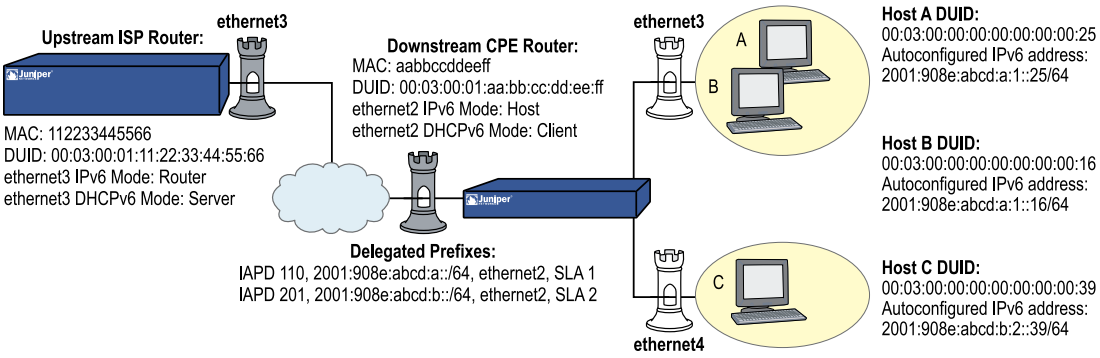


Configuring a DHCPv6 Server

In environments, such as an ISP, where you allow end users access to outside networks, you can set up security devices to delegate different prefixes with different SLAs. Figure 505 on page 2127 shows an upstream IPv6 router set to delegate DHCPv6 prefixes to downstream routers.

The IPv6 router delegates two prefixes, each with a different SLA. Downstream CPE routers are able to use the two different SLAs to assign prefixes to two different networks.

Figure 505: DHCPv6 Prefix Delegation



In this example, you configure interface ethernet3 as a DHCPv6 server and specify two prefixes on the upstream ISP router. The prefixes are then available for delegation to a downstream Customer Premises Equipment (CPE) router.

- Specifies downstream client DUID 00:03:00:01:aa:bb:cc:dd:ee:ff. This allows the client to request prefix delegation from the server.
- Delegates the following prefixes:
  - 2001:908e:abcd:a::/64, SLA 1, IAPD 110, preferred lifetime 259200 seconds, valid lifetime 345600 seconds
  - 2001:908e:abcd:b::/64, SLA 2, IAPD 201, preferred lifetime 172800 seconds, valid lifetime 345600 seconds
- Because of the nonzero IAPD numbers, the server delegates both prefixes statically (not dynamically).

NOTE: The WebUI section lists only the navigational paths to the device configuration pages. For specific values, see the CLI section that follows it.

WebUI

1. **Server Interface**

   Network > Interfaces > Edit (for ethernet3) > IPv6

   Network > DHCPV6 > Edit (for ethernet3)

2. **Downstream Client Identification**

   Network > DHCPv6 > DUID/Prefix Delegation List (for ethernet3) > New DUID

3. **Delegated Prefixes**

   Network > DHCPv6 > DUID/Prefix Delegation List (for ethernet3) > Add Prefix Entry

CLI

1. **Server Interface**

```
set interface ethernet3 ipv6 mode router
set interface ethernet3 ipv6 enable
set interface ethernet3 dhcp6 server
```

2. **Downstream Client Identification**

```
set interface ethernet3 dhcp6 server options client-duid
  00:03:00:01:aa:bb:cc:dd:ee:ff
```

3. **Delegated Prefixes**

```
set interface ethernet3 dhcp6 server options pd duid
  00:03:00:01:aa:bb:cc:dd:ee:ff iapd-id 110 prefix 2001:908e:abcd:a::/64
  259200 345600
set interface ethernet3 dhcp6 server options pd duid
  00:03:00:01:aa:bb:cc:dd:ee:ff iapd-id 201 prefix 2001:908e:abcd:b::/64
  172800 345600
save
```

Configuring a DHCPv6 Client

In the following example, you configure a downstream CPE router to be a DHCPv6 client. The router can then request prefixes from the upstream router, and automatically push IPv6 addresses derived from the prefixes to local hosts. See Figure 505 on page 2127.

- Specifies preferred server with DUID 00:03:00:01:11:22:33:44:55:66
- Requests the following preferred prefixes from the server:
  - 2001:908e:abcd:a::/64, SLA 1, IAPD 110, preferred lifetime 259200 seconds (3 days), valid lifetime 345600 seconds (4 days)
  - 2001:908e:abcd:b::/64, SLA 2, IAPD 201, preferred lifetime 172800 seconds (2 days), valid lifetime 345600 seconds (4 days)
- Designates ethernet3 and ethernet4 as downstream interfaces.
  - Prefixes with SLA of 1 go to ethernet3
  - Prefixes with SLA of 2 go to ethernet4

NOTE: The WebUI section lists only the navigational paths to the device configuration pages. For specific values, see the CLI section that follows it.

WebUI

1. **Downstream CPE Interfaces**

   Network > Interfaces > Edit (for ethernet4)

   Network > Interfaces > Edit (for ethernet4) > IPv6

   Network > Interfaces > Edit (for ethernet3)

   Network > Interfaces > Edit (for ethernet3) > IPv6

2. **Client Interface (to Upstream Router)**

   Network > Interfaces > Edit (for ethernet2)

   Network > Interfaces > Edit (for ethernet2) > IPv6

   Network > DHCPv6 > Edit (for ethernet2)

3. **Preferred Prefixes (from Upstream Router)**

   Network > DHCPv6 > Prefix Assignment List (for ethernet2) > Learned (Suggested) Prefix add

4. **Downstream Interfaces**

   Network > DHCPv6 > Prefix Assignment List (for ethernet2) > Prefix Distribution add

CLI

1. **Downstream CPE Interfaces**

```
set interface ethernet4 zone trust
set interface ethernet4 ipv6 mode router
set interface ethernet4 ipv6 enable
set interface ethernet3 zone trust
set interface ethernet3 ipv6 mode router
set interface ethernet3 ipv6 enable
```

2. **Client Interface (to Upstream Router)**

```
set interface ethernet2 ipv6 mode host
set interface ethernet2 ipv6 enable
set interface ethernet2 dhcp6 client
set interface ethernet2 dhcp6 client options request pd
set interface ethernet2 dhcp6 client prefer-server 00:03:00:01:11:22:33:44:55:66
```

3. **Preferred Prefixes (from Upstream Router)**

```
set interface ethernet2 dhcp6 client pd iapd-id 110 prefix 2001:908e:abcd:a::/64
  259200 345600
```

```
set interface ethernet2 dhcp6 client pd iapd-id 201 prefix 2001:908e:abcd:b::/64
  172800 345600
```

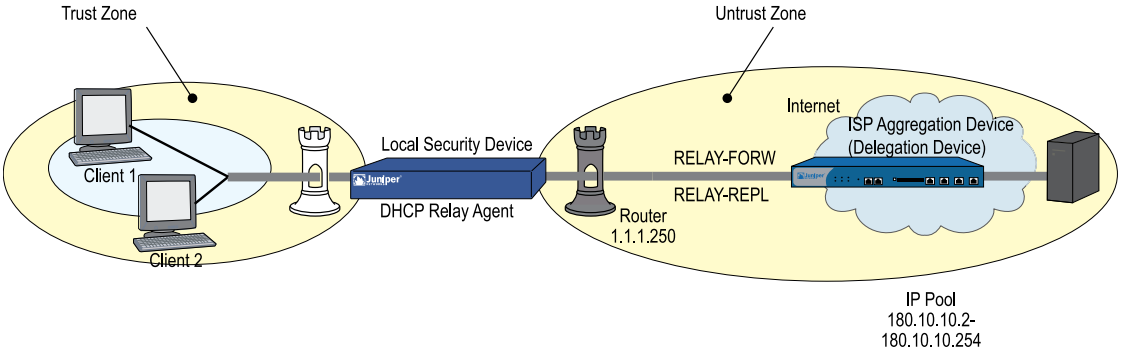
4. **Downstream Interfaces**

```
set interface ethernet2 dhcp6 client pd iapd-id 110 ra-interface ethernet3 sla-id
  1 sla-len 2
set interface ethernet2 dhcp6 client pd iapd-id 201 ra-interface ethernet4 sla-id
  2 sla-len 2
set interface ethernet2 dhcp6 client enable
save
```

Configuring DHCPv6 Relay Agent

When a DHCPv6 client sends a message to a DHCPv6 server that is not attached to the same link as the client, a DHCPv6 relay agent configured on the client's link will relay the messages between the client and the server. You can configure up to three servers for a relay agent. The relay agent unicasts each message from the client to all three servers at the same time.

Figure 506: Configuring DHCPv6 Relay Agent



You can set up the relay agent on any interface that has an IPv6 address and is in router mode. The following sections explain the steps involved in configuring a DHCPv6 relay agent on the device.

Setting up a DHCPv6 relay agent

1. Enable IPv6:

```
set env ipv6=yes
```

2. Reset the security device.
3. Enable IPv6 on interface1:

```
set interface interface1 ipv6 mode router
set interface interface1 ipv6 ip ipv6 address/num
```

Setting up DHCPv6 relay

You can set up the DHCPv6 relay with the WebUI or CLI:

WebUI:

1. **Enable the DHCPv6 relay:**

   Network > DHCPv6

   DHCP Relay Agent: (Select)

2. **Setting Up Server IP Addresses:**

   Relay agent Server IP: Enter the server IP addresses to which the relay agent has to relay the messages in the respective fields.

CLI

1. **Enable the DHCPv6 relay:**

   set interface interface1 dhcp6 relay enable

2. **Setting Up Server IP Addresses:**

   set interface interface1 dhcp6 relay server-ip *IPv6 Address*

Relay Agent Behavior

A relay agent relays messages from the client to the server and also relays Relay-Forward messages from other relay agents.

When a relay agent receives a message to be relayed, it constructs a Relay-Forward message to be sent to the server. The following table defines the fields in a Relay-Forward message:

msg-type	RELAY-FORW
hop-count	Number of relay agents that have relayed this message.
link-address	Address used by the server to identify the link on which the client is located.
peer-address	The address of the client or relay agent from which the message to be relayed was received.

The Relay-Forward message includes a Relay Message option to relay messages either to the server or to other relay agent. In addition to this, the relay agent can include

an Interface-id option in the Relay-Forward message, to identify the interface on which the client message was received. The server will include this interface-id option in its Relay-Reply message.

When the client sends a message to the relay agent, the relay agent will fill the link address field of the Relay-Forward message with the address of the client's interface from which the message was received. It sets the hop-count value in the Relay-Forward message to 0. The link-address is used by the server to identify the link on which the client is located.

When the relay agent receives a Relay-Forward message from another relay agent, it copies the source address of the client, which it extracts from the IP datagram into the peer-address field in the Relay-Forward message. It also sets the hop count value to the value of hop count field in the received message incremented by 1.

NOTE: If the message received by the relay agent is a Relay-Forward message and the hop count in the message is greater than or equal to the HOP_COUNT_LIMIT, the relay agent will discard the received message.

Server Behavior

When a server receives a Relay-Forward message, it sends a Relay-Reply message to the relay agent containing a response message to the client. The relay agent relays the response message to the client. The following table defines the fields in a Relay-Reply message:

msg-type	RELAY-REPL
hop-count	Copied from the Relay-Forward message
link-address	Copied from the Relay-Forward message
peer-address	Copied from the Relay-Forward message

The Relay-Reply message includes a Relay Message option to relay messages to the relay agent. The server encapsulates the client message as an option in the Relay-Reply message, which the relay agent extracts and relays to the client.

If more than one relay agent was involved when sending the Relay-Forward message to the server, the server follows the exact path when returning the response message to the client. The server creates a Relay-Reply message that includes a Relay Message option containing the message for the next relay agent in return path to the client. That Relay-Reply message contains another Relay Message option to be sent to the next relay agent and so on. The server records the contents in the peer address fields of the received messages and constructs Relay reply messages carrying the response from the server.

NOTE: The DHCPvs servers should be able to identify the two types of messages. They should respond to the Relay-forward messages and discard the Relay-reply messages.

Both the relay agent and server supports two specific options:

- Relay Message
- Interface ID

These two options appear only in the Relay-forward and Relay-reply messages.

Relay Message Option

In a Relay-Forward message, the Relay Message option carries the message received from the client and relays it verbatim to the next relay or the server.

In a Relay-Reply message the Relay Message option carries the message to be copied and relayed to the relay agent or client whose address is in the peer-address field of the Relay-Reply message.

Interface-ID option

Interface-id option is an optional value that can be sent by the relay agent to identify the interface on which the client message was received. The server copies the Interface-Id option from the Relay-Forward message into the Relay-Reply message and sends it to the relay agent in response to the Relay-Forward message.

Viewing DHCPv6 settings

You can view DHCPv6 relay agent settings from the command line interface. To view DHCPv6 relay settings:

get interface *interface_name* dhcp6 relay

Sample output:

```
device-> get interface e0/3 dhcp6 relay
DHCPv6 Configuration, enabled.
------------------------------------------------------------------------------
        Mode           : relay
        Interface-id   : enabled
        Server-ip list : 3000::217:cbff:fef7:e307/48
                         3002::a09:cbff:fef4:1219/48
                         3003::f21:cbff:fef5:f521/48
.
------------------------------------------------------------------------------
```

Viewing DHCPv6 Settings

You can view DHCPv6 server and client settings from the command line interface.

To view DHCPv6 settings:

```
get interface interface_name dhcp6
```

Sample output:

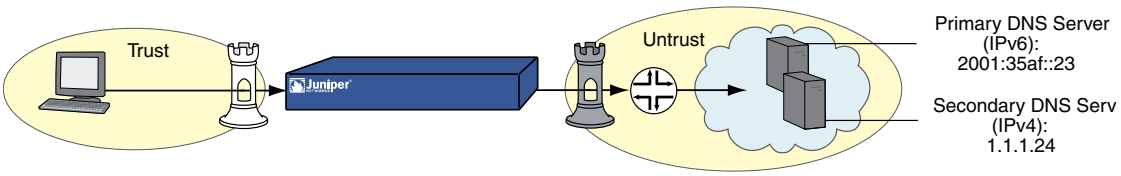
```
device-> get interface ethernet1/2 dhcp6
DHCPv6 Configuration, enabled.
------------------------------------------------------------------
        Mode             : server
        DUID             : 00:03:00:01:00:10:db:7a:c1:08
        Interface        : ethernet1/2
        Allow rapid-commit: yes
        Preference       : 255
------------------------------------------------------------------
```

Configuring Domain Name System Servers

In order to use Domain Name System (DNS) for domain name-to-address resolution, you must configure the device by entering an IP address for a primary and secondary DNS server. The DNS servers you choose to configure might use IPv4 or IPv6 addresses.

Figure 507 on page 2134 shows a primary and secondary DNS server in the Untrust zone behind a router.

Figure 507: Domain Name System Servers



In the following example, you designate an IPv6 server as the primary server and an IPv4 server as the secondary server.

- Primary server residing at IPv6 address 2001:35af::23
- Secondary server residing at IPv4 address 1.1.1.24
- DNS refresh every day at 11:00pm

WebUI

Network > DNS > Host

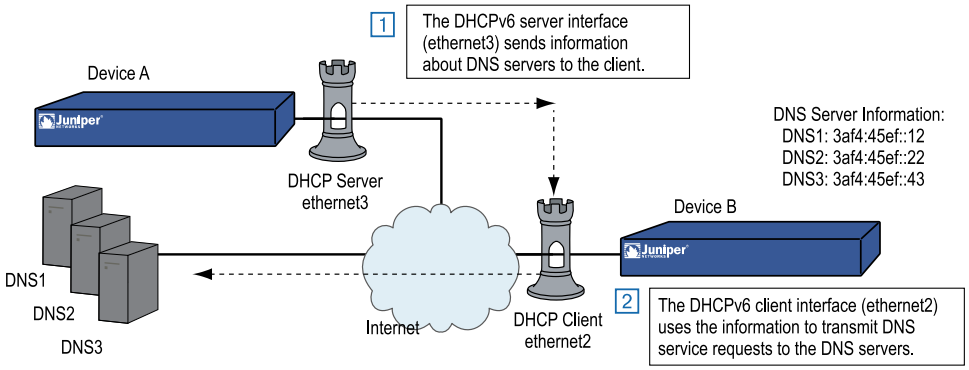
CLI

```
set dns host dns1 2001:35af::23
set dns host dns2 1.1.1.24
set dns host schedule 23:00
save
```

Requesting DNS and DNS Search List Information

Figure 508 on page 2135 shows a DHCPv6 server and a DHCPv6 client. The DHCPv6 client receives information about the DNS devices from interface ethernet3 on the DHCPv6 server. This interface automatically transmits information about the primary, secondary, and tertiary DNS servers. The DHCPv6 client can then send DNS service requests to any of the DNS servers.

Figure 508: DNS Servers and DHCPv6 Client



WebUI (Server)

Network > Interfaces > Edit (for ethernet3) > IPv6

Network > DHCPv6 > Edit (for ethernet3)

WebUI (Client)

Network > Interfaces > Edit (for ethernet3) > IPv6

Network > DHCPv6 > Edit (for ethernet3)

CLI (Server)

```
set interface ethernet3 ipv6 mode router
set interface ethernet3 ipv6 enable
set interface ethernet3 dhcp6 server
set interface ethernet3 dhcp6 server enable
set interface ethernet3 dhcp6 server options dns dns1 3af4:45ef::12
set interface ethernet3 dhcp6 server options dns dns2 3af4:45ef::22
set interface ethernet3 dhcp6 server options dns dns3 3af4:45ef::43
save
```

CLI (Client)

```
set interface ethernet2 ipv6 mode host
set interface ethernet2 ipv6 enable
```

```
set interface ethernet2 dhcp6 client
set interface ethernet2 dhcp6 client enable
set interface ethernet2 dhcp6 client options request dns
set interface ethernet2 dhcp6 client options request search-list
save
```

Setting Proxy DNS Address Splitting

ScreenOS supports proxy DNS address splitting, by which you direct selected domain name queries to specific DNS servers. Address splitting has two advantages:

- Reduces overhead. You can relieve DNS servers from processing irrelevant service requests.
- Protects traffic. You can perform IPsec tunneling on queries sent to DNS servers.

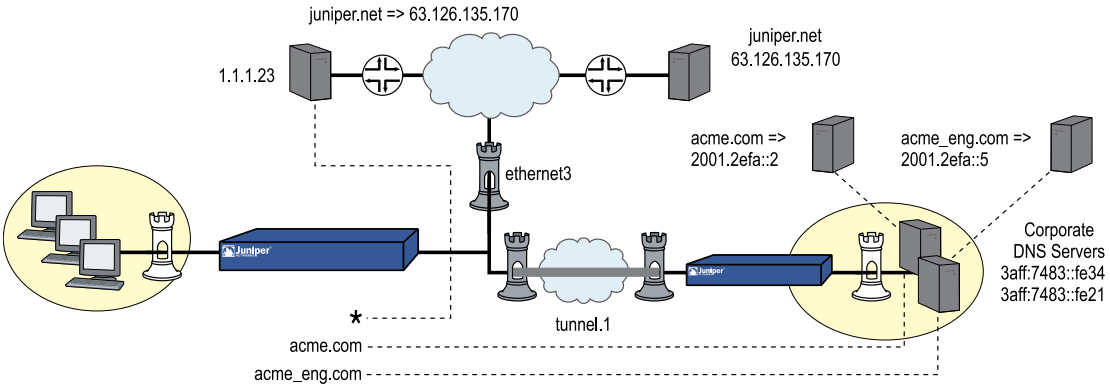
The DNS proxy selectively redirects the DNS queries to specific DNS servers, according to partial or complete domain names. This is useful when VPN tunnels or PPPoE virtual links provide multiple network connectivity, and it is necessary to direct some DNS queries to one network while directing other queries to another network.

Some advantages of a DNS proxy are as follows:

- Domain lookups are usually more efficient. For example, DNS queries meant for the corporate domain (such as marketing.acme.com) could go to the corporate DNS server exclusively, while all others go to the ISP DNS server, which reduces the load on the corporate server. In addition, this can prevent corporate domain information from leaking into the Internet.
- DNS proxy allows you to transmit selected DNS queries through a tunnel interface, which prevents malicious users from learning about internal network configuration. For example, DNS queries bound for the corporate server can pass through a tunnel interface and use security features such as authentication, encryption, and anti-replay.

In the following example, you create two proxy-DNS entries that selectively forward DNS queries to different servers. See Figure 509 on page 2136.

Figure 509: Proxy DNS Using Split Servers



- Any DNS query with an FQDN containing the domain name acme.com goes out through tunnel interface tunnel.1 to the corporate DNS server at IPv6 address 3aff:7483::fe34.

  For example, if a host sends a DNS query to www.acme.com, the device automatically directs the query to this server. (For this example, assume that the server resolves the query to IPv6 address 2001.2efa::2.)

- Any DNS query with an FQDN containing the domain name acme_engineering.com goes out through tunnel interface tunnel.1 to the DNS server at IPv6 address 3aff:7483::fe21.

  For example, if a host sends a DNS query to the intranet.acme_eng.com, the device directs the query to this server. (For this example, assume that the server resolves the query to IPv6 address 2001.2efa::5.)

- All other DNS queries (denoted by an asterisk) bypass the corporate servers and go out through interface ethernet3 to the DNS server at IPv4 address 1.1.1.23.

  For example, if the host and domain name is www.juniper.net, the device automatically bypasses the corporate servers and directs the query to this server, which resolves the query to IPv4 address 63.126.135.170.

WebUI

Network > DNS > Proxy

Network > DNS > Proxy > New

CLI

```
set dns proxy
set dns proxy enable
set dns server-select domain .acme.com outgoing-interface tunnel.1 primary-server
  3aff:7483::fe34
set dns server-select domain .acme_eng.com outgoing-interface tunnel.1 primary-server
  3aff:7483::fe21
set dns server-select domain * outgoing-interface ethernet3 primary-server 1.1.1.23
```

Configuring PPPoE

An IPv6-enabled interface can be a Point-to-Point Protocol over Ethernet (PPPoE) client, which allows members of an IPv6 Ethernet LAN to make individual PPP connections with their ISP. As with IPv4 implementations, the device encapsulates each outgoing IP packet within a PPP payload, then encapsulates the PPP payload inside a PPPoE payload. PPPoE allows devices to operate compatibly on digital subscriber lines (DSL), Ethernet Direct, and cable networks run by ISPs using PPPoE for their clients' Internet access.

You set PPPoE at the interface level and then enable the interface. The interface negotiates with an access concentrator for an Internet Protocol Control Protocol (IPCP) prefix, then an IPv6 Control Protocol (IPv6CP) prefix.

NOTE: PPPoE for IPv6 supports Netscreen Redundancy Protocol (NSRP). For more information about PPPoE in dual-stack IPv6 environments, refer to RFC 4241.

Setting up an interface for PPPoE (dual-stack mode) consists of the following steps:

1. Create a PPPoE instance with username and password.
2. Specify that the PPPoE instance obtains IPv4 and IPv6 prefixes from the PPPoE access concentrator (AC).
3. Configure the device for IPv6 host mode.

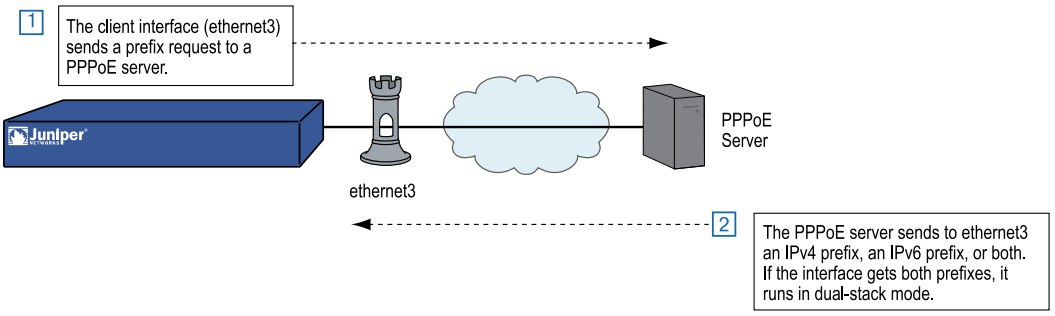
After you enable PPPoE for the interface, the interface negotiates with an AC for an IPCP prefix, then an IPv6CP prefix. Either or both of these attempts could be successful. If both are successful, the interface operates in dual-stack mode. Otherwise, it operates in single-stack mode.

In the following example, you configure interface ethernet3 for PPPoE with the following settings:

- You name the PPPoE instance **NY_Office** and bind it to **ethernet3**.
- You set the PPPoE username as **Richard_B** and the password as **er5cmdj**.
- You configure the PPPoE instance to obtain an IPv4 address first and then an IPv6 address.

Figure 510 on page 2138 shows a PPPoE client and server exchange.

Figure 510: PPPoE Client and Server



NOTE: The WebUI section lists only the navigational paths to the device configuration pages. For specific values, see the CLI section that follows it.

WebUI

1. **PPPoE Instance**

   Network > PPPoE > New

2. **Client Interface**

   Network > Interfaces > Edit (for ethernet3)

   Network > Interfaces > Edit (for ethernet3) > IPv6

CLI

1. **PPPoE Instance**

```
set pppoe name NY_Office username richard_b password er5cmdj
set pppoe name NY_Office ppp ipcp ipv6cp
set pppoe name NY_Office interface ethernet3
```

2. **Client Interface**

```
set interface ethernet3 zone untrust
set interface ethernet3 ipv6 mode host
set interface ethernet3 ipv6 enable
```

Setting Fragmentation

An IPv6 router running ScreenOS uses the Path MTU in combination with the MTU to determine fragmentation.

NOTE: For more information about Path MTU for IPv6, refer to RFC 1981.

An IPv6 interface initially determines the Path MTU to equal the MTU of the first hop in the path. If any packet sent on that path is too large to be forwarded by a router along the path, that router discards the packet and returns an Internet Control Message Protocol version 6 (ICMPv6) Packet Too Big (PTB) message to the source router. Upon receipt of a PTB message, the source router reduces its Path MTU to match the MTU reported in the PTB message. This process might occur several times before the Path MTU discovery process ends because the packet might have to pass through other routers with smaller MTUs further along the path.

The device applies the Path MTU to traffic going through the device and to traffic originating in the device. For through traffic, if any fragmentation is required before forwarding a packet out of an interface at the flow level, the interface sends back an ICMPv6 PTB message with the MTU set to the outgoing interface's MTU.

NOTE: For devices that do not allow you to configure Path MTU, the value is 1500.

For fragmentation to occur, you must enable Path MTU for the interface. For devices that accept a configured value, you can set a value from 1280 to 1500.

WebUI

Network > Interfaces > Edit

In the Maximum Transfer Unit (MTU) Admin MTU: Enter the MTU value, then click **OK**.

CLI

```
set interface trust trust-vr mtu 1280
save
```

NOTE: For advanced configuration examples for manual tunneling with fragmentation enabled, see “Manual Tunneling with Fragmentation Enabled” on page 2216.

Chapter 66
Static and Dynamic Routing

ScreenOS supports static routing and dynamic routing with Routing Information Protocol next-generation (RIPng).

This chapter contains the following sections:

Overview

Dual-stack architecture allows the security device to receive, process, and forward IPv6 traffic and supports assignment of an IPv4 address, at least one IPv6 prefix, or both. This allows the device to exchange packets between dissimilar networks and supports security policies between remote networks over dissimilar WAN backbones.

The next sections explain IPv4/IPv6 routing tables, and static and dynamic routing concepts.

Dual Routing Tables

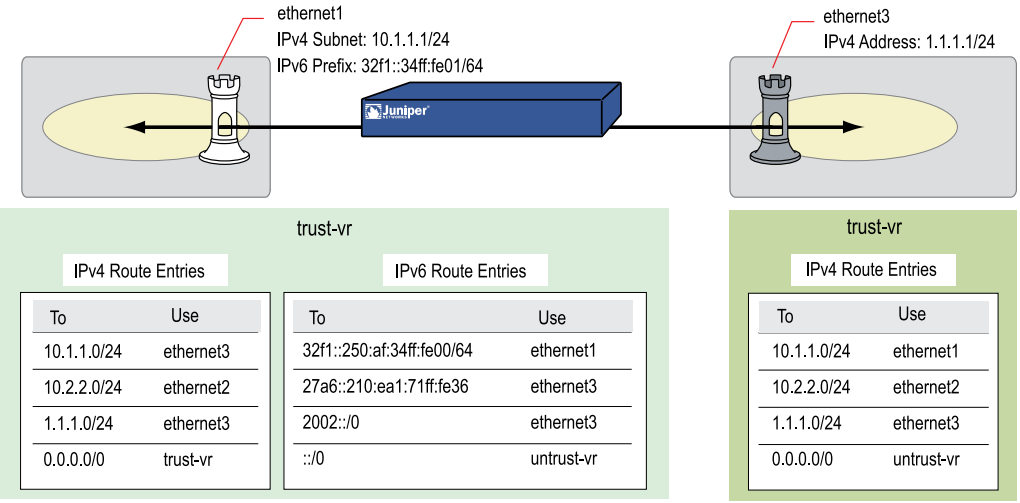
If you configure an interface to have both IPv4 and IPv6 addresses, the virtual router to which the interface is bound contains two separate routing tables: one for IPv4 entries and one for IPv6 entries. Both tables reside in the same virtual router.

For example, if you bind an interface to a zone bound to the trust-vr virtual router, and then you configure the interface for dual-stack mode, the trust-vr virtual router builds and maintains an IPv4 routing table and an IPv6 routing table.

Figure 511 on page 2142 shows a security device with two Ethernet interfaces:

- Ethernet1 runs in dual-stack mode and maintains two routing tables.
- Ethernet3 runs in IPv4 mode only. The trust-vr for ethernet3 maintains only one routing table.

Figure 511: Dual-Stack Router Behavior



Static and Dynamic Routing

ScreenOS supports static routing and dynamic routing with Routing Information Protocol next-generation (RIPng), an Interior Gateway Protocol (IGP) that uses a distance-vector algorithm to determine the best route to a destination, using the hop count as the metric. RIPng supports route redistribution to import known routes, from a router running a different protocol, into the current RIPng routing instance. For example, you can import static routes from a virtual router into a RIPng instance. RIPng is intended only for use in IPv6 networks.

NOTE: For more information about RIPng, refer to RFCs 2080 and 2081.

Upstream and Downstream Prefix Delegation

In some network topologies, devices delegate prefixes to downstream devices, which can use the prefixes to autoconfigure other devices. For example, a corporate WAN might use multiple security devices used in the following ways:

- Customer Premises Equipment (CPE) routers to delegate IP addresses to local hosts
- Gateway routers to delegate subnetwork prefixes to the CPE routers
- Internet Service Provider (ISP) routers further upstream to delegate network prefixes to gateway routers

Static Routing

The process for configuring static routes for IPv6 interfaces is the same as for IPv4 interfaces. The difference is the address notation. For more information about IPv4 static routing, see *"Static Routing" on page 1221*.

In the following example, you configure two interfaces.

- ethernet2 IPv4 address 1.1.2.1/24 (optional)
- ethernet2 configured in router mode with IPv6 address 32f1:250a::02/48
- ethernet2 bound to trust zone and enabled for route advertising to local hosts
- ethernet3 configured in router mode with IPv6 address 32f1:250a::03/48
- Two static routing entries:
  - 32f1:250a::01/48 for ethernet2, advertised to local hosts
  - ::/0 for ethernet3, external gateway 27a6::210:ea1:71ff:fe36

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (ethernet2)

   Network > Interfaces > Edit (ethernet2) Static IP: (select)

   Network > Interfaces > Edit (ethernet2) > IPv6

   Network > Interfaces > Edit (ethernet2) > IPv6 > ND/RA Settings

   Network > Interfaces > Edit (ethernet3)

2. **Routes**

   Network > Routing > Routing Entries New (trust-vr)

   Network > Routing > Routing Entries New (trust-vr) > Gateway: (select)

CLI

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 1.1.2.1/24
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 32f1:250a::02/48
set interface ethernet2 ipv6 ra transmit
set interface ethernet2 ipv6 ra link-mtu
set interface ethernet2 ipv6 ra link-address
```

```
set interface ethernet3 zone untrust
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet3 ipv6 32f1:250a::03/48
```

2. **Routes**

```
set vrouter trust-vr route 32f1:250a::01/48 interface ethernet2
set vrouter trust-vr route ::/0 interface ethernet3 gateway 27a6::210:ea1:71ff:fe36
```

RIPng Configuration

You create RIPng on a per-virtual router basis on a security device. If you have multiple virtual routers (VRs) within a system, you can enable multiple instances of RIPng.

NOTE: Before you configure a dynamic routing protocol on a security device, you should assign a VR ID. For more information, see *"Fundamentals" on page 15*.

This section describes the following basic steps to configure RIPng on a security device:

1. Create the RIPng routing instance in a VR.
2. Enable the RIPng instance.
3. Enable RIPng on interfaces that connect to other RIPng routers.
4. Redistribute routes learned from different routing protocols (such as OSPF, BGP, or statically configured routes) into the RIPng instance.

This section describes how to perform each of these tasks using the CLI or the WebUI.

Optionally, you can configure RIPng parameters such as the following:

- Global parameters, such as timers and trusted RIPng neighbors, that are set at the VR level for RIPng (see "Global RIP Parameters" on page 1316)
- Interface parameters that are set on a per-interface basis for RIPng (see "Configuring RIP Interface Parameters" on page 1318)

Creating and Deleting a RIPng Instance

You create and enable a RIPng routing instance on a specific virtual router (VR) on a security device. When you create and enable a RIPng routing instance on a VR, RIPng transmits and receives packets on all RIPng-enabled interfaces in the VR.

Deleting a RIPng routing instance in a VR removes the corresponding RIPng configurations for all interfaces that are in the VR.

For more information about VRs and configuring a VR on security devices, see *"Routing" on page 1219*.

Creating a RIPng Instance

You create a RIPng routing instance on the trust-vr and then enable RIPng.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit: Enter a Virtual Router ID, then select **Create RIPng Instance**.

Select Enable RIPng, then click **OK**.

CLI

1. **Router ID**

```
set vrouter trust-vr router-id 10
```

2. **RIPng Routing Instance**

```
set vrouter trust-vr protocol ripng
set vrouter trust-vr protocol ripng enable
save
```

Deleting a RIPng Instance

In this example, you disable the RIPng routing instance in the trust-vr. RIPng stops transmitting and processing packets on all RIPng-enabled interfaces of the trust-vr.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIPng Instance: Deselect Enable RIPng and then click **OK**.

Network > Routing > Virtual Router (trust-vr) > Edit > Delete RIPng Instance and then click **OK** at the confirmation prompt.

CLI

```
unset vrouter trust-vr protocol ripng enable
unset vrouter trust-vr protocol ripng
save
```

Enabling and Disabling RIPng on an Interface

By default, RIPng is disabled on all interfaces in the virtual router (VR), and you must explicitly enable it on an interface. When you disable RIPng at the interface level, RIPng does not transmit or receive packets on the specified interface. Interface configuration parameters are preserved when you disable RIPng on an interface.

NOTE: If you disable the RIPng routing instance in the VR (see "Deleting a RIP Instance" on page 1309), RIPng stops transmitting and processing packets on all interfaces in the VR.

Enabling RIPng on an Interface

In this example, you enable RIPng on the Trust interface.

WebUI

Network > Interface > Edit (for Trust) > RIPng: Select Protocol RIPng **Enable**, then click **Apply**.

CLI

```
set interface trust protocol ripng enable
save
```

Disabling RIPng on an Interface

In this example, you disable RIPng on the Trust interface. The **unset interface** *interface_name* **protocol ripng** command unbinds the existing instance of RIPng from the interface. To disable RIP for the interface enter the **unset interface** *interface_name* **protocol ripng enable** command before saving.

WebUI

Network > Interface (for Trust) > RIPng: Clear Protocol RIPng **Enable**, then click **Apply**.

CLI

```
unset interface trust protocol ripng
unset interface trust protocol ripng enable
save
```

Global RIPng Parameters

This section describes RIPng global parameters that you can configure at the virtual router (VR) level. When you configure a RIPng parameter at the VR level, the parameter setting affects operations on all RIPng-enabled interfaces. You can modify global parameter settings through the RIPng routing protocol context in the CLI or by using the WebUI.

Table 142 on page 2147 lists the RIPng global parameters and their default values.

Table 142: Global RIPng Parameters and Default Values

RIPng Global Parameter	Description	Default Value(s)
Advertise default route	Specifies whether the default route (::/0) is advertised.	Disabled
Default metric	Default metric value for routes imported into RIPng from other protocols, such as OSPF and BGP.	10
Flush timer	Specifies, in seconds, when a route is removed from the time the route is invalidated.	120 seconds
Invalid timer	Specifies, in seconds, when a route becomes invalid from the time a neighbor stops advertising the route.	180 seconds
Maximum neighbors	The maximum number of RIPng neighbors allowed. To list the neighbors for a particular interface, enter the **get interface** *interface_name* **protocol ripng** command.	256
Redistribute routes	Provides a way to route traffic coming from another dynamic routing protocol, such as Open Shortest Path First (OSPF), or static routes.	Disabled
Reject default routes	Specifies whether RIPng rejects a default route learned from another protocol. See "Rejecting Default Routes" on page 1321.	Disabled
Incoming route map	Specifies the filter for routes to be learned by RIPng.	None
Outgoing route map	Specifies the filter for routes to be advertised by RIPng.	None
Update threshold	Specifies the number of updates the device processes before the update timer expires. If the threshold is set as 5 and if it receives 10 updates (not number of routes) before the update timer expires, the device drops the last 5 updates. After the update timer is reset another 5 updates are accepted and so on.	Zero (0) seconds
Trusted neighbors	Specifies an access list that defines RIPng neighbors. If no neighbors are specified, RIPng uses multicasting or broadcasting to detect neighbors on an interface. See "Configuring Trusted Neighbors" on page 1320.	All neighbors are trusted
Update timer	Specifies, in seconds, when to issue updates of RIPng routes to neighbors.	30 seconds

Advertising the Default Route

You can change the RIPng configuration to include the advertisement of the default route (a non-RIPng route) and change the metric associated with the default route present in a particular VR routing table.

By default, the default route (::/0) is not advertised to RIPng neighbors. The following command advertises the default route to RIPng neighbors in the trust-vr VR with a metric of 5 (you must enter a metric value). The default route must exist in the routing table.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIPng Instance: Enter the following, then click **OK**:

Advertising Default Route: (select)
Metric: 5

CLI

```
set vrouter trust-vr protocol ripng adv-default-route metric number 5
save
```

Rejecting Default Routes

In a Route Detour Attack, a router inserts a default route (::/0) into the routing domain in order to detour packets to itself. The router can then drop the packets, causing service disruption, or it can obtain sensitive information in the packets before forwarding them. On Juniper Networks security devices, RIPng by default accepts any default routes that are learned in RIPng and adds the default route to the routing table.

In the following example, you configure the RIPng routing instance running in trust-vr to reject any default routes that are learned in RIPng.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIPng Instance: Enter the following, then click **OK**:

Reject Default Route Learnt by RIPng: (select)

CLI

```
set vrouter trust-vr protocol ripng reject-default-route
save
```

Configuring Trusted Neighbors

Multi-access environments can allow devices, including routers, to be connected into a network relatively easily. This can cause stability or performance issues if the connected device is not reliable. To prevent this problem, you can use an access list to filter the devices that are allowed to become RIPng neighbors. By default, RIPng neighbors are limited to devices that are on the same subnet as the virtual router (VR).

In this example, you configure the following global parameters for the RIPng routing instance running in the trust-vr:

- Maximum number of RIPng neighbors is 1.
- The IP address of the trusted neighbor, 2eee::5/64, is specified in an access-list.

WebUI

Network > Routing > Virtual Router (trust-vr) > Access List > New: Enter the following, then click **OK**:

Access List ID: 10
Sequence No.: 1
IP/Netmask: 2eee::5/64
Action: Permit (select)

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIPng Instance: Enter the following, then click **OK**:

Trusted Neighbors: (select), 10
Maximum Neighbors: 1

CLI

```
set vrouter trust-vr
device(trust-vr)-> set access-list 10 permit ip 2eee::5/64 1
device(trust-vr)-> set protocol ripng
device(trust-vr/ripng)-> set max-neighbor-count 1
device(trust-vr/ripng)-> set trusted-neighbors 10
device(trust-vr/ripng)-> exit
device(trust-vr)-> exit
save
```

Redistributing Routes

Route redistribution is the exchange of route information between routing protocols. For example, you can redistribute the following types of routes into the RIPng routing instance in the same virtual router (VR):

- Routes learned from BGP
- Routes learned from OSPF
- Directly connected routes
- Imported routes
- Statically configured routes

You need to configure a route map to filter the routes that are redistributed. For more information about creating route maps for route redistribution, see *"Routing" on page 1219*.

Routes imported into RIPng from other protocols have a default metric of 10. You can change the default metric (see "Global RIP Parameters" on page 1316).

In this example, you redistribute static routes that are in the subnetwork 2eee::/64 to RIPng neighbors in the trust-vr. To do this, you first create an access list to permit addresses in the 2eee::/64 subnetwork. Then, configure a route map that permits addresses that match the access list you configured. Use the route map to specify the redistribution of static routes into the RIPng routing instance.

WebUI

Network > Routing > Virtual Router (trust-vr) > Access List > New: Enter the following, then click **OK**:

Access List ID: 20
Sequence No.: 1
IP/Netmask: 2eee::/64
Action: Permit (select)

Network > Routing > Virtual Router (trust-vr) > Route Map > New: Enter the following, then click **OK**:

Map Name: rtmap1
Sequence No.: 1
Action: Permit (select)
Match Properties:
Access List: (select), 20 (select)

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIPng Instance > Redistributable Rules: Enter the following, then click **Add**:

Route Map: rtmap1 (select)
Protocol: Static (select)

CLI

```
set vrouter trust-vr access-list 20 permit ip 2eee::/64 1
set vrouter trust-vr route-map name rtmap1 permit 1
set vrouter trust-vr route-map rtmap1 1 match ip 20
set vrouter trust-vr protocol ripng redistribute route-map rtmap1 protocol static
save
```

Protecting Against Flooding by Setting an Update Threshold

A malfunctioning or compromised router can flood its neighbors with RIPng routing update packets. On virtual router (VRs), you can configure the maximum number of update packets that can be received on a RIPng interface within an update interval to avoid flooding of update packets. All update packets that exceed the configured update threshold are dropped. If you do not set an update threshold, all update packets are accepted.

In networks where neighbors have large routing tables, specifying an update threshold can impair the security device's ability to learn valid routes. Large routing tables accept and generate a large number of routing updates including flash updates within a given period. Update packets to the security device that exceed the changed threshold will be dropped and important routes might not be learned.

In this example, you set the maximum number of routing update packets that RIPng can receive on an interface to 4.

WebUI

Network > Routing > Virtual Router (trust-vr) > Edit > Edit RIPng Instance: Enter the following, then click **OK**:

Maximum Number Packets per Update Time: (select), 4

CLI

```
set vrouter trust-vr protocol ripng threshold-update 4
save
```

RIPng Interface Parameters

This section describes RIPng parameters that you configure at the interface level. When you configure a RIPng parameter at the interface level, the parameter setting affects the RIPng operation only on the specific interface. You can modify interface parameter settings with **interface** commands in the CLI or by using the WebUI.

Table 143 on page 2151 lists the RIPng interface parameters and their default values.

Table 143: RIPng Interface Parameters and Default Values

RIPng Interface Parameter	Description	Default Value
RIPng metric	Specifies the RIPng metric for the interface.	1
Passive mode	Specifies that the interface is to receive but not transmit RIPng packets.	No
Incoming route map	Specifies the filter for routes to be learned by RIPng.	None.
Outgoing route map	Specifies the filter for routes to be advertised by RIPng.	None.
Split-horizon	Specifies whether to enable split-horizon (do not advertise routes learned from an interface in updates sent to the same interface). If split horizon is enabled with the poison-reverse option, routes that are learned from an interface are advertised with a metric of 16 in updates sent to the same interface.	Split-horizon is enabled. Poison reverse is disabled.

Route, Interface, and Offset Metrics

A RIPng uses a positive integer to indicate the number of hops needed to reach a router. A device uses three values to calculate this number:

- The *route* metric is an integer associated with a particular route prefix.
- The *interface cost* metric is an integer associated with a particular interface configured for RIPng operation.
- The *offset* metric is a value added to the total.

The method by which the device derives the total metric depends on the interface through which the traffic flows.

- If the interface receives incoming traffic, the total metric is:

  route metric + interface cost metric + offset metric

- If the interface transmits outgoing traffic, the interface cost metric does not apply, so the total metric is:

  route metric + offset metric

Access Lists and Route Maps

In this example, you set up an IPv6 access list for the trust-vr with an access control list (ACL) ID of 10 and create a route map with prefix ::/0.

NOTE: To create an IPv4 access list and route map, see *"Configuring a Route Map" on page 1262*.

WebUI

Network > Routing > Virtual Routers > Access Lists > New (trust-vr)

Network > Routing > Virtual Routers > Route Maps (trust-vr) > New

CLI

```
set vrouter trust-vr
set access-list ipv6 10
set access-list 10 permit ip ::/0 10
set route-map ipv6 name rm_six permit 10
set match ip 10
end
```

Static Route Redistribution

In the following example, you configure two devices for RIPng, Device A and Device B. For Device B, you add 2 to the metric value received by a route map (default is 1).

You configure the devices as follows:

- Device A to transmit outgoing traffic to Device B with these metrics:
  - Route metric (5) +
  - Offset metric (1 by default)

    The total RIPng metric for outgoing packets is 6.
- Device B receives traffic from Device A with these metrics:
  - Route metric (4) +
  - Interface cost metric (7) +
  - Offset metric (2)

    The total RIPng metric for incoming packets is 13.

WebUI (Device A)

1. **IPv6 Interface**

   Network > Interfaces > Edit (for ethernet3)

   Network > Interfaces > Edit (for ethernet2) > IPv6

   Network > Interfaces > Edit (for ethernet2) > IPv6 > ND/RA Settings

   Network > Interfaces > Edit (for ethernet2) > RIPng

2. **RIPng Virtual Routing**

   Network > Routing > Virtual Routers > Edit (for trust-vr) > Create RIPng Instance

3. **RIPng Interface Metric**

   Network > Interfaces > Edit (for ethernet2) > IPv6

CLI (Device A)

1. **IPv6 Interface**

```
set interface ethernet3 zone trust
set interface ethernet3 ipv6 mode router
set interface ethernet3 ipv6 enable
set interface ethernet3 ipv6 ip 2eee::1/64
set interface ethernet3 ipv6 interface-id 1111111111111111
set interface ethernet3 ipv6 ra transmit
set interface ethernet3 protocol ripng enable
```

2. **RIPng Virtual Routing**

```
set vrouter trust-vr protocol ripng
set vrouter trust-vr protocol ripng enable
set vrouter trust-vr protocol ripng advertise-def-route always metric 5
```

3. **RIPng Interface Metric**

```
set interface ethernet3 protocol ripng metric 1
```

WebUI (Device B)

1. **IPv6 Interface**

   Network > Interfaces > Edit (for Trust)

   Network > Interfaces > Edit (for Trust) > IPv6

   Network > Interfaces > Edit (for Trust) > IPv6 > ND/RA Settings

2. **Static Routes**

   Network > Routing > Routing Entries

3. **RIPng Virtual Router**

   Network > Routing > Virtual Routers > Edit (for trust-vr) > Create RIPng Instance

   Network > Interfaces > Edit (for Trust) > RIPng

   Network > Routing > Virtual Routers > Access List > New (for trust-vr)

4. **IPv6 Route-Map**

   Network > Routing > Virtual Routers > Route Map > New (for trust-vr)

   Network > Routing > Virtual Router > Edit (for trust-vr) > Create RIPng Instance

5. **Static Route Redistribution**

   Network > Routing > Virtual Router > Edit (for trust-vr) > Edit RIPng Instance > Redistributable Rules

CLI (Device B)

1. **IPv6 Interface**

```
set interface trust zone trust
set interface trust ipv6 mode host
set interface trust ipv6 enable
set interface trust ipv6 interface-id 2222222222222222
set interface trust ipv6 ra accept
```

2. **Static Routes**

```
set route 3a51:94ef::1/48 interface trust metric 2 tag 23
set route 3a51:ee45::1/48 interface trust metric 3 tag 37
```

3. **RIPng Virtual Router**

```
set vrouter trust-vr protocol ripng
set vrouter trust-vr protocol ripng enable
set interface trust protocol ripng enable
set interface trust protocol ripng metric 7
```

```
set vrouter trust-vr access-list ipv6 10
set vrouter trust-vr access-list ipv6 10 permit ip 3a51::1/16 10
```

4. **IPv6 Route-Map**

```
set vrouter trust-vr route-map ipv6 name abc permit 25
set vrouter trust-vr route-map abc 25 match ip 10
set vrouter trust-vr route-map abc 25 metric 4
set vrouter trust-vr route-map abc 25 offset-metric 2
set vrouter trust-vr protocol ripng route-map abc out
set interface trust protocol ripng route-map abc out
```

5. **Static Route Redistribution**

```
set vrouter trust-vr protocol ripng redistribute route-map abc protocol static
save
```

Configuring Split Horizon with Poison Reverse

In the following example, you configure split horizon with poison reverse for the interface.

WebUI

Network > Interfaces > Edit (for Trust) > RIPng: Enter the following, then click **OK**:

Split Horizon: Enabled with poison reverse (select)

CLI

```
set interface trust protocol ripng split-horizon poison-reverse
save
```

Viewing Routing and RIPng Information

After modifying RIPng parameters, you can view the following types of RIPng details:

- Database, which shows routing information
- Protocol, which gives RIPng and interface details for a virtual router (VR)
- Neighbor

Viewing the Routing Table

You can view route information from the CLI.

WebUI

NOTE: You must use the CLI to view the complete routing table.

CLI

get route v6

Sample output:

```
untrust-vr (0 entries)
--------------------------------------------------------------------------------
C - Connected, S - Static, A - Auto-Exported, I - Imported, R - RIP
iB - IBGP, eB - EBGP, O - OSPF, E1 - OSPF external type 1
E2 - OSPF external type 2
trust-vr (6 entries)
--------------------------------------------------------------------------------
   ID          IP-Prefix      Interface         Gateway   P Pref    Mtr     Vsys
--------------------------------------------------------------------------------
*   5               ::/0          trust fe80::210:dbff:fe22:3017   U  251      2     none
*   2         3abc::1/64          trust fe80::210:dbff:fe22:3017   R  100     11     Root
*   3         2abc::1/64          trust fe80::210:dbff:fe22:3017   R  100     11     Root
*   6 2eee::210:dbff:fe20          trust              ::   C    1      0     Root
    1         2eee::1/64          trust fe80::210:dbff:fe22:3017   R  100      2     Root
```

Viewing the RIPng Database

You can verify RIPng routing information from the CLI. You can choose to view a complete list of all RIPng database entries or a single entry.

In this example, you view detailed information from the RIPng database. You can choose to view all database entries or limit the output to a single database entry by appending the IP address and mask of the desired VR.

In this example, you specify the trust-vr and append the prefix and IP address 10.10.10.0/24 to view only a single table entry.

WebUI

NOTE: You must use the CLI to view the RIPng database.

CLI

get vrouter trust-vr protocol ripng database prefix 10.10.10.0/24

After you enter the following CLI command, you can view the RIPng database entry:

device-> **get vrouter trust-vr protocol ripng database 10.10.10.0/24**

The RIPng database contains the following fields:

- **DBID**, the database identifier for the entry
- **Prefix**, the IP address and prefix

- **Nexthop**, the address of the next-hop (router)
- **If**, the type of connection (Ethernet or tunnel)
- Cost metric assigned to indicate the distance form the source

Flags can be one or more of the following: multipath (M), RIPng (R), Redistributed (I), Advertised default (D), Permanent (P), Summary (S), Unreachable (U), or Hold (H).

In this example, the database identifier is 7, the IP address and prefix is 2eee::/64, and the next hop is 2eee::100/64. It is an Ethernet connection with a cost of 2. The flags are M and R and indicate that this route is multipath and uses RIPng.

Viewing RIPng Details by Virtual Router

You can view complete RIPng information for a virtual router to check a configuration or verify that saved changes are active. You can limit output to only the interface summary table by appending interface to the CLI command.

WebUI

NOTE: You must use the CLI to view the RIPng details.

CLI

```
get vrouter trust-vr protocol ripng
```

Sample output:

```
device->  get vrouter trust-vr protocol ripng
get vrouter trust-vr protocol ripng
VR: trust-vr
----------------------------------
State: enabled
Version: 1
Default metric for routes redistributed into RIPng: 10
Maximum neighbors per interface: 256
Next RIPng update scheduled after: 7 sec
Advertising default route: disabled
Default routes learnt by RIPng will be accepted
Incoming routes filter and offset-metric: not configured
Outgoing routes filter and offset-metric: not configured
Update packet threshold is not configured
Total number of RIPng interfaces created on vr(trust-vr): 1

Update Invalid Flush (Timers in seconds)
-----------------------------------------------------------
    30     180   120

Flags: Split Horizon - S, Split Horizon with Poison Reverse - P, Passive - I

Interface   IP-Prefix        Admin      State    Flags   NbrCnt Metric Ver-Rx/Tx
```

```
--------------------------------------------------------------------------------
ethernet3/4 fe80::210:dbff:fe8d enabled    enabled  S            1      1   1/1
```

You can view RIPng settings, packet details, RIPng timer information, and a summarized interface table.

Viewing RIPng Details by Interface

You can view complete RIPng information for a specific interface to check a configuration or verify that saved changes are active.

WebUI

NOTE: You must use the CLI to view the RIPng details.

CLI

```
get interface ethernet3/4 protocol ripng
```

Sample output:

```
VR: trust-vr
---------------------------------
Interface: ethernet3/4, IP: fe80::210:dbff:fe8d:3ea0/64, RIPng: enabled, Router:
 enabled
Receive version 1, Send Version 1
State: Up, Passive: No
Metric: 1, Split Horizon: enabled, Poison Reverse: disabled
Incoming routes filter and offset-metric: not configured
Outgoing routes filter and offset-metric: not configured
Current neighbor count: 1
Next update after: 16 sec
Transmit Updates: 30 (3 triggered), Receive Updates: 68
Update packets dropped because flooding: 0
Bad packets: 0, Bad routes: 0
Neighbors on interface ethernet3/4
--------------------------------------------------------------------------------
IpAddress       Version  Age          Expires      BadPackets BadRoutes
--------------------------------------------------------------------------------
fe80::210:dbf.. 1        00:12:13     00:02:39              0         0
```

Viewing RIPng Neighbor Information

You can view details about RIPng neighbors for a virtual router (VR). You can retrieve a list of information about all neighbors or an entry for a specific neighbor by appending the IP address of the desired neighbor. You can check the status of a route and verify the connection between the neighbor and the security device from these statistics.

In the following example you view RIPng neighbor information for the trust-vr.

WebUI

NOTE: You must use the CLI to view RIPng neighbor information.

CLI

```
get vrouter trust-vr protocol ripng neighbors
```

This command produces output similar to the following output:

```
n2-> get vr trust protocol ripng neighbors
VR: trust-vr
---------------------------------
Neighbors on interface trust
-------------------------------------------------------------------------------
IpAddress       Version  Age          Expires      BadPackets BadRoutes
-------------------------------------------------------------------------------
fe80::210:dbf.. 1        00:06:52     00:02:29              0         0
```

In addition to viewing the IP address, you can view the following RIPng neighbor information:

- Age of the entry
- Expiration time
- Number of bad packets
- Number of bad routes
- Flags: static (S), demand circuit (T), NHTB (N), down (D), up (U), poll (P), or demand circuit init (I)

Configuration Examples

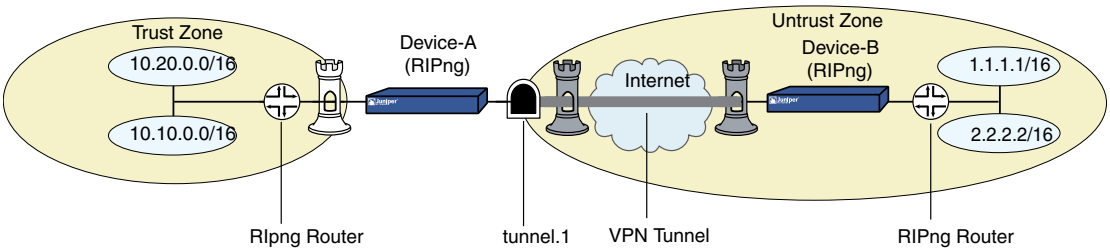
This section contains examples for using RIPng and static routing with tunnels with multiple security devices that behave as IPv6 hosts or IPv6 routers placed upstream or downstream from other devices.

Enabling RIPng on Tunnel Interfaces

The following example creates and enables a RIPng routing instance in trust-vr, on the Device-A device. You enable RIPng on both the VPN tunnel interface and the Trust zone interface. Only routes that are in the subnet 2eee::/64 are advertised to the RIPng neighbor on Device-B. This is done by first configuring an access list that permits only addresses in the subnet 2eee::/64, then specifying a route map abcd that permits routes that match the access list. You then specify the route map to filter the routes that are advertised to RIPng neighbors.

Figure 330 on page 1322 shows the described network scenario.

Figure 512: Tunnel Interface with RIPng Example



WebUI (Device A)

Network > Routing > Virtual Router > Edit (for trust-vr) > Create RIPng Instance: Select **Enable RIPng**, then click **OK**.

Network > Routing > Virtual Router > Access List (for trust-vr) > New: Enter the following, then click **OK**:

Access List ID: 10
Sequence No.: 10
IP/Netmask: 2eee::/64
Action: Permit

Network > Routing > Virtual Router > Route Map (for trust-vr) > New: Enter the following, then click **OK**:

Map Name: abcd
Sequence No.: 10
Action: Permit
Match Properties:
Access List: (select), 10

Network > Routing > Virtual Router > Edit (for trust-vr) > Edit RIPng Instance: Select the following, then click **OK**:

Outgoing Route Map Filter: abcd

Network > Interfaces > Edit (for tunnel.1) > RIPng: Enter the following, then click **Apply**:

Enable RIPng: (select)

Network > Interfaces > Edit (for trust) > RIPng: Enter the following, then click **Apply**:

Enable RIPng: (select)

CLI (Device A)

```
set vrouter trust-vr protocol ripng
set vrouter trust-vr protocol ripng enable
set interface tunnel.1 protocol ripng enable
set interface trust protocol ripng enable
```

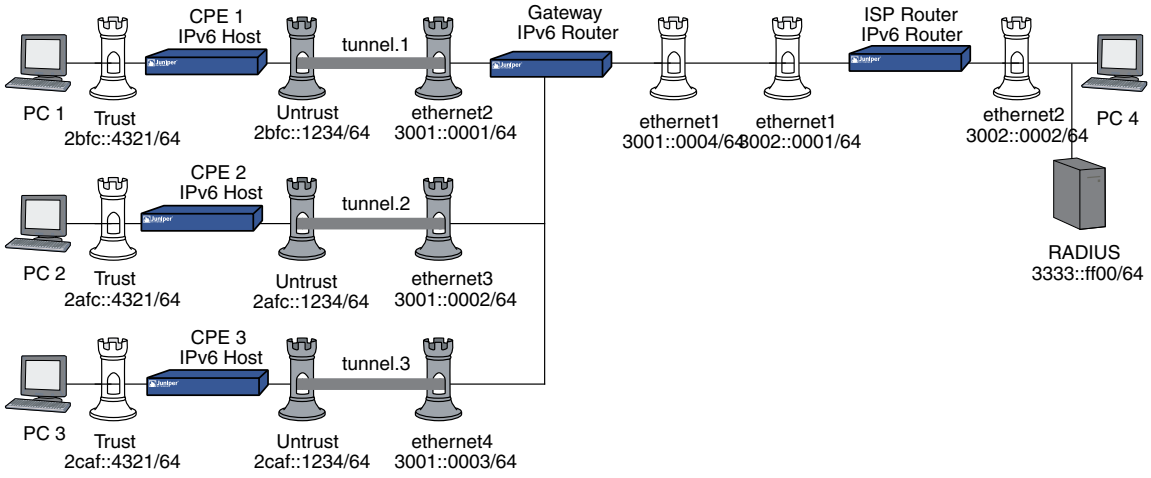
```
set vrouter trust-vr access-list 10 permit ipv6 2eee::/64 10
set vrouter trust-vr route-map ipv6 name abcd permit 10
set vrouter trust-vr route-map ipv6 abcd 10 match ip 10
set vrouter trust-vr protocol ripng route-map abcd out
save
```

Avoiding Traffic Loops to an ISP Router

Figure 513 on page 2161 shows multiple devices configured to avoid traffic loop to the upstream ISP router. The ISP and gateway routers are in IPv6 router mode. The three customer premises equipment (CPE) devices are IPv6 hosts and receive delegated IPv6 prefixes from the ISP router through the gateway router. The RADIUS server resides in the Trust zone of the Gateway device.

In this example, you enable the **route-deny** feature on the interface of a gateway connected to the ISP router. This feature prevents the default route on the gateway from creating a traffic loop to the ISP router.

Figure 513: RADIUSv6 IKE Example



Configuring the Customer Premises Equipment

This section lists the configuration steps for setting up the CPE 1,CPE 2, and CPE 3. The CPE devices are IPv6 hosts. The IPv6 addresses are placeholders for the values that each device could autoconfigure.

1. Configure IPv6 interfaces for all CPE devices. When configuring a CPE device interface in IPv6 host mode, you do not have to explicitly assign the IPv6 prefix. The device accepts the prefix from the gateway router. For each CPE device configuration, we show and recommend assigning each IPv6 interface an interface ID.
2. Configure the following tunnels:
   a. CPE 1 with interface tunnel.1.
   b. CPE 2 with interface tunnel.2.

c. CPE 3 with interface tunnel.3.

3. Define a VPN for each CPE.
4. Define an XAuth client user with username and password for hosts.
5. Configure the Gateway router.

NOTE: In your network, if one or more CPE devices do not specify an IPv6 prefix and use one or more IPv4 addresses, then you need to assign an IPv4 address to the gateway router interface.

6. Configure the ISP router.

NOTE: The WebUI section lists the navigational paths to the device configuration pages for CPE 1, CPE 2, and CPE3. For specific values, see the specific CPE CLI sections that follow it.

WebUI (CPE 1, CPE 2, CPE 3)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.*n*) > IPv6

3. **Route**

   Network > Routing > Routing Table > New (trust-vr)

4. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit

5. **VPN**

   VPNs > AutoKey IKE > New

VPNs > AutoKey IKE > Edit

6. **Policies**

   Policies > (From: Untrust, To: Trust) > New

   Policies > (From: Trust, To: Untrust) > New

CLI (CPE 1)

1. **Interfaces**

```
set interface trust zone trust
set interface trust ipv6 mode host
set interface trust ipv6 enable
set interface trust ipv6 interface-id 3333333333333333
set interface trust ipv6 ra accept

set interface trust manage
set interface trust route

set interface untrust zone trust
set interface untrust ipv6 mode host
set interface untrust ipv6 enable
set interface untrust ipv6 interface-id 4444444444444444
set interface untrust ipv6 ra accept

set interface untrust manage
set interface untrust route
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
```

3. **Static Routes**

```
set vrouter trust-vr route 3002::0001/64 interface tunnel.1
set vrouter trust-vr route ::/0 interface untrust gateway 3001::0001/64
```

4. **IKE**

```
set ike gateway CPE1_G add 3001::0001/64 main outgoing-interface untrust
  preshare abc123 sec-level standard
set ike gateway CPE1_G xauth client any username PC1 password PC1
```

5. **VPN**

```
set vpn CPE1toG gateway CPE1_G sec-level standard
set vpn CPE1toG bind interface tunnel.1
```

6. **Policies**

```
set policy from trust to untrust any any any permit
set policy from untrust to trust any any any permit
```

CLI (CPE 2)

1. **Interfaces**

```
set interface trust zone trust
set interface trust ipv6 mode host
set interface trust ipv6 enable
set interface trust ipv6 interface-id 3333333333333333
set interface trust ipv6 ra accept

set interface trust manage
set interface trust route

set interface untrust zone trust
set interface untrust ipv6 mode host
set interface untrust ipv6 enable
set interface untrust ipv6 interface-id 4444444444444444
set interface untrust ipv6 ra accept

set interface untrust manage
set interface untrust route
```

2. **Tunnel**

```
set interface tunnel.2 zone untrust
```

3. **Static Routes**

```
set vrouter trust-vr route 3002::0001/64 interface tunnel.2
set vrouter trust-vr route ::/0 interface untrust gateway 3001::0001/64
```

4. **IKE**

```
set ike gateway CPE2_G add 3001::0001/64 main outgoing-interface untrust
  preshare abc123 sec-level standard
set ike gateway CPE2_G xauth client any username PC1 password PC1
```

5. **VPN**

```
set vpn CPE2toG gateway CPE2_G sec-level standard
set vpn CPE2toG bind interface tunnel.2
```

6. **Policies**

```
set policy from trust to untrust any any any permit
set policy from untrust to trust any any any permit
```

CLI (CPE 3)

1. **Interfaces**

```
set interface trust zone trust
set interface trust ipv6 mode host
set interface trust ipv6 enable
```

```
set interface trust ipv6 interface-id 5555555555555555
set interface trust ipv6 ra accept

set interface trust manage
set interface trust route

set interface untrust zone trust
set interface untrust ipv6 mode host
set interface untrust ipv6 enable
set interface untrust ipv6 interface-id 6666666666666666
set interface untrust ipv6 ra accept

set interface untrust manage
set interface untrust route
```

2. **Tunnel**

```
set interface tunnel.3 zone untrust
```

3. **Static Routes**

```
set vrouter trust-vr route 3002::0001/64 interface tunnel.3
set vrouter trust-vr route ::/0 interface untrust gateway 3001::0001/64
```

4. **IKE**

```
set ike gateway CPE3_G add 3001::0001/64 main outgoing-interface untrust
  preshare abc123 sec-level standard
set ike gateway CPE3_G xauth client any username PC3 password PC3
```

5. **VPN**

```
set vpn CPE3toG gateway CPE3_G sec-level standard
set vpn CPE3toG bind interface tunnel.3
```

6. **Policies**

```
set policy from trust to untrust any any any permit
set policy from untrust to trust any any any permit
```

Configuring the Gateway

This section lists the configuration steps for setting up the gateway router, an IPv6 router.

1. Configure the ethernet1 interface of gateway with the route-deny feature.
2. Configure the gateway with three tunnels (tunnel.1, tunnel.2, and tunnel.3).
3. Configure the XAuth server with users for the three IKE gateways (from CPE 1, CPE 2, and CPE 3). Specify RADIUS shared secret (juniper) and port number (1812).

4. Define static routes between the RADIUS server and gateway and define a default route from the gateway to the ISP router.
5. Define an IP pool (P1) with an IP range from 2fba::5555 to 2fba::8888 on the gateway (Device 2).

WebUI (Gateway)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.6) > IPv6

3. **Route**

   Network > Routing > Routing Table > New (trust-vr)

4. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit

5. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

6. **Policies**

   Policies > (From: Untrust, To: Trust) > New

   Policies > (From: Trust, To: Untrust) > New

CLI (Gateway)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ipv6 mode router
set interface ethernet1 ipv6 enable
set interface ethernet1 ipv6 interface-id 1111111111111111
set interface ethernet1 ipv6 ip 2eee::1/64
set interface ethernet1 ipv6 ra transmit
```

```
set interface ethernet1 manage
set interface ethernet1 route
set interface ethernet1 route-deny

set interface ethernet2 zone untrust
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 interface-id 1111111111111111
set interface ethernet2 ipv6 ip 2eee::1/64
set interface ethernet2 ipv6 ra transmit
set interface ethernet2 manage
set interface ethernet2 route

set interface ethernet3 zone untrust
set interface ethernet3 ipv6 mode router
set interface ethernet3 ipv6 enable
set interface ethernet3 ipv6 interface-id 1111111111111111
set interface ethernet3 ipv6 ip 2eee::1/64
set interface ethernet3 ipv6 ra transmit
set interface ethernet3 zone untrust
set interface ethernet3 manage
set interface ethernet3 route

set interface ethernet4 zone untrust
set interface ethernet4 ipv6 mode router
set interface ethernet4 ipv6 enable
set interface ethernet4 ipv6 interface-id 1111111111111111
set interface ethernet4 ipv6 ip 2eee::1/64
set interface ethernet4 ipv6 ra transmit
set interface ethernet4 manage
set interface ethernet4 route
```

2. **Tunnels**

```
set interface tunnel.1 zone untrust
set interface tunnel.2 zone untrust
set interface tunnel.3 zone untrust
```

3. **Routes**

```
set vrouter trust-vr route 2fbc::4321/64 interface tunnel.1
set vrouter trust-vr route 2afc::4321/64 interface tunnel.2
set vrouter trust-vr route 2caf::4321/64 interface tunnel.3
set vrouter trust-vr route ::/0 interface ethernet1 gateway 3002::0001
```

4. **RADIUS**

```
set ippool P1 2fba::5555 2fba::8888
set auth-server RAD1 id 1
set auth-server RAD1 server-name 3333::ff00
set auth-server RAD1 account-type xauth
set auth-server RAD1 radius secret juniper
set auth-server RAD1 radius port 1812
set xauth default ippool P1
```

5. **IKE**

```
set ike gateway G_CPE1 add 20.1.1.1 main outgoing-interface ethernet2 preshare
  abc123 sec-level standard
set ike gateway G_CPE1 xauth server RAD1 query-config user PC4
set ike gateway G_CPE1 modecfg server action add-route
set ike gateway G_CPE2 add 60.1.1.1 main outgoing-interface ethernet3 preshare
  abc123 sec-level standard
set ike gateway G_CPE2 xauth server RAD1 query-config user PC4
set ike gateway G_CPE2 modecfg server action add-route
set ike gateway G_CPE3 add 80.1.1.1 main outgoing-interface ethernet4 preshare
  abc123 sec-level standard
set ike gateway G_CPE3 xauth server RAD1 user PC4
set ike gateway G_CPE3 modecfg server action add-route
```

6. **VPN**

```
set vpn GtoCPE1 gateway NS2_NS1 sec-level standard
set vpn G2toCPE1 bind interface tunnel.1
set vpn GtoCPE2 gateway NS2_NS3 sec-level standard
set vpn GtoCPE2 bind interface tunnel.2
set vpn GtoCPE3 gateway NS2_NS4 sec-level standard
set vpn GtoCPE3 bind interface tunnel.3
```

7. **Policies**

```
set policy from trust to untrust any any any permit
set policy from untrust to trust any any any permit
```

Configuring the ISP Router

This section lists the configuration steps for setting up the ISP router, an IPv6 router.

1. Configure the ISP router to reach the gateway router and the RADIUS server.
2. Configure accounts for the RADIUS server.
3. Define a static route on the ISP router to reach the 100.100.100.0/24 network. Make the next hop the ethernet1 interface IP address of the gateway.

WebUI (ISP Router)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Route**

   Network > Routing > Routing Table > New (trust-vr)

CLI (ISP Router)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 ipv6 mode router
set interface ethernet1 ipv6 enable
set interface ethernet1 ipv6 interface-id 1111111111111111
set interface ethernet1 ipv6 ip 3002::0001/64
set interface ethernet1 ipv6 ra transmit
set interface ethernet1 manage
set interface ethernet1 route
set interface ethernet2 zone trust
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 interface-id 2222222222222222
set interface ethernet2 ipv6 ip 3002::0002/64
set interface ethernet2 ipv6 ra transmit
set interface ethernet2 manage
set interface ethernet2 route
```

2. **Route**

```
set vrouter trust-vr route 2fba::5555/64 interface ethernet1 gateway
  3001::0004/64
```

Setting a Null Interface Redistribution to OSPF

In this example, you set a static route to a null interface and redistribute the routes to the ISP router using Open Shortest Path First (OSPF) routing protocol. Any traffic sent to this route is subject to redistribution to OSPF and becomes available to outside OSPF devices.

1. Configure CPE 1, CPE 2, and CPE 3 as described in “Configuring the Customer Premises Equipment” on page 2161.
2. Configure the gateway router as described in “Configuring the Gateway” on page 2165.
3. Configure the ISP router as described in “Configuring the ISP Router” on page 2168.

WebUI (OSPF for Gateway Router)

Network > Routing > Virtual Router > Edit

CLI (Gateway)

```
set vrouter trust-vr route 2fba::5555/64 interface null
set vrouter trust-vr
set protocol ospf
set area 0
```

```
set enable
set interface ethernet1 protocol ospf area 0
set interface ethernet1 protocol ospf enable
set vrouter trust-vr
set route-map name abc permit 10
set match ip 10
set access-list 10 permit ip 2fba::5555/64 10
set vrouter trust-vr protocol ospf redistribute route-map abc protocol static
```

WebUI (ISP)

Network > Routing > Virtual Router > Edit

CLI (ISP)

```
set vrouter trust-vr
set protocol ospf
set area 0
set enable
set interface ethernet1 protocol ospf area 0
set interface ethernet1 protocol ospf enable
```

Redistributing Discovered Routes to OSPF

You can configure the gateway router to redistribute discovered routes to Open Shortest Path First (OSPF) routing protocol. To do this, append the following command information to the configuration that appears on "Avoiding Traffic Loops to an ISP Router" on page 2161.

WebUI (Gateway)

Network > Routing > Virtual Router > Edit

CLI (Gateway)

```
set vrouter trust-vr protocol ospf redistribute route-map abc protocol discovered
```

Setting Up OSPF-Summary Import

In this example, you set up the gateway and ISP routers to redistribute discovered routes to OSPF routing protocol. See "Avoiding Traffic Loops to an ISP Router" on page 2161 for the full configuration and illustration.

1. Configure CPE 1, CPE 2, and CPE 3 as described in "Configuring the Customer Premises Equipment" on page 2161 and the ISP router as described in "Configuring the ISP Router" on page 2168.
2. Configure the gateway as described in "Configuring the Gateway" on page 2165 and then redistribute the static routes to the ISP router. Append the added configuration below.
3. Configure the import summary feature for OSPF.

WebUI (Gateway)

Network > Routing > Routing Table > New (trust-vr)

Network > Routing > Virtual Router > Edit

CLI (Gateway)

1. **OSPF**

```
set vrouter trust-vr
set protocol ospf
set area 0
set enable
set interface ethernet1 protocol ospf area 0
set interface ethernet1 protocol ospf enable
set vrouter trust-vr
set route-map name abc permit 10
set match ip 10
exit
set access-list 10 permit ip 2fba::5555/64 10
set vrouter trust-vr protocol ospf redistribute route-map abc protocol discovered
set vrouter trust protocol ospf summary-import 2fba::5555/64
```

Chapter 67

Address Translation

When two hosts using different IP stacks exchange service requests through a security device, the device must perform address translation for all packets before it can transmit them across an IPv4/IPv6 boundary. This chapter describes the translation process. It contains the following sections:

Overview

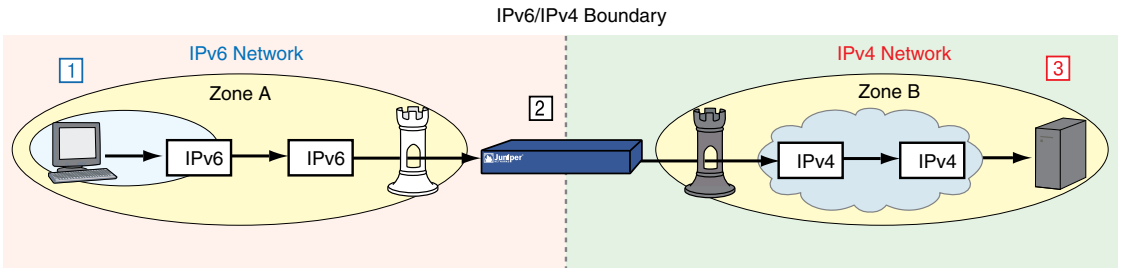
Network Address Translation with Port Translation (NAT-PT) is a mechanism that performs address translation for packets transmitted between hosts that use incompatible IP stacks. For example, a packet generated by an IPv6 host, transmitted over an IPv4-only backbone to an IPv4 host, must have IPv4 source and destination addresses. NAT-PT provides such addresses by translating the original IPv6 source and destination addresses to IPv4.

Devices perform address translations using the following mechanisms.

- Dynamic IP (DIP), which generates new source addresses from a defined address pool.
- Mapped IP (MIP), which translates destination addresses by performing direct mapping between one address and another. MIP can also map addresses to specified subnets.

Figure 514 on page 2173 shows how a device might use NAT-PT to transmit an IPv6 service request packet to an IPv4 host.

Figure 514: Network Address Translation (NAT) Across an IPv4/IPv6 Boundary



1. The IPv6 host transmits an IPv6 service request packet to the local IPv6 interface of a security device.
2. The security device translates the packet addresses to IPv4. It uses DIP to translate the source address and uses MIP to translate the destination address.
3. The destination host receives and processes the IPv4 packet.

NAT-PT converts IPv6 packets into IPv4 packets or IPv4 packets into IPv6 packets, which makes them routable across the IPv6/IPv4 boundary.

When an IPv4 host translates addresses between IPv4 and IPv6, it uses a MIP to generate an *IPv4-mapped* destination address. An IPv4-mapped address is a global unicast IPv6 address that contains an embedded IPv4 address. IPv4-mapped addresses are in the format *IPv6*::*a.b.c.d*, where:

- *IPv6* is the IPv6 subnet portion of the destination address.
- *a.b.c.d* is the embedded IPv4 address (expressed in decimal notation).

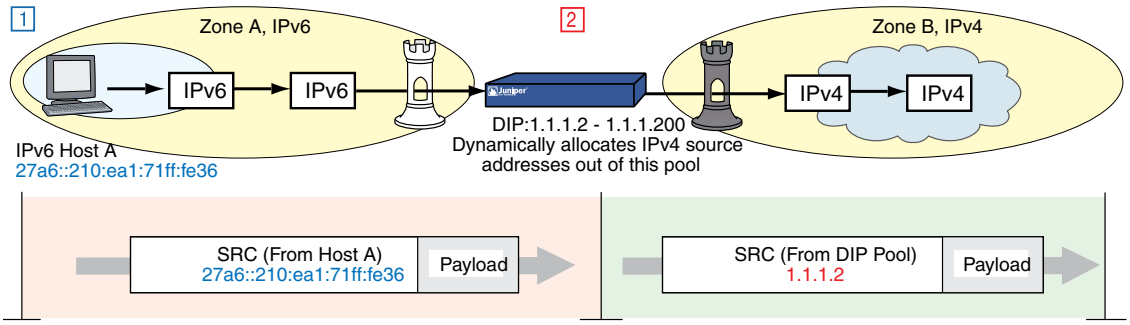
Translating Source IP Addresses

When two hosts that use dissimilar IP stacks exchange packets through a device, the device must translate the source address of each packet to an address compatible with the destination host. For example, an IPv6 packet transmitted into an IPv4 network needs an IPv4 source address; otherwise, the packet cannot successfully traverse the IPv4/IPv6 boundary. To translate source addresses, devices can use dynamic IP (DIP), a mechanism that automatically generates source IP addresses from a defined pool of addresses.

DIP from IPv6 to IPv4

Figure 515 on page 2174 shows an IPv6 host sending a request packet to an IPv4 server over an IPv4 backbone. The security device, an IPv6 router, replaces the IPv6 source address in the packet header with an IPv4 address.

Figure 515: DIP from IPv6 to IPv4

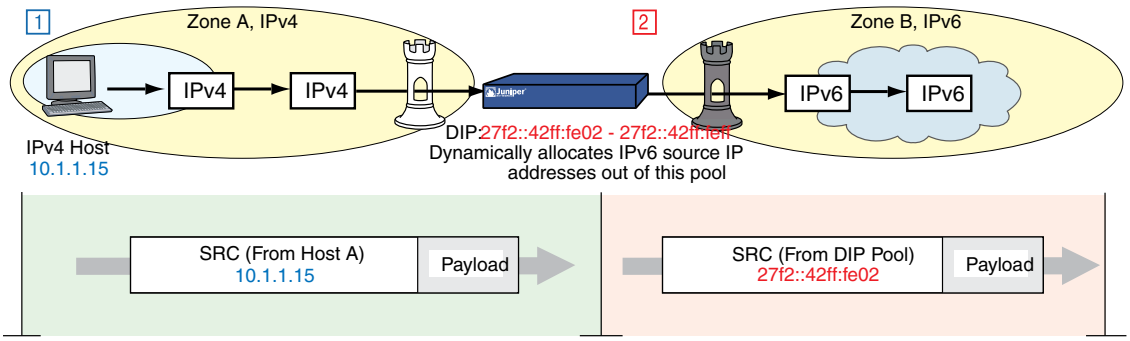


In this example, the device takes IPv4 source addresses 1.1.1.2 from the DIP pool and uses it to replace the original IPv6 source address (27a6::210:ea1:71ff:fe36). It then transmits the packet across the IPv6/IPv4 boundary.

DIP from IPv4 to IPv6

Figure 516 on page 2175 shows an IPv4 host sending a request packet to an IPv6 server in zone B over an IPv6 backbone. The security device replaces the IPv4 source address in the packet header with an IPv6 address.

Figure 516: DIP from IPv4 to IPv6



In this example, the device generates IPv6 source addresses 27f2::42ff:fe02 from the DIP pool and uses it to replace the original IPv4 source address (10.1.1.15). It then sends the packet across the IPv4/IPv6 boundary.

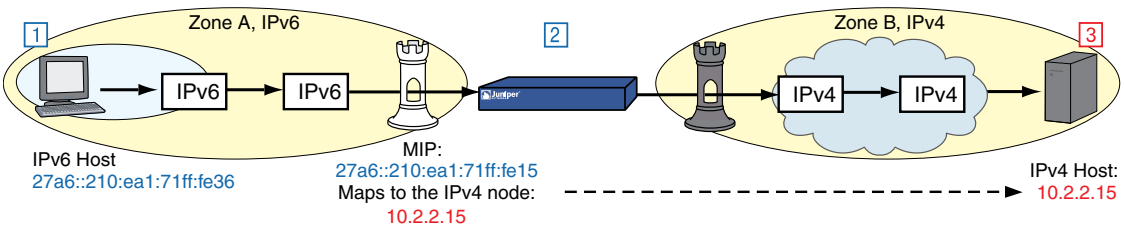
Translating Destination IP Addresses

When two hosts that use dissimilar IP stacks exchange packets through a device, the device must translate the destination address of each packet to an address compatible with the destination host. For example, an IPv6 packet transmitted into an IPv4 network needs an IPv4 destination address; otherwise, the packet cannot cross the IPv4/IPv6 boundary. To perform this translation, devices use mapped IP (MIP).

MIP from IPv6 to IPv4

Figure 517 on page 2175 shows an IPv6 host sending a request packet to an IPv4 server over an IPv4 backbone, the device must use a destination address that matches the IP address format of the destination host.

Figure 517: MIP from IPv6 to IPv4



1. An IPv6 Host generates an IPv6 service request packet and sends it to the security device. The packet enters Zone A.

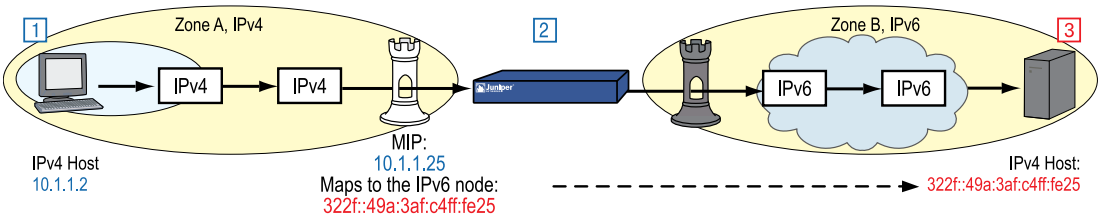
2. The device replaces the destination address with the MIP address and sends the packet to Zone B.
3. The destination device receives the packet somewhere in Zone B. To this device, the packet is an IPv4 packet.

In this example, the IPv6 host addresses the packet to the MIP address (27a6::210:ea1:71ff:fe15). The device translates this MIP to the address of the remote IPv4 host (10.2.2.15).

MIP from IPv4 to IPv6

Figure 518 on page 2176 shows an IPv4 host sending a request packet to an IPv6 server over an IPv6 backbone. The security device uses a destination address that matches the IP address format of the destination host.

Figure 518: MIP from IPv4 to IPv6



1. An IPv4 Host generates an IPv4 service request packet and sends it to the security device. The packet enters Zone A.
2. The security device replaces the destination address with the MIP address and sends the packet to Zone B.
3. The destination device receives the packet somewhere in Zone B. To this device, the packet is an IPv6 packet.

In this example, the IPv4 host addresses the packet to the MIP address (10.1.1.25). The device translates this MIP to the address of the remote IPv6 host (322f::49a:3af:c4ff:fe25).

A MIP can also map to IPv6 networks and subnets.

Configuration Examples

The following sections contain examples of NAT-PT scenarios and translation with domain name services.

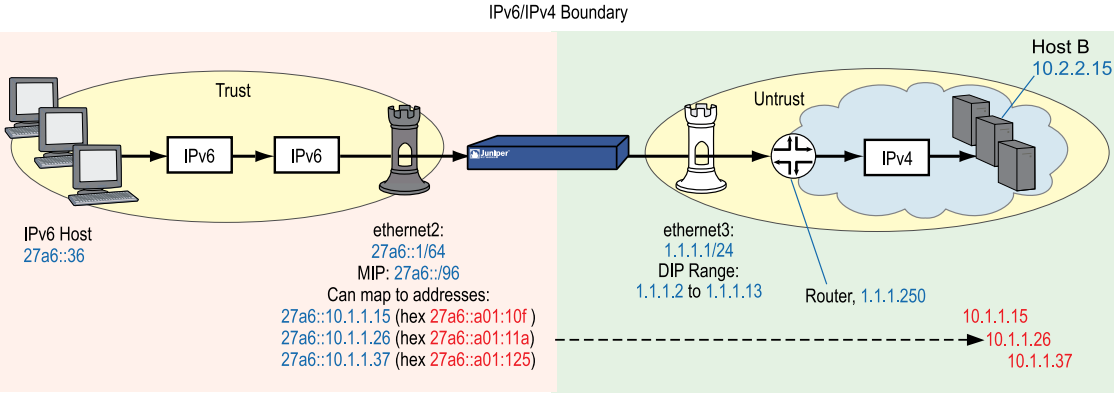
When a device transmits a service request packet from a host and forwards it to another host that uses a different IP stack, the device can use a NAT-PT policy to translate the IPv6 source and destination addresses of the outgoing packet. For example, a device residing at the border between an IPv6 network and an IPv4 WAN might transmit an outgoing IPv6 service request sent by an IPv6 host, using a NAT-PT policy to translate the source and destination addresses to IPv4.

IPv6 Hosts to Multiple IPv4 Hosts

When you need to send service requests from IPv6 hosts to multiple IPv4 destination hosts, define a policy that uses *IPv6-to-IPv4 network mapping* to translate the destination address.

Figure 519 on page 2177 shows the NAT-PT mechanism mapping local IPv6 addresses to IPv4 addresses in a remote IPv4 network. When a local IPv6 host transmits an outgoing service request packet through the device, the device uses MIP to generate an IPv6 destination address that uses IPv4-mapped format. When the device transmits packets across the IPv4/IPv6 boundary, it translates these destination addresses into IPv4 addresses.

Figure 519: IPv4-Mapped Addresses



IPv6 hosts can send HTTP requests through a device across an IPv6/IPv4 boundary. To send such a request, the user enters an IPv4-mapped address (such as 27a6::10.1.1.15) in the URL field of the browser. Such an entry might look like http://[27ab::10.1.1.15]. The device automatically translates this address into its hexadecimal equivalent. To IPv6 hosts the destinations appear to be IPv6-compatible devices. However, before transmitting the HTTP request packet over the boundary, the device translates the address to its IPv4 equivalent, which allows the packet to traverse the IPv4 WAN. To the target IPv4 servers, the packets are completely IPv4-compatible.

In the following example, you configure a device to allow hosts in an island IPv6 network to send HTTP service requests to hosts in a remote IPv4 network. Because the destination address is a network instead of a single host, the device uses *IPv4-mapped addresses* to represent individual remote nodes.

Host A generates an IPv6 service request packet and addresses it to Host B. It specifies an IPv6 destination address in the MIP table. The packet then goes from Zone A to Zone B.

Device A gets a source address from the DIP pool and translates the MIP address to the IPv4 address of the destination host. It then sends the packet out the interface to Device B.

The device translates the IPv4-mapped address to its IPv4 equivalent and then assigns it to the destination address of the outgoing packet.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2)

   Network > Interfaces > Edit (for ethernet2) > IPv6

   Network > Interfaces > Edit (for ethernet3)

2. **MIP**

   Network > Interfaces > Edit (for ethernet2) > IPv6 > MIP > New

3. **DIP**

   Network > Interfaces > Edit (for ethernet3) > DIP > New

4. **Router**

   Network > Routing > Routing Entries New (for trust-vr)

5. **Policy**

   Policies > New (for Trust to Untrust)

   Policies > Edit (for policy) > Advanced

CLI

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ipv6 mode host
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 27a6::1/64
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **MIP**

```
set interface ethernet2 mip 27a6::/96 ipv6 ipv4
```

3. **DIP**

```
set interface ethernet3 dip 7 1.1.1.2 1.1.1.13
```

4. **Router**

```
set vrouter trust-vr route 0.0.0.0/0 gateway 1.1.1.250
```

5. **Policy**

```
set policy from trust to untrust any-ipv6 mip(27a6::/96) http nat src dip-id 7 permit
save
```

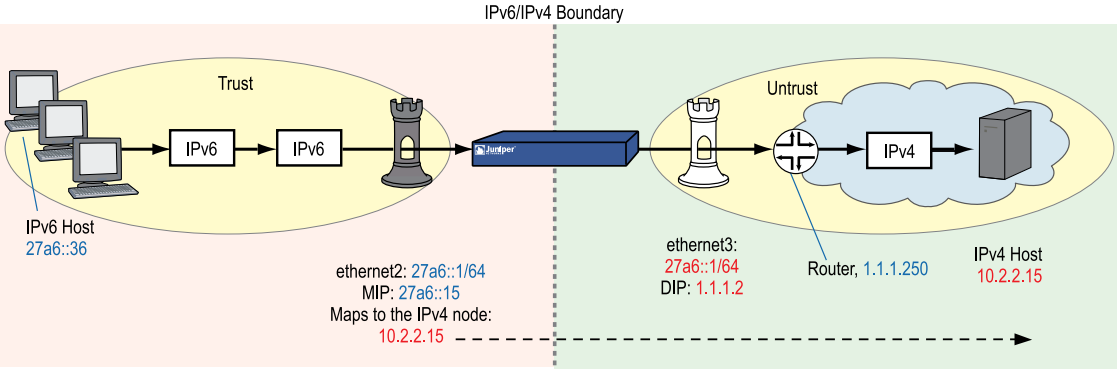
IPv6 Hosts to a Single IPv4 Host

When you need to send service requests from IPv6 hosts to a single IPv4 host, you can define a policy that uses *IPv6-to-IPv4 host mapping* to translate the destination address.

This NAT-PT mechanism maps an IPv6 address to the IPv4 address of a single remote IPv4 host. When a local IPv6 host transmits an outgoing service request packet through the device, the device uses the mapped address for the IPv4 destination address in the packet header. This translation allows the outgoing packet to traverse the IPv6/IPv4 boundary and establish communication with the remote host.

Figure 520 on page 2179 shows IPv6 hosts sending HTTP requests to a single IPv4 server through a device across an IPv6/IPv4 boundary.

Figure 520: IPv6-to-IPv4 Host Mapping



To send a request, the user enters an IPv6 address (such as 27a6::15) in the URL field of a browser. To IPv6 hosts the destination appears to be an IPv6-compatible device. However, before transmitting the HTTP request packet over the IPv6/IPv4 boundary, the device translates the address to its mapped IPv4 address, which allows the packet to traverse the IPv4 WAN. To the target IPv4 server, the packet is completely IPv4-compatible.

In the following example, you configure a device to allow hosts in an island IPv6 network to send HTTP requests to a single remote IPv4 host (10.2.2.15).

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2)

   Network > Interfaces > Edit (for ethernet2) > IPv6

   Network > Interfaces > Edit (for ethernet3)

2. **MIP**

   Network > Interfaces > Edit (for ethernet2) > IPv6 > MIP > New

3. **DIP**

   Network > Interfaces > Edit (for ethernet3) > DIP > New

4. **Router**

   Network > Routing > Routing Entries New (for trust-vr)

5. **Policy**

   Policies > New (for Trust to Untrust)

   Policies > Edit (for policy) > Advanced

CLI

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ipv6 mode host
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 27a6::1/64
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **MIP**

```
set interface ethernet2 mip 27a6::15 ipv6 host 10.2.2.15
```

3. **DIP**

```
set interface ethernet3 dip 7 1.1.1.2 1.1.1.13
```

4. **Routers**

```
set vrouter trust-vr route 0.0.0.0/0 gateway 1.1.1.250
```

5. **Policy**

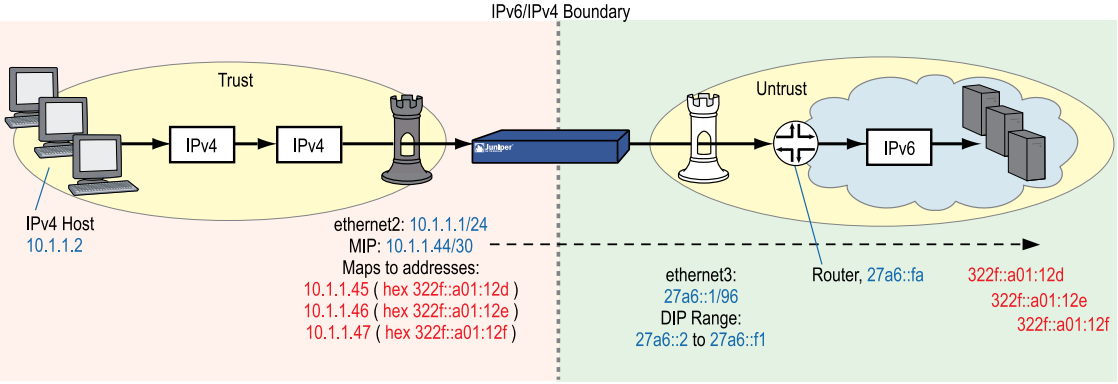
```
set policy from trust to untrust any-ipv6 mip(27a6::15) http nat src dip-id 7 permit
```

IPv4 Hosts to Multiple IPv6 Hosts

When you need to send service requests from local IPv4 hosts to multiple remote IPv6 hosts, you can define a policy that uses *IPv4-to-IPv6 network mapping*.

Figure 521 on page 2181 shows NAT-PT mapping a local IPv4 address to an IPv6 network (or subnet). When a local IPv4 host transmits an outgoing IPv4 service request packet through the device, the device translates the mapped address to the IPv6 address and assigns it to the destination address of the packet header. This translation allows the outgoing packet to traverse the IPv4/IPv6 boundary to establish communication with the remote IPv6 host.

Figure 521: IPv4-to-IPv6 Network Mapping



The mapped IPv4 address indirectly represents the remote IPv6 host. In effect, the local IPv4 hosts can view the remote host as part of the local IPv4 network.

Because the MIP belongs to the same subnet as the local IPv4 hosts, the hosts can view the remote host as part of the local IPv4 network.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2)

   Network > Interfaces > Edit (for ethernet3)

   Network > Interfaces > Edit (for ethernet3) > IPv6

2. **MIP**

   Network > Interfaces > Edit (for ethernet2) > IPv6 > MIP > New

3. **DIP**

   Network > Interfaces > Edit (for ethernet3) > DIP > New

4. **Router**

Network > Routing > Routing Entries New (for trust-vr)

5. **Policy**

   Policies > New (for Trust to Untrust)

   Policies > Edit (for policy) > Advanced

CLI

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ipv6 mode host
set interface ethernet3 ipv6 enable
set interface ethernet3 ipv6 ip 27a6::1/64
```

2. **MIP**

```
set interface ethernet2 mip 10.1.1.44 ipv6 prefix 322f::44/96 netmask
  255.255.255.252
```

3. **DIP**

```
set interface ethernet3 dip 7 27a6::2 27a6::f1
```

4. **Routers**

```
set vrouter trust-vr route ::/0 interface ethernet3 gateway 27a6::fa
```

5. **Policy**

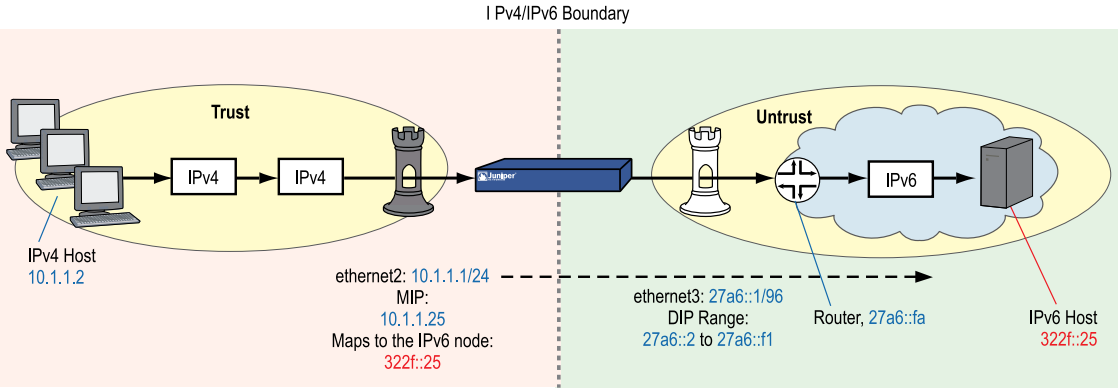
```
set policy from trust to untrust any-ipv4 mip(10.1.1.44/30) http nat src dip-id 7
  permit
```

IPv4 Hosts to a Single IPv6 Host

When you need to send service requests from local IPv4 hosts to a single remote IPv6 host, define a policy that uses *IPv4-to-IPv6 host mapping*.

Figure 522 on page 2183 shows an IPv4 host transmitting an outgoing IPv4 service request packet through the device. The device translates the mapped address to an IPv6 address, and assigns it to the destination address of the packet header. This translation allows the outgoing packet to traverse the IPv4/IPv6 boundary to establish communication with the remote IPv6 host.

Figure 522: IPv4-to-IPv6 Host Mapping



The MIP belongs to the same subnet as the local IPv4 hosts, so hosts view the remote host as part of the local IPv4 network.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2)

   Network > Interfaces > Edit (for ethernet3)

   Network > Interfaces > Edit (for ethernet3) > IPv6

2. **MIP**

   Network > Interfaces > Edit (for ethernet2) > IPv6 > MIP > New

3. **DIP**

   Network > Interfaces > Edit (for ethernet3) > DIP > New

4. **Router**

   Network > Routing > Routing Entries New (for trust-vr)

5. **Policy**

   Policies > New (for Trust to Untrust)

   Policies > Edit (for policy) > Advanced

CLI

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ipv6 mode host
```

```
set interface ethernet3 ipv6 enable
set interface ethernet3 ipv6 ip 27a6::1/96
```

2. **MIP**

```
set interface ethernet2 mip 10.1.1.25 ipv6 host 322f::25
```

3. **DIP**

```
set interface ethernet3 dip 7 27a6::2 27a6::f1
```

4. **Routers**

```
set vrouter trust-vr route ::/0 interface ethernet3 gateway 27a6::fa
```

5. **Policy**

```
set policy from trust to untrust any-ipv4 mip(10.1.1.25) http nat src dip-id 7 permit
```

Translating Addresses for Domain Name System Servers

Domain Name System (DNS) allows network devices to identify each other using domain names instead of IP addresses. An external DNS server keeps a table of domain names, each name having at least one associated IP address. You can use these domain names (as well as IP addresses) for identifying endpoints in policy definitions.

Some domain names can have both IPv4 and IPv6 addresses. For example, the hypothetical domain names acme.com and juniper.net could have the following addresses:

Domain Name	IPv4 Addresses	IPv6 Addresses
www.acme.com	1.2.2.15	27a6::e02:20f
	1.2.2.62	27a6::23ee:51a
	2.3.5.89	27a6::5542:225
www.juniper.net	3.1.1.21	3090::e02:20f
	3.20.7.96	3090::23ee:51a
	4.6.5.89	2e66::3354:722
	4.200.7.88	2e66::3354:722

When you define a policy that uses domain names as endpoints, and both domain names have IPv4 and IPv6 addresses, the device can establish secure communication between the endpoints using either protocol. For example, the following commands define a policy between the domain names juniper.net and acme.com:

```
set address trust juniper www.juniper.net
set address untrust acme www.acme.com
set policy from trust to untrust juniper acme any permit
```

The two endpoints can exchange either type of traffic because the domain names each have IPv4 and IPv6 addresses.

However, if one domain name uses an IP stack that the other domain name does not, you must use NAT-PT to translate the source and destination addresses for transmitted DNS requests. For example, if juniper.net has only IPv6 addresses and acme.com has only IPv4 addresses, any service request sent from one to the other must undergo NAT-PT address translation.

In addition to translating the addresses, the NAT-PT DNS ALG also modifies the DNS request before forwarding it to another domain that has only IPv4 addresses. You can enable or disable the NAT-PT DNS ALG to modify DNS requests received from the IPv6 domain. If enabled, the NAT-PT DNS ALG modifies the DNS request for AAAA records going into the IPv4 domain. By default, this option is disabled, and the DNS request from the IPv6 domain undergoes only NAT-PT address translation. But if the DNS reply message indicates an error, the NAT-PT DNS ALG modifies the DNS request for AAAA records and sends it again to the IPv4 domain.

To enable the NAT-PT DNS ALG to modify DNS requests:

WebUI

Security > ALG > DNS: Enter the following, then click **Apply**:

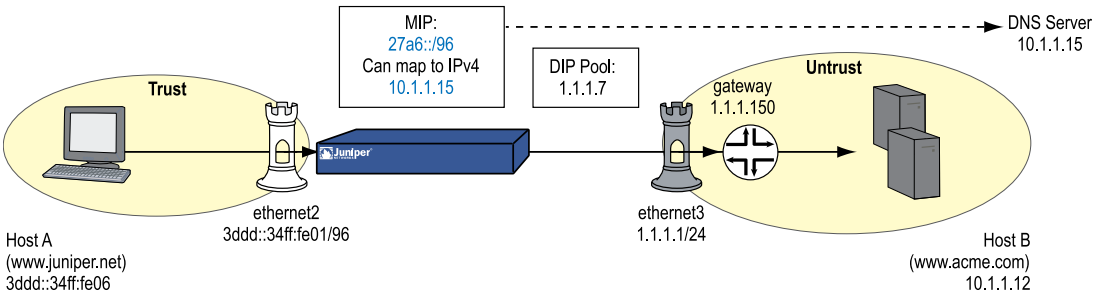
```
DNS Enable: (select)
DNS Inhibit-AAAA-Request Enable: (select)
```

CLI

```
set alg dns inhibit-aaaa-request
save
```

In the following example, an IPv6 host (www.juniper.net) sends service requests to an IPv4 host (www.acme.com). It obtains the IPv4 address of the destination host from an IPv4 DNS server.

Figure 523: NAT-PT DNS Example



The device requires two policies:

- A policy that permits outgoing service requests and that performs NAT-PT on the source and destination addresses
- A policy that permits outgoing service requests from www.juniper.net to www.acme.com

In the following example, you configure a device to allow a Host A (www.juniper.net) to send service requests to a Host B (www.acme.com). The host domain names serve as the communication endpoints.

WebUI

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2)

   Network > Interfaces > Edit (for ethernet2) > IPv6

   Network > Interfaces > Edit (for ethernet3)

2. **MIP**

   Network > Interfaces > Edit (for ethernet2) > IPv6 > MIP > New

3. **DIP**

   Network > Interfaces > Edit (for ethernet3) > DIP > New

4. **Routers**

   Network > Routing > Routing Entries New (for trust-vr)

5. **Addresses**

   Policy > Policy Elements > Addresses > List > New (for Trust)

   Policy > Policy Elements > Addresses > List > New (for Untrust)

6. **Policy**

   Policies > New (for Trust to Untrust)

   Policies > Edit (for policy) > Advanced > Advanced

CLI

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ipv6 mode host
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 3ddd::34ff:fe01/96
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **MIP**

   ```
   set interface ethernet2 mip 3ddd::/96 ipv6 ipv4 vrouter trust-vr
   ```

3. **DIP**

   ```
   set interface ethernet3 dip 7 1.1.1.7
   ```

4. **Addresses**

   ```
   set address trust juniper.net 3ddd::34ff:fe06/96
   set address untrust acme www.acme.com
   ```

5. **Routers**

   ```
   set vrouter trust-vr route 0.0.0.0/0 gateway 1.1.1.250
   ```

6. **Policy**

   ```
   set policy from trust to untrust any-ipv6 mip(3ddd::/96) any nat dip-id 7 permit
   set policy from trust to untrust any-ipv4 acme any permit
   save
   ```

Chapter 68

IPv6 in an IPv4 Environment

6over4 tunneling is the process of encapsulating IPv6 packets within IPv4 packet headers so that IPv6 packets can traverse an IPv4 wide area network (WAN).

This chapter contains the following sections:

Overview

6over4 tunneling is a way to send IPv6 traffic over an IPv4 wide area network (WAN) when you don't need authentication and encryption for the exchanged traffic. For example, your organization might need to exchange traffic between island IPv6 networks over an IPv4 WAN. In this case, the security device provides only firewall services.

There are two kinds of 6over4 tunneling:

- *Manual tunneling* uses a manually configured, static remote-end tunnel termination point.
- *6to4 tunneling* derives the remote-end tunnel termination point dynamically.

NOTE: For more information about 6to4 tunneling, refer to RFC 3056.

In most cases, the kind of tunneling to use depends on the kinds of routers and other devices present in a WAN infrastructure. For example, some ISPs (Internet Service Providers) provide manual tunnels to their customers as addendum services, and might have no need for 6to4 tunneling. In addition, there might be insufficient 6to4-configured relays in the IPv6 backbone to support 6to4 tunneling for your organization over the WAN.

6to4 tunneling can be appropriate if your organization needs to set up a one-to-many configuration, where one IPv6 network can access unspecified, multiple island IPv6 networks through multiple 6to4 relay routers. In such cases, 6to4 tunneling might be the best solution.

Configuring Manual Tunneling

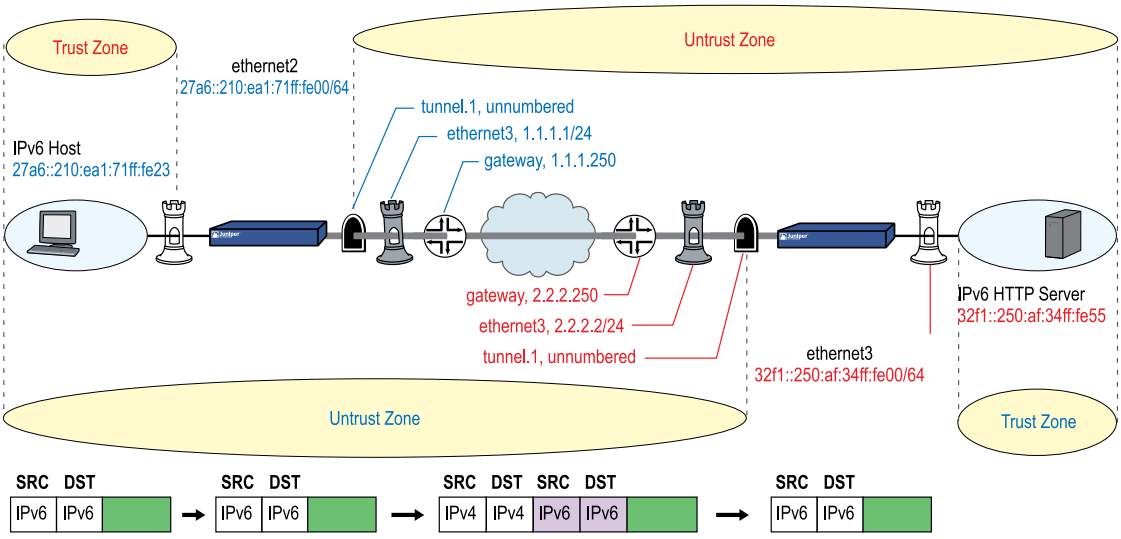
Manual tunneling is best suited for applications that require strict control and usually provide more security than 6to4 tunneling. Manual tunneling does not require address translation because you specify explicitly the IPv4 address of the destination gateway.

IPv6 hosts use the tunnel to communicate with an IPv6 server over an IPv4 WAN backbone. The endpoints of the tunnel are interfaces on devices A and B.

When you configure a device for manual tunneling, the device encapsulates each outgoing IPv6 packet inside an IPv4 packet before transmitting it into IPv4 network space.

Figure 524 on page 2190 shows a manual tunnel between two static, defined endpoints.

Figure 524: IPv6 Tunneling Using IPv4 Encapsulation Example



In the following example, you set up two devices (A and B) as endpoints for a manual 6to4 tunnel.

WebUI (Device A)

Interfaces

Network > Interfaces > Edit (for ethernet2)

Network > Interfaces > Edit (for ethernet2) > IPv6

Network > Interfaces > Edit (for ethernet3)

Tunnel Interfaces

Network > Interfaces > New (Tunnel IF)

Network > Interfaces > Edit (for tunnel.1) > IPv6

Network > Interfaces > Edit (for tunnel.1) > Encapsulation

Routers

Network > Routing > Routing Table > New (trust-vr)

Network > Routing > Route Table > New (trust-vr):

Address Book Entries

Policy > Policy Elements > Addresses > List (Trust) > New

Policy

Policies > (From: Untrust, To: Trust) > New

WebUI (Device B)

Interfaces

Network > Interfaces > Edit (for ethernet2)

Network > Interfaces > Edit (for ethernet2) > IPv6

Network > Interfaces > Edit (for ethernet3)

Tunnel Interfaces

Network > Interfaces > New (Tunnel IF)

Network > Interfaces > Edit (for tunnel.1) > IPv6

Network > Interfaces > Edit (for tunnel.1) > Encapsulation

Routers

Network > Routing > Routing Table > New (trust-vr)

Network > Routing > Route Table > New (trust-vr)

Address Book Entries

Policy > Policy Elements > Addresses > List (Trust) > New

Policy

Policies > (From: Untrust, To: Trust) > New

CLI (Device A)

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 10.1.1.1/24
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 2abc::1/64
set interface ethernet2 ipv6 ra link-mtu
set interface ethernet2 ipv6 ra link-address
set interface ethernet2 ipv6 ra transmit
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Tunnel Interface**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.1 ipv6 mode host
set interface tunnel.1 ipv6 enable
set interface tunnel.1 tunnel encap ip6in4 manual
set interface tunnel.1 tunnel local-if ethernet3 dst-ip 2.2.2.2
```

3. **Routers**

```
set vrouter trust-vr route 3abc::/64 interface tunnel.1
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 1.1.1.250
```

4. **Addresses**

```
set address trust L_Hosts 2abc::/64
set address untrust R_Server 3abc::100/128
```

5. **Policy**

```
set policy from trust to untrust L_Hosts R_Server http permit
```

CLI (Device B)

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 1.100.2.1/24
set interface ethernet2 ipv6 mode host
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 3abc::1/64
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.2/24
```

2. **Tunnel Interface**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.1 ipv6 mode host
set interface tunnel.1 ipv6 enable
set interface tunnel.1 tunnel encap ip6in4 manual
set interface tunnel.1 tunnel local-if ethernet3 dst-ip 1.1.1.1
```

3. **Routers**

```
set vrouter trust-vr route 2abc::/64 interface tunnel.1
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3 gateway 2.2.2.250
```

4. **Addresses**

```
set address trust L_Server 3abc::100/128
set address untrust R_Clients 2abc::/64
```

5. **Policy**

```
set policy from untrust to trust R_Clients L_Server http permit
```

Configuring 6to4 Tunneling

When a device uses *6to4 tunneling* , the device determines remote-end tunnel termination points (gateways) dynamically from routing table entries. In contrast with manual tunneling, which explicitly designates a single termination point for a VPN tunnel, 6to4 tunneling can allow any number of devices to serve as remote gateways for the tunnel. This allows one-to-many communication between a protected island IPv6 network and multiple external IPv6 networks.

For a network device to function as a 6to4 host, it must have an interface configured with a 6to4 address. Any IPv6 host without such an address is said to be *native* , or non-6to4. For example, if a host has interfaces with global aggregate IPv6 addresses but none with a 6to4 address, it is a native host. By contrast, if the host has an interface configured with a 6to4 address (whether or not it has other kinds of IPv6 addresses), it can function as a 6to4 host.

To set up a 6to4 tunnel between two devices, you must configure each communicating interface with a 6to4 address. The interfaces then serve as virtual border routers that handle the transition between IPv4 space and IPv6 space.

There are two kinds of 6to4 virtual router.

- **6to4** routers function as border routers between IPv4 WANs and 6to4 networks.
- **6to4 relay** routers function as border routers between IPv4 WANs and native networks.

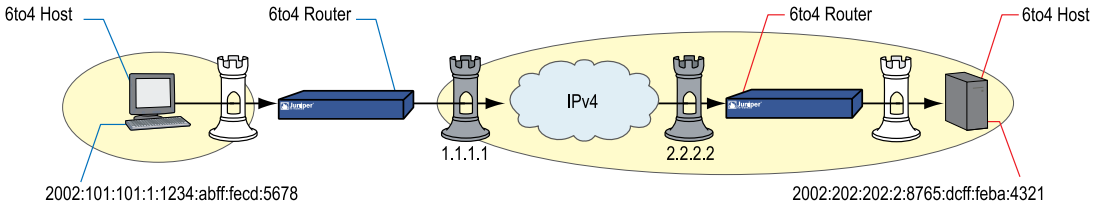
6to4 Routers

A *6to4 router* is a router configured to support exchange of packets between 6to4 hosts and other 6to4 hosts over an IPv4 WAN. You can make a device function as a 6to4 router by configuring an IPv4 interface for 6to4.

When a packet from a 6to4 host passes through the device, the 6to4 router encapsulates the packet inside an IPv4 packet, then transmits it into the IPv4 network space. 6to4 routers can also receive such encapsulated packets, decapsulate them, and forward them to 6to4 devices.

Figure 259 on page 967 shows a 6to4 host sending a service request across an IPv4 WAN to another 6to4 host. The 6to4 router on Device A encapsulates the outgoing packets, and the 6to4 router on Device B decapsulates them.

Figure 525: 6to4 Routers

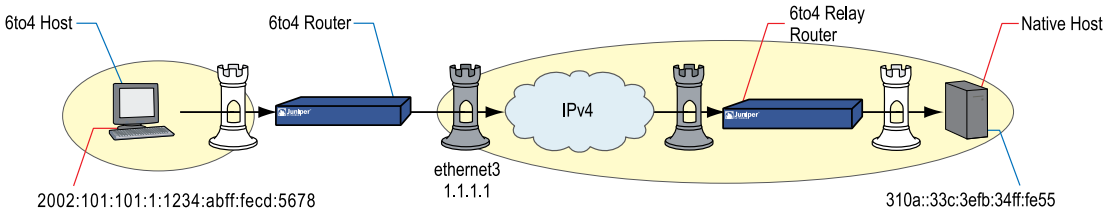


6to4 Relay Routers

A *6to4 relay router* is a router configured to support exchange of packets between 6to4 hosts and native (non-6to4) hosts over an IPv4 WAN. As with 6to4 routers, you can make a device function as a 6to4 router by configuring an IPv4 interface to handle 6to4 tunneling. When a 6to4 relay router receives an incoming encapsulated 6to4 packet, the router decapsulates the packet and forwards it to the native destination host. When a native host transmits an outgoing packet, the 6to4 relay router encapsulates the packet inside an IPv4 packet, then transmits it into the IPv4 network space.

In Figure 526 on page 2194, a 6to4 host sends a service request across an IPv4 WAN to a native (non-6to4) host. Device B decapsulates incoming packets, using the native IPv6 address as the destination address.

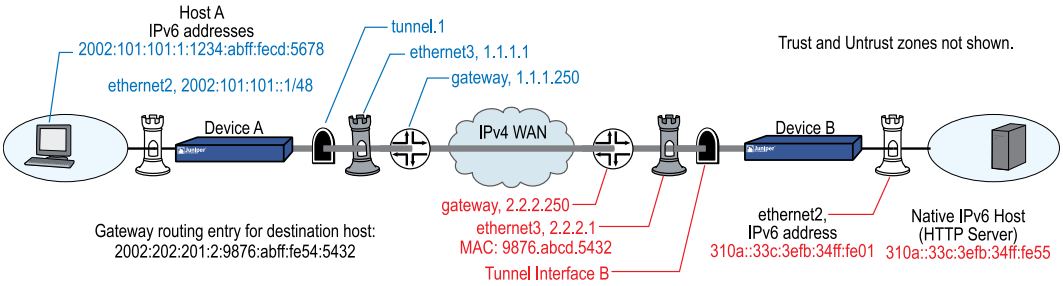
Figure 526: 6to4 Routers with Native Addresses



Tunnels to Remote Native Hosts

When a 6to4 host sends a service request to a remote native host, the device derives the remote termination point (gateway IPv4 address) from a routing table entry. Figure 527 on page 2195 shows Device A sending an HTTP service request to the native server protected by Device B.

Figure 527: 6over6 Manual Tunneling



The 6to4 router on Device A derives the remote gateway IPv4 address (2.2.2.1) from a configured routing table gateway entry 2002:0202:201:2:9876:abff:fecd:5432. (For this example, the MAC address of the remote gateway interface is 9876abcd5432.)

A device addresses packets that pass between a 6to4 host and a native host over an IPv4 WAN in the following manner:

1. The IPv6 host (Host A) generates an IPv6 packet and sends it to Device A.
2. Device A, a 6to4 router, encapsulates the outgoing IPv6 packet inside an IPv4 packet and sends it out the tunnel interface over the IPv4 WAN to Device B.
3. Device B decapsulates the packet and forwards it to Host B.
4. Host B generates a reply packet. Device B encapsulates it and forwards it to Device A.
5. Device A receives the encapsulated reply packet through the tunnel interface. Device A decapsulates the packet and forwards it to Host A.

In this example, you configure a device to send HTTP requests from IPv6 hosts to a native host (an HTTP server) over an IPv4 WAN infrastructure.

WebUI (Device A)

Interfaces

Network > Interfaces > Edit (for ethernet2)

Network > Interfaces > Edit (for ethernet2) > IPv6

Network > Interfaces > Edit (for ethernet3)

Tunnel Interfaces

Network > Interfaces > New (Tunnel IF)

Network > Interfaces > Edit (for tunnel.1) > IPv6

Network > Interfaces > Edit (for tunnel.1) > Encapsulation

Routers

Network > Routing > Routing Table > New (trust-vr)

Address Book Entries

Policy > Policy Elements > Addresses > List (Trust) > New

Policy

Policies > (From: Untrust, To: Trust) > New

WebUI (Device B)

Interfaces

Network > Interfaces > Edit (for ethernet2)

Network > Interfaces > Edit (for ethernet2) > IPv6

Network > Interfaces > Edit (for ethernet3)

Tunnel Interfaces

Network > Interfaces > New (Tunnel IF)

Network > Interfaces > Edit (for tunnel.1) > IPv6

Network > Interfaces > Edit (for tunnel.1) > Encapsulation

Routers

Network > Routing > Routing Table > New (trust-vr)

Network > Routing > Routing Table > trust-vr New: Enter the following, then click **OK**:

Network > Routing > Routing Table > trust-vr Edit > Gateway: (select)

Address Book Entries

Policy > Policy Elements > Addresses > List (Trust) > New

Policy > Policy Elements > Addresses > List (Untrust) > New

Policy

Policies > (From: Untrust, To: Trust) > New: Enter the following, then click **OK**:

Policies > (From: Untrust, To: Trust) > Edit > Advanced:

CLI (Device A)

1. **Interfaces**

```
set interface ethernet2 zone trust
```

```
set interface ethernet2 ip 10.1.1.1/24
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 2002:101:101::1/48
set interface ethernet2 ipv6 ra link-mtu
set interface ethernet2 ipv6 ra link-address
set interface ethernet2 ipv6 ra transmit
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Tunnel Interface**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.1 ipv6 mode router
set interface tunnel.1 ipv6 enable
set interface tunnel.1 tunnel encap ip6in4 6to4
set interface tunnel.1 tunnel local-if ethernet3
```

3. **Routers**

```
set vrouter trust-vr route 0.0.0.0/0 gateway 1.1.1.250
set vrouter trust-vr route 3abc::/16 interface tunnel.1 gateway 2002:0202:201::1
```

4. **Address Book Entries**

```
set address trust L_Hosts 2002:101:101::/48
set address untrust R_Server 3abc::100/128
```

5. **Policy**

```
set policy from trust to untrust L_Hosts R_Server http permit
```

CLI (Device B)

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 10.2.2.1/24
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 310a::33c:3efb:34ff:fe01/64
set interface ethernet2 ipv6 ra link-mtu
set interface ethernet2 ipv6 ra link-address
set interface ethernet2 ipv6 ra transmit
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.1/24
```

2. **Tunnel Interface**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.1 ipv6 mode router
set interface tunnel.1 ipv6 enable
set interface tunnel.1 tunnel encap ip6in4 6to4
```

```
set interface tunnel.1 tunnel local-if ethernet3
```

3. **Routers**

```
set vrouter trust-vr route 0.0.0.0/0 gateway 2.2.2.250
set vrouter trust-vr route 2002::/16 interface tunnel.1
```

4. **Address Book Entries**

```
set address trust L_Server 310a::33c:3efb:34ff:fe55/128
set address untrust R_Clients 2002:101:101::/48
```

5. **Policy**

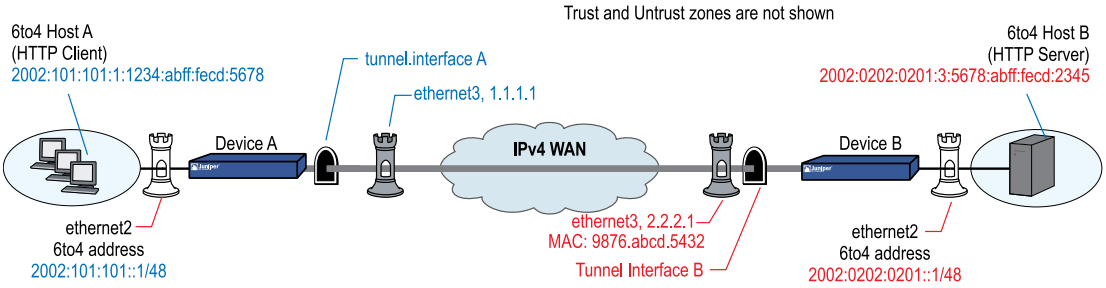
```
set policy from untrust to trust R_Clients L_Server http permit
```

Tunnels to Remote 6to4 Hosts

When a 6to4 host sends a service request to remote a 6to4 host, the device derives the remote termination point (gateway IPv4 address) from the destination IP address of the outgoing request packet.

Figure 528 on page 2198 shows Host A sending an HTTP service request to a server protected by Device B.

Figure 528: 6to4 Tunnel



In this example, the 6to4 destination address of the outgoing packet is 2002:0202:0201:3:5678:abff:fecd:2345, which is the address of 6to4 Host B. The 6to4 router (on Device A) derives the remote gateway IPv4 address (2.2.2.1) from this destination address.

A device addresses packets that pass between a 6to4 host and another 6to4 host over an IPv4 WAN in the following manner:

1. The IPv6 Host A generates an IPv6 packet and sends it to Device A.
2. Device A derives the IPv4 destination address of the remote gateway (2.2.2.1) from the destination address of the outgoing packet.
3. Device A encapsulates the outgoing 6to4 packet inside an IPv4 packet and sends it out the tunnel interface into Zone B.
4. Device B decapsulates the packet and forwards it to Host B.

5. Host B generates a reply packet.
6. Device B encapsulates it and forwards it to Device A.

In this example, you configure a device to send HTTP requests from IPv6 hosts to an IPv6 server over an IPv4 WAN infrastructure.

WebUI (Device A)

Interfaces

Network > Interfaces > Edit (for ethernet2)

Network > Interfaces > Edit (for ethernet2) > IPv6

Network > Interfaces > Edit (for ethernet3)

Tunnel Interfaces

Network > Interfaces > New (Tunnel IF)

Network > Interfaces > Edit (for tunnel.1) > IPv6

Network > Interfaces > Edit (for tunnel.1) > Encapsulation

Routers

Network > Routing > Routing Table > New (trust-vr)

Network > Routing > Routing Table > New (untrust-vr)

Address Book Entries

Policy > Policy Elements > Addresses > List (Trust) > New

Policy

Policies > (From: Untrust, To: Trust) > New

WebUI (Device B)

Interfaces

Network > Interfaces > Edit (for ethernet2)

Network > Interfaces > Edit (for ethernet2) > IPv6

Network > Interfaces > Edit (for ethernet3)

Tunnel Interfaces

Network > Interfaces > New (Tunnel IF)

Network > Interfaces > Edit (for tunnel.1) > IPv6

Network > Interfaces > Edit (for tunnel.1) > Encapsulation

Routers

Network > Routing > Routing Table > untrust-vr New

Address Book Entries

Policy > Policy Elements > Addresses > List (Trust) > New

Policy > Policy Elements > Addresses > List (Untrust) > New

Policy

Policies > (From: Untrust, To: Trust) > New

CLI (Device A)

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 10.1.1.1/24
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 2002:101:101::1/48
set interface ethernet2 ipv6 ra link-mtu
set interface ethernet2 ipv6 ra link-address
set interface ethernet2 ipv6 ra transmit
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Tunnel Interface**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.1 ipv6 mode router
set interface tunnel.1 ipv6 enable
set interface tunnel.1 tunnel encap ip6in4 6to4
set interface tunnel.1 tunnel local-if ethernet3
```

3. **Routers**

```
set vrouter trust-vr route 2002:0202:0201::/48 interface tunnel.1
```

4. **Address Book Entries**

```
set address trust L_Hosts 2002:101:101::/48
set address untrust R_Server 2002:202:201:0:5678:abff:fecd:2345/128
```

5. **Policy**

```
set policy from trust to untrust L_Hosts R_Server any permit
```

CLI (Device B)

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 10.2.2.1/24
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 2002:202:0201::1/48
set interface ethernet2 ipv6 ra link-mtu
set interface ethernet2 ipv6 ra link-address
set interface ethernet2 ipv6 ra transmit
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.1/24
```

2. **Tunnel Interface**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.1 ipv6 mode router
set interface tunnel.1 ipv6 enable
set interface tunnel.1 tunnel encap ip6in4 6to4
set interface tunnel.1 tunnel local-if ethernet3
```

3. **Routers**

```
set vrouter trust-vr route 2002::/16 interface tunnel.1
```

4. **Address Book Entries**

```
set address untrust R_Clients 2002:101:101::/48
set address trust L_Server 2002:202:0201:3:5678:abff:fecd:2345/128
```

5. **Policy**

```
set policy from untrust to trust R_Clients L_Server http permit
```

Chapter 69
IPsec Tunneling

6in6, 4in6, and 6in4 tunneling allow you to create virtual private networks. These mechanisms automatically perform packet encapsulation and ensure secure packet exchange over the WAN.

This chapter contains the following sections:

Overview

ScreenOS allows you to create virtual private networks (VPNs). A VPN connection can link two local area networks (LANs) or a remote dialup user with a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. To secure VPN communication traffic, the two participants create an IP security (IPsec) tunnel.

You create policies to provide IPsec security services such as authentication and encryption or use Network Address Translation (NAT) to define private address space. For more information about NAT, see *"Address Translation" on page 1467*. For information about security concepts and VPNs, see *"Fundamentals" on page 15* and *"Virtual Private Networks" on page 705*.

This chapter includes configuration examples that use VPN tunnels in a purely IPv6 environment and other scenarios for IPv4 or IPv6 island networks or hosts.

IPsec 6in6 Tunneling

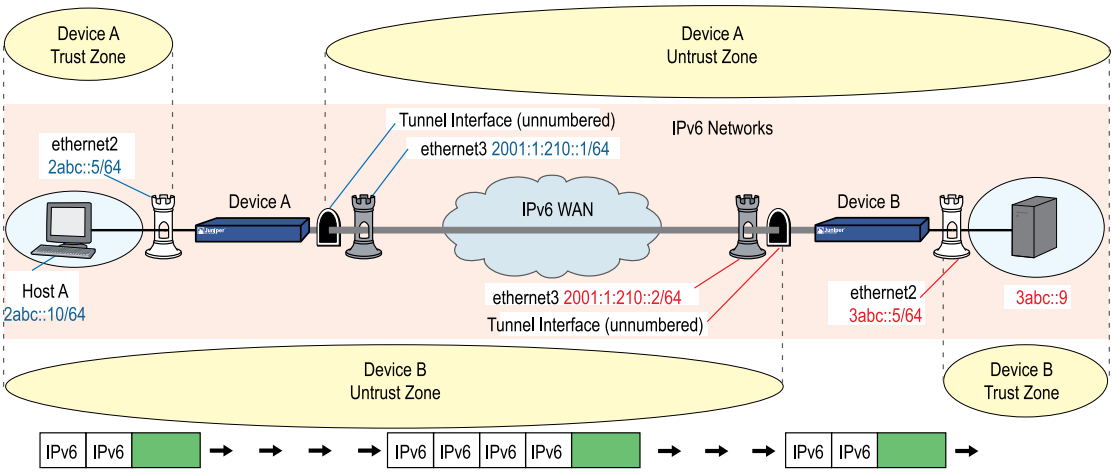
6in6 tunneling is a method to encapsulates IPv6 packets inside IPv6 packets and is used where the exchanged information must travel within an IPv6 domain and requires protection to ensure data confidentiality and integrity.

Use IPsec in combination with 6in6 tunneling when you need to establish secure communication between IPv6 hosts and other IPv6 hosts over an IPv6 infrastructure,

For example, you might need to exchange secured traffic between an IPv6 network and an IPv6 server over an IPv6 WAN.

Figure 529 on page 2204 shows an IPv6 host sending an HTTP service request to an IPv6 Web server over an IPv6 WAN infrastructure. Both devices have policies that use IPsec security.

Figure 529: IPsec with 6in6 Tunnel Example



The following lists the steps to how a device configured for IPv4-to-IPv6 host mapping translates the source and destination addresses of an outgoing service request packet.

1. Host A generates the service request packet and assigns it the IPv6 destination address (3abc::9).
2. Device A encapsulates the entire IPv6 packet inside of another IPv6 packet. It then sends it out the tunnel interface to Device B.
3. The remote IPv6 host device sends a reply packet.
4. Device A decapsulates the replay packet and sends it to the IPv6 host.

In the following example, you set up IPsec policies that allow HTTP communication between an IPv6 network and a remote IPv6 network or a subnet. The traffic passes over an IPv6 WAN environment.

WebUI (Device A)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2)

   Network > Interfaces > Edit (for ethernet2) > IPv6

   Network > Interfaces > Edit (for ethernet3)

   Network > Interfaces > Edit (for ethernet3) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (For tunnel.1) > IPv6

3. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit (for gw-test) > Advanced

4. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit (For vpn-test) > Advanced

5. **Route**

   Network > Routing > Routing Entries > New

6. **Policies**

   Policies > New (Trust to Untrust)

   Policies > New (Untrust to Trust)

CLI (Device A)

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 2abc::5/64
set interface ethernet3 ipv6 mode host
set interface ethernet3 ipv6 enable
set interface ethernet3 ipv6 ip 2001:1:210::1/64
```

2. **Tunnel Interface**

```
set interface tunnel.1 zone untrust
```

```
set interface tunnel.1 ipv6 mode host
set interface tunnel.1 ipv6 enable
```

3. **IKE**

```
set ike gateway gw-test address 2001:1:210::2 main outgoing-interface ethernet3
  local-address 2001:1:210::1 preshare abcd1234 proposal pre-g2-des-md5
```

4. **VPN**

```
set vpn vpn-test gateway gw-test proposal g2-esp-des-md5
set vpn vpn-test bind interface tunnel.1
set vpn vpn-test proxy-id local-ip ::/0 remote-ip ::/0 any
```

5. **Route**

```
set route 3abc::/16 interface tunnel.1
```

6. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set policy from untrust to trust any-ipv6 any-ipv6 any permit
```

WebUI (Device B)

1. **Interfaces**

   Network > Interfaces > Edit (for ethernet2)

   Network > Interfaces > Edit (for ethernet2) > IPv6

   Network > Interfaces > Edit (for ethernet3)

   Network > Interfaces > Edit (for ethernet3) > IPv6

2. **Tunnel Interface**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (For tunnel.1) > IPv6

3. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit (for gw-test) > Advanced

4. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit (For vpn-test) > Advanced

5. **Route**

Network > Routing > Routing Entries > New

6. **Policies**

   Policies > New (Trust to Untrust)

   Policies > New (Untrust to Trust)

CLI (Device B)

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 3abc::5/64
set interface ethernet3 ipv6 mode host
set interface ethernet3 ipv6 enable
set interface ethernet3 ipv6 ip 2001:1:210::2/64
```

2. **Tunnel Interface**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ipv6 mode host
set interface tunnel.1 ipv6 enable
```

3. **IKE**

```
set ike gateway gw-test address 2001:1:210::1 main outgoing-interface ethernet3
  local-address 2001:1:210::2 preshare abcd1234 proposal pre-g2-des-md5
```

4. **VPN**

```
set vpn vpn-test gateway gw-test proposal g2-esp-des-md5
set vpn vpn-test bind interface tunnel.1
set vpn vpn-test proxy-id local-ip ::/0 remote-ip ::/0 ANY
```

5. **Route**

```
set route 2abc::/16 interface tunnel.1
```

6. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set policy from untrust to trust any-ipv6 any-ipv6 any permit
```

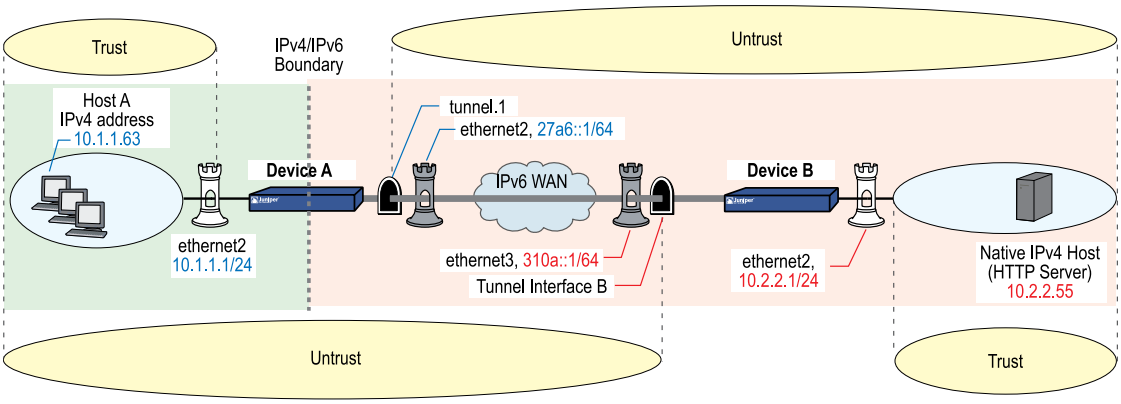
IPsec 4in6 Tunneling

4in6 tunneling allows you to establish communication between IPv4 networks over an IPv6 backbone. This transition mechanism uses IPsec to encapsulate packets exchanged between IPv4 networks over an IPv6 backbone. Encapsulation makes exchanged packets routable across the IPv6 network space.

In addition to encapsulating the packets, IPsec can perform authentication and encryption.

Figure 530 on page 2208 shows an IPsec tunnel between remote IPv4 networks. In this example, you define a tunnel interface on each gateway device and bind the tunnel to a zone that borders IPv6 network space.

Figure 530: IPsec 4in6 Tunnel Example



A device addresses packets that pass between an IPv4 host in an IPv4 island network and another IPv4 host over an IPv6 WAN backbone in the following manner:

1. Host A generates an IPv6 service request packet and addresses it to Host B. The packet goes from Zone A to Zone B.
2. Device A encapsulates the outgoing IPv4 packet inside an IPv6 packet and sends it out the tunnel interface.
3. Device B decapsulates the packet and forwards it to Host B.
4. Device B generates a reply packet, encapsulates it, and forwards it to Device A.
5. Device A receives the encapsulated reply packet through the tunnel interface. Device A decapsulates the packet and forwards it to Host A.

The steps to set up 4in6 tunneling are as follows:

1. Configure an interface for communication with the protected IPv4 network.
   - Bind the interface to a zone (typically the Trust zone).
   - Assign the interface an IPv4 address and subnet mask.
2. Configure an interface for communication over the IPv6 WAN.
   - Bind the interface to a zone (typically the Untrust zone).
   - Configure the interface for IPv6 host mode.
   - Assign the interface an IPv6 address and subnet mask.

- Create a tunnel interface (unnumbered) in the zone and bind it to the IPv6 interface.
- Configure the tunnel interface for IPv6 host mode.

3. Set up interfaces on the peer device in a similar manner.
4. Set up IPsec between the peer devices. For more examples of IPsec, see *"Virtual Private Networks" on page 705*.
5. On each device, create address book entries that identify the IPv6 host, subnet, or network.
6. Set up routing entries that allow the hosts to access each other.
7. Set up security policies.

In the following example, you create an IPsec tunnel between IPv6 endpoints. IPv4 devices behind Device A transmit service requests hosts behind Device B. Device A use IPsec to encapsulate the service request packets inside IPv6 packets. Device B receives and decapsulates the packets.

WebUI (Device A)

Interfaces

Network > Interfaces > Edit (for ethernet2)

Network > Interfaces > Edit (for ethernet3)

Network > Interfaces > Edit (for ethernet3) > IPv6

Tunnel

Network > Interfaces > New (Tunnel IF)

Network > Interfaces > Edit (for tunnel.1) > IPv6

IKE

VPNs > Autokey Advanced > Gateway > New

VPN

VPNs > Autokey IKE > New

Routers

Network > Routing > Routing Table > New (trust-vr)

Network > Routing > Routing Table > New (untrust-vr)

Policy

Policies > (From: Trust, To: Untrust) > New

CLI (Device A)

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 10.1.1.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
set interface ethernet3 ipv6 mode host
set interface ethernet3 ipv6 enable
set interface ethernet3 ipv6 ip 27a6::1/64
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.1 ipv6 mode host
set interface tunnel.1 ipv6 enable
```

3. **IKE**

```
set ike gateway IPSec_Servers ip 310a::1 main outgoing-interface ethernet3
  local-address 27a6::1 proposal rsa-g2-aes128-sha
```

4. **VPN**

```
set vpn Tunnel_Servers gateway IPSec_Servers no-replay tunnel sec-level standard
set vpn Tunnel_Servers bind interface tunnel.1
```

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3
set vrouter trust-vr route 310a::0/64 interface tunnel.1 gateway 310a::1
```

6. **Policy**

```
set policy from trust to untrust any any any permit
```

WebUI (Device B)

Interfaces

Network > Interfaces > Edit (for ethernet2)

Network > Interfaces > Edit (for ethernet3)

Network > Interfaces > Edit (for ethernet3) > IPv6

Tunnel

Network > Interfaces > New (Tunnel IF)

Network > Interfaces > Edit (for tunnel.1) > IPv6

IKE

VPNs > Autokey Advanced > Gateway > New

VPN

VPNs > Autokey IKE > New

Routers

Network > Routing > Routing Table > New (trust-vr)

Addresses

Policy > Policy Elements > Addresses > List (Trust) > New

Policy > Policy Elements > Addresses > List (Untrust) > New

Policy

Policies > (From: Trust, To: Untrust) > New

CLI (Device B)

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 10.2.2.1/24
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.1/24
set interface ethernet3 ipv6 mode host
set interface ethernet3 ipv6 enable
set interface ethernet3 ipv6 ip 310a::1/96
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.1 ipv6 mode host
set interface tunnel.1 ipv6 enable
```

3. **IKE**

```
set ike gateway IPSec_Clients ip 27a6::1 main outgoing-interface ethernet3
  local-address 310a::1 proposal rsa-g2-aes128-sha
```

4. **VPN**

```
set vpn Tunnel_Clients id 1 gateway IPSec_Clients no-replay tunnel sec-level
  standard
set vpn Tunnel_Clients id 2 bind interface tunnel.1
```

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 vrouter untrust-vr
```

```
set vrouter untrust-vr route 27a6::0/64 interface tunnel.1 gateway 27a6::1
```

6. **Addresses**

```
set address trust L_Server 10.2.2.55/32
set address untrust R_Clients 10.1.1.1/24
```

7. **Policy**

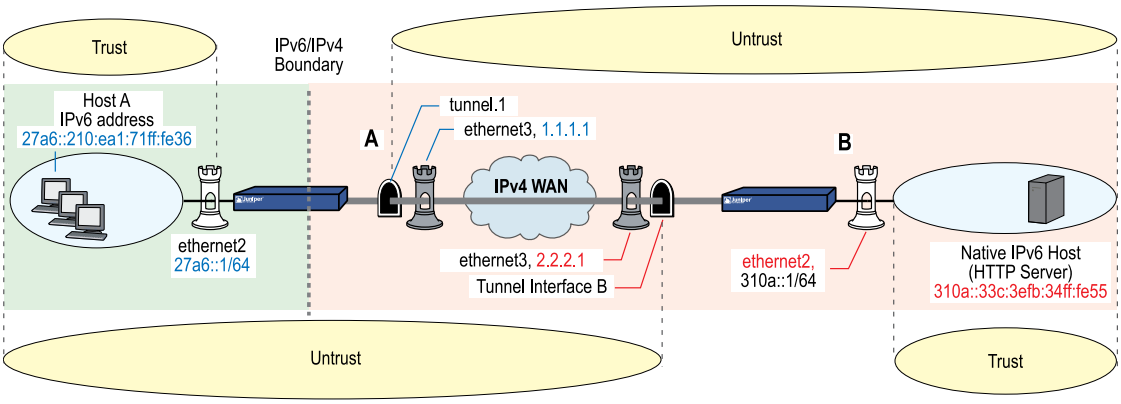
```
set policy from untrust to trust R_Clients L_Server any permit
```

IPsec 6in4 Tunneling

You can use IPsec tunneling, which supports authentication and encryption, to encapsulate packets as the security device transmits them between remote IPv6 island networks over an IPv4 WAN.

Figure 531 on page 2212 shows an IPsec tunnel between remote IPv6 island networks. In this example, you first define a tunnel interface on each gateway device; then you bind the tunnel to a zone that borders IPv4 network space.

Figure 531: Tunnel Interface and Zone Example



A device addresses packets that pass between a host in an IPv6 island network and another IPv6 host over an IPv4 WAN backbone in the following manner:

1. Host A generates an IPv6 service request packet and addresses it to Host B. The packet goes from Zone A to Zone B.
2. Device A encapsulates the outgoing IPv6 packet inside an IPv4 packet and sends it out the tunnel interface into Zone B.
3. Device B decapsulates the packet and forwards it to Host B.
4. Device B generates a reply packet, encapsulates it, and forwards it to Device A.
5. Device A receives the encapsulated reply packet through the tunnel interface. Device A decapsulates the packet and forwards it to Host A.

In most cases, the necessary setup tasks are as follows:

1. Configure an interface for communication with the protected island IPv6 network.
   - Bind the interface to a zone (typically the Trust zone).
   - Configure the interface for IPv6, host mode.
   - Assign the interface a global unicast prefix. (For information about global unicast addresses, see “Address Types” on page 2090.)
2. Configure an interface for communication over the IPv4 WAN.
   - Bind the interface to a zone (typically the Untrust zone).
   - Assign the interface an IPv4 address and subnet mask.
   - Create a tunnel interface (unnumbered) in the zone and bind it to the IPv4 interface.
3. Follow steps 1 and 2 to set up interfaces on the peer device.
4. Set up IPsec between the peer devices. For more information about IPsec, see *“Virtual Private Networks” on page 705*.
5. On each device, create address book entries that identify the IPv6 host, subnet, or network.
6. Set up routing entries that allow the hosts to access each other.
7. Set up security policies.

   In the following example, you create an IPsec tunnel between IPv4 endpoints. IPv6 devices behind Device A transmit service requests to hosts behind Device B. Device A use IPsec to encapsulate the service request packets inside IPv4 packets. Device B receives and decapsulates the packets. See Figure 531 on page 2212.

   The devices perform Phase 1 of the AutoKey IKE tunnel negotiation as follows:
   - RSA authentication
   - Diffie-Hellman Group 2
   - AES128 encryption algorithm
   - SHA hashing algorithm

   The devices perform Phase 2 of the tunnel negotiation uses the Standard proposal.
   - Diffie-Hellman Group 2
   - ESP (Encapsulating Security Payload) tunneling
   - 3DES encryption algorithm
   - SHA hashing algorithm

WebUI (Device A)

Interfaces

Network > Interfaces > Edit (for ethernet2)

Network > Interfaces > Edit (for ethernet2) > IPv6

Network > Interfaces > Edit (for ethernet3)

Tunnels

Network > Interfaces > New (Tunnel IF)

IKE

VPNs > Autokey Advanced > Gateway > New

VPN

VPNs > Autokey IKE > New

Routers

Network > Routing > Routing Table > New (trust-vr)

Addresses

Policy > Policy Elements > Addresses > List (Trust) > New

Policy > Policy Elements > Addresses > List (Untrust) > New

Policy

Policies > (From: Trust, To: Untrust) > New

CLI (Device A)

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 10.1.1.1/24
set interface ethernet2 ipv6 mode host
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 27a6::1/64
set interface ethernet3 zone untrust
set interface ethernet3 ip 1.1.1.1/24
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.1 ipv6 mode host
set interface tunnel.1 ipv6 enable
```

3. **IKE**

```
set ike gateway IPSec_Servers ip 2.2.2.1 main outgoing-interface ethernet3
  proposal rsa-g2-aes128-sha
```

4. **VPN**

```
set vpn Tunnel_Servers gateway IPSec_Servers no-replay tunnel sec-level standard
set vpn Tunnel_Servers bind interface tunnel.1
```

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3
set vrouter trust-vr route ::/0 interface tunnel.1
```

6. **Addresses**

```
set address trust L_Clients 27a6::210:ea1:71ff:fe36/64
set address untrust R_Servers 32f1::250:af:34ff:fe34/64
```

7. **Policy**

```
set policy from trust to untrust L_Clients R_Servers any permit
```

WebUI (Device B)

Interfaces

Network > Interfaces > Edit (for ethernet2)

Network > Interfaces > Edit (for ethernet2) > IPv6

Network > Interfaces > Edit (for ethernet3)

Tunnels

Network > Interfaces > New (Tunnel IF)

IKE

VPNs > Autokey Advanced > Gateway > New

VPN

VPNs > Autokey IKE > New

Routers

Network > Routing > Routing Table > New (trust-vr)

Network > Routing > Route Table > New (untrust-vr)

Addresses

Policy > Policy Elements > Addresses > List (Trust) > New

Policy > Policy Elements > Addresses > List (Untrust) > New

Policy

Policies > (From: Untrust, To: Trust) > New

CLI (Device B)

1. **Interfaces**

```
set interface ethernet2 zone trust
set interface ethernet2 ip 10.2.2.1/24
set interface ethernet2 ipv6 mode host
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 ip 310a::1/64
set interface ethernet3 zone untrust
set interface ethernet3 ip 2.2.2.1/24
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ip unnumbered interface ethernet3
set interface tunnel.1 ipv6 mode host
set interface tunnel.1 ipv6 enable
```

3. **IKE**

```
set ike gateway IPSec_Clients ip 1.1.1.1 main outgoing-interface ethernet3
  proposal rsa-g2-aes128-sha
```

4. **VPN**

```
set vpn Tunnel_Clients id 1 gateway IPSec_Clients no-replay tunnel sec-level
  standard
set vpn Tunnel_Clients id 2 bind interface tunnel.1
```

5. **Routes**

```
set vrouter trust-vr route 0.0.0.0/0 interface ethernet3
set vrouter trust-vr route ::/0 interface tunnel.1
```

6. **Addresses**

```
set address trust L_Server 310a::33c:3efb:34ff:fe55/128
set address untrust R_Clients 27a6::210:ea1:71ff:fe36/64
```

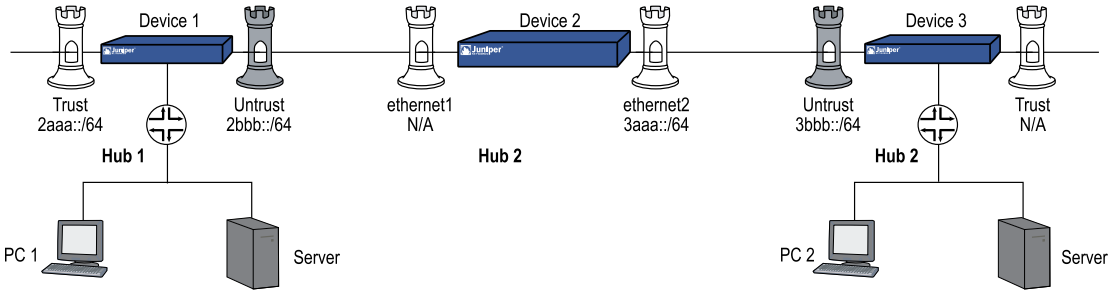
7. **Policy**

```
set policy from untrust to trust R_Clients L_Server any tunnel vpn Tunnel_Clients
```

Manual Tunneling with Fragmentation Enabled

Figure 532 on page 2217 shows the configuration of the network used in the following tunneling examples.

Figure 532: Manual Tunneling Example



IPv6 to IPv6 Route-Based VPN Tunnel

In the following example, you send IPv6 packets through an IPv6 tunnel, with a route-based VPN that uses 3DES encryption, and you enable fragmentation.

CLI (Device 1)

1. **General**

```
set console time 0
unset zone untrust block
```

2. **Interfaces**

```
set interface ethernet1/1 zone trust
set interface ethernet1/1 ipv6 mode router
set interface ethernet1/1 ipv6 ip 2aaa::/64
set interface ethernet1/1 ipv6 enable
set interface ethernet1/1 ipv6 interface-id 0000000000000001
set interface ethernet1/1 ipv6 ra transmit
set interface ethernet1/1 route
set interface ethernet1/1 manage

set interface ethernet1/2 zone untrust
set interface ethernet1/2 ipv6 mode host
set interface ethernet1/2 ipv6 ip 2bbb::/64
set interface ethernet1/2 ipv6 enable
set interface ethernet1/2 ipv6 interface-id 0000000000000002
set interface ethernet1/2 route
set interface ethernet1/2 manage
```

3. **IKE**

```
set ike gateway ton6 address 3aaa::5 outgoing-interface ethernet1/2 local-address
   2bbb::2 preshare abc sec-level standard
set vpn vpn4 gateway ton6 sec-level standard
```

4. **Tunnel**

```
set interface tunnel.6 zone untrust
set interface tunnel.6 ipv6 mode host
set interface tunnel.6 ipv6 enable
```

```
set interface tunnel.6 ip unnumbered interface ethernet1/2
set vpn vpn4 bind interface tunnel.6
```

5. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set policy from untrust to trust any-ipv6 any-ipv6 any permit
```

6. **Routes**

```
set vrouter trust-vr route ::/0 interface ethernet1/2 gateway 2bbb::3
set vrouter trust-vr route 3bbb::/64 interface tunnel.6
```

CLI (Device 2)

1. **General**

```
set console time 0
unset zone untrust block
```

2. **Interfaces**

```
set interface untrust zone untrust
set interface untrust ipv6 mode host
set interface untrust ipv6 enable
set interface untrust ipv6 ip 2bbb::/64
set interface untrust ipv6 interface-id 0000000000000003
set interface untrust manage
set interface untrust route

set interface trust zone trust
set interface trust ipv6 mode host
set interface trust ipv6 ip 3aaa::/64
set interface trust ipv6 enable
set interface trust ipv6 interface-id 0000000000000004
set interface trust route
set interface trust manage
```

3. **Routes**

```
set vrouter trust-vr route 3bbb::/64 interface trust gateway 3aaa::5
set vrouter trust-vr route 2aaa::/64 interface untrust gateway 2bbb::2
```

4. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set policy from untrust to trust any-ipv6 any-ipv6 any permit
```

CLI (Device 3)

1. **General**

```
set console time 0
unset zone untrust block
```

2. **Interfaces**

```
set interface untrust zone untrust
set interface untrust ipv6 mode host
set interface untrust ipv6 ip 3aaa::/64
set interface untrust ipv6 interface-id 0000000000000005
set interface untrust ipv6 enable
set interface untrust route
set interface untrust manage

set interface trust zone trust
set interface trust ipv6 mode router
set interface trust ipv6 ip 3bbb::/64
set interface trust ipv6 interface-id 0000000000000006
set interface trust ipv6 enable
set interface trust ipv6 ra transmit
set interface trust route
set interface trust manage
```

3. **Tunnel**

```
set interface tunnel.6 zone untrust
set interface tunnel.6 ipv6 mode host
set interface tunnel.6 ipv6 enable
set interface tunnel.6 ip unnumbered interface untrust
```

4. **IKE**

```
set ike gateway ton4 address 2bbb::2 outgoing-interface untrust local-address
  3aaa::5 preshare abc sec-level standard
```

5. **VPN**

```
set vpn vpn6 gateway ton4 sec-level standard
set vpn vpn6 bind interface tunnel.6
```

6. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set vpn vpn6 bind interface tunnel.6
```

7. **Routes**

```
set vrouter trust-vr route ::/0 interface untrust gateway 3aaa::4
set vrouter trust-vr route 2aaa::/64 interface tunnel.6
```

IPv4 to IPv6 Route-Based VPN Tunnel

In the following example, you send IPv4 packets through an IPv4 tunnel, with a route-based VPN that uses 3DES encryption, and you enable fragmentation.

CLI (Device 1)

1. **General**

```
set console time 0
unset zone untrust block
```

2. **Interfaces**

```
set interface ethernet1/1 zone trust
set interface ethernet1/1 ipv6 mode router
set interface ethernet1/1 ipv6 ip 2aaa::/64
set interface ethernet1/1 ipv6 enable
set interface ethernet1/1 ipv6 interface-id 0000000000000001
set interface ethernet1/1 ipv6 ra transmit
set interface ethernet1/1 route
set interface ethernet1/1 manage

set interface ethernet1/2 zone untrust
set interface ethernet1/2 ipv6 mode router
set interface ethernet1/2 ipv6 ip 2bbb::/64
set interface ethernet1/2 ipv6 enable
set interface ethernet1/2 ipv6 interface-id 0000000000000002
set interface ethernet1/2 ipv6 ra transmit
set interface ethernet1/2 route
set interface ethernet1/2 manage
```

3. **IKE**

```
set ike gateway ton6 address 3aaa::5 outgoing-interface ethernet1/2 local-address
  2bbb::2 preshare abc sec-level standard
set vpn vpn4 gateway ton6 sec-level standard
```

4. **Tunnel**

```
set interface tunnel.6 zone untrust
set interface tunnel.6 ipv6 mode router
set interface tunnel.6 ipv6 enable
set interface tunnel.6 ip unnumbered interface ethernet1/2
set interface tunnel.6 ipv6 ra transmit
```

5. **VPN**

```
set vpn vpn4 bind interface tunnel.6
```

6. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set policy from untrust to trust any-ipv6 any-ipv6 any permit
```

7. **Routes**

```
set vrouter trust-vr route ::/0 interface ethernet1/2 gateway 2bbb::3
set vrouter trust-vr route 3bbb::/64 interface tunnel.6
set interface ethernet1/1 ip 1.1.1.1/24
set route 4.1.1.0/24 interface tunnel.6
```

8. **Policies**

```
set policy from trust to untrust any any any permit
set policy from untrust to trust any any any permit
```

CLI (Device 2)

1. **General**

```
set console time 0
unset zone untrust block
```

2. **Interfaces**

```
set interface untrust zone untrust
set interface untrust ipv6 mode host
set interface untrust ipv6 enable
set interface untrust ipv6 interface-id 0000000000000003
set interface untrust ipv6 ra accept
set interface untrust manage
set interface untrust route

set interface trust zone trust
set interface trust ipv6 mode router
set interface trust ipv6 ip 3aaa::/64
set interface trust ipv6 enable
set interface trust ipv6 interface-id 0000000000000004
set interface trust ipv6 ra transmit
set interface trust route
set interface untrust manage
```

3. **Routes**

```
set vrouter trust-vr route 3bbb::/64 interface trust gateway 3aaa::5
set vrouter trust-vr route 2aaa::/64 interface untrust gateway 2bbb::2
```

4. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set policy from untrust to trust any-ipv6 any-ipv6 any permit
```

CLI (Device 3)

1. **General**

```
set console time 0
unset zone untrust block
```

2. **Interfaces**

```
set interface untrust zone untrust
set interface untrust ipv6 mode host
set interface untrust ipv6 interface-id 0000000000000005
set interface untrust ipv6 enable
set interface untrust ipv6 ra accept
set interface untrust route
set interface untrust manage

set interface trust zone trust
```

```
set interface trust ipv6 mode router
set interface trust ipv6 ip 3bbb::/64
set interface trust ipv6 interface-id 0000000000000006
set interface trust ipv6 enable
set interface trust ipv6 ra transmit
set interface trust route
set interface trust manage
```

3. **Tunnel**

```
set interface tunnel.6 zone untrust
set interface tunnel.6 ipv6 mode host
set interface tunnel.6 ipv6 enable
set interface tunnel.6 ip unnumbered interface untrust
set interface tunnel.6 ipv6 ra accept
```

4. **IKE**

```
set ike gateway ton4 address 2bbb::2 outgoing-interface untrust local-address
   3aaa::5 preshare abc sec-level standard
set vpn vpn6 gateway ton4 sec-level standard
set vpn vpn6 bind interface tunnel.6
```

5. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set policy from trust to untrust any any any permit

set policy from trust to untrust any any any permit
set policy from untrust to trust any any any permit
```

6. **Routes**

```
set vrouter trust-vr route ::/0 interface untrust gateway 3aaa::4
set vrouter trust-vr route 2aaa::/64 interface tunnel.6
set interface trust ip 4.1.1.1/24
set route 1.1.1.0/24 interface tunnel.6
```

Chapter 70
IPv6 XAuth User Authentication

This chapter describes IPv6 user authentication and Dead Peer Detection (DPD) and provides configuration examples. It contains the following sections:

Overview

ScreenOS XAuth for IPv6 allows you to authenticate individual users and user groups after Phase 1 IKE negotiations.

You can use XAuth to authenticate remote virtual private network (VPN) users (or user groups) individually instead of only authenticating VPN gateways and/or devices or to assign TCP/IP configuration information, such as IP address, netmask, DNS server, and WINS server assignments.

RADIUSv6

ScreenOS supports RADIUSv6, the next-generation version of RADIUS. When a device functions as an XAuth server, it sends a username and password to a RADIUS server enabled for IPv4, IPv6, or both.

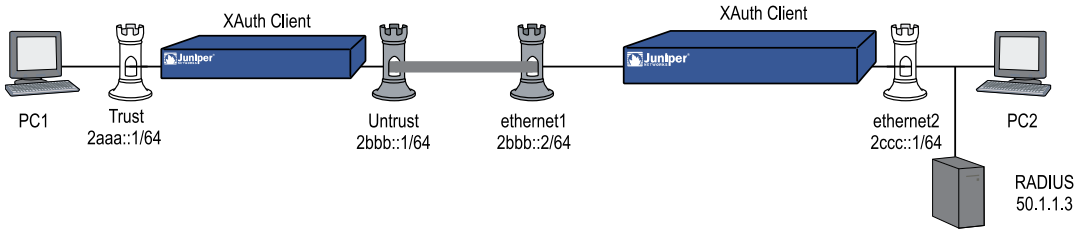
NOTE: For more information about RADIUS and IPv6, refer to RFCs 3162 and 2882.

The next sections show various scenarios. Configuration examples follow these scenarios.

Single Client, Single Server

Figure 533 on page 2224 shows a single client device and a single server interacting with a RADIUS server performing XAuth authentication.

Figure 533: RADIUS with a Single Client and Single Server

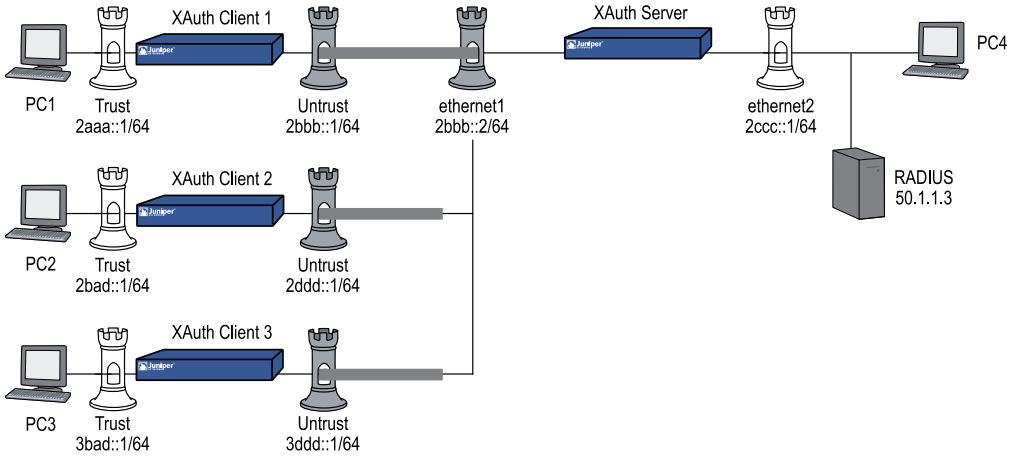


The RADIUS server resides in the Trust zone of the XAuth server. Each protected subnet can contain more than a single protected host, but the figure shows only one.

Multiple Clients, Single Server

Figure 534 on page 2224 shows multiple client devices and a single server interacting with a RADIUS server performing XAuth authentication.

Figure 534: RADIUS with Multiple Clients and a Single Server

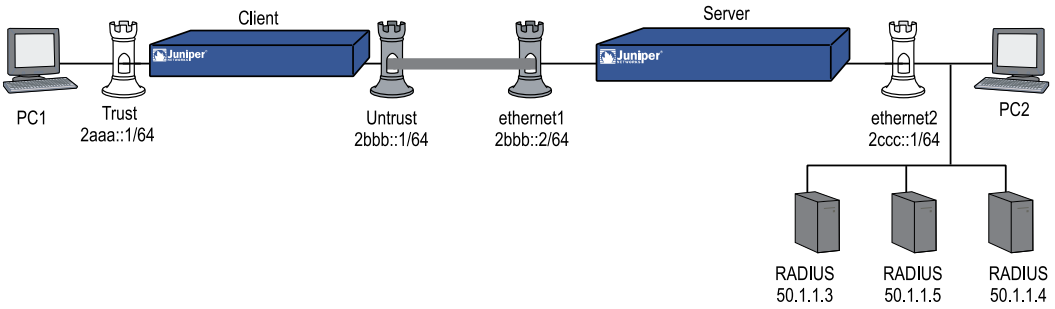


The RADIUS server resides in the Trust zone of the server. Each client device protects a different subnet. Each protected subnet can contain more than a single host, but the figure only shows one.

Single Client, Multiple Servers

Figure 535 on page 2225 shows a single client device and a single server interacting with multiple RADIUS servers.

Figure 535: RADIUS with a Single Client and Multiple Servers

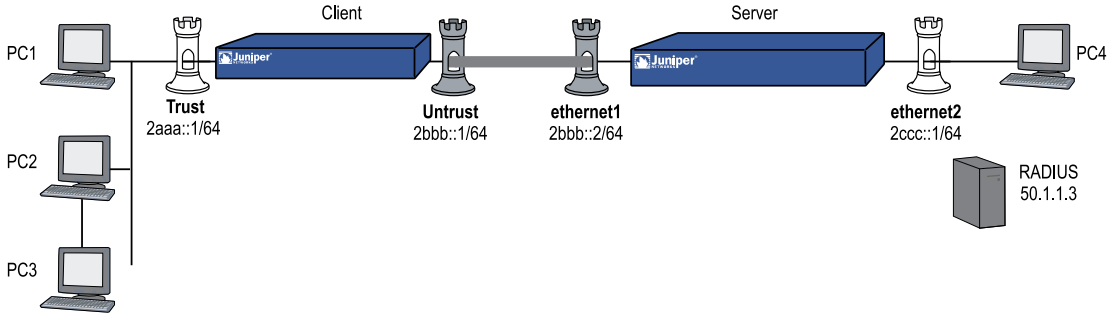


The RADIUS servers reside in the Trust zone of Device 2. Each protected subnet can contain more than a single host, but the figure only shows one.

Multiple Hosts, Single Server

Figure 536 on page 2225 shows a single client device (Device 1) protecting multiple hosts and a single server (Device 2) interacting with a RADIUS server.

Figure 536: RADIUS with Multiple Hosts and a Single Server



The RADIUS server resides in the Trust zone of the security server.

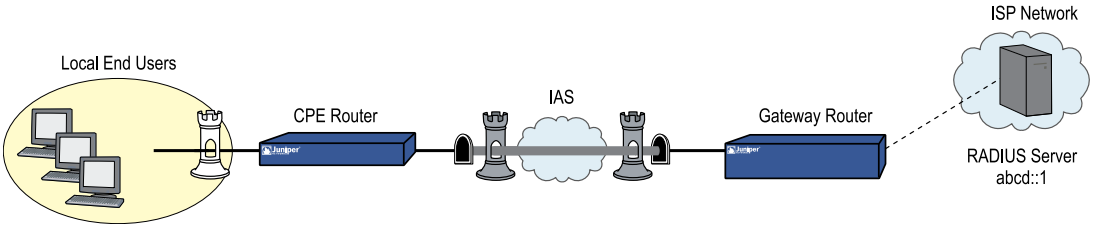
IPsec Access Session Management

You can use several security devices to allow user networks to access outside networks by setting up one or more local customer premises equipment (CPE) routers and one or more gateway routers. To protect packets transmitted over the access network, the CPE and gateway devices can establish an IPsec tunnel. The gateway, together with the RADIUS server manages and records accounting for the network access sessions.

IPsec Access Session

The time interval during which a network access session exists is called an IPsec Access Session (IAS). The IAS time interval begins when the first end user connects to the access network and ends when the last user disconnects from the network. Figure 537 on page 2226 shows how an ISP might use security devices as CPE and gateway routers.

Figure 537: IPsec Access Session with RADIUS Server



To initiate an IAS, the CPE device performs a Phase 1 Internet Key Exchange (IKE) and at least one Phase 2 IKE.

NOTE: For more information about ScreenOS security concepts and IKE, see *"Internet Protocol Security" on page 707*.

Each IAS has the following characteristics.

- A CPE device can have only one IAS Phase 1 Security Association (SA) for each IP address. For example, if the CPE device has three IPv6 addresses, it can host three Phase 1 SAs. However, there might be multiple Phase 2 SAs for each Phase 1 SA.
- During an IAS lifetime, the CPE and gateway devices might perform many IPsec operations, such as IKE Phase 1 and Phase 2 rekeying.
- During the IAS lifetime, the CPE initiates Phase 1 SA rekeys whenever the Phase 1 session expires.
- During the IAS lifetime, the CPE must initiate a new Phase 2 SA rekey *before* a Phase 2 session expires to ensure that SA is always on.

While it exists, each IAS contains and uses the following components:

Information Field	Description
CPE IPv6 Address	The address of the CPE device. The address is globally unique; that is, there cannot be more than one IAS with the same CPE IP address.
IKE Phase I ID	The ID that identifies the Phase 1 gateway for the IAS. This ID can consist of any of the following: ■ An alphanumeric string that identifies the gateway device ■ The ASN1 domain name of the gateway device ■ The Fully Qualified Domain Name (FQDN) of the gateway device ■ The IP address (IPv4 or IPv6) of the gateway device ■ The user-Fully Qualified Domain Name (u_FQDN) of the gateway device
XAuth User Name	The name of the XAuth user.
IKE Phase 1 Cookie Pair	The IKE Phase 1 SA cookies, which show the Phase 1 SA associated with the IAS.

Information Field	Description
Number of Phase 2 SAs	The number of Phase 2 SAs that currently exist in the IAS. For the IAS to exist, there must be one or more Phase 2 SAs.
Assigned IP address	The IP4 address assigned from the gateway to the CPE during the IKE mode configuration phase.
Phase 2 SA SPIs	Security Parameter Index (SPI) values, including both new and old SPIs.
IAS Start Time	The time the IAS began.

When a CPE gateway device initiates an IAS successfully, it starts RADIUS accounting for that IAS. When a Phase 2 SA expires or fails, the CPE device must establish another SA to continue secure communication with the gateway device. How the RADIUS server handles a Phase 2 rekey operation depends on whether IAS functionality is currently enabled or disabled.

When the IAS feature is disabled, the RADIUS accounting session starts over after the CPE establishes a new SA. Each time a Phase 2 session fails or times out, a new accounting session is necessary; and the RADIUS server must generate multiple billings (and, in some cases, other account-related features).

When the IAS feature is enabled, however, the original RADIUS accounting session resumes after each Phase 2 SA rekeying operation occurs. This allows the RADIUS server to maintain a single continuous accounting session to simplify billing and other accounting-related features.

Enabling and Disabling IAS Functionality

To enable the CPE gateway device for IAS functionality, enter the following command:

```
set ipsec-access-session enable
```

To disable the CPE gateway device for IAS functionality, enter the following command:

```
unset ipsec-access-session enable
```

Releasing an IAS Session

You can release an IPsec access session with the **clear ike all** command.

Limiting IAS Settings

The following settings limit the number of concurrently active IASs:

- The *maximum* specifies the maximum number of concurrent IASs the device allows. The default is 5000 sessions, and the range is from 0 to 5000 sessions.

  To configure this limitation, enter the following command:

  ```
  set ipsec access-session maximum maximum_number
  ```

- The *lower IAS threshold* specifies the minimum number of concurrent IASs the device allows before triggering a SNMP trap. The default is 1000 sessions, and the range is from 1 to 5000 sessions. This value must be less than the upper-threshold value.

  To configure this limitation, enter the following command:

  set ipsec access-session lower-threshold *minimum_number*

- The *upper IAS threshold* specifies the minimum number of concurrent IASs the device allows before triggering a SNMP trap. The default is 1000 sessions, and the range is from 1 to 5000 sessions. This value must be greater than the lower-threshold value.

  To configure this limitation, enter the following command:

  set ipsec access-session upper-threshold *maximum_number*

Dead Peer Detection

Dead Peer Detection (DPD) is a protocol that verifies the existence and liveliness of other IPsec peer devices when no incoming IPsec traffic from peers is received during a specified interval.

A device performs DPD by sending encrypted IKE Phase 1 notification payloads (R-U-THERE) to peers and waiting for DPD acknowledgements (R-U-THERE-ACK) from its peers. The device sends an R-U-THERE request only if it has not received any traffic from the peer during a specified DPD interval. If a DPD-enabled device receives traffic on a tunnel, it resets the R-U-THERE counter for that tunnel to start a new interval. If the device receives an R-U-THERE-ACK from the peer device during this interval, it determines the peer to be alive. If the device does not receive an R-U-THERE-ACK response during the interval, it determines the peer to be dead.

The device removes the Phase 1 SA and all Phase 2 SAs for dead peers. . In previous ScreenOS releases, the active tunnel could not fail over to another tunnel and gateway in the group. To maintain continuous connectivity even if a peer goes dead, in the current ScreenOS release the dead peer fails over the tunnel to another group member with the second highest weight. Meanwhile, the device attempts to renegotiate the tunnel with the peer identified as dead. Once the tunnel is successfully negotiated, the tunnel automatically fails back to the first member. The weighting system always causes the best ranked gateway in the group to handle the VPN data whenever possible

NOTE: The device also renegotiates the tunnel if the lifetime of the Phase 1 SA is about to expire.

You can configure the following DPD parameters with the CLI or the WebUI:

- The **interval** parameter specifies the DPD interval. This interval is the amount of time (expressed in seconds) the device allows to pass before considering a peer to be dead. A setting of zero disables DPD.
- The **always-send** parameter instructs the device to send DPD requests regardless of whether there is IPsec traffic with the peer.
- The **retry** parameter specifies the maximum number of times to send the R-U-THERE request before considering the peer to be dead. As with an IKE heartbeat configuration, the default number of transmissions is 5 times, with a permissible range from 1to 128 attempts.
- The **reconnect** parameter instructs the device to renegotiate the tunnel at the specified interval with the peer considered to be dead. As with an IKE recovery configuration, the minimum setting is 60 seconds. You can set the interval at any value between 60 and 9999 seconds.

NOTE: You can enable DPD or IKE heartbeat but not both. If you attempt to configure DPD after enabling the IKE heartbeat, the security device generates the following error message:

```
IKE: DPD cannot co-exist with IKE heartbeat.
```

The device generates a similar error if you attempt to configure an IKE heartbeat after enabling DPD.

Configuration Examples

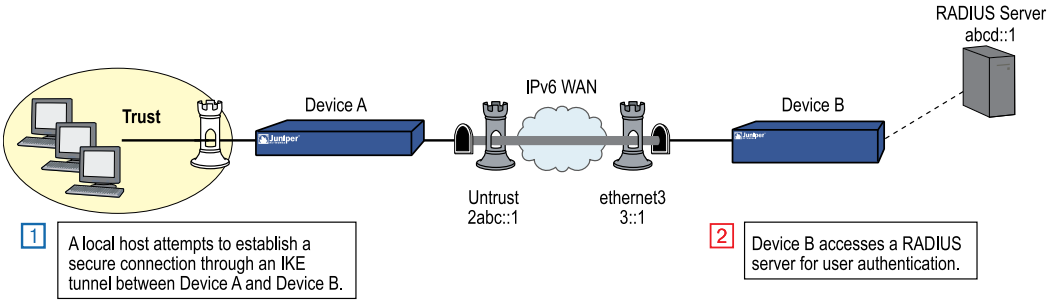
This section contains examples for RADIUS, IAS, and DPD.

NOTE: The WebUI section of each example lists only the navigational paths to the device configuration pages. For specific values, see the CLI section that follows it.

XAuth with RADIUS

Figure 538 on page 2230 shows device A performing XAuth authentication on users in its local network and a device B using a RADIUS server to perform XAuth in response to authentication through an IKE tunnel.

Figure 538: XAuth Example



WebUI (XAuth Client)

VPNs > AutoKey IKE > New

VPNs > AutoKey IKE > Edit

VPNs > AutoKey IKE > Advanced

CLI (XAuth Client)

```
set ike gateway client_gw6 address 3::1 main outgoing-interface untrust local-address
   2abc::1 preshare 1234 proposal pre-g2-des-md5
set ike gateway client_gw6 cert peer-cert-type x509-sig
set ike gateway client_gw6 xauth client any username chris password swordfish
```

WebUI (XAuth Server)

1. **RADIUS**

   Configuration > Auth > Auth Servers > Edit

2. **IKE**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

   VPNs > AutoKey IKE > Advanced

CLI (XAuth Server)

1. **RADIUS**

   ```
   set auth-server xauth-rad id 1
   set auth-server xauth-rad server-name 1.1.1.1
   set auth-server xauth-rad account-type xauth
   set auth-server xauth-rad radius secret "juniper"
   set xauth default auth server xauth-rad query-config
   ```

2. **IKE**

```
set ike gateway server_gw6 address 2abc::1 aggressive outgoing-interface
   ethernet3 local-address 3::1 preshare 1234 proposal pre-g2-des-md5
set ike gateway server_gw6 cert peer-cert-type x509-sig
set ike gateway server_gw6 xauth server xauth-rad query-config
```

RADIUS with XAuth Route-Based VPN

In this example, an XAuth client performs authentication and authorization using XAuth and RADIUS, with prefix delegation and mode config, over a route-based VPN. This delegates the IP address obtained from the RADIUS server to a tunnel interface on XAuth client (unless the client specifies another interface).

NOTE: The VPN in this example is route-based. In a policy-based VPN, or if there is a framed netmask other than "/32" , the device installs the IP address on the tunnel interface.

See Figure 533 on page 2224.

The steps to configure this example are as follows:

1. Configure an XAuth client to perform IPv6 CPE operation (for VPN).
2. Configure an XAuth IPv6 gateway (for VPN) to perform server configuration.
3. Enable an XAuth server (RAD1) with IP address 50.1.1.3. Give RADIUS a shared secret, with port number 1812 for RADIUS.
4. Specify a valid username (PC1) and a valid password (PC1) for the user from the XAuth client.

WebUI (XAuth Client)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.6) > IPv6

3. **Route**

   Network > Routing > Routing Table > New (trust-vr)

4. **IKE**

VPNs > AutoKey Advanced > Gateway > New

VPNs > AutoKey Advanced > Gateway > Edit

5. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

6. **Policies**

   Policies > (From: Untrust, To: Trust) > New

   Policies > (From: Trust, To: Untrust) > New

CLI (XAuth Client)

1. **Interfaces**

```
set interface trust ipv6 mode router
set interface trust manage
set interface trust ipv6 enable
set interface trust ipv6 interface-id 1111111111111111
set interface trust ipv6 ip 2aaa::1/64
set interface trust ipv6 ra transmit
set interface trust route
set interface untrust ipv6 mode router
set interface untrust manage
set interface untrust ipv6 enable
set interface untrust ipv6 interface-id 1222222222222222
set interface untrust ipv6 ip 2bbb::1/64
set interface untrust route
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ipv6 mode router
set interface tunnel.1 ipv6 enable
set interface tunnel.1 ipv6 unnumbered
```

3. **Routes**

```
set vrouter trust route ::/0 interface untrust gateway 2bbb::2
set vrouter trust route 2ccc::0/64 interface tunnel.1
```

4. **IKE**

```
set ike gateway NS1_NS2 add 2bbb::2 outgoing-interface untrust local-address
  2bbb::1 preshare abc123 sec-level standard
set ike gateway NS1_NS2 xauth client any username PC1 password PC1
```

5. **VPN**

```
set vpn NS1toNS2 gateway NS1_NS2 sec-level standard
set vpn NS1toNS2 bind interface tunnel.1
```

6. **Policies**

```
set policy from untrust to trust any-ipv6 any-ipv6 any permit
set policy from trust to untrust any-ipv6 any-ipv6 any permit
```

WebUI (XAuth Server)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.6) > IPv6

3. **Route**

   Network > Routing > Routing Table > New (trust-vr)

4. **RADIUS**

   Configuration > Auth > Auth Servers > New

   Configuration > Auth > Auth Servers > Edit

5. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit

6. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

7. **Policies**

   Policies > (From: Untrust, To: Trust) > New

   Policies > (From: Trust, To: Untrust) > New

CLI (XAuth Server)

1. **Interfaces**

```
set interface ethernet1 zone untrust
set interface ethernet1 ipv6 mode router
set interface ethernet1 manage
set interface ethernet1 ipv6 enable
set interface ethernet1 ipv6 interface-id 2111111111111111
set interface ethernet1 ipv6 ip 2bbb::2/64
set interface ethernet1 route
set interface ethernet2 zone trust
set interface ethernet2 ipv6 mode router
set interface ethernet2 manage
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 interface-id 2222222222222222
set interface ethernet2 ipv6 ip 2ccc::1/64
set interface ethernet2 ip 50.1.1.1/24
set interface ethernet2 route
set interface ethernet2 ipv6 ra transmit
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ipv6 mode router
set interface tunnel.1 ipv6 enable
```

3. **Routes**

```
set vrouter trust route ::/0 interface ethernet1 gateway 2bbb::1
set vrouter trust route 2aaa::0/64 interface tunnel.1
```

4. **RADIUS**

```
set auth-server RAD1 id 1
set auth-server RAD1 server-name 50.1.1.3
set auth-server RAD1 account-type xauth
set auth-server RAD1 radius secret juniper
set auth-server RAD1 radius port 1812
```

5. **IKE**

```
set ike gateway NS2_NS1 add 2bbb::1 outgoing-interface ethernet1 local-address
   2bbb::2 preshare abc123 sec-level standard
set ike gateway NS2_NS1 xauth server RAD1 query-config user PC1
```

6. **VPN**

```
set vpn NS2toNS1 gateway NS2_NS1 sec-level standard
set vpn NS2toNS1 bind interface tunnel.1
```

7. **Policies**

```
set policy from untrust to trust any-ipv6 any-ipv6 any permit
set policy from trust to untrust any-ipv6 any-ipv6 any permit
```

RADIUS with XAuth and Domain Name Stripping

In the following example, you use the domain name checking feature, which checks the domain name (starting from the right most character to the left most until it finds the separator character). This feature enables the device to allow users from a particular domain only.

See Figure 533 on page 2224.

The steps to configure this example are as follows:

1. Enable IPv6 CPE configuration (for VPN) and configure XAuth on the XAuth client.
2. Enable IPv6 gateway configuration (for VPN) on the XAuth server.
3. Enable the XAuth server as Radius RAD1, with IP address 50.1.1.3. Give it RADIUS shared secret (juniper) and port number (1812).
4. Enter commands that do the following:

- On the XAuth client: Provide the username (PC1@juniper.net) and password (PC1) for that user.
- On the XAuth server:
  - Configure the username that comes from the XAuth client to be PC1@juniper.net.
  - Allow only users from domain “juniper.net.”
  - Strip the username from right to left until one separator character (@) is found.

WebUI (XAuth Client)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.6) > IPv6

3. **Route**

   Network > Routing > Routing Table > New (trust-vr)

4. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit

5. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

6. **Policies**

   Policies > (From: Untrust, To: Trust) > New

   Policies > (From: Trust, To: Untrust) > New

CLI (XAuth Client)

1. **Interfaces**

```
set interface trust ipv6 mode router
set interface trust manage
set interface trust ipv6 enable
set interface trust ipv6 interface-id 1111111111111111
set interface trust ipv6 ip 2aaa::1/64
set interface trust ipv6 ra transmit
set interface trust route
set interface untrust ipv6 mode router
set interface untrust manage
set interface untrust ipv6 enable
set interface untrust ipv6 interface-id 1222222222222222
set interface untrust ipv6 ip 2bbb::1/64
set interface untrust route
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ipv6 mode router
set interface tunnel.1 ipv6 enable
```

3. **IKE**

```
set ike gateway NS1_NS2 add 2bbb::2 outgoing-interface untrust local-address
  2bbb::1 preshare abc123 sec-level standard
set ike gateway NS1_NS2 xauth client any username PC1@juniper.net password
  PC1
```

4. **VPN**

```
set vpn NS1toNS2 gateway NS1_NS2 sec-level standard
set vpn NS1toNS2 bind interface tunnel.1
```

5. **Routers**

```
set vrouter trust route ::/0 interface untrust gateway 2bbb::2
set vrouter trust route 2ccc::0/64 interface tunnel.1
```

6. **Policies**

```
set policy from untrust to trust any-ipv6 any-ipv6 any permit
set policy from trust to untrust any-ipv6 any-ipv6 any permit
```

WebUI (XAuth Server)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.6) > IPv6

3. **Route**

   Network > Routing > Routing Table > New (trust-vr)

4. **RADIUS**

   Configuration > Auth > Auth Servers > New

   Configuration > Auth > Auth Servers > Edit

5. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit

6. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

7. **Policies**

   Policies > (From: Untrust, To: Trust) > New

   Policies > (From: Trust, To: Untrust) > New

CLI (XAuth Server)

1. **Interfaces**

```
set interface ethernet1 zone untrust
set interface ethernet1 ipv6 mode router
set interface ethernet1 manage
set interface ethernet1 ipv6 enable
set interface ethernet1 ipv6 interface-id 2111111111111111
set interface ethernet1 ipv6 ip 2bbb::2/64
set interface ethernet1 route
set interface ethernet2 zone trust
set interface ethernet2 ipv6 mode router
set interface ethernet2 manage
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 interface-id 2222222222222222
set interface ethernet2 ipv6 ip 2ccc::1/64
set interface ethernet2 ip 50.1.1.1/24
set interface ethernet2 route
set interface ethernet2 ipv6 ra transmit
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ipv6 mode router
set interface tunnel.1 ipv6 enable
```

3. **Routes**

```
set vrouter trust route ::/0 interface ethernet1 gateway 2bbb::1
set vrouter trust route 2aaa::0/64 interface tunnel.1
```

4. **RADIUS**

```
set auth-server RAD1 id 1
set auth-server RAD1 server-name 50.1.1.3
set auth-server RAD1 account-type xauth
set auth-server RAD1 radius secret juniper
set auth-server RAD1 radius port 1812
set auth-server RAD1 username domain juniper.net
set auth-server RAD1 username separator @ number 1
```

5. **IKE**

```
set ike gateway NS2_NS1 add 2bbb::1 outgoing-interface ethernet1 local-address
   2bbb::2 preshare abc123 sec-level standard
set ike gateway NS2_NS1 xauth server RAD1 query-config user PC1@juniper.net
```

6. **VPN**

```
set vpn NS2toNS1 gateway NS2_NS1 sec-level standard
set vpn NS2toNS1 bind interface tunnel.1
```

7. **Policies**

```
set policy from untrust to trust any-ipv6 any-ipv6 any permit
set policy from trust to untrust any-ipv6 any-ipv6 any permit
```

IP Pool Range Assignment

A device assigns a user an IP address from its local IP pool when a RADIUS server returns a framed-ip-address of 255.255.255.254. If all the IP addresses in range contained in the default IP pool are used up, the device uses IP addresses from the next IP range.

See Figure 534 on page 2224.

The steps to configure this example are as follows:

1. Enable IPv6 CPE configuration (for VPN) and XAuth client configuration on Client 1, Client 2, and Client 3.
2. On the XAuth server:
   - Enable the IPv6 gateway configuration (for VPN).
   - Configure a tunnel for Client 1, Client 2, and Client 3.
   - Enable XAuth server (RAD1) with IP address 50.1.1.3. Specify the RADIUS shared secret (juniper) and port number (1812).
   - Define a local IP pool (P1) containing two IP ranges.
     - 80.1.1.1 to 80.1.1.2
     - 80.1.1.3 to 80.1.1.4.
   - Enable the default XAuth IP pool (P1).
   - Specify valid usernames (PC4) and valid passwords (PC4) for those users from the XAuth clients (Device 1, Device 2, and Device 3).
3. On the RADIUS server, insert a framed IP address with a subnet mask of 255.255.255.254 for the user PC4.

WebUI (XAuth Client 1, XAuth Client 2, and XAuth Client 3)

NOTE: The WebUI paths are the same for each XAuth client. The values you enter for some fields differ. See the CLI commands for the correct values.

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.6) > IPv6

3. **Route**

   Network > Routing > Routing Table > New (trust-vr)

4. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit

5. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

6. **Policies**

   Policies > (From: Untrust, To: Trust) > New

   Policies > (From: Trust, To: Untrust) > New

CLI (XAuth Client 1)

1. **Interfaces**

```
set interface trust ipv6 mode router
set interface trust manage
set interface trust ipv6 enable
set interface trust ipv6 interface-id 1111111111111111
set interface trust ipv6 ip 2aaa::1/64
set interface trust ipv6 ra transmit
set interface trust route
set interface untrust ipv6 mode router
set interface untrust manage
set interface untrust ipv6 enable
set interface untrust ipv6 interface-id 1222222222222222
set interface untrust ipv6 ip 2bbb::1/64
set interface untrust route
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ipv6 mode router
set interface tunnel.1 ipv6 enable
```

3. **Routes**

```
set vrouter trust route ::/0 interface untrust gateway 2bbb::2
set vrouter trust route 2ccc::0/64 interface tunnel.1
```

4. **IKE**

```
set ike gat NS1_NS2 add 2bbb::2 outgoing-interface untrust local-add 2bbb::1
  preshare abc123 sec-level standard
set ike gateway NS1_NS2 xauth client any username PC4 password PC4
```

5. **VPN**

```
set vpn NS1toNS2 gateway NS1_NS2 sec-level standard
set vpn NS1toNS2 bind interface tunnel.1
```

6. **Policies**

```
set policy id 1 from untrust to trust any-ipv6 any-ipv6 any permit
set policy id 2 from trust to untrust any-ipv6 any-ipv6 any permit
```

CLI (XAuth Client 2)

1. **Interfaces**

```
set interface trust ipv6 mode router
set interface trust manage
set interface trust ipv6 enable
set interface trust ipv6 interface-id 3111111111111111
set interface trust ipv6 ip 2bad::1/64
set interface trust ipv6 ra transmit
set interface trust route
set interface untrust ipv6 mode router
set interface untrust manage
set interface untrust ipv6 enable
set interface untrust ipv6 interface-id 3222222222222222
set interface untrust ipv6 ip 2ddd::1/64
set interface untrust route
```

2. **Tunnel**

```
set interface tunnel.2 zone untrust
set interface tunnel.2 ipv6 mode router
set interface tunnel.2 ipv6 enable
```

3. **Routes**

```
set vrouter trust route ::/0 interface untrust gateway 2ddd::2
set vrouter trust route 2ccc::0/64 interface tunnel.2
```

4. **IKE**

```
set ike gateway NS3_NS2 add 2ddd::2 outgoing-interface untrust local-add 2ddd::1
  preshare abc123 sec-level standard
set ike gateway NS3_NS2 xauth client any username PC4 password PC4
```

5. **VPN**

```
set vpn NS3toNS2 gateway NS3_NS2 sec-level standard
set vpn NS3toNS2 bind interface tunnel.2
```

6. **Policies**

```
set policy id 1 from untrust to trust any-ipv6 any-ipv6 any permit
set policy id 2 from trust to untrust any-ipv6 any-ipv6 any permit
```

CLI (XAuth Client 3)

1. **Interfaces**

```
set interface trust ipv6 mode router
set interface trust manage
set interface trust ipv6 enable
set interface trust ipv6 interface-id 4111111111111111
set interface trust ipv6 ip 3bad::1/64
set interface trust ipv6 ra transmit
set interface trust route
set interface untrust ipv6 mode router
set interface untrust manage
set interface untrust ipv6 enable
set interface untrust ipv6 interface-id 4222222222222222
set interface untrust ipv6 ip 3ddd::1/64
set interface untrust route
```

2. **Tunnel**

```
set interface tunnel.3 zone untrust
set interface tunnel.3 ipv6 mode router
set interface tunnel.3 ipv6 enable
```

3. **Routes**

```
set vrouter trust route ::/0 interface untrust gateway 3ddd::2
set vrouter trust route 2ccc::0/64 interface tunnel.3
```

4. **IKE**

```
set ike gateway NS4_NS2 add 3ddd::2 outgoing-interface untrust local-add 3ddd::1
  preshare abc123 sec-level standard
set ike gateway NS4_NS2 xauth client any username PC4 password PC4
```

5. **VPN**

```
set vpn NS4toNS2 gateway NS4_NS2 sec-level standard
set vpn NS4toNS2 bind interface tunnel.3
```

6. **Policies**

```
set policy id 1 from untrust to trust any-ipv6 any-ipv6 any permit
set policy id 2 from trust to untrust any-ipv6 any-ipv6 any permit
```

WebUI (XAuth Server)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.6) > IPv6

3. **Route**

   Network > Routing > Routing Table > New (trust-vr)

4. **RADIUS**

   Configuration > Auth > Auth Servers > New

   Configuration > Auth > Auth Servers > Edit

5. **XAuth**

   Objects > IP Pools > New

   Objects > IP Pools > Edit

   VPNs > L2TP > Default Settings

6. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit

7. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

8. **Policies**

   Policies > (From: Untrust, To: Trust) > New

Policies > (From: Trust, To: Untrust) > New

CLI (XAuth Server)

1. **Interfaces**

```
set interface ethernet1 zone untrust
set interface ethernet1 ipv6 mode router
set interface ethernet1 manage
set interface ethernet1 ipv6 enable
set interface ethernet1 ipv6 interface-id 2111111111111111
set interface ethernet1 ipv6 ip 2bbb::2/64
set interface ethernet1 route
set interface ethernet2 zone trust
set interface ethernet2 manage
set interface ethernet2 ipv6 mode router
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 interface-id 2222222222222222
set interface ethernet2 ipv6 ip 2ccc::1/64
set interface ethernet2 ipv6 ra transmit
set interface ethernet2 ip 50.1.1.1/24
set interface ethernet2 route
set interface ethernet3 zone untrust
set interface ethernet3 manage
set interface ethernet3 ipv6 mode router
set interface ethernet3 ipv6 enable
set interface ethernet3 ipv6 interface-id 2333333333333333
set interface ethernet3 ipv6 ip 2ddd::2/64
set interface ethernet3 route
set interface ethernet4 zone untrust
set interface ethernet4 manage
set interface ethernet4 ipv6 mode router
set interface ethernet4 ipv6 enable
set interface ethernet4 ipv6 interface-id 2444444444444444
set interface ethernet4 ipv6 ip 3ddd::2/64
set interface ethernet4 route
```

2. **Tunnels**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ipv6 mode router
set interface tunnel.1 ipv6 enable
set interface tunnel.2 zone untrust
set interface tunnel.2 ipv6 mode router
set interface tunnel.2 ipv6 enable
set interface tunnel.3 zone untrust
set interface tunnel.3 ipv6 mode router
set interface tunnel.3 ipv6 enable
```

3. **Routes**

```
set vrouter trust route 2aaa::0/64 interface ethernet1 gateway 2bbb::1 metric 2
set vrouter trust route 2aaa::0/16 interface tunnel.1
set vrouter trust route 2bad::0/64 interface ethernet3 gateway 2ddd::1 metric 2
set vrouter trust route 2bad::0/16 interface tunnel.2
```

```
set vrouter trust route 3bad::0/64 interface ethernet4 gateway 2ddd::1 metric 2
set vrouter trust route 3bad::0/16 interface tunnel.3
```

4. **RADIUS**

```
set auth-server RAD1 id 1
set auth-server RAD1 server-name 50.1.1.3
set auth-server RAD1 account-type xauth
set auth-server RAD1 radius secret juniper
set auth-server RAD1 radius port 1812
```

5. **XAuth**

```
set ippool P1 80.1.1.1 80.1.1.2
set ippool P1 80.1.1.3 80.1.1.4
set xauth default ippool P1
```

6. **IKE**

```
set ike gateway NS2_NS1 add 2bbb::1 outgoing-interface ethernet1 local-add
  2bbb::2 preshare abc123 sec-level standard
set ike gateway NS2_NS3 add 2ddd::1 outgoing-interface ethernet3 local-add
  2ddd::2 preshare abc123 sec-level standard
set ike gateway NS2_NS4 add 3ddd::1 outgoing-interface ethernet4 local-add
  3ddd::2 preshare abc123 sec-level standard
set ike gateway NS2_NS1 xauth server RAD1 query-config user PC4
set ike gateway NS2_NS3 xauth server RAD1 query-config user PC4
set ike gateway NS2_NS4 xauth server RAD1 query-config user PC4
```

7. **VPN**

```
set vpn NS2toNS1 gateway NS2_NS1 sec-level standard
set vpn NS2toNS1 bind interface tunnel.1
set vpn NS2toNS3 gateway NS2_NS3 sec-level standard
set vpn NS2toNS3 bind interface tunnel.2
set vpn NS2toNS4 gateway NS2_NS4 sec-level standard
set vpn NS2toNS4 bind interface tunnel.3
```

8. **Policies**

```
set policy id 1 from untrust to trust any-ipv6 any-ipv6 any permit
set policy id 2 from trust to untrust any-ipv6 any-ipv6 any permit
```

RADIUS Retries

In an environment with a single XAuth client and XAuth server with multiple RADIUS servers, you can configure the number of RADIUS server retries before the XAuth server tries a backup server. The default number of retries is 3 (4 total). For example, if you configure the device to 20 retries, the device sends a total of 21 requests to the primary RADIUS server before trying the backup server.

this example, based on the topology shown in Figure 535 on page 2225, only shows the RADIUS configuration for the XAuth server.

WebUI (XAuth Server, RADIUS Configuration)

Configuration > Auth > Auth Servers > New

Configuration > Auth > Auth Servers > Edit

CLI (XAuth Server, RADIUS Configuration)

```
set auth-server RAD1 id 1
set auth-server RAD1 server-name 50.1.1.3
set auth-server RAD1 backup1 50.1.1.5
set auth-server RAD1 account-type xauth
set auth-server RAD1 radius secret juniper
set auth-server RAD1 radius port 1812
set auth-server RAD1 radius retries 20
```

Calling-Station-Id

The Calling-Station-Id attribute cannot be seen in the Access-Request packet and in the Accounting-Request packet by default. You can configure this attribute to appear in packets. The ike-ip address will be sent in the access request packet.

In the following example based on Figure 533 on page 2224, you configure an XAuth server to include the Calling-Station-Id attribute in packets. This example only shows the RADIUS configuration for the XAuth server.

WebUI (Device 2)

Configuration > Auth > Auth Servers > New

Configuration > Auth > Auth Servers > Edit

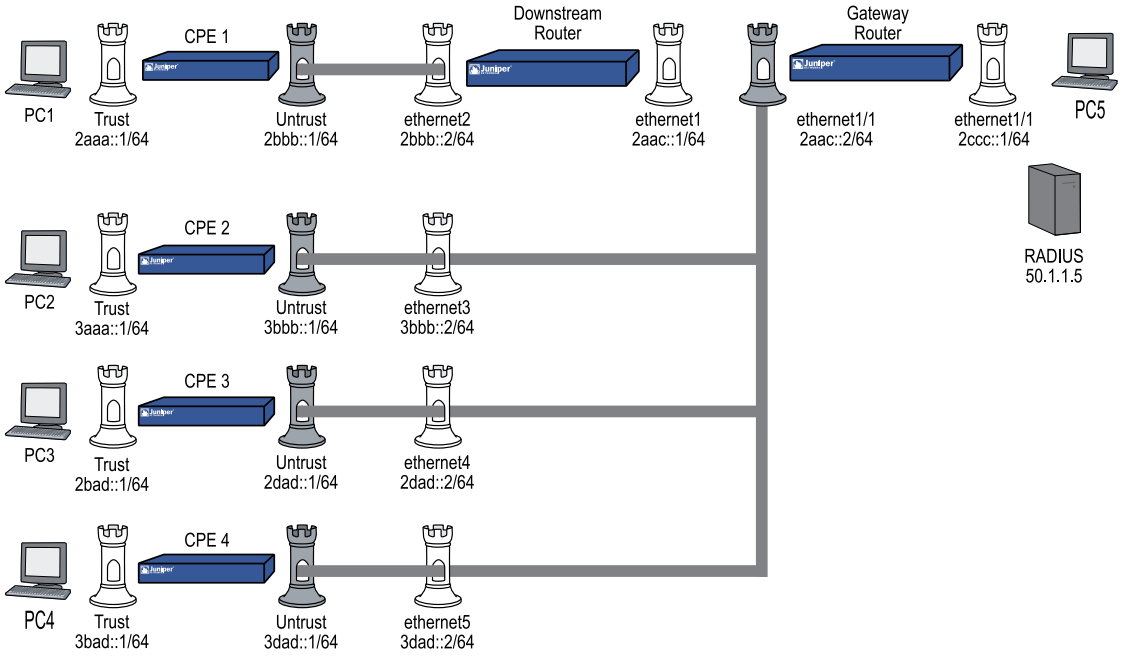
CLI (Device 2)

```
set auth-server RAD1 id 1
set auth-server RAD1 server-name 50.1.1.3
set auth-server RAD1 backup1 50.1.1.5
set auth-server RAD1 account-type xauth
set auth-server RAD1 radius secret juniper
set auth-server RAD1 radius port 1812
set auth-server RAD1 radius attribute calling-station-id
```

IPsec Access Session

Figure 539 on page 2247 shows four devices, configured as CPE routers, interacting with an upstream gateway router (Device 5) through a downstream router (Device 2). In all cases, ipsec-access-session feature is enabled, which allows the RADIUS server to maintain a single continuous RADIUS accounting session.

Figure 539: IPsec Access Session Example



In the following example, four CPEs establish tunnels to the same interface on the Gateway router (ethernet1, IP address 2aac::2). You initiate an IAS session on the Gateway, which allows the four different XAuth users (one on each CPE router), identifying themselves using separate usernames and passwords.

NOTE: The WebUI section lists only the navigational paths to the device configuration pages. For specific values, see the CLI section that follows it.

The basic steps to perform this example are as follows:

1. Configure all four CPE devices, the router, and the Gateway as described in the CLI below.
2. From each of the four CPEs, send an XAuth user (PC1, PC2, PC3, and PC11 with passwords PC1, PC2, PC3, and PC11, respectively). In addition, configure all the CPEs as XAuth clients.
3. From all the CPEs send domain names as “juniper.net” and configure VPN for aggressive mode negotiations.
4. On the Gateway, do the following:
   a. Perform the XAuth server configuration.
   b. Specify the RADIUS shared secret as juniper, and specify the port as 1812.
   c. Add a user group (group1) and add user (n1) with domain name “juniper.net” and share limit 10.

   On the RADIUS server, do the following:

a. Add users PC1, PC2, PC3, and PC4 and specify passwords PC1, PC2, PC3, and PC4 respectively.

b. Specify all the required attributes (such as framed-ip-address) to all the users.

WebUI (CPE 1, CPE 2, CPE 3, and CPE 4)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.6) > IPv6

3. **Routes**

   Network > Routing > Routing Table > New (trust-vr)

4. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit

5. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

6. **Policies**

   Policies > (From: Untrust, To: Trust) > New

   Policies > (From: Trust, To: Untrust) > New

CLI (CPE 1)

1. **Interfaces**

```
set interface ethernet1 zone trust
set interface ethernet1 manage
set interface ethernet1 ipv6 mode router
set interface ethernet1 ipv6 enable
set interface ethernet1 ipv6 interface-id 6111111111111111
set interface ethernet1 ipv6 ip 2aaa::1/64
```

```
set interface ethernet1 ipv6 ra transmit
set interface ethernet1 route

set interface ethernet2 zone untrust
set interface ethernet2 manage
set interface ethernet2 ipv6 mode host
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 interface-id 6222222222222222
set interface ethernet2 ipv6 ip 2bbb::1/64
set interface ethernet2 route
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ipv6 mode host
set interface tunnel.1 ipv6 enable
```

3. **Routes**

```
set vrouter trust-vr route ::/0 interface ethernet2 gateway 2bbb::2
set vrouter trust-vr route 2ccc::/64 interface tunnel.1
```

4. **IKE**

```
set ike gateway touniversal add 2aac::2 aggressive local-id juniper.net
  outgoing-interface ethernet2 local-address 2bbb::1 preshare abc123 sec-level
  standard
set ike gateway touniversal xauth client any username PC1 password PC1
```

5. **VPN**

```
set vpn vpn1 gateway touniversal sec-level standard
set vpn vpn1 bind interface tunnel.1
```

6. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set policy from untrust to trust any-ipv6 any-ipv6 any permit
```

CLI (CPE 2)

1. **Interfaces**

```
set interface trust zone trust
set interface trust manage
set interface trust ipv6 mode router
set interface trust ipv6 enable
set interface trust ipv6 interface-id 3111111111111111
set interface trust ipv6 ip 3aaa::1/64
set interface trust ipv6 ra transmit
set interface trust route
set interface untrust zone untrust
set interface untrust manage
set interface untrust ipv6 mode host
set interface untrust ipv6 enable
```

```
set interface untrust ipv6 interface-id 3222222222222222
set interface untrust ipv6 ip 3bbb::1/64
set interface untrust route
```

2. **Tunnel**

```
set interface tunnel.2 zone untrust
set interface tunnel.2 ipv6 mode host
set interface tunnel.2 ipv6 enable
```

3. **Routes**

```
set vrouter trust-vr route ::/0 interface untrust gateway 3bbb::2
set vrouter trust-vr route 2ccc::/64 interface tunnel.2
```

4. **IKE**

```
set ike gateway touniversal add 2aac::2 aggressive local-id juniper.net
  outgoing-interface untrust local-address 3bbb::1 preshare abc123 sec-level
  standard
set ike gateway touniversal xauth client any username PC2 password PC2
```

5. **VPN**

```
set vpn vpn2 gateway touniversal sec-level standard
set vpn vpn2 bind interface tunnel.2
```

6. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set policy from untrust to trust any-ipv6 any-ipv6 any permit
```

CLI (CPE 3)

1. **Interfaces**

```
set interface trust zone trust
set interface trust manage
set interface trust ipv6 mode router
set interface trust ipv6 enable
set interface trust ipv6 interface-id 4111111111111111
set interface trust ipv6 ip 2bad::1/64
set interface trust ipv6 ra transmit
set interface trust route
set interface untrust zone untrust
set interface untrust manage
set interface untrust ipv6 mode host
set interface untrust ipv6 enable
set interface untrust ipv6 interface-id 4222222222222222
set interface untrust ipv6 ip 2dad::1/64
set interface untrust route
```

2. **Tunnel**

```
set interface tunnel.3 zone untrust
```

```
set interface tunnel.3 ipv6 mode host
set interface tunnel.3 ipv6 enable
```

3. **Routes**

```
set vrouter trust-vr route ::/0 interface untrust gateway 2dad::2
set vrouter trust-vr route 2ccc::/64 interface tunnel.3
```

4. **IKE**

```
set ike gateway touniversal add 2aac::2 aggressive local-id juniper.net
  outgoing-interface untrust local-address 2dad::1 preshare abc123 sec-level
  standard
set ike gateway touniversal xauth client any username PC3 password PC3
```

5. **VPN**

```
set vpn vpn3 gateway touniversal sec-level standard
set vpn vpn3 bind interface tunnel.3
```

6. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set policy from untrust to trust any-ipv6 any-ipv6 any permit
```

CLI (CPE 4)

1. **Interfaces**

```
set interface trust zone trust
set interface trust manage
set interface trust ipv6 mode router
set interface trust ipv6 enable
set interface trust ipv6 interface-id 5111111111111111
set interface trust ipv6 ip 3bad::1/64
set interface trust ipv6 ra transmit
set interface trust route
set interface untrust zone untrust
set interface untrust manage
set interface untrust ipv6 mode host
set interface untrust ipv6 enable
set interface untrust ipv6 interface-id 5222222222222222
set interface untrust ipv6 ip 3dad::1/64
set interface untrust route
```

2. **Tunnel**

```
set interface tunnel.4 zone untrust
set interface tunnel.4 ipv6 mode host
set interface tunnel.4 ipv6 enable
```

3. **Routes**

```
set vrouter trust-vr route ::/0 interface untrust gateway 3dad::2
set vrouter trust-vr route 2ccc::/64 interface tunnel.4
```

4. **IKE**

```
set ike gateway touniversal add 2aac::2 aggressive local-id juniper.net
  outgoing-interface untrust local-address 3dad::1 preshare abc123 sec-level
  standard
set ike gateway touniversal xauth client any username PC11 password PC11
```

5. **VPN**

```
set vpn vpn4 gateway touniversal sec-level standard
set vpn vpn4 bind interface tunnel.4
```

6. **Policies**

```
set policy from trust to untrust any-ipv6 any-ipv6 any permit
set policy from untrust to trust any-ipv6 any-ipv6 any permit
```

WebUI (Device 2, Router)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Routes**

   Network > Routing > Routing Table > New (trust-vr)

CLI (Device 2, Router)

1. **Internet**

```
set interface ethernet1 zone trust
set interface ethernet1 manage
set interface ethernet1 ipv6 mode host
set interface ethernet1 ipv6 enable
set interface ethernet1 ipv6 interface-id 1111111111111111
set interface ethernet1 ipv6 ip 2aac::1/64
set interface ethernet1 route
set interface ethernet2 zone trust
set interface ethernet2 manage
set interface ethernet2 ipv6 mode host
set interface ethernet2 ipv6 enable
set interface ethernet2 ipv6 interface-id 1222222222222222
set interface ethernet2 ipv6 ip 2bbb::2/64
set interface ethernet2 route
set interface ethernet3 zone trust
set interface ethernet3 manage
set interface ethernet3 ipv6 mode host
```

```
set interface ethernet3 ipv6 enable
set interface ethernet3 ipv6 interface-id 1333333333333333
set interface ethernet3 ipv6 ip 3bbb::2/64
set interface ethernet3 route
set interface ethernet4 zone trust
set interface ethernet4 manage
set interface ethernet4 ipv6 mode host
set interface ethernet4 ipv6 enable
set interface ethernet4 ipv6 interface-id 1444444444444444
set interface ethernet4 ipv6 ip 2dad::2/64
set interface ethernet4 route
set interface ethernet5 zone trust
set interface ethernet5 manage
set interface ethernet5 ipv6 mode host
set interface ethernet5 ipv6 enable
set interface ethernet5 ipv6 interface-id 1555555555555555
set interface ethernet5 ipv6 ip 3dad::2/64
set interface ethernet5 route
unset zone trust block
```

2. **Routes**

```
set vrouter trust-vr route 2ccc::/64 interface ethernet1 gateway 2aac::2
set vrouter trust-vr route 2aaa::/64 interface ethernet2 gateway 2bbb::1
set vrouter trust-vr route 3aaa::/64 interface ethernet3 gateway 3bbb::1
set vrouter trust-vr route 2bad::/64 interface ethernet4 gateway 2dad::1
set vrouter trust-vr route 3bad::/64 interface ethernet5 gateway 3dad::1
```

WebUI (Gateway Router)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.6) > IPv6

3. **Routes**

   Network > Routing > Routing Table > New (trust-vr)

4. **Users**

   Objects > Users > Local > New

   Objects > Users > Local > Edit

Objects > User > Local Groups > New

Objects > User > Local Groups > Edit

5. **RADIUS**

   Configuration > Auth > Auth Servers > New

   Configuration > Auth > Auth Servers > Edit

6. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit

7. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

8. **Policies**

   Policies > (From: Untrust, To: Trust) > New

   Policies > (From: Trust, To: Untrust) > New

CLI (Gateway Router)

1. **Interfaces**

```
set interface ethernet1/1 zone untrust
set interface ethernet1/1 manage
set interface ethernet1/1 ipv6 mode host
set interface ethernet1/1 ipv6 en
set interface ethernet1/1 ipv6 interface-id 2111111111111111
set interface ethernet1/1 ipv6 ip 2aac::2/64
set interface ethernet1/1 route
set interface ethernet1/2 zone trust
set interface ethernet1/2 manage
set interface ethernet1/2 ipv6 mode router
set interface ethernet1/2 ipv6 enable
set interface ethernet1/2 ipv6 interface-id 2222222222222222
set interface ethernet1/2 ipv6 ip 2ccc::1/64
set interface ethernet1/2 route
set interface ethernet1/2 ipv6 ra transmit
set interface ethernet1/2 ip 50.1.1.1/24
```

2. **Tunnel**

```
set interface tunnel.1 zone untrust
set interface tunnel.1 ipv6 mode host
set interface tunnel.1 ipv6 enable
```

3. **Routes**

```
set vrouter trust-vr route ::/0 interface ethernet1/1 gateway 2aac::1
set vrouter trust-vr route 2aaa::/64 interface tunnel.1
set vrouter trust-vr route 3aaa::/64 interface tunnel.1
set vrouter trust-vr route 2bad::/64 interface tunnel.1
set vrouter trust-vr route 3bad::/64 interface tunnel.1
set ipsec access-session enable
```

4. **Users**

```
set user n1 uid 1
set user n1 ike-id fqdn juniper.net share-limit 10
set user n1 type ike
set user n1 enable
set user-group group1 id 1
set user-group group1 user n1
```

5. **RADIUS**

```
set auth-server RAD1 id 1
set auth-server RAD1 server-name 50.1.1.5
set auth-server RAD1 account-type xauth
set auth-server RAD1 radius port 1812
set auth-server RAD1 radius secret juniper
```

6. **IKE**

```
set ike gateway universal dialup group1 aggressive outgoing-interface ethernet1/1
  local-address 2aac::2 preshare abc123 sec-level standard
set ike gateway universal xauth server RAD1 query-config
set ike gateway universal dpd interval 15
set ike gateway universal dpd retry 2
```

7. **VPN**

```
set vpn VPN1 gateway universal sec-level standard
set vpn VPN1 bind interface tunnel.1
```

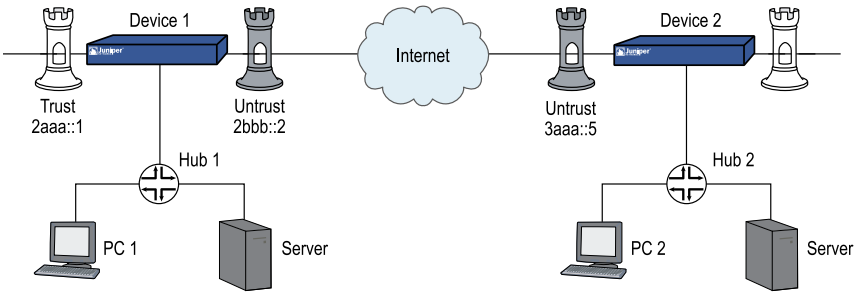
8. **Policies**

```
set policy id 1 from Untrust to Trust "Dial-up VPN ipv6" any-ipv6 any permit
set policy id 2 from trust to unTrust any-ipv6 "Dial-up VPN ipv6" any permit
set policy id 3 from untrust to Trust any-ipv6 any-ipv6 any permit
```

Dead Peer Detection

Figure 540 on page 2256 shows two devices configured for Dead Peer Detection (DPD).

Figure 540: Dead Peer Detection Example



NOTE: The WebUI section lists only the navigational paths to the device configuration pages. For specific values, see the CLI section that follows it.

On Device 1:

- Configure Trust and Untrust interfaces of Device 1 as an IPv6 router.
  - Configure the Trust interface to advertise prefix 2aaa::.
  - Configure the Untrust interface to advertise prefix 2bbb::.
- Configure an IKE gateway.
- Configure a VPN.
- Set the DPD interval time to 15 seconds and the retry setting to 10 retries.

On Device 2:

- Configure Untrust interface of as an IPv6 host so it can accept RAs.
- Configure the Trust interface as an IPv6 router, advertising prefix 3bbb::.
- Configure an IKE gateway.
- Configure a VPN.

NOTE: Optionally, you can set DPD Always Send by entering the **set ike gateway ton4 dpd always-send** command.

WebUI (Device 1)

1. **Interfaces**

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

2. **Tunnels**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.6) > IPv6

3. **Routes**

   Network > Routing > Routing Table > New (trust-vr)

4. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit

5. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

6. **Policy**

   Policies > (From: Untrust, To: Trust) > New

   Policies > (From: Trust, To: Untrust) > New

CLI (Device 1)

1. **Interfaces**

```
unset zone untrust block
set interface trust ipv6 mode router
set interface trust ipv6 ip 2aaa::/64
set interface trust ipv6 interface-id 0000000000000001
set interface trust ipv6 enable
set interface trust ipv6 ra transmit
set interface trust route
set interface trust manage
set interface untrust ipv6 mode router
set interface untrust ipv6 ip 2bbb::/64
set interface untrust ipv6 interface-id 0000000000000002
```

```
set interface untrust ipv6 enable
set interface untrust ipv6 ra transmit
set interface untrust route
set interface untrust manage
```

2. **Tunnel**

```
set interface tunnel.6 zone untrust
set interface tunnel.6 ipv6 mode host
set interface tunnel.6 ipv6 enable
set interface tunnel.6 ipv6 ip 3ddd::/64
```

3. **Routes**

```
set vrouter trust-vr route 3bbb::/64 interface tunnel.6
set vrouter trust-vr route ::/0 interface untrust gateway 2bbb::3
```

4. **IKE**

```
set ike gateway ton4 address 3aaa::5 outgoing-interface untrust local-address
  2bbb::2 preshare abc sec-level standard
set ike gateway ton4 dpd interval 15
set ike gateway ton4 dpd retry 10
```

5. **VPN**

```
set vpn vpn1 gateway ton4 no-replay sec-level standard
set vpn vpn1 bind interface tunnel.6
```

6. **Policies**

```
set policy from untrust to trust any-ipv6 any-ipv6 any permit
set policy from trust to untrust any-ipv6 any-ipv6 any permit
```

WebUI (Device 2)

1. **Interfaces**

   Network > Interfaces > Edit (for untrust)

   Network > Interfaces > Edit (for untrust) > IPv6

   Network > Interfaces > Edit (for trust)

   Network > Interfaces > Edit (for trust) > IPv6

2. **Tunnel**

   Network > Interfaces > New (Tunnel IF)

   Network > Interfaces > Edit (for tunnel.6) > IPv6

3. **Routes**

   Network > Routing > Routing Table > New (trust-vr)

4. **IKE**

   VPNs > AutoKey Advanced > Gateway > New

   VPNs > AutoKey Advanced > Gateway > Edit

5. **VPN**

   VPNs > AutoKey IKE > New

   VPNs > AutoKey IKE > Edit

6. **Policy**

   Policies > (From: Untrust, To: Trust) > New

   Policies > (From: Trust, To: Untrust) > New

CLI (Device 2)

1. **Interfaces**

```
unset zone untrust block
set interface untrust ipv6 mode host
set interface untrust ipv6 interface-id 0000000000000005
set interface untrust ipv6 enable
set interface untrust ipv6 ra accept
set interface untrust route
set interface untrust manage
set interface trust ipv6 mode router
set interface trust ipv6 ip 3bbb::/64
set interface trust ipv6 interface-id 0000000000000006
set interface trust ipv6 enable
set interface trust ipv6 ra transmit
set interface trust route
set interface trust manage
```

2. **Tunnel**

```
set interface tunnel.6 zone untrust
set interface tunnel.6 ipv6 mode host
set interface tunnel.6 ipv6 ip 3ddd::/64
set interface tunnel.6 ipv6 enable
```

3. **Routes**

```
set vrouter trust-vr route ::/0 interface untrust gateway 3aaa::4
set vrouter trust-vr route 2aaa::/64 interface tunnel.6
```

4. **IKE**

```
set ike gateway ton1 address 2bbb::2 outgoing-interface untrust local-address
   3aaa::5 preshare abc sec-level standard
set ike gateway ton1 dpd interval 15
set ike gateway ton1 dpd retry 5
```

5. **VPN**

```
set vpn vpn4 gateway ton1 no-replay sec-level standard
set vpn vpn4 bind interface tunnel.6
```

6. **Policies**

```
set policy from untrust to trust any-ipv6 any-ipv6 any permit
set policy from trust to untrust any-ipv6 any-ipv6 any permit
```

Part 15
Appendixes

Appendix A
Contexts for User-Defined Signatures

Contexts for User-Defined Signatures

The context defines the location in the packet where the Deep Inspection (DI) module searches for a signature matching the attack object pattern. When defining a stateful signature attack object, you can specify any of the contexts in the following lists. After you define an attack object, you must then put it in a user-defined attack object group for use in policies.

NOTE: A user-defined attack object group can contain only user-defined attack objects. You cannot mix predefined and user-defined attack objects in the same attack object group.

When the DI module examines traffic for TCP stream signatures, it does so without regard for contexts. TCP stream signatures look for patterns anywhere in any kind of TCP traffic regardless of the application protocol in use. Stream signatures are defined on NetScreen-5000 and 2000 series systems only. Stream256, however, looks for patterns in the first 256 bytes of data.

Table 144: Contexts for User-Defined Signatures

Protocol	Context	Description (Sets the Context As...)
AIM	aim-chat-room-desc	The description of a chat room in an America Online Instant Messenger (AIM) or ICQ (I Seek You) session.
	aim-chat-room-name	The name of a chat room in an AIM or ICQ session.
	aim-get-file	The name of a file that a user is transferring from a peer.
	aim-nick-name	The nickname of an AIM or ICQ user.
	aim-put-file	The name of a file that a user is transferring to a peer.
	aim-screen-name	The screen name of an AIM or ICQ user.
DNS	dns-cname	The CNAME (canonical name) in a Domain Name System (DNS) request or response, as defined in RFC 1035, *Domain Names—Implementation and Specification*).

Table 144: Contexts for User-Defined Signatures *(continued)*

Protocol	Context	Description (Sets the Context As...)
FTP	ftp-command	One of the FTP commands specified in RFC 959, *File Transfer Protocol (FTP)*.
	ftp-password	An FTP login password.
	ftp-pathname	A directory or filename in any FTP command.
	ftp-username	The name that a user enters when logging into an FTP server.
Gnutella	gnutella-http-get-filename	The name of a file that a Gnutella client intends to retrieve.
HTTP	http-authorization	The username and password decoded from an Authorization: Basic header in an HyperText Transfer Protocol (HTTP) request, as specified in RFC 1945, *HyperText Transfer Protocol—HTTP/1.0*.
	http-header-user-agent	The user-agent field in the header of an HTTP request. (When users visit a website, they provide information about their browsers in this field.)
	http-request	An HTTP request line.
	http-status	The status line in an HTTP reply. (The status line is a three-digit code that a Web server sends a client to communicate the state of a connection. For example, 401 means “Unauthorized” and 404 means “Not found”.)
	http-text-html	The text, or HyperText Markup Language (HTML) data, in an HTTP transaction.
	http-url	The uniform resource locator (URL) in an HTTP request as it appears in the data stream.
	http-url-parsed	A “normalized” text string decoded from a unicode string that comprises a URL used in HTTP.
	http-url-variable-parsed	A decoded common gateway interface (CGI) variable in the URL of an HTTP-GET request.
IMAP	imap-authenticate	an argument in an Internet Mail Access Protocol (IMAP) AUTHENTICATE command. The argument indicates the type of authentication mechanism that the IMAP client proposes to the server. Examples are KERBEROS_V4, GSSAPI (see RFC 1508, *Generic Security Service Application Program Interface*), and SKEY. For information about IMAP, see RFC 1730, *Internet Message Access Protocol - Version 4*, and RFC 1731, *IMAP4 Authentication Mechanisms*.
	imap-login	Either the username or plaintext password in an IMAP LOGIN command.
	imap-mailbox	The mailbox text string in an IMAP SELECT command.
	imap-user	The username in an IMAP LOGIN command.
MSN Messenger	msn-display-name	The display name of a user in a Microsoft Network (MSN) Instant Messaging session.
	msn-get-file	The name of a file that a client is downloading from a peer.
	msn-put-file	The name of a file that a client is sending to a peer.

Table 144: Contexts for User-Defined Signatures *(continued)*

Protocol	Context	Description (Sets the Context As...)
	msn-sign-in-name	The screen name (login name) of an MSN Instant Messaging user.
POP3	pop3-auth	The AUTH command in a Post Office Protocol, version 3 (POP3) session. For information about POP3, see RFC 1939, *Post Office Protocol—Version 3*.
	pop3-header-from	The text string in the "From:" header of an email in a POP3 transaction.
	pop3-header-line	The text string in any header line of an email in a POP3 transaction.
	pop3-header-subject	The text string in the "Subject:" header of an email in a POP3 transaction.
	pop3-header-to	The text string in the "To:" header of an email in a POP3 transaction.
	pop3-mime-content-filename	The content filename of a Multipurpose Internet Mail Extensions (MIME) attachment in a POP3 session.
	pop3-user	The username in a POP3 session.
SMB	smb-account-name	The name of a Server Message Blocks (SMB) account in a SESSION_SETUP_ANDX request in an SMB session.
	smb-connect-path	The connect path in the TREE_CONNECT_ANDX request in an SMB session.
	smb-connect-service	The name of the connect service in the TREE_CONNECT_ANDX request in an SMB session.
	smb-copy-filename	The name of a file in a COPY request in an SMB session.
	smb-delete-filename	The name of a file in a DELETE request in an SMB session.
	smb-open-filename	The name of a file in the NT_CREATE_ANDX and OPEN_ANDX requests in an SMB session.
SMTP	smtp-from	The text string in a "MAIL FROM" command line in a Simple Mail Transfer Protocol (SMTP) session, as described in RFC 2821, *Simple Mail Transfer Protocol*.
	smtp-header-from	The text string in the "From:" header in an SMTP session.
	smtp-header-line	The text string in any header line in an SMTP session.
	smtp-header-subject	The text string in the "Subject:" header in an SMTP session.
	smtp-header-to	The text string in the "To:" header in an SMTP session.
	smtp-mime-content-filename	The content filename of a Multipurpose Internet Mail Extensions (MIME) attachment in an SMTP session.
	smtp-rcpt	The text string in a "RCPT TO" command line in an SMTP session.
–	stream256	The first 256 bytes of a reassembled, normalized TCP data stream.
Yahoo! Messenger	ymsg-alias	The alternate identifying name associated with the main username of a Yahoo! Instant Messaging user.

Table 144: Contexts for User-Defined Signatures *(continued)*

Protocol	Context	Description (Sets the Context As...)
	ymsg-chatroom-message	The text in messages exchanged in a Yahoo! Instant Messaging chatroom.
	ymsg-chatroom-name	The name of a Yahoo! Instant Messaging chatroom.
	ymsg-nickname	The nickname of a Yahoo! Instant Messaging user.

Appendix B

Wireless Information

This appendix lists information that might affect your deployment of a wireless LAN (WLAN). It contains the following sections:

802.11a Channel Numbers

This section applies only to security devices with two radios.

The regulatory domains are as follows:

- Telecom Engineering Center (TELEC)—Japan
- Federal Communications Commission (FCC)—US
- European Telecommunications Standards Institute (ETSI)—Europe
- WORLD—All countries

Table 145 on page 2267 lists the countries and channel numbers for 802.11a.

Table 145: 802.11a Channel Numbers

Country	Country Code	Regulatory Domain	Channel
Argentina	AR	WORLD	56, 60, 64, 149, 153, 157, 161
Australia	AU	WORLD	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
Austria	AT	ETSI	36, 40, 44, 48
Belgium	BE	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Bulgaria	BG	ETSI	36, 40, 44, 48, 52, 56, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Canada	CA	FCC	36, 40, 44, 48, 52, 56, 60, 64
Chile	CL	WORLD	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165

Table 145: 802.11a Channel Numbers *(continued)*

Country	Country Code	Regulatory Domain	Channel
Colombia	CO	WORLD	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
Cyprus	CY	ETSI	36, 40, 44, 48, 52, 56, 60, 64
Czech Republic	CZ	ETSI	36, 40, 44, 48, 52, 56, 60, 64
Denmark	DK	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Estonia	EE	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Finland	FI	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
France	FR	ETSI	36, 40, 44, 48, 52, 56, 60, 64
Germany	DE	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Greece	GR	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Hong Kong	HK	WORLD	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
Hungary	HU	ETSI	36, 40, 44, 48, 52, 56, 60, 64
Iceland	IS	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
India	IN	WORLD	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
Ireland	IE	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Italy	IT	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Japan	JP	TELEC	36, 40, 44, 48, 52, 56, 60, 64
Latvia	LV	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Liechtenstein	LI	ETSI	36, 40, 44, 48, 52, 56, 60, 64
Lithuania	LT	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Luxembourg	LU	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Malta	MT	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140

Table 145: 802.11a Channel Numbers *(continued)*

Country	Country Code	Regulatory Domain	Channel
Mexico	MX	WORLD	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
Monaco	MC	ETSI	36, 40, 44, 48, 52, 56, 60, 64
Netherlands	NL	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
New Zealand	NZ	WORLD	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
Norway	NO	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Panama	PA	WORLD	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
Peru	PE	WORLD	149, 153, 157, 161, 165
Philippines	PH	WORLD	149, 153, 157, 161, 165
Poland	PL	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Portugal	PT	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Saudi Arabia	SA	WORLD	Not available
Slovak Republic	SK	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Slovenia	SI	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
South Africa	ZA	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Spain	ES	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Sweden	SE	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
Switzerland	CH	ETSI	36, 40, 44, 48, 52, 56, 60, 64
Thailand	TH	WORLD	Not available
Turkey	TR	WORLD	36, 40, 44, 48, 52, 56, 60, 64
Ukraine	UA	WORLD	Not available
United Kingdom	GB	ETSI	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
United States	US	FCC	36, 40, 44, 48, 52, 56, 60, 64

Table 145: 802.11a Channel Numbers *(continued)*

Country	Country Code	Regulatory Domain	Channel
Venezuela	VE	WORLD	149, 153, 157, 161

802.11b and 802.11g Channels

All supported countries listed in Table 145 on page 2267 and Table 146 on page 2270, except for the following list, support channels 1 through 13 for 802.11b and 802.11g. The listed countries support channels 1 through 11 for 802.11b and 802.11g.

- Canada
- Colombia
- Dominican Republic
- Guatemala
- Mexico
- Panama
- Puerto Rico
- United States
- Uzbekistan

Turbo-Mode Channel Numbers

Table 146 on page 2270 lists the channels for 802.11a Turbo and 802.11g Turbo modes. The 802.11a Turbo column applies only to wireless security devices with two radios. The 802.11g Turbo column applies to security devices with one or two radios.

Table 146: Channels for 802.11a and 802.11g Turbo Modes

Country	Country Code	802.11a Turbo	802.11g Turbo
Albania	AL	Not available	6
Algeria	DZ	Not available	6
Argentina	AR	Not available	Not available
Armenia	AM	Not available	6
Australia	AU	42, 50, 58, 152, 160	6
Austria	AT	Not available	6
Azerbaijan	AZ	Not available	6
Bahrain	BH	Not available	6

Table 146: Channels for 802.11a and 802.11g Turbo Modes *(continued)*

Country	Country Code	802.11a Turbo	802.11g Turbo
Belarus	BY	Not available	6
Belgium	BE	Not available	6
Belize	BZ	Not available	6
Bolivia	BO	Not available	6
Brazil	BR	Not available	6
Brunei Darussalam	BN	Not available	6
Bulgaria	BG	Not available	6
Canada	CA	42, 50, 58	6
Chile	CL	42, 50, 58, 152, 160	Not available
Colombia	CO	Not available	6
Costa Rica	CR	Not available	6
Croatia	HR	Not available	6
Cyprus	CY	Not available	6
Czech Republic	CZ	Not available	6
Denmark	DK	Not available	6
Dominican Republic	DO	Not available	6
Ecuador	EC	Not available	6
Egypt	EG	Not available	6
El Salvador	SV	Not available	6
Estonia	EE	Not available	6
Finland	FI	Not available	6
France	FR	Not available	6
France_Res	F2	Not available	6
Georgia	GE	Not available	6
Germany	DE	Not available	6
Greece	GR	Not available	6
Guatemala	GT	Not available	6

Table 146: Channels for 802.11a and 802.11g Turbo Modes *(continued)*

Country	Country Code	802.11a Turbo	802.11g Turbo
Honduras	HN	Not available	6
Hong Kong	HK	42, 50, 58, 152, 160	6
Hungary	HU	Not available	6
Iceland	IS	Not available	6
India	IN	Not available	6
Indonesia	ID	Not available	6
Iran	IR	Not available	6
Ireland	IE	Not available	6
Israel	IL	Not available	6
Italy	IT	Not available	6
Japan	JP	Not available	Not available
Japan1	J1	Not available	Not available
Japan2	J2	Not available	Not available
Japan3	J3	Not available	Not available
Japan4	J4	Not available	Not available
Japan5	J5	Not available	Not available
Jordan	JO	Not available	6
Kazakhstan	KZ	Not available	6
North Korea	KP	Not available	6
Korea Republic2	K2	Not available	6
Kuwait	KW	Not available	6
Latvia	LV	Not available	6
Lebanon	LB	Not available	6
Liechtenstein	LI	Not available	6
Lithuania	LT	Not available	6
Luxembourg	LU	Not available	6
Macao	MO	Not available	6

Table 146: Channels for 802.11a and 802.11g Turbo Modes *(continued)*

Country	Country Code	802.11a Turbo	802.11g Turbo
Macedonia	MK	Not available	6
Malaysia	MY	Not available	6
Malta	MT	Not available	6
Mexico	MX	42, 50, 58, 152, 160	6
Monaco	MC	Not available	6
Morocco	MA	Not available	6
Netherlands	NL	Not available	6
New Zealand	NZ	Not available	6
Norway	NO	Not available	6
Oman	OM	Not available	6
Pakistan	PK	Not available	6
Panama	PA	42, 50, 58, 152, 160	6
Peru	PE	Not available	6
Philippines	PH	Not available	6
Poland	PL	Not available	6
Portugal	PT	Not available	6
Puerto Rico	PR	Not available	6
Qatar	QA	Not available	6
Romania	RO	Not available	6
Russia	RU	Not available	6
Saudi Arabia	SA	Not available	6
Slovak Republic	SK	Not available	6
Slovenia	SI	Not available	6
South Africa	ZA	Not available	6
Spain	ES	Not available	6
Sweden	SE	Not available	6
Switzerland	CH	Not available	6

Table 146: Channels for 802.11a and 802.11g Turbo Modes *(continued)*

Country	Country Code	802.11a Turbo	802.11g Turbo
Syria	SY	Not available	6
Thailand	TH	Not available	6
Trinidad & Tobago	TT	Not available	6
Tunisia	TN	Not available	6
Turkey	TR	Not available	6
Ukraine	UA	Not available	6
United Arab Emirates	AE	Not available	6
United Kingdom	GB	Not available	6
United States	US	42, 50, 58	6
Uruguay	UY	Not available	6
Uzbekistan	UZ	Not available	6
Venezuela	VE	Not available	6
Vietnam	VN	Not available	6
Yemen	YE	Not available	6
Zimbabwe	ZW	Not available	6

Appendix C
Switching

Switching

You can use the security device as a switch to pass IPv6 traffic along a path. To do this, you must place the device in transparent mode. For more information about transparent mode, see *"Fundamentals" on page 15*.

Table 147 on page 2275 lists the ScreenOS commands that can allow IPv6 traffic and other non-IP traffic to pass through a security device operating in transparent mode.

Table 147: Transparent Mode Commands to Bypass Non-IP Traffic

Command	Description
set interface vlan1 bypass-non-ip	Allows all Layer 2 non-IP traffic to pass through the security device.
unset interface vlan1 bypass-non-ip	Blocks all non-IP and non-ARP unicast traffic (default behavior).
unset interface vlan1 bypass-non-ip-all	Overwrites the unset interface vlan1 bypass-non-ip command when both commands appear in the configuration file. If you previously entered the unset interface vlan1 bypass-non-ip-all command, and you now want the device to revert to its default behavior of blocking only the non-IP and non-ARP unicast traffic, you first enter the set interface vlan1 bypass-non-ip command to allow all non-IP and non-ARP traffic, including multicast, unicast, and broadcast traffic to pass through the device. Then you enter the unset interface vlan1 bypass-non-ip command to block only the non-IP, non-ARP unicast traffic.
set interface vlan1 bypass-others-ipsec	Allows a device to pass IPsec traffic without attempting to terminate it, use the command. The device then allows the IPsec traffic to pass through to other VPN termination points.

Part 16
Index

■ Index on page 2279

Index

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z