Concepts & Examples
ScreenOS Reference Guide

Volume 7:
Routing

Release 6.2.0, Rev. 01

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net

Part Number: 530-023769-01, Revision 01

Copyright Notice

All specifications are subject to change without notice. Juniper Networks assumes no responsibility for any inaccuracies in this document or for any obligation to update information in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

FCC Statement

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. The equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

The following information is for FCC compliance of Class B devices: The equipment described in this manual generates and may radiate radio-frequency energy. If it is not installed in accordance with Juniper Networks' installation instructions, it may cause interference with radio and television reception. This equipment has been tested and found to comply with the limits for a Class B digital device in accordance with the specifications in part 15 of the FCC rules. These specifications are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Consult the dealer or an experienced radio/TV technician for help.
- Connect the equipment to an outlet on a circuit different from that to which the receiver is connected.

Caution: Changes or modifications to this product could void the user's warranty and authority to operate this device.

Disclaimer

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR JUNIPER NETWORKS REPRESENTATIVE FOR A COPY.

Table of Contents

Chapter 6 Policy-Based Routing 137

Chapter 7 Multicast Routing 155

Chapter 8 Internet Group Management Protocol 163

About This Volume

Volume 7: Routing contains the following chapters:

- Chapter 1, “Static Routing,” explains route tables and how to configure static routes for destination-based routing, Source Interface-Based Routing (SIBR), or source-based routing.
- Chapter 2, “Routing,” explains how to configure virtual routers on security devices and how to redistribute routing table entries between protocols or between virtual routers.
- Chapter 3, “Open Shortest Path First,” describes how to configure the OSPF dynamic routing protocol on security devices.
- Chapter 4, “Routing Information Protocol,” explains how to configure Routing Information Protocol l (RIP).
- Chapter 5, “Border Gateway Protocol,” explains how to configure Border Gateway Protocol (BGP).
- Chapter 6, “Policy-Based Routing,” 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.
- Chapter 7, “Multicast Routing,” explains multicast routing basics, including how to configure static multicast routes.
- Chapter 8, “Internet Group Management Protocol,” explains how to configure Internet Group Management Protocol (IGMP).
- Chapter 9, “Protocol Independent Multicast,” explains how to configure Protocol Independent Multicast-Sparse Mode (PIM-SM) and Protocol Independent Multicast-Source Specific Multicast (PIM-SSM).
- Chapter 10, “ICMP Router Discovery Protocol,” explains how to set up an Internet Control Message Protocol (ICMP) exchange between a host and a router.

Document Conventions

This document uses the conventions described in the following sections:

- "Web User Interface Conventions" on page x
- "Command Line Interface Conventions" on page x
- "Naming Conventions and Character Types" on page xi
- "Illustration Conventions" on page xii

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 left 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 left 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":

  ```
  set interface { ethernet1 | ethernet2 | ethernet3 } manage
  ```

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.

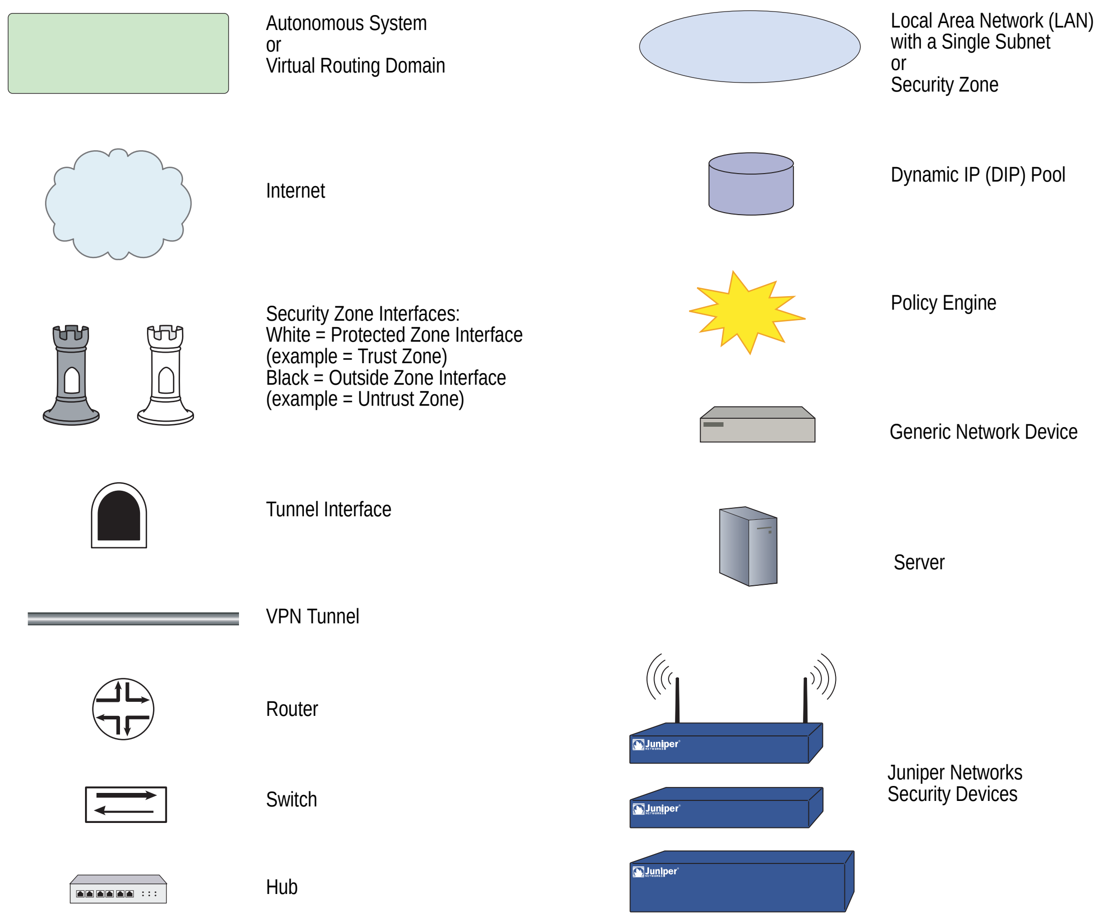
- ASCII characters from 32 (0x20 in hexadecimals) to 255 (0xff), except double quotes ("), which have special significance as an indicator of the beginning or end of a name string that includes spaces.

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 1 shows the basic set of images used in illustrations throughout this volume.

Figure 1: 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/
- 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.

Chapter 1

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 2 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 2: Static Routing Example

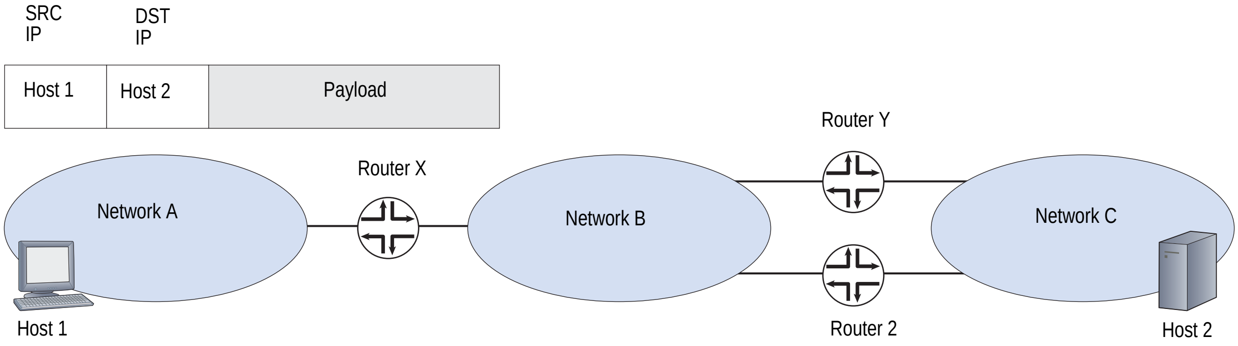


Table 1 summarizes the routing table of each router.

Table 1: 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 1, 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 11.

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” on page 5
- “Setting a Static Route for a Tunnel Interface” on page 9

Setting Static Routes

In Figure 3 on page 7, 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

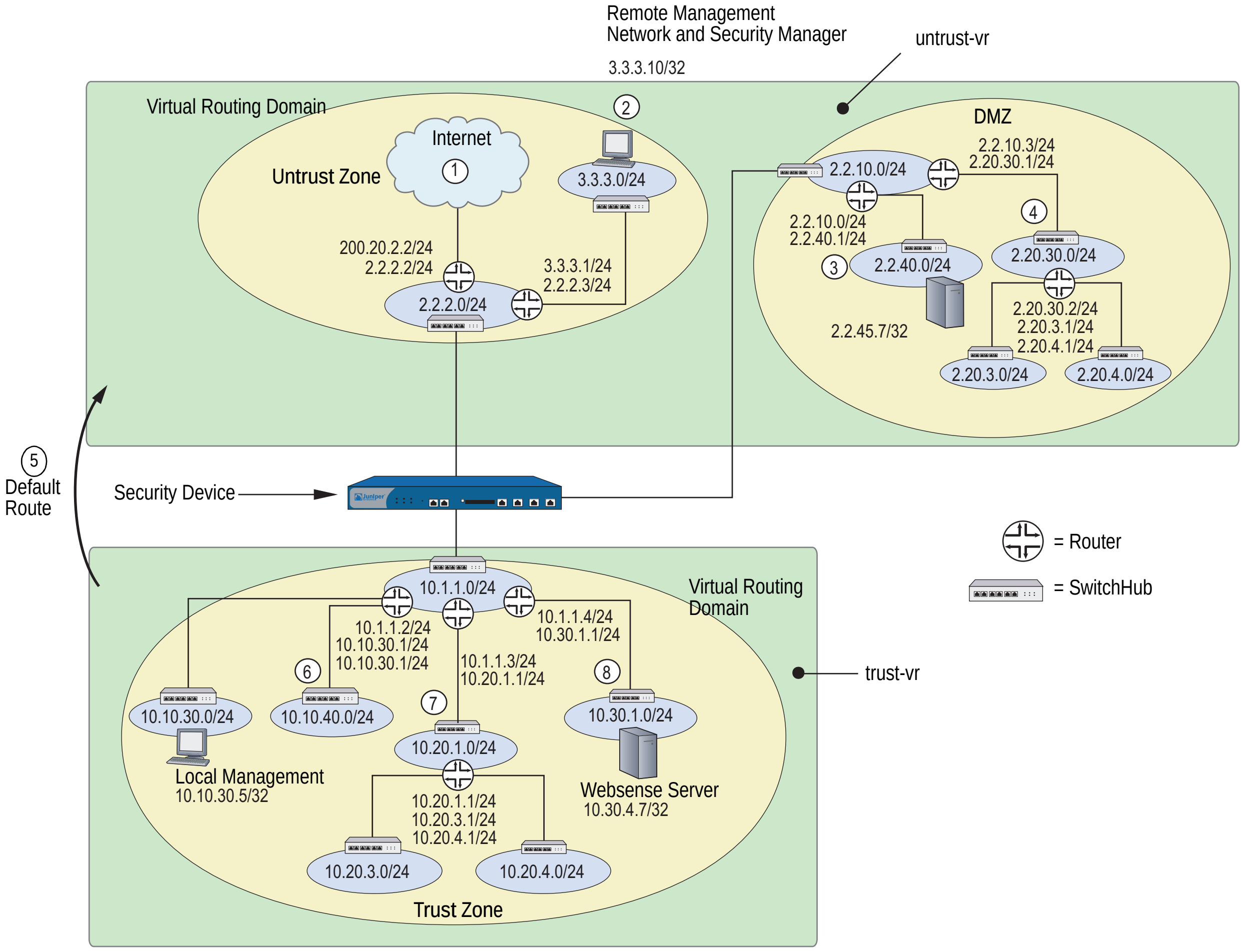
5. untrust-vr for all addresses not found in the trust-vr routing table (default route for the VR)

6. 10.10.0.0/16 subnet in the Trust zone

7. 10.20.0.0/16 subnet in the Trust zone

8. 10.30.1.0/24 subnet in the Trust zone

Figure 3: 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
```

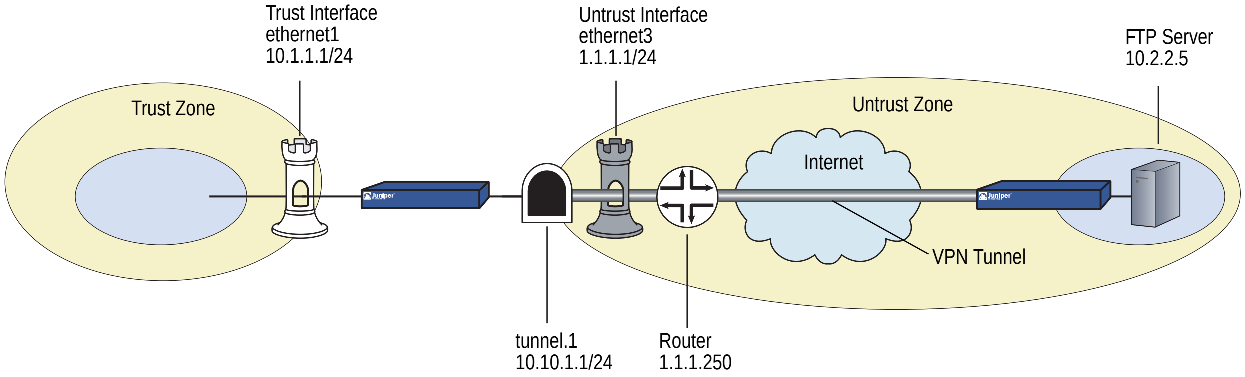
2. **trust-vr**

```
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 4, 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 4: 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 21.) 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 38 for more information.

Chapter 2
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 2-25.

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 19.
- 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 21.

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 33 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 29. 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 5.

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 this 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.

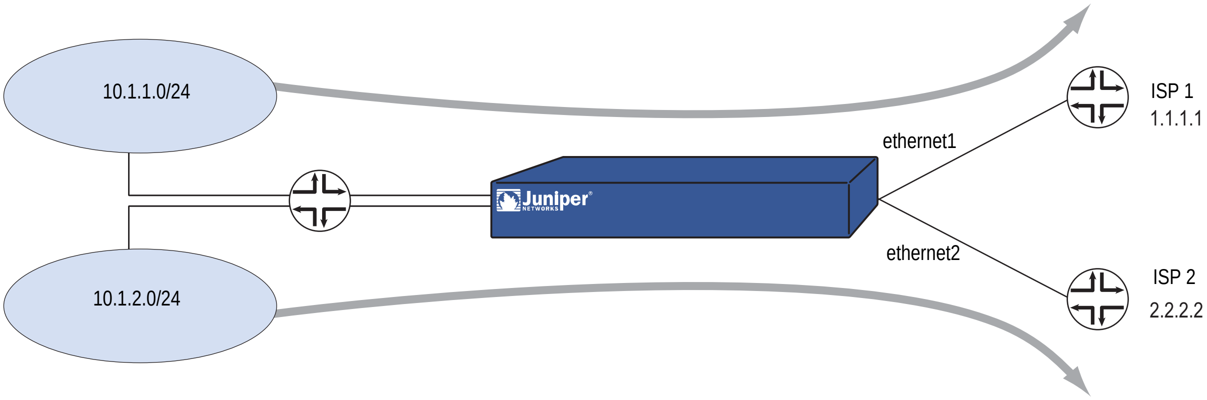
2. Enable source-based routing for the VR. The security device uses the source IP of the packet for route lookup in the source-based routing table. If no route is found for the source IP address, the destination IP address is used for the routing table lookup.

In Figure 5, 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 5: 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:

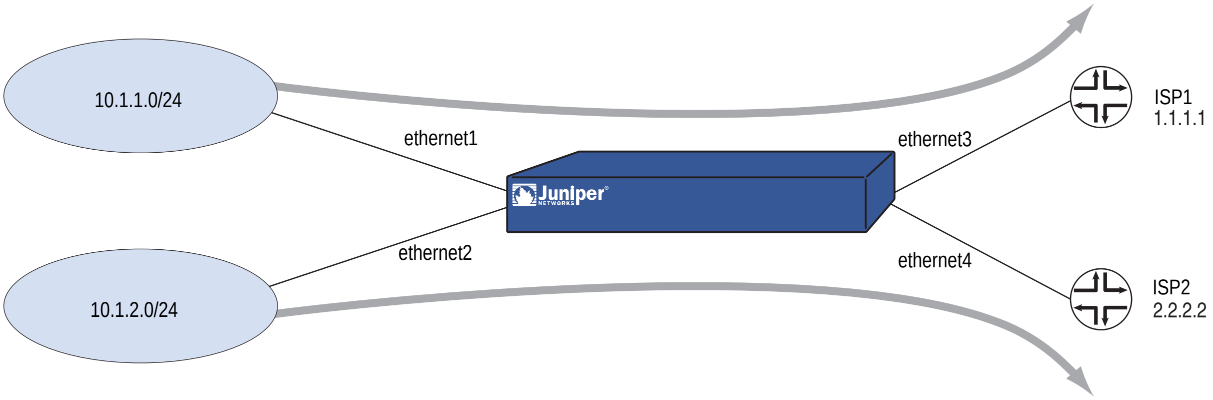
   - The name of the VR in which SIBR applies.
   - The source interface on which the security device performs a lookup in the SIBR table. (The interface appears as an entry in the routing table.)
   - The IP address and netmask prefix for the route.
   - The name of the outgoing interface on which the 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 6, 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 6: 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 32).
- Maximum number of entries allowed in the routing table.
- Preference value for routes, based on protocol (see “Setting a Route Preference” on page 33).
- 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 19.)
- 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 *Volume 11: High Availability* 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 volume.

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 5.

- 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 44.

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 2-35.)

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:

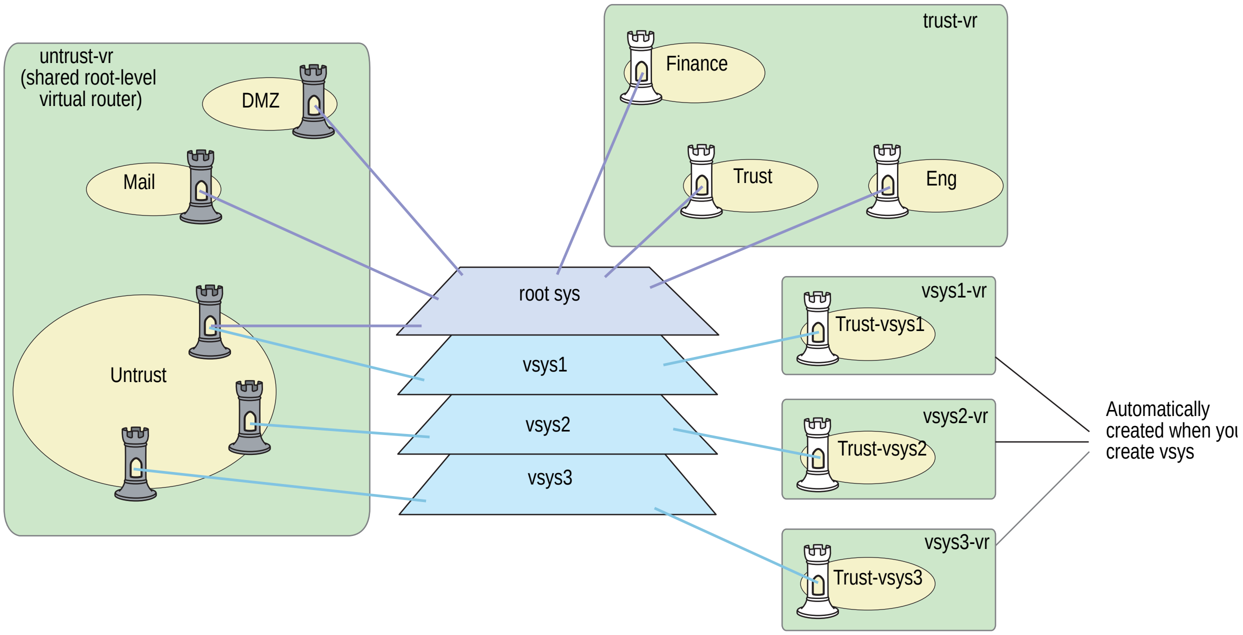
- Any root-level VRs that have been defined as sharable. The untrust-vr is, by default, a shared VR that is accessible by any vsys. You can configure other root-level VRs to be sharable.

- 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 *Volume 10: Virtual Systems*. In Figure 7, each of the three vsys has two VRs associated with it: a vsys-level VR named *vsysname*-vr and the untrust-vr.

Figure 7: 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 33
- “Configuring Equal Cost Multipath Routing” on page 38
- “Route Redistribution” on page 40
- “Exporting and Importing Routes Between Virtual Routers” on page 44

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 2 lists the default preference values for routes of each protocol.

Table 2: 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

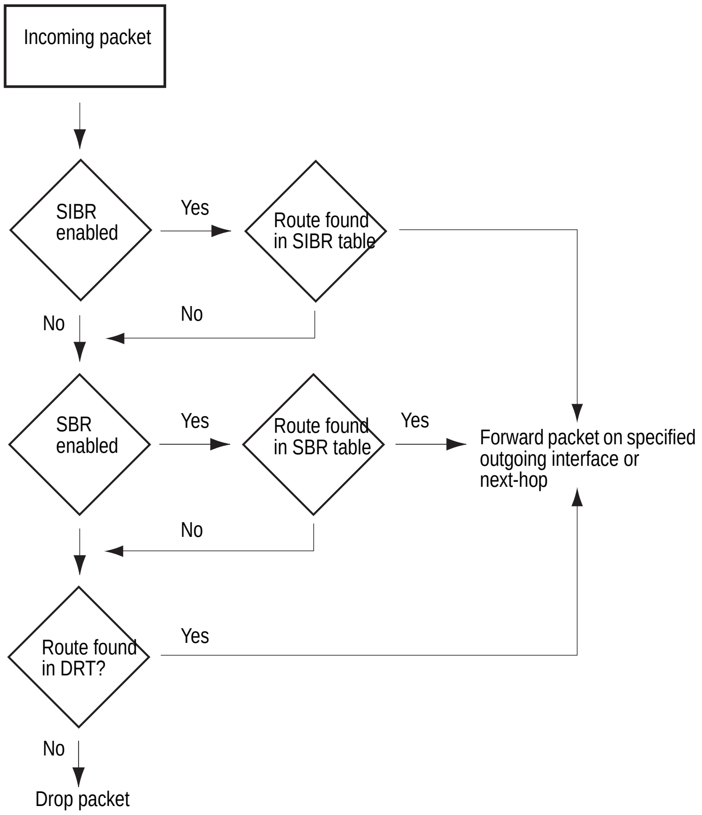
When routes are learned dynamically, the neighboring router from which the route originates provides the metric. The default metric for connected routes is always 0. The default metric for static routes is 1.

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 8 shows the default route lookup sequence.

Figure 8: 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:

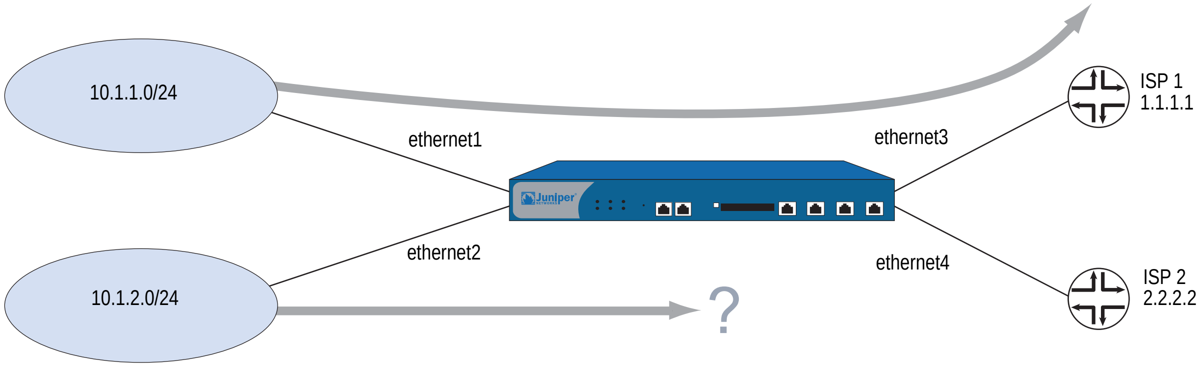
- A source-based routing entry for the subnetwork 10.1.1.0/24, with ethernet3 as the forwarding interface, and the router at 1.1.1.1 as the next-hop
- A default route, with the untrust-vr as the next-hop.

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 9 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 9: 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 2-75.

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. Table 3 lists route map match conditions and gives a description of each.

Table 3: Route Map Match Conditions

Match Condition	Description
BGP AS Path	Matches a specified AS path access list. See "Route Filtering" on page 41.
BGP Community	Matches a specified community list. See "Route Filtering" on page 41.
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 41.
Metric	Matches a specified route metric value.
Next-hop	Matches a specified access list. See "Route Filtering" on page 41.
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 4 lists route map attributes and descriptions of each.

Table 4: 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.
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” 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 122 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 130 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 159.

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 122.) 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 3
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 61.

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 70.

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 5 lists LSA types, where each type is flooded, and the contents of each type of LSA.

Table 5: 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 15.

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.

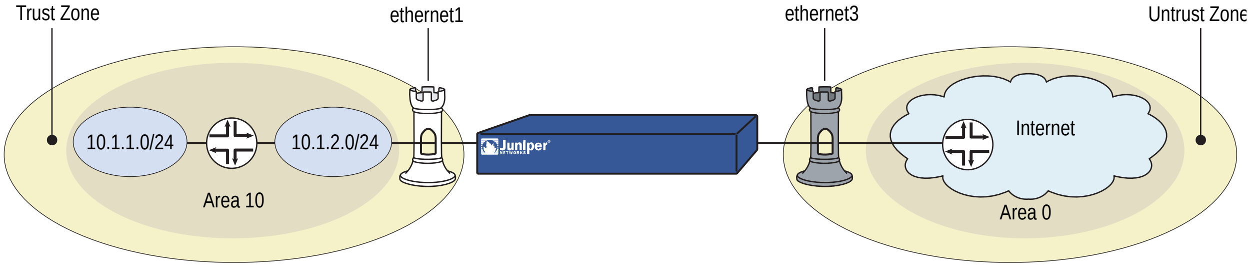
3. Assign one or more interfaces to each OSPF area. You must explicitly add interfaces to an OSPF area, including the backbone area.

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 10.

Figure 10: 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 60).
- Interface parameters, such as authentication, that are set on a per-interface basis for the OSPF protocol (see “Setting OSPF Interface Parameters” on page 64).
- Security-related OSPF parameters that are set at either the VR level or on a per-interface basis (see “Security Configuration” on page 66).

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 15.)

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 48 for more information about these types of areas.

Table 6 lists area parameters, with descriptions of each parameter, and gives the default value for each.

Table 6: 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 53), 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 53).

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 15.

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 15.

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 7 lists global OSPF parameters and their default values.

Table 7: 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.
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 38.	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 8 lists optional parameters for virtual links.

Table 8: Optional Parameters for Virtual Links

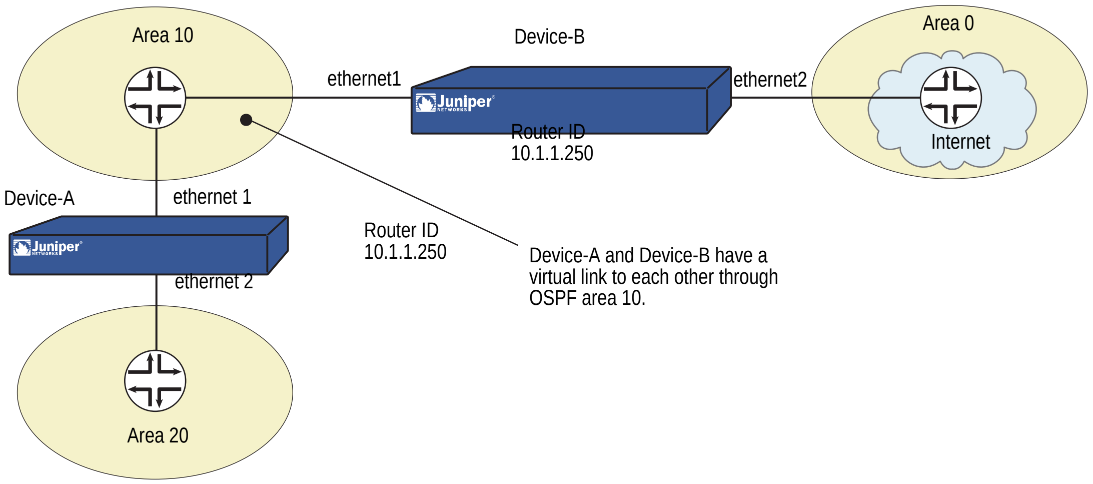
Virtual Link Parameter	Description	Default Value
Authentication	Specifies either clear text password or MD5 authentication.	No authentication
Dead interval	Specifies the number of seconds that elapses with no response from an OSPF neighbor before the neighbor is determined to be not running.	40 seconds
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 15 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 11 shows the example network setup for a virtual link.

Figure 11: 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 9 lists optional OSPF interface parameters and their default values.

Table 9: 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 70.	Ethernet interfaces are treated as broadcast interfaces. Tunnel interfaces bound to OSPF areas are point-to-point by default.
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.

OSPF Interface Parameter	Description	Default Value
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 69.	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 69). You can, however, configure the interface for reduced LSA flooding.

For an example of binding multiple tunnels to a tunnel interface, "Binding Automatic Route and NHTB Table Entries" on page 5-294. The following sections include examples for:

- Setting the link-type (see "Setting the OSPF Link-Type" on page 70)
- Setting the route-deny feature (see "Disabling the Route-Deny Restriction" on page 71)
- Configuring a network with a point-to-multipoint tunnel interface (see "Creating a Point-to-Multipoint Network" on page 71)

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 12 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.

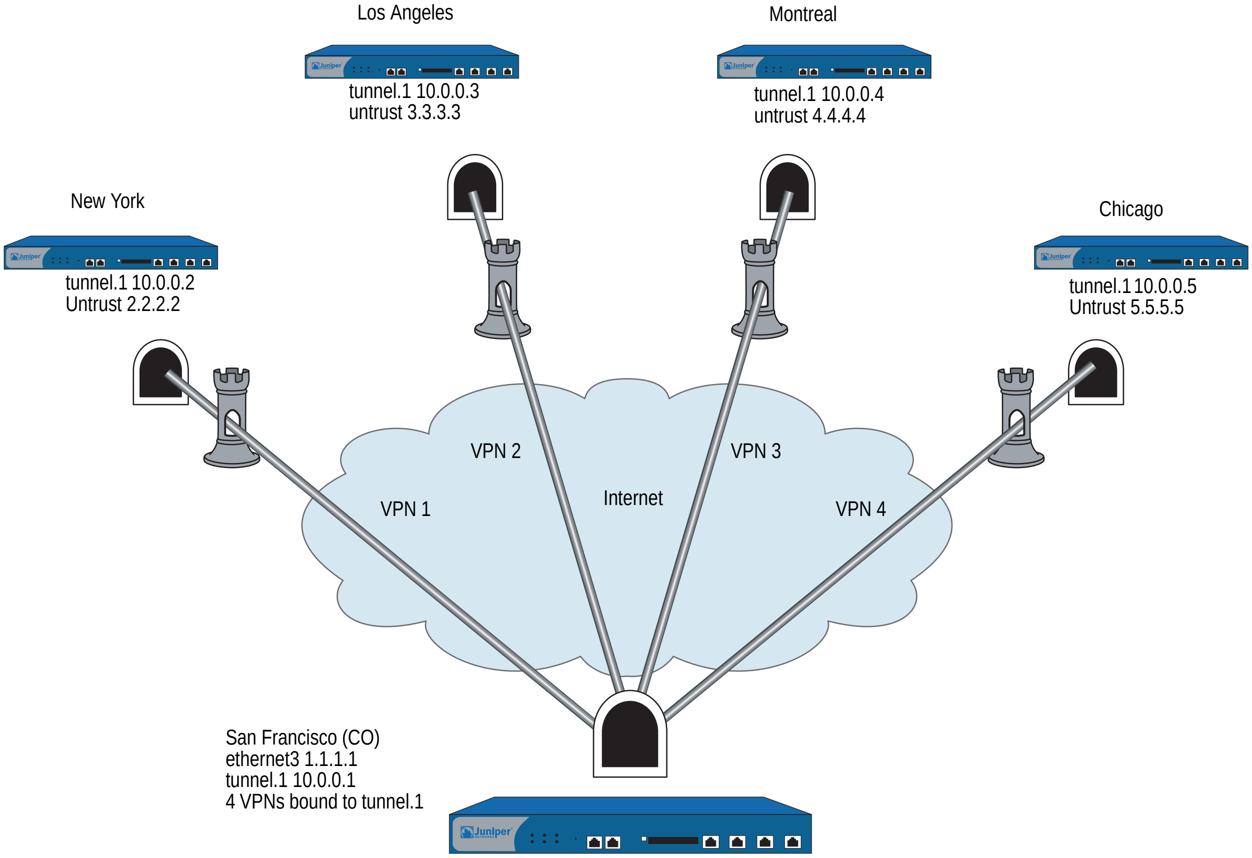
The following are the configuration requirements particular to the remote security devices:

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 12 shows the described network scenario.

Figure 12: Point-to-MultiPoint Network Example



In Figure 12, 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 12 on page 72, 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.

Chapter 4

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 95.

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 15.

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 84)
- Interface parameters, such as neighbor authentication, that are set on a per-interface basis for RIP (see “Configuring RIP Interface Parameters” on page 86)
- Security-related RIP parameters, that are set at either the VR level or on a per-interface basis (see “Security Configuration” on page 87)

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 15.

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 78), 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 15.

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 84).

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 10 lists the RIP global parameters and their default values.

Table 10: 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 88.	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 89.	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

RIP Global Parameter	Description	Default Value(s)
Maximum alternate routes	Specifies the maximum number of RIP routes for the same prefix that can be added into the RIP route database. See "Setting Alternate Routes" on page 94.	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 93.	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 91.	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 11 lists the RIP interface parameters and their default values.

Table 11: 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 87.	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 91.	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 93.	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 95.	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 40.

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

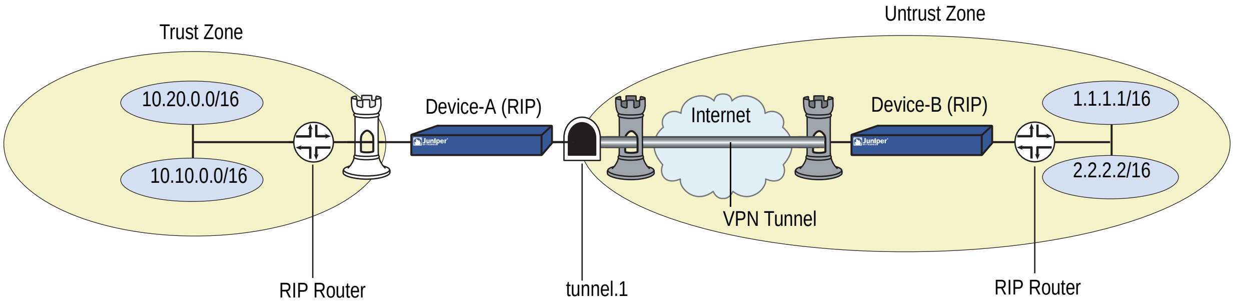
```
set vrouter trust-vr protocol rip threshold-update 4
save
```

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 13 shows the described network scenario.

Figure 13: 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 12.

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 95.

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 38) 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 page 98.

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 12 lists suggestions for troubleshooting performance issues influenced by timer settings.

Table 12: 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 14 on page 99.

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:

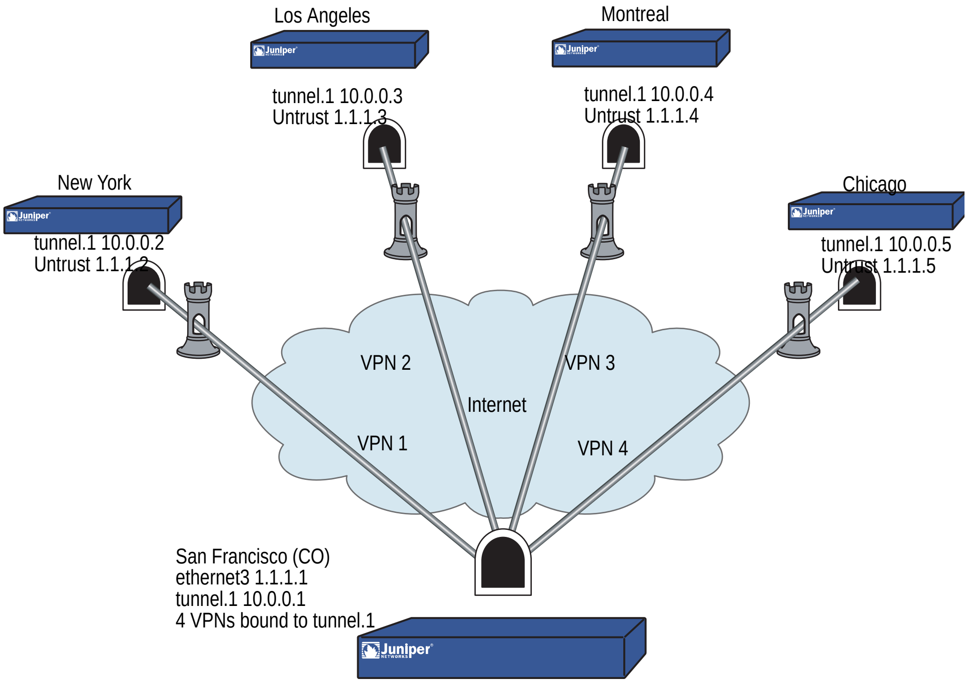
1. Configure the VR to run an instance of RIP and enable RIP and then configure tunnel.1 interface.

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 14: Point-to-MultiPoint with Tunnel Interface Network Example



In the network diagram shown in Figure 14, 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 page 99, 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 5

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 17.

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 2.

- 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 118.

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 15.

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 15 (IPv4) and Figure 16 (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 15: IPv4 BGP Configuration Example

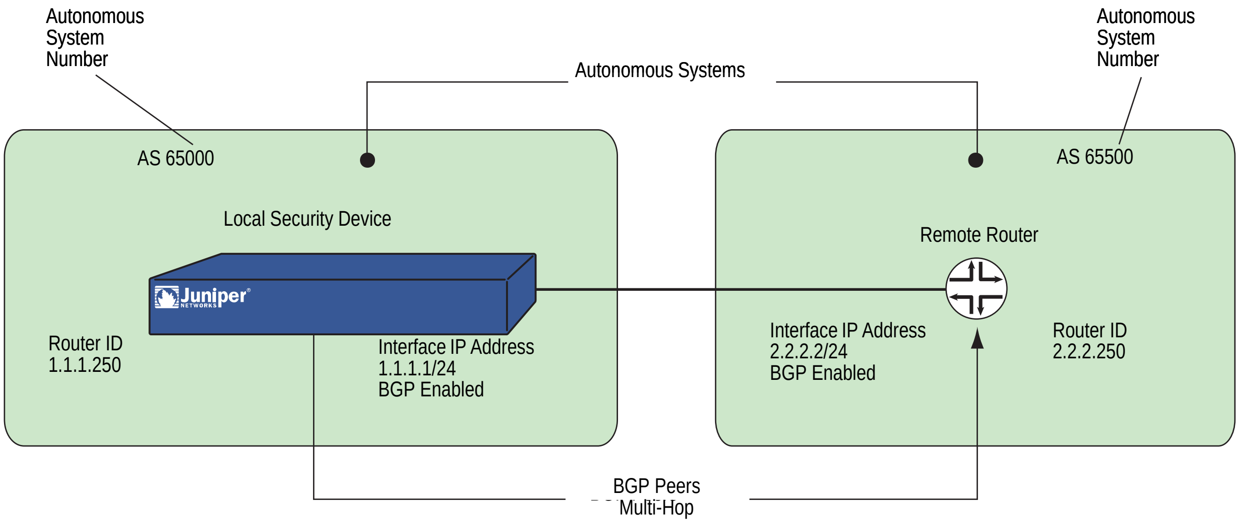
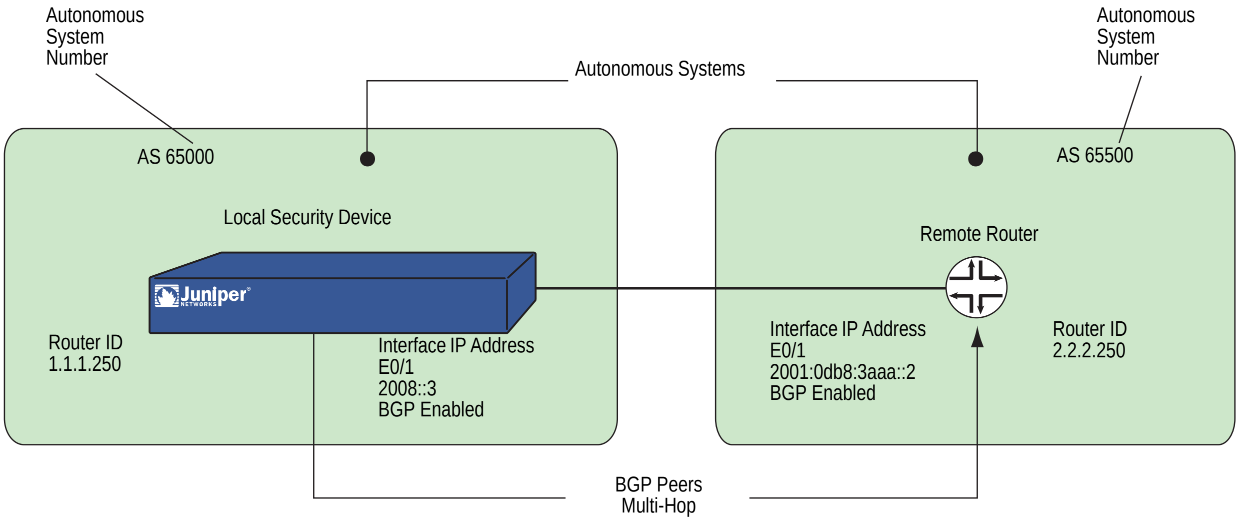


Figure 16: IPv6 BGP Configuration Example



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 15.)

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 13 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 13: BGP Peer and Peer Group Parameters and Default Values (page 1 of 2)

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 118.	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
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

Table 13: BGP Peer and Peer Group Parameters and Default Values (page 2 of 2)

BGP Parameter	Peer	Peer Group	Description	Default Value
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 129). 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 15.

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 14 describes BGP parameters and their default values.

Table 14: 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 131.	—
Always compare MED	Compare MED values in routes.	Disabled
AS confederation	Create confederations. See “Configuring a Confederation” on page 129.	—
AS path access list	Create an AS path access list to permit or deny routes. See “Configuring an AS-Path Access List” on page 122.	—
Community list	Create community lists. See “BGP Communities” on page 130.	—
Equal cost multipath (ECMP)	Equal cost multiple routes can be added to provide load-balancing capabilities. See “Configuring Equal Cost Multipath Routing” on page 38.	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 123.	—
Route redistribution	Import routes into BGP from other routing protocols. See “Redistributing Routes into BGP” on page 121.	—
Reflector	Configure the local BGP instance as a route reflector to clients. See “Configuring Route Reflection” on page 127.	Disabled
Reject default route	Ignore default route advertisements from BGP peers. See “Rejecting Default Routes” on page 120.	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 15.

In the following example, you redistribute a route that originated from a RIP routing domain into the current 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
```

Configuring an AS-Path Access List

The AS-path attribute contains a list of the 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 pattern 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

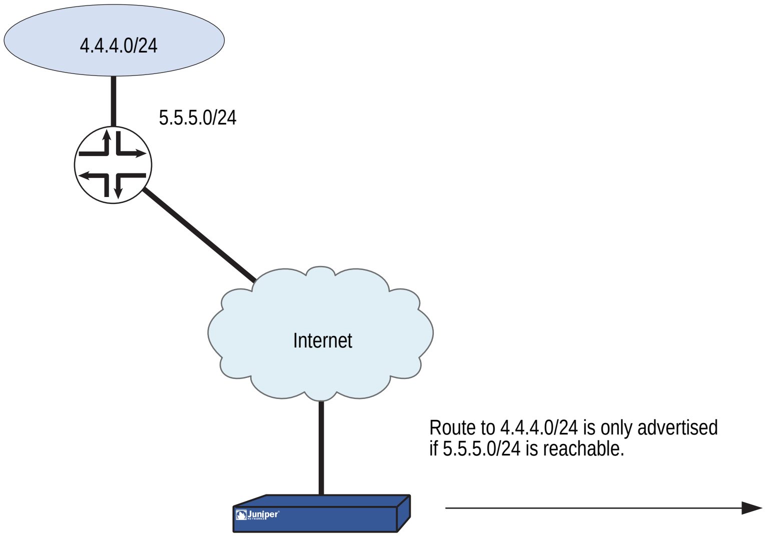
To allow BGP to advertise network routes, you need to redistribute the routes from the source protocol into the advertising protocol (BGP) in the same virtual router (VR). You can also add static routes directly into BGP. If the network prefix is reachable from the VR, BGP advertises this route to peers 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 40). 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 17.

Figure 17: Conditional BGP Route Advertisement Example


4.4.4.0/24
5.5.5.0/24
Internet
Route to 4.4.4.0/24 is only advertised if 5.5.5.0/24 is reachable.
Juniper

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 107), 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.

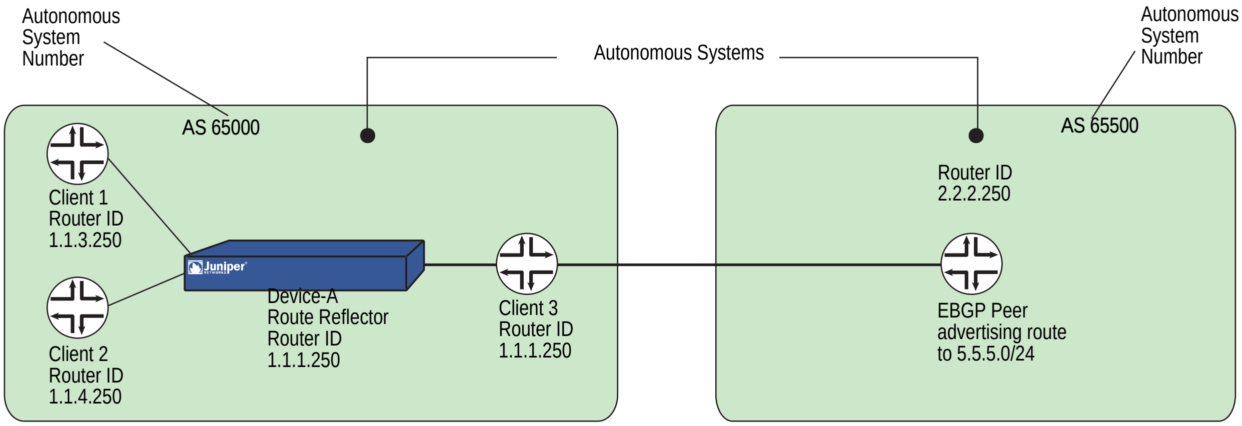
The local virtual router (VR) of the security device can act as a route reflector and can be assigned a cluster ID. If you specify a cluster ID, the BGP routing instance appends the cluster ID to the Cluster-List attribute of a route. The cluster ID helps prevent routing loops as the local BGP routing 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 18.

Figure 18: 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.

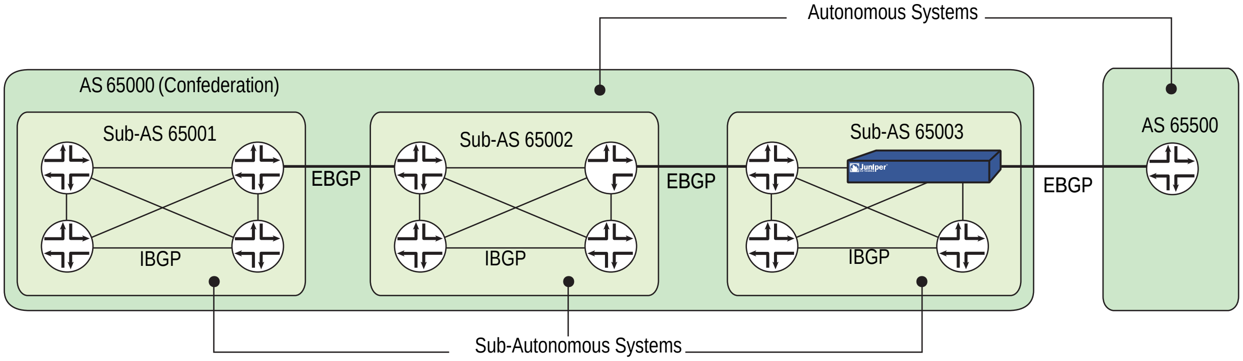
CLI

```
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 127), *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 19 illustrates BGP confederations.

Figure 19: BGP Confederations



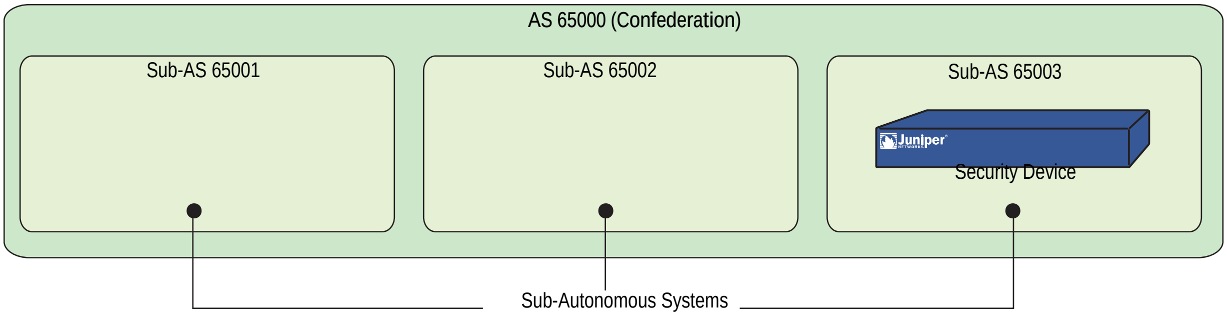
For each router in a confederation, you need to specify the following:

- 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 106) 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 20 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 20: 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.

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 6

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:

- "Advanced PBR Example" on page 148
- "Advanced PBR with High Availability and Scalability" on page 152

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.

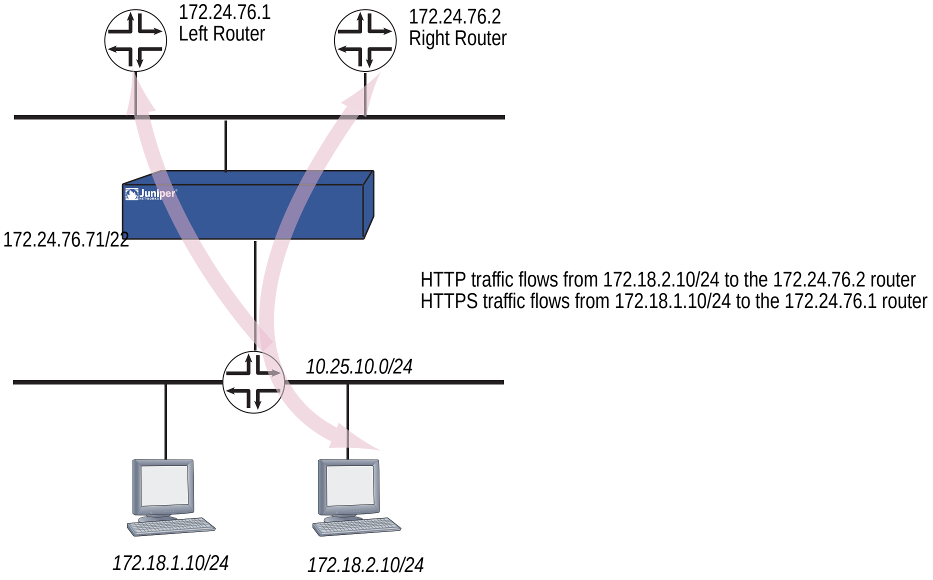
NOTE: For more information about route lookup, see *Volume 2: Fundamentals*.

Configuring PBR

Figure 21 shows one way PBR differentiates service-traffic paths by sending HTTP traffic along one path and HTTPS traffic along another. Figure 21 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 21: Routing HTTP and HTTPS Traffic with Policy Based Routing


172.24.76.1
Left Router
172.24.76.2
Right Router
Juniper
172.24.76.71/22
HTTP traffic flows from 172.18.2.10/24 to the 172.24.76.2 router
HTTPS traffic flows from 172.18.1.10/24 to the 172.24.76.1 router
10.25.10.0/24
172.18.1.10/24
172.18.2.10/24

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 page 141, 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:

Creating Access List 10

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:

Creating Access List 10

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 page 141, 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 page 141 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
```

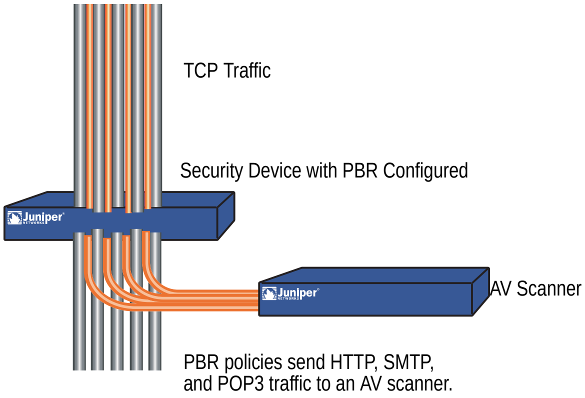
Advanced PBR Example

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 22 shows a security device running PBR to segregate AV traffic from all other traffic (right).

Figure 22: Selective Routing by Traffic Type


TCP Traffic
Security Device with PBR Configured
Juniper
Juniper
AV Scanner
PBR policies send HTTP, SMTP, and POP3 traffic to an AV scanner.

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 *Volume 4: Attack Detection and Defense Mechanisms*.

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 15, you set up four 10/100 Ethernet interfaces.

Table 15: 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 *Volume 11: High Availability*.

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 7
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 163.)
- 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 189.)

  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 171.)

The following sections introduce basic concepts used in multicast routing.

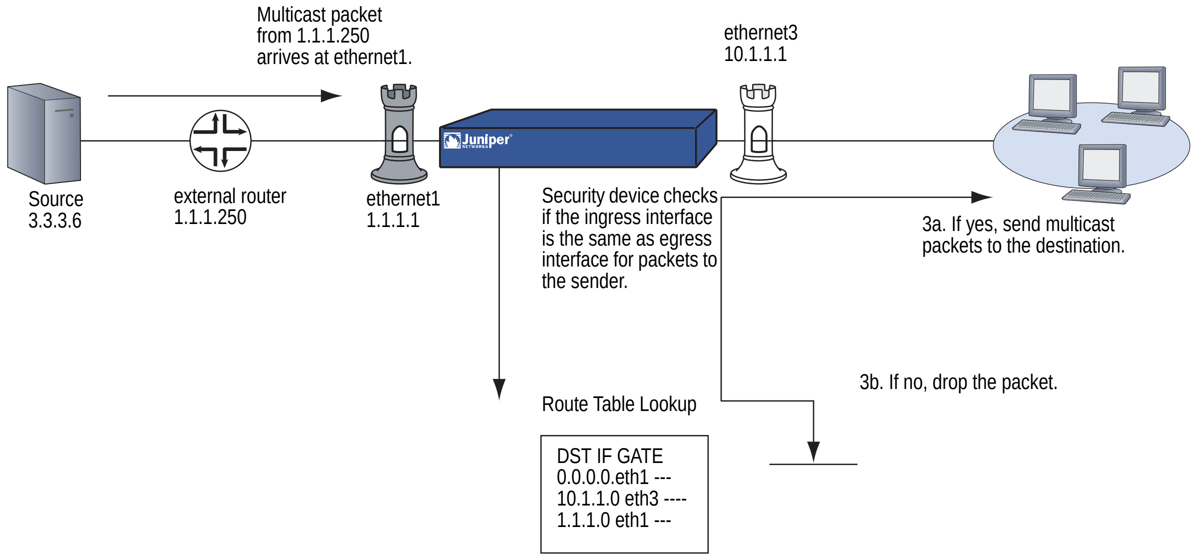
Multicast Addresses

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 23 shows the security device and the multicast packet processing flow.

Figure 23: 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 1.) 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 171.) 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 42.)

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.

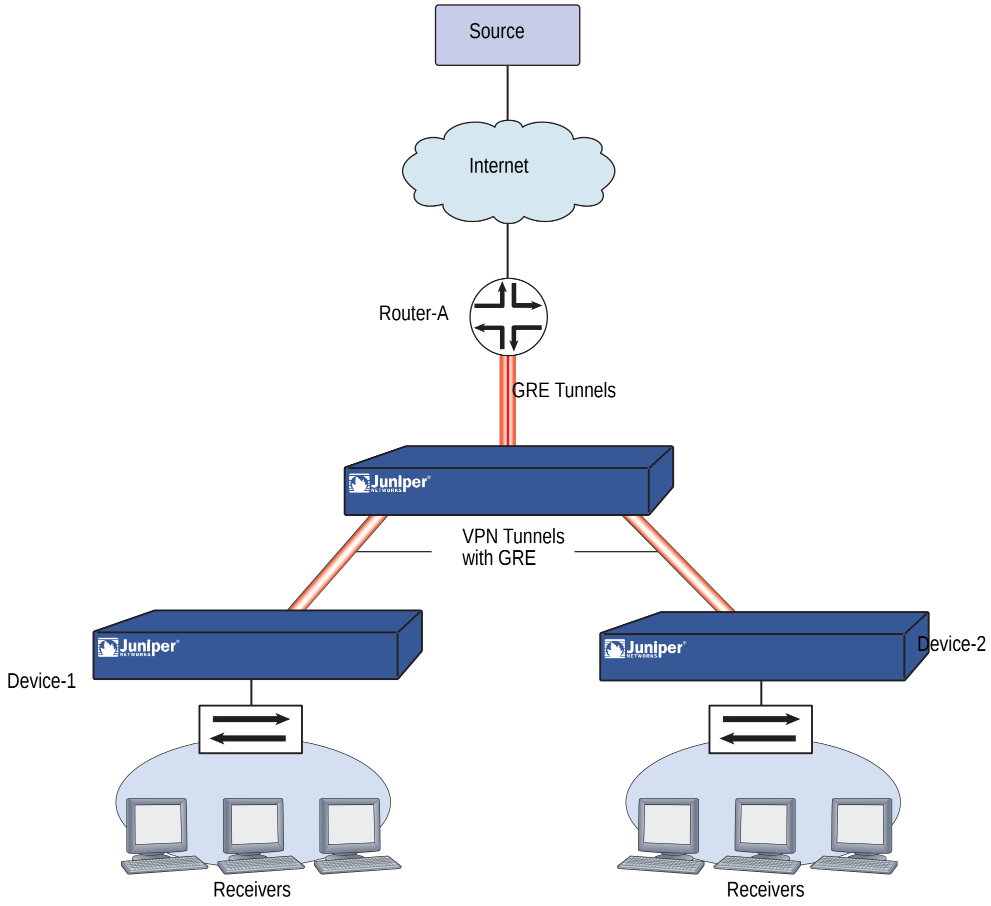
NOTE: You can enable GRE on a tunnel interface that is bound to a loopback interface as long as the loopback interface is on the same zone as the outgoing interface. For information about loopback interfaces, see "Loopback Interfaces" on page 2-58.

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 24, 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 24: 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 *Volume 5: Virtual Private Networks*.)

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 163. For information about PIM, see “Protocol Independent Multicast” on page 189.)

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 *Volume 2: Fundamentals*.)

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 175. For an example of how to configure a multicast policy for PIM messages, see "Defining a Multicast Group Policy for PIM-SM" on page 199.

Chapter 8

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 196.)

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 16 lists the IGMP messages that hosts send and the destination of the messages.

Table 16: 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 17 describes the messages that a querier sends and destinations.

Table 17: 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
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 42.)

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
```

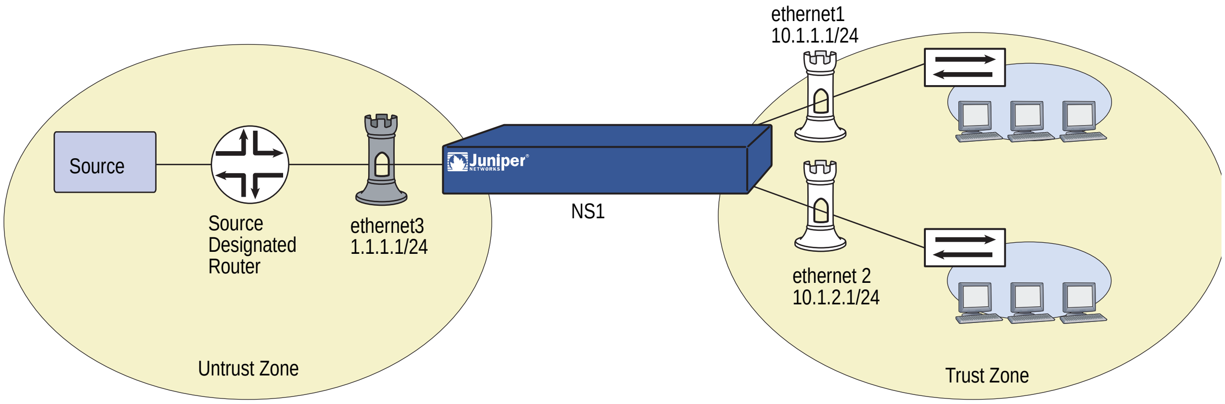
Configuring IGMP

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 25, 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 25: 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 189.)

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 18 lists the IGMP querier interface parameters and their defaults.

Table 18: 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 *Volume 10: Virtual Systems*.)

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.

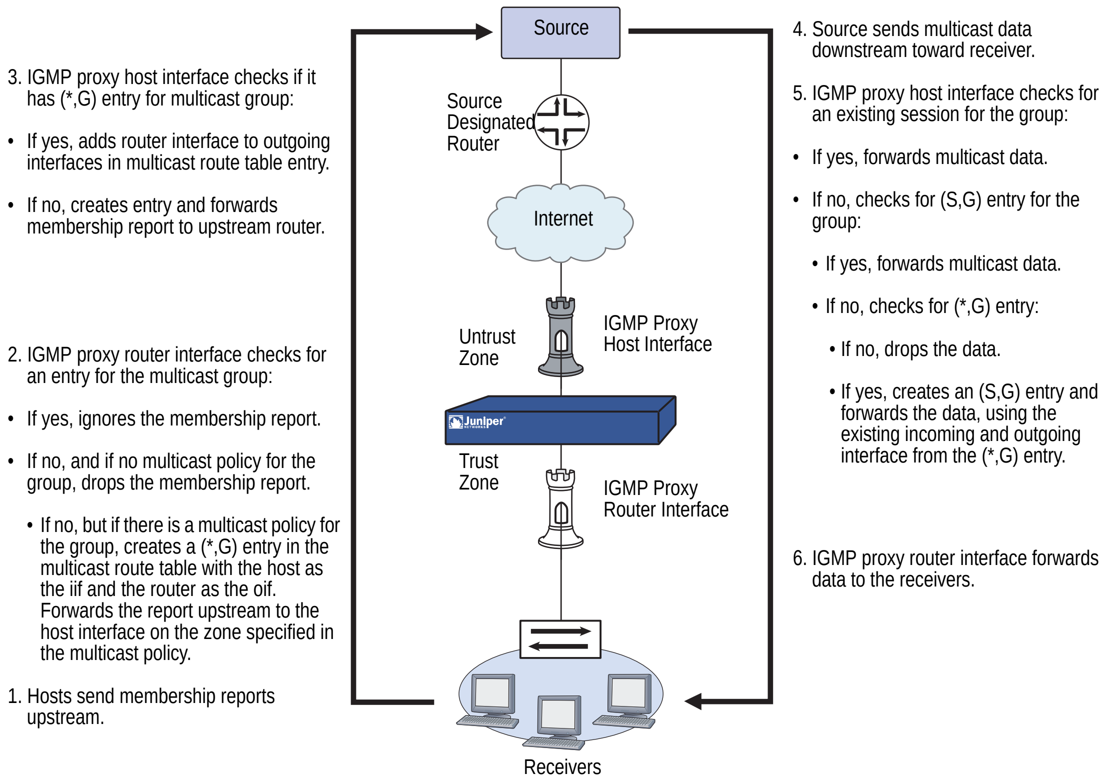
  - If there is no (S, G) entry, the interface checks if there is a (*, G) entry for the group.

  - 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 26 shows an example of an IGMP proxy host configuration.

Figure 26: IGMP Proxy Host Configuration


Source
4. Source sends multicast data downstream toward receiver.
3. IGMP proxy host interface checks if it has (*,G) entry for multicast group:
• If yes, adds router interface to outgoing interfaces in multicast route table entry.
• If no, creates entry and forwards membership report to upstream router.
Source Designated Router
5. IGMP proxy host interface checks for an existing session for the group:
• If yes, forwards multicast data.
• If no, checks for (S,G) entry for the group:
• If yes, forwards multicast data.
• If no, checks for (*,G) entry:
• If no, drops the data.
• If yes, creates an (S,G) entry and forwards the data, using the existing incoming and outgoing interface from the (*,G) entry.
Internet
Untrust Zone
IGMP Proxy Host Interface
2. IGMP proxy router interface checks for an entry for the multicast group:
• If yes, ignores the membership report.
• If no, and if no multicast policy for the group, drops the membership report.
• If no, but if there is a multicast policy for the group, creates a (*,G) entry in the multicast route table with the host as the iif and the router as the oif. Forwards the report upstream to the host interface on the zone specified in the multicast policy.
Juniper
Trust Zone
IGMP Proxy Router Interface
6. IGMP proxy router interface forwards data to the receivers.
1. Hosts send membership reports upstream.
Receivers

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

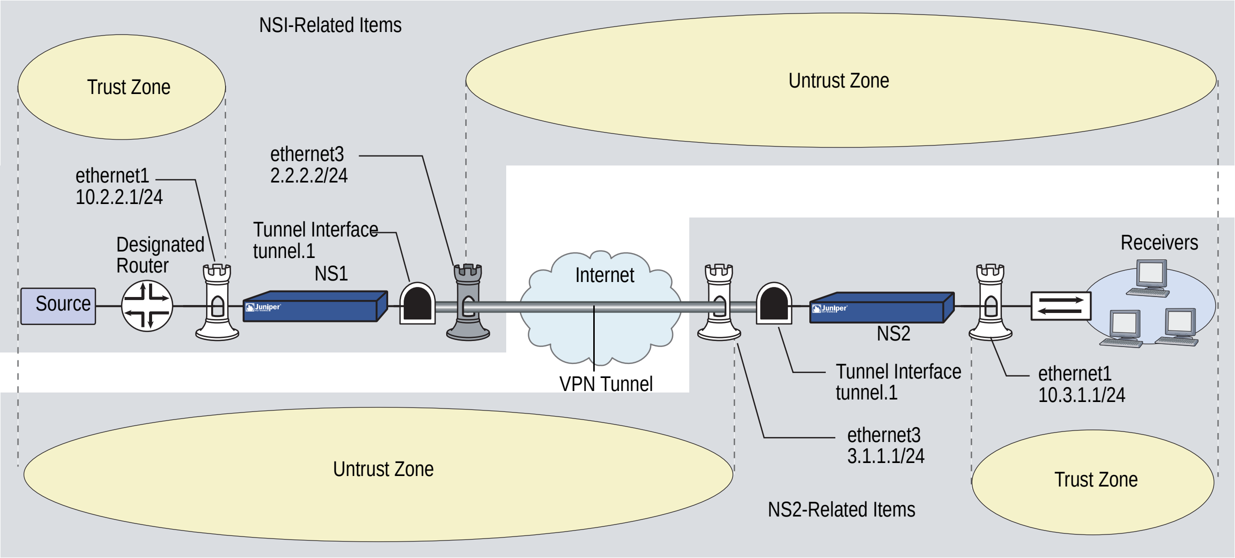
```
set multicast-group-policy from trust mgroup 224.2.202.99/32 to untrust
    igmp-message bi-directional
save
```

Creating an IGMP Proxy Configuration

As shown in Figure 27, 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 27: 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

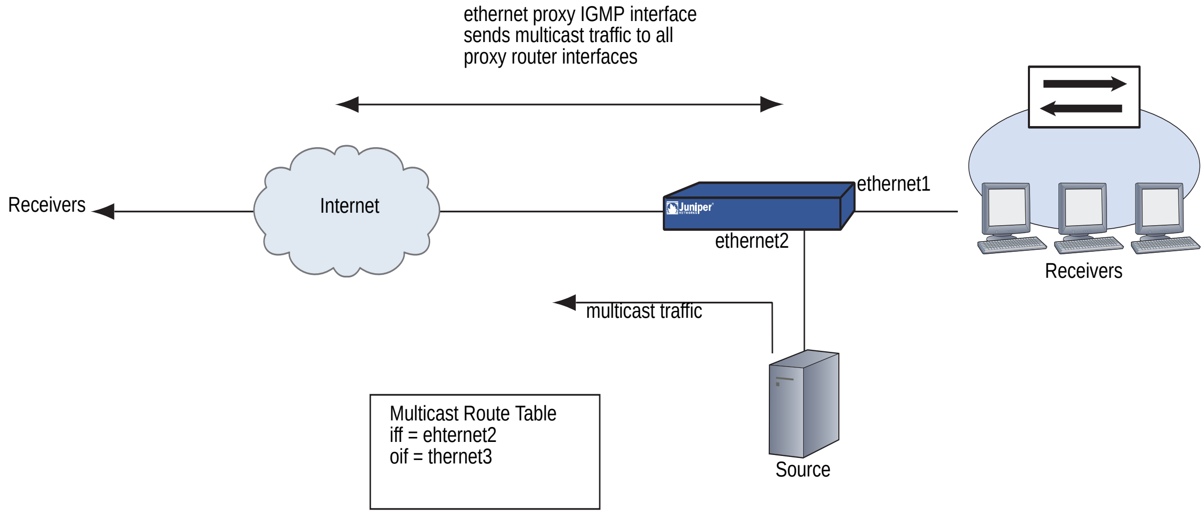
In IGMP proxy, the multicast traffic usually travels downstream from the host interface to the router interface. In certain situations, the source can be in the same network as the router interface. When a source connected to an interface that is on the same network as the IGMP router proxy interface 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 28 shows an example of IGMP sender proxy.

Figure 28: IGMP Sender Proxy



In Figure 29, 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.

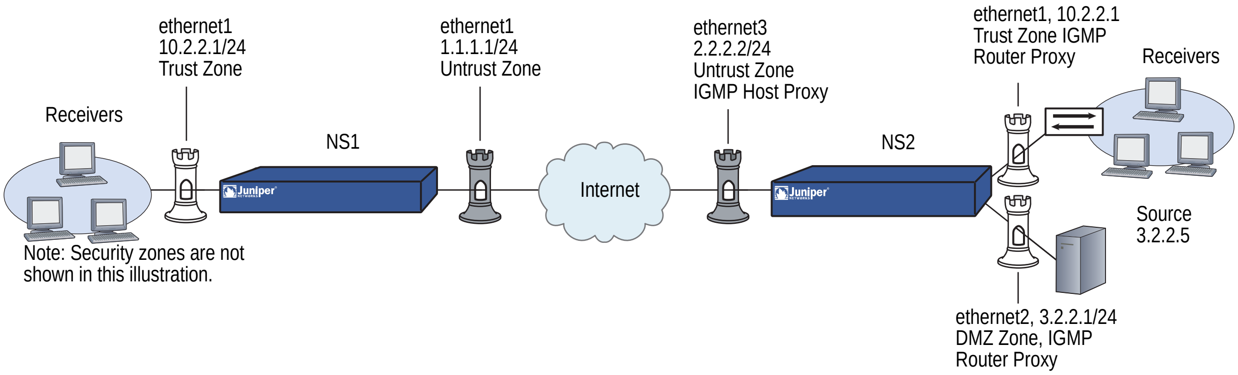
5. Set up the default route.

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 29: 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
```

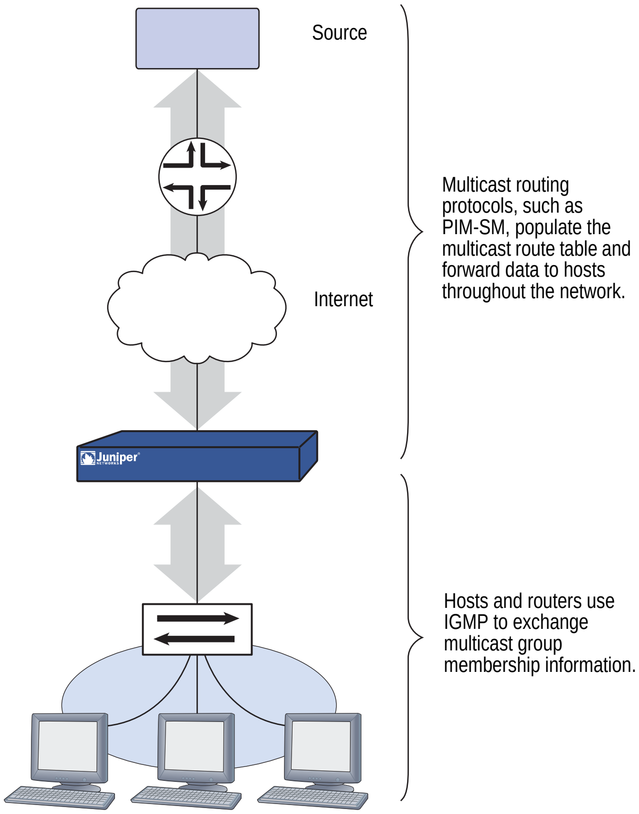
Chapter 9
Protocol Independent Multicast

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 163.)

Figure 30: IGMP


Source
Internet
Juniper
Multicast routing protocols, such as PIM-SM, populate the multicast route table and forward data to hosts throughout the network.
Hosts and routers use IGMP to exchange multicast group membership information.

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 156.)

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 192.)

- 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.” For information about PIM-SSM, see “PIM-SSM” on page 196.

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 192.) 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.

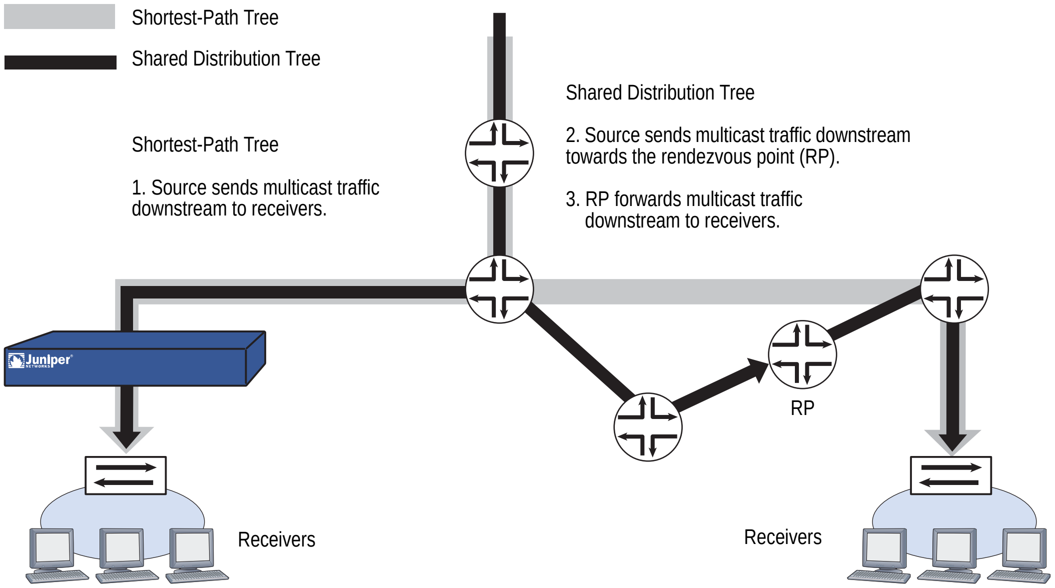
Multicast Distribution Trees

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 31: PIM


Shortest-Path Tree
Shared Distribution Tree
Shared Distribution Tree
2. Source sends multicast traffic downstream towards the rendezvous point (RP).
3. RP forwards multicast traffic downstream to receivers.
Shortest-Path Tree
1. Source sends multicast traffic downstream to receivers.
Juniper
RP
Receivers
Receivers

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*.

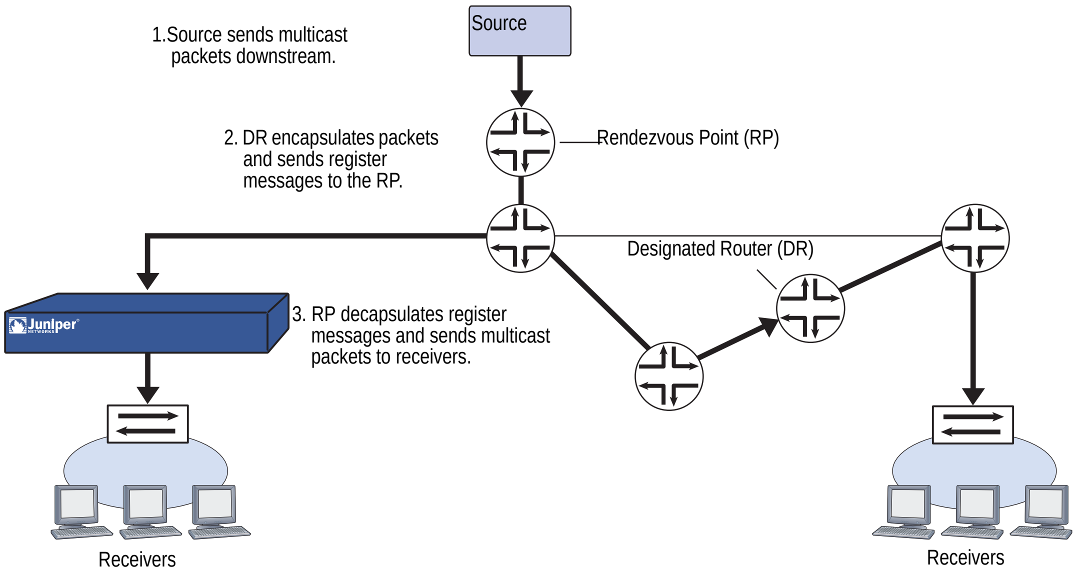
Forwarding Traffic on the Distribution Tree

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 32: Source Sending Data

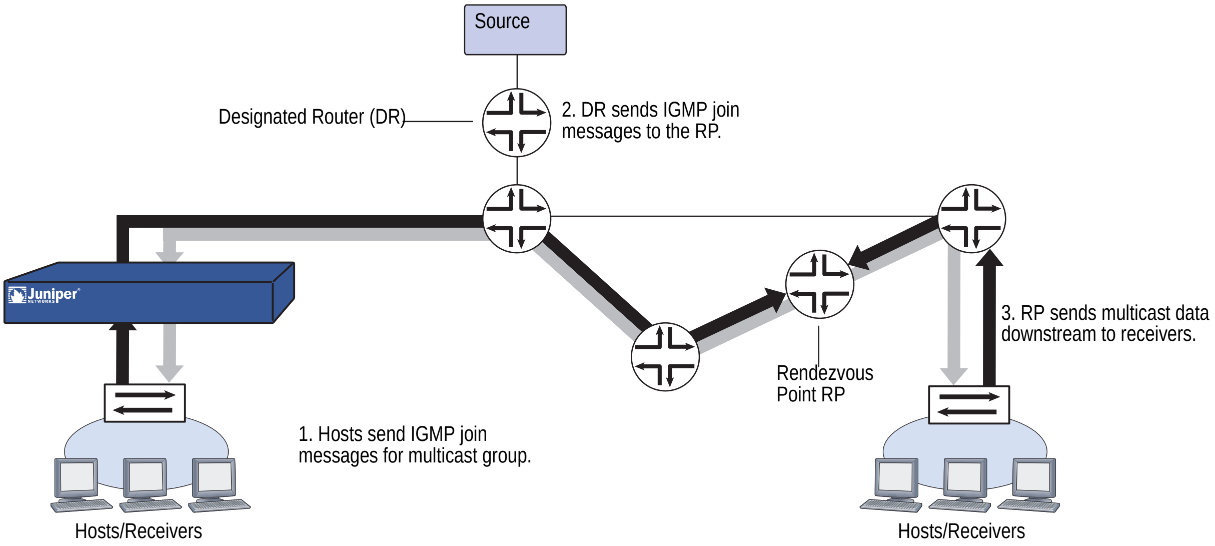

1.Source sends multicast packets downstream.
Source
2. DR encapsulates packets and sends register messages to the RP.
Rendezvous Point (RP)
Designated Router (DR)
Juniper NETWORKS
3. RP decapsulates register messages and sends multicast packets to receivers.
Receivers
Receivers

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 33: Host Joining a Group


Source
Designated Router (DR)
2. DR sends IGMP join messages to the RP.
Juniper NETWORKS
3. RP sends multicast data downstream to receivers.
Rendezvous Point RP
1. Hosts send IGMP join messages for multicast group.
Hosts/Receivers
Hosts/Receivers

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 15.) Security devices support the following dynamic routing protocols:

- Open Shortest Path First (OSPF)—For more information, see “Open Shortest Path First” on page 47.
- Routing Information Protocol (RIP)—For more information, see “Routing Information Protocol” on page 75.
- Border Gateway Protocol (BGP)—For more information, see “Border Gateway Protocol” on page 103.

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 163.)

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.

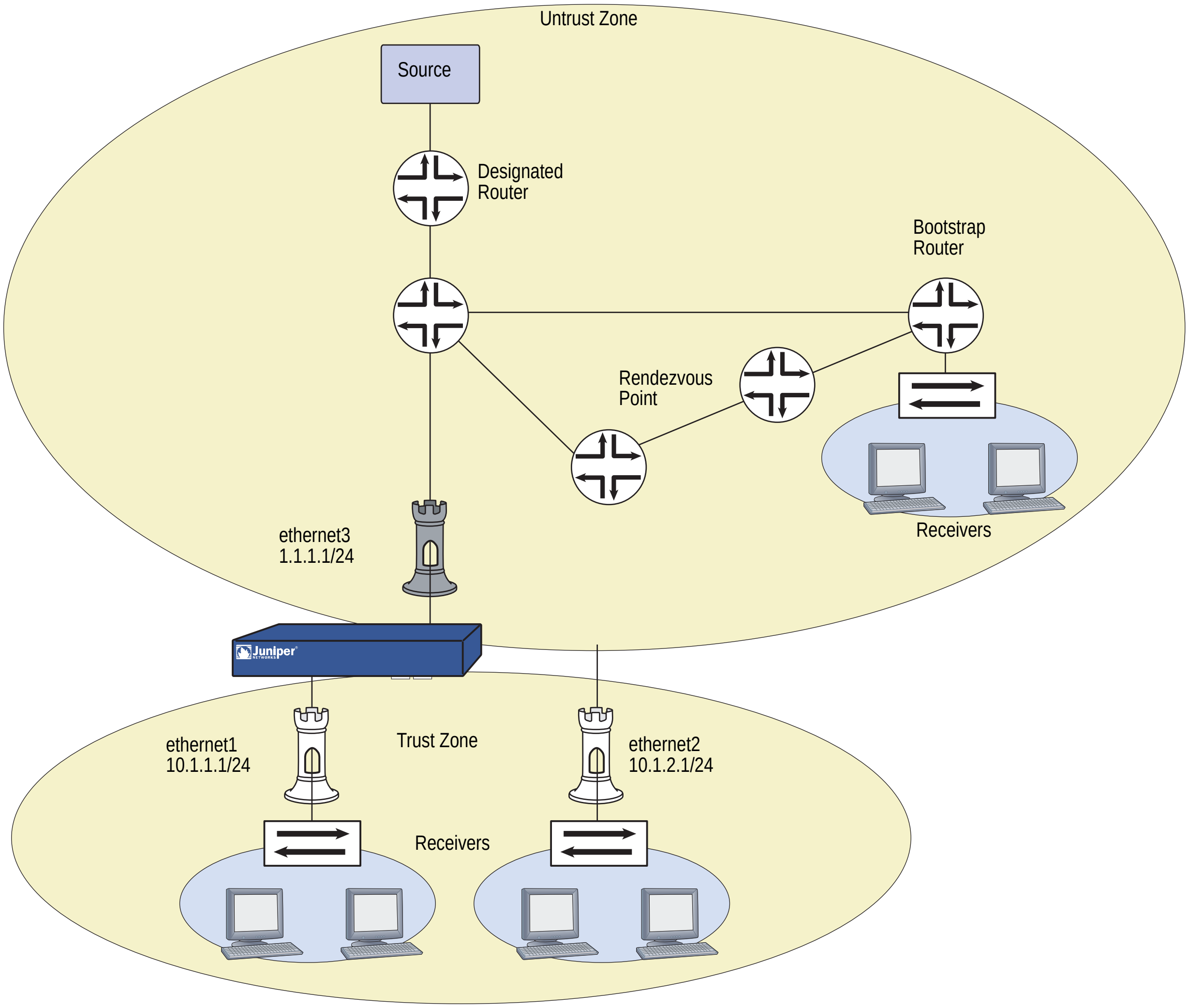
When you configure PIM-SM across two VRs, you must configure the RP in the zone of the VR in which the RP is located. Then, configure a multicast group policy allowing join-prune and BSR-static-RP messages between the zones in each VR. You must also export unicast routes between the two VRs to ensure the accuracy of the reverse path forwarding (RPF) information. For information about exporting routes, see “Exporting and Importing Routes Between Virtual Routers” on page 44.

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 *Volume 10: Virtual Systems*.) 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 212.)

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 34: 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

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 159.)

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 19 describes the PIM-SM interface parameters and their defaults.

Table 19: 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 211.	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 212.	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 206.)

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.

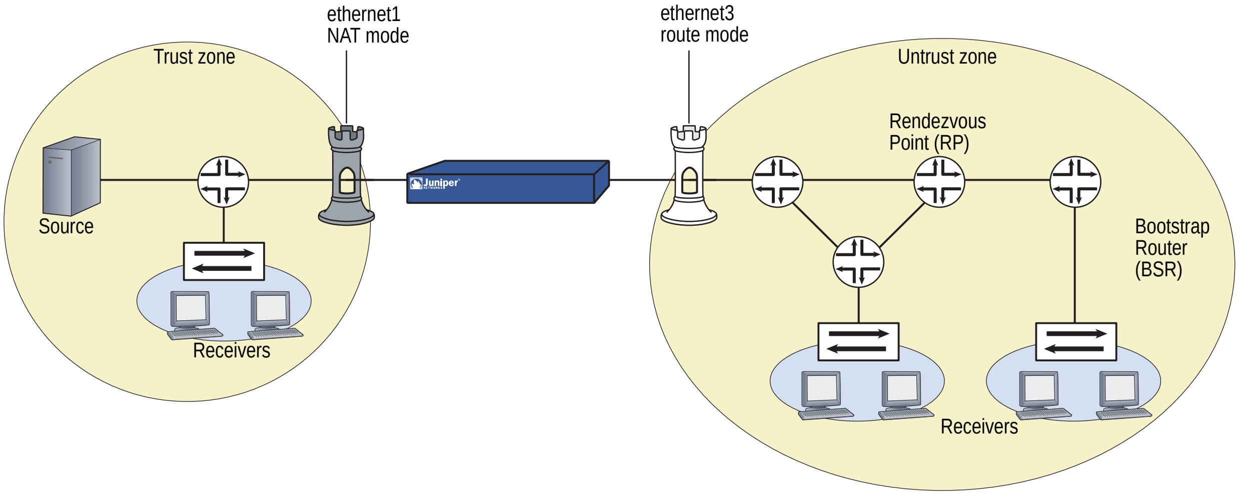
Proxy RP supports the use of Mapped IPs (MIP) for source address translation. A MIP is a direct one-to-one mapping of one IP address to another. You can configure a MIP when you want the security device to translate a private address in a zone whose interfaces are in NAT mode to another address. When 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 35: 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.

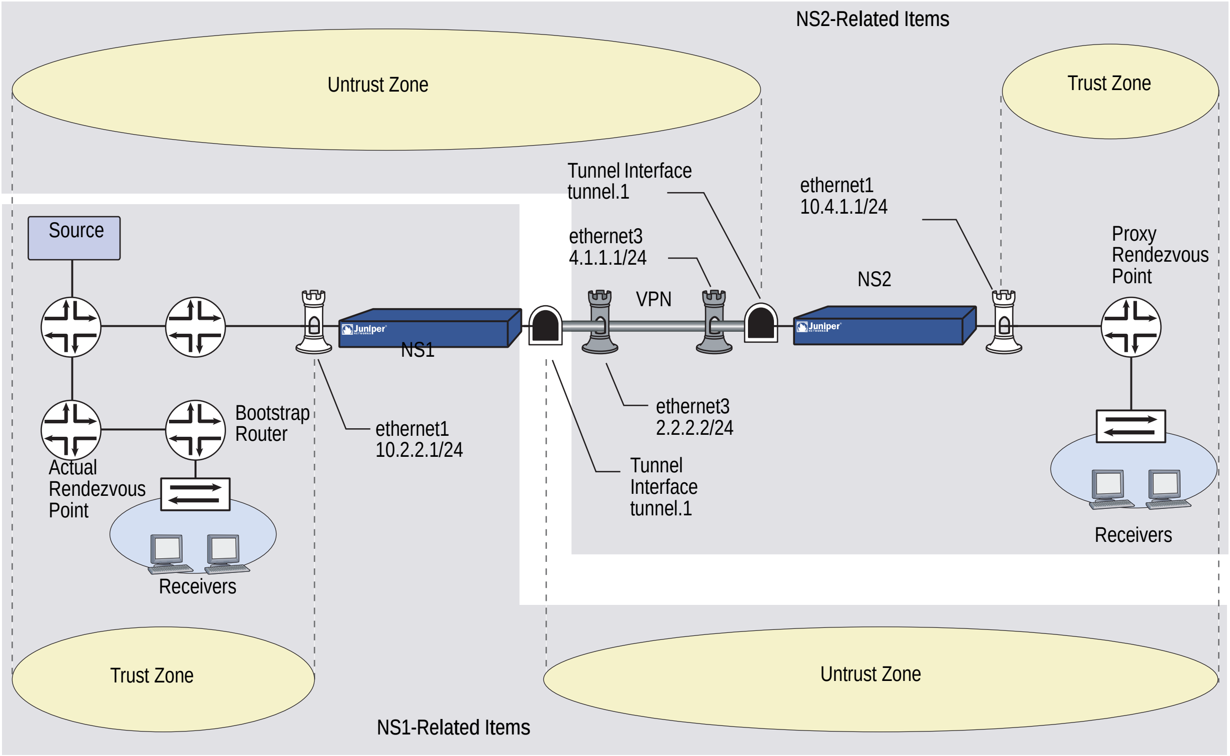
2. Enable PIM-SM on the appropriate interfaces.

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 36: 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 10

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 with NetScreen Redundancy Protocol (NSRP).

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 2-35.

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 20 lists the IRDP parameters, default values, and available settings.

Table 20: 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
```

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
```

Index