Configuring Unicast RPF

For interfaces that carry IPv4 or IPv6 traffic, you can reduce the impact of denial of service (DoS) attacks by configuring unicast reverse path forwarding (RPF). Unicast RPF helps determine the source of attacks and rejects packets from unexpected source addresses on interfaces where unicast RPF is enabled.

NOTE: If you want to configure unicast RPF, your routing platform must be equipped with the Internet Processor II application-specific integrated circuit (ASIC).

If you enable unicast RPF on live traffic, some packets are dropped while the packet forwarding components are updating.

For transit packets exiting the router through the tunnel, forwarding path features, such as RPF, forwarding table filtering, source class usage, and destination class usage are not supported on the interfaces you configure as the output interface for tunnel traffic. For firewall filtering, you must allow the output tunnel packets through the firewall filter applied to input traffic on the interface that is the next-hop interface towards the tunnel destination.

The following sections describe unicast RPF in detail:

Configuring Unicast RPF Strict Mode

In strict mode, unicast RPF checks whether the incoming packet has a source address that matches a prefix in the routing table, and whether the interface expects to receive a packet with this source address prefix.

If the incoming packet fails the unicast RPF check, the packet is not accepted on the interface. When a packet is not accepted on an interface, unicast RPF counts the packet and sends it to an optional fail filter. If the fail filter is not configured, the default action is to silently discard the packet.

The optional fail filter allows you to apply a filter to packets that fail the unicast RPF check. You can define the fail filter to perform any filter operation, including accepting, rejecting, logging, sampling, or policing.

When unicast RPF is enabled on an interface, Bootstrap Protocol (BOOTP) packets and Dynamic Host Configuration Protocol (DHCP) packets are not accepted on the interface. To allow the interface to accept BOOTP packets and DHCP packets, you

must apply a fail filter that accepts all packets with a source address of 0.0.0.0 and a destination address of 255.255.255.255. For a configuration example, see "Example: Configuring Unicast RPF" on page 5.

For more information about unicast RPF, see the *JUNOS Routing Protocols Configuration Guide*. For more information about defining fail filters, see the *JUNOS Policy Framework Configuration Guide*.

To configure unicast RPF, include the rpf-check statement:

```
rpf-check <fail-filter filter-name>;
```

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6)]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (inet | inet6)]

Using unicast RPF can have several consequences when implemented with traffic filters:

- RPF fail filters are evaluated after input filters and before output filters.
- If you configure a filter counter for packets dropped by an input filter, and you want to know the total number of packets dropped, you must also configure a filter counter for packets dropped by the RPF check.
- To count packets that fail the RPF check and are accepted by the RPF fail filter, you must configure a filter counter.
- If an input filter forwards packets anywhere other than the inet.0 or inet6.0 routing tables, the unicast RPF check is not performed.
- If an input filter forwards packets anywhere other than the routing instance the input interface is configured for, the unicast RPF check is not performed.

Configuring Unicast RPF Loose Mode

By default, unicast RPF uses strict mode. Unicast RPF loose mode is similar to unicast RPF strict mode and has the same configuration restrictions. The only check in loose mode is whether the packet has a source address with a corresponding prefix in the routing table; loose mode does not check whether the interface expects to receive a packet with a specific source address prefix. If a corresponding prefix is not found, unicast RPF loose mode does not accept the packet. As in strict mode, loose mode counts the failed packet and optionally forwards it to a fail filter, which either accepts, rejects, logs, samples, or polices the packet.

To configure unicast RPF loose mode, include the mode:

```
mode loose;
```

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6) rpf-check <fail-filter *filter-name*>]

- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (inet | inet6) rpf-check <fail-filter *filter-name*>]

Unicast RPF and Default Routes

When the active route cannot be chosen from the routes in a routing table, the routing platform chooses a default route. A default route is equivalent to an IP address of 0.0.0.0/0. If you configure a default route, and you configure unicast RPF on an interface that the default route uses, unicast RPF behaves differently than it does otherwise. For information about configuring default routes, see the *JUNOS Routing Protocols Configuration Guide*.

To determine whether the default route uses an interface, enter the show route command:

```
user@host> show route address
```

address is the next-hop address of the configured default route. The default route uses the interfaces shown in the output of the show route command.

The following sections describe how unicast RPF behaves when a default route uses an interface and when a default route does not use an interface:

- Unicast RPF Behavior with a Default Route on page 3
- Unicast RPF Behavior Without a Default Route on page 4

Unicast RPF Behavior with a Default Route

If you configure a default route that uses an interface configured with unicast RPF, unicast RPF behaves as follows:

- Loose mode—All packets are automatically accepted. For this reason, we recommend that you not configure unicast RPF loose mode on interfaces that the default route uses.
- Strict mode—The packet is accepted when either of the following is true:
  - The source address of the packet matches any of the routes (either default or learned) that can be originated from the interface. Note that routes can have multiple destinations associated with them; therefore, if one of the destinations matches the incoming interface of the packet, the packet is accepted.
  - The source address of the packet does not match any of the routes.

The packet is not accepted when either of the following is true:

- The source address of the packet does not match a prefix in the routing table.

- The interface does not expect to receive a packet with this source address prefix.

Unicast RPF Behavior Without a Default Route

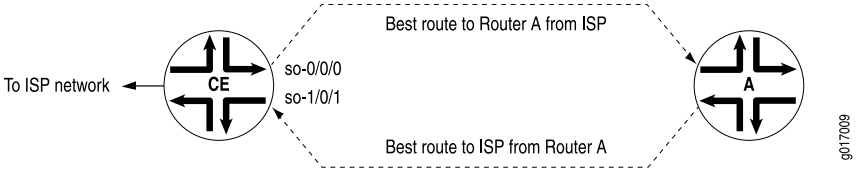
If you do not configure a default route, or if the default route does not use an interface configured with unicast RPF, unicast RPF behaves as described in "Configuring Unicast RPF Strict Mode" on page 1 and "Configuring Unicast RPF Loose Mode" on page 2. To summarize, unicast RPF without a default route behaves as follows:

- Strict mode—The packet is not accepted when either of the following is true:
  - The packet has a source address that does not match a prefix in the routing table.
  - The interface does not expect to receive a packet with this source address prefix.
- Loose mode—The packet is not accepted when the packet has a source address that does not match a prefix in the routing table.

Unicast RPF with Routing Asymmetry

In general, we recommend that you not enable unicast RPF on interfaces that are internal to the network because internal interfaces are likely to have *routing asymmetry*. Routing asymmetry means that a packet's outgoing and return paths are different. Routers in the core of the network are more likely to have asymmetric reverse paths than routing platforms at the customer or provider edge. Figure 1 shows unicast RPF in an environment with routing asymmetry.

Figure 1: Unicast RPF with Routing Asymmetry



In Figure 1, if you enable unicast RPF on interface so-0/0/0, traffic destined for Router A is not rejected. If you enable unicast RPF on interface so-1/0/1, traffic from Router A is rejected.

If you need to enable unicast RPF in an asymmetric routing environment, you can use fail filters to allow the routing platform to accept incoming packets that are known to be arriving by specific paths. For an example of a fail filter that accepts packets with a specific source and destination address, see "Example: Configuring Unicast RPF" on page 5.

Configuring Unicast RPF on a VPN

You can configure unicast RPF on a VPN interface by enabling unicast RPF on the interface and including the interface statement at the [edit routing-instances *routing-instance-name*] hierarchy level.

You can configure unicast RPF only on the interfaces you specify in the routing instance. This means the following:

- For Layer 3 VPNs, unicast RPF is supported on the CE router interface.
- Unicast RPF is not supported on core-facing interfaces.
- For virtual-router routing instances, unicast RPF is supported on all interfaces you specify in the routing instance.
- If an input filter forwards packets anywhere other than the routing instance the input interface is configured for, the unicast RPF check is not performed.

For unicast RPF configuration guidelines, see "Configuring Unicast RPF" on page 1. For more information about VPNs and virtual-router routing instances, see the *JUNOS VPNs Configuration Guide*. For more information about FBF, see the *JUNOS Routing Protocols Configuration Guide*.

Example: Configuring Unicast RPF on a VPN

Configure unicast RPF on a Layer 3 VPN interface:

```
[edit interfaces]
so-0/0/0 {
   unit 0 {
      family inet {
         rpf-check;
      }
   }
}
[edit routing-instance]
VPN-A {
   interface so-0/0/0.0;
}
```

Example: Configuring Unicast RPF

Configure unicast RPF strict mode, and apply a fail filter that allows the interface to accept BOOTP packets and DHCP packets. The filter accepts all packets with a source address of 0.0.0.0 and a destination address of 255.255.255.255.

```
[edit firewall]
filter rpf-special-case-dhcp-bootp {
   term allow-dhcp-bootp {
      from {
         source-address {
            0.0.0.0/32;
```

```
            }
            address {
               255.255.255.255/32;
            }
         }
         then {
            count rpf-dhcp-bootp-traffic;
            accept;
         }
      }
      term default {
         then {
            log;
            reject;
         }
      }
   }
   [edit]
   interfaces {
      so-0/0/0 {
         unit 0 {
            family inet {
               rpf-check fail-filter rpf-special-case-dhcp-bootp;
            }
         }
      }
   }
```