# Zone-Based Policy Firewall IPv6 Support

The zone-based policy firewall provides advanced traffic filtering or inspection of IPv4 packets. With IPv6 support, the zone-based policy firewall supports the inspection of IPv6 packets. Prior to IPv6 support, the firewall supported only the inspection of IPv4 packets. Only Layer 4 protocols, Internet Control Messaging Protocol (ICMP), TCP, and UDP packets are subject to IPv6 packet inspection.

This module describes the firewall features that are supported and how to configure a firewall for IPv6 packet inspection.



## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

## Restrictions for Zone-Based Policy Firewall IPv6 Support

The following functionalities are not supported:

- Application-level gateways (ALGs)
- Box-to-box high availability (HA)

- Distributed Denial-of-Service attacks
- Firewall resource management
- Layer 7 inspection
- Multicast packets
- Per-subscriber firewall or the broadband-based firewall
- Stateless Network Address Translation 64 (NAT64)
- VRF-Aware Software Infrastructure (VASI)
- Wide Area Application Services (WAAS) and Web Cache Communication Protocol (WCCP)

# Information About IPv6 Zone-Based Firewall Support over VASI Interfaces

## IPv6 Support for Firewall Features

The firewall features described in the table below are supported by IPv6 packet inspection:

***Table 1: Firewall Features Supported on IPv6***

| Feature | Configuration Information |
|---|---|
| Class maps | *Zone-Based Policy Firewall* module. |
| Internet Control Message Protocol Version 6 (ICMPv6), TCP, and UDP protocols | • *Firewall Stateful Inspection of ICMP* module.<br>• *Zone-Based Policy Firewall* module. |
| IP fragmentation | *Virtual Fragmentation Reassembly* module. |
| Intrachassis HA | — |
| Logging of error messages | *Zone-Based Policy Firewall* module. |
| Nested class maps | *Nested Class Map Support for Zone-Based Policy Firewall* module. |
| Out-of-order packet handling | The “Out-of-Order Packet Handling” section in the *Zone-Based Policy Firewall* module. |
| Parameter-maps—For inspect type parameter maps, the number of sessions defined in the parameter map will be cumulative for IPv4 and IPv6 sessions | *Zone-Based Policy Firewall* module. |

| Feature | Configuration Information |
|---|---|
| Policy maps | *Zone-Based Policy Firewall* module. |
| Port-to-application mapping | — |
| Stateful Network Address Translation 64 (NAT64) | The *Stateful Network Address Translation 64* module in the *IP Addressing: NAT Configuration Guide*. |
| TCP SYN Cookie | *Configuring Firewall TCP SYN Cookie* module. |
| VPN routing and forwarding (VRF)-aware firewall | *VRF-Aware Cisco IOS XE Firewall* module. |
| Virtual fragmentation reassembly (VFR) | *Virtual Fragmentation Reassembly* module. |
| Zone, default zone, and zone pair | *Zone-Based Policy Firewall* module. |

# Dual-Stack Firewalls

A dual-stack firewall is a firewall running IPv4 and IPv6 traffic at the same time. A dual-stack firewall can be configured in the following scenarios:

- One firewall zone running IPv4 traffic and another running IPv6 traffic.
- IPv4 and IPv6 coexist when deployed with stateful Network Address Translation 64 (NAT64). In this scenario, the traffic flows from IPv6 to IPv4 and vice versa.
- The same zone pair allows both IPv4 and IPv6 traffic.

# Firewall Actions for IPv6 Header Fields

The firewall actions for IPv6 header fields (in the order they are available in the IPv6 header) are described in the following table:

***Table 2: IPv6 Header Fields***

| IPv6 Header Field | IPv6 Header Field Description | Firewall Action |
|---|---|---|
| Version | Similar to the Version field in the IPv4 packet header, except that this field lists number 6 for IPv6, instead of number 4 for IPv4. | Must be IPv6. |
| Traffic Class | Similar to the Type of Service (ToS) field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services. | Not inspected. |

| IPv6 Header Field | IPv6 Header Field Description | Firewall Action |
|---|---|---|
| Flow Label | A new field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer. | Not inspected. |
| Payload Length | Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet. | The firewall uses this field on a limited basis to calculate the length of some of the Layer 4 protocols, such as ICMP and TCP. |
| Next Header Length | Similar to the Protocol field in the IPv4 packet header. The value of the Next Header Length field determines the type of information that follows the basic IPv6 header. The type of information following the basic IPv6 header can be a transport-layer packet, for example, a TCP or a UDP packet, or an extension header. | The firewall must recognize this field to create a session. |
| Hop Limit | Similar to the Time-to-Live (TTL) field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of devices that an IPv6 packet can pass through before the packet is considered invalid. Each device decrements the Hop Limit value by one. Because the IPv6 header does not have a checksum, the device can decrement the value without recalculating the checksum. | Not inspected. |

# IPv6 Firewall Sessions

To perform stateful inspection of traffic, the firewall creates internal sessions for each traffic flow. The session information includes IP source and destination addresses, UDP or TCP source and destination ports or ICMP types, the Layer 4 protocol type (ICMP, TCP, or UDP), and VPN routing and forwarding (VRF) IDs. For an IPv6 firewall, the source and the destination addresses contain 128 bits of the IPv6 address.

The firewall creates a TCP session after receiving the first packet when the packet matches the configured policy. The firewall tracks the TCP sequence numbers and drops the TCP packets whose sequence numbers are not within the configured range. Sessions are removed when the TCP idle timer expires or when a Reset (RST) or Finish-Acknowledge (FIN-ACK) packet is received with the appropriate sequence numbers.

The firewall creates UDP sessions when the first UDP packet that matches the configured policy arrives and removes sessions when the UDP idle timer expires. The firewall does not create TCP or UDP sessions for IPv6 packets with multicast IPv6 or unknown IPv6 addresses.

# Firewall Inspection of Fragmented Packets

The firewall supports the inspection of fragmented IPv6 packets. IP fragmentation is the process of breaking up a single IP datagram into multiple packets of smaller size. In IPv6, end nodes perform a path maximum transmission unit (MTU) discovery to determine the maximum size of the packet that is to be sent and generate IPv6 packets with the fragment extension header for packets larger than the MTU size.

The firewall inspects fragmented packets by using Virtual Fragmentation Reassembly (VFR). VFR examines the fragment extension header for out-of-sequence fragments and puts them in the correct order for inspection. When you enable the firewall on an interface by adding the interface to a zone, VFR is configured automatically on the same interface. If you explicitly disable VFR, the firewall only inspects the first fragments with Layer 4 headers and passes the rest of the fragments without inspection.

The fragment extension header appears in the following order of headers:

- IPv6 header
- Hop-by-hop options header
- Destination options header
- Routing header
- Fragment extension header

Cisco Express Forwarding checks IPv6 packets with fragment extension headers so that the firewall need not do further checks before processing the packets.

# ICMPv6 Messages

IPv6 uses ICMPv6 to perform diagnostic functions, error reporting, and neighbor discovery. ICMPv6 messages are grouped into informational and error messages.

The firewall inspects only the following ICMPv6 messages:

- ECHO REQUEST
- ECHO REPLY
- DESTINATION UNREACHABLE
- PACKET TOO BIG
- PARAMETER PROBLEM
- TIME EXCEEDED

**Note** Neighbor discovery packets are passed and not inspected by the firewall.

# Firewall Support of Stateful NAT64

The zone-based policy firewall supports Stateful NAT64. Stateful NAT64 translates IPv6 packets into IPv4 packets and vice versa. When both the firewall and Stateful NAT64 are configured on a router, the firewall uses IP addresses in an access control list (ACL) to filter packets. However, ACL does not support a mix of IPv4 and IPv6 addresses. Before the firewall and Stateful NAT64 can work together, you must use an IPv6 ACL and the IPv4 address must be embedded in the IPv6 ACL.

**Note** You cannot use VRF along with a firewall and a Stateful NAT64 configuration because Stateful NAT64 is not VRF-aware.

When a firewall class map uses an ACL, the ACL must use the real IP addresses on the host to configure packet flows. If only a source or a destination address is needed, either the IPv4 address or the IPv6 address is used in the class map ACL. Before the packet flow can be filtered based on both the source and destination addresses, the IPv6 address must be used and the IPv4 address must be embedded in the ACL. The ACL has to use IPv6 addresses to filter Stateful NAT64 packets.

**Note** Stateless NAT64 with firewall is not supported.

# Port-to-Application Mapping

Port-to-application mapping (PAM) allows you to customize TCP or UDP port numbers for network services or applications. The firewall uses PAM to correlate TCP or UDP port numbers to specific network services or applications. By mapping port numbers to network services or applications, an administrator can force firewall inspection on custom configurations that are not defined by using well known ports. Use the **ip port-map** command to configure PAM.

# High Availability and ISSU

The IPv6 firewall supports Intrabox HA. Firewall sessions are synchronized to the standby Embedded Services Processors (ESP) for a switchover. In Service Software Upgrade (ISSU) is also supported by the IPv6 firewall.

# Pass Action for a Traffic Class

In a firewall, a traffic class identifies a set of packets based on its contents. You can define a class and apply an action to the identified traffic that reflects a policy. An action is a specific functionality that is associated with a traffic class. You can configure inspect, drop, and pass actions for a class.

The pass action passes the traffic from one zone to another. When the pass action is configured, the firewall does not inspect the traffic; it passes the traffic. In the IPv6 firewall, you must explicitly configure the pass action for the return traffic by defining a zone pair and a policy map with pass action.

The following example shows how to configure the pass action for policy maps, outside-to-inside-policy, and inside-to-outside-policy for IPv6 traffic:

```
policy-map type inspect outside-to-inside-policy
 class type inspect ipv6-class
  pass (Defines pass action for the ipv6-class from the outside to the inside)
!
 class class-default
!
policy-map type inspect inside-to-outside-policy
 class type inspect ipv4-class
  inspect (Defines inspect action for ipv4-class)
 class type inspect v6_class
  pass (Defines pass action for ipv6-class from the inside to the outside)
 class class-default
!
!
zone security inside
!
zone security outside
!
zone-pair security in-out source inside destination outside
 service-policy type inspect inside-to-outside-policy
!
zone-pair security out-in source outside destination inside
 service-policy type inspect outside-to-inside-policy
```

# How to Configure Zone-Based Policy Firewall IPv6 Support

## Configuring an IPv6 Firewall

The steps to configure an IPv4 firewall and an IPv6 firewall are the same. To configure an IPv6 firewall, you must configure the class map in such a way that only an IPv6 address family is matched.

The **match protocol** command applies to both IPv4 and IPv6 traffic and can be included in either an IPv4 policy or an IPv6 policy.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf-definition** *vrf-name*
4. **address-family ipv6**
5. **exit-address-family**
6. **exit**
7. **parameter-map type inspect** *parameter-map-name*
8. **sessions maximum** *sessions*
9. **exit**
10. **ipv6 unicast-routing**
11. **ip port-map** *appl-name* **port** *port-num* **list** *list-name*
12. **ipv6 access-list** *access-list-name*
13. **permit ipv6 any any**
14. **exit**
15. **class-map type inspect match-all** *class-map-name*
16. **match access-group name** *access-group-name*
17. **match protocol** *protocol-name*
18. **exit**
19. **policy-map type inspect** *policy-map-name*
20. **class type inspect** *class-map-name*
21. **inspect** [*parameter-map-name*]
22. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enters privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **vrf-definition** *vrf-name*<br><br>**Example:**<br>`Device(config)# vrf-definition VRF1` | Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **address-family ipv6**<br><br>**Example:**<br>`Device(config-vrf)# address-family ipv6` | Enters VRF address family configuration mode and configures sessions that carry standard IPv6 address prefixes. |
| **Step 5** | **exit-address-family**<br><br>**Example:**<br>`Device(config-vrf-af)# exit-address-family` | Exits VRF address family configuration mode and enters VRF configuration mode. |
| **Step 6** | **exit**<br><br>**Example:**<br>`Device(config-vrf)# exit` | Exits VRF configuration mode and enters global configuration mode. |
| **Step 7** | **parameter-map type inspect** *parameter-map-name*<br><br>**Example:**<br>`Device(config)# parameter-map type inspect ipv6-param-map` | Enables a global inspect-type parameter map for the firewall to connect thresholds, timeouts, and other parameters that pertain to the inspect action, and enters parameter-map type inspect configuration mode. |
| **Step 8** | **sessions maximum** *sessions*<br><br>**Example:**<br>`Device(config-profile)# sessions maximum 10000` | Sets the maximum number of allowed sessions that can exist on a zone pair. |
| **Step 9** | **exit**<br><br>**Example:**<br>`Device(config-profile)# exit` | Exits parameter-map type inspect configuration mode and enters global configuration mode. |
| **Step 10** | **ipv6 unicast-routing**<br><br>**Example:**<br>`Device(config)# ipv6 unicast-routing` | Enables the forwarding of IPv6 unicast datagrams. |
| **Step 11** | **ip port-map** *appl-name* **port** *port-num* **list** *list-name*<br><br>**Example:**<br>`Device(config)# ip port-map ftp port 8090 list ipv6-acl` | Establishes a port to application mapping (PAM) by using the IPv6 access control list (ACL). |
| **Step 12** | **ipv6 access-list** *access-list-name*<br><br>**Example:**<br>`Device(config)# ipv6 access-list ipv6-acl` | Defines an IPv6 access list and enters IPv6 access list configuration mode. |
| **Step 13** | **permit ipv6 any any**<br><br>**Example:**<br>`Device(config-ipv6-acl)# permit ipv6 any any` | Sets permit conditions for an IPv6 access list. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 14** | **exit**<br><br>**Example:**<br>`Device(config-ipv6-acl)# exit` | Exits IPv6 access list configuration mode and enters global configuration mode. |
| **Step 15** | **class-map type inspect match-all** *class-map-name*<br><br>**Example:**<br>`Device(config)# class-map type inspect match-all ipv6-class` | Creates an application-specific inspect type class map and enters QoS class-map configuration mode. |
| **Step 16** | **match access-group name** *access-group-name*<br><br>**Example:**<br>`Device(config-cmap)# match access-group name ipv6-acl` | Configures the match criteria for a class map on the basis of the specified ACL. |
| **Step 17** | **match protocol** *protocol-name*<br><br>**Example:**<br>`Device(config-cmap)# match protocol tcp` | Configures a match criterion for a class map on the basis of the specified protocol. |
| **Step 18** | **exit**<br><br>**Example:**<br>`Device(config-cmap)# exit` | Exits QoS class-map configuration mode and enters global configuration mode. |
| **Step 19** | **policy-map type inspect** *policy-map-name*<br><br>**Example:**<br>`Device(config)# policy-map type inspect ipv6-policy` | Creates a protocol-specific inspect type policy map and enters QoS policy-map configuration mode. |
| **Step 20** | **class type inspect** *class-map-name*<br><br>**Example:**<br>`Device(config-pmap)# class type inspect ipv6-class` | Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode. |
| **Step 21** | **inspect** [*parameter-map-name*]<br><br>**Example:**<br>`Device(config-pmap-c)# inspect ipv6-param-map` | Enables stateful packet inspection. |
| **Step 22** | **end**<br><br>**Example:**<br>`Device(config-pmap-c)# end` | Exits QoS policy-map class configuration mode and enters privileged EXEC mode. |

# Configuring Zones and Applying Zones to Interfaces

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **exit**
5. **zone security** *zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **ipv6 address** *ipv6-address*/*prefix-length*
12. **encapsulation dot1q** *vlan-id*
13. **zone-member security** *zone-name*
14. **end**
15. **show policy-map type inspect zone-pair sessions**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enters privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **zone security** *zone-name*<br><br>**Example:**<br>`Device(config)# zone security z1` | Creates a security zone and enters security zone configuration mode. |
| **Step 4** | **exit**<br><br>**Example:**<br>`Device(config-sec-zone)# exit` | Exits security zone configuration mode and enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **zone security** *zone-name*<br><br>**Example:**<br>`Device(config)# zone security z2` | Creates a security zone and enters security zone configuration mode. |
| **Step 6** | **exit**<br><br>**Example:**<br>`Device(config-sec-zone)# exit` | Exits security zone configuration mode and enters global configuration mode. |
| **Step 7** | **zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** *destination-zone*]<br><br>**Example:**<br>`Device(config)# zone-pair security in-2-out source z1 destination z2` | Creates a zone pair and enters security zone-pair configuration mode. |
| **Step 8** | **service-policy type inspect** *policy-map-name*<br><br>**Example:**<br>`Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy` | Attaches a policy map to a top-level policy map. |
| **Step 9** | **exit**<br><br>**Example:**<br>`Device(config-sec-zone-pair)# exit` | Exits security zone-pair configuration mode and enters global configuration mode. |
| **Step 10** | **interface** *type number*<br><br>**Example:**<br>`Device(config)# interface gigabitethernet 0/0/0.1` | Configures a subinterface and enters subinterface configuration mode. |
| **Step 11** | **ipv6 address** *ipv6-address*/*prefix-length*<br><br>**Example:**<br>`Device(config-subif)# ipv6 address 2001:DB8:2222:7272::72/64` | Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface or a subinterface. |
| **Step 12** | **encapsulation dot1q** *vlan-id*<br><br>**Example:**<br>`Device(config-subif)# encapsulation dot1q 2` | Sets the encapsulation method used by the interface. |
| **Step 13** | **zone-member security** *zone-name*<br><br>**Example:**<br>`Device(config-subif)# zone member security z1` | Configures the interface as a zone member.<br>• For the *zone-name* argument, you must configure one of the zones that you had configured by using the **zone security** command. |

| | Command or Action | Purpose |
|---|---|---|
| | | • When an interface is in a security zone, all traffic to and from that interface (except traffic going to the device or initiated by the device) is dropped by default. To permit traffic through an interface that is a zone member, you must make that zone part of the zone pair to which you apply a policy. If the policy permits traffic (via inspect or pass actions), traffic can flow through the interface. |
| **Step 14** | **end**<br><br>**Example:**<br>`Device(config-subif)# end` | Exits subinterface configuration mode and enters privileged EXEC mode. |
| **Step 15** | **show policy-map type inspect zone-pair sessions**<br><br>**Example:**<br>`Device# show policy-map type inspect zone-pair sessions` | Displays the stateful packet inspection sessions created because a policy map is applied on a specified zone pair.<br><br>• The output of this command displays both IPv4 and IPv6 firewall sessions. |

### Example

The following sample output from the **show policy-map type inspect zone-pair sessions** command displays the translation of packets from an IPv6 address to an IPv4 address and vice versa:

```
Device# show policy-map type inspect zone-pair sessions

  Zone-pair: in-to-out
  Service-policy inspect : in-to-out

    Class-map: ipv6-class (match-any)
      Match: protocol ftp
      Match: protocol tcp
      Match: protocol udp
      Inspect
        Established Sessions
         Session 110D930C [2001:DB8:1::103]:32847=>(209.165.201.2:21) ftp SIS_OPEN
          Created 00:00:00, Last heard 00:00:00
          Bytes sent (initiator:responder) [37:84]

        Half-open Sessions
         Session 110D930C [2001:DB8:1::104]:32848=>(209.165.201.2:21) ftp SIS_OPENING
          Created 00:00:00, Last heard 00:00:00
          Bytes sent (initiator:responder) [0:0]
```

The following sample output from the **show policy-map type inspect zone-pair sessions** command displays the translation of packets from an IPv6 address to an IPv6 address:

```
Device# show policy-map type inspect zone-pair sessions

  Zone-pair: in-to-out
  Service-policy inspect : in-to-out

    Class-map: ipv6-class (match-any)
      Match: protocol ftp
      Match: protocol tcp
      Match: protocol udp
```

```
Inspect
  Established Sessions
   Session 110D930C [2001:DB8:1::103]:63=>[2001:DB8:2::102]:63 udp SIS_OPEN
    Created 00:00:02, Last heard 00:00:01
    Bytes sent (initiator:responder) [162:0]
```

# Configuring an IPv6 Firewall and Stateful NAT64 Port Address Translation

The following task configures an IPv6 firewall with Stateful NAT64 dynamic port address translation (PAT).

A PAT configuration maps multiple IPv6 hosts to a pool of available IPv4 addresses on a first-come first-served basis. The dynamic PAT configuration directly helps conserve the scarce IPv4 address space while providing connectivity to the IPv4 Internet.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **no ip address**
6. **zone-member security** *zone-name*
7. **negotiation auto**
8. **ipv6 address** *ipv6-address*/*prefix-length*
9. **ipv6 enable**
10. **nat64 enable**
11. **exit**
12. **interface** *type number*
13. **ip address** *ip-address mask*
14. **zone member security** *zone-name*
15. **negotiation auto**
16. **nat64 enable**
17. **exit**
18. **ipv6 access-list** *access-list-name*
19. **permit ipv6 host** *source-ipv6-address* **host** *destination-ipv6-address*
20. **exit**
21. **ipv6 route** *ipv6-prefix*/*length interface-type interface-number*
22. **ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address*
23. **nat64 v4 pool** *pool-name start-ip-address end-ip-address*
24. **nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload**
25. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enters privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **ipv6 unicast-routing**<br><br>**Example:**<br>`Device(config)# ipv6 unicast-routing` | Enables the forwarding of IPv6 unicast datagrams. |
| **Step 4** | **interface** *type number*<br><br>**Example:**<br>`Device(config)# interface gigabitethernet 0/0/0` | Configures an interface and enters interface configuration mode. |
| **Step 5** | **no ip address**<br><br>**Example:**<br>`Device(config-if)# no ip address` | Removes an IP address or disables IP processing. |
| **Step 6** | **zone-member security** *zone-name*<br><br>**Example:**<br>`Device(config-if)# zone member security z1` | Attaches an interface to a security zone. |
| **Step 7** | **negotiation auto**<br><br>**Example:**<br>`Device(config-if)# negotiation auto` | Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface. |
| **Step 8** | **ipv6 address** *ipv6-address*/*prefix-length*<br><br>**Example:**<br>`Device(config-if)# ipv6 address 2001:DB8:1::2/96` | Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface. |
| **Step 9** | **ipv6 enable**<br><br>**Example:**<br>`Device(config-if)# ipv6 enable` | Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 10** | **nat64 enable**<br><br>**Example:**<br>`Device(config-if)# nat64 enable` | Enables NAT64 on an interface. |
| **Step 11** | **exit**<br><br>**Example:**<br>`Device(config-if)# exit` | Exits interface configuration mode and enters global configuration mode. |
| **Step 12** | **interface** *type number*<br><br>**Example:**<br>`Device(config)# interface gigabitethernet 0/0/1` | Configures an interface and enters interface configuration mode. |
| **Step 13** | **ip address** *ip-address mask*<br><br>**Example:**<br>`Device(config-if)# ip address 209.165.201.25 255.255.255.0` | Sets a primary or secondary IP address for an interface. |
| **Step 14** | **zone member security** *zone-name*<br><br>**Example:**<br>`Device(config-if)# zone member security z2` | Attaches an interface to a security zone. |
| **Step 15** | **negotiation auto**<br><br>**Example:**<br>`Device(config-if)# negotiation auto` | Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface. |
| **Step 16** | **nat64 enable**<br><br>**Example:**<br>`Device(config-if)# nat64 enable` | Enables NAT64 on an interface. |
| **Step 17** | **exit**<br><br>**Example:**<br>`Device(config-if)# exit` | Exits interface configuration mode and enters global configuration mode. |
| **Step 18** | **ipv6 access-list** *access-list-name*<br><br>**Example:**<br>`Device(config)# ipv6 access-list ipv6-ipv4-pair` | Defines an IPv6 access list and enters IPv6 access list configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 19** | **permit ipv6 host** *source-ipv6-address* **host** *destination-ipv6-address*<br><br>**Example:**<br>`Device(config-ipv6-acl)# permit ipv6 host 2001:DB8:1::2 host 209.165:201.25` | Sets permit conditions for an IPv6 access list, a source IPv6 host address, and a destination IPv6 host address. |
| **Step 20** | **exit**<br><br>**Example:**<br>`Device(config-ipv6-acl)# exit` | Exits IPv6 access list configuration mode and enters global configuration mode. |
| **Step 21** | **ipv6 route** *ipv6-prefix/length interface-type interface-number*<br><br>**Example:**<br>`Device(config)# ipv6 route 2001:DB8:1::2/96 gigabitethernet 0/0/0` | Establishes static IPv6 routes. |
| **Step 22** | **ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address*<br><br>**Example:**<br>`Device(config)# ipv6 neighbor 2001:DB8:1::2/96 gigabitethernet 0/0/0 0000.29f1.4841` | Configures a static entry in the IPv6 neighbor discovery cache. |
| **Step 23** | **nat64 v4 pool** *pool-name start-ip-address end-ip-address*<br><br>**Example:**<br>`Device(config)# nat64 v4 pool pool1 209.165.201.25 209.165.201.125` | Defines a Stateful NAT64 IPv4 address pool. |
| **Step 24** | **nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload**<br><br>**Example:**<br>`Device(config)# nat64 v6v4 list nat64-ipv6-any pool pool1 overload` | Enables NAT64 PAT or overload address translation. |
| **Step 25** | **end**<br><br>**Example:**<br>`Device(config)# end` | Exits global configuration mode and enters privileged EXEC mode. |

# Configuration Examples for Zone-Based Policy Firewall IPv6 Support

## Example: Configuring an IPv6 Firewall

```
Device# configure terminal
Device(config)# vrf-definition VRF1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
Device(config)# parameter-map type inspect ipv6-param-map
Device(config-profile)# sessions maximum 10000
Device(config-profile)# exit
Device(config)# ipv6 unicast-routing
Device(config)# ip port-map ftp port 8090 list ipv6-acl
Device(config)# ipv6 access-list ipv6-acl
Device(config-ipv6-acl)# permit ipv6 any any
Device(config-ipv6-acl)# exit
Device(config)# class-map type inspect match-all ipv6-class
Device(config-cmap)# match access-group name ipv6-acl
Device(config-cmap)# match protocol tcp
Device(config-cmap)# exit
Device(config)# policy-map type inspect ipv6-policy
Device(config-pmap)# class type inspect ipv6-class
Device(config-pmap-c)# inspect ipv6-param-map
Device(config-pmap-c)# end
```

## Example: Configuring Zones and Applying Zones to Interfaces

```
Device# configure terminal
Device(config)# zone security z1
Device(config-sec-zone)# exit
Device(config)# zone security z2
Device(config-sec-zone)# exit
Device(config)# zone-pair security in-to-out source z1 destination z2
Device(config-sec-zone-pair)# service-policy type inspect ipv6-policy
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 0/0/0.1
Device(config-if)# ipv6 address 2001:DB8:2222:7272::72/64
Device(config-if)# encapsulation dot1q 2
Device(config-if)# zone member security z1
Device(config-if)# end
```

## Example: Configuring an IPv6 Firewall and Stateful NAT64 Port Address Translation

```
configure terminal
 ipv6 unicast-routing
 interface gigabitethernet 0/0/0
  no ip address
  zone member security z1
  negotiation auto
  ipv6 address 2001:DB8:1::2/96
  ipv6 enable
  nat64 enable
```

```
!
 interface gigabitethernet 0/0/1
  ip address 209.165.201.25 255.255.255.0
  zone member security z2
  negotiation auto
  nat64 enable
!
 ipv6 access-list ipv6-ipv4-pair
  permit ipv6 host 2001:DB8:1::2 host 209.165:201.25
!
 ipv6 route 2001:DB8:1::2/96 gigabitethernet 0/0/0
 ipv6 neighbor 2001:DB8:1::2/96 gigabitethernet 0/0/0 0000.29f1.4841
 nat64 v4 pool pool1 209.165.201.25 209.165.201.125
 nat64 v6v4 list nat64-ipv6-any pool pool1 overload
```

# Additional References for Zone-Based Policy Firewall IPv6 Support

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Master Commands List, All Releases |
| Security commands | • Security Command Reference: Commands A to C<br>• Security Command Reference: Commands D to L<br>• Security Command Reference: Commands M to R<br>• Security Command Reference: Commands S to Z |
| Stateful NAT64 | Stateful Network Address Translation 64 |

### Standards and RFCs

| Standard/RFC | Title |
|---|---|
| RFC 2460 | *Internet Protocol, Version 6 (IPv6) Specification* |
| RFC 2473 | *Generic Packet Tunneling in IPv6 Specification* |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

# Feature Information for Zone-Based Policy Firewall IPv6 Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

***Table 3: Feature Information for Zone-Based Policy Firewall IPv6 Support***

| Feature Name | Releases | Feature Information |
|---|---|---|
| Zone-Based Policy Firewall IPv6 Support | Cisco IOS XE Release 3.6S | The Zone-Based Policy firewall supports the inspection of IPv6 packets.<br>The following commands were introduced or modified: **ip port-map** and **show policy-map type inspect zone-pair**. |