# Configuring IPv6 ACLs

When the advanced IP services image is installed on the Catalyst 3750 switch, you can filter IP Version 6 (IPv6) traffic by creating IPv6 access control lists (ACLs) and applying them to interfaces similarly to the way that you create and apply IP Version 4(IPv4) named ACLs. This chapter includes information about configuring IPv6 ACLs on the switch. Unless otherwise noted, the term *switch* refers to a standalone switch and to a switch stack.

**Note** To use IPv6, the switch or stack master must be running the advanced IP services image and you must configure the dual IPv4 and IPv6 Switch Database Management (SDM) template on the switch. You select the template by entering the **sdm prefer dual-ipv4-and-ipv6** {**default** | **vlan**}[**desktop**] global configuration command.

For related information, see these chapters:

- For more information about SDM templates, see Chapter 8, "Configuring SDM Templates."
- For information about IPv6 on the switch, seeChapter 36, "Configuring IPv6 Unicast Routing.".
- For information about ACLs on the switch, see Chapter 32, "Configuring Network Security with ACLs.".

**Note** For complete syntax and usage information for the commands used in this chapter, see the command reference for this release or the Cisco IOS documentation referenced in the procedures.

This chapter contains these sections:

# Understanding IPv6 ACLs

The switch supports two types of IPv6 ACLs:

- IPv6 router ACLs are supported on outbound or inbound traffic on Layer 3 interfaces, which can be routed ports, switch virtual interfaces (SVIs), or Layer 3 EtherChannels. IPv6 router ACLs apply only to IPv6 packets that are routed.
- IPv6 port ACLs are supported on inbound traffic on Layer 2 interfaces only. IPv6 port ACLs are applied to all IPv6 packets entering the interface.

The switch does not support VLAN ACLs (VLAN maps) for IPv6 traffic.

**Note** For more information about ACL support on the switch, see Chapter 32, "Configuring Network Security with ACLs.".

You can apply both IPv4 and IPv6 ACLs to an interface.

As with IPv4 ACLs, IPv6 port ACLs take precedence over router ACLs:

- When an input router ACL and input port ACL exist in an SVI, packets received on ports to which a port ACL is applied are filtered by the port ACL. Routed IP packets received on other ports are filtered by the router ACL. Other packets are not filtered.
- When an output router ACL and input port ACL exist in an SVI, packets received on the ports to which a port ACL is applied are filtered by the port ACL. Outgoing routed IPv6 packets are filtered by the router ACL. Other packets are not filtered.

**Note** If *any* port ACL (IPv4, IPv6, or MAC) is applied to an interface, that port ACL is used to filter packets, and any router ACLs attached to the SVI of the port VLAN are ignored.

These sections describe some characteristics of IPv6 ACLs on the switch:

- Supported ACL Features, page 38-2
- IPv6 ACL Limitations, page 38-3
- IPv6 ACLs and Switch Stacks, page 38-3

## Supported ACL Features

IPv6 ACLs on the switch have these characteristics:

- Fragmented frames (the **fragments** keyword as in IPv4) are supported.
- The same statistics supported in IPv4 are supported for IPv6 ACLs.
- If the switch runs out of TCAM space, packets associated with the ACL label are forwarded to the CPU, and the ACLs are applied in software.
- Routed or bridged packets with hop-by-hop options have IPv6 ACLs applied in software.
- Logging is supported for router ACLs, but not for port ACLs.

## IPv6 ACL Limitations

With IPv4, you can configure standard and extended numbered IP ACLs, named IP ACLs, and MAC ACLs. IPv6 supports only named ACLs.

The switch supports most Cisco IOS-supported IPv6 ACLs with some exceptions:

- IPv6 source and destination addresses—ACL matching is supported only on prefixes from /0 to /64 and host addresses (/128) that are in the extended universal identifier (EUI)-64 format. The switch supports only these host addresses with no loss of information:
  - aggregatable global unicast addresses
  - link local addresses
- The switch does not support matching on these keywords: **flowlabel**, **routing header**, and **undetermined-transport**.
- The switch does not support reflexive ACLs (the **reflect** keyword).
- This release supports only port ACLs and router ACLs for IPv6; it does not support VLAN ACLs (VLAN maps).
- The switch does not apply MAC-based ACLs on IPv6 frames.
- You cannot apply IPv6 port ACLs to Layer 2 EtherChannels.
- For this release, IPv6 ACLs are supported only on switch stacks that are running the advanced IP services image.
- When configuring an ACL, there is no restriction on keywords entered in the ACL, regardless of whether or not they are supported on the platform. When you apply the ACL to an interface that requires hardware forwarding (physical ports or SVIs), the switch checks to determine whether or not the ACL can be supported on the interface. If not, attaching the ACL is rejected.
- If an ACL is applied to an interface and you attempt to add an ACE with an unsupported keyword, the switch does not allow the ACE to be added to the ACL that is currently attached to the interface.

## IPv6 ACLs and Switch Stacks

The stack master supports IPv6 ACLs in hardware and distributes the IPv6 ACLs to the stack members.

**Note** For IPv6 functionality in a switch stack, all stack members must be running the advanced IP services image.

If a new switch takes over as stack master, it distributes the ACL configuration to all stack members. The member switches sync up the configuration distributed by the new stack master and flush out entries that are not required.

When an ACL is modified, attached to, or detached from an interface, the stack master distributes the change to all stack members.

# Configuring IPv6 ACLs

Before configuring IPv6 ACLs, you must select one of the dual IPv4 and IPv6 SDM templates. If you are going to configure router ACLs, you must also enable IPv6 routing on the switchstack.

To filter IPv6 traffic, you perform these steps:

**Step 1** Create an IPv6 ACL, and enter IPv6 access list configuration mode.

**Step 2** Configure the IPv6 ACL to block (deny) or pass (permit) traffic.

**Step 3** Apply the IPv6 ACL to an interface. For router ACLs, you must also configure an IPv6 address on the Layer 3 interface to which the ACL is applied.

These sections describe how to configure and apply IPv6 ACLs:

- Default IPv6 ACL Configuration, page 38-4
- Interaction with Other Features, page 38-4
- Creating IPv6 ACLs, page 38-5
- Applying an IPv6 ACL to an Interface, page 38-8

## Default IPv6 ACL Configuration

There are no IPv6 ACLs configured or applied.

## Interaction with Other Features

Configuring IPv6 ACLs has these interactions with other features or switch characteristics:

- If an IPv6 router ACL is configured to deny a packet, the packet is not routed. A copy of the packet is sent to the Internet Control Message Protocol (ICMP) queue to generate an ICMP unreachable message for the frame.
- If a bridged frame is to be dropped due to a port ACL, the frame is not bridged.
- You can create both IPv4 and IPv6 ACLs on a stack, and you can apply both IPv4 and IPv6 ACLs to the same interface. Each ACL must have a unique name; an error message appears if you try to use a name that is already configured.

  You use different commands to create IPv4 and IPv6 ACLs and to attach IPv4 or IPv6 ACLs to the same Layer 2 or Layer 3 interface. If you use the wrong command to attach an ACL (for example, an IPv4 command to attach an IPv6 ACL), you receive an error message.
- You cannot use MAC ACLs to filter IPv6 frames. MAC ACLs can only filter non-IP frames.
- If the TCAM is full, for any additional configured ACLs, packets are forwarded to the CPU, and the ACLs are applied in software.

## Creating IPv6 ACLs

Beginning in privileged EXEC mode, follow these steps to create an IPv6 ACL:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **ipv6 access-list** *access-list-name* | Define an IPv6 access list using a name, and enter IPv6 access-list configuration mode. |

| | Command | Purpose |
|---|---|---|
| **Step 3a** | **deny** \| **permit** *protocol* {*source-ipv6-prefix*/*prefix-length* \| **any** \| **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/ *prefix-length* \| **any** \| **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dscp** *value*] [**fragments**] [**log**] [**log-input**] [**sequence** *value*] [**time-range** *name*] | Enter **deny** or **permit** to specify whether to deny or permit the packet if conditions are matched. These are the conditions:<br>• For *protocol*, enter the name or number of an Internet protocol: **ahp**, **esp**, **icmp**, **ipv6**, **pcp**, **stcp**, **tcp**, or **udp**, or an integer in the range 0 to 255 representing an IPv6 protocol number.<br>**Note** For additional specific parameters for ICMP, TCP, and UDP, see Steps 3b through 3d.<br>• The *source-ipv6-prefix/prefix-length* or *destination-ipv6-prefix/ prefix-length* is the source or destination IPv6 network or class of networks for which to set deny or permit conditions, specified in hexadecimal and using 16-bit values between colons (see RFC 2373).<br>**Note** Although the CLI help shows a prefix-length range of /0 to /128, the switch supports IPv6 address matching only for prefixes in the range of /0 to /64 and EUI-based /128 prefixes for aggregatable global unicast and link-local host addresses.<br>• Enter **any** as an abbreviation for the IPv6 prefix ::/0.<br>• For **host** *source-ipv6-address* or *destination-ipv6-address,* enter the source or destination IPv6 host address for which to set deny or permit conditions, specified in hexadecimal using 16-bit values between colons.<br>• (Optional) For *operator*, specify an operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range**.<br>If the operator follows the *source-ipv6-prefix/prefix-length* argument, it must match the source port. If the operator follows the *destination-ipv6-prefix/prefix-length* argument, it must match the destination port.<br>• (Optional) The *port-number* is a decimal number from 0 to 65535 or the name of a TCP or UDP port. You can use TCP port names only when filtering TCP. You can use UDP port names only when filtering UDP.<br>• (Optional) Enter **dscp** *value* to match a differentiated services code point value against the traffic class value in the Traffic Class field of each IPv6 packet header. The acceptable range is from 0 to 63.<br>• (Optional) Enter **fragments** to check noninitial fragments. This keyword is visible only if the protocol is **ipv6**.<br>• (Optional) Enter **log** to cause an logging message to be sent to the console about the packet that matches the entry. Enter **log-input** to include the input interface in the log entry. Logging is supported only for router ACLs.<br>• (Optional) Enter **sequence** *value* to specify the sequence number for the access list statement. The acceptable range is from 1 to 4294967295.<br>• (Optional) Enter **time-range** *name* to specify the time range that applies to the deny or permit statement. |

| | Command | Purpose |
|---|---|---|
| **Step 3b** | **deny** \| **permit tcp** {*source-ipv6-prefix*/*prefix-length* \| **any** \| **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length* \| **any** \| **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**ack**] [**dscp** *value*] [**established**] [**fin**] [**log**] [**log-input**] [**neq** {*port* \| *protocol*}] [**psh**] [**range** {*port* \| *protocol*}] [**rst**] [**sequence** *value*] [**syn**] [**time-range** *name*] [**urg**] | (Optional) Define a TCP access list and the access conditions.<br>Enter **tcp** for Transmission Control Protocol. The parameters are the same as those described in Step 3a, with these additional optional parameters:<br>• **ack**—Acknowledgment bit set.<br>• **established**—An established connection. A match occurs if the TCP datagram has the ACK or RST bits set.<br>• **fin**—Finished bit set; no more data from sender.<br>• **neq** {*port* \| *protocol*}—Matches only packets that are not on a given port number.<br>• **psh**—Push function bit set.<br>• **range** {*port* \| *protocol*}—Matches only packets in the port number range.<br>• **rst**—Reset bit set.<br>• **syn**—Synchronize bit set.<br>• **urg**—Urgent pointer bit set. |
| **Step 3c** | **deny** \| **permit udp** {*source-ipv6-prefix*/*prefix-length* \| **any** \| **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length* \| **any** \| **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dscp** *value*] [**log**] [**log-input**] [**neq** {*port* \| *protocol*}] [**range** {*port* \| *protocol*}] [**sequence** *value*] [**time-range** *name*] | (Optional) Define a UDP access list and the access conditions.<br>Enter **udp** for the User Datagram Protocol. The UDP parameters are the same as those described for TCP, except that the [*operator* [*port*]] port number or name must be a UDP port number or name, and the **established** parameter is not valid for UDP. |
| **Step 3d** | **deny** \| **permit icmp** {*source-ipv6-prefix*/*prefix-length* \| **any** \| **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length* \| **any** \| **host** *destination-ipv6-address*} [*operator* [*port-number*]] [*icmp-type* [*icmp-code*] \| *icmp-message*] [**dscp** *value*] [**log**] [**log-input**] [**sequence** *value*] [**time-range** *name*] | (Optional) Define an ICMP access list and the access conditions.<br>Enter **icmp** for Internet Control Message Protocol. The ICMP parameters are the same as those described for most IP protocols in Step 3a, with the addition of the ICMP message type and code parameters. These optional keywords have these meanings:<br>• *icmp-type*—Enter to filter by ICMP message type, a number from 0 to 255.<br>• *icmp-code*—Enter to filter ICMP packets that are filtered by the ICMP message code type, a number from 0 to 255.<br>• *icmp-message*—Enter to filter ICMP packets by the ICMP message type name or the ICMP message type and code name. To see a list of ICMP message type names and code names, use the ? key or see command reference for this release. |
| **Step 4** | **end** | Return to privileged EXEC mode. |
| **Step 5** | **show ipv6 access-list** | Verify the access list configuration. |
| **Step 6** | **copy running-config startup-config** | (Optional) Save your entries in the configuration file. |

Use the **no deny | permit** IPv6 access-list configuration commands with keywords to remove the deny or permit conditions from the specified access list.

This example configures the IPv6 access list named CISCO. The first deny entry in the list denies all packets that have a destination TCP port number greater than 5000. The second deny entry denies packets that have a source UDP port number less than 5000. The second deny also logs all matches to the console. The first permit entry in the list permits all ICMP packets. The second permit entry in the list permits all other traffic. The second permit entry is necessary because an implicit deny -all condition is at the end of each IPv6 access list.

```
Switch(config)# ipv6 access-list CISCO
Switch(config-ipv6-acl)# deny tcp any any gt 5000
Switch config-ipv6-acl)# deny ::/0 lt 5000 ::/0 log
Switch(config-ipv6-acl)# permit icmp any any
Switch(config-ipv6-acl)# permit any any
```

## Applying an IPv6 ACL to an Interface

This section describes how to apply IPv6 ACLs to network interfaces. You can apply an ACL to outbound or inbound traffic on Layer 3 interfaces. You can apply ACLs only to inbound traffic on Layer 2 interfaces.

Beginning in privileged EXEC mode, follow these steps to control access to an interface:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **interface** *interface-id* | Identify a Layer 2 interface (for port ACLs) or Layer 3 interface (for router ACLs) on which to apply an access list, and enter interface configuration mode. |
| Step 3 | **no switchport** | If applying a router ACL, change the interface from Layer 2 mode (the default) to Layer 3 mode. |
| Step 4 | **ipv6 address** *ipv6-address* | Configure an IPv6 address on a Layer 3 interface (for router ACLs).<br>**Note** This command is not required on Layer 2 interfaces or if the interface has already been configured with an explicit IPv6 address. |
| Step 5 | **ipv6 traffic-filter** *access-list-name* {**in** \| **out**} | Apply the access list to incoming or outgoing traffic on the interface.<br>**Note** The **out** keyword is not supported for Layer 2 interfaces (port ACLs). |
| Step 6 | **end** | Return to privileged EXEC mode. |
| Step 7 | **show running-config** | Verify the access list configuration. |
| Step 8 | **copy running-config startup-config** | (Optional) Save your entries in the configuration file. |

Use the **no ipv6 traffic-filter** *access-list-name* interface configuration command to remove an access list from an interface.

This example shows how to apply the access list *Cisco* to outbound traffic on a Layer 3 interface:

```
Switch(config)# interface gigabitethernet 1/0/3
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001::/64 eui-64
Switch(config-if)# ipv6 traffic-filter CISCO out
```

# Displaying IPv6 ACLs

You can display information about all configured access lists, all IPv6 access lists, or a specific access list by using one or more of the privileged EXEC commands in Table 38-1.

*Table 38-1 Commands for Displaying IPv6 Access List Information*

| Command | Purpose |
|---|---|
| **show access-lists** | Display all access lists configured on the switch. |
| **show ipv6 access-list** [*access-list-name*] | Display all configured IPv6 access list or the access list specified by name. |

This is an example of the output from the **show access-lists** privileged EXEC command. The output shows all access lists that are configured on the switch stack.

```
Switch #show access-lists
Extended IP access list hello
    10 permit ip any any
IPv6 access list ipv6
    permit ipv6 any any sequence 10
```

This is an example of the output from the **show ipv6 access-lists** privileged EXEC command. The output shows only IPv6 access lists configured on the switch stack.

```
Switch# show ipv6 access-list
IPv6 access list inbound
    permit tcp any any eq bgp (8 matches) sequence 10
    permit tcp any any eq telnet (15 matches) sequence 20
    permit udp any any sequence 30

IPv6 access list outbound
    deny udp any any sequence 10
    deny tcp any any eq telnet sequence 20
```