



CHAPTER 32

Configuring IPv6 ACLs

When the Cisco ME 3600/ME 3800 switch is running the metro IP access image, 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. You can also create and apply input router ACLs to filter Layer 3 management traffic.

For related information, see these chapters:

- For information about IPv6 on the switch, see [Chapter 38, “Configuring IPv6 Unicast Routing.”](#)
- For information about ACLs on the switch, see [Chapter 31, “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, page 32-2](#)
- [Configuring IPv6 ACLs, page 32-3](#)
- [Displaying IPv6 ACLs, page 32-7](#)

Understanding IPv6 ACLs

A switch running the metro IP access image 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 routed IPv6 packets.
- IPv6 port ACLs are supported only on inbound traffic on Layer 2 interfaces. IPv6 port ACLs are applied to all IPv6 packets entering the interface.
- IPv6 EVC ACLs are supported only on inbound traffic on EVCs.

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

If you configure unsupported IPv6 ACLs, an error message appears, and the configuration does not take affect.



Note

For more information about IPv4 ACL support on the switch, see [Chapter 31, “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

IPv6 ACLs on EVC cannot be configured where mac or IPv4 ACL are configured.



Note

If *any* port ACL (IPv4, IPv6, or MAC) is applied to an interface, that port ACL filters 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 32-2](#)
- [IPv6 ACL Limitations, page 32-3](#)

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 hardware space, packets associated with the ACL 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.
- The switch supports IPv6 address-matching for a full range of prefix-lengths.

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:

- 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).
- The switch supports tcp/udp port matching only for the **eq** keyword. The **range**, **neq**, **lt** and **gt** keywords are not supported.
- 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.
- The switch does not support output port ACLs.
- 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 determines whether or not the ACL can be supported on the interface. If not, the ACL attachment is rejected.
- If an ACL is applied to an interface and you attempt to add an access control entry (ACE) with an unsupported keyword, the switch does not allow the ACE to be added to the attached ACL.

Configuring IPv6 ACLs

Before configuring IPv6 ACLs, you must select one of the dual IPv4 and IPv6 SDM templates.

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 on which the ACL is applied. |
-

- [Default IPv6 ACL Configuration, page 32-4](#)
- [Interaction with Other Features and Switches, page 32-4](#)
- [Creating IPv6 ACLs, page 32-4](#)
- [Applying an IPv6 ACL to an Interface, page 32-6](#)

Default IPv6 ACL Configuration

There are no IPv6 ACLs configured or applied.

Interaction with Other Features and Switches

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 switch, 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 hardware memory 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 <i>access-list-name</i>	Define an IPv6 access list using a name, and enter IPv6 access-list configuration mode.

Command	Purpose
<p>Step 3</p> <pre>{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] [routing] [sequence value] [time-range name]</pre>	<p>Enter deny or permit to specify whether to deny or to permit the packet if conditions are matched. These are the conditions:</p> <ul style="list-style-type: none"> For <i>protocol</i>, 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. <p>Note Only TCP or UDP protocols are supported</p> <p>Note For additional specific parameters for TCP, and UDP, see Steps 3b through 3d.</p> <ul style="list-style-type: none"> The <i>source-ipv6-prefix/prefix-length</i> or <i>destination-ipv6-prefix/prefix-length</i> 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. Enter any as an abbreviation for the IPv6 prefix <code>::/0</code>. For host <i>source-ipv6-address</i> or <i>destination-ipv6-address</i>, enter the source or destination IPv6 host address for which to set deny or permit conditions, specified in hexadecimal and using 16-bit values between colons. (Optional) For <i>operator</i>, 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. <p>Note Only the eq (equal) operator is supported in this release.</p> <p>If the operator follows the <i>source-ipv6-prefix/prefix-length</i> argument, it must match the source port. If the operator follows the <i>destination-ipv6-prefix/prefix-length</i> argument, it must match the destination port.</p> <ul style="list-style-type: none"> (Optional) The <i>port-number</i> 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. (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. (Optional) Enter fragments to check noninitial fragments. This keyword is visible only if the protocol is ipv6. (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. (Optional) Enter routing to specify that IPv6 packets be routed. (Optional) Enter sequence value to specify the sequence number for the access list statement. The acceptable range is from 1 to 4294967295. (Optional) Enter time-range name to specify the time range that applies to the deny or permit statement.
<p>Step 4</p> <pre>end</pre>	<p>Return to privileged EXEC mode.</p>

	Command	Purpose
Step 5	<code>show ipv6 access-list</code>	Verify the access list configuration.
Step 6	<code>copy running-config startup-config</code>	(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 equal to 5000. The second deny entry denies packets that have a source UDP port number equal to 5000. The second deny also logs all matches to the console. The first permit entry in the list permits all packets that have a destination port number equal to 4000. 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 eq 5000
Switch config-ipv6-acl)# deny ::/0 eq 5000 ::/0 log
Switch(config-ipv6-acl)# permit udp any any eq 4000
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, or 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	<code>configure terminal</code>	Enter global configuration mode.
Step 2	<code>interface interface-id</code>	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	<code>no switchport</code>	If applying a router ACL, change the interface from Layer 2 mode (the default) to Layer 3 mode.
Step 4	<code>ipv6 address ipv6-address</code>	Configure an IPv6 address on a Layer 3 interface (for router ACLs). 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	<code>ipv6 traffic-filter access-list-name {in out}</code>	Apply the access list to incoming or outgoing traffic on the interface. Note The out keyword is not supported for Layer 2 interfaces (port ACLs).
Step 6	<code>end</code>	Return to privileged EXEC mode.
Step 7	<code>show running-config</code>	Verify the access list configuration.
Step 8	<code>copy running-config startup-config</code>	(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 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 32-1](#).

Table 32-1 Commands for Displaying IPv6 Access List Information

Command	Purpose
show access-lists	Display all access lists configured on the switch.
show ipv6 access-list [<i>access-list-name</i>]	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 or 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 or 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
```

