# IPv6 ACLs



## Restrictions for IPv6 ACLs

IPv6 supports only named ACLs. With IPv4 ACLs, you can configure standard and extended numbered IP ACLs, named IP ACLs, and MAC 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 does not apply MAC-based ACLs on IPv6 frames.
- When configuring an ACL, there is no restriction on keywords that are entered in the ACL, regardless of whether they are supported or not 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 ACL can be supported on the interface or not. If the ACL is not supported on the interface, the ACL 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 ACL that is currently attached to the interface.
- When you apply a scale ACL to an interface that does not program TCAM for a protocol and the ACLs that have been unloaded, it can impact the existing normal movement of traffic for other protocols. The restriction is applicable to IPv6 and MAC address traffic.
- Time-to-live (TTL) classification is not supported on ACLs.
- If a downloadable ACL contains any type of duplicate entries, the entries are not auto merged. As a result, the 802.1X session authorization fails. Ensure that the downloadable ACL is optimized without any duplicate entries, for example port-based and name-based entries for the same port.

- Egress ACL lookup is not supported for injected traffic that is forwarded by the software.

# Information About IPv6 ACLs

The following sections provide information about IPv6 ACLs.

## IPv6 ACL Overview

This topic provides an overview of IPv6 ACL.

An access control list (ACL) is a set of rules that are used to limit access to a particular interface. ACLs are configured on the device and applied to the management interface and to any of the dynamic interfaces.

You can also create a preauthentication ACL for web authentication. Such an ACL is used to allow certain types of traffic before authentication is complete.

IPv6 ACLs support the same options as IPv4 ACLs including source, destination, source, and destination ports.

## Supported ACLs

The switch supports three types of ACLs to filter the traffic:

- Port ACLs access-control traffic entering a Layer 2 interface. You can apply port ACLs to a Layer 2 interface in each direction to each access list type—IPv4 and MAC.
- Router ACLs access-control traffic routed between VLANs and are applied to Layer 3 interfaces in a specific direction (inbound or outbound).
- VLAN ACLs or VLAN maps are applied only to Layer 2 VLANs and impact bridged traffic only. You can use VLAN maps to filter traffic between devices in the same VLAN. VLAN maps are configured to provide access control based on Layer 3 addresses for IPv4. Unsupported protocols are access-controlled through MAC addresses using Ethernet ACEs. After a VLAN map is applied to a VLAN, all packets (routed or bridged) entering the VLAN are checked against the VLAN map. Packets can either enter the VLAN through a switch port or through a routed port after being routed.

## Types of ACL

The following sections provide information on the types of ACL:

### Per-User IPv6 ACL

For the per-user ACL, the full access control entries (ACE) as the text strings are configured on the Cisco Secure Access Control Server (Cisco Secure ACS).

### Filter ID IPv6 ACL

For the filter-Id ACL, the full ACEs and the `acl name(filter-id)` is configured on the device and only the `filter-id` is configured on the Cisco Secure ACS.

## Downloadable IPv6 ACL

For the downloadable ACL (dACL), all the full ACEs and the dacl name are configured only on the Cisco Secure ACS.

The Cisco Secure ACS sends the dacl name to the device in its `ACCESS-Accept` attribute, which takes the `dacl` name and sends the `dACL` name back to the Cisco Secure ACS for the ACEs, using the `ACCESS-request` attribute.

# Switch Stacks and IPv6 ACLs

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

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

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

# ACL Precedence

When VLAN maps, Port ACLs, and router ACLs are configured on the same switch, the filtering precedence, from greatest to least for ingress traffic is port ACL, VLAN map, and then router ACL. For egress traffic, the filtering precedence is router ACL, VLAN map, and then port ACL.

The following examples describe simple use cases:

- When both an input port ACL and a VLAN map are applied, incoming packets that are received on ports with a port ACL applied are filtered by the port ACL. Other packets are filtered by the VLAN map
- When an input router ACL and input port ACL exist in a switch virtual interface (SVI), incoming packets that are received on ports to which a port ACL is applied are filtered by the port ACL. Incoming 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, incoming packets that are received on the ports to which a port ACL is applied are filtered by the port ACL. Outgoing routed IP packets are filtered by the router ACL. Other packets are not filtered.
- When a VLAN map, input router ACL, and input port ACL exist in an SVI, incoming packets that are received on the ports to which a port ACL is applied are only filtered by the port ACL. Incoming routed IP packets received on other ports are filtered by both the VLAN map and the router ACL. Other packets are filtered only by the VLAN map.
- When a VLAN map, output router ACL, and input port ACL exist in an SVI, incoming packets that are received on the ports to which a port ACL is applied are only filtered by the port ACL. Outgoing routed IP packets are filtered by both the VLAN map and the router ACL. Other packets are filtered only by the VLAN map.

# VLAN Maps

VLAN ACLs or VLAN maps are used to control the network traffic within a VLAN. You can apply VLAN maps to all packets that are bridged within a VLAN in the switch or switch stack. VACLs are strictly for the

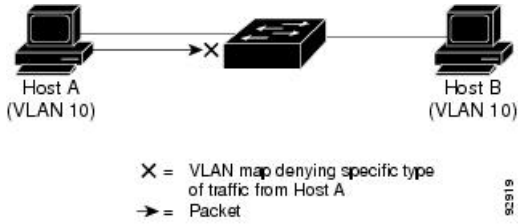
security packet filtering and for redirecting traffic to specific physical interfaces. VACLs are not defined by direction (ingress or egress).

All non-IP protocols are access-controlled through MAC addresses and Ethertype using MAC VLAN maps. (IP traffic is not access-controlled by MAC VLAN maps.) You can enforce VLAN maps only on packets going through the switch; you cannot enforce VLAN maps on traffic between hosts on a hub or on another switch that is connected to this switch.

With VLAN maps, forwarding of packets is permitted or denied, based on the action specified in the map.

***Figure 1: Using VLAN Maps to Control Traffic***

This figure shows how a VLAN map is applied to prevent a specific type of traffic from Host A in VLAN 10 from being forwarded. You can apply only one VLAN map to a VLAN.




# Interactions with Other Features and Switches

- 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 or switch 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 hardware memory is full, packets are dropped on the interface and an unload error message is logged.

  If the hardware memory is full, for any additional configured ACLs, packets are dropped to the CPU, and the ACLs are applied in software. When the hardware is full a message is printed to the console indicating the ACL has been unloaded and the packets will be dropped on the interface.

# How to Configure an IPv6 ACL

The following sections display information on how to configure an IPv6 ACL.

# Default Configuration for IPv6 ACLs

The default IPv6 ACL configuration is as follows:

```
Device# show access-lists preauth_ipv6_acl

IPv6 access list preauth_ipv6_acl (per-user)
permit udp any any eq domain sequence 10
permit tcp any any eq domain sequence 20
permit icmp any any nd-ns sequence 30
permit icmp any any nd-na sequence 40
permit icmp any any router-solicitation sequence 50
permit icmp any any router-advertisement sequence 60
permit icmp any any redirect sequence 70
permit udp any eq 547 any eq 546 sequence 80
permit udp any eq 546 any eq 547 sequence 90
deny ipv6 any any sequence 100
```

# Configuring IPv6 ACLs

To filter IPv6 traffic, perform this procedure:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **ipv6 access-list** {*list-name* \| **log-update threshold** \| **role-based** *list-name*}<br>**Example:**<br>`Device(config)# ipv6 access-list example_acl_list` | Defines an IPv6 ACL name, and enters IPv6 access list configuration mode. |
| **Step 4** | {**deny** \| **permit**} protocol {*source-ipv6-prefix/ \|prefix-length* \|**any threshold**\| **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*]<br>**Example:**<br>`Device(config-ipv6-acl)# permit tcp 2001:DB8:0300:0201::/32 eq telnet any` | Specifies permit or deny conditions for an IPv6 ACL.<br>• For protocol, enter the name or number of an IP: **ahp**, **esp**, **icmp**, **ipv6**, **pcp**, **stcp**, **tcp**, or **udp**, or an integer in the range 0 to 255 representing an IPv6 protocol number.<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 |

| | Command or Action | Purpose |
|---|---|---|
| | | hexadecimal and using 16-bit values between colons (see RFC 2373).<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 |

| | Command or Action | Purpose |
|---|---|---|
| | | list statement. The acceptable range is from 1 to 4,294,967,295.<br>• (Optional) Enter **time-range** name to specify the time range that applies to the deny or permit statement. |
| **Step 5** | {**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**]<br>**Example:**<br>`Device(config-ipv6-acl)# deny tcp host 2001:DB8:1::1 any log-input` | Specifies permit or deny conditions for an IPv6 ACL.<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 6** | **end**<br>**Example:**<br>`Device(config-ipv6-acl)# end` | Exits IPv6 access list configuration mode and returns to privileged EXEC mode. |
| **Step 7** | **show ipv6 access-list**<br>**Example:**<br>`Device# show ipv6 access-list` | Verifies that IPv6 ACLs are configured correctly. |

# Attaching an IPv6 ACL to an Interface

You can apply an ACL to outbound or inbound traffic on Layer 3 interfaces, or to inbound traffic on Layer 2 interfaces. You can also apply ACLs only to inbound management traffic on Layer 3 interfaces.

Follow these steps to control access to an interface.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *interface-id*<br>**Example:**<br>`Device(config)# interface gigabitethernet 1/0/1` | Identifies a Layer 2 interface (for port ACLs) or Layer 3 interface (for router ACLs) on which to apply an access list, and enters interface configuration mode. |
| **Step 4** | **no switchport**<br>**Example:**<br>`Device(config-if)# no switchport` | Returns the interface to the routed-interface status and erases all further Layer 2 configuration. |
| **Step 5** | **ipv6 address** *ipv6-address*<br>**Example:**<br>`Device(config-if)# ipv6 address 2001:DB8::1` | Configures an IPv6 address on a Layer 3 interface (for router ACLs). |
| **Step 6** | **ipv6 traffic-filter** *access-list-name* {**in** \| **out**}<br>**Example:**<br>`Device(config-if)# ipv6 traffic-filter acl1 in` | Applies the access list to incoming or outgoing traffic on the interface. |
| **Step 7** | **end**<br>**Example:**<br>`Device(config-ipv6-acl)# end` | Exits interface configuration mode and returns to privileged EXEC mode. |

# Configuring a VLAN Map

To create a VLAN map and apply it to one or more VLANs, perform these steps:

### Before you begin

Create the IPv6 ACL that you want to apply to the VLAN.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **vlan access-map** *name* [**number**]<br>**Example:**<br>`Device(config)# vlan access-map map_1 20` | Creates a VLAN map, and enters VLAN access-map command mode<br>VLAN map can have a name or (optionally) a number. The number is the sequence number of the entry within the map.<br>When you create VLAN maps with the same name, numbers are assigned sequentially in increments of 10. When modifying or deleting maps, you can enter the number of the map entry that you want to modify or delete.<br>VLAN maps do not use the specific permit or deny keywords. To deny a packet by using VLAN maps, create an ACL that would match the packet, and set the action to drop. A permit in the ACL counts as a match. A deny in the ACL means no match. |
| **Step 4** | **match** {**ip** \| **ipv6** \| **mac**} **address** {*name* \| *number*} [*name* \| *number*]<br>**Example:**<br>`Device(config-access-map)# match ipv6 address ip_net` | Matches the packet against one or more access lists. Note that packets are only matched against access lists of the correct protocol type. IP packets are matched against IP access lists. Non-IP packets are only matched against named MAC access lists.<br>**Note** If the VLAN map is configured with a match clause for a type of packet (IP or MAC) and the map action is drop, all packets that match the type are dropped. If the VLAN map has no match clause, and the configured action is drop, all IP and Layer 2 packets are dropped. |
| **Step 5** | Enter one of the following commands to specify an IP packet or a non-IP packet (with only a known MAC address) and to match the packet against one or more ACLs: | Sets the action for the map entry. |

| | Command or Action | Purpose |
|---|---|---|
| | • **action** { **forward**}<br>`Device(config-access-map)# action forward`<br>• **action** { **drop**}<br>`Device(config-access-map)# action drop` | |
| **Step 6** | **vlan filter** *mapname* **vlan-list** *list*<br>**Example:**<br>`Device(config)# vlan filter map 1 vlan-list 20-22` | Applies the VLAN map to one or more VLAN IDs.<br>The list can be a single VLAN ID (22), a consecutive list (10-22), or a string of VLAN IDs (12, 22, 30). Spaces around the comma and hyphen are optional. |
| **Step 7** | **end**<br>**Example:**<br>`Device(config)# end` | Exits global configuration mode and returns to privileged EXEC mode. |

# Applying a VLAN Map to a VLAN

To apply a VLAN map to one or more VLANs, perform these steps.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **vlan filter** *mapname* **vlan-list** *list*<br>**Example:**<br>`Device(config)# vlan filter map 1 vlan-list 20-22` | Applies the VLAN map to one or more VLAN IDs.<br>The list can be a single VLAN ID (22), a consecutive list (10-22), or a string of VLAN IDs (12, 22, 30). Spaces around the comma and hyphen are optional. |
| **Step 4** | **end**<br>**Example:**<br>`Device(config)# end` | Exits global configuration mode and returns to privileged EXEC mode. |

# Monitoring 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 shown in the table below:

***Table 1: show ACL commands***

| Command | Purpose |
|---|---|
| **show access-lists** | Displays all access lists configured on the switch. |
| **show ipv6 access-list** [*access-list-name*] | Displays all configured IPv6 access lists or the access list specified by name. |
| **show vlan access-map** [*map-name*] | Displays VLAN access map configuration. |
| **show vlan filter** [**access-map** *access-map* \| **vlan** *vlan-id*] | Displays the mapping between VACLs and VLANs. |

# Configuration Examples for IPv6 ACL

This following sections display configuration examples for IPv6 ACL:

## Example: Creating an IPv6 ACL

This example configures the IPv6 access list named IPv6-ACL. 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.

**Note** Logging is supported only on Layer 3 interfaces.

```
Device> enable
Device(config)# ipv6 access-list IPv6_ACL
Device(config-ipv6-acl)# deny tcp any any gt 5000
Device (config-ipv6-acl)# deny ::/0 lt 5000 ::/0 log
Device(config-ipv6-acl)# permit icmp any any
Device(config-ipv6-acl)# permit any any
Device(config-ipv6-acl)# end
```

## Example: Displaying IPv6 ACLs

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.

```
Device# 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.

```
Device# 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
```

# Example: Displaying VLAN Access Map Configuration

This is an example of the output from the **show vlan access-map** privileged EXEC command.

```
Device# show vlan access-map

Vlan access-map "m1"  10
  Match clauses:
    ipv6 address: ip2
  Action: drop
```

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.

```
Device# 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
```

# Feature History for IPv6 ACLs

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

| Release | Feature | Feature Information |
|---|---|---|
| Cisco IOS XE Everest 16.5.1a | IPv6 ACLs | You can filter IPv6 traffic by creating IPv6 ACLs and applying them to interfaces similar to how you create and apply IPv4 named ACLs. You can also create and apply input router ACLs to filter Layer 3 management traffic. |
| Cisco IOS XE Gibraltar 16.11.1 | IPv6 Downloadable ACLs | IPv6 dACLs are supported. |

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.