



CHAPTER 24

Configuring IP Version 6

Internet Protocol version 6 (IPv6), formerly called IPng (next generation), is the latest version of IP. IPv6 offers many advantages over the previous version of IP, including a larger address space. IPv6 has been available on other Cisco platforms; with the release of Cisco IOS release 12.2(28)SB, it is available on the Cisco 10000 series routers running the PRE2 processor.

For information about how to configure and use these IPv6 features on Cisco platforms, see the *Cisco IOS IPv6 Implementation Library*, located at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00805766e4.html.

This chapter the following information for the IPv6 feature:

- [Feature History for IPv6, page 24-1](#)
- [Supported Features, page 24-1](#)
- [Limitations for IPv6, page 24-3](#)
- [IPv6 Extended ACLs, page 24-4](#)

Feature History for IPv6

Cisco IOS Release	Description	Required PRE
12.2(28)SB	This feature was introduced on the Cisco 10000 series router.	PRE2
12.2(31)SB2	Support was added for the PRE3. Support was added for extended ACLs.	PRE3

Supported Features

The Cisco 10000 series routers support the following IPv6 PXF features:

- Coexistence with IPv4
- IPv6 Addressing
- IPv6 extension header. PXF handling of extension headers includes:
 - Diversion of packets with hop-by-hop extension header

- Ability to match on fragment and presence of routing headers
 - Skipping extension headers to get to layer 4 information
 - Flag setting to match on the “undetermined-transport” ACL flag
- IPv6 Internet Control Message Protocol (ICMP)
- IPv6 NDP
- IPv6 Layer 2 encapsulation:
 - Point to Point Protocol (PPP)
 - Multilink PPP
 - High-level Data Link Control (HDLC)
 - VLAN
 - Point-to-point Frame Relay
 - Point-to-point ATM
- IPv6 Routing:
 - Static
 - Routing Information Protocol (RIPng)
 - Open Shortest Path First (OSPFv3)
 - Border Gateway Protocol (BGP4+)
 - Intermediate System-to-Intermediate System (ISISv6)
- IPv6 Tunneling (manual and generic routing encapsulation (GRE))
 - Manually configured bi-directional IPv6-in-IPv4 GRE tunnels
 - Manually configured bi-directional IPv4-over-IPv4 tunnels
 Maximum of 1000 IPIP or GRE tunnels
- HA/ISSU coexistence; IPv6 support is RPR+
- IPv6 Unicast Forwarding

The Cisco 10000 series router maintains the following global (unless otherwise specified) IPv6-specific packet counters:

- forwarded—number of IPv6 packets forwarded
- no adjacency—number of IPv6 packets punted due to adj_index=0. Statistics per VCCI will be collected for this specific punt case (diversion cause).
- adj_discard— number of IPv6 packets dropped due to discard adjacency. Statistics per VCCI will be collected for this specific drop (column 5).
- adj_punt—number of IPv6 packets punted due to punt adjacency
- adj_glean— number of IPv6 packets punted due to glean adjacency
- adj_drop—number of IPv6 packets punted due to drop adjacency (RP generates ICMP and drops after that)
- adj_null—number of IPv6 packets punted due to null adjacency
- adj_receive—number of IPv6 packets punted due to receive adjacency
- adj_unknown—number of IPv6 packets punted due to unknown adjacency (e.g. 0x80)
- Strict Reverse Path Forwarding (RPF)

RPF strict check mode verifies that the source IP address exists in the FIB table and verifies that the source IP address is reachable through the input port

- Security ACLs

For IPv6, ACEs include the following new fields:

- Flow Label
- Presence of Routing Header
- “Undetermined Transport”

- QoS

QoS matching is performed only on the following subset of fields, which are common to IPv4 and IPv6:

- dscp/precedence
- access group (matches only on ACE entries common to IPv4 and IPv6)
- class
- qos group
- mpls
- input if
- l2 cos
- discard class

The **match protocol** command now includes the **ipv6** keyword to specify this protocol as a matching criterion. The **match ip dscp** and **match ip precedence** commands apply only to IPv4 traffic. The **match dscp** and **match precedence** commands apply to both IPv4 and IPv6 traffic.

For marking packets, the **set ip dscp** and **set ip precedence** commands have been changed to **set dscp** and **set precedence**. They now apply to both IPv4 and IPv6 traffic.

- ICMP handling and generation are performed on the route processor and are not handled in PFX

Limitations for IPv6

Not all types of IPv6 Tunneling are supported on the Cisco 10000 routers with this release. Among those not supported are the following:

- Automatic 6to4
- ISATAP
- Automatic IPv4-compatible
- IPv6 over L2TPv3
- 6over4 (RFC 2529)
- IPv6 in IPv6 GRE
- IPv6 over UTI

The following security ACL features are not supported for IPv6:

- Incremental compilation (The Cisco 10000 routers use pre-compiled ACLs.)
- Single-step classification

- ACL logging
- Time-based ACLs
- Reflexive ACLs
- Receive Path ACLs
- MiniACLs

QoS matching is not provided on the following two fields, which are IPv6-specific:

- IPv6 src/dst address
- IPv6 ACL

IPv6 Extended ACLs

Access lists determine what traffic is blocked and what traffic is forwarded at router interfaces and allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Each access list has an implicit deny statement at the end. IPv6 ACLs are defined and their deny and permit conditions are set using the **ipv6 access-list** command with the **deny** and **permit** keywords in global configuration mode.

In Cisco IOS Release 12.2(31)SB2 and later releases, the standard IPv6 ACL functionality is extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control (functionality similar to extended ACLs in IPv4).

Prerequisites

In Cisco IOS Release 12.2(13)T and 12.0(23)S or later releases, for backward compatibility, the **ipv6 access-list** command with the **deny** and **permit** keywords in global configuration mode is still supported; however, an IPv6 ACL defined with deny and permit conditions in global configuration mode is translated to IPv6 access list configuration mode. See the [“Create and Apply IPv6 ACL: Examples”](#) section for an example of a translated IPv6 ACL configuration.

Restrictions

IPv6 ACLs are defined by a unique name (IPv6 does not support numbered ACLs). An IPv4 ACL and an IPv6 ACL cannot share the same name.

- Each IPv6 ACL contains implicit permit rules to enable IPv6 neighbor discovery. These rules can be overridden by the user by placing a **deny ipv6 any any** statement within an ACL. The IPv6 neighbor discovery process makes use of the IPv6 network layer service; therefore, by default, IPv6 ACLs implicitly allow IPv6 neighbor discovery packets to be sent and received on an interface. In IPv4, the Address Resolution Protocol (ARP), which is equivalent to the IPv6 neighbor discovery process, makes use of a separate data link layer protocol; therefore, by default, IPv4 ACLs implicitly allow ARP packets to be sent and received on an interface.

Configuring IPv6 Traffic Filtering

To enable IPv6 traffic filtering, you must perform the following steps:

1. Create an IPv6 ACL
2. Configure the IPv6 ACL to pass or block traffic
3. Apply the IPv6 ACL to an interface

Creating and Configuring the IPv6 ACL

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list *access-list-name***
4. **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*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**reflect** *name*] [**timeout** *value*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
or
deny *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*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	• Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

Command or Action	Purpose
Step 3 <pre>ipv6 access-list access-list-name</pre> Example: <pre>Router(config)# ipv6 access-list outbound</pre>	Defines an IPv6 ACL and enters IPv6 access list configuration mode. The router prompt changes to Router(config-ipv6-acl)#. The <i>access-list name</i> argument specifies the name of the IPv6 ACL. IPv6 ACL names cannot contain a space or quotation mark, or begin with a numeral.
Step 4 <pre>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]] [dest-option-type [doh-number doh-type]] [dscp value] [flow-label value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number mh-type]] [reflect name [timeout value]] [routing] [routing-type routing-number] [sequence value] [time-range name]</pre> or <pre>deny 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]] [dest-option-type [doh-number doh-type]] [dscp value] [flow-label value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number mh-type]] [reflect name [timeout value]] [routing] [routing-type routing-number] [sequence value] [time-range name] [undetermined-transport]</pre> Example: <pre>Router(config-ipv6-acl)# permit tcp 2001:0DB8:0300:0201::/32 eq telnet any reflect reflectout</pre> Example: <pre>Router(config-ipv6-acl)# deny tcp host 2001:0db8:1::1 any log-input</pre>	Specifies permit or deny conditions for an IPv6 ACL. - The <i>protocol</i> argument specifies the name or number of an Internet protocol. It can be one of the keywords ahp, esp, icmp, ipv6, pcp, setp, tcp, or udp, or an integer in the range from 0 to 255 representing an IPv6 protocol number. - The <i>source-ipv6-prefix/prefix-length</i> and <i>destination-ipv6-prefix/prefix-length</i> arguments specify the source and destination IPv6 network or class of networks about which to set permit conditions. Note These arguments must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. - The any keyword is an abbreviation for the IPv6 prefix ::/0. - The host source-ipv6-address keyword and argument specify the source IPv6 host address about which to set permit conditions. - The <i>source-ipv6-address</i> argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. For information on supported arguments and keywords, see the permit and deny commands in the <i>IPv6 for Cisco IOS Command Reference</i> document.

Applying the IPv6 ACL to an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type number**
4. **ipv6 traffic-filter access-list-name {in | out}**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0	Specifies the interface type and number, and enters interface configuration mode.
Step 4	ipv6 traffic-filter <i>access-list-name</i> { in out } Example: Router(config-if)# ipv6 traffic-filter outbound out	Applies the specified IPv6 access list to the interface specified in the previous step. - The in keyword filters incoming IPv6 traffic on the specified interface. - The out keyword filters outgoing IPv6 traffic on the specified interface.

Verifying IPv6 ACLs

In the following example, the **show ipv6 access-list** command is used to verify that IPv6 ACLs are configured correctly:

```
Router> show ipv6 access-list
```

```
IPv6 access list inbound
  permit tcp any any eq bgp reflect tcptraffic (8 matches) sequence 10
  permit tcp any any eq telnet reflect tcptraffic (15 matches) sequence 20
  permit udp any any reflect udptraffic sequence 30

IPv6 access list tcptraffic (reflexive) (per-user)
  permit tcp host 2001:0DB8:1::32 eq bgp host 2001:0DB8:2::32 eq 11000 timeout 300 (time
    left 243) sequence 1
  permit tcp host 2001:0DB8:1::32 eq telnet host 2001:0DB8:2::32 eq 11001 timeout 300
    (time left 296) sequence 2

IPv6 access list outbound
  evaluate udptraffic
  evaluate tcptraffic
```

**Note**

For a description of each output display field, see the **show ipv6 access-list** command in the *IPv6 for Cisco IOS Command Reference* document.

Create and Apply IPv6 ACL: Examples

The following example configures two IPv6 ACLs named OUTBOUND and INBOUND and applies both ACLs to outbound and inbound traffic on Ethernet interface 0. The first and second permit entries in the OUTBOUND list permit all TCP and User Datagram Protocol (UDP) packets from network 2001:0DB8:0300:0201::/32 to exit out of Ethernet interface 0. The entries also configure the temporary IPv6 reflexive ACL named REFLECTOUT to filter returning (incoming) TCP and UDP packets on Ethernet interface 0. The first deny entry in the OUTBOUND list keeps all packets from the network fec0:0:0:0201::/64 (packets that have the site-local prefix fec0:0:0:0201 as the first 64 bits of their source IPv6 address) from exiting out of Ethernet interface 0.

The **evaluate** command in the INBOUND list applies the temporary IPv6 reflexive ACL named REFLECTOUT to inbound TCP and UDP packets on Ethernet interface 0. When outgoing TCP or UDP packets are permitted on Ethernet interface 0 by the OUTBOUND list, the INBOUND list uses the REFLECTOUT list to match (evaluate) the returning (incoming) TCP and UDP packets.

```
ipv6 access-list OUTBOUND
 permit tcp 2001:0DB8:0300:0201::/32 any reflect REFLECTOUT
 permit udp 2001:0DB8:0300:0201::/32 any reflect REFLECTOUT
 deny fec0:0:0:0201::/64 any

ipv6 access-list INBOUND
 evaluate REFLECTOUT

interface ethernet 0
 ipv6 traffic-filter OUTBOUND out
 ipv6 traffic-filter INBOUND in
```

**Note**

Given that a **permit any any** statement is not included as the last entry in the OUTBOUND or INBOUND ACL, only TCP and UDP packets matching the configured permit entries in the ACL and ICMP packets matching the implicit permit conditions in the ACL are permitted out of and in to Ethernet interface 0 (the implicit deny all condition at the end of the ACL denies all other packet types on the interface).

The following example configures HTTP access to be restricted to certain hours during the day, and to log any activity outside of the permitted hours.

```
time-range lunchtime
 periodic weekdays 12:00 to 13:00

ipv6 access-list OUTBOUND
 permit tcp any any eq www time-range lunchtime
 deny tcp any any eq www log-input
 permit tcp 2001:0DB8::/32 any
 permit udp 2001:0DB8::/32 any
```