



Implementing Traffic Filters and Firewalls for IPv6 Security

First Published: June 7, 2001

Last Updated: August 18, 2008

This module describes how to configure Cisco IOS IPv6 traffic filter and firewall features for your Cisco networking devices. These security features can protect your network from degradation or failure and also from data loss or compromised security resulting from intentional attacks and from unintended but damaging mistakes by well-meaning network users.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 for Implementing Traffic Filters and Firewalls for IPv6 Security” section on page 31](#).

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

Contents

- [Prerequisites for Implementing Traffic Filters and Firewalls for IPv6 Security, page 2](#)
- [Restrictions for Implementing Traffic Filters and Firewalls for IPv6 Security, page 2](#)
- [Information About Implementing Traffic Filters and Firewalls for IPv6 Security, page 2](#)
- [How to Implement Traffic Filters and Firewalls for IPv6 Security, page 5](#)
- [Configuration Examples for Implementing Traffic Filters and Firewalls for IPv6 Security, page 26](#)
- [Additional References, page 29](#)



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2001–2009 Cisco Systems, Inc. All rights reserved.

- [Feature Information for Implementing Traffic Filters and Firewalls for IPv6 Security, page 31](#)

Prerequisites for Implementing Traffic Filters and Firewalls for IPv6 Security

You should be familiar with IPv6 addressing and basic configuration. Refer to the [Implementing IPv6 Addressing and Basic Connectivity](#) module for more information.

Restrictions for Implementing Traffic Filters and Firewalls for IPv6 Security

Cisco IOS Release 12.2(2)T through Cisco IOS Release 12.2(13)T and Cisco IOS Release 12.0(22)S and later releases support only standard IPv6 access control list (ACL) functionality. In Cisco IOS Release 12.0(23)S and 12.2(13)T or 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).

Information About Implementing Traffic Filters and Firewalls for IPv6 Security

To implement security features for IPv6, you need to understand the following concepts:

- [Access Control Lists for IPv6 Traffic Filtering, page 2](#)
- [Cisco IOS Firewall for IPv6, page 3](#)

Access Control Lists for IPv6 Traffic Filtering

The standard ACL functionality in IPv6 is similar to standard ACLs in IPv4. 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 12.0(23)S and 12.2(13)T or 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).

IPv6 ACL Extensions for IPsec Authentication Header

This feature provides the ability to match on the upper layer protocol (ULP) (for example, TCP, User Datagram Protocol [UDP], ICMP, SCTP) regardless of whether an authentication header (AH) is present or absent.

TCP or UDP traffic can be matched to the upper-layer protocol (ULP) (for example, TCP, UDP, ICMP, SCTP) if an AH is present or absent. Before this feature was introduced, this function was only available if an AH was absent.

This feature introduces the keyword **auth** to the **permit** and **deny** commands. The **auth** keyword allows matching traffic against the presence of the authentication header in combination with the specified protocol; that is, TCP or UDP.

IPv6 traffic can be matched to a ULP when an AH header is present. To perform this function, enter the **ahp** option for the *protocol* argument when using the **permit** or **deny** command.

Access Class Filtering in IPv6

Filtering incoming and outgoing connections to and from the router based on an IPv6 ACL is performed using the **ipv6 access-class** command in line configuration mode. The **ipv6 access-class** command is similar to the **access-class** command, except the IPv6 ACLs are defined by a name. If the IPv6 ACL is applied to inbound traffic, the source address in the ACL is matched against the incoming connection source address and the destination address in the ACL is matched against the local router address on the interface. If the IPv6 ACL is applied to outbound traffic, the source address in the ACL is matched against the local router address on the interface and the destination address in the ACL is matched against the outgoing connection source address. We recommend that identical restrictions are set on all the virtual terminal lines because a user can attempt to connect to any of them.

Cisco IOS Firewall for IPv6

The Cisco IOS Firewall feature provides advanced traffic filtering functionality as an integral part of a network's firewall. Cisco IOS Firewall for IPv6 enables you to implement Cisco IOS Firewall in IPv6 networks. Cisco IOS Firewall coexists with Cisco IOS Firewall for IPv4 networks and is supported on all dual-stack routers.

Cisco IOS Firewall for IPv6 features are as follows:

- Fragmented packet inspection—The fragment header is used to trigger fragment processing. Cisco IOS Firewall virtual fragment reassembly (VFR) examines out-of-sequence fragments and switches the packets into correct order, examines the number of fragments from a single IP given a unique identifier (Denial of Service [DoS] attack), and performs virtual reassembly to move packets to upper-layer protocols.
- IPv6 DoS attack mitigation—Mitigation mechanisms have been implemented in the same fashion as for IPv4 implementation, including SYN half-open connections.
- Tunneled packet inspection—Tunneled IPv6 packets terminated at a Cisco IOS firewall router can be inspected by the Cisco IOS Firewall for IPv6.
- Stateful packet inspection—The feature provides stateful packet inspection of TCP, UDP, Internet Control Message Protocol version 6 (ICMPv6), and FTP sessions.
- Stateful inspection of packets originating from the IPv4 network and terminating in an IPv6 environment—This feature uses IPv4-to-IPv6 translation services.
- Interpretation or recognition of most IPv6 extension header information—The feature provides IPv6 extension header information including routing header, hop-by-hop options header, and fragment header is interpreted or recognized.
- Port-to-application mapping (PAM)—Cisco IOS Firewall for IPv6 includes PAM.

PAM in Cisco IOS Firewall for IPv6

PAM allows you to customize TCP or UDP port numbers for network services or applications. PAM uses this information to support network environments that run services using ports that are different from the registered or well-known ports associated with an application.

Using the port information, PAM establishes a table of default port-to-application mapping information at the firewall. The information in the PAM table enables Context-based Access Control (CBAC) supported services to run on nonstandard ports. CBAC is limited to inspecting traffic using only the well-known or registered ports associated with an application, whereas PAM allows network administrators to customize network access control for specific applications and services.

PAM also supports host- or subnet-specific port mapping, which allows you to apply PAM to a single host or subnet using standard ACLs. Host- or subnet-specific port mapping is done using standard ACLs.

Cisco IOS Firewall Alerts, Audit Trails, and System Logging

Cisco IOS Firewall generates real-time alerts and audit trails based on events tracked by the firewall. Enhanced audit trail features use system logging to track all network transactions; to record time stamps, source host, destination host, and ports used; and to record the total number of transmitted bytes for advanced, session-based reporting. Real-time alerts send system logging error messages to central management consoles when the system detects suspicious activity. Using Cisco IOS Firewall inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. For example, if you want to generate audit trail information for TCP traffic, you can specify the generation of this information in the Cisco IOS Firewall rule that defines TCP inspection.

The Cisco IOS Firewall provides audit trail messages to record details about inspected sessions. Audit trail information is configurable on a per-application basis using the CBAC inspection rules. To determine which protocol was inspected, use the port number associated with the responder. The port number appears immediately after the address.

IPv6 Packet Inspection

The following header fields are all used for IPv6 inspection—traffic class, flow label, payload length, next header, hop limit, and source or destination address. For further information on and descriptions of the IPv6 header fields, see RFC 2474.

Tunneling Support

IPv6 packets tunneled in IPv4 are not inspected. If a tunnel terminates on a router, and IPv6 traffic exiting the tunnel is nonterminating, then the traffic is inspected.

Virtual Fragment Reassembly

When VFR is enabled, VFR processing begins after ACL input lists are checked against incoming packets. The incoming packets are tagged with the appropriate VFR information.

Cisco IOS Firewall Restrictions

Cisco IOS Intrusion Detection System (IDS) is not supported for IPv6.

How to Implement Traffic Filters and Firewalls for IPv6 Security

The tasks in the following sections explain how to configure security features for IPv6:

- [Configuring IPv6 Traffic Filtering, page 5](#)
- [Controlling Access to a vty, page 8](#)
- [Configuring TCP or UDP Matching, page 11](#)
- [Creating an IPv6 ACL for Traffic Filtering in Cisco IOS Release 12.2\(11\)T, 12.0\(22\)S, or Earlier Releases, page 12](#)
- [Configuring the Cisco IOS Firewall for IPv6, page 14](#)
- [Configuring the Cisco IOS Firewall for IPv6, page 14](#)
- [Verifying IPv6 Security Configuration and Operation, page 19](#)
- [Troubleshooting IPv6 Security Configuration and Operation, page 21](#)

Configuring IPv6 Traffic Filtering

The following sections describe how enable IPv6 traffic filtering:

- [Creating and Configuring an IPv6 ACL for Traffic Filtering, page 5](#)
- [Applying the IPv6 ACL to an Interface, page 7](#)

Restrictions

- If you are running Cisco IOS Release 12.2(13)T, 12.0(23)S, or later releases, proceed to the [“Creating and Configuring an IPv6 ACL for Traffic Filtering”](#) section. If you are running Cisco IOS Release 12.2(11)T, 12.0(22)S, or earlier releases, proceed to the [“Creating an IPv6 ACL for Traffic Filtering in Cisco IOS Release 12.2\(11\)T, 12.0\(22\)S, or Earlier Releases”](#) section.
- 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.

Creating and Configuring an IPv6 ACL for Traffic Filtering

This part describes how to configure your networking devices to filter traffic, function as a firewall, or detect potential viruses. The following task explains how to create an IPv6 ACL and configure the IPv6 ACL to filter traffic in Cisco IOS Release 12.2(13)T and 12.0(23)S or later releases.

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 [“Creating and Applying IPv6 ACLs: Examples”](#) section for an example of a translated IPv6 ACL configuration.

Restrictions

- 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.
- Time-based and reflexive ACLs are not supported for IPv4 or IPv6 on the Cisco 12000 series platform. The **reflect**, **timeout**, and **time-range** keywords of the **permit** command in IPv6 are excluded on the Cisco 12000 series.

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* | **auth** } [*operator* [*port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth** } [*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* | **auth** } [*operator* [*port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth** } [*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 Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>ipv6 access-list <i>access-list-name</i></p> <p>Example: Router(config)# ipv6 access-list outbound</p>	<p>Defines an IPv6 ACL, and enters IPv6 access list configuration mode.</p> <ul style="list-style-type: none"> 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	<p>permit protocol {<i>source-ipv6-prefix/prefix-length</i> any host source-ipv6-address auth} [<i>operator</i> [<i>port-number</i>]] {<i>destination-ipv6-prefix/prefix-length</i> any host destination-ipv6-address auth} [<i>operator</i> [<i>port-number</i>]] [dest-option-type [<i>doh-number</i> <i>doh-type</i>]] [dscp <i>value</i>] [flow-label <i>value</i>] [fragments] [log] [log-input] [mobility] [mobility-type [<i>mh-number</i> <i>mh-type</i>]] [reflect name [<i>timeout value</i>]] [routing] [routing-type <i>routing-number</i>] [sequence <i>value</i>] [time-range <i>name</i>]</p> <p>or</p> <p>deny protocol {<i>source-ipv6-prefix/prefix-length</i> any host source-ipv6-address auth} [<i>operator</i> [<i>port-number</i>]] {<i>destination-ipv6-prefix/prefix-length</i> any host destination-ipv6-address auth} [<i>operator</i> [<i>port-number</i>]] [dest-option-type [<i>doh-number</i> <i>doh-type</i>]] [dscp <i>value</i>] [flow-label <i>value</i>] [fragments] [log] [log-input] [mobility] [mobility-type [<i>mh-number</i> <i>mh-type</i>]] [routing] [routing-type <i>routing-number</i>] [sequence <i>value</i>] [time-range <i>name</i>] [undetermined-transport]</p> <p>Example: Router(config-ipv6-acl)# permit tcp 2001:0DB8:0300:0201::/32 eq telnet any reflect reflectout</p> <p>or</p> <p>Example: Router(config-ipv6-acl)# deny tcp host 2001:0db8:1::1 any log-input</p>	<p>Specifies permit or deny conditions for an IPv6 ACL.</p>

Applying the IPv6 ACL to an Interface

This task describes how to apply the IPv6 ACL to an interface in Cisco IOS Release 12.2(13)T and 12.0(23)S or later releases.

SUMMARY STEPS

- enable**
- configure terminal**
- interface** *type number*

4. `ipv6 traffic-filter access-list-name {in | out}`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# configure terminal	Enters global configuration mode.
Step 3	<code>interface type number</code> Example: Router(config)# interface ethernet 0	Specifies the interface type and number, and enters interface configuration mode.
Step 4	<code>ipv6 traffic-filter access-list-name {in out}</code> Example: Router(config-if)# ipv6 traffic-filter outbound out	Applies the specified IPv6 access list to the interface specified in the previous step.

Controlling Access to a vty

The following tasks explain how to restrict access to a vty on a router:

- [Creating an IPv6 ACL to Provide Access Class Filtering, page 8](#)
- [Applying an IPv6 ACL to the Virtual Terminal Line, page 10](#)

Creating an IPv6 ACL to Provide Access Class Filtering

The following task explains how to control access to a vty on a router by creating an IPv6 ACL to provide access class filtering.

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 | auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address | auth} [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 | auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address | auth} [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 Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<pre>ipv6 access-list access-list-name</pre> <p>Example: Router(config)# ipv6 access-list cisco</p>	Defines an IPv6 ACL, and enters IPv6 access list configuration mode.
Step 4	<pre>permit protocol {source-ipv6-prefix/prefix-length any host source-ipv6-address auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length any host destination-ipv6-address auth} [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> <p>or</p> <pre>deny protocol {source-ipv6-prefix/prefix-length any host source-ipv6-address auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length any host destination-ipv6-address auth} [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]</pre> <p>Example: Router(config-ipv6-acl)# permit ipv6 host 2001:0DB8:0:4::32 any eq telnet</p> <p>or</p> <p>Example: Router(config-ipv6-acl)# deny ipv6 host 2001:0DB8:0:6::6/32 any</p>	Specifies permit or deny conditions for an IPv6 ACL.

Applying an IPv6 ACL to the Virtual Terminal Line

After you have created the IPv6 ACL for access class filtering, you must apply it to a specified virtual terminal line. The following task describes how to apply the ACL to the virtual terminal line.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line [aux | console | tty | vty] line-number [ending-line-number]**
4. **ipv6 access-class ipv6-access-list-name {in | out}**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	line [aux console tty vty] line-number [ending-line-number] Example: Router(config)# line vty 0 4	Identifies a specific line for configuration and enters line configuration mode. <ul style="list-style-type: none"> In this example, the vty keyword is used to specify the virtual terminal lines for remote console access.
Step 4	ipv6 access-class ipv6-access-list-name {in out} Example: Router(config-line)# ipv6 access-class cisco in	Filters incoming and outgoing connections to and from the router based on an IPv6 ACL.

Configuring TCP or UDP Matching

TCP or UDP traffic can be matched to the ULP (for example, TCP, UDP, ICMP, SCTP) if an AH is present or absent. Before this feature was introduced, this function was only available if an AH was absent.

Use of the keyword **auth** with the **permit icmp** and **deny icmp** commands allows TCP or UDP traffic to be matched to the ULP if an AH is present. TCP or UDP traffic without an AH will not be matched.

IPv6 traffic can be matched to a ULP when an AH header is present. To perform this function, enter the **ahp** option for the *protocol* argument when using the **permit** or **deny** command.

This task shows how to allow TCP or UDP traffic to be matched to the ULP if an AH is present.

SUMMARY STEPS

- enable**
 - configure terminal**
 - ipv6 access-list access-list-name**
 - permit icmp auth**
- or
- deny icmp auth**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router# <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# <code>configure terminal</code>	Enters global configuration mode.
Step 3	<code>ipv6 access-list access-list-name</code> Example: Router(config)# <code>ipv6 access-list list1</code>	Defines an IPv6 access list and places the router in IPv6 access list configuration mode.
Step 4	<code>permit icmp auth</code> or <code>deny icmp auth</code> Example: Router(config-ipv6-acl)# <code>permit icmp auth</code>	Specifies permit or deny conditions for an IPv6 ACL using the auth keyword, which is used to match against the presence of the AH.

Creating an IPv6 ACL for Traffic Filtering in Cisco IOS Release 12.2(11)T, 12.0(22)S, or Earlier Releases

The following tasks describe how to create and apply ACLs in Cisco IOS Release 12.2(11)T, 12.0(22)S, or earlier releases.

- [Creating an IPv6 ACL in Cisco IOS Release 12.2\(11\)T, 12.0\(22\)S, or Earlier Releases, page 12](#)
- [Applying the IPv6 ACL to an Interface in Cisco IOS Release 12.2\(11\)T, 12.0\(22\)S, or Earlier Releases, page 13](#)

Creating an IPv6 ACL in Cisco IOS Release 12.2(11)T, 12.0(22)S, or Earlier Releases

This task explains how to create an IPv6 ACL and configure the IPv6 ACL to pass or block traffic in Cisco IOS Release 12.2(11)T, 12.0(22)S, or earlier releases.

Restrictions

- The *source-ipv6-prefix* argument filters traffic by packet source address, and the *destination-ipv6-prefix* argument filters traffic by packet destination address.

- The Cisco IOS software compares an IPv6 prefix against the permit and deny condition statements in the access list. Every IPv6 access list, including access lists that do not have any permit and deny condition statements, has an implicit deny any any statement as its last match condition. The priority or sequence value applied to each condition statement dictates the order in which the statement is applied in the access list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name* {**permit** | **deny**} {*source-ipv6-prefix/prefix-length* | **any**} {*destination-ipv6-prefix/prefix-length* | **any**} [**priority value**]

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	ipv6 access-list <i>access-list-name</i> { permit deny } { <i>source-ipv6-prefix/prefix-length</i> any } { <i>destination-ipv6-prefix/prefix-length</i> any } [priority value] Example: Router(config)# ipv6 access-list list2 deny fec0:0:0:2::/64 any	Creates an IPv6 ACL and sets deny or permit conditions for the ACL.

Applying the IPv6 ACL to an Interface in Cisco IOS Release 12.2(11)T, 12.0(22)S, or Earlier Releases

This task describes how to apply the IPv6 ACL to an interface in Cisco IOS Release 12.2(11)T, 12.0(22)S, or earlier releases.

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. <ul style="list-style-type: none"> 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 {in out}</i> Example: Router(config-if)# ipv6 traffic-filter list2 out	Applies the specified IPv6 access list to the interface specified in the previous step.

Configuring the Cisco IOS Firewall for IPv6

This task shows how to configure the Cisco IOS Firewall for IPv6 environments. This configuration scenario uses both packet inspection and ACLs.

SUMMARY STEPS

- enable**
- configure terminal**
- ipv6 unicast-routing**
- ipv6 inspect name** *inspection-name protocol [alert {on | off}] [audit-trail {on | off}] [timeout seconds]*
- interface** *type number*
- ipv6 address** *{ipv6-address/prefix-length | prefix-name sub-bits/prefix-length}*
- ipv6 enable**
- ipv6 traffic-filter** *access-list-name {in | out}*
- ipv6 inspect** *inspect-name*
- ipv6 access-list** *access-list-name*
- permit** *protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address | auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address | auth} [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 | auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address | auth} [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 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	ipv6 unicast-routing Example: Router(config)# ipv6 unicast-routing	Enables IPv6 unicast routing.
Step 4	ipv6 inspect name inspection-name protocol [alert {on off}] [audit-trail {on off}] [timeout seconds] Example: Router(config)# ipv6 inspect name ipv6_test icmp timeout 60	Defines a set of IPv6 inspection rules for the firewall.
Step 5	interface type number Example: Router(config)# interface FastEthernet0/0	Specifies the interface on which the inspection will occur.
Step 6	ipv6 address {ipv6-address/prefix-length prefix-name sub-bits/prefix-length} Example: Router(config-if)# ipv6 address 3FFE:C000:0:7::/64 eui-64	Provides the address for the inspection interface.
Step 7	ipv6 enable Example: Router(config-if)# ipv6 enable	Enables IPv6 routing. Note This step is optional if the IPv6 address is specified in step 6.

	Command or Action	Purpose
Step 8	<p>ipv6 traffic-filter <i>access-list-name</i> {in out}</p> <p>Example: Router(config-if)# ipv6 traffic-filter outbound out</p>	Applies the specified IPv6 access list to the interface specified in the previous step.
Step 9	<p>ipv6 inspect <i>inspection-name</i> {in out}</p> <p>Example: Router(config)#ipv6 inspect ipv6_test in</p>	Applies the set of inspection rules.
Step 10	<p>ipv6 access-list <i>access-list-name</i></p> <p>Example: Router(config)# ipv6 access-list outbound</p>	Defines an IPv6 ACL and enters IPv6 access list configuration mode. The router prompt changes to Router(config-ipv6-acl)#.
Step 11	<p>permit protocol {<i>source-ipv6-prefix/prefix-length</i> any host source-ipv6-address auth} [<i>operator</i> [<i>port-number</i>]] {<i>destination-ipv6-prefix/prefix-length</i> any host destination-ipv6-address auth} [<i>operator</i> [<i>port-number</i>]] [dest-option-type [<i>doh-number</i> <i>doh-type</i>]] [dscp <i>value</i>] [flow-label <i>value</i>] [fragments] [log] [log-input] [mobility] [mobility-type [<i>mh-number</i> <i>mh-type</i>]] [reflect name [<i>timeout value</i>]] [routing] [routing-type <i>routing-number</i>] [sequence <i>value</i>] [time-range <i>name</i>]</p> <p>or</p> <p>deny protocol {<i>source-ipv6-prefix/prefix-length</i> any host source-ipv6-address auth} [<i>operator</i> [<i>port-number</i>]] {<i>destination-ipv6-prefix/prefix-length</i> any host destination-ipv6-address auth} [<i>operator</i> [<i>port-number</i>]] [dest-option-type [<i>doh-number</i> <i>doh-type</i>]] [dscp <i>value</i>] [flow-label <i>value</i>] [fragments] [log] [log-input] [mobility] [mobility-type [<i>mh-number</i> <i>mh-type</i>]] [routing] [routing-type <i>routing-number</i>] [sequence <i>value</i>] [time-range <i>name</i>] [undetermined-transport]</p> <p>Example: Router(config-ipv6-acl)# permit tcp 2001:0DB8:0300:0201::/32 any reflect reflectout or</p> <p>Example: Router(config-ipv6-acl)# deny tcp fec0:0:0:0201::/64 any</p>	Specifies permit or deny conditions for an IPv6 ACL.

Configuring PAM for IPv6

The tasks in the following sections explain how to configure PAM for IPv6.

- [Creating an IPv6 Access Class Filter for PAM, page 17](#)
- [Applying the IPv6 Access Class Filter to PAM, page 18](#)

Creating an IPv6 Access Class Filter for PAM

The following task explains how to create an IPv6 access class filter to use in PAM configuration:

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* | **auth** } [*operator* [*port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth** } [*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* | **auth** } [*operator* [*port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth** } [*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 Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<pre>ipv6 access-list access-list-name</pre> <p>Example: Router(config)# ipv6 access-list outbound</p>	Defines an IPv6 ACL and enters IPv6 access list configuration mode. The router prompt changes to Router(config-ipv6-acl)#.
Step 4	<pre>permit protocol {source-ipv6-prefix/prefix-length any host source-ipv6-address auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length any host destination-ipv6-address auth} [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> <p>or</p> <pre>deny protocol {source-ipv6-prefix/prefix-length any host source-ipv6-address auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length any host destination-ipv6-address auth} [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]</pre> <p>Example: Router(config-ipv6-acl)# permit tcp 2001:0DB8:0300:0201::/32 any reflect reflectout or</p> <p>Example: Router(config-ipv6-acl)# deny tcp fec0:0:0:0201::/64 any</p>	Specifies permit or deny conditions for an IPv6 ACL.

Applying the IPv6 Access Class Filter to PAM

Once you have created an IPv6 access class filter, use the following task to apply the filter to PAM.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 port-map application-name port port-num [list acl-name]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 port-map <i>application-name</i> port <i>port-num</i> [list <i>acl-name</i>] Example: Router(config)# ipv6 port-map ftp port 8090 list PAMACL	Establishes PAM for the system.

Verifying IPv6 Security Configuration and Operation

This task explains how to display information to verify the configuration and operation of IPv6 security options. Use the following commands as needed to verify configuration and operation.

SUMMARY STEPS

- show crypto ipsec sa** [**map** *map-name* | **address** | **identity** | **interface** *interface-type* *interface-number* | **peer** [**vrf** *fvr-name*] **address** | **vrf** *ivr-name* | **ipv6** [*interface-type* *interface-number*]] [**detail**]
- show crypto isakmp peer** [**config** | **detail**]
- show crypto isakmp profile**
- show crypto isakmp sa** [**active** | **standby** | **detail** | **nat**]
- show ipv6 access-list** [*access-list-name*]
- show ipv6 inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all**}
- show ipv6 port-map** [*application* | **port** *port-number*]
- show ipv6 prefix-list** [**detail** | **summary**] [*list-name*]
- show ipv6 virtual-reassembly interface** *interface-type*
- show logging** [**slot** *slot-number* | **summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>show crypto ipsec sa [map map-name address identity interface interface-type interface-number peer [vrf fvrf-name] address vrf ivrf-name ipv6 [interface-type interface-number]] [detail]</pre> <p>Example: Router# show crypto ipsec sa ipv6</p>	Displays the settings used by current SAs.
Step 2	<pre>show crypto isakmp peer [config detail]</pre> <p>Example: Router# show crypto isakmp peer</p>	Displays peer descriptions.
Step 3	<pre>show crypto isakmp profile</pre> <p>Example: Router# show crypto isakmp profile</p>	Lists all the ISAKMP profiles that are defined on a router.
Step 4	<pre>show crypto isakmp sa [active standby detail nat]</pre> <p>Example: Router# show crypto isakmp sa</p>	Displays current IKE SAs.
Step 5	<pre>show ipv6 access-list [access-list-name]</pre> <p>Example: Router# show ipv6 access-list</p>	Displays the contents of all current IPv6 access lists.
Step 6	<pre>show ipv6 inspect {name inspection-name config interfaces session [detail] all}</pre> <p>Example: Router# show ipv6 inspect interfaces</p>	Displays CBAC configuration and session information.
Step 7	<pre>show ipv6 port-map [application port port-number]</pre> <p>Example: Router# show ipv6 port-map ftp</p>	Displays PAM configuration.
Step 8	<pre>show ipv6 prefix-list [detail summary] [list-name]</pre> <p>Example: Router# show ipv6 prefix-list</p>	Displays information about an IPv6 prefix list or IPv6 prefix list entries.

	Command or Action	Purpose
Step 9	<p>show ipv6 virtual-reassembly interface <i>interface-type</i></p> <p>Example: Router# show ipv6 virtual-reassembly interface e1/1</p>	Displays configuration and statistical information of VFR.
Step 10	<p>show logging [<i>slot slot-number</i> summary]</p> <p>Example: Router# show logging</p>	<p>Displays the state of system logging (syslog) and the contents of the standard system logging buffer.</p> <ul style="list-style-type: none"> Access list entries with the log or log-input keywords will be logged when a packet matches the access list entry.

Troubleshooting IPv6 Security Configuration and Operation


This optional task explains how to display information to troubleshoot the configuration and operation of IPv6 security options. Use the following commands only as needed to verify configuration and operation.

SUMMARY STEPS

- enable
- clear ipv6 access-list [*access-list-name*]
- clear ipv6 inspect {*session session-number* | all}
- clear ipv6 prefix-list [*prefix-list-name*] [*ipv6-prefix/prefix-length*]
- debug crypto ipsec
- debug crypto engine packet [*detail*]
- debug ipv6 inspect {*function-trace* | *object-creation* | *object-deletion* | *events* | *timers* | *protocol* | *detailed*}
- debug ipv6 packet [*access-list access-list-name*] [*detail*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router# enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>clear ipv6 access-list [<i>access-list-name</i>]</p> <p>Example: Router# clear ipv6 access-list tin</p>	Resets the IPv6 access list match counters.

	Command or Action	Purpose
Step 3	<pre>clear ipv6 inspect {session session-number all}</pre> <p>Example: Router# clear ipv6 inspect all</p>	Removes a specific IPv6 session or all IPv6 inspection sessions.
Step 4	<pre>clear ipv6 prefix-list [prefix-list-name] [ipv6-prefix/prefix-length]</pre> <p>Example: Router# clear ipv6 prefix-list</p>	Resets the hit count of the IPv6 prefix list entries.
Step 5	<pre>debug crypto ipsec</pre> <p>Example: Router# debug crypto ipsec</p>	Displays IPsec network events.
Step 6	<pre>debug crypto engine packet [detail]</pre> <p>Example: Router# debug crypto engine packet</p>	Displays the contents of IPv6 packets.  Caution Using this command could flood the system and increase CPU usage if several packets are being encrypted.
Step 7	<pre>debug ipv6 inspect {function-trace object-creation object-deletion events timers protocol detailed}</pre> <p>Example: Router# debug ipv6 inspect timers</p>	Displays messages about Cisco IOS Firewall events.
Step 8	<pre>debug ipv6 packet [access-list access-list-name] [detail]</pre> <p>Example: Router# debug ipv6 packet access-list PAK-ACL</p>	Displays debugging messages for IPv6 packets.

Examples

This section provides the following output examples:

- [Sample Output from the show crypto ipsec sa ipv6 Command, page 23](#)
- [Sample Output from the show crypto isakmp peer Command, page 24](#)
- [Sample Output from the show crypto isakmp profile Command, page 24](#)
- [Sample Output from the show crypto isakmp sa Command, page 24](#)
- [Sample Output from the show ipv6 access-list Command, page 25](#)
- [Sample Output from the show ipv6 prefix-list Command, page 25](#)
- [Sample Output from the show ipv6 virtual-reassembly Command, page 25](#)
- [Sample Output from the show logging Command, page 26](#)
- [Sample Output from the clear ipv6 access-list Command, page 26](#)

Sample Output from the show crypto ipsec sa ipv6 Command

The following is sample output from the **show crypto ipsec sa ipv6** command:

```
Router# show crypto ipsec sa ipv6

interface: Tunnel0
  Crypto map tag: Tunnel0-head-0, local addr 3FFE:2002::A8BB:CCFF:FE01:9002

protected vrf: (none)
local ident (addr/mask/prot/port): (::/0/0/0)
remote ident (addr/mask/prot/port): (::/0/0/0)
current_peer 3FFE:2002::A8BB:CCFF:FE01:2C02 port 500
  PERMIT, flags={origin_is_acl,}
#pkts encaps: 133, #pkts encrypt: 133, #pkts digest: 133
#pkts decaps: 133, #pkts decrypt: 133, #pkts verify: 133
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 60, #recv errors 0

local crypto endpt.: 3FFE:2002::A8BB:CCFF:FE01:9002,
remote crypto endpt.: 3FFE:2002::A8BB:CCFF:FE01:2C02
path mtu 1514, ip mtu 1514
current outbound spi: 0x28551D9A(676666778)

inbound esp sas:
  spi: 0x2104850C(553944332)
    transform: esp-des ,
    in use settings = {Tunnel, }
    conn id: 93, flow_id: SW:93, crypto map: Tunnel0-head-0
    sa timing: remaining key lifetime (k/sec): (4397507/148)
    IV size: 8 bytes
    replay detection support: Y
    Status: ACTIVE

inbound ah sas:
  spi: 0x967698CB(2524354763)
    transform: ah-sha-hmac ,
    in use settings = {Tunnel, }
    conn id: 93, flow_id: SW:93, crypto map: Tunnel0-head-0
    sa timing: remaining key lifetime (k/sec): (4397507/147)
    replay detection support: Y
    Status: ACTIVE

inbound pcp sas:

outbound esp sas:
  spi: 0x28551D9A(676666778)
    transform: esp-des ,
    in use settings = {Tunnel, }
    conn id: 94, flow_id: SW:94, crypto map: Tunnel0-head-0
    sa timing: remaining key lifetime (k/sec): (4397508/147)
    IV size: 8 bytes
    replay detection support: Y
    Status: ACTIVE

outbound ah sas:
  spi: 0xA83E05B5(2822636981)
    transform: ah-sha-hmac ,
    in use settings = {Tunnel, }
    conn id: 94, flow_id: SW:94, crypto map: Tunnel0-head-0
    sa timing: remaining key lifetime (k/sec): (4397508/147)
    replay detection support: Y
```

```
Status: ACTIVE
```

```
outbound pcp sas:
```

Sample Output from the show crypto isakmp peer Command

The following sample output shows peer descriptions on an IPv6 router:

```
Router# show crypto isakmp peer detail

Peer: 2001:0DB8:0:1::1 Port: 500 Local: 2001:0DB8:0:2::1
Phase1 id: 2001:0DB8:0:1::1
flags:
NAS Port: 0 (Normal)
IKE SAs: 1 IPsec SA bundles: 1
last_locker: 0x141A188, last_last_locker: 0x0
last_unlocker: 0x0, last_last_unlocker: 0x0
```

Sample Output from the show crypto isakmp profile Command

The following sample output shows the ISAKMP profiles that are defined on an IPv6 router:

```
Router# show crypto isakmp profile

ISAKMP PROFILE tom
Identities matched are:
ipv6-address 2001:0DB8:0:1::1/32
Certificate maps matched are:
Identity presented is: ipv6-address fqdn
keyring(s): <none>
trustpoint(s): <all>
```

Sample Output from the show crypto isakmp sa Command

The following sample output shows the SAs of an active IPv6 device. The IPv4 device is inactive:

```
Router# show crypto isakmp sa detail

Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption
IPv4 Crypto ISAKMP SA

C-id Local Remote I-VRF Status Encr Hash Auth DH
Lifetime Cap.

IPv6 Crypto ISAKMP SA

dst: 3FFE:2002::A8BB:CCFF:FE01:2C02
src: 3FFE:2002::A8BB:CCFF:FE01:9002
conn-id: 1001 I-VRF: Status: ACTIVE Encr: des Hash: sha Auth:
psk
DH: 1 Lifetime: 23:45:00 Cap: D Engine-id:Conn-id = SW:1

dst: 3FFE:2002::A8BB:CCFF:FE01:2C02
src: 3FFE:2002::A8BB:CCFF:FE01:9002
conn-id: 1002 I-VRF: Status: ACTIVE Encr: des Hash: sha Auth:
psk
DH: 1 Lifetime: 23:45:01 Cap: D Engine-id:Conn-id = SW:2
```

Sample Output from the show ipv6 access-list Command

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
```

Sample Output from the show ipv6 prefix-list Command

The following example shows the output of the **show ipv6 prefix-list** command with the **detail** keyword:

```
Router# show ipv6 prefix-list detail

Prefix-list with the last deletion/insertion: bgp-in
ipv6 prefix-list 6to4:
  count: 1, range entries: 0, sequences: 5 - 5, refcount: 2
  seq 5 permit 2001:0db8::/32 (hit count: 313, refcount: 1)
ipv6 prefix-list aggregate:
  count: 2, range entries: 2, sequences: 5 - 10, refcount: 30
  seq 5 deny 3FFE:C00::/24 ge 25 (hit count: 568, refcount: 1)
  seq 10 permit ::/0 le 48 (hit count: 31310, refcount: 1)
ipv6 prefix-list bgp-in:
  count: 6, range entries: 3, sequences: 5 - 30, refcount: 31
  seq 5 deny 5F00::/8 le 128 (hit count: 0, refcount: 1)
  seq 10 deny ::/0 (hit count: 0, refcount: 1)
  seq 15 deny ::/1 (hit count: 0, refcount: 1)
  seq 20 deny ::/2 (hit count: 0, refcount: 1)
  seq 25 deny ::/3 ge 4 (hit count: 0, refcount: 1)
  seq 30 permit ::/0 le 128 (hit count: 240664, refcount: 0)
```

Sample Output from the show ipv6 virtual-reassembly Command

The following example shows the output of the **show ipv6 virtual-reassembly** command with the **interface** keyword:

```
Router# show ipv6 virtual-reassembly interface e1/1

Configuration Information:
-----
Virtual Fragment Reassembly (VFR) is ENABLED...
Maximum number of datagram that can be reassembled at a time: 64
Maximum number of fragments per datagram: 8
Timeout value of a datagram: 3 seconds

Statistical Information:
-----
Number of datagram being reassembled:12
Number of fragments being processed:48
Total number of datagram reassembled:6950
Total number of datagram failed: 9
```

Sample Output from the show logging Command

In the following example, the **show logging** command is used to display logging entries that match the first line (sequence 10) of the access list named list1:

```
Router> show logging

00:00:36: %IPV6-6-ACCESSLOGP: list list1/10 permitted tcp 2001:0db8:1::1(11001)
(Ethernet0/0) -> 2001:0db8:1::2(179), 1 packet
```

Sample Output from the clear ipv6 access-list Command

In the following example, the **show ipv6 access-list** command is used to display some match counters for the access list named list1. The **clear ipv6 access-list** command is issued to reset the match counters for the access list named list1. The **show ipv6 access-list** command is used again to show that the match counters have been reset.

```
Router> show ipv6 access-list list1

IPv6 access list list1
  permit tcp any any log-input (6 matches) sequence 10
  permit icmp any any echo-request log-input sequence 20
  permit icmp any any echo-reply log-input sequence 30

Router# clear ipv6 access-list list1

Router# show ipv6 access-list list1

IPv6 access list list1
  permit tcp any any log-input sequence 10
  permit icmp any any echo-request log-input sequence 20
  permit icmp any any echo-reply log-input sequence 30
```

Configuration Examples for Implementing Traffic Filters and Firewalls for IPv6 Security

This section provides the following configuration examples:

- [Creating and Applying IPv6 ACLs: Examples, page 26](#)
- [Controlling Access to a vty: Example, page 28](#)
- [Configuring TCP or UDP Matching: Example, page 28](#)
- [Configuring Cisco IOS Firewall for IPv6: Example, page 28](#)

Creating and Applying IPv6 ACLs: Examples

The following sections provide examples for creating and applying ipv6 ACLs:

- [Creating and Applying an IPv6 ACL for Release 12.2\(13\)T or 12.0\(23\)S: Example, page 26](#)
- [Creating and Applying an IPv6 ACL for 12.2\(11\)T, 12.0\(22\)S, or Earlier Releases: Example, page 27](#)

Creating and Applying an IPv6 ACL for Release 12.2(13)T or 12.0(23)S: Example

The following example is from a router running Cisco IOS Release 12.2(13)T.

The 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 can be run on a router running Cisco IOS Release 12.2(13)T or 12.0(23)S.

The 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
```

Creating and Applying an IPv6 ACL for 12.2(11)T, 12.0(22)S, or Earlier Releases: Example

The following example is from a router running Cisco IOS Release 12.2(11)T, 12.0(22)S, or earlier releases.

The example configures the IPv6 ACL named list2 and applies the ACL to outbound traffic on Ethernet interface 0. Specifically, the first ACL entry keeps all packets from the network fec0:0:0:2::/64 (packets that have the site-local prefix fec0:0:0:2 as the first 64 bits of their source IPv6 address) from exiting out of Ethernet interface 0. The second entry in the ACL permits all other traffic to exit out of Ethernet interface 0. The second entry is necessary because an implicit deny all condition is at the end of each IPv6 ACL.

```
ipv6 access-list list2 deny fec0:0:0:2::/64 any
```

```

ipv6 access-list list2 permit any any

interface ethernet 0
  ipv6 traffic-filter list2 out

```

If the same configuration was used on a router running Cisco IOS Release 12.2(13)T, 12.0(23)S, or later releases, the configuration would be translated into IPv6 access list configuration mode as follows:

```

ipv6 access-list list2
  deny ipv6 fec0:0:0:2::/64 any
  permit ipv6 any any

interface ethernet 0
  ipv6 traffic-filter list2 out

```

**Note**

IPv6 is automatically configured as the protocol type in **permit any any** and **deny any any** statements that are translated from global configuration mode to IPv6 access list configuration mode.

Controlling Access to a vty: Example

In the following example, incoming connections to the virtual terminal lines 0 to 4 are filtered based on the IPv6 access list named acl1:

```

ipv6 access-list acl1
  permit ipv6 host 2001:0DB8:0:4::2/32 any
  !
line vty 0 4
  ipv6 access-class acl1 in

```

Configuring TCP or UDP Matching: Example

The following example allows any TCP traffic regardless of whether or not an AH is present:

```

IPv6 access list example1
  permit tcp any any

```

The following example allows TCP or UDP parsing only when an AH header is present. TCP or UDP traffic without an AH will not be matched:

```

IPv6 access list example2
  deny tcp host 2001::1 any log sequence 5
  permit tcp any any auth sequence 10
  permit udp any any auth sequence 20

```

The following example allows any IPv6 traffic containing an authentication header:

```

IPv6 access list example3
  permit ahp any any

```

Configuring Cisco IOS Firewall for IPv6: Example

This Cisco IOS Firewall configuration example uses inbound and outbound filters for inspection and makes use of access lists to manage the traffic. The inspect mechanism is the method of permitting return traffic based upon a packet being valid for an existing session for which the state is being maintained:

```

enable
configure terminal

```

```

ipv6 unicast-routing
ipv6 inspect name ipv6_test icmp timeout 60
ipv6 inspect name ipv6_test tcp timeout 60
ipv6 inspect name ipv6_test udp timeout 60

interface FastEthernet0/0
  ipv6 address 3FFE:C000:0:7::/64 eui-64
  ipv6 enable
  ipv6 traffic-filter INBOUND out
  ipv6 inspect ipv6_test in

interface FastEthernet0/1
  ipv6 address 3FFE:C000:1:7::/64 eui-64
  ipv6 enable
  ipv6 traffic-filter OUTBOUND in

! This is used for 3745b connection to tftpboot server
interface FastEthernet4/0
  ip address 192.168.17.33 255.255.255.0
  duplex auto
  speed 100

ip default-gateway 192.168.17.8
! end of tftpboot server config

! Access-lists to deny everything except for Neighbor Discovery ICMP messages
ipv6 access-list INBOUND
  permit icmp any any nd-na
  permit icmp any any nd-ns
  deny ipv6 any any log

ipv6 access-list OUTBOUND
  permit icmp any any nd-na
  permit icmp any any nd-ns
  deny ipv6 any any log

```

Additional References

The following sections provide references related to the Implementing Traffic Filters and Firewalls for IPv6 Security feature.

Related Documents

Related Topic	Document Title
IPv6 IPsec	“Implementing IPsec in IPv6 Security,” Cisco IOS IPv6 Configuration Guide
Basic IPv6 configuration	“Implementing IPv6 Addressing and Basic Connectivity,” Cisco IOS IPv6 Configuration Guide
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2401	<i>Security Architecture for the Internet Protocol</i>
RFC 2402	<i>IP Authentication Header</i>
RFC 2428	<i>FTP Extensions for IPv6 and NATs</i>
RFC 2460	<i>Internet Protocol, Version 6 (IPv6) Specification</i>
RFC 2474	<i>Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers</i>
RFC 3576	<i>Change of Authorization</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Feature Information for Implementing Traffic Filters and Firewalls for IPv6 Security

[Table 1](#) lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(2)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifics for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

**Note**

[Table 1](#) lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Implementing Traffic Filters and Firewalls for IPv6 Security

Feature Name	Releases	Feature Information
IPv6 services: standard access control lists	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(17a)SX1 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>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.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Restrictions for Implementing Traffic Filters and Firewalls for IPv6 Security, page 2 • Access Control Lists for IPv6 Traffic Filtering, page 2 • PAM in Cisco IOS Firewall for IPv6, page 4 • How to Implement Traffic Filters and Firewalls for IPv6 Security, page 5 • Configuration Examples for Implementing Traffic Filters and Firewalls for IPv6 Security, page 26
IPv6 services: extended access control lists ¹	12.0(23)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(17a)SX1 12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Restrictions for Implementing Traffic Filters and Firewalls for IPv6 Security, page 2 • Access Control Lists for IPv6 Traffic Filtering, page 2 • How to Implement Traffic Filters and Firewalls for IPv6 Security, page 5 • Configuration Examples for Implementing Traffic Filters and Firewalls for IPv6 Security, page 26
IPv6 services: IPv6 IOS Firewall	12.3(7)T 12.4 12.4(2)T	<p>This feature provides advanced traffic filtering functionality as an integral part of a network's firewall.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Cisco IOS Firewall for IPv6, page 3 • Configuring the Cisco IOS Firewall for IPv6, page 14

Table 1 Feature Information for Implementing Traffic Filters and Firewalls for IPv6 Security (continued)

Feature Name	Releases	Feature Information
IPv6 services: IPv6 IOS Firewall FTP application support	12.3(11)T 12.4 12.4(2)T	IPv6 supports this feature. The following section provides information about this feature: <ul style="list-style-type: none"> • Cisco IOS Firewall for IPv6, page 3
IPv6 ACL extensions for IPsec Authentication Header	12.4(20)T	The IPv6 ACL extensions for IPsec authentication headers feature allows TCP or UDP parsing when an IPv6 IPsec authentication header is present. The following section provides information about this feature: <ul style="list-style-type: none"> • IPv6 ACL Extensions for IPsec Authentication Header, page 2

1. IPv6 extended access control lists and IPv6 provider edge router over Multiprotocol Label Switching (MPLS) are implemented with hardware acceleration on the Cisco 12000 series Internet router IP service engineer (ISE) line cards in Cisco IOS routers in Cisco IOS Release 12.0(25)S and later releases.

CCDE, CCENT, CCSI, Cisco Eos, Cisco HealthPresence, Cisco IronPort, the Cisco logo, Cisco Nurse Connect, Cisco Pulse, Cisco SensorBase, Cisco StackPower, Cisco StadiumVision, Cisco TelePresence, Cisco Unified Computing System, Cisco WebEx, DCE, Flip Channels, Flip for Good, Flip Mino, Flipshare (Design), Flip Ultra, Flip Video, Flip Video (Design), Instant Broadband, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn, Cisco Capital, Cisco Capital (Design), Cisco:Financed (Stylized), Cisco Store, Flip Gift Card, and One Million Acts of Green are service marks; and Access Registrar, Aironet, AllTouch, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Lumin, Cisco Nexus, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, Continuum, EtherFast, EtherSwitch, Event Center, Explorer, Follow Me Browsing, GainMaker, iLYNX, IOS, iPhone, IronPort, the IronPort logo, Laser Link, LightStream, Linksys, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, PCNow, PIX, PowerKEY, PowerPanels, PowerTV, PowerTV (Design), PowerVu, Prisma, ProConnect, ROSA, SenderBase, SMARTnet, Spectrum Expert, StackWise, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0910R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.

