Implementing Traffic Filters and Firewalls for IPv6 Security

Last Updated: August 1, 2012

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 caveats and feature information, see Bug Search Tool and 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 table at the end of this module.

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

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

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.

IPv6 extended ACLs augments standard IPv6 ACL functionality 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.

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 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, page 3
- Cisco IOS Firewall Alerts Audit Trails and System Logging, page 4
- IPv6 Packet Inspection, page 4
- Cisco IOS Firewall Restrictions, page 4

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.

Cisco IOS Firewall Restrictions

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

Zone-Based Policy Firewall IPv6 Support

The zone-based policy firewall for IPv6 coexists with the zone-based policy firewall for IPv4 in order to support IPv6 traffic. The feature provides MIB support for TCP, UDP, ICMPv6, and FTP sessions.

ACL--Hardware and Software Counters Granularity for IPv4 and IPv6 ACL Statistics

Each IPv6 and IPv4 ACL entry maintains a global counter per entry for the number of matches applied to the ACL entry. The counters reflect all matches applied to the ACL, regardless of where the match was applied (such as on the platform or in the software feature path). This feature allows both IPv4 and IPv6 ACLs on the Cisco Catalyst 6500 platform to update the ACL entry statistics with a platform entry count.

How to Implement Traffic Filters and Firewalls for IPv6 Security

Configuring IPv6 Traffic Filtering

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, page 5 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, page 12 section.

Creating and Configuring an IPv6 ACL for Traffic Filtering

This section describes how to configure your networking devices to filter traffic, function as a firewall, or detect potential viruses.

Note

- 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. Do one of the following:
   - **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*]
   -
   -
   - **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 access-list** *access-list-name* **Example:** `Router(config)# ipv6 access-list outbound`	Defines an IPv6 ACL, and enters IPv6 access list configuration mode. • The *access-list name* argument specifies the name of the IPv6 ACL. IPv6 ACL names cannot contain a space or quotation mark, or begin with a numeral.

Command or Action	Purpose

Step 4 Do one of the following:
• **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*]
•
•
• **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**]
Example:
`Router(config-ipv6-acl)# permit tcp 2001:DB8:0300:0201::/32 eq telnet any reflect reflectout`
Example:
Example:
Example:
`Router(config-ipv6-acl)# deny tcp host 2001:DB8:1::1 any log-input`
	Specifies permit or deny conditions for an IPv6 ACL.

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** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**ipv6 traffic-filter** *access-list-name* {**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.

Controlling Access to a vty

Creating an IPv6 ACL to Provide Access Class Filtering

Perform this task 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. Do one of the following:
   - **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*]
   -
   -
   - **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 access-list** *access-list-name* **Example:** `Router(config)# ipv6 access-list cisco`	Defines an IPv6 ACL, and enters IPv6 access list configuration mode.

Command or Action	Purpose

Step 4 Do one of the following:
• **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*]
•
•
• **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**]
Example:
`Router(config-ipv6-acl)# permit ipv6 host 2001:DB8:0:4::32 any eq telnet`
Example:
Example:
`Router(config-ipv6-acl)# deny ipv6 host 2001:DB8:0:6::6/32 any`
	Specifies permit or deny conditions for an IPv6 ACL.

Applying an IPv6 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. • 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. • 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.

Perform this task to allow TCP or UDP traffic to be matched to the ULP if an AH is present.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **permit icmp auth**

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** *access-list-name* **Example:** `Router(config)# ipv6 access-list list1`	Defines an IPv6 access list and places the router in IPv6 access list configuration mode.

Step 4	**permit icmp auth**
	
	Example:
	
	Example:
	`or`
	
	Example:
	deny icmp auth
	
	Example:
	`Router(config-ipv6-acl)# permit icmp auth`
		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

Perform the following tasks 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

Perform this task 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.

Note

- 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.

	Command or Action	Purpose
Step 3	**ipv6 access-list** *access-list-name* {**permit** | **deny**} {*source-ipv6-prefix* / *prefix-length*| **any**} {*destination-ipv6-prefix* / *prefix-length*| **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

Perform this task 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. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies the interface type and number, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ipv6 traffic-filter** *access-list-name* {**in**| **out**} **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 configuration scenario uses both packet inspection and ACLs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **ipv6 inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail**{**on** | **off**}] [**timeout** *seconds*]
5. **interface** *type number*
6. **ipv6 address** {*ipv6-address* / *prefix-length* | *prefix-name sub-bits* / *prefix-length*
7. **ipv6 enable**
8. **ipv6 traffic-filter** *access-list-name* {**in** | **out**
9. **ipv6 inspect** *inspection-name* {**in** | **out**
10. **ipv6 access-list** *access-list-name*
11. Do one of the following:
    - **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*]
    -
    -
    - **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	**ipv6 traffic-filter** *access-list-name* {**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.
Step 9	**ipv6 inspect** *inspection-name* {**in** | **out** **Example:** `Router(config)# ipv6 inspect ipv6_test in`	Applies the set of inspection rules.
Step 10	**ipv6 access-list** *access-list-name* **Example:** `Router(config)# ipv6 access-list outbound`	Defines an IPv6 ACL and enters IPv6 access list configuration mode. The router prompt changes to Router(config-ipv6-acl)#.

Command or Action	Purpose

Step 11 Do one of the following:
• **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*]
•
•
• **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**]
Example:
`Router(config-ipv6-acl)# permit tcp 2001:DB8:0300:0201::/32 any reflect reflectout`
Example:
Example:
`Router(config-ipv6-acl)# deny tcp fec0:0:0:0201::/64 any`
	Specifies permit or deny conditions for an IPv6 ACL.

• Configuring PAM for IPv6, page 18

Configuring PAM for IPv6

Creating an IPv6 Access Class Filter for PAM

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. Do one of the following:
   - **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*]
   -
   -
   - **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 access-list** *access-list-name* **Example:** `Router(config)# ipv6 access-list outbound`	Defines an IPv6 ACL and enters IPv6 access list configuration mode.

Command or Action	Purpose

Step 4 Do one of the following:
• **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*]
•
•
• **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**
Example:
`Router(config-ipv6-acl)# permit tcp 2001:DB8:0300:0201::/32 any reflect reflectout`
Example:
Example:
`Router(config-ipv6-acl)# deny tcp fec0:0:0:0201::/64 any`
	Specifies permit or deny conditions for an IPv6 ACL.

Applying the IPv6 Access Class 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. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 port-map** *application-name* **port** *port-num* [**list** *acl-name* **Example:** `Router(config)# ipv6 port-map ftp port 8090 list PAMACL`	Establishes PAM for the system.

Configuring Zone-Based Firewall in IPv6

Configuring an Inspect-Type Parameter Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** {*parameter-map-name* | **global** | **default**}
4. **sessions maximum** *sessions*
5. **ipv6 routing-enforcement-header loose**

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	**parameter-map type inspect** {*parameter-map-name* | **global** | **default**} **Example:** `Router(config)# parameter-map type inspect v6-param-map`	Configures an inspect type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the inspect action, and places the router in parameter map configuration mode.
Step 4	**sessions maximum** *sessions* **Example:** `Router(config-profile)# sessions maximum 10000`	Sets the maximum number of allowed sessions that can exist on a zone pair.
Step 5	**ipv6 routing-enforcement-header loose** **Example:** `Router(config-profile)# ipv6 routing-enforcement-header loose`	Provides backward compatibility with legacy IPv6 inspection.

Creating and Using an Inspect-Type Class Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** {**match-any** | **match-all**} *class-map-name*
4. **match protocol tcp**
5. **match protocol udp**
6. **match protocol icmp**
7. **match protocol ftp**

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	**class-map type inspect** {**match-any** | **match-all**} *class-map-name* **Example:** `Router(config-profile)# class-map type inspect match-any v6-class`	Create an inspect type class map, and places the router in lass-map configuration mode.
Step 4	**match protocol tcp** **Example:** `Router(config-cmap)# match protocol tcp`	Configures the match criterion for a class map based on TCP.
Step 5	**match protocol udp** **Example:** `Router(config-cmap)# match protocol udp`	Configures the match criterion for a class map based on UDP.

	Command or Action	Purpose
Step 6	**match protocol icmp** **Example:** `Router(config-cmap)# match protocol icmp`	Configures the match criterion for a class map based on ICMP.
Step 7	**match protocol ftp** **Example:** `Router(config-cmap)# match protocol ftp`	Configures the match criterion for a class map based on FTP.

Creating and Using an Inspect-Type Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *policy-map-name*
4. **class type inspect** *class-map-name*
5. **inspect** [*parameter-map-name*]

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	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect v6-policy`	Creates an inspect-type policy map, and places the router in policy-map configuration mode.

	Command or Action	Purpose
Step 4	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect v6-class`	Specifies the traffic (class) on which an action is to be performed.
Step 5	**inspect** [*parameter-map-name*] **Example:** `Router(config-pmap)# inspect`	Enables Cisco IOS stateful packet inspection.

Creating Security Zones and Zone Pairs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **zone security** {*zone-name* | **default**}
5. **zone-pair security** *zone-pair-name* **source** {*source-zone-name* | **self** | **default**} **destination** {*destination-zone-name* | **self** | **default**}
6. **service-policy type inspect** *policy-map-name*

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	**zone security** {*zone-name* | **default**} **Example:** `Router(config)# zone security 1`	Creates a security zone. • Cisco recommends that you create at least two security zones so that you can create a zone pair.
Step 4	**zone security** {*zone-name* | **default**} **Example:** `Router(config)# zone security 2`	Creates a security zone. • Cisco recommends that you create at least two security zones so that you can create a zone pair.
Step 5	**zone-pair security** *zone-pair-name* **source** {*source-zone-name* | **self** | **default**} **destination** {*destination-zone-name* | **self** | **default**} **Example:** `Router(config)# zone-pair security zp source z1 destination z2`	Creates a zone pair, and places the router in zone-pair configuration mode.
Step 6	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect v6-policy`	Attaches a firewall policy map to a zone pair.

Configuring ACL Hardware and Software Counters Granularity for IPv4 and IPv6 ACL Statistics

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **hardware statistics**

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** *access-list-name* **Example:** `Router(config)# ipv6 access-list outbound`	Defines an IPv6 ACL and enters IPv6 access list configuration mode.
Step 4	**hardware statistics** **Example:** `Router(config-ipv6-acl)# hardware statistics`	Enables the collection of hardware statistics.

Verifying IPv6 Security Configuration and Operation

SUMMARY STEPS

1. **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**
2. **show crypto isakmp peer** [**config** | **detail**]
3. **show crypto isakmp profile**
4. **show crypto isakmp sa** [**active** | **standby** | **detail** | **nat**
5. **show ipv6 access-list** [*access-list-name*]
6. **show ipv6 inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all**}
7. **show ipv6 port-map** [*application* | **port** *port-number*
8. **show ipv6 prefix-list** [**detail** | **summary**] [*list-name*
9. **show ipv6 virtual-reassembly interface** *interface-type*
10. **show logging** [**slot** *slot-number* | **summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**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** **Example:** `Router# show crypto ipsec sa ipv6`	Displays the settings used by current SAs.
Step 2	**show crypto isakmp peer** [**config** | **detail**] **Example:** `Router# show crypto isakmp peer`	Displays peer descriptions.
Step 3	**show crypto isakmp profile** **Example:** `Router# show crypto isakmp profile`	Lists all the ISAKMP profiles that are defined on a router.
Step 4	**show crypto isakmp sa** [**active** | **standby** | **detail** | **nat** **Example:** `Router# show crypto isakmp sa`	Displays current IKE SAs.
Step 5	**show ipv6 access-list** [*access-list-name*] **Example:** `Router# show ipv6 access-list`	Displays the contents of all current IPv6 access lists.
Step 6	**show ipv6 inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all**} **Example:** `Router# show ipv6 inspect interfaces`	Displays CBAC configuration and session information.

	Command or Action	Purpose
Step 7	**show ipv6 port-map** [*application* | **port** *port-number* **Example:** `Router# show ipv6 port-map ftp`	Displays PAM configuration.
Step 8	**show ipv6 prefix-list** [**detail** | **summary**] [*list-name* **Example:** `Router# show ipv6 prefix-list`	Displays information about an IPv6 prefix list or IPv6 prefix list entries.
Step 9	**show ipv6 virtual-reassembly interface** *interface-type* **Example:** `Router# show ipv6 virtual-reassembly interface e1/1`	Displays configuration and statistical information of VFR.
Step 10	**show logging** [**slot** *slot-number* | **summary** **Example:** `Router# show logging`	Displays the state of system logging (syslog) and the contents of the standard system logging buffer. • 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

SUMMARY STEPS

1. **enable**
2. **clear ipv6 access-list** [*access-list-name*]
3. **clea r ipv6 inspect** {**session** *session-number* | all}
4. **clear ipv6 prefix-list** [*prefix-list-name*] [*ipv6-prefix* / *prefix-length*
5. **debug crypto ipsec**
6. **debug crypto engine packet** [**detail**
7. **debug ipv6 inspect** {**function-trace** | **object-creation** | **object-deletion** | **events** | **timers** | **protocol** | **detailed**
8. **debug ipv6 packet** [**access-list** *access-list-name*] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 access-list** [*access-list-name*] **Example:** `Router# clear ipv6 access-list tin`	Resets the IPv6 access list match counters.
Step 3	**clea r ipv6 inspect** {**session** *session-number* | all} **Example:** `Router# clear ipv6 inspect all`	Removes a specific IPv6 session or all IPv6 inspection sessions.
Step 4	**clear ipv6 prefix-list** [*prefix-list-name*] [*ipv6-prefix* / *prefix-length* **Example:** `Router# clear ipv6 prefix-list`	Resets the hit count of the IPv6 prefix list entries.
Step 5	**debug crypto ipsec** **Example:** `Router# debug crypto ipsec`	Displays IPsec network events.
Step 6	**debug crypto engine packet** [**detail** **Example:** `Router# debug crypto engine packet`	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	**debug ipv6 inspect** {**function-trace** | **object-creation** | **object-deletion** | **events** | **timers** | **protocol** | **detailed** **Example:** `Router# debug ipv6 inspect timers`	Displays messages about Cisco IOS Firewall events.

Command or Action	Purpose
Step 8 **debug ipv6 packet** [**access-list** *access-list-name*] [**detail**] **Example:** `Router# debug ipv6 packet access-list PAK-ACL`	Displays debugging messages for IPv6 packets.

- Examples, page 31

Examples

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:DB8:0:1::1 Port: 500 Local: 2001:DB8:0:2::1
Phase1 id: 2001:DB8:0:1::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:DB8:0:1::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:DB8:1::32 eq bgp host 2001:DB8:2::32 eq 11000 timeout 300
(time         left 243) sequence 1
    permit tcp host 2001:DB8:1::32 eq telnet host 2001:DB8:2::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:DB8::/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:DB8:1::1(11001)
(Ethernet0/0) -> 2001:DB8:1::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

Examples Creating and Applying IPv6 ACLs

Example: Creating and Applying an IPv6 ACL

This 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:DB8:0300:0201::/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:DB8:0300:0201::/32 any reflect REFLECTOUT
 permit udp 2001:DB8:0300:0201::/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 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:DB8::/32 any
 permit udp 2001:DB8::/32 any
```

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

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.

Example Controlling Access to a vty

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:DB8:0:4::2/32 any
```

```
!
line vty 0 4
 ipv6 access-class acl1 in
```

Example: Configuring TCP or UDP Matching

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
```

Example: Configuring Cisco IOS Firewall for IPv6

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
```

Example: Configuring Cisco IOS Zone-Based Firewall for IPv6

```
parameter-map type inspect v6-param-map
 sessions maximum 10000
 ipv6 routing-header-enforcement loose
!
!
class-map type inspect match-any v6-class
 match protocol tcp
 match protocol udp
 match protocol icmp
 match protocol ftp
!
!
policy-map type inspect v6-policy
 class type inspect v6-class
  inspect
!
zone security z1
zone security z2
!
zone-pair security zp source z1 destination z2
 service-policy type inspect v6-policy
```

Additional References

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*
Zone-based firewalls	" Zone-Based Policy Firewall ," *Cisco IOS Security Configuration Guide: Securing the Data Plane*
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
• CISCO-UNIFIED-FIREWALL-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

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

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Implementing Traffic Filters and Firewalls for IPv6 Security

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

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

Feature Name	Releases	Feature Information
ACL--Hardware and Software Counters Granularity for IPv4 and IPv6 ACL Statistics	12.2(50)SY	This feature allows both IPv4 and IPv6 ACLs on the Cisco Catalyst 6500 platform to update the ACL entry statistics with a platform entry count.
IOS Zone-Based Firewall	15.1(2)T	Cisco IOS Zone-Based Firewall for IPv6 coexists with Cisco IOS Zone-Based Firewall for IPv4 in order to support IPv6 traffic.
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.
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)SX112.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T 15.0(1)S	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.
IPv6 Services--IPv6 IOS Firewall	12.3(7)T 12.4 12.4(2)T	This feature provides advanced traffic filtering functionality as an integral part of a network's firewall.
IPv6 Services--IPv6 IOS Firewall FTP Application Support	12.3(11)T 12.4 12.4(2)T	IPv6 supports this feature.

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 engine (ISE) line cards in Cisco IOS routers in Cisco IOS Release 12.0(25)S and later releases.

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 15.0(1)S	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.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned 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. (1110R)

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

© 2012 Cisco Systems, Inc. All rights reserved.