ipv6-i4

ipv6 snooping attach-policy

To apply an IPv6 snooping policy to a target, use the **ipv6 snooping attach-policy** command in IPv6 snooping configuration mode. To remove a policy from a target, use the **no** form of this command.

ipv6 snooping policy attach-policy *snooping-policy*

Syntax Description

snooping-policy	User-defined name of the snooping policy. The policy name can be a symbolic string (such as Engineering) or an integer (such as 0).

Command Default

An IPv6 snooping policy is not attached to a target.

Command Modes

IPv6 snooping configuration (config-ipv6-snooping)

Command History

Release	Modification
15.0(2)SE	This command was introduced.
15.3(1)S	This command was integrated into Cisco IOS Release 15.3(1)S.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

Once a policy has been identified or configured, it is applied on a target using the **ipv6 snooping attach-policy** command. This command is applied on any target, which varies depending on the platform. Examples of targets (depending on the platform used) include device ports, switchports, Layer 2 interfaces, Layer 3 interfaces, and VLANs.

Examples

The following examples shows how to apply an IPv6 snooping policy named policy1 to a target:

```
Device(config)# ipv6 snooping policy policy1
Device(config-ipv6-snooping)# ipv6 snooping attach-policy policy1
```

Related Commands

Command	Description
ipv6 snooping policy	Configures an IPv6 snooping policy and enters IPv6 snooping configuration mode.

ipv6 snooping policy

To configure an IPv6 snooping policy and enter IPv6 snooping configuration mode, use the **ipv6 snooping policy** command in global configuration mode. To delete an IPv6 snooping policy, use the **no** form of this command.

ipv6 snooping policy *snooping-policy*

no ipv6 snooping policy *snooping-policy*

Syntax Description

snooping-policy	User-defined name of the snooping policy. The policy name can be a symbolic string (such as Engineering) or an integer (such as 0).

Command Default

An IPv6 snooping policy is not configured.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.0(2)SE	This command was introduced.
15.3(1)S	This command was integrated into Cisco IOS Release 15.3(1)S.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

Use the **ipv6 snooping policy** command to create an IPv6 snooping policy. When the **ipv6 snooping policy** command is enabled, the configuration mode changes to IPv6 snooping configuration mode. In this mode, the administrator can configure the following IPv6 first-hop security commands:

- The **data-glean**/**destination-glean** command enables IPv6 first-hop security binding table recovery using data or destination address gleaning.
- The **device-role** command specifies the role of the device attached to the port.
- The **limit address-count** *maximum* command limits the number of IPv6 addresses allowed to be used on the port.
- **security-level** specifies the level of security enforced.
- The **tracking** command overrides the default tracking policy on a port.

- The **trusted-port** command configures a port to become a trusted port; that is, limited or no verification is performed when messages are received.

Once a policy has been identified or configured, it is applied on a device using the **ipv6 snooping attach-policy** command.

Examples

The following examples show hows to configure an IPv6 snooping policy:

```
Device(config)# ipv6 snooping policy policy1
```

Related Commands

Command	Description
ipv6 snooping attach-policy	Applies an IPv6 snooping policy to a target.

ipv6 traffic-filter

To filter incoming or outgoing IPv6 traffic on an interface, use the **ipv6 traffic-filter**command in interface configuration mode. To disable the filtering of IPv6 traffic on an interface, use the **no** form of this command.

ipv6 traffic-filter *access-list-name* {**in**| **out**}

no ipv6 traffic-filter *access-list-name*

Syntax Description

access-list-name	Specifies an IPv6 access name.
in	Specifies incoming IPv6 traffic.
out	Specifies outgoing IPv6 traffic.

Command Default

Filtering of IPv6 traffic on an interface is not configured.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
12.2(2)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 series routers.
12.2(33)SXI4	The **out** keyword and therefore filtering of outgoing traffic is not supported in IPv6 port-based access list (PACL) configuration.
12.2(54)SG	This command was modified. Support for Cisco IOS Release 12.2(54)SG was added.

Release	Modification
12.2(50)SY	This command was modified. The **out** keyword is not supported.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Examples

The following example filters inbound IPv6 traffic on Ethernet interface 0/0 as defined by the access list named cisco:

```
Router(config)# interface ethernet 0/0
Router(config-if)# ipv6 traffic-filter cisco in
```

Related Commands

Command	Description
ipv6 access-list	Defines an IPv6 access list and sets deny or permit conditions for the defined access list.
show ipv6 access-list	Displays the contents of all current IPv6 access lists.
show ipv6 interface	Displays the usability status of interfaces configured for IPv6.

ipv6 verify unicast source reachable-via

To verify that a source address exists in the FIB table and enable Unicast Reverse Path Forwarding (Unicast RPF), use the **ipv6 verify unicast source reachable-via** command in interface configuration mode. To disable URPF, use the **no** form of this command.

ipv6 verify unicast source reachable-via {**rx**| **any**} **[allow-default] [allow-self-ping]** [*access-list-name*]

no ipv6 verify unicast

Syntax Description

rx	Source is reachable through the interface on which the packet was received.
any	Source is reachable through any interface.
allow-default	(Optional) Allows the lookup table to match the default route and use the route for verification.
allow-self-ping	(Optional) Allows the router to ping a secondary address.
access-list-name	(Optional) Name of the IPv6 access list. Names cannot contain a space or quotation mark, or begin with a numeral.

Command Default

Unicast RPF is disabled.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Aggregation Services Routers.

Usage Guidelines

The **ipv6 verify unicast reverse-path** command is used to enable Unicast RPF for IPv6 in loose checking mode.

Use the **ipv6 verify unicast source reachable-via**command to mitigate problems caused by malformed or forged (spoofed) IP source addresses that pass through an IPv6 router. Malformed or forged source addresses can indicate denial-of-service (DoS) attacks based on source IPv6 address spoofing.

The URPF feature checks to see if any packet received at a router interface arrives on one of the best return paths to the source of the packet. The feature does this by doing a reverse lookup in the CEF table. If URPF does not find a reverse path for the packet, U RPF can drop or forward the packet, depending on whether an access control list (ACL) is specified in the **ipv6 verify unicast source reachable-via** command. If an ACL is specified in the command, then when (and only when) a packet fails the URPF check, the ACL is checked to see if the packet should be dropped (using a deny statement in the ACL) or forwarded (using a permit statement in the ACL). Whether a packet is dropped or forwarded, the packet is counted in the global IP traffic statistics for U RPF drops and in the interface statistics for Unicast RPF.

If no ACL is specified in the **ipv6 verify unicast source reachable-via** command, the router drops the forged or malformed packet immediately and no ACL logging occurs. The router and interface Unicast RPF counters are updated.

U RPF events can be logged by specifying the logging option for the ACL entries used by the **ipv6 verify unicast source reachable-via** command. Log information can be used to gather information about the attack, such as source address, time, and so on.

Examples

The following example enables Unicast RPF on any interface:

```
ipv6 verify unicast source reachable-via any
```

Related Commands

Command	Description
ipv6 access-list	Defines an IPv6 access list and places the router in IPv6 access list configuration mode.
show ipv6 interface	Displays the usability status of interfaces configured for IPv6.

managed-config-flag

To verify the advertised managed address configuration parameter, use the **managed-config-flag** command in RA guard policy configuration mode.

managed-config-flag {**on**| **off**}

Syntax Description

on	Verification is enabled.
off	Verification is disabled.

Command Default

Verification is not enabled.

Command Modes

RA guard policy configuration (config-ra-guard)

Command History

Release	Modification
12.2(50)SY	This command was introduced.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.
15.0(2)SE	This command was integrated into Cisco IOS Release 15.0(2)SE.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

The **managed-config-flag** command enables verification of the advertised managed address configuration parameter (or "M" flag). This flag could be set by an attacker to force hosts to obtain addresses through a DHCPv6 server that may not be trustworthy.

Examples

The following example shows how the command defines a router advertisement (RA) guard policy name as raguard1, places the router in RA guard policy configuration mode, and enables M flag verification:

```
Router(config)# ipv6 nd raguard policy raguard1
Router(config-ra-guard)# managed-config-flag on
```

Related Commands

Command	Description
ipv6 nd raguard policy	Defines the RA guard policy name and enters RA guard policy configuration mode.

match ipv6

To configure one or more of the IPv6 fields as a key field for a flow record, use the **match ipv6** command in Flexible NetFlow flow record configuration mode. To disable the use of one or more of the IPv6 fields as a key field for a flow record, use the **no** form of this command.

match ipv6 {**dscp**| **flow-label**| **next-header**| **payload-length**| **precedence**| **protocol**| **traffic-class**| **version**}

no match ipv6 {**dscp**| **flow-label**| **next-header**| **payload-length**| **precedence**| **protocol**| **traffic-class**| **version**}

Cisco Catalyst 6500 Switches in Cisco IOS Release 12.2(50)SY

match ipv6 {**dscp**| **precedence**| **protocol**| **tos**}

no match ipv6 {**dscp**| **precedence**| **protocol**| **tos**}

Cisco IOS XE Release 3.2SE

match ipv6 {**protocol**| **traffic-class**| **version**}

no match ipv6 {**protocol**| **traffic-class**| **version**}

Syntax Description

dscp	Configures the IPv6 differentiated services code point DSCP (part of type of service (ToS)) as a key field.
flow-label	Configures the IPv6 flow label as a key field.
next-header	Configures the IPv6 next header as a key field.
payload-length	Configures the IPv6 payload length as a key field.
precedence	Configures the IPv6 precedence (part of ToS) as a key field.
protocol	Configures the IPv6 protocol as a key field.
tos	Configures the IPv6 ToS as a key field.
traffic-class	Configures the IPv6 traffic class as a key field.
version	Configures the IPv6 version from IPv6 header as a key field.

Command Default

The IPv6 fields are not configured as a key field.

Command Modes

Flexible Netflow flow record configuration (config-flow-record)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.2(33)SRE	This command was modified. Support for this command was implemented on the Cisco 7200 and Cisco 7300 Network Processing Engine (NPE) series routers.
12.2(50)SY	This command was modified. The **flow-label**, **next-header**, **payload-length,traffic-class**, and **version** keywords were removed.
15.2(2)T	This command was modified. Support for the Cisco Performance Monitor was added.
Cisco IOS XE Release 3.5S	This command was modified. Support for the Cisco Performance Monitor was added.
Cisco IOS XE Release 3.2SE	This command was modified. The **dscp**, **flow-label**, **next-header**, **payload-length**, and **precedence** keywords were removed.

Usage Guidelines

This command can be used with both Flexible NetFlow and Performance Monitor. These products use different commands to enter the configuration mode in which you issue this command, however the mode prompt is the same for both products. For Performance Monitor, you must first enter the **flow record type performance-monitor** command before you can use this command.

Because the mode prompt is the same for both products, here we refer to the command mode for both products as flow record configuration mode. However, for Flexible NetFlow, the mode is also known as Flexible NetFlow flow record configuration mode; and for Performance Monitor, the mode is also known as Performance Monitor flow record configuration mode.

A flow record requires at least one key field before it can be used in a flow monitor. The key fields differentiate flows, with each flow having a unique set of values for the key fields. The key fields are defined using the **match** command.

Note Some of the keywords of the **match ipv6** command are documented as separate commands. All of the keywords for the **match ipv6** command that are documented separately start with **match ipv6**. For example, for information about configuring the IPv6 hop limit as a key field for a flow record, refer to the **match ipv6 hop-limit** command.

Examples

The following example configures the IPv6 DSCP field as a key field:

```
Router(config)# flow record FLOW-RECORD-1
Router(config-flow-record)# match ipv6 dscp
```

The following example configures the IPv6 DSCP field as a key field:

```
Router(config)# flow record type performance-monitor RECORD-1
Router(config-flow-record)# match ipv6 dscp
```

Related Commands

Command	Description
flow record	Creates a flow record, and enters Flexible NetFlow flow record configuration mode.
flow record type performance-monitor	Creates a flow record, and enters Performance Monitor flow record configuration mode.

match ipv6 access-list

To verify the sender's IPv6 address in inspected messages from the authorized prefix list, use the **match ipv6 access-list** command in RA guard policy configuration mode.

match ipv6 access-list *ipv6-access-list-name*

Syntax Description

ipv6-access-list-name	The IPv6 access list to be matched.

Command Default

Senders' IPv6 addresses are not verified.

Command Modes

RA guard policy configuration (config-ra-guard)

Command History

Release	Modification
12.2(50)SY	This command was introduced.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.
15.0(2)SE	This command was integrated into Cisco IOS Release 15.0(2)SE.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

The **match ipv6 access-list** command enables verification of the sender's IPv6 address in inspected messages from the configured authorized router source access list. If the **match ipv6 access-list** command is not configured, this authorization is bypassed.

An access list is configured using the **ipv6 access-list** command. For instance, to authorize the router with link-local address FE80::A8BB:CCFF:FE01:F700 only, define the following IPv6 access list:

```
Router(config)# ipv6 access-list list1
Router(config-ipv6-acl)# permit host FE80::A8BB:CCFF:FE01:F700 any
```

Note The access list is used here as a convenient way to define several explicit router sources, but it should not be considered to be a port-based access list (PACL). The **match ipv6 access-list** command verifies the IPv6 source address of the router messages, so specifying a destination in the access list is meaningless and the destination of the access control list (ACL) entry should always be "any." If a destination is specified in the access list, then matching will fail.

Examples

The following example shows how the command defines a router advertisement (RA) guard policy name as raguard1, places the router in RA guard policy configuration mode, and matches the IPv6 addresses in the access list named list1:

```
Router(config)# ipv6 nd raguard policy raguard1
Router(config-ra-guard)# match ipv6 access-list list1
```

Related Commands

Command	Description
ipv6 nd raguard policy	Defines the RA guard policy name and enters RA guard policy configuration mode.
ipv6 access-list	Defines an IPv6 access list and places the router in IPv6 access list configuration mode.

match ipv6 address

To distribute IPv6 routes that have a prefix permitted by a prefix list or to specify an IPv6 access list to be used to match packets for policy-based routing (PBR) for IPv6, use the **match ipv6 address** command in route-map configuration mode. To remove the **match ipv6 address** entry, use the **no** form of this command.

match ipv6 address {**prefix-list** *prefix-list-name*| *access-list-name*}

no match ipv6 address

Syntax Description

prefix-list *prefix-list-name*	Specifies the name of an IPv6 prefix list.
access-list-name	Name of the IPv6 access list. Names cannot contain a space or quotation mark or begin with a numeric.

Command Default

No routes are distributed based on the destination network number or an access list.

Command Modes

Route-map configuration (config-route-map)

Command History

Release	Modification
12.2(2)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.3(7)T	This command was modified. The *access-list-name* argument was added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SXI4	This command was modified. The **prefix-list** *prefix-list-name* keyword-argument pair argument is not supported in Cisco IOS Release 12.2(33)SXI4.
Cisco IOS XE Release 3.2S	This command was integrated into Cisco IOS XE Release 3.2S.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

Use the **route-map** command and the **match** and **set** commands to define the conditions for redistributing routes from one routing protocol to another. Each **route-map** command has a list of **match** and **set** commands associated with it. The **match** commands specify the match criteria--the conditions under which redistribution is allowed for the current **route-map** command. The **set** commands specify the set actions, which are the particular redistribution actions to be performed if the criteria enforced by the **match** commands are met.

The **match ipv6 address** command can be used to specify either an access list or a prefix list. When using PBR, you must use the *access-list-name* argument; the **prefix-list** *prefix-list-name* keyword-argument pair argument will not work.

Examples

In the following example, IPv6 routes that have addresses specified by the prefix list named marketing are matched:

```
Device(config)# route-map name
Device(config-route-map)# match ipv6 address prefix-list marketing
```

In the following example, IPv6 routes that have addresses specified by an access list named marketing are matched:

```
Device(config)# route-map
Device(config-route-map)# match ipv6 address marketing
```

Related Commands

Command	Description
match as-path	Matches a BGP autonomous system path access list.
match community	Matches a BGP community.
match ipv6 address	Specifies an IPv6 access list to be used to match packets for PBR for IPv6.
match ipv6 next-hop	Distributes IPv6 routes that have a next-hop prefix permitted by a prefix list.
match ipv6 route-source	Distributes IPv6 routes that have been advertised by routers at an address specified by a prefix list.
match length	Bases policy routing on the Level 3 length of a packet.
match metric	Redistributes routes with the specified metric.
match route-type	Redistributes routes of the specified type.
route-map	Defines conditions for redistributing routes from one routing protocol into another.
set as-path	Modifies an autonomous system path for BGP routes.
set community	Sets the BGP community attribute.

Command	Description
set default interface	Specifies the default interface to output packets that pass a match clause of a route map for policy routing and have no explicit route to the destination.
set interface	Specifies the default interface to output packets that pass a match clause of a route map for policy routing.
set ipv6 default next-hop	Specifies an IPv6 default next hop to which matching packets will be forwarded.
set ipv6 next-hop (PBR)	Indicates where to output IPv6 packets that pass a match clause of a route map for policy routing.
set ipv6 precedence	Sets the precedence value in the IPv6 packet header.
set level	Indicates where to import routes.
set local preference	Specifies a preference value for the autonomous system path.
set metric	Sets the metric value for a routing protocol.
set metric-type	Sets the metric type for the destination routing protocol.
set tag	Sets a tag value of the destination routing protocol.
set weight	Specifies the BGP weight for the routing table.

match ipv6 destination

To configure the IPv6 destination address as a key field for a flow record, use the **match ipv6 destination** command in Flexible Netflow flow record configuration mode. To disable the IPv6 destination address as a key field for a flow record, use the **no** form of this command.

match ipv6 destination {**address**| {**mask**| **prefix**} [**minimum-mask** *mask*]}

no match ipv6 destination {**address**| {**mask**| **prefix**} [**minimum-mask** *mask*]}

Cisco Catalyst 6500 Switches in Cisco IOS Release 12.2(50)SY

match ipv6 destination address

no match ipv6 destination address

Cisco IOS XE Release 3.2SE

match ipv6 destination address

no match ipv6 destination address

Syntax Description

address	Configures the IPv6 destination address as a key field.
mask	Configures the mask for the IPv6 destination address as a key field.
prefix	Configures the prefix for the IPv6 destination address as a key field.
minimum-mask *mask*	(Optional) Specifies the size, in bits, of the minimum mask. Range: 1 to 128.

Command Default

The IPv6 destination address is not configured as a key field.

Command Modes

Flexible NetFlow flow record configuration (config-flow-record)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.2(33)SRE	This command was modified. Support for this command was implemented on the Cisco 7200 and Cisco 7300 Network Processing Engine (NPE) series routers.

Release	Modification
12.2(50)SY	This command was modified. The **mask**, **prefix**, and **minimum-mask** keywords were removed.
15.2(2)T	This command was modified. Support for the Cisco Performance Monitor was added.
Cisco IOS XE Release 3.5S	This command was modified. Support for the Cisco Performance Monitor was added.
Cisco IOS XE Release 3.2SE	This command was modified. The **mask**, **prefix**, and **minimum-mask** keywords were removed.

Usage Guidelines

This command can be used with both Flexible NetFlow and Performance Monitor. These products use different commands to enter the configuration mode in which you issue this command, however the mode prompt is the same for both products. For Performance Monitor, you must first enter the **flow record type performance-monitor** command before you can use this command.

Because the mode prompt is the same for both products, here we refer to the command mode for both products as flow record configuration mode. However, for Flexible NetFlow, the mode is also known as Flexible NetFlow flow record configuration mode; and for Performance Monitor, the mode is also known as Performance Monitor flow record configuration mode.

A flow record requires at least one key field before it can be used in a flow monitor. The key fields differentiate flows, with each flow having a unique set of values for the key fields. The key fields are defined using the **match** command.

Examples

The following example configures a 16-bit IPv6 destination address prefix as a key field:

```
Router(config)# flow record FLOW-RECORD-1
Router(config-flow-record)# match ipv6 destination prefix minimum-mask 16
```

The following example specifies a 16-bit IPv6 destination address mask as a key field:

```
Router(config)# flow record FLOW-RECORD-1
Router(config-flow-record)# match ipv6 destination mask minimum-mask 16
```

The following example configures a 16-bit IPv6 destination address mask as a key field:

```
Router(config)# flow record type performance-monitor RECORD-1
Router(config-flow-record)# match ipv6 destination mask minimum-mask 16
```

Related Commands

Command	Description
flow record	Creates a flow record, and enters Flexible NetFlow flow record configuration mode.
flow record type performance-monitor	Creates a flow record, and enters Performance Monitor flow record configuration mode.

match ipv6 hop-limit

To configure the IPv6 hop limit as a key field for a flow record, use the **match ipv6 hop-limit** command in Flexible NetFlow flow record configuration mode. To disable the use of a section of an IPv6 packet as a key field for a flow record, use the **no** form of this command.

match ipv6 hop-limit

no match ipv6 hop-limit

Syntax Description

This command has no arguments or keywords.

Command Default

The use of the IPv6 hop limit as a key field for a user-defined flow record is not enabled by default.

Command Modes

Flexible NetFlow flow record configuration (config-flow-record)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.2(33)SRE	This command was modified. Support for this command was implemented on the Cisco 7200 and Cisco 7300 Network Processing Engine (NPE) series routers.
15.2(2)T	This command was modified. Support for the Cisco Performance Monitor was added.
Cisco IOS XE Release 3.5S	This command was modified. Support for the Cisco Performance Monitor was added.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

This command can be used with both Flexible NetFlow and Performance Monitor. These products use different commands to enter the configuration mode in which you issue this command, however the mode prompt is the same for both products. For Performance Monitor, you must first enter the **flow record type performance-monitor** command before you can use this command.

Because the mode prompt is the same for both products, here we refer to the command mode for both products as flow record configuration mode. However, for Flexible NetFlow, the mode is also known as Flexible NetFlow flow record configuration mode; and for Performance Monitor, the mode is also known as Performance Monitor flow record configuration mode.

A flow record requires at least one key field before it can be used in a flow monitor. The key fields differentiate flows, with each flow having a unique set of values for the key fields. The key fields are defined using the **match** command.

Examples

The following example configures the hop limit of the packets in the flow as a key field:

```
Router(config)# flow record FLOW-RECORD-1
Router(config-flow-record)# match ipv6 hop-limit
```

The following example configures the hop limit of the packets in the flow as a key field:

```
Router(config)# flow record type performance-monitor RECORD-1
Router(config-flow-record)# match ipv6 hop-limit
```

Related Commands

Command	Description
flow record	Creates a flow record, and enters Flexible NetFlow flow record configuration mode.
flow record type performance-monitor	Creates a flow record, and enters Performance Monitor flow record configuration mode.

match ra prefix-list

To verify the advertised prefixes in inspected messages from the authorized prefix list, use the **match ra prefix-list** command in RA guard policy configuration mode.

match ra prefix-list *ipv6-prefix-list-name*

Syntax Description

ipv6-prefix-list-name	The IPv6 prefix list to be matched.

Command Default

Advertised prefixes are not verified.

Command Modes

RA guard policy configuration (config-ra-guard)

Command History

Release	Modification
12.2(50)SY	This command was introduced.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.
15.0(2)SE	This command was integrated into Cisco IOS Release 15.0(2)SE.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

The **match ra prefix-list** command enables verification of the advertised prefixes in inspected messages from the configured authorized prefix list. Use the **ipv6 prefix-list** command to configure an IPv6 prefix list. For instance, to authorize the 2001:101::/64 prefixes and deny the 2001:100::/64 prefixes, define the following IPv6 prefix list:

```
Router(config)# ipv6 prefix-list listname1 deny 2001:0DB8:101:/64
Router(config)# ipv6 prefix-list listname1 permit 2001:0DB8:100::/64
```

Examples

The following example shows how the command defines an router advertisement (RA) guard policy name as raguard1, places the router in RA guard policy configuration mode, and verifies the advertised prefixes in listname1:

```
Router(config)# ipv6 nd raguard policy raguard1
Router(config-ra-guard)# match ra prefix-list listname1
```

Related Commands

Command	Description
ipv6 nd raguard policy	Defines the RA guard policy name and enters RA guard policy configuration mode.
ipv6 prefix-list	Creates an entry in an IPv6 prefix list.

max-through

To limit multicast router advertisements (RAs) per VLAN per throttle period, use the **max-through** command in IPv6 RA throttle policy configuration mode. To reset the command to its defaults, use the **no** form of this command.

max-through {*mt-value*| **inherit**| **no-limit**}

Syntax Description

mt-value	Number of multicast RAs allowed on the VLAN before throttling occurs. The range is from 0 through 256.
inherit	Merges the setting between target policies.
no-limit	Multicast RAs are not limited on the VLAN.

Command Default

10 RAs per VLAN per 10 minutes

Command Modes

IPv6 RA throttle policy configuration (config-nd-ra-throttle)

Command History

Release	Modification
Cisco IOS XE Release 3.2XE	This command was introduced.

Usage Guidelines

The **max-through** command limits the amount of multicast RAs that are passed through to the VLAN per throttle period. This command can be configured only on a VLAN.

Examples

```
Device(config)# ipv6 nd ra-throttle policy policy1
Device(config-nd-ra-throttle)# max-through 25
```

medium-type

To indicate whether a device is wired or wireless, use the **medium-type** command in IPv6 RA throttle policy configuration mode. To reset the command to its defaults, use the **no** form of this command.

medium-type {**access-point**| **wired**}

Syntax Description

access-point	The attached device is a radio access point and is throttled.
wired	The attached device is wired and is not throttled.

Command Default

Wired

Command Modes

IPv6 RA throttle policy configuration (config-nd-ra-throttle)

Command History

Release	Modification
Cisco IOS XE Release XE3.2S	This command was introduced.

Usage Guidelines

The **medium-type** command indicates the type of access on a port only. The VLAN ignores any values specified by the **medium-type** command.

Examples

```
Device(config)# ipv6 nd ra-throttle policy policy1
Device(config-nd-ra-throttle)# medium-type wired
```

mode dad-proxy

To enable duplicate address detection (DAD) proxy mode for IPv6 Neighbor Discovery (ND) suppress, use the **mode dad-proxy** command in ND suppress policy configuration mode. To disable this feature, use the **no** form of this command.

mode dad-proxy

Syntax Description

This command has no arguments or keywords.

Command Default

All multicast neighbor solicitation (NS) messages are suppressed.

Command Modes

ND suppress policy configuration mode (config-nd-suppress)

Command History

Release	Modification
15.1(2)SG	This command was introduced.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

The IPv6 Dad proxy feature responds on behalf of the address's owner when an address is already in use. Use the **mode dad-proxy** command to enable IPv6 DAD proxy when using IPv6 ND suppress. If your device does not support IPv6 multicast suppress, you can enable IPv6 DAD proxy by entering the **ipv6 nd dad-proxy** command in global configuration mode.

Examples

```
Device(config)# ipv6 nd suppress policy policy1
Device(config-nd-suppress)# mode dad-proxy
```

Related Commands

Command	Description
ipv6 nd dad-proxy	Enables the IPv6 ND DAD proxy feature on the device.
ipv6 nd suppress policy	Enables IPv6 ND multicast suppress and enters ND suppress policy configuration mode.

network (IPv6)

To configure the network source of the next hop to be used by the PE VPN, use the network command in router configuration mode. To disable the source, use the **no** form of this command.

network *ipv6-address/prefix-length*

no network *ipv6-address/prefix-length*

Syntax Description

ipv6-address	The IPv6 address to be used.
/ *prefix-length*	The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.

Command Default

Next-hop network sources are not configured.

Command Modes

Address family configuration Router configuration

Command History

Release	Modification
12.2(33)SRB	This command was introduced.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.
Cisco IOS XE Release 3.1S	This command was integrated into Cisco IOS XE Release 3.1S.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

The *ipv6-address* argument in this command configures the IPv6 network number.

Examples

The following example places the router in address family configuration mode and configures the network source to be used as the next hop:

```
Router(config)# router bgp 100
Router(config-router)# network 2001:DB8:100::1/128
```

Related Commands

Command	Description
address-family ipv6	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv6 address prefixes.
address-family vpnv6	Places the router in address family configuration mode for configuring routing sessions that use standard VPNv6 address prefixes.

other-config-flag

To verify the advertised “other” configuration parameter, use the **other-config-flag** command in RA guard policy configuration mode.

other-config-flag {**on**| **off**}

Syntax Description

on	Verification is enabled.
off	Verification is disabled.

Command Default

Verification is not enabled.

Command Modes

RA guard policy configuration (config-ra-guard)

Command History

Release	Modification
12.2(50)SY	This command was introduced.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.
15.0(2)SE	This command was integrated into Cisco IOS Release 15.0(2)SE.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

The **other-config-flag** command enables verification of the advertised "other" configuration parameter (or "O" flag). This flag could be set by an attacker to force hosts to retrieve other configuration information through a Dynamic Host Configuration Protocol for IPv6 (DHCPv6) server that may not be trustworthy.

Examples

The following example shows how the command defines a router advertisement (RA) guard policy name as raguard1, places the router in RA guard policy configuration mode, and enables O flag verification:

```
Router(config)# ipv6 nd raguard policy raguard1
Router(config-ra-guard)# other-config-flag on
```

Related Commands

Command	Description
ipv6 nd raguard policy	Defines the RA guard policy name and enters RA guard policy configuration mode.

passive-interface (IPv6)

To disable sending routing updates on an interface, use the **passive-interface** command in router configuration mode. To reenable the sending of routing updates, use the **no** form of this command.

passive-interface [**default**| *interface-type interface-number*]

no passive-interface [**default**| *interface-type interface-number*]

Syntax Description

default	(Optional) All interfaces become passive.
interface-type interface-number	(Optional) Interface type and number. For more information, use the question mark (**?**) online help function.

Command Default

No interfaces are passive. Routing updates are sent to all interfaces on which the routing protocol is enabled.

Command Modes

Router configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.4(6)T	Support for Enhanced Internal Gateway Routing Protocol (EIGRP) IPv6 was added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

If you disable the sending of routing updates on an interface, the particular address prefix will continue to be advertised to other interfaces, and updates from other routers on that interface continue to be received and processed.

The **default** keyword sets all interfaces as passive by default. You can then configure individual interfaces where adjacencies are desired using the **no passive-interface** command. The **default** keyword is useful in Internet service provider (ISP) and large enterprise networks where many of the distribution routers have more than 200 interfaces.

OSPF for IPv6 routing information is neither sent nor received through the specified router interface. The specified interface address appears as a stub network in the OSPF for IPv6 domain.

For the Intermediate System-to-Intermediate System (IS-IS) protocol, this command instructs IS-IS to advertise the IP addresses for the specified interface without actually running IS-IS on that interface. The **no** form of this command for IS-IS disables advertising IP addresses for the specified address.

Examples

The following example sets all interfaces as passive, then activates Ethernet interface 0:

```
Router(config-router)# passive-interface default
Router(config-router)# no passive-interface ethernet0/0
```

passive-interface (OSPFv3)

To suppress sending routing updates on an interface when using an IPv4 Open Shortest Path First version 3 (OSPFv3) process, use the **passive-interface** command in router configuration mode. To reenable the sending of routing updates, use the **no** form of this command.

passive-interface [**default**| *interface-type interface-number*]

no passive-interface [**default**| *interface-type interface-number*]

Syntax Description

default	(Optional) All interfaces become passive.
interface-type interface-number	(Optional) Interface type and number. For more information, use the question mark (**?**) online help function.

Command Default

No interfaces are passive. Routing updates are sent to all interfaces on which the routing protocol is enabled.

Command Modes

OSPFv3 router configuration mode (config-router)

Command History

Release	Modification
15.1(3)S	This command was introduced.
Cisco IOS XE Release 3.4S	This command was integrated into Cisco IOS XE Release 3.4S.
15.2(1)T	This command was integrated into Cisco IOS Release 15.2(1)T.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

If you suppress the sending of routing updates on an interface, the particular address prefix will continue to be advertised to other interfaces, and updates from other routers on that interface continue to be received and processed.

The **default** keyword sets all interfaces as passive by default. You can then configure individual interfaces where adjacencies are desired using the **no passive-interface** command. The **default** keyword is useful in Internet service provider (ISP) and large enterprise networks where many of the distribution routers have more than 200 interfaces.

Examples

The following example sets all interfaces as passive, then activates Ethernet interface 0/0:

```
Router(config-router)# passive-interface default
Router(config-router)# no passive-interface ethernet0/0
```

Related Commands

Command	Description
default (OSPFv3)	Returns an OSPFv3 parameter to its default value.
router ospfv3	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.

permit (IPv6)

To set permit conditions for an IPv6 access list, use the **permit** command in IPv6 access list configuration mode. To remove the permit conditions, use the **no** form of this command.

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] [hbh] [log] [log-input] [mobility]** [**mobility-type** [*mh-number*| *mh-type*]] [**reflect** *name* [**timeout** *value*]] **[routing]** [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]

no 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] [hbh] [log] [log-input] [mobility]** [**mobility-type** [*mh-number*| *mh-type*]] [**reflect** *name* [**timeout** *value*]] **[routing]** [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]

Internet Control Message Protocol

permit icmp {*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*]] [*icmp-type* [*icmp-code*]| *icmp-message*] [**dest-option-type** [*doh-number*| *doh-type*]] [**dscp** *value*] [**flow-label** *value*] **[fragments] [hbh] [log] [log-input] [mobility]** [**mobility-type** [*mh-number*| *mh-type*]] **[routing]** [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]

Transmission Control Protocol

permit tcp {*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*]] **[ack]** [**dest-option-type** [*doh-number*| *doh-type*]] [**dscp** *value*] **[established] [fin]** [**flow-label** *value*] **[fragments] [hbh] [log] [log-input] [mobility]** [**mobility-type** [*mh-number*| *mh-type*]] [**neq** {*port*| *protocol*}] **[psh]** [**range** {*port*| *protocol*}] [**reflect** *name* [**timeout** *value*]] **[routing]** [**routing-type** *routing-number*] **[rst]** [**sequence** *value*] **[syn]** [**time-range** *name*] **[urg]**

User Datagram Protocol

permit udp {*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] [hbh] [log] [log-input] [mobility]** [**mobility-type** [*mh-number*| *mh-type*]] [**neq** {*port*| *protocol*}] [**range** {*port*| *protocol*}] [**reflect** *name* [**timeout** *value*]] **[routing]** [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]

Syntax Description

protocol	Name or number of an Internet protocol. It can be one of the keywords **ahp**, **esp**, **icmp**, **ipv6**, **pcp**, **sctp**, **tcp**, **udp**, or **hbh**, or an integer in the range from 0 to 255 representing an IPv6 protocol number.

source-ipv6-prefix/*prefix-length*	The source IPv6 network or class of networks about which to set permit conditions. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
any	An abbreviation for the IPv6 prefix ::/0.
host *source-ipv6-address*	The source IPv6 host address about which to set permit conditions. This *source-ipv6-address* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
auth	Allows matching traffic against the presence of the authentication header in combination with any protocol.
operator [*port-number*]	(Optional) Specifies an operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). If the operator is positioned after the *source-ipv6-prefix*/*prefix-length* argument, it must match the source port. If the operator is positioned after the *destination-ipv6-prefix*/*prefix-length* argument, it must match the destination port. The **range** operator requires two port numbers. All other operators require one port number. The optional *port-number* argument is a decimal number or the name of a TCP or UDP port. A port number is a number from 0 to 65535. TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.
destination-ipv6-prefix/ *prefix-length*	The destination IPv6 network or class of networks about which to set permit conditions. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.

host *destination-ipv6-address*	The destination IPv6 host address about which to set permit conditions. This *destination-ipv6-address* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
dest-option-type	(Optional) Matches IPv6 packets against the destination extension header within each IPv6 packet header.
doh-number	(Optional) Integer in the range from 0 to 255 representing an IPv6 destination option extension header.
doh-type	(Optional) Destination option header types. The possible destination option header type and its corresponding *doh-number* value are home-address—201.
dscp *value*	(Optional) Matches a differentiated services codepoint value against the traffic class value in the Traffic Class field of each IPv6 packet header. The acceptable range is from 0 to 63.
flow-label *value*	(Optional) Matches a flow label value against the flow label value in the Flow Label field of each IPv6 packet header. The acceptable range is from 0 to 1048575.
fragments	(Optional) Matches non-initial fragmented packets where the fragment extension header contains a non-zero fragment offset. The **fragments** keyword is an option only if the *operator* [*port-number*] arguments are not specified. When this keyword is used, it also matches when the first fragment does not have Layer 4 information.
hbh	(Optional) Matches IPv6 packets against the hop-by-hop extension header within each IPv6 packet header.

log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) The message includes the access list name and sequence number, whether the packet was permitted; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted in the prior 5-minute interval.
log-input	(Optional) Provides the same function as the **log** keyword, except that the logging message also includes the input interface.
mobility	(mobility) Matches IPv6 packets against the mobility extension header within each IPv6 packet header.
mobility-type	(Optional) Matches IPv6 packets against the mobility-type extension header within each IPv6 packet header. Either the *mh-number* or *mh-type* argument must be used with this keyword.
mh-number	(Optional) Integer in the range from 0 to 255 representing an IPv6 mobility header type.
mh-type	(Optional) Mobility header types. Possible mobility header types and their corresponding *mh-number* value are as follows: • 0—bind-refresh • 1—hoti • 2—coti • 3—hot • 4—cot • 5—bind-update • 6—bind-acknowledgment • 7—bind-error

reflect *name*	(Optional) Specifies a reflexive IPvi6 access list. Reflexive IPv6 access lists are created dynamically when an IPv6 packets matches a permit statement that contains the **reflect** keyword. The reflexive IPv6 access list mirrors the permit statement and times out automatically when no IPv6 packets match the permit statement. Reflexive IPv6 access lists can be applied to the TCP, UDP, SCTP, and ICMP for IPv6 packets.
timeout *value*	(Optional) Interval of idle time (in seconds) after which a reflexive IPv6 access list times out. The acceptable range is from 1 to 4294967295. The default is 180 seconds.
routing	(Optional) Matches source-routed packets against the routing extension header within each IPv6 packet header.
routing-type	(Optional) Matches IPv6 packets against the routing-type extension header within each IPv6 packet header. The *routing-number* argument must be used with this keyword.
routing-number	Integer in the range from 0 to 255 representing an IPv6 routing header type. Possible routing header types and their corresponding *routing-number* value are as follows: • 0—Standard IPv6 routing header • 2—Mobile IPv6 routing header
sequence *value*	(Optional) Specifies the sequence number for the access list statement. The acceptable range is from 1 to 4294967295.
time-range *name*	(Optional) Specifies the time range that applies to the permit statement. The name of the time range and its restrictions are specified by the **time-range** and **absolute** or **periodic** commands, respectively.

icmp-type	(Optional) Specifies an ICMP message type for filtering ICMP packets. ICMP packets can be filtered by ICMP message type. The ICMP message type can be a number from 0 to 255, some of which include the following predefined strings and their corresponding numeric values: • 144—dhaad-request • 145—dhaad-reply • 146—mpd-solicitation • 147—mpd-advertisement
icmp-code	(Optional) Specifies an ICMP message code for filtering ICMP packets. ICMP packets that are filtered by ICMP message type can also be filtered by the ICMP message code. The code is a number from 0 to 255.
icmp-message	(Optional) Specifies an ICMP message name for filtering ICMP packets. ICMP packets can be filtered by an ICMP message name or ICMP message type and code. The possible names are listed in the "Usage Guidelines" section.
ack	(Optional) For the TCP protocol only: acknowledgment (ACK) bit set.
established	(Optional) For the TCP protocol only: Indicates an established connection. A match occurs if the TCP datagram has the ACK or RST bits set. The nonmatching case is that of the initial TCP datagram to form a connection.
fin	(Optional) For the TCP protocol only: Fin bit set; no more data from sender.
neq {*port* | *protocol*}	(Optional) Matches only packets that are not on a given port number.
psh	(Optional) For the TCP protocol only: Push function bit set.
{**range** *port* | *protocol*}	(Optional) Matches only packets in the range of port numbers.
rst	(Optional) For the TCP protocol only: Reset bit set.
syn	(Optional) For the TCP protocol only: Synchronize bit set.

urg	(Optional) For the TCP protocol only: Urgent pointer bit set.

Command Default No IPv6 access list is defined.

Command Modes IPv6 access list configuration (config-ipv6-acl)#

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.4(2)T	The *icmp-type* argument was enhanced. The **dest-option-type**, **mobility**, **mobility-type**, and **routing-type** keywords were added. The *doh-number*, *doh-type*, *mh-number*, *mh-type*, and *routing-number* arguments were added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 series routers.
12.4(20)T	The **auth** keyword was added.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.
15.2(3)T	This command was modified. Support was added for the **hbh** keyword.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines The **permit** (IPv6) command is similar to the **permit** (IP) command, except that it is IPv6-specific.

Use the **permit** (IPv6) command following the **ipv6 access-list** command to define the conditions under which a packet passes the access list or to define the access list as a reflexive access list.

Specifying IPv6 for the *protocol* argument matches against the IPv6 header of the packet.

By default, the first statement in an access list is number 10, and the subsequent statements are incremented by 10.

You can add **permit**, **deny**, **remark**, or **evaluate** statements to an existing access list without retyping the entire list. To add a new statement anywhere other than at the end of the list, create a new statement with an appropriate entry number that falls between two existing entry numbers to indicate where it belongs.

In Cisco IOS Release 12.2(2)T or later releases, 12.0(21)ST, and 12.0(22)S, IPv6 access control lists (ACLs) are defined and their deny and permit conditions are set by using the **ipv6 access-list** command with the **deny** and **permit** keywords in global configuration mode. In Cisco IOS Release 12.0(23)S or later releases, IPv6 ACLs are defined by using the **ipv6 access-list** command in global configuration mode and their permit and deny conditions are set by using the **deny** and **permit** commands in IPv6 access list configuration mode. Refer to the **ipv6 access-list** command for more information on defining IPv6 ACLs.

Note In Cisco IOS Release 12.0(23)S or later releases, every IPv6 ACL has implicit **permit icmp any any nd-na**, **permit icmp any any nd-ns**, and **deny ipv6 any any** statements as its last match conditions. (The former two match conditions allow for ICMPv6 neighbor discovery.) An IPv6 ACL must contain at least one entry for the implicit **deny ipv6 any any** statement to take effect. 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.

Both the *source-ipv6-prefix*/*prefix-length* and *destination-ipv6-prefix*/*prefix-length* arguments are used for traffic filtering (the source prefix filters traffic based upon the traffic source; the destination prefix filters traffic based upon the traffic destination).

Note IPv6 prefix lists, not access lists, should be used for filtering routing protocol prefixes.

The **fragments** keyword is an option only if the *operator* [*port-number*] arguments are not specified.

The following is a list of ICMP message names:

- beyond-scope
- destination-unreachable
- echo-reply
- echo-request
- header
- hop-limit
- mld-query
- mld-reduction
- mld-report
- nd-na
- nd-ns

- next-header
- no-admin
- no-route
- packet-too-big
- parameter-option
- parameter-problem
- port-unreachable
- reassembly-timeout
- renum-command
- renum-result
- renum-seq-number
- router-advertisement
- router-renumbering
- router-solicitation
- time-exceeded
- unreachable

Defining Reflexive Access Lists

To define an IPv6 reflexive list, a form of session filtering, use the **reflect** keyword in the **permit** (IPv6) command. The **reflect** keyword creates an IPv6 reflexive access list and triggers the creation of entries in the reflexive access list. The **reflect** keyword must be an entry (condition statement) in an IPv6 access list.

Note For IPv6 reflexive access lists to work, you must nest the reflexive access list using the **evaluate** command.

If you are configuring IPv6 reflexive access lists for an external interface, the IPv6 access list should be one that is applied to outbound traffic.

If you are configuring an IPv6 reflexive access list for an internal interface, the IPv6 access list should be one that is applied to inbound traffic.

IPv6 sessions that originate from within your network are initiated with a packet exiting your network. When such a packet is evaluated against the statements in the IPv6 access list, the packet is also evaluated against the IPv6 reflexive permit entry.

As with all IPv6 access list entries, the order of entries is important, because they are evaluated in sequential order. When an IPv6 packet reaches the interface, it will be evaluated sequentially by each entry in the access list until a match occurs.

If the packet matches an entry prior to the reflexive permit entry, the packet will not be evaluated by the reflexive permit entry, and no temporary entry will be created for the reflexive access list (session filtering will not be triggered).

The packet will be evaluated by the reflexive permit entry if no other match occurs first. Then, if the packet matches the protocol specified in the reflexive permit entry, the packet is forwarded and a corresponding

temporary entry is created in the reflexive access list (unless the corresponding entry already exists, indicating that the packet belongs to a session in progress). The temporary entry specifies criteria that permit traffic into your network only for the same session.

Characteristics of Reflexive Access List Entries

The **permit** (IPv6) command with the **reflect** keyword enables the creation of temporary entries in the same IPv6 reflexive access list that was defined by the **permit** (IPv6) command. The temporary entries are created when an IPv6 packet exiting your network matches the protocol specified in the **permit** (IPv6) command. (The packet "triggers" the creation of a temporary entry.) These entries have the following characteristics:

- The entry is a permit entry.
- The entry specifies the same IP upper-layer protocol as the original triggering packet.
- The entry specifies the same source and destination addresses as the original triggering packet, except that the addresses are swapped.
- If the original triggering packet is TCP or UDP, the entry specifies the same source and destination port numbers as the original packet, except that the port numbers are swapped.
- If the original triggering packet is a protocol other than TCP or UDP, port numbers do not apply, and other criteria are specified. For example, for ICMP, type numbers are used: The temporary entry specifies the same type number as the original packet (with only one exception: if the original ICMP packet is type 8, the returning ICMP packet must be type 0 to be matched).
- The entry inherits all the values of the original triggering packet, with exceptions only as noted in the previous four bullets.
- IPv6 traffic entering your internal network will be evaluated against the entry, until the entry expires. If an IPv6 packet matches the entry, the packet will be forwarded into your network.
- The entry will expire (be removed) after the last packet of the session is matched.
- If no packets belonging to the session are detected for a configured length of time (the timeout period), the entry will expire.

Examples

The following example configures two IPv6 access lists named OUTBOUND and INBOUND and applies both access lists to outbound and inbound traffic on Ethernet interface 0. The first and second permit entries in the OUTBOUND list permit all TCP and UDP packets from network 2001:ODB8:0300:0201::/64 to exit out of Ethernet interface 0. The entries also configure the temporary IPv6 reflexive access list 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 third permit entry in the OUTBOUND list permits all ICMP packets to exit out of Ethernet interface 0.

The permit entry in the INBOUND list permits all ICMP packets to enter Ethernet interface 0. The **evaluate** command in the list applies the temporary IPv6 reflexive access list 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. Refer to the **evaluate** command for more information on nesting IPv6 reflexive access lists within IPv6 ACLs.

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

```
 deny FEC0:0:0:0201::/64 any
 permit icmp any any
ipv6 access-list INBOUND
 permit icmp any any
 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 access list, only TCP, UDP, and ICMP packets will be permitted out of and in to Ethernet interface 0 (the implicit deny all condition at the end of the access list denies all other packet types on the interface).

The following example shows how to allow the matching of any UDP traffic. The authentication header may be present.

```
permit udp any any sequence 10
```

The following example shows how to allow the matching of only TCP traffic if the authentication header is also present.

```
permit tcp any any auth sequence 20
```

The following example shows how to allow the matching of any IPv6 traffic where the authentication header is present.

```
permit ahp any any sequence 30
```

Related Commands

Command	Description
deny (IPv6)	Sets deny conditions for an IPv6 access list.
evaluate (IPv6)	Nests an IPv6 reflexive access list within an IPv6 access list.
ipv6 access-list	Defines an IPv6 access list and enters IPv6 access list configuration mode.
ipv6 traffic-filter	Filters incoming or outgoing IPv6 traffic on an interface.
show ipv6 access-list	Displays the contents of all current IPv6 access lists.

prefix-glean

To enable the device to glean prefixes from IPv6 router advertisements (RAs) or Dynamic Host Configuration Protocol (DHCP), use the **prefix-glean** command in IPv6 snooping configuration mode. To learn only prefixes gleaned in one of these protocols and exclude the other, use the **no** form of this command.

prefix-glean [**only**]

no prefix-glean [**only**]

Syntax Description

only	(Optional) Only prefixes are gleaned.

Command Default

Prefixes are not learned through RA or DHCP.

Command Modes

IPv6 snooping configuration mode (config-ipv6-snooping)

Command History

Release	Modification
15.0(2)SE	This command was introduced.
15.3(1)S	This command was integrated into Cisco IOS Release 15.3(1)S.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

The **prefix-glean** command enables the device to learn prefixes in RA and DHCP traffic.

Examples

The following example shows how to enable the device to learn prefixes:

```
Device(config)# ipv6 snooping policy policy1
Device(config-ipv6-snooping)# prefix-glean
```

Related Commands

Command	Description
ipv6 snooping attach-policy	Applies an IPv6 snooping policy to a target.
ipv6 snooping policy	Configures an IPv6 snooping policy and enters IPv6 snooping configuration mode.

protocol (IPv6)

To specify that addresses should be gleaned with Dynamic Host Configuration Protocol (DHCP) or Neighbor Discovery Protocol (NDP) or to associate the protocol with an IPv6 prefix list, use the **protocol** command. To disable address gleaning with DHCP or NDP, use the **no** form of the command.

protocol {**dhcp** | **ndp**} [**prefix-list** *prefix-list-name*]

no protocol {**dhcp** | **ndp**}

Syntax Description

dhcp	Specifies that addresses should be gleaned in Dynamic Host Configuration Protocol (DHCP) packets.
ndp	Specifies that addresses should be gleaned in Neighbor Discovery Protocol (NDP) packets.
prefix-list *prefix-list-name*	(Optional) Specifies that a prefix list of protected prefixes be used.

Command Default

Snooping and recovery are attempted using both DHCP and NDP. No prefix list is used, all address ranges are accepted.

Command Modes

IPv6 snooping configuration mode (config-ipv6-snooping)

Command History

Release	Modification
15.2(4)S	This command was introduced.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

If an address does not match the prefix list associated with DHCP or NDP, then control packets will be dropped and recovery of the binding table entry will not be attempted with that protocol.

- If there is no prefix list specified, all protocols are supported by default. There is no check and all addresses are accepted.
- Using the **no protocol** {**dhcp** | **ndp**} command indicates that a protocol will not to be used for snooping or gleaning.
- However, if the **no protocol dhcp** command is used, DHCP can still be used for binding table recovery.
- The NDP prefix list should be a superset of the DHCP prefix list, as addresses obtained by DHCP must be confirmed by NDP later.

- When a prefix list is given and a protocol packet indicates an address that does not match the prefix list for that protocol, the packet is dropped (unless the security level is “glean”).
- Data glean can recover with DHCP and NDP, though destination guard will only recovery through DHCP.

Note Before you configure the **protocol** command, it is essential that you provide a value for the **ge** *ge-value* option when configuring a prefix list using the **ipv6 prefix-list** command.

Examples

The following example shows a valid configuration for an IPv6 prefix list (“abc”) and shows that DHCP will be used to recover addresses that match the prefix list abc:

```
Device(config)# ipv6 prefix-list abc seq 5 permit 2001:DB8::/64 ge 128
!
Device(config-ipv6-snooping)# protocol dhcp prefix-list abc
```

Related Commands

Command	Description
ipv6 prefix-list	Creates an entry in an IPv6 prefix list.
ipv6 snooping policy	Enters IPv6 snooping configuration mode.

redistribute (IPv6)

To redistribute IPv6 routes from one routing domain into another routing domain, use the **redistribute** command in address family configuration or router configuration mode. To disable redistribution, use the **no** form of this command.

redistribute source-protocol [*process-id*] [**include-connected** {**level-1**| **level-1-2**| **level-2**}] [*as-number*] [**metric** {*metric-value*| **transparent**}] [**metric-type** *type-value*] [**match** {**external** [**1**| **2**]| **internal**| **nssa-external** [**1**| **2**]}] [**tag** *tag-value*] [**route-map** *map-tag*]

no redistribute source-protocol [*process-id*] **[include-connected]** {**level-1**| **level-1-2**| **level-2**} [*as-number*] [**metric** {*metric-value*| **transparent**}] [**metric-type** *type-value*] [**match** {**external** [**1**| **2**]| **internal**| **nssa-external** [**1**| **2**]}] [**tag** *tag-value*] [**route-map** *map-tag*]

Syntax Description

source-protocol	Source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, connected, **eigrp**, **isis**, **ospf**, **rip**, or **static**.
process-id	(Optional) For the **bgp** or **eigrp**keyword, the process ID is a Border Gateway Protocol (BGP) autonomous system number, which is a 16-bit decimal number. For the **isis**keyword, the process ID is an optional value that defines a meaningful name for a routing process. You can specify only one IS-IS process per router. Creating a name for a routing process means that you use names when configuring routing. For the **ospf** keyword, the process ID is the number assigned administratively when the Open Shortest Path First (OSPF) for IPv6 routing process is enabled. For the **rip**keyword, the process ID is an optional value that defines a meaningful name for an IPv6 Routing Information Protocol (RIP) routing process.
include-connected	(Optional) Allows the target protocol to redistribute routes learned by the source protocol and connected prefixes on those interfaces over which the source protocol is running.
level-1	Specifies that, for Intermediate System-to-Intermediate System (IS-IS), Level 1 routes are redistributed into other IP routing protocols independently.
level-1-2	Specifies that, for IS-IS, both Level 1 and Level 2 routes are redistributed into other IP routing protocols.

level-2	Specifies that, for IS-IS, Level 2 routes are redistributed into other IP routing protocols independently.
as-number	(Optional) Autonomous system number for the redistributed route.
metric *metric-value*	(Optional) When redistributing from one OSPF process to another OSPF process on the same router, the metric will be carried through from one process to the other if no metric value is specified. When redistributing other processes to an OSPF process, the default metric is 20 when no metric value is specified.
metric transparent	(Optional) Causes RIP to use the routing table metric for redistributed routes as the RIP metric.
metric-type *type-value*	(Optional) For OSPF, specifies the external link type associated with the default route advertised into the OSPF routing domain. It can be one of two values: • **1** --Type 1 external route • **2** --Type 2 external route If no value is specified for the **metric-type** keyword, the Cisco IOS software adopts a Type 2 external route. For IS-IS, the link type can be one of two values: • **internal** --IS-IS metric that is < 63. • **external** --IS-IS metric that is > 64 < 128. The default is **internal**.
match {**external [1 | 2]**| **internal | nssa-external [1 | 2]**	(Optional) For OSPF, routes are redistributed into other routing domains using the match keyword. It is used with one of the following: • **external [1 | 2]** --Routes that are external to the autonomous system, but are imported into OSPF as Type 1 or Type 2 external routes. • **internal** --Routes that are internal to a specific autonomous system. • **nssa-external** [**1**| **2**]-- Routes that are external to the autonomous system but are imported into OSPF, in a not so stubby area (NSSA), for IPv6 as Type 1 or Type 2 external routes.

tag *tag-value*	(Optional) Specifies the 32-bit decimal value attached to each external route. This is not used by OSPF itself. It may be used to communicate information between Autonomous System Boundary Routers (ASBRs). If none is specified, then the remote autonomous system number is used for routes from BGP and Exterior Gateway Protocol (EGP); for other protocols, zero (0) is used.
route-map	(Optional) Specifies the route map that should be checked to filter the importation of routes from this source routing protocol to the current routing protocol. If the route-map keyword is not specified, all routes are redistributed. If this keyword is specified, but no route map tags are listed, no routes will be imported.
map-tag	(Optional) Identifier of a configured route map.

Command Default Route redistribution is disabled.

Command Modes Address family configuration Router configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.4(6)T	Support for Enhanced Internal Gateway Routing Protocol (EIGRP) IPv6 was added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines Changing or disabling any keyword will not affect the state of other keywords.

A router receiving an IPv6 IS-IS route with an internal metric will consider the cost of the route from itself to the redistributing router plus the advertised cost to reach the destination. An external metric considers only the advertised metric to reach the destination.

IS-IS will ignore any configured redistribution of routes configured with the include-connected keyword. IS-IS will advertise a prefix on an interface if either IS-IS is running over the interface or the interface is configured as passive.

Routes learned from IPv6 routing protocols can be redistributed into IPv6 IS-IS at Level 1 into an attached area or at Level 2. The **level-1-2** keyword allows both Level 1 and Level 2 routes in a single command.

For IPv6 RIP, use the **redistribute**command to advertise static routes as if they were directly connected routes.

Caution Advertising static routes as directly connected routes can cause routing loops if improperly configured.

Redistributed IPv6 RIP routing information should always be filtered by the **distribute-list prefix-list**router configuration command. Use of the **distribute-list prefix-list**command ensures that only those routes intended by the administrator are passed along to the receiving routing protocol.

Note The **metric** value specified in the **redistribute** command for IPv6 RIP supersedes the **metric** value specified using the **default-metric** command.

Note In IPv4, if you redistribute a protocol, by default you also redistribute the subnet on the interfaces over which the protocol is running. In IPv6 this is not the default behavior. To redistribute the subnet on the interfaces over which the protocol is running in IPv6, use the include-connected keyword. In IPv6 this functionality is not supported when the source protocol is BGP.

When the no redistribute command is configured, the parameter settings are ignored when the client protocol is IS-IS or EIGRP.

IS-IS redistribution will be removed completely when IS-IS level 1 and level 2 are removed by the user. IS-IS level settings can be configured using the redistribute command only.

The default redistribute type will be restored to OSPF when all route type values are removed by the user.

Examples

The following example configures IPv6 IS-IS to redistribute IPv6 BGP routes. The metric is specified as 5, and the metric type will be set to external, indicating that it has lower priority than internal metrics.

```
Router(config)# router isis
Router(config-router)# address-family ipv6
Router(config-router-af)# redistribute bgp 64500 metric 5 metric-type external
```

The following example redistributes IPv6 BGP routes into the IPv6 RIP routing process named cisco:

```
Router(config)# ipv6 router rip cisco
Router(config-router)# redistribute bgp 42
```

The following example redistributes IS-IS for IPv6 routes into the OSPF for IPv6 routing process 1:

```
Router(config)# ipv6 router ospf 1
Router(config-router)# redistribute isis 1 metric 32 metric-type 1 tag 85
```

In the following example, ospf 1 redistributes the prefixes 2001:1:1::/64 and 2001:99:1::/64 and any prefixes learned through rip 1:

```
interface ethernet0/0
 ipv6 address 2001:1:1::90/64
 ipv6 rip 1 enable
interface ethernet1/1
 ipv6 address 2001:99:1::90/64
 ipv6 rip 1 enable
interface ethernet2/0
 ipv6 address 2001:1:2::90/64
 ipv6 ospf 1 area 1
ipv6 router ospf 1
  redistribute rip 1 include-connected
```

The following configuration example and output show the no redistribute command parameters when the last route type value is removed:

```
Router(config-router)# redistribute rip process1 metric 7
Router(config-router)# do show run | include redistribute
 redistribute rip process1 metric 7
Router(config-router)# no redistribute rip process1 metric 7
Router(config-router)# do show run | include redistribute
 redistribute rip process1
Router(config-router)#
```

Related Commands

Command	Description
default-metric	Specifies a default metric for redistributed routes.
distribute-list prefix-list (IPv6 EIGRP)	Applies a prefix list to EIGRP for IPv6 routing updates that are received or sent on an interface.
distribute-list prefix-list (IPv6 RIP)	Applies a prefix list to IPv6 RIP routing updates that are received or sent on an interface.
redistribute isis (IPv6)	Redistributes IPv6 routes from one routing domain into another routing domain using IS-IS as both the target and source protocol.

router-preference maximum

To verify the advertised default router preference parameter value, use the **router-preference maximum** command in RA guard policy configuration mode.

router-preference maximum {**high**| **low**| **medium**}

Syntax Description

high	Default router preference parameter value is higher than the specified limit.
medium	Default router preference parameter value is equal to the specified limit.
low	Default router preference parameter value is lower than the specified limit.

Command Default

The router preference maximum value is not configured.

Command Modes

RA guard policy configuration (config-ra-guard)

Command History

Release	Modification
12.2(50)SY	This command was introduced.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.
15.0(2)SE	This command was integrated into Cisco IOS Release 15.0(2)SE.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

The **router-preference maximum** command enables verification that the advertised default router preference parameter value is lower than or equal to a specified limit. You can use this command to give a lower priority to default routers advertised on trunk ports, and to give precedence to default routers advertised on access ports.

The **router-preference maximum** command limit are high, medium, or low. If, for example, this value is set to **medium** and the advertised default router preference is set to **high** in the received packet, then the packet is dropped. If the command option is set to **medium** or **low** in the received packet, then the packet is not dropped.

Examples

The following example shows how the command defines a router advertisement (RA) guard policy name as raguard1, places the router in RA guard policy configuration mode, and configures router-preference maximum verification to be high:

```
Router(config)# ipv6 nd raguard policy raguard1
Router(config-ra-guard)# router-preference maximum high
```

Related Commands

Command	Description
ipv6 nd raguard policy	Defines the RA guard policy name and enters RA guard policy configuration mode.