Configuring IPv6 First Hop Security

Prerequisites for IPv6 First Hop Security

You have configured the necessary IPv6 enabled SDM template.

Restrictions for IPv6 First Hop Security

The following restrictions apply when applying FHS policies to EtherChannel interfaces (Port Channels):

- A physical port with an FHS policy attached cannot join an EtherChannel group.
- An FHS policy cannot be attached to an physical port when it is a member of an EtherChannel group.

Information About IPv6 First Hop Security

IPv6 FHS is composed of the following IPv6 security features: IPv6 Snooping, IPv6 Neighbor Discovery Inspection, IPv6 Router Advertisement Guard , IPv6 DHCP Guard, IPv6 Source Guard, IPv6 Prefix Guard, IPv6 Destination Guard.

Each one of these security features addresses a different aspect of first hop security. In order to use a security feature, the corresponding policy must be configured. Policies specify a particular behavior. They must also be attached to a target, which can be a physical interface, an EtherChannel interface, or a VLAN. An IPv6 software policy database service stores and accesses these policies. When a policy is configured or modified, the attributes of the policy are stored or updated in the software policy database, and applied as specified.

In addition to the security features, the IPv6 FHS infrastructure has an IPv6 FHS Binding Table, which is a database table of IPv6 neighbors connected to the device. A binding entry includes information such as the

IP and MAC address of the host, the interface, VLAN, state of the entry etc. This database or binding table is used by other features (such as IPv6 ND Inspection) to validate the link-layer address (LLA), the IPv4 or IPv6 address, and prefix binding of the neighbors, to prevent spoofing and redirect attacks. The binding table is updated through the IPv6 Snooping feature, and through manually added static binding entries.

Note The IPv6 FHS Binding Table is supported through the Switch Integrated Security Feature (SISF) feature. For more information, see the *Configuring Switch Integrated Security Features* chapter in this guide.

IPv6 Snooping

Note The IPv6 Snooping feature is deprecated and the SISF feature replaces it and offers the same capabilities. While the IPv6 Snooping commands are still available on the CLI and the existing configuration continues to be supported, the commands will be removed from the CLI in a later release. For more information about the replacement feature, see the *Configuring Switch Integrated Security Features* chapter in this guide.

IPv6 Snooping acts as a container that enables most of the features available with FHS in IPv6 including following capabilities and functions:

- Neighbor Discovery Snooping: IPv6 Neighbor Discovery Snooping analyzes and verifies IPv6 Neighbor Discovery Protocol (NDP) traffic. During inspection, it gleans address bindings (IP, MAC, port, etc) and stores it in the binding table.
- DHCPv6 Snooping: DHCPv6 Snooping traps DHCPv6 packets between DHCPv6 Client and DHCPv6 Server. From the packets snooped, assigned addresses are learnt and stored in the binding table.
- Device tracking: IPv6 Snooping also tracks the movement of hosts from one port to another, verifies their existence using Duplicate Address Detection (DAD).
- With the IPv6 Snooping feature one can limit the number of addresses any node on the link can claim. This feature can be used to protect the switch binding table against denial of service flooding attacks.

By default, a snooping policy has a security-level of guard. When a snooping policy is configured on an access switch, external IPv6 Router Advertisement (RA) or Dynamic Host Configuration Protocol for IPv6 (DHCPv6) server packets are blocked, even though the uplink port facing the device or DHCP server or relay is configured as a trusted port. To allow IPv6 RA or DHCPv6 server messages, do the following:

- Apply an IPv6 RA-guard policy (for RA) or IPv6 DHCP-guard policy (for DHCP server messages) on the uplink port.
- Configure a snooping policy with a lower security-level, for example glean or inspect. This is a less preferable option, because the benefits of FHS features are not effective.

To use this feature, configure an IPv6 Snooping policy and attach it to a target. See Configuring an IPv6 Snooping Policy, on page 5.

IPv6 Neighbor Discovery Inspection

Note Starting with Cisco IOS XE Amsterdam 17.1.1, the IPv6 Neighbor Discovery Inspection (IPv6 ND Inspection) feature is deprecated and the SISF feature replaces it and offers the same capabilities. While the IPv6 ND Inspection commands are still available on the CLI and the existing configuration continues to be supported, the commands will be removed from the CLI in a later release. For more information about the replacement feature, see the *Configuring Switch Integrated Security Features* chapter in this guide.

The IPv6 ND Inspection feature learns and secures bindings for stateless auto-configuration addresses in Layer 2 neighbor tables. It analyzes neighbor discovery messages in order to build a trusted binding table database. IPv6 neighbor discovery messages that do not conform are dropped. An neighbor discovery message is considered trustworthy if its IPv6-to-MAC mapping is verifiable.

This feature mitigates some of the inherent vulnerabilities of the ND mechanism, such as attacks on DAD, address resolution, router discovery, and the neighbor cache.

To use this feature, configure an IPv6 ND Inspection policy and attach it to a target. See Configuring an IPv6 Neighbor Discovery Inspection Policy, on page 11.

IPv6 Router Advertisement Guard

This feature enables the network administrator to block or reject unwanted or rogue Router Advertisement (RA) guard messages that arrive at the network device platform. RAs are used by devices to announce themselves on the link. The RA Guard feature analyzes the RAs and filters out bogus RAs sent by unauthorized devices. In host mode, all router advertisement and router redirect messages are disallowed on the port. The RA guard feature compares configuration information on the Layer 2 device with the information found in the received RA frame. Once the Layer 2 device has validated the content of the RA frame and router redirect frame against the configuration, it forwards the RA to its unicast or multicast destination. If the RA frame content is not validated, the RA is dropped.

SISF-based device-tracking forwards router solicitation packets only on interfaces that have the RA guard policy configured and are also designated as router-facing interfaces. If no such interface exists, the router solicitation messages are dropped, which might delay the router discovery for onboarding hosts as they will be unable to discover the router until it sends a periodic unsolicited router advertisement.

To use this feature, configure an IPv6 RA Guard policy and attach it to a target. See Configuring an IPv6 Router Advertisement Guard Policy, on page 16.

IPv6 DHCP Guard

The IPv6 DHCP Guard feature blocks reply and advertisement messages that come from unauthorized DHCPv6 servers and relay agents. IPv6 DHCP guard can prevent forged messages from being entered in the binding table and block DHCPv6 server messages when they are received on ports that are not explicitly configured as facing a DHCPv6 server or DHCP relay.

To use this feature, configure an IPv6 DHCP Guard policy and attach it to a target. See Configuring an IPv6 DHCP Guard Policy , on page 20.

To debug DHCP guard packets, use the **debug ipv6 snooping dhcp-guard** privileged EXEC command.

IPv6 Source Guard

The IPv6 Source Guard feature validates the source of IPv6 traffic to prevent source address spoofing. It deals exclusively with data packet traffic. You can use this feature to deny traffic from unknown sources, traffic from sources not assigned by a DHCP server, etc.

It involves a hardware-programmed (TCAM table) filter which allows or denies traffic based on its source address. For the filter to work this way, an entry (of the source address) in the binding table is required. If the source address is in the binding table, the filter allows the packet into the network; if the address is not in the binding table, entry is denied and the packet is dropped. When an entry is removed from the binding table, the filter is also removed, and subsequent packets with that source address are dropped.

When configuring this feature, consider the following:

- The IPv6 Source Guard and Prefix Guard features are supported only in the ingress direction and not supported in the egress direction.
- You cannot use IPv6 Source Guard and Prefix Guard together. When you attach the policy to an interface, it should be "validate address" or "validate prefix" but not both.
- PVLAN and Source or Prefix Guard cannot be applied together.
- IPv6 Source Guard and Prefix Guard is supported on EtherChannels
- An IPv6 source guard policy cannot be attached to a VLAN. It is supported only at the interface level.
- When you configure IPv4 and IPv6 source guard together on an interface, it is recommended to use **ip verify source mac-check** command instead of **ip verify source tracking mac-check** command. IPv4 connectivity on a given port might break due to two different filtering rules set: one for IPv4 (IP-filter) and the other for IPv6 (IP-MAC filter).
- When IPv6 source guard is enabled on a switch port, NDP or DHCP snooping must be enabled on the interface to which the switch port belongs. Otherwise, all data traffic from this port will be blocked.
- Binding information is normally gleaned from IPv6 NDP traffic and DHCP packets. If you rely only on a DHCP server for source addresses of hosts, ensure that you also configure a data-glean recovery function to counteract a situation where entries are prematurely removed from the binding table (for various reasons) before the DHCP lease timer expires. This way, the recovery function *restores* binding entries of valid hosts and you can be sure that that the IPv6 Source Guard feature allows only packets with a DHCP server-assigned source address. See Example: Using the Data-Glean Recovery Function, on page 32.

To use this feature, you must configure an IPv6 Source Guard policy and attach it to a target. See Configuring IPv6 Source Guard, on page 25 .

To debug source-guard packets, use the **debug ipv6 snooping source-guard** privileged EXEC command.

IPv6 Prefix Guard

The IPv6 Prefix Guard feature works within the IPv6 Source Guard feature to enable the device to deny traffic originated from non-topologically correct addresses. IPv6 Prefix Guard is often used when IPv6 prefixes are delegated to devices (for example, home gateways) using DHCP prefix delegation. The feature discovers ranges of addresses assigned to the link and blocks any traffic sourced with an address outside this range.

In order to use this feature, you must configure an IPv6 Prefix Guard policy and attach it to a target. See Configuring IPv6 Prefix Guard, on page 27.

Note Ensure that you have read the configuration considerations listed in the **IPv6 Source Guard** section above - some of them apply to the IPv6 Prefix Guard feature as well.

IPv6 Destination Guard

The IPv6 Destination Guard feature works with IPv6 neighbor discovery to ensure that the device performs address resolution only for those addresses that are known to be active on the link. It relies on the address glean functionality to populate all destinations active on the link into the binding table and then blocks resolutions before they happen when the destination is not found in the binding table.

Note We recommend that you apply an IPv6 Destination Guard policy on all Layer 2 VLANs with an SVI configured.

In order to use this feature, you must configure an IPv6 Destination Guard policy and attach it to a target. See Configuring an IPv6 Destination Guard Policy, on page 30.

How to Configure IPv6 First Hop Security

Configuring an IPv6 Snooping Policy

Note The IPv6 Snooping Policy feature has been deprecated. Although the commands are visible on the CLI and you can configure them, we recommend that you use the Switch Integrated Security Feature (SISF)-based Device Tracking feature instead.

Beginning in privileged EXEC mode, follow these steps to configure IPv6 Snooping Policy :

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 snooping policy** *policy-name* **Example:** `Device(config)# ipv6 snooping policy example_policy`	Creates a snooping policy and enters IPv6 snooping policy configuration mode.

	Command or Action	Purpose
Step 4	{[**default**] | [**device-role** {**node** | **switch**}] | [**limit address-count** *value*] | [**no**] | [**protocol** {**dhcp** | **ndp**}] | [**security-level** {**glean** | **guard** | **inspect**}] | [**tracking** {**disable** [**stale-lifetime** [*seconds* | **infinite**] | **enable** [**reachable-lifetime** [*seconds* | **infinite**] }] | [**trusted-port**] } **Example:** `Device(config-ipv6-snooping)# security-level inspect` **Example:** `Device(config-ipv6-snooping)# trusted-port`	Enables data address gleaning, validates messages against various criteria, specifies the security level for messages. • (Optional) **default**: Sets all to default options. • (Optional) **device-role**{**node**] | **switch**}: Specifies the role of the device attached to the port. Default is **node**. • (Optional) **limit address-count** *value*: Limits the number of addresses allowed per target. • (Optional) **no**: Negates a command or sets it to defaults. • (Optional) **protocol**{**dhcp** | **ndp**}: Specifies which protocol should be redirected to the snooping feature for analysis. The default, is **dhcp** and **ndp**. To change the default, use the **no protocol** command. • (Optional) **security-level**{**glean**|**guard**|**inspect**}: Specifies the level of security enforced by the feature. Default is **guard.** **glean**: Gleans addresses from messages and populates the binding table without any verification. **guard**: Gleans addresses and inspects messages. In addition, it rejects RA and DHCP server messages. This is the default option. **inspect**: Gleans addresses, validates messages for consistency and conformance, and enforces address ownership. • (Optional) **tracking** {**disable** | **enable**}: Overrides the default tracking behavior and specifies a tracking option. • (Optional) **trusted-port**: Sets up a trusted port. It disables the guard on applicable targets. Bindings learned through a trusted port have preference over bindings learned through any other port. A trusted port is given preference in case of a collision while making an entry in the table.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-ipv6-snooping)# end`	Exits IPv6 snooping policy configuration mode and returns to privileged EXEC mode.
Step 6	**show ipv6 snooping policy** *policy-name* **Example:** `Device#show ipv6 snooping policy example_policy`	Displays the snooping policy configuration.

What to do next

Attach an IPv6 Snooping policy to interfaces or VLANs.

Attaching an IPv6 Snooping Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Snooping policy on an interface or VLAN:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface_type stack/module/port* **Example:** `Device(config)# interface gigabitethernet 1/1/4`	Specifies an interface type and identifier and enters the interface configuration mode.
Step 4	**switchport** **Example:**	Enters the Switchport mode.

	Command or Action	Purpose
	`Device(config-if)# switchport`	**Note** To configure Layer 2 parameters, if the interface is in Layer 3 mode, you must enter the **switchport** interface configuration command without any parameters to change the interface into Layer 2 mode. This shuts down the interface and then re-enables it, which might generate messages on the device to which the interface is connected. When change the interface mode from Layer 3 to Layer 2 mode, the previous configuration information related to the affected interface might be lost, and the interface is returned to its default configuration. The command prompt displays as (config-if)# in Switchport configuration mode.

Step 5	**ipv6 snooping** [**attach-policy** *policy_name* [**vlan** {*vlan_id* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids*}] | **vlan** {*vlan_id* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`Device(config-if)# ipv6 snooping`
	`Device(config-if)# ipv6 snooping attach-policy example_policy`
	`Device(config-if)# ipv6 snooping vlan 111,112`
	`Device(config-if)# ipv6 snooping attach-policy example_policy vlan 111,112`
		Attaches a custom IPv6 snooping policy to the interface or the specified VLANs on the interface. To attach the default policy to the interface, use the **ipv6 snooping** command without the **attach-policy** keyword. To attach the default policy to VLANs on the interface, use the **ipv6 snooping vlan** command. The default policy is, security-level **guard**, device-role **node**, protocol **ndp** and **dhcp.**

Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Device# show running-config`	Verifies that the policy is attached to the specified interface without exiting the interface configuration mode.

Attaching an IPv6 Snooping Policy to a Layer 2 EtherChannel Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Snooping policy on an EtherChannel interface or VLAN:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface range** *interface_name* **Example:** `Device(config)# interface range Port-channel 11`	Specifies the port-channel interface name assigned when the EtherChannel was created. Enters the interface range configuration mode. **Tip** Enter the **show interfaces summary** command for quick reference to interface names and types.

Step 4	**ipv6 snooping** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`Device(config-if-range)# ipv6 snooping attach-policy example_policy`
	`Device(config-if-range)# ipv6 snooping attach-policy example_policy vlan 222,223,224`
	`Device(config-if-range)# ipv6 snooping vlan 222, 223,224`
		Attaches the IPv6 Snooping policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

Step 5	**end** **Example:** `Device(config-if-range)# end`	Exits interface range configuration mode and returns to privileged EXEC mode.
Step 6	**show running-config interface***portchannel_interface_name* **Example:** `Device# show running-config interface portchannel 11`	Confirms that the policy is attached to the specified interface.

Attaching an IPv6 Snooping Policy to VLANs Globally

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Snooping Policy to VLANs across multiple interfaces:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vlan configuration** *vlan_list* **Example:** `Device(config)# vlan configuration 333`	Specifies the VLANs to which the IPv6 Snooping policy will be attached, and enters the VLAN interface configuration mode.
Step 4	**ipv6 snooping** [**attach-policy** *policy_name*] **Example:** `Device(config-vlan-config)#ipv6 snooping attach-policy example_policy`	Attaches the IPv6 Snooping policy to the specified VLANs across all device interfaces. The default policy is attached if the **attach-policy** option is not used. The default policy is, security-level **guard**, device-role **node**, protocol **ndp** and **dhcp.**
Step 5	**end** **Example:** `Device(config-vlan-config)# end`	Exits VLAN interface configuration mode and returns to privileged EXEC mode.

Configuring the IPv6 Binding Table Content

Beginning in privileged EXEC mode, follow these steps to configure IPv6 Binding Table Content :

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **ipv6 neighbor binding** [**vlan** *vlan-id* {*ipv6-address* **interface** interface_type *stack/module/port hw_address* [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] | [**tracking**{ [default | disable] [	Adds a static entry to the binding table database.

	Command or Action	Purpose
	reachable-lifetimevalue [*seconds* | **default** | **infinite**] | [**enable** [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] | [**retry-interval** {*seconds*| **default** [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] }] **Example:** `Device(config)# ipv6 neighbor binding`	
Step 4	[**no**] **ipv6 neighbor binding max-entries** *number* [**mac-limit** *number* | **port-limit** *number* [**mac-limit** *number*] | **vlan-limit** *number* [[**mac-limit** *number*] | [**port-limit** *number* [**mac-limit***number*]]]] **Example:** `Device(config)# ipv6 neighbor binding max-entries 30000`	Specifies the maximum number of entries that are allowed to be inserted in the binding table cache.
Step 5	**ipv6 neighbor binding logging** **Example:** `Device(config)# ipv6 neighbor binding logging`	Enables the logging of binding table main events.
Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 7	**show ipv6 neighbor binding** **Example:** `Device# show ipv6 neighbor binding`	Displays contents of a binding table.

Configuring an IPv6 Neighbor Discovery Inspection Policy

Starting with Cisco IOS XE Amsterdam 17.1.1 the IPv6 ND Inspection feature is deprecated and the SISF-based device tracking feature replaces it and offers the same capabilities. For the corresponding replacement task, see *Creating a Custom Device Tracking Policy with Custom Settings* under the *Configuring SISF-Based Device Tracking* chapter in this document.

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 ND Inspection Policy:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 nd inspection policy** *policy-name* **Example:** `Device(config)# ipv6 nd inspection policy example_policy`	Specifies the ND inspection policy name and enters ND Inspection Policy configuration mode.
Step 4	**device-role** {**host** | **switch**} **Example:** `Device(config-nd-inspection)# device-role switch`	Specifies the role of the device attached to the port. The default is **host**.
Step 5	**limit address-count** *value* **Example:** `Device(config-nd-inspection)# limit address-count 1000`	Limits the number of IPv6 addresses allowed to be used on the port.
Step 6	**tracking** {**enable** [**reachable-lifetime** {*value* | **infinite**}] | **disable** [**stale-lifetime** {*value* | **infinite**}]} **Example:** `Device(config-nd-inspection)# tracking disable stale-lifetime infinite`	Overrides the default tracking policy on a port.
Step 7	**trusted-port** **Example:** `Device(config-nd-inspection)# trusted-port`	Configures a port to become a trusted port.
Step 8	**validate source-mac** **Example:** `Device(config-nd-inspection)# validate source-mac`	Checks the source media access control (MAC) address against the link-layer address.
Step 9	**no** {**device-role** | **limit address-count** | **tracking** | **trusted-port** | **validate source-mac**} **Example:** `Device(config-nd-inspection)# no validate source-mac`	Removes the current configuration of a parameter with the **no** form of the command.
Step 10	**default** {**device-role** | **limit address-count** | **tracking** | **trusted-port** | **validate source-mac**} **Example:**	Restores configuration to the default values.

	Command or Action	Purpose
	`Device(config-nd-inspection)# default limit address-count`	
Step 11	**end** **Example:** `Device(config-nd-inspection)# end`	Exits ND Inspection Policy configuration mode and returns to privileged EXEC mode.
Step 12	**show ipv6 nd inspection policy** *policy_name* **Example:** `Device# show ipv6 nd inspection policy example_policy`	Verifies the ND inspection configuration.

Attaching an IPv6 Neighbor Discovery Inspection Policy to an Interface

Starting with Cisco IOS XE Amsterdam 17.1.1 the IPv6 ND Inspection feature is deprecated and the SISF-based device tracking feature replaces it and offers the same capabilities. For the corresponding replacement task, see *Attaching a Device Tracking Policy to an Interface* under the *Configuring SISF-Based Device Tracking* chapter in this document.

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 ND Inspection policy to an interface or VLANs on an interface :

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Device(config)# interface gigabitethernet 1/1/4`	Specifies an interface type and identifier; enters the interface configuration mode.
Step 4	**ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] **Example:** `Device(config-if)# ipv6 nd inspection attach-policy example_policy`	Attaches the Neighbor Discovery Inspection policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

	Command or Action	Purpose
	`Device(config-if)# ipv6 nd inspection attach-policy example_policy vlan 222,223,2` `Device(config-if)# ipv6 nd inspection vlan 222, 223,224`	
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Attaching an IPv6 Neighbor Discovery Inspection Policy to a Layer 2 EtherChannel Interface

Starting with Cisco IOS XE Amsterdam 17.1.1 the IPv6 ND Inspection feature is deprecated and the SISF-based device tracking feature replaces it and offers the same capabilities. For the corresponding replacement task, see *Attaching a Device Tracking Policy to an Interface* under the *Configuring SISF-Based Device Tracking* chapter in this document.

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Neighbor Discovery Inspection policy on an EtherChannel interface or VLAN:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface range** *interface_name* **Example:** `Device(config)# interface range Port-channel 11`	Specifies the port-channel interface name assigned when the EtherChannel was created. Enters interface range configuration mode. **Tip** Enter the **show interfaces summary** command for quick reference to interface names and types.
Step 4	**ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] **Example:** `Device(config-if-range)# ipv6 nd inspection attach-policy example_policy`	Attaches the ND Inspection policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

	Command or Action	Purpose
	`Device(config-if-range)# ipv6 nd inspection vlan 222, 223,224` `Device(config-if-range)# ipv6 nd inspection attach-policy example_policy vlan 222,223,224`	
Step 5	**end** **Example:** `Device(config-if-range)# end`	Exits interface range configuration mode and returns to privileged EXEC mode.

Attaching an IPv6 Neighbor Discovery Inspection Policy to VLANs Globally

Starting with Cisco IOS XE Amsterdam 17.1.1 the IPv6 ND Inspection feature is deprecated and the SISF-based device tracking feature replaces it and offers the same capabilities. For the corresponding replacement task, see *Attaching a Device Tracking Policy to a VLAN* under the *Configuring SISF-Based Device Tracking* chapter in this document.

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 ND Inspection policy to VLANs across multiple interfaces:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vlan configuration** *vlan_list* **Example:** `Device(config)# vlan configuration 334`	Specifies the VLANs to which the IPv6 Snooping policy will be attached, and enters VLAN interface configuration mode.
Step 4	**ipv6 nd inspection** [**attach-policy** *policy_name*] **Example:** `Device(config-vlan-config)#ipv6 nd inspection attach-policy example_policy`	Attaches the IPv6 Neighbor Discovery policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the **attach-policy** option is not used. The default policy is, device-role **host**, no drop-unsecure, limit address-count disabled, sec-level minimum is disabled, tracking is disabled, no trusted-port, no validate source-mac.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-vlan-config)# end`	Exits VLAN interface configuration mode and returns to privileged EXEC mode.

Configuring an IPv6 Router Advertisement Guard Policy

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 Router Advertisement policy :

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 nd raguard policy** *policy-name* **Example:** `Device(config)# ipv6 nd raguard policy example_policy`	Specifies the RA guard policy name and enters RA guard policy configuration mode.
Step 4	[**no**]**device-role** {**host** | **monitor** | **router** | **switch**} **Example:** `Device(config-nd-raguard)# device-role switch`	Specifies the role of the device attached to the port. The default is **host**. **Note** For a network with both host-facing ports and router-facing ports, along with a RA guard policy configured with **device-role host** on host-facing ports or vlan, it is mandatory to configure a RA guard policy with **device-role router** on router-facing ports to allow the RA Guard feature to work properly.
Step 5	**hop-limit** {**maximum** | **minimum**} *value* **Example:** `Device(config-nd-raguard)# hop-limit maximum 33`	Enables filtering of Router Advertisement messages by the Hop Limit value. A rogue RA message may have a low Hop Limit value (equivalent to the IPv4 Time to Live) that when accepted by the host, prevents the host from generating traffic to destinations beyond the rogue RA message generator. An RA message with an unspecified Hop Limit value is blocked.

	Command or Action	Purpose
		(1–255) Range for Maximum and Minimum Hop Limit values. If not configured, this filter is disabled. Configure **minimum** to block RA messages with Hop Limit values lower than the value you specify. Configure **maximum**to block RA messages with Hop Limit values greater than the value you specify.
Step 6	**managed-config-flag** {**off** | **on**} **Example:** `Device(config-nd-raguard)# managed-config-flag on`	Enables filtering of Router Advertisement messages by the managed address configuration, or "M" flag field. A rouge RA message with an M field of 1 can cause a host to use a rogue DHCPv6 server. If not configured, this filter is disabled. **On**: Accepts and forwards RA messages with an M value of 1, blocks those with 0. **Off**: Accepts and forwards RA messages with an M value of 0, blocks those with 1.
Step 7	**match** {**ipv6 access-list** *list* | **ra prefix-list** *list*} **Example:** `Device(config-nd-raguard)# match ipv6 access-list example_list`	Matches a specified prefix list or access list.
Step 8	**other-config-flag** {**on** | **off**} **Example:** `Device(config-nd-raguard)# other-config-flag on`	Enables filtering of Router Advertisement messages by the Other Configuration, or "O" flag field. A rouge RA message with an O field of 1 can cause a host to use a rogue DHCPv6 server. If not configured, this filter is disabled. **On**: Accepts and forwards RA messages with an O value of 1, blocks those with 0. **Off**: Accepts and forwards RA messages with an O value of 0, blocks those with 1.
Step 9	[**no**]**router-preference maximum** {**high** | **medium** | **low**} **Example:** `Device(config-nd-raguard)# router-preference maximum high`	Enables filtering of Router Advertisement messages by the router preference flag. If not configured, this filter is disabled. • **high**: Accepts RA messages with the router preference set to high, medium, or low. • **medium**: Blocks RA messages with the router preference set to high. • **low**: Blocks RA messages with the router preference set to medium and high.

	Command or Action	Purpose
Step 10	**trusted-port** **Example:** `Device(config-nd-raguard)# trusted-port`	When configured as a trusted port, all attached devices are trusted, and no further message verification is performed.
Step 11	**default** {**device-role** | **hop-limit** {**maximum** | **minimum**} | **managed-config-flag** | **match** {**ipv6 access-list** | **ra prefix-list** } | **other-config-flag** | **router-preference maximum**| **trusted-port**} **Example:** `Device(config-nd-raguard)# default hop-limit`	Restores a command to its default value.
Step 12	**end** **Example:** `Device(config-nd-raguard)# end`	Exits RA Guard policy configuration mode and returns to privileged EXEC mode.
Step 13	**show ipv6 nd raguard policy** *policy_name* **Example:** `Device# show ipv6 nd raguard policy example_policy`	(Optional) Displays the ND guard policy configuration.

Attaching an IPv6 Router Advertisement Guard Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Router Advertisement policy to an interface or to VLANs on the interface :

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/1/4`	Specifies an interface type and identifier; enters the interface configuration mode.
Step 4	**ipv6 nd raguard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except**	Attaches the Neighbor Discovery Inspection policy to the interface or the specified VLANs

	Command or Action	Purpose

	vlan_ids | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`Device(config-if)# ipv6 nd raguard attach-policy example_policy`
	`Device(config-if)# ipv6 nd raguard attach-policy example_policy vlan 222,223,224`
	`Device(config-if)# ipv6 nd raguard vlan 222, 223,224`
		on that interface. The default policy is attached if the **attach-policy** option is not used.

Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Attaching an IPv6 Router Advertisement Guard Policy to a Layer 2 EtherChannel Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Router Advertisement Guard Policy on an EtherChannel interface or VLAN:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface range** *type number* **Example:** `Device(config)# interface Port-channel 11`	Specifies the port-channel interface name assigned when the EtherChannel was created. Enters interface range configuration mode. **Tip** Enter the **show interfaces summary** command in privileged EXEC mode for quick reference to interface names and types.
Step 4	**ipv6 nd raguard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]	Attaches the RA Guard policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

	Command or Action	Purpose

	Example:
	`Device(config-if-range)# ipv6 nd raguard attach-policy example_policy`
	`Device(config-if-range)# ipv6 nd raguard attach-policy example_policy vlan 222,223,224`
	`Device(config-if-range)# ipv6 nd raguard vlan 222, 223,224`	

Step 5	**end** **Example:** `Device(config-if-range)# end`	Exits interface range configuration mode and returns to privileged EXEC mode.

Attaching an IPv6 Router Advertisement Guard Policy to VLANs Globally

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Router Advertisement policy to VLANs regardless of interface:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vlan configuration** *vlan_list* **Example:** `Device(config)# vlan configuration 335`	Specifies the VLANs to which the IPv6 RA Guard policy will be attached, and enters VLAN interface configuration mode.
Step 4	**ipv6 dhcp guard** [**attach-policy** *policy_name*] **Example:** `Device(config-vlan-config)# ipv6 nd raguard attach-policy example_policy`	Attaches the IPv6 RA Guard policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the **attach-policy** option is not used.
Step 5	**end** **Example:** `Device(config-vlan-config)# end`	Exits VLAN interface configuration mode and returns to privileged EXEC mode.

Configuring an IPv6 DHCP Guard Policy

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 DHCP (DHCPv6) Guard policy:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp guard policy** *policy-name* **Example:** `Device(config)# ipv6 dhcp guard policy example_policy`	Specifies the DHCPv6 Guard policy name and enters DHCPv6 Guard Policy configuration mode.
Step 4	**device-role** {**client** | **server**} **Example:** `Device(config-dhcp-guard)# device-role server`	(Optional) Filters out DHCPv6 replies and DHCPv6 advertisements on the port that are not from a device of the specified role. Default is **client**. • **client**: Default value, specifies that the attached device is a client. Server messages are dropped on this port. • **server**: Specifies that the attached device is a DHCPv6 server. Server messages are allowed on this port.

Step 5	**match server access-list** *ipv6-access-list-name*
	Example:
	`;;Assume a preconfigured IPv6 Access List as follows:`
	`Device(config)# ipv6 access-list my_acls`
	`Device(config-ipv6-acl)# permit host 2001:BD8:::1 any`
	
	`;;configure DCHPv6 Guard to match approved access list.`
	`Device(config-dhcp-guard)# match server access-list my_acls`
		(Optional). Enables verification that the advertised DHCPv6 server or relay address is from an authorized server access list (The destination address in the access list is 'any'). If not configured, this check will be bypassed. An empty access list is treated as a permit all.

Step 6	**match reply prefix-list** *ipv6-prefix-list-name*
	Example:
	
	`;;Assume a preconfigured IPv6 prefix list as follows:`
	`Device(config)# ipv6 prefix-list my_prefix permit 2001:DB8::/64 le 128`
	
	`;; Configure DCHPv6 Guard to match prefix`
		(Optional) Enables verification of the advertised prefixes in DHCPv6 reply messages from the configured authorized prefix list. If not configured, this check will be bypassed. An empty prefix list is treated as a permit.

	Command or Action	Purpose
	`Device(config-dhcp-guard)# match reply prefix-list my_prefix`	

Step 7	**preference** { **max** *limit* | **min** *limit* }
	Example:
	`Device(config-dhcp-guard)# preference max 250`
	`Device(config-dhcp-guard)#preference min 150`
		Configure **max** and **min** when **device-role** is **server**to filter DCHPv6 server advertisements by the server preference value. The defaults permit all advertisements.
		max *limit*—(0 to 255) (Optional) Enables verification that the advertised preference (in preference option) is less than the specified limit. Default is 255. If not specified, this check will be bypassed.
		min *limit*—(0 to 255) (Optional) Enables verification that the advertised preference (in preference option) is greater than the specified limit. Default is 0. If not specified, this check will be bypassed.

Step 8	**trusted-port** **Example:** `Device(config-dhcp-guard)# trusted-port`	(Optional) **trusted-port**—Sets the port to a trusted mode. No further policing takes place on the port. **Note** If you configure a trusted port then the device-role option is not available.
Step 9	**default** {**device-role** | **trusted-port**} **Example:** `Device(config-dhcp-guard)# default device-role`	(Optional) **default**—Sets a command to its defaults.
Step 10	**end** **Example:** `Device(config-dhcp-guard)# end`	Exits DHCPv6 Guard Policy configuration mode and returns to privileged EXEC mode.
Step 11	**show ipv6 dhcp guard policy** *policy_name* **Example:** `Device# show ipv6 dhcp guard policy example_policy`	(Optional) Displays the configuration of the IPv6 DHCP guard policy. Omitting the *policy_name* variable displays all DHCPv6 policies.

Attaching an IPv6 DHCP Guard Policy to an Interface or a VLAN on an Interface

Beginning in privileged EXEC mode, follow these steps to configure IPv6 Binding Table Content :

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/1/4`	Specifies an interface type and identifier, and enters interface configuration mode.

Step 4	**ipv6 dhcp guard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`Device(config-if)# ipv6 dhcp guard attach-policy example_policy`
	`Device(config-if)# ipv6 dhcp guard attach-policy example_policy vlan 222,223,224`
	`Device(config-if)# ipv6 dhcp guard vlan 222, 223,224`
		Attaches the DHCP Guard policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Attaching an IPv6 DHCP Guard Policy to a Layer 2 EtherChannel Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 DHCP Guard policy on an EtherChannel interface or VLAN:

Procedure

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

	Command or Action	Purpose
Step 3	**interface range** *Interface_name* **Example:** `Device(config)# interface Port-channel 11`	Specify the port-channel interface name assigned when the EtherChannel was created. Enters interface range configuration mode. **Tip** Enter the **show interfaces summary** command in privileged EXEC mode for quick reference to interface names and types.

Step 4	**ipv6 dhcp guard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`Device(config-if-range)# ipv6 dhcp guard attach-policy example_policy`
	`Device(config-if-range)# ipv6 dhcp guard attach-policy example_policy vlan 222,223,224`
	`Device(config-if-range)# ipv6 dhcp guard vlan 222, 223,224`
		Attaches the DHCP Guard policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

Step 5	**end** **Example:** `Device(config-if-range)# end`	Exits interface range configuration mode and returns to privileged EXEC mode.

Attaching an IPv6 DHCP Guard Policy to VLANs Globally

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 DHCP Guard policy to VLANs across multiple interfaces:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vlan configuration** *vlan_list* **Example:** `Device(config)# vlan configuration 334`	Specifies the VLANs to which the IPv6 Snooping policy will be attached, and enters VLAN interface configuration mode.

	Command or Action	Purpose
Step 4	**ipv6 dhcp guard** [**attach-policy** *policy_name*] **Example:** `Device(config-vlan-config)# ipv6 dhcp guard attach-policy example_policy`	Attaches the IPv6 Neighbor Discovery policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the **attach-policy** option is not used. The default policy is, device-role **client**, **no** trusted-port.
Step 5	**end** **Example:** `Device(config-vlan-config)# end`	Exits VLAN interface configuration mode and returns to privileged EXEC mode.

Configuring IPv6 Source Guard

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 source-guard policy** *policy_name* **Example:** `Device(config)# ipv6 source-guard policy example_policy`	Specifies the IPv6 Source Guard policy name and enters IPv6 Source Guard policy configuration mode.
Step 4	[**deny global-autoconf**] [**permit link-local**] [**default**{. . . }] [**exit**] [**no**{. . . }] **Example:** `Device(config-sisf-sourceguard)# deny global-autoconf`	(Optional) Defines the IPv6 Source Guard policy. • **deny global-autoconf**: Denies data traffic from auto-configured global addresses. This is useful when all global addresses on a link are DHCP-assigned and the administrator wants to block hosts with self-configured addresses to send traffic. • **permit link-local**: Allows all data traffic that is sourced by a link-local address. **Note** Trusted option under source guard policy is not supported.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-sisf-sourceguard)# end`	Exits of IPv6 Source Guard policy configuration mode and returns to privileged EXEC mode.
Step 6	**show ipv6 source-guard policy** *policy_name* **Example:** `Device# show ipv6 source-guard policy example_policy`	Shows the policy configuration and all the interfaces where the policy is applied.

What to do next

Apply the IPv6 Source Guard policy to an interface.

Attaching an IPv6 Source Guard Policy to an Interface

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/1/4`	Specifies an interface type and identifier; enters interface configuration mode.
Step 4	**ipv6 source-guard** [**attach-policy** <*policy_name*>] **Example:** `Device(config-if)# ipv6 source-guard attach-policy example_policy`	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the **attach-policy** option is not used.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show ipv6 source-guard policy** *policy_name* **Example:** `Device#(config)# show ipv6 source-guard policy example_policy`	Shows the policy configuration and all the interfaces where the policy is applied.

Attaching an IPv6 Source Guard Policy to a Layer 2 EtherChannel Interface

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface port-channel** *port-channel-number* **Example:** `Device(config)# interface Port-channel 4`	Specifies an interface type and port number and places the switch in the port channel configuration mode.
Step 4	**ipv6 source-guard** [**attach-policy** <*policy_name*>] **Example:** `Device(config-if)# ipv6 source-guard attach-policy example_policy`	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the **attach-policy** option is not used.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show ipv6 source-guard policy** *policy_name* **Example:** `Device# show ipv6 source-guard policy example_policy`	Shows the policy configuration and all the interfaces where the policy is applied.

Configuring IPv6 Prefix Guard

Note To allow routing protocol control packets sourced by a link-local address when prefix guard is applied, enable the **permit link-local** command in the source-guard policy configuration mode.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 source-guard policy** *source-guard-policy* **Example:** `Device(config)# ipv6 source-guard policy my_snooping_policy`	Defines an IPv6 source-guard policy name and enters switch integrated security features source-guard policy configuration mode.
Step 4	**validate address** **Example:** `Device(config-sisf-sourceguard)# no validate address`	Disables the validate address feature and enables the IPv6 prefix guard feature to be configured.
Step 5	**validate prefix** **Example:** `Device(config-sisf-sourceguard)# validate prefix`	Enables IPv6 source guard to perform the IPv6 prefix-guard operation.
Step 6	**exit** **Example:** `Device(config-sisf-sourceguard)# exit`	Exits switch integrated security features source-guard policy configuration mode and returns to privileged EXEC mode.
Step 7	**show ipv6 source-guard policy** [*source-guard-policy*] **Example:** `Device# show ipv6 source-guard policy policy1`	Displays the IPv6 source-guard policy configuration.

Attaching an IPv6 Prefix Guard Policy to an Interface

Procedure

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

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/1/4`	Specifies an interface type and identifier, and enters interface configuration mode.
Step 4	**ipv6 source-guard attach-policy** *policy_name* **Example:** `Device(config-if)# ipv6 source-guard attach-policy example_policy`	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the **attach-policy** option is not used.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show ipv6 source-guard policy** *policy_name* **Example:** `Device(config-if)# show ipv6 source-guard policy example_policy`	Shows the policy configuration and all the interfaces where the policy is applied.

Attaching an IPv6 Prefix Guard Policy to a Layer 2 EtherChannel Interface

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface port-channel** *port-channel-number* **Example:** `Device(config)# interface Port-channel 4`	Specifies an interface type and port number and places the switch in the port channel configuration mode.
Step 4	**ipv6 source-guard** [**attach-policy** <*policy_name*>] **Example:** `Device(config-if)# ipv6 source-guard attach-policy example_policy`	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the **attach-policy** option is not used.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show ipv6 source-guard policy** *policy_name* **Example:** `Device(config)# show ipv6 source-guard policy example_policy`	Shows the policy configuration and all the interfaces where the policy is applied.

Configuring an IPv6 Destination Guard Policy

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 destination guard policy:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 destination-guard policy** *policy-name* **Example:** `Device(config)# ipv6 destination-guard policy pol1`	Defines the destination guard policy name and enters destination-guard configuration mode.
Step 4	**enforcement** {**always** | **stressed**} **Example:** `Device(config-destguard)# enforcement always`	Sets the enforcement level for the target address.
Step 5	**exit** **Example:** `Device(config-destguard)# exit`	Exits destination-guard configuration mode and returns to global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/1`	Enters interface configuration mode.

	Command or Action	Purpose
Step 7	**ipv6 destination-guard attach-policy** [*policy-name*] **Example:** `Device(config-if)# ipv6 destination-guard attach-policy pol1`	Attaches a destination guard policy to an interface.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC configuration mode.
Step 9	**show ipv6 destination-guard policy** [*policy-name*] **Example:** `Device# show ipv6 destination-guard policy pol1`	(Optional) Displays the policy configuration and all interfaces where the policy is applied.

Configuration Examples for IPv6 First Hop Security

Example: Configuring an IPv6 DHCP Guard Policy

Example of DHCPv6 Guard Configuration

```
Device> enable
Device# configure terminal
Device(config)# ipv6 access-list acl1
Device(config-ipv6-acl)# permit host 2001:DB8:0000:
0000:0000:0000:0000:0001 any
Device(config-ipv6-acl)# exit
Device(config)# ipv6 prefix-list abc permit 2001:0DB8::/64 le 128
Device(config)# ipv6 dhcp guard policy pol1
Device(config-dhcp-guard)# device-role server
Device(config-dhcp-guard)# match server access-list acl1
Device(config-dhcp-guard)# match reply prefix-list abc
Device(config-dhcp-guard)# preference min 0
Device(config-dhcp-guard)# preference max 255
Device(config-dhcp-guard)# trusted-port
Device(config-dhcp-guard)# exit
Device(config)# interface GigabitEthernet 0/2/0
Device(config-if)# switchport
Device(config-if)# ipv6 dhcp guard attach-policy pol1 vlan add 1
Device(config-if)# exit
Device(config)# vlan 1
Device(config-vlan)# ipv6 dhcp guard attach-policy pol1
Device(config-vlan)# end
```

Examples: Attaching an IPv6 Source Guard Policy to a Layer 2 EtherChannel Interface

The following example shows how to attach an IPv6 Source Guard Policy to a Layer 2 EtherChannel Interface:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 source-guard policy POL
Device(config-sisf-sourceguard) # validate address
Device(config-sisf-sourceguard)# exit
Device(config)# interface Port-Channel 4
Device(config-if)# ipv6 snooping
Device(config-if)# ipv6 source-guard attach-policy POL
Device(config-if)# end
Device#
```

Examples: Attaching an IPv6 Prefix Guard Policy to a Layer 2 EtherChannel Interface

The following example shows how to attach an IPv6 Prefix Guard Policy to a Layer 2 EtherChannel Interface:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 source-guard policy POL
Device (config-sisf-sourceguard)# no validate address
Device((config-sisf-sourceguard)# validate prefix
Device(config-sisf-sourceguard)# exit
Device(config)# interface Po4
Device(config-if)# ipv6 snooping
Device(config-if)# ipv6 source-guard attach-policy POL

Device(config-if)# end
```

Example: Using the Data-Glean Recovery Function

Binding entries can be removed from the binding table for various reasons: the switch may have reset, or you may have used the **clear** commands, and so on. The following example shows how you can use the data-glean recovery function to restore valid binding entries in the binding table.

The scenario used in this example involves interaction between the IPv6 Source Guard, IEEE 802.1x authentication, and SISF-based device-tracking features. Described below is the set-up we are using for this example, along with sample configuration, followed by a description of situations that can cause premature removal of valid entries from the binding table, and finally, the configuration that you must have in-place, for such entries to be restored.

The key aspects of this example set-up are outlined below:

- An IPv6 Source Guard policy is configured and attached to an interface.

  This means that if the source address of an incoming packet is in the binding table, the filter allows the packet into the network. If the address is not in the binding table, entry is denied and the packet entry is dropped. When an entry is removed from the binding table, the filter is also removed, and subsequent packets from that source are dropped.

```
Device# show ipv6 source-guard policy src-guard-policy
Source guard policy src-guard-policy configuration:
  validate address
Policy src-guard-policy is applied on the following targets:
Target               Type  Policy              Feature        Target range
Gi1/0/1              PORT  src-guard-policy    Source guard   vlan all
```

- A custom SISF-based device-tracking policy, which allows gleaning of only DHCP packets and not NDP packets is attached to the same interface as the source guard policy.

  This means that any host in the network can use only a DHCP-assigned IP address to communicate.

```
Device# show device-tracking policy glean_only_DHCP
Device-tracking policy glean_only_DHCP configuration:
  security-level guard
  device-role node
  NOT gleaning from Neighbor Discovery
  gleaning from DHCP6
  NOT gleaning from ARP
  NOT gleaning from DHCP4
  NOT gleaning from protocol unkn
Policy glean_only_DHCP is applied on the following targets:
Target               Type  Policy              Feature        Target range
Gi1/0/1              PORT  glean_only_DHCP     Device-tracking vlan all
```

- IEEE 802.1x authentication is enabled.

  This means only authenticated hosts are allowed to request addresses from the DHCP server and attach themselves to the network.

Note The following 802.1x configuration is for example purposes only.

```
<output truncated>

 interface GigabitEthernet 1/0/1
 description 802.1x+MAB+IPT

 authentication control-direction in
 authentication event server dead action authorize vlan <vlan id>
 authentication event no-response action authorize vlan <vlan id>
 authentication event server alive action reinitialize
 authentication host-mode multi-domain
 authentication port-control auto
 authentication periodic
 authentication timer reauthenticate server
 authentication violation protect
 mab
 trust device cisco-phone
 dot1x pae authenticator
 dot1x timeout quiet-period 30
 dot1x timeout server-timeout 5
 dot1x timeout tx-period 1
 dot1x max-req 1
 dot1x max-reauth-req 1
<output truncated>
```

Events that cause a change in the configuration occur in any typical network. For example, a host may be unplugged from one port and then plugged back into another port, or an interface may flap, or you may have configured the **shutdown**, followed by the **no shutdown** interface configuration commands. For the duration

that the host is not connected, or the interface is down, the host or interface is considered "unauthenticated". Because of this absence of host or interface authentication, the corresponding binding table entry is removed from the binding table.

When such a host connects back to the network or when such an interface is restored, the client does not reinstantiate the DHCP sequence until the DHCP lease time expires. Until the DHCP sequence is reinstantiated, a valid address fails to be stored in the binding table. If the entry is not in the binding table, the IPv6 Source Guard's filter function drops all packets initiated by that host.

In order to prevent such a situation, configure the data-glean recovery function.

To configure data-glean recovery, create a custom SISF-based device-tracking policy, configure the data-glean policy parameter to recover binding information from DHCP Server, and attach it to the necessary targets.

Note When configuring data-glean recovery from DHCP, for binding information retrieval to work as expected, the DHCPv6 Leasequery configuration (as in RFC 5007), is required. Ensure that the leasequery configuration is enabled on the DHCP Server.

The following sample configuration shows how to add the required "data-glean" policy parameter to the existing custom SISF-based device-tracking policy (`glean_only_DHCP`), to recover binding information. It remains attached to the same target as the IPv6 Source Guard policy, that is, Gigabit Ethernet 1/0/1:

```
Device# configure terminal
Device(config)# device-tracking policy glean_only_DHCP
Device(config-device-tracking)#  data-glean recovery dhcp
Device(config-device-tracking)# exit

Device# show device-tracking policy glean_only_DHCP
Device-tracking policy glean_only_DHCP configuration:
  security-level guard
  device-role node
  data-glean recovery dhcp                        <<< Recovery of binding information is
configured.
  NOT gleaning from Neighbor Discovery
  gleaning from DHCP6
  NOT gleaning from ARP
  NOT gleaning from DHCP4
  NOT gleaning from protocol unkn
Policy glean_only_DHCP is applied on the following targets:
Target               Type  Policy               Feature        Target range
Gi1/0/1              PORT  glean_only_DHCP      Device-tracking vlan all

Device# show device-tracking policies interface Gi1/0/1
Target               Type  Policy               Feature        Target range
Gi1/0/1              PORT  glean_only_DHCP      Device-tracking vlan all
Gi1/0/1              PORT  src-guard-policy     Source guard   vlan all
```

With this additional configuration, valid entries are automatically restored in the binding table if they are removed prematurely.

Additional References for IPv6 First Hop Security

Related Documents

Related Topic	Document Title
SISF	Configuring SISF-Based Device Tracking chapter of the *Security Configuration Guide*

Technical Assistance

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

Feature History for IPv6 First Hop Security

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

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

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	IPv6 First Hop Security	First Hop Security in IPv6 is a set of IPv6 security features, the policies of which can be attached to a physical interface, an EtherChannel interface, or a VLAN. An IPv6 software policy database service stores and accesses these policies. When a policy is configured or modified, the attributes of the policy are stored or updated in the software policy database, then applied as was specified. The IPv6 Snooping Policy feature has been deprecated. Although the commands are visible on the CLI and you can configure them, we recommend that you use the Switch Integrated Security Feature (SISF)-based Device Tracking feature instead.

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.1.1	IPv6 ND Inspection	Starting with this release, the IPv6 ND Inspection feature is deprecated and the SISF- based device tracking feature replaces it and offers the same capabilities. While the IPv6 ND Inspection commands are still available on the CLI and the existing configuration continues to be supported, the commands will be removed from the CLI in a later release. For more information about the replacement feature, see the *Configuring SISF-Based Device Tracking* chapter in this guide.

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