CHAPTER 36

Configuring IPv6 Host Functions

This chapter describes how to configure IPv6 host functions on the Catalyst 2960, 2960-S, or 2960-C switch.

For information about configuring IPv6 Multicast Listener Discovery (MLD) snooping, see Chapter 37, "Configuring IPv6 MLD Snooping."

To enable dual stack environments (supporting both IPv4 and IPv6) on a Catalyst 2960 switch, you must configure the switch to use the a dual IPv4 and IPv6 switch database management (SDM) template. See the "Dual IPv4 and IPv6 Protocol Stacks" section on page 36-8. This template is not required on Catalyst 2960-S switches.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS documentation referenced in the procedures.

Understanding IPv6

IPv4 users can move to IPv6 and receive services such as end-to-end security, quality of service (QoS), and globally unique addresses. The IPv6 address space reduces the need for private addresses and Network Address Translation (NAT) processing by border routers at network edges.

For information about how Cisco Systems implements IPv6, go to:
http://www.cisco.com/en/US/products/ps6553/products_ios_technology_home.html

For information about IPv6 and other features in this chapter:

- See the *Cisco IOS IPv6 Configuration Library*:
  http://www.cisco.com/en/US/docs/ios/12_2t/ipv6/ipv6_vgf.html
- Use the Search field to locate the Cisco IOS software documentation. For example, if you want information about static routes, you can enter *Implementing Static Routes for IPv6* in the search field to get this document about static routes:
  http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-stat_routes_ps6441_TSD_Products_Configuration_Guide_Chapter.html

These sections describe IPv6 implementation on the switch.

IPv6 Addresses

The switch supports only IPv6 unicast addresses. It does not support site-local unicast addresses, anycast addresses, or multicast addresses.

The IPv6 128-bit addresses are represented as a series of eight 16-bit hexadecimal fields separated by colons in the format: n:n:n:n:n:n:n:n. This is an example of an IPv6 address:

2031:0000:130F:0000:0000:09C0:080F:130B

For easier implementation, leading zeros in each field are optional. This is the same address without leading zeros:

2031:0:130F:0:0:9C0:80F:130B

You can also use two colons (::) to represent successive hexadecimal fields of zeros, but you can use this short version only once in each address:

2031:0:130F::09C0:080F:130B

For more information about IPv6 address formats, address types, and the IPv6 packet header, see the "Implementing IPv6 Addressing and Basic Connectivity" chapter of *Cisco IOS IPv6 Configuration Library* on Cisco.com.

In the "Implementing Addressing and Basic Connectivity" chapter, these sections apply to the Catalyst 2960, 2960-S, or 2960-C switch:

- IPv6 Address Formats
- IPv6 Address Output Display
- Simplified IPv6 Packet Header

Supported IPv6 Host Features

These sections describe the IPv6 protocol features supported by the switch:

Support on the switch includes expanded address capability, header format simplification, improved support of extensions and options, and hardware parsing of the extension header. The switch supports hop-by-hop extension header packets, which are routed or bridged in software.

128-Bit Wide Unicast Addresses

The switch supports aggregatable global unicast addresses and link-local unicast addresses. It does not support site-local unicast addresses.

- Aggregatable global unicast addresses are IPv6 addresses from the aggregatable global unicast prefix. The address structure enables strict aggregation of routing prefixes and limits the number of routing table entries in the global routing table. These addresses are used on links that are aggregated through organizations and eventually to the Internet service provider.

  These addresses are defined by a global routing prefix, a subnet ID, and an interface ID. Current global unicast address allocation uses the range of addresses that start with binary value 001 (2000::/3). Addresses with a prefix of 2000::/3(001) through E000::/3(111) must have 64-bit interface identifiers in the extended unique identifier (EUI)-64 format.

- Link local unicast addresses can be automatically configured on any interface by using the link-local prefix FE80::/10(1111 1110 10) and the interface identifier in the modified EUI format. Link-local addresses are used in the neighbor discovery protocol (NDP) and the stateless autoconfiguration process. Nodes on a local link use link-local addresses and do not require globally unique addresses to communicate. IPv6 routers do not forward packets with link-local source or destination addresses to other links.

For more information, see the section about IPv6 unicast addresses in the "Implementing IPv6 Addressing and Basic Connectivity" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

DNS for IPv6

IPv6 supports Domain Name System (DNS) record types in the DNS name-to-address and address-to-name lookup processes. The DNS AAAA resource record types support IPv6 addresses and are equivalent to an A address record in IPv4. The switch supports DNS resolution for IPv4 and IPv6.

ICMPv6

The Internet Control Message Protocol (ICMP) in IPv6 generates error messages, such as ICMP destination unreachable messages, to report errors during processing and other diagnostic functions. In IPv6, ICMP packets are also used in the neighbor discovery protocol and path MTU discovery.

Neighbor Discovery

The switch supports NDP for IPv6, a protocol running on top of ICMPv6, and static neighbor entries for IPv6 stations that do not support NDP. The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), to verify the reachability of the neighbor, and to keep track of neighboring routers.

The switch supports ICMPv6 redirect for routes with mask lengths less than 64 bits. ICMP redirect is not supported for host routes or for summarized routes with mask lengths greater than 64 bits.

Neighbor discovery throttling ensures that the switch CPU is not unnecessarily burdened while it is in the process of obtaining the next hop forwarding information to route an IPv6 packet. The switch drops any additional IPv6 packets whose next hop is the same neighbor that the switch is actively trying to resolve. This drop avoids further load on the CPU.

First Hop Security in IPv6

This section provides information about configuring the functions that comprise the first hop security (FHS) feature in IPv6.

The functions available under FHS are also called as IPv6 policies. Policies can be applied at the interface or VLAN level. IPv6 policies provide policy database services to features with regard to storing and accessing those policies. Every time a policy is configured, the attributes of the policy are stored in the software policy database. The policy is then applied to an interface and the software policy database entry is updated to include this interface to which the policy is applied. You can use the following IPv6 policies:

- IPv6 Snooping, page 36-5
- IPv6 First-Hop Security Binding Table, page 36-5
- NDP Address Gleaning, page 36-5
- IPv6 DHCP Address Gleaning, page 36-5
- IPv6 DHCP Address Gleaning, page 36-5
- IPv6 ND Inspection, page 36-6
- IPv6 Device Tracking, page 36-7
- IPv6 Port-Based Access List Support, page 36-7
- IPv6 Router Advertisement Guard, page 36-7
- IPv6 Device Tracking, page 36-7
- IPv6 Source Guard, page 36-7

Note Prerequisites for Implementing First Hop Security in IPv6:

- You have configured the necessary IPv6 enabled SDM template.
- You should be familiar with the IPv6 neighbor discovery feature. For information , see "Implementing IPv6 Addressing and Basic Connectivity" chapter of the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Note Restrictions for Implementing First Hop Security in IPv6:

- This feature is supported only on gigabitEthernet switches.
- The Catalyst 2960-S LAN Lite image supports only IPv6 RA guard. Further, you cannot attach an IPv6 ACL to the RA guard policy as the switch does not support IPv6 ACL.
- First Hop Security is supported only on Catalyst 2960-CG series switches.
- VLAN targets are not supported in a mixed stack scenario.

IPv6 Snooping

IPv6 snooping acts as a container policy that enables most of the features available with FHS in IPv6. For more information, see the "Configuring an IPv6 Snooping Policy" section on page 36-13.

IPv6 First-Hop Security Binding Table

A database table of IPv6 neighbors connected to the switch is created from multiple sources of information, For example, Neighbor Discovery Protocol (NDP) snooping and Dynamic Host Configuration Protocol (DHCP) snooping. This database or binding table is used by various IPv6 guard features, such as, IPv6 Neighbor Discovery (ND) Inspection (to validate the link-layer address (LLA)), per-port address limit (to validate the IPv4 or IPv6 addresses), IPv6 device tracking (to prefix binding of the neighbors to prevent spoofing and redirect attacks).

These categories of traffic carry information that the binding table snoops for:

- ND traffic—For more information, see the "NDP Address Gleaning" section on page 36-5.
- DHCP traffic—For more information, see the "IPv6 DHCP Address Gleaning" section on page 36-5.
- Data traffic—For more informaiton, see the "IPv6 DHCP Address Gleaning" section on page 36-5.

NDP Address Gleaning

The NDP address gleaning feature is enabled by default when you configure the **ipv6 snooping policy** global configuration command. To disable this function, enter the **no protocol ndp** global configuration command and attach the policy to the target port or VLAN.

IPv6 DHCP Address Gleaning

The IPv6 DHCP address gleaning feature provides the ability to extract addresses from DHCP messages and populate the binding table. The switch extracts address binding information from the following types of DHCPv6 exchanges (using User Datagram Protocol (UDP), ports 546 and 547):

- DHCP-REQUEST
- DHCP-CONFIRM
- DHCP- RENEW
- DHCP-REBIND
- DHCP-REPLY
- DHCP-RELEASE
- DHCP-DECLINE

After a switch receives a DHCP-REQUEST message from a client, one of the following can happen:

- The switch receives a DHCP-REPLY message from DHCP server and a binding table entry is created in the REACHABLE state and completed. The reply contains the IP address and the MAC address in the Layer 2 (L2) DMAC field.

  Creating an entry in the binding table allows the switch to learn addresses assigned by DHCP. A binding table can have one of the following states:

  - INCOMPLETE—Address resolution is in progress and the link-layer address is not yet known.
  - REACHABLE—The table is known to be reachable within the last reachable time interval.
  - STALE—The table requires re-resolution.
  - SEARCH—The feature creating the entry does not have the L2 address and requests the binding table to search for the L2 address.
  - VERIFY—The L2 and Layer 3 (L3) addresses are known and a duplicate address detection (DAD) Neighbor solicitation (NS) unicast is sent to the L2 and L3 destinations, to verify the addresses.
  - DOWN—The interface from which the entry was learnt is down, preventing verification.
- The DHCP server sends a DHCP-DECLINE or DHCP release message and the entry is deleted.
- The client sends a DHCP-RENEW message to the server that allocated the address or a DHCP-REBIND message to any server and the lifespan of the entry is extended.
- The server does not reply and the session is timed-out.

To enable this feature, configure a policy using the **ipv6 snooping policy** *policy-name* global configuration command. For more information, see the "Configuring an IPv6 Snooping Policy" section on page 36-13.

You can configure a policy and attach it to a DHCP guard to prevent the binding table from being filled with forged DHCP messages. For more information, see the "IPv6 DHCP Guard" section on page 36-7 and "Configuring IPv6 DHCP Guard" section on page 36-15.

IPv6 Data Address Gleaning

The IPv6 data address gleaning feature provides the ability to extract addresses from redirected data traffic, to discover neighbors and to populate binding tables.

When a port receives a data packet where the binding is unknown, that is, the neighbor is in an INCOMPLETE state and the link-layer addres is not yet known, the switch sends a DAD NS NDP unicast message to the port from which the data packet was received.

After the host replies with a DAD Neighbor Advertisement (NA) NDP message, the binding table is updated and a Private VLAN ACL (PVACL) is installed in the hardware for this binding.

If the host does not reply with a DAD NA, after the binding table timer expires, the hardware is notified and any resources associated with that binding are released.

To enable this feature, configure a policy with **data-glean** and attach the policy to a target port. To debug the policy, use the **debug ipv6 snooping** privileged EXEC command.

IPv6 ND Inspection

IPv6 ND inspection learns and secures bindings for stateless autoconfiguration addresses in L2 neighbor tables. IPv6 ND inspection analyzes neighbor discovery messages in order to build a trusted binding table database and IPv6 neighbor discovery messages that do not conform are dropped. An SA ND message is considered trustworthy if its IPv6-to-Media Access Control (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.

IPv6 Device Tracking

The IPv6 device tracking feature provides IPv6 host liveness tracking so that a neighbor table can be updated when an IPv6 host disappears. The feature tracks the liveness of the neighbors connected through the L2 switch on regular basis in order to revoke network access privileges as they become inactive.

IPv6 Port-Based Access List Support

The IPv6 port-based access lists (PACL) feature provides the ability to provide access control (permit or deny) on L2 switch ports for IPv6 traffic. IPv6 PACLs are similar to IPv4 PACLs, which provide access control on L2 switch ports for IPv4 traffic.

With Catalyst 3750-E, 3750X, 3560E, 3560-X, 3750v2, and 3560 v2 switches, this feature is supported in hardware and only in ingress direction. In a mixed stack scenario where the stack has a switch that does not support IPv6 FHS, the VLAN target is disabled on the whole switch, for security. Port targets are allowed on the IPv6 FHS-capable ports of the switch. If a nonsupporting switch becomes the stack master then the IPv6 FHS functions are still supported on the IPv6 FHS-capable ports of the switch.

Access lists determine which traffic is blocked and which traffic is forwarded at switch interfaces and allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Each access list has an implicit deny statement at the end. To configure an IPv6 PACL you have to create an IPv6 access list and then configure the PACL mode on the specified IPv6 L2 interface.

PACL can filter ingress traffic on L2 interfaces based on L3 and Layer 4 (L4) header information or non-IP L2 information.

IPv6 Router Advertisement Guard

The IPv6 Router Advertisement (RA) guard feature enables the network administrator to block or reject unwanted or rogue RA guard messages that arrive at the network switch platform. RAs are used by routers to announce themselves on the link. The RA Guard feature analyzes the RAs and filters out bogus RAs sent by unauthorized routers. In host mode, all router advertisement and router redirect messages are disallowed on the port. The RA guard feature compares configuration information on the L2 device with the information found in the received RA frame. Once the L2 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.

IPv6 DHCP Guard

You can use the DHCP guard to prevent forged messages from being entered in the binding table. The DHCP guard blocks DHCP server messages when they are received on ports that are not explicitly configured as facing a DHCP server or DHCP relay.

To use this feature, configure a policy and attach it to a DHCP guard. To debug DHCP guard packets, use the **debug ipv6 snooping dhcp-guard** privileged EXEC command.

IPv6 Source Guard

A source guard programs the hardware to allow or deny traffic based on source or destination addresses. It deals exclusively with data packet traffic.

The IPv6 source gaurd feature provides the ability to use the IPv6 binding table to install PACLs to prevent a host from sending packets with an invalid IPv6 source address.

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

Note The IPv6 PACL feature is supported only in the ingress direction; it is not supported in the egress direction.

The following restrictions apply:

- When IPv6 source guard is enabled on a switchport, NDP or DHCP snooping must be enabled on the interface to which the switchport belongs. Otherwise all data traffic from this port will be blocked.
- An IPv6 source guard policy cannot be attached to a VLAN.
- IPv6 source guard is not supported on EtherChannels.

For information about configuring IPv6 access lists, see the "Implementing Traffic Filters and Firewalls for IPv6 Security" chapter of the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

IPv6 Stateless Autoconfiguration and Duplicate Address Detection

The switch uses stateless autoconfiguration to manage link, subnet, and site addressing changes, such as management of host and mobile IP addresses. A host autonomously configures its own link-local address, and booting nodes send router solicitations to request router advertisements for configuring interfaces.

For more information about autoconfiguration and duplicate address detection, see the “Implementing IPv6 Addressing and Basic Connectivity” chapter of *Cisco IOS IPv6 Configuration Library* on Cisco.com.

IPv6 Applications

The switch has IPv6 support for these applications:

- Ping, traceroute, and Telnet
- Secure Shell (SSH) over an IPv6 transport
- HTTP server access over IPv6 transport
- DNS resolver for AAAA over IPv4 transport
- Cisco Discovery Protocol (CDP) support for IPv6 addresses

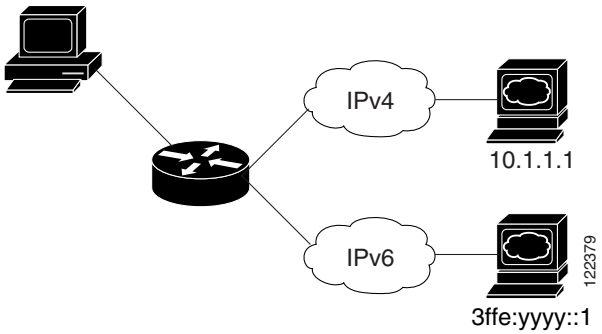
For more information about managing these applications, see the “Managing Cisco IOS Applications over IPv6” chapter and the “Implementing IPv6 Addressing and Basic Connectivity” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Dual IPv4 and IPv6 Protocol Stacks

On a Catalyst 2960 switch, you must use the dual IPv4 and IPv6 template to allocate ternary content addressable memory (TCAM) usage to both IPv4 and IPv6 protocols.

Figure 36-1 shows a router forwarding both IPv4 and IPv6 traffic through the same interface, based on the IP packet and destination addresses.

Figure 36-1 Dual IPv4 and IPv6 Support on an Interface



Use the dual IPv4 and IPv6 switch database management (SDM) template on a Catalyst 2960 switch to enable dual stack environments (supporting both IPv4 and IPv6). For more information about the dual IPv4 and IPv6 SDM template, see Chapter 10, "Configuring SDM Templates."

The dual IPv4 and IPv6 templates on Catalyst 2960 switches allow the switch to be used in dual stack environments.

- If you try to configure IPv6 without first selecting a dual IPv4 and IPv6 template, a warning message appears.
- In IPv4-only environments, the switch applies IPv4 QoS and ACLs in hardware. IPv6 packets are not supported.
- In dual IPv4 and IPv6 environments, the switch applies IPv4 QoS and ACLs in hardware.
- IPv6 QoS and ACLs are not supported.
- If you do not plan to use IPv6, do not use the dual stack template because this template results in less TCAM capacity for each resource.

For more information about IPv4 and IPv6 protocol stacks, see the "Implementing IPv6 Addressing and Basic Connectivity" chapter of *Cisco IOS IPv6 Configuration Library* on Cisco.com.

onfiguring IPSec on OSPFv3SNMP and Syslog Over IPv6

To support both IPv4 and IPv6, IPv6 network management requires both IPv6 and IPv4 transports. Syslog over IPv6 supports address data types for these transports.

SNMP and syslog over IPv6 provide these features:

- Support for both IPv4 and IPv6
- IPv6 transport for SNMP and to modify the SNMP agent to support traps for an IPv6 host
- SNMP- and syslog-related MIBs to support IPv6 addressing
- Configuration of IPv6 hosts as trap receivers

For support over IPv6, SNMP modifies the existing IP transport mapping to simultaneously support IPv4 and IPv6. These SNMP actions support IPv6 transport management:

- Opens User Datagram Protocol (UDP) SNMP socket with default settings
- Provides a new transport mechanism called *SR_IPV6_TRANSPORT*
- Sends SNMP notifications over IPv6 transport
- Supports SNMP-named access lists for IPv6 transport
- Supports SNMP proxy forwarding using IPv6 transport

- Verifies SNMP Manager feature works with IPv6 transport

For information on SNMP over IPv6, including configuration procedures, see the "Managing Cisco IOS Applications over IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

For information about syslog over IPv6, including configuration procedures, see the "Implementing IPv6 Addressing and Basic Connectivity" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

HTTP(S) Over IPv6

The HTTP client sends requests to both IPv4 and IPv6 HTTP servers, which respond to requests from both IPv4 and IPv6 HTTP clients. URLs with literal IPv6 addresses must be specified in hexadecimal using 16-bit values between colons.

The accept socket call chooses an IPv4 or IPv6 address family. The accept socket is either an IPv4 or IPv6 socket. The listening socket continues to listen for both IPv4 and IPv6 signals that indicate a connection. The IPv6 listening socket is bound to an IPv6 wildcard address.

The underlying TCP/IP stack supports a dual-stack environment. HTTP relies on the TCP/IP stack and the sockets for processing network-layer interactions.

Basic network connectivity (**ping**) must exist between the client and the server hosts before HTTP connections can be made.

For more information, see the "Managing Cisco IOS Applications over IPv6" chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

IPv6 and Switch Stacks

The switch supports IPv6 forwarding across the stack and IPv6 host functionality on the stack master. The stack master runs IPv6 host functionality and IPv6 applications.

While the new stack master is being elected and is resetting, the switch stack does not forward IPv6 packets. The stack MAC address changes, which also changes the IPv6 address. When you specify the stack IPv6 address with an extended unique identifier (EUI) by using the **ipv6 address** *ipv6-prefix/prefix length* **eui-64** interface configuration command, the address is based on the interface MAC address. See the "Configuring IPv6 Addressing and Enabling IPv6 Host" section on page 36-11.

If you configure the persistent MAC address feature on the stack and the stack master changes, the stack MAC address does not change for approximately 4 minutes. For more information, see the "Enabling Persistent MAC Address" section on page 9-18 in Chapter 9, "Managing Switch Stacks."

Configuring IPv6

These sections contain this IPv6 forwarding configuration information:

Default IPv6 Configuration

Table 36-1 shows the default IPv6 configuration.

Table 36-1 Default IPv6 Configuration

Feature	Default Setting
SDM template	Default
IPv6 addresses	None configured.

Configuring IPv6 Addressing and Enabling IPv6 Host

This section describes how to assign IPv6 addresses to individual Layer 3 interfaces and to globally forward IPv6 traffic on the switch.

Before configuring IPv6 on the switch, consider these guidelines:

- Be sure to select a dual IPv4 and IPv6 SDM template.
- In the **ipv6 address** interface configuration command, you must enter the *ipv6-address* and *ipv6-prefix* variables with the address specified in hexadecimal using 16-bit values between colons. The *prefix-length* variable (preceded by a slash [/]) is a decimal value that shows how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address).

To forward IPv6 traffic on an interface, you must configure a global IPv6 address on that interface. Configuring an IPv6 address on an interface automatically configures a link-local address and activates IPv6 for the interface. The configured interface automatically joins these required multicast groups for that link:

- solicited-node multicast group FF02:0:0:0:0:1:ff00::/104 for each unicast address assigned to the interface (this address is used in the neighbor discovery process.)
- all-nodes link-local multicast group FF02::1
- all-routers link-local multicast group FF02::2

For more information about configuring IPv6, see the “Implementing Addressing and Basic Connectivity for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Beginning in privileged EXEC mode, follow these steps to assign an IPv6 address to a Layer 3 interface and enable IPv6forwarding:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**sdm prefer dual-ipv4-and-ipv6 default**	Select the SDM template that supports IPv4 and IPv6.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**reload**	Reload the operating system.
Step 5	**configure terminal**	Enter global configuration mode after the switch reloads.
Step 6	**interface** *interface-id*	Enter interface configuration mode, and specify the interface to configure.

	Command	Purpose
Step 7	**ipv6 address** *ipv6-prefix/prefix length* **eui-64** or **ipv6 address** *ipv6-address/prefix length* or **ipv6 address** *ipv6-address* **link-local** or **ipv6 enable**	Specify a global IPv6 address with an extended unique identifier (EUI) in the low-order 64 bits of the IPv6 address. Specify only the network prefix; the last 64 bits are automatically computed from the switch MAC address. This enables IPv6 processing on the interface. Manually configure an IPv6 address on the interface. Specify a link-local address on the interface to be used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface. This command enables IPv6 processing on the interface. Automatically configure an IPv6 link-local address on the interface, and enable the interface for IPv6 processing. The link-local address can only be used to communicate with nodes on the same link.
Step 8	**exit**	Return to global configuration mode.
Step 9	**end**	Return to privileged EXEC mode.
Step 10	**show ipv6 interface** *interface-id*	Verify your entries.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove an IPv6 address from an interface, use the **no ipv6 address** *ipv6-prefix/prefix length* **eui-64** or **no ipv6 address** *ipv6-address* **link-local** interface configuration command. To remove all manually configured IPv6 addresses from an interface, use the **no ipv6 address** interface configuration command without arguments. To disable IPv6 processing on an interface that has not been explicitly configured with an IPv6 address, use the **no ipv6 enable** interface configuration command. To globally disable IPv6 routing, use the **no ipv6 unicast-routing** global configuration command.

This example shows how to enable IPv6 with both a link-local address and a global address based on the IPv6 prefix 2001:0DB8:c18:1::/64. The EUI-64 interface ID is used in the low-order 64 bits of both addresses. Output from the **show ipv6 interface** EXEC command shows how the interface ID (20B:46FF:FE2F:D940) is appended to the link-local prefix FE80::/64 of the interface.

```
Switch(config)# sdm prefer dual-ipv4-and-ipv6 default
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64
Switch(config-if)# end
Switch# show ipv6 interface gigabitethernet1/0/1
GigabitEthernet1/0/1 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::20B:46FF:FE2F:D940
  Global unicast address(es):
  2001:0DB8:c18:1:20B:46FF:FE2F:D940, subnet is 2001:0DB8:c18:1::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2F:D940
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
```

Configuring First Hop Security in IPv6

Configuring an IPv6 Snooping Policy

	Action or Command	Purpose
Step 1	**enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enters the global configuration mode.
Step 3	**ipv6 snooping policy** *policy-name*	Creates a snooping policy in global configuration mode.

	Action or Command	Purpose
Step 4	[**data-glean** | **default** | **device-role** [**node** | **switch**] | **limit** {**address-count** *value* }| **no** | **protocol** [**all** | **dhcp** | **ndp**] | **security-level** [**glean** | **guard** | **inspect**]| **tracking** [**disable** | **enable**]| **trusted-port**}	Enables data address gleaning, validates messages against various criteria, specifies the security level for messages. • (Optional) **data-glean**—Enables data address gleaning. This option is disabled by default. • (Optional) **default**—Sets all default options. • (Optional) **device-role** [**node** | **switch**]—Qualifies the role of the device attached to the port. • (Optional) **limit** {**address-count** *value*}—Limits the number of addresses allowed per target. • (Optional) **no**—Negates a command or set its defaults. • (Optional) **protocol** [**all** | **dhcp** | **ndp**]—Specifies which protocol should be redirected to the snooping feature for analysis. The default, is **all**. To change the default, use the **no protocol** command. • (Optional) **security-level** [**glean** | **guard** | **inspect**]—Specifies the level of security enforced by the feature. – **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 learnt through a trusted port have preference over bindings learnt through any other port. A trusted port is also given preference in case of a collision while making an entry in the table.
Step 5	**exit**	Exits the snooping policyconfiguration mode.
Step 6	**show ipv6 snooping policy** *policy-name*	Displays the snooping policy configuration.

To attach a snooping policy to an interface or VLAN, complete the following steps:

	Action or Command	Purpose
Step 1	**enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enters the global configuration mode.
Step 3	**interface** *type number*	Specifies an interface type and number, and enters the interface configuration mode.

	Action or Command	Purpose
Step 4	**switchport** **ipv6 snooping attach-policy** *policy-name* OR **vlan configuration** *vlan list* **ipv6 snooping attach-policy** *policy-name*	Attaches the snooping policy (where data gleaning is enabled) to an interface. Specifies the port and the policy that is attached to the port. **Note** If you have enabled **data-glean** on a snooping policy, you must attach it to an interface and not a VLAN.
Step 5	**show ipv6 snooping policy** *policy-name*	Displays the snooping policy configuration.
Step 6	**show ipv6 neighbors binding**	Displays the binding table entries populated by the snooping policy.

Configuring IPv6 DHCP Guard

	Action or Command	Purpose
Step 1	**enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enters the global configuration mode.
Step 3	**ipv6 dhcp guard policy** *policy-name*	Creates a policy in global configuration mode and enters the DHCP guard policy global configuration mode.
Step 4	[**default** | **device-role** [**client** | **server**] | **no** | **exit** | **trusted-port**]	Configures the parameters for the DHCP guard policy. • (Optional) **default**—Set a command to its defaults. • (Optional) **device-role** [**client** | **server**]—Qualifies the role of the device attached to the port. – **client**—Specifies that the attached device is a client. This is the default. Any server messages are dropped on this port. – **server**—Specifies that the attached device is a DHCP server. Server messages are allowed on this port. • (Optional) **no**—Removes the configured policy parameters. • (Optional) **exit**—Exits the DHCP guard policy global configuration mode. • (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 5	**exit**	Exits the DHCP guard policy global configuration mode.
Step 6	**interface** *type number*	Specifies an interface type and number and enters the interface configuration mode.

	Action or Command	Purpose
Step 7	**ipv6 dhcp guard attach-policy** *policy-name* Or **vlan configuration** *vlan-id*	Attaches the DHCP guard policy to an interface or VLAN.
Step 8	**show ipv6 dhcp guard policy** *policy-name*	Displays the DHCP guard policy configuration.

Configuring IPv6 Source Guard

	Action or Command	Purpose
Step 1	**enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enters the global configuration mode.
Step 3	**ipv6 source-guard policy** *policy-name*	Specifies the source guard policy name and enters the source guard policy configuration mode.
Step 4	**permit link-local**	Allows all data traffic that is sourced by a link-local address.
Step 5	**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.
Step 6	**ipv6 source-guard** [**attach-policy** *policy-name*]	Specifies the policy name. (Optional) **attach-policy** *policy-name*—Filters based on the policy name
Step 7	**exit**	Exits the source guard policy configuration mode.
Step 8	**show ipv6 source-guard policy** *policy name*	Shows the policy configuration and all the interfaces where the policy is applied.

Configuration Examples for Implementing First Hop Security in IPv6

This example shows you how to attach a snooping policy to a VLAN and to configure an RA trusted router port and DHCP trusted server port:

```
Switch(config)# vlan configuration 100
Switch(config-vlan-config)# ipv6 snooping
Switch(config-vlan-config)# exit

Switch(config)# ipv6 nd raguard policy router
Switch(config-nd-raguard)# device-role router
Switch(config-nd-raguard)# exit

Switch(config)# ipv6 dhcp guard policy server
Switch(config-dhcp-guard)# device-role server
Switch(config-dhcp-guard)# exit
```

Here, 2/1/2 is a router-facing port:

```
Switch(config)# interface fastethernet 2/1/2
Switch(config-if)# switchport
Switch(config-if)# swithport access vlan 100
Switch(config-if)# ipv6 nd raguard attach-policy router
Switch(config-if)# exit
```

Here, 1/0/17 is a DHCP server-facing port:

```
Switch(config)# interface gigabitethernet 1/0/17
Switch(config-if)# switchport access vlan 100
Switch(config-if)# ipv6 dhcp guard attach-policy server
Switch(config-if)# exit
Switch(config)# exit

Switch# show ipv6 snooping policies
Target               Type  Policy              Feature        Target range
Gi1/0/17             PORT  server              DHCP Guard     vlan all
Te2/1/2              PORT  router              RA guard       vlan all
vlan 100             VLAN  default             Snooping       vlan all
```

This example shows you how to create a snooping policy called *Test* and enable data address gleaning on it:

```
Switch(config)# ipv6 snooping policy Test
Switch(config-ipv6-snooping)# data-glean
Switch(config-ipv6-snooping)# device-role node
Switch(config-ipv6-snooping)# limit address-count 1
Switch(config-ipv6-snooping)# protocol dhcp
Switch(config-ipv6-snooping)# security-level glean
Switch(config-ipv6-snooping)# tracking enable
Switch(config-ipv6-snooping)# no trusted-port
Switch(config-ipv6-snooping)# exit
```

This example shows you how to configure snooping policy *Test*, enable data address gleaning on the policy, and enable source guard where link-local addresses are permitted and global autoconfiguration addresses are denied entry:

```
Switch(config)# ipv6 snooping policy Test
Switch(config-ipv6-snooping)# data-glean
Switch(config-ipv6-snooping)# exit
Switch(config)# ipv6 source-guard policy Test
Switch(config-sisf-sourceguard)# permit link-local
Switch(config-sisf-sourceguard)# deny global-autoconf
Switch(config-sisf-sourceguard)# exit
```

This example shows you how to attach a snooping policy with source guard, to an interface:

```
Switch(config)# interface gigabitethernet2/0/3
Switch(config-if)# ipv6 snooping attach-policy Test
Switch(config-if)# ipv6 source-guard attach-policy Test

Switch# show ipv6 source-guard policy Test
Policy Test configuration:
    permit link-local
    deny global-autoconf
Policy Test is applied on the following targets:
    Target  Type  Policy Feature      Target range
    Gi2/0/3 PORT  Test   Source guard vlan all
```

This example shows you how to configure a DHCP guard policy *Test* and attach it to an interface:

```
Switch(config)# ipv6 dhcp-guard policy Test
Switch(config-dhcp-guard)# no trusted-port
Switch(config-dhcp-guard)# exit

Switch(config)# interface gigabitethernet2/0/3
Switch(config-if)# ipv6 dhcp guard attach-policy Test
Switch(config-if)# exit
```

OR

```
Switch(config)# vlan configuration 1-10
Switch(config-vlan-config)# ipv6 dhcp guard attach-policy Test
Switch(config-vlan-config)# exit

Switch# show ipv6 dhcp-guard policy Test
Dhcp guard policy: Test
Device Role: dhcp server
Target: Gi2/0/3 vlan 1 vlan 2 vlan 3 vlan 4 vlan 5 vlan 6 vlan 7 vlan 8 vlan 9 vlan 10
Max Preference: 255
Min Preference: 0
```

This example shows how you can enable the FHS feature on an interface or VLAN, without creating a policy.

Note Creating a policy gives you the flexibility to configure as per your needs. If you enable the feature without creating a policy then the default policy configuration is applied:

```
Switch(config)# interface GigabitEthernet1/0/9
Switch(config-if)# ipv6 nd inspection
Switch(config-if)# ipv6 nd raguard
Switch(config-if)# ipv6 snooping
Switch(config-if)# ipv6 dhcp guard
Switch(config-if)# ipv6 source-guard
Switch(config-if)# end
```

OR

```
Switch(config)# vlan configuration 1
Switch(config-vlan-config)# ipv6 nd inspection
Switch(config-vlan-config)# ipv6 nd raguard
Switch(config-vlan-config)# ipv6 dhcp guard
Switch(config-vlan-config)# ipv6 snooping
```

Note You cannot apply a source-guard policy to the VLAN.

For more examples, see the Configuration Examples for Implementing First Hop Security in IPv6 section of the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Configuring IPv6 ICMP Rate Limiting

ICMP rate limiting is enabled by default with a default interval between error messages of 100 milliseconds and a bucket size (maximum number of tokens to be stored in a bucket) of 10.

Beginning in privileged EXEC mode, follow these steps to change the ICMP rate-limiting parameters:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ipv6 icmp error-interval** *interval* [*bucketsize*]	Configure the interval and bucket size for IPv6 ICMP error messages: • *interval*—The interval (in milliseconds) between tokens being added to the bucket. The range is from 0 to 2147483647 milliseconds. • *bucketsize*—(Optional) The maximum number of tokens stored in the bucket. The range is from 1 to 200.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show ipv6 interface** [*interface-id*]	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default configuration, use the **no ipv6 icmp error-interval** global configuration command.

This example shows how to configure an IPv6 ICMP error message interval of 50 milliseconds and a bucket size of 20 tokens.

```
Switch(config)#ipv6 icmp error-interval 50 20
```

Configuring Static Routes for IPv6

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 static route:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ipv6 route** *ipv6-prefix/prefix length* {*ipv6-address* | *interface-id* [*ipv6-address*]} [*administrative distance*]	Configure a static IPv6 route. • *ipv6-prefix*—The IPv6 network that is the destination of the static route. It can also be a hostname when static host routes are configured. • */prefix length*—The length of the IPv6 prefix. A decimal value that shows 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. • *ipv6-address*—The IPv6 address of the next hop that can be used to reach the specified network. The IPv6 address of the next hop need not be directly connected; recursion is done to find the IPv6 address of the directly connected next hop. The address must be specified in hexadecimal using 16-bit values between colons. • *interface-id*—Specify direct static routes from point-to-point and broadcast interfaces. With point-to-point interfaces, there is no need to specify the IPv6 address of the next hop. With broadcast interfaces, you should always specify the IPv6 address of the next hop, or ensure that the specified prefix is assigned to the link, specifying a link-local address as the next hop. You can optionally specify the IPv6 address of the next hop to which packets are sent. **Note** You must specify an *interface-id* when using a link-local address as the next hop (the link-local next hop must also be an adjacent router). • *administrative distance*—(Optional) An administrative distance. The range is 1 to 254; the default value is 1, which gives static routes precedence over any other type of route except connected routes. To configure a floating static route, use an administrative distance greater than that of the dynamic routing protocol.
Step 3	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 4	**show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix length*] [**interface** *interface-id*] [**recursive**] [**detail**] or **show ipv6 route static** [*updated*]	Verify your entries by displaying the contents of the IPv6 routing table. • **interface** *interface-id*—(Optional) Display only those static routes with the specified interface as an egress interface. • **recursive**—(Optional) Display only recursive static routes. The **recursive** keyword is mutually exclusive with the **interface** keyword, but it can be used with or without the IPv6 prefix included in the command syntax. • **detail**—(Optional) Display this additional information: – For valid recursive routes, the output path set, and maximum resolution depth. – For invalid routes, the reason why the route is not valid.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove a configured static route, use the **no ipv6 route** *ipv6-prefix/prefix length* {*ipv6-address* | *interface-id* [*ipv6-address*]} [*administrative distance*] global configuration command.

This example shows how to configure a floating static route with an administrative distance of 130 to an interface:

```
Switch(config)# ipv6 route 2001:0DB8::/32 gigabitethernet0/1 130
```

For more information about configuring static IPv6 routing, see the “Implementing Static Routes for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Displaying IPv6

For complete syntax and usage information on these commands, see the Cisco IOS command reference publications.

Table 36-2 shows the privileged EXEC commands for monitoring IPv6 on the switch.

Table 36-2 Commands for Monitoring IPv6

Command	Purpose
show ipv6 access-list	Display a summary of access lists.
show ipv6 interface *interface-id*	Display IPv6 interface status and configuration.
show ipv6 mtu	Display IPv6 MTU per destination cache.
show ipv6 neighbors	Display IPv6 neighbor cache entries.
show ipv6 prefix-list	Display a list of IPv6 prefix lists.
show ipv6 protocols	Display IPv6 routing protocols on the switch.
show ipv6 route	Display the IPv6 route table entries.
show ipv6 static	Display IPv6 static routes.
show ipv6 traffic	Display IPv6 traffic statistics.

Table 36-3 shows the privileged EXEC commands for displaying information about IPv4 and IPv6 address types.

Table 36-3 Commands for Displaying IPv4 and IPv6 Address Types

Command	Purpose
show ip http server history	Display the previous 20 connections to the HTTP server, including the IP address accessed and the time when the connection was closed.
show ip http server connection	Display the current connections to the HTTP server, including the local and remote IP addresses being accessed.
show ip http client connection	Display the configuration values for HTTP client connections to HTTP servers.
show ip http client history	Display a list of the last 20 requests made by the HTTP client to the server.

This is an example of the output from the **show ipv6 interface** privileged EXEC command:

```
Switch# show ipv6 interface
Vlan1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20B:46FF:FE2F:D940
  Global unicast address(es):
    3FFE:C000:0:1:20B:46FF:FE2F:D940, subnet is 3FFE:C000:0:1::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF2F:D940
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
<output truncated>
```

This is an example of the output from the **show ipv6 protocols** privileged EXEC command:

```
Switch# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip fer"
  Interfaces:
    Vlan6
    GigabitEthernet2/0/4
    GigabitEthernet2/0/
    GigabitEthernet1/0/12
Redistribution:
    None
```

This is an example of the output from the **show ipv6 static** privileged EXEC command:

```
Switch# show ipv6 static
IPv6 Static routes
Code: * - installed in RIB
* ::/0 via nexthop 3FFE:C000:0:7::777, distance 1
```

This is an example of the output from the **show ipv6 neighbor** privileged EXEC command:

```
Switch# show ipv6 neighbors
IPv6 Address                                Age Link-layer Addr State Interface
3FFE:C000:0:7::777                            - 0007.0007.0007  REACH Vl7
3FFE:C101:113:1::33                           - 0000.0000.0033  REACH Fa1/0/13
```

This is an example of the output from the **show ipv6 route** privileged EXEC command:

```
Switch# show ipv6 route
IPv6 Routing Table - Default - 1 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
L   FF00::/8 [0/0]
     via Null0, receive
```

This is an example of the output from the **show ipv6 traffic** privileged EXEC command.

```
Switch# show ipv6 traffic
IPv6 statistics:
  Rcvd:  1 total, 1 local destination
         0 source-routed, 0 truncated
         0 format errors, 0 hop count exceeded
         0 bad header, 0 unknown option, 0 bad source
         0 unknown protocol, 0 not a router
         0 fragments, 0 total reassembled
         0 reassembly timeouts, 0 reassembly failures
  Sent:  36861 generated, 0 forwarded
         0 fragmented into 0 fragments, 0 failed
         0 encapsulation failed, 0 no route, 0 too big
         0 RPF drops, 0 RPF suppressed drops
  Mcast: 1 received, 36861 sent

ICMP statistics:
  Rcvd: 1 input, 0 checksum errors, 0 too short
        0 unknown info type, 0 unknown error type
        unreach: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
        parameter: 0 error, 0 header, 0 option
        0 hopcount expired, 0 reassembly timeout,0 too big
        0 echo request, 0 echo reply
        0 group query, 0 group report, 0 group reduce
        1 router solicit, 0 router advert, 0 redirects
        0 neighbor solicit, 0 neighbor advert
  Sent: 10112 output, 0 rate-limited
        unreach: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
        parameter: 0 error, 0 header, 0 option
        0 hopcount expired, 0 reassembly timeout,0 too big
        0 echo request, 0 echo reply
        0 group query, 0 group report, 0 group reduce
        0 router solicit, 9944 router advert, 0 redirects
        84 neighbor solicit, 84 neighbor advert

UDP statistics:
  Rcvd: 0 input, 0 checksum errors, 0 length errors
        0 no port, 0 dropped
  Sent: 26749 output

TCP statistics:
  Rcvd: 0 input, 0 checksum errors
  Sent: 0 output, 0 retransmitted
```