clear ipv6 mobile traffic

To clear statistics associated with Mobile IPv6 traffic, use the **clear ipv6 mobile traffic** command in privileged EXEC mode.

clear ipv6 mobile traffic

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.

Usage Guidelines

The **clear ipv6 mobile traffic** command clears the statistics about the received binding updates and transmitted binding acknowledgments on a mobile node.

Examples

In the following example, statistics about binding updates and binding acknowledgments are cleared:

```
Router# clear ipv6 mobile traffic

Router# show ipv6 mobile traffic

MIPv6 statistics:
  Rcvd: 0 total
      0 truncated, 0 format errors
      0 checksum errors
    Binding Updates received:0
      0 no HA option, 0 BU's length
      0 options' length, 0 invalid CoA
  Sent: 0 generated
    Binding Acknowledgements sent:0
      0 accepted (0 prefix discovery required)
      0 reason unspecified, 0 admin prohibited
      0 insufficient resources, 0 home reg not supported
      0 not home subnet, 0 not home agent for node
      0 DAD failed, 0 sequence number
    Binding Errors sent:0
      0 no binding, 0 unknown MH
```

```
Home Agent Traffic:
  0 registrations, 0 deregistrations
  unknown time since last accepted HA registration
  unknown time since last failed HA registration
  unknown last failed registration code
  Traffic forwarded:
    0 tunneled, 0 reversed tunneled
  Dynamic Home Agent Address Discovery:
    0 requests received, 0 replies sent
  Mobile Prefix Discovery:
    0 solicitations received, 0 advertisements sent
```

Related Commands

Command	Description
binding	Configures binding options for the Mobile IPv6 home agent feature in home agent configuration mode.
show ipv6 mobile home-agent	Displays neighboring home agents.

clear ipv6 mtu

To clear the maximum transmission unit (MTU) cache of messages, use the **clear ipv6 mtu** command in privileged EXEC mode.

clear ipv6 mtu

Syntax Description

This command has no arguments or keywords.

Command Default

Messages are not cleared from the MTU cache.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 2.6	This command was introduced.

Usage Guidelines

If a router is flooded with ICMPv6 toobig messages, the router is forced to create an unlimited number of entries in the MTU cache until all available memory is consumed. Use the **clear ipv6 mtu** command to clear messages from the MTU cache.

Examples

The following example clears the MTU cache of messages:

```
Router# clear ipv6 mtu
```

Related Commands

Command	Description
ipv6 flowset	Configures flow-label marking in 1280-byte or larger packets sent by the router.

clear ipv6 multicast aaa authorization

To clear authorization parameters that restrict user access to an IPv6 multicast network, use the **clear ipv6 multicast aaa authorization** command in privileged EXEC mode.

clear ipv6 multicast aaa authorization [*interface-type interface-number*]

Syntax Description

interface-type interface-number	Interface type and number. For more information, use the question mark (**?**) online help function.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(4)T	This command was introduced.

Usage Guidelines

Using the **clear ipv6 multicast aaa authorization** command without the optional *interface-type* and *interface-number* arguments will clear all authorization parameters on a network.

Examples

The following example clears all configured authorization parameters on an IPv6 network:

```
Router# clear ipv6 multicast aaa authorization FastEthernet 1/0
```

Related Commands

Command	Description
aaa authorization multicast default	Sets parameters that restrict user access to an IPv6 multicast network.

clear ipv6 nat translation

To clear dynamic Network Address Translation—Protocol Translation (NAT-PT) translations from the dynamic state table, use the **clear ipv6 nat translation** command in privileged EXEC mode.

clear ipv6 nat translation *

Syntax Description

*	Clears all dynamic NAT-PT translations.

Command Default

Entries are deleted from the dynamic translation state table when they time out.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(13)T	This command was introduced.

Usage Guidelines

Use this command to clear entries from the dynamic translation state table before they time out. Static translation configuration is not affected by this command.

Examples

The following example shows the NAT-PT entries before and after the dynamic translation state table is cleared. Note that all the dynamic NAT-PT mappings are cleared, but the static NAT-PT configurations remain.

```
Router# show ipv6 nat translations

Prot  IPv4 source              IPv6 source
      IPv4 destination         IPv6 destination
---   ---                      ---
      192.168.123.2            2001::2

---   ---                      ---
      192.168.122.10           2001::10

tcp   192.168.124.8,11047      3002::8,11047
      192.168.123.2,23         2001::2,23

udp   192.168.124.8,52922      3002::8,52922
      192.168.123.2,69         2001::2,69

Router# clear ipv6 nat translation *
```

```
Router# show ipv6 nat translations

Prot  IPv4 source              IPv6 source
      IPv4 destination         IPv6 destination
---   ---                      ---
      192.168.123.2            2001::2

---   ---                      ---
      192.168.122.10           2001::10
```

Related Commands

Command	Description
ipv6 nat	Designates that traffic originating from or destined for the interface is subject to NAT-PT.
show ipv6 nat translations	Displays active NAT-PT translations.

clear ipv6 neighbors

To delete all entries in the IPv6 neighbor discovery cache, except static entries, use the **clear ipv6 neighbors** command in privileged EXEC mode.

Syntax for Releases 15.0(1)M, 12.2(33)SXH, and 12.2(33)SRC, and Later Releases

clear ipv6 neighbors [**interface** *type number* [**ipv6** *ipv6-address*] | **statistics** | **vrf** *table-name* [*ipv6-address* | **statistics**]]

Syntax for Release Cisco IOS XE Release 2.1 and Later Releases

clear ipv6 neighbors

Syntax Description

interface *type number*	(Optional) Clears the IPv6 neighbor discovery cache in the specified interface.
ipv6 *ipv6-address*	(Optional) Clears the IPv6 neighbor discovery cache that matches the specified IPv6 address on the specified interface.
statistics	(Optional) Clears the IPv6 neighbor discovery entry cache.
vrf	(Optional) Clears entries for a virtual private network (VPN) routing or forwarding instance.
table-name	(Optional) Table name or identifier. The value range is from 0x0 to 0xFFFFFFFF (0 to 65535 in decimal).

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(2)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
15.0(1)M	This command was modified in a release earlier than Cisco IOS Release 15.0(1)M. The **vrf** keyword and *table-name* argument were added.
12.2(33)SRC	This command was integrated into a release earlier than Cisco IOS Release 12.2(33)SRC.
Cisco IOS XE Release 2.1	This command was implemented on the Cisco ASR 1000 Series Aggregation Services Routers.

Examples

The following example deletes all entries, except static entries, in the neighbor discovery cache:

```
Router# clear ipv6 neighbors
```

The following example clears all IPv6 neighbor discovery cache entries, except static entries, on Ethernet interface 0/0:

```
Router# clear ipv6 neighbors interface Ethernet 0/0
```

The following examples clears a neighbor discovery cache entry for 2001:0DB8:1::1 on Ethernet interface 0/0:

```
Router# clear ipv6 neighbors interface Ethernet0/0 ipv6 2001:0DB8:1::1
```

Related Commands

Command	Description
ipv6 neighbor	Configures a static entry in the IPv6 neighbor discovery cache.
show ipv6 neighbors	Displays IPv6 neighbor discovery cache information.

clear ipv6 nhrp

To clear all dynamic entries from the Next Hop Resolution Protocol (NHRP) cache, use the **clear ipv6 nhrp** command in privileged EXEC mode.

clear ipv6 nhrp [*ipv6-address* | **counters**]

Syntax Description

ipv6-address	(Optional) The IPv6 network to delete.
counters	(Optional) Specifies NHRP counters to delete.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(20)T	This command was introduced.

Usage Guidelines

This command does not clear any static (configured) IPv6-to-nonbroadcast multiaccess (NBMA) address mappings from the NHRP cache.

Examples

The following example shows how to clear all dynamic entries from the NHRP cache for the interface:

```
Router# clear ipv6 nhrp
```

Related Commands

Command	Description
show ipv6 nhrp	Displays the NHRP cache.

clear ipv6 ospf

To clear the Open Shortest Path First (OSPF) state based on the OSPF routing process ID, use the **clear ipv6 ospf** command in privileged EXEC mode.

clear ipv6 ospf [*process-id*] {**process** | **force-spf** | **redistribution**}

Syntax Description

process-id	(Optional) Internal identification. It is locally assigned and can be any positive integer. The number used here is the number assigned administratively when enabling the OSPF routing process.
process	Restarts the OSPF process.
force-spf	Starts the shortest path first (SPF) algorithm without first clearing the OSPF database.
redistribution	Clears OSPF route redistribution.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS XE Release 2.1.
15.0(1)M	This command was integrated into Cisco IOS Release 12.5(1)M.

Usage Guidelines

When the **process** keyword is used with the **clear ipv6 ospf** command, the OSPF database is cleared and repopulated, and then the shortest path first (SPF) algorithm is performed. When the **force-spf** keyword is used with the **clear ipv6 ospf** command, the OSPF database is not cleared before the SPF algorithm is performed.

Use the *process-id* option to clear only one OSPF process. If the *process-id* option is not specified, all OSPF processes are cleared.

Examples

The following example starts the SPF algorithm without clearing the OSPF database:

```
Router# clear ipv6 ospf force-spf
```

clear ipv6 ospf counters

To clear the Open Shortest Path First (OSPF) state based on the OSPF routing process ID, use the **clear ipv6 ospf** command in privileged EXEC mode.

clear ipv6 ospf [*process-id*] **counters** [**neighbor** [*neighbor-interface* | *neighbor-id*]]

Syntax Description

process-id	(Optional) Internal identification. It is locally assigned and can be any positive integer. The number used here is the number assigned administratively when enabling the OSPF routing process.
neighbor	(Optional) Neighbor statistics per interface or neighbor ID.
neighbor-interface	(Optional) Neighbor interface.
neighbor-id	(Optional) IPv6 or IP address of the neighbor.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

Use the **neighbor** *neighbor-interface* option to clear counters for all neighbors on a specified interface. If the **neighbor** *neighbor-interface* option is not used, all OSPF counters are cleared.

Use the **neighbor** *neighbor-id* option to clear counters at a specified neighbor. If the **neighbor** *neighbor-id* option is not used, all OSPF counters are cleared.

Examples

The following example provides detailed information on a neighbor router:

```
Router# show ipv6 ospf neighbor detail

 Neighbor 10.0.0.1
    In the area 1 via interface Serial19/0
    Neighbor:interface-id 21, link-local address FE80::A8BB:CCFF:FE00:6F00
    Neighbor priority is 1, State is FULL, 6 state changes
    Options is 0x194AE05
    Dead timer due in 00:00:37
    Neighbor is up for 00:00:15
    Index 1/1/1, retransmission queue length 0, number of retransmission 1
    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

The following example clears all neighbors on the specified interface:

```
Router# clear ipv6 ospf counters neighbor s19/0
```

The following example now shows that there have been 0 state changes since the **clear ipv6 ospf counters neighbor s19/0** command was used:

```
Router# show ipv6 ospf neighbor detail

 Neighbor 10.0.0.1
    In the area 1 via interface Serial19/0
    Neighbor:interface-id 21, link-local address FE80::A8BB:CCFF:FE00:6F00
    Neighbor priority is 1, State is FULL, 0 state changes
    Options is 0x194AE05
    Dead timer due in 00:00:39
    Neighbor is up for 00:00:43
    Index 1/1/1, retransmission queue length 0, number of retransmission 1
    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

Related Commands

Command	Description
show ipv6 ospf neighbor	Displays OSPF neighbor information on a per-interface basis.

clear ipv6 ospf events

To clear the Open Shortest Path First (OSPF) for IPv6 event log content based on the OSPF routing process ID, use the **clear ipv6 ospf events** command in privileged EXEC mode.

clear ipv6 ospf [*process-id*] **events**

Syntax Description

process-id	(Optional) Internal identification. It is locally assigned and can be any positive integer. The number used here is the number assigned administratively when enabling the OSPF routing process.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRC	This command was introduced.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 series routers.
12.2(33)XNE	This command was modified. It was integrated into Cisco IOS Release 12.2(33)XNE.

Usage Guidelines

Use the optional *process-id* argument to clear the IPv6 event log content of a specified OSPF routing process. If the *process-id* argument is not used, all event log content is cleared.

Examples

The following example enables the clearing of OSPF for IPv6 event log content for routing process 1:

```
Router# clear ipv6 ospf 1 events
```

clear ipv6 pim counters

To reset the Protocol Independent Multicast (PIM) traffic counters, use the **clear ipv6 pim counters** command in privileged EXEC mode.

clear ipv6 pim counters

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

Usage Guidelines

Using the **clear ipv6 pim counters** command will reset all PIM traffic counters.

Examples

The following example resets the PIM traffic counters:

```
Router# clear ipv6 pim counters
```

Related Commands

Command	Description
show ipv6 pim traffic	Displays the PIM traffic counters.

clear ipv6 pim limit

To clear Protocol Independent Multicast (PIM) statistics, use the **clear ipv6 pim limit** command in privileged EXEC mode.

clear ipv6 pim [**vrf** *vrf-name*] **limit** [*interface*]

Syntax Description

vrf *vrf-name*	(Optional) Specifies a virtual routing and forwarding (VRF) configuration.
interface	(Optional) Specific interface for which statistics will be cleared.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRE	This command was introduced.
15.1(4)M	The **vrf** *vrf-name* keyword and argument were added.

Usage Guidelines

The **clear ipv6 pim limit** command clears IPv6 PIM interface statistics. If the optional *interface* argument is enabled, only statistics for the specified interface are cleared.

Examples

The following example clears PIM interface limit statistics:

```
Router# clear ipv6 pim limit
```

Related Commands

Command	Description
ipv6 multicast limit	Configures per-interface mroute state limiters in IPv6.
ipv6 multicast limit cost	Applies a cost to mroutes that match per interface mroute state limiters in IPv6.

clear ipv6 pim reset

To delete all entries from the topology table and reset the Multicast Routing Information Base (MRIB) connection, use the **clear ipv6 pim reset** command in privileged EXEC mode.

clear ipv6 pim [**vrf** *vrf-name*] **reset**

Syntax Description

vrf *vrf-name*	(Optional) Specifies a virtual routing and forwarding (VRF) configuration.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(2)T	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
15.1(4)M	The **vrf** *vrf-name* keyword and argument were added.

Usage Guidelines

Using the **clear ipv6 pim reset** command breaks the PIM-MRIB connection, clears the topology table, and then reestablishes the PIM-MRIB connection. This procedure forces MRIB resynchronization.

Caution Use the **clear ipv6 pim reset** command with caution, as it clears all PIM protocol information from the PIM topology table. Use of the **clear ipv6 pim reset** command should be reserved for situations where PIM and MRIB communication are malfunctioning.

Examples

The following example deletes all entries from the topology table and resets the MRIB connection:

```
Router# clear ipv6 pim reset
```

clear ipv6 pim topology

To clear the Protocol Independent Multicast (PIM) topology table, use the **clear ipv6 pim topology** command in privileged EXEC mode.

clear ipv6 pim [**vrf** *vrf-name*] **topology** [*group-name* | *group-address*]

Syntax Description

vrf *vrf-name*	(Optional) Specifies a virtual routing and forwarding (VRF) configuration.
group-name | *group-address*	(Optional) IPv6 address or name of the multicast group.

Command Default

When the command is used with no arguments, all group entries located in the PIM topology table are cleared of PIM protocol information.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(2)T	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.
15.1(4)M	The **vrf** *vrf-name* keyword and argument were added.

Usage Guidelines

This command clears PIM protocol information from all group entries located in the PIM topology table. Information obtained from the MRIB table is retained. If a multicast group is specified, only those group entries are cleared.

Examples

The following example clears all group entries located in the PIM topology table:

```
Router# clear ipv6 pim topology
```

clear ipv6 pim traffic

To clear the Protocol Independent Multicast (PIM) traffic counters, use the **clear ipv6 pim traffic** command in privileged EXEC mode.

clear ipv6 pim [**vrf** *vrf-name*] **traffic**

Syntax Description

vrf *vrf-name*	(Optional) Specifies a virtual routing and forwarding (VRF) configuration.

Command Default

When the command is used with no arguments, all traffic counters are cleared.

Command Modes

Privileged EXEC

Command History

Release	Modification
15.1(4)M	This command was introduced.

Usage Guidelines

This command clears PIM traffic counters. If the **vrf** *vrf-name* keyword and argument are used, only those counters are cleared.

Examples

The following example clears all PIM traffic counter:

```
Router# clear ipv6 pim traffic
```

clear ipv6 prefix-list

To reset the hit count of the IPv6 prefix list entries, use the **clear ipv6 prefix-list** command in privileged EXEC mode.

clear ipv6 prefix-list [*prefix-list-name*] [*ipv6-prefix*/*prefix-length*]

Syntax Description

prefix-list-name	(Optional) The name of the prefix list from which the hit count is to be cleared.
ipv6-prefix	(Optional) The IPv6 network from which the hit count is to be cleared. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
/*prefix-length*	(Optional) The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.

Command Default

The hit count is automatically cleared for all IPv6 prefix lists.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

Usage Guidelines

The **clear ipv6 prefix-list** command is similar to the **clear ip prefix-list** command, except that it is IPv6-specific.

The hit count is a value indicating the number of matches to a specific prefix list entry.

Examples

The following example clears the hit count from the prefix list entries for the prefix list named first_list that match the network mask 2001:0DB8::/35.

```
Router# clear ipv6 prefix-list first_list 2001:0DB8::/35
```

Related Commands

Command	Description
ipv6 prefix-list	Creates an entry in an IPv6 prefix list.
ipv6 prefix-list sequence-number	Enables the generation of sequence numbers for entries in an IPv6 prefix list.
show ipv6 prefix-list	Displays information about an IPv6 prefix list or prefix list entries.

clear ipv6 rip

To delete routes from the IPv6 Routing Information Protocol (RIP) routing table, use the **clear ipv6 rip** command in privileged EXEC mode.

clear ipv6 rip [*name*]

Syntax Description

name	(Optional) Name of an IPv6 RIP process.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(22)S	This command was introduced.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

When the *name* argument is specified, only routes for that process are deleted from the IPv6 RIP routing table and, if installed, from the IPv6 routing table. If no *name* argument is specified, all IPv6 RIP routes are deleted.

Use the **show ipv6 rip** command to display IPv6 RIP routes.

Examples

The following example deletes all the IPv6 routes for the RIP process called one:

```
Router# clear ipv6 rip one
```

Related Commands

Command	Description
show ipv6 rip	Displays the current contents of the IPv6 RIP routing table.

clear ipv6 route

To delete routes from the IPv6 routing table, use the **clear ipv6 route** command in privileged EXEC mode.

clear ipv6 route {*ipv6-address* | *ipv6-prefix*/*prefix-length* | *}

Syntax Description

ipv6-address	The address of the IPv6 network to delete from the table. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
ipv6-prefix	The IPv6 network number to delete from the table. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
/*prefix-length*	The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
*	Clears all IPv6 routes.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

The **clear ipv6 route** command is similar to the **clear ip route** command, except that it is IPv6-specific.

When the *ipv6-address* or *ipv6-prefix*/*prefix-length* argument is specified, only that route is deleted from the IPv6 routing table. When the * keyword is specified, all routes are deleted from the routing table (the per-destination maximum transmission unit [MTU] cache is also cleared).

Examples

The following example deletes the IPv6 network 2001:0DB8::/35:

```
Router# clear ipv6 route 2001:0DB8::/35
```

Related Commands

Command	Description
ipv6 route	Establishes static IPv6 routes.
show ipv6 route	Displays the current contents of the IPv6 routing table.

clear ipv6 snooping counters

To remove counter entries, use the **clear ipv6 snooping counters** command in privileged EXEC mode.

clear ipv6 snooping counters [**interface** *type number*]

Syntax Description

interface *type number*	(Optional) Clears the counter of entries that match the specified interface type and number.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(50)SY	This command was introduced.

Usage Guidelines

The **clear ipv6 snooping counters** command removes counters from all the configured interfaces. You can use the optional **interface** *type number* keyword and argument to remove counters from the specified interface.

Examples

The following example shows how to remove entries from the counter:

```
Router# clear ipv6 snooping counters
```

clear ipv6 spd

To clear the most recent Selective Packet Discard (SPD) state transition, use the **clear ipv6 spd** command in privileged EXEC mode.

clear ipv6 spd

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(3)T	This command was introduced.

Usage Guidelines

The **clear ipv6 spd** command removes the most recent SPD state transition and any trend historical data.

Examples

The following example shows how to clear the most recent SPD state transition:

```
Router# clear ipv6 spd
```

clear ipv6 traffic

To reset IPv6 traffic counters, use the **clear ipv6 traffic** command in privileged EXEC mode.

clear ipv6 traffic [*interface-type interface-number]*

Syntax Description

interface-type interface-number	Interface type and number. For more information, use the question mark (**?**) online help function.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S and output fields were added.
12.2(13)T	The modification to add output fields was integrated into this release.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2(33)XN	The optional *interface-type* and *interface-number* arguments were added.

Usage Guidelines

Using this command resets the counters in the output from the **show ipv6 traffic** command.

Examples

The following example resets the IPv6 traffic counters. The output from the **show ipv6 traffic** command shows that the counters are reset:

```
Router# clear ipv6 traffic

Router# 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:  1 generated, 0 forwarded
         0 fragmented into 0 fragments, 0 failed
         0 encapsulation failed, 0 no route, 0 too big
  Mcast: 0 received, 0 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
        0 router solicit, 0 router advert, 0 redirects
        0 neighbor solicit, 1 neighbor advert
Sent: 1 output
        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, 0 router advert, 0 redirects
        0 neighbor solicit, 1 neighbor advert

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

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

Related Commands

Command	Description
show ipv6 traffic	Displays IPv6 traffic statistics.

clear mls cef ipv6 accounting per-prefix

To clear information about the IPv6 per-prefix accounting statistics, use the **clear mls cef ipv6 accounting per-prefix** command in privileged EXEC mode.

clear mls cef ipv6 accounting per-prefix {**all** | *ipv6-address*/*mask* [*instance*]}

Syntax Description

all	Clears all per-prefix accounting statistics information.
ipv6-address/*mask*	Entry IPv6 address and mask. The format used is X:X:X:X::X/*mask,* where the valid values for *mask* are from 0 to 128.
instance	(Optional) VPN routing and forwarding instance name.

Command Default

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(17a)SX	This command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to Release 12.2(17d)SXB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

When entering the *ipv6-address*/*mask* arguments, use this format, X:X:X:X::X/*mask,* where the valid values for *mask* are from 0 to 128.

Examples

This example shows how to clear all information about the per-prefix accounting statistics:

```
Router# clear mls cef ipv6 accounting per-prefix all
```

clear ospfv3 counters

To clear Open Shortest Path First version 3 (OSPFv3) counters, use the **clear ospfv3 counters** command in privileged EXEC mode.

clear ospfv3 [*process-id*] [*address-family*] **counters** [**neighbor** [*neighbor-interface* | *neighbor-id*]]

Syntax Description

process-id	(Optional) Internal identification. The number used here is the number assigned administratively when enabling the OSPFv3 routing process and can be a value from 1 through 65535.
address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.
neighbor	(Optional) Neighbor statistics per interface or neighbor ID.
neighbor-interface	(Optional) Specified neighbor interface.
neighbor-id	(Optional) IPv6 or IPv4 address of the neighbor.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

Use the **neighbor** *neighbor-interface* option to clear counters for all neighbors on a specified interface. If the **neighbor** *neighbor-interface* option is not used, all OSPFv3 counters are cleared.

Examples

The following example clears all neighbors on the serial 19/0 interface:

```
Router# clear ospfv3 counters neighbor s19/0
```

clear ospfv3 force-spf

To run shortest path first (SPF) calculations for an Open Shortest Path First version 3 (OSPFv3) process, use the **clear ospfv3 counters** command in privileged EXEC mode.

clear ospfv3 [*process-id*] [*address-family*] **force-spf**

Syntax Description

process-id	(Optional) Internal identification. The number used here is the number assigned administratively when enabling the OSPFv3 routing process and can be a value from 1 through 65535.
address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

Use the **clear ospv3 force-spf** command to run SPF calculations for either an IPv6 or an IPv4 OSPFv3 instance. If the optional *process-ID* argument is not used, SPF runs on all instances on the interface.

Examples

The following example enables SPF calculations for process 1:

```
Router# clear ospfv3 1 force-spf
```

clear ospfv3 process

To reset an Open Shortest Path First version 3 (OSPFv3) process, use the **clear ospfv3 process** command in privileged EXEC mode.

clear ospfv3 [*process-id*] [*address-family*] **process**

Syntax Description

process-id	(Optional) Internal identification. The number used here is the number assigned administratively when enabling the OSPFv3 routing process and can be a value from 1 through 65535.
address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

Use the **clear ospv3 process** command to reset either an IPv6 or IPv4 OSPFv3 process. If the optional *process-ID* argument is not used, all OSPFv3 processes are reset.

Examples

The following example resets the OSPFv3 process 2:

```
Router# clear ospfv3 2 process
```

clear ospfv3 redistribution

To clear Open Shortest Path First version 3 (OSPFv3) route redistribution, use the **clear ospfv3 process** command in privileged EXEC mode.

clear ospfv3 [*process-id*] [*address-family*] **redistribution**

Syntax Description

process-id	(Optional) Internal identification. The number used here is the number assigned administratively when enabling the OSPFv3 routing process and can be a value from 1 through 65535.
address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

Use the **clear ospv3 process** command to clear either IPv6 or IPv4 OSPFv3 redistribution. If the optional *process-ID* argument is not used, all processes on the interface are cleared.

Examples

The following example clears OSPFv3 redistribution on all OSPFv3 processes:

```
Router# clear ospfv3 redistribution
```

clear ospfv3 traffic

To reset counters and clear Open Shortest Path First version 3 (OSPFv3) traffic statistics, use the **clear ospfv3 traffic** command privileged EXEC mode.

clear ospfv3 [*process-id*] [*address-family*] **traffic** [*interface*]

Syntax Description

process-id	(Optional) Internal identification. The number used here is the number assigned administratively when enabling the OSPFv3 routing process and can be a value from 1 through 65535.
address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.
interface	(Optional) Specified interface from which to clear traffic statistics.

Command Modes

Privileged EXEC

Command History

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

Usage Guidelines

Use the **clear ospv3 traffic** command to reset traffic statistics for an IPv6 or IPv4 OSPFv3 process. If the optional *process-ID* argument is not used, all traffic statistics are cleared.

Examples

The following example resets the counters and clears the OSPFv3 traffics statistics:

```
Router# clear ospfv3 traffic
```

codec (DSP farm profile)

To specify the codecs that are supported by a digital signal processor (DSP) farm profile, use the **codec** command in DSP farm profile configuration mode. To remove the codec, use the **no** form of this command.

codec {*codec-type* [*resolution*] | [**frame-rate** *framerate*] | [**bitrate** *bitrate*] | [**rfc-2190**] | **pass-through**}

no codec {*codec-type* [*resolution*] | [**frame-rate** *framerate*] | [**bitrate** *bitrate*] | [**rfc-2190**] | **pass-through**}

Syntax Description

codec-type	Specifies the codec preferred. • **g711alaw**—G.711 a-law 64,000 bits per second (bps) • **g711ulaw**—G.711 mu-law 64,000 bps • **g722r-64**—G.722-64 at 64,000 bps • **g729abr8**—G.729 ANNEX A and B 8000 bps • **g729ar8**—G.729 ANNEX A 8000 bps • **g729br8**—G.729 ANNEX B 8000 bps • **g729r8**—G.729 8000 bps • **h263**—H.263 video codec • **h264**—H.264 video codec • **ilbc**—Internet Low Bitrate Codec (iLBC) • **isac**—Cisco internet Speech Audio Codec (iSAC) codec
resolution	Specifies the supported video resolution. The valid entries are: • For H.263—**qcif** and **cif** • For H.264—**qcif**, **cif**, **vga**, **w360p**, **w448p**, **4cif**, and **720p** **Note** 720p option applies only to homogeneous video conferences.
frame-rate *framerate*	Specifies the frame rate. The valid entries are 15 fps or 30 fps. This option applies to homogeneous conferences only.
bitrate *bitrate*	Specifies the bitrate. This option applies to homogeneous conferences only.
rfc-2190	Specifies the payload format follow RFC-2190.
pass-through	Enables codec pass-through. Supported for transcoding and media termination point (MTP) profiles.

Command Default

The following transcoding defaults apply when you are configuring audio profiles only. When you configure video transcoding, you must specify the audio codecs.

Transcoding

- **g711alaw**

- **g711ulaw**
- **g729abr8**
- **g729ar8**

Conferencing

- **g711alaw**
- **g711ulaw**
- **g729abr8**
- **g729ar8**
- **g729br8**
- **g729r8**

MTP

- **g711ulaw**

Command Modes

DSP farm profile configuration (config-dspfarm-profile)

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.4(4)T	The **pass-through** keyword was added.
12.4(11)XJ2	The **gsmefr** and **gsmfr** keywords were removed as configurable codec options for all platforms.
12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)T.
12.4(15)XY	The **g722r-64** keyword was added.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.
12.4(22)T	Support for IPv6 was added.
15.1(1))T	This command was modified. The **isac** keyword was added.
15.1(4)M	This command was modified. The **frame-rate**, **bitrate**, **rfc-2190**, and **pass-through** keywords were added and codec support was added for **ilbc**, **h.263** and **h.264**.

Usage Guidelines

Only one codec is supported for each MTP profile. To support multiple codecs, you must define a separate MTP profile for each codec.

For homogeneous video profiles, only one video format is supported

For heterogeneous and heterogeneous guaranteed-audio video profiles, multiple video formats and audio codecs are supported.

To change the configured codec in the profile, you must first enter a **no maximum session** command.

Table 8 shows the relationship between DSP farm functions and codecs.

Table 8 DSP Farm Functions and Codec Relationships

DSP Farm Function	Supported Codec
Transcoding	• **g711alaw** • **g711ulaw** • **g729abr8** • **g729ar8** • **iSAC** • **h263** • **h264**
Conferencing	• **g711alaw** • **g711ulaw** • **g722r-64** • **g729abr8** • **g729ar8** • **g729br8** • **g729r8** • **h263** • **h264** • **ilbc**
MTP	• **g711ulaw** • **iSAC**

Hardware MTPs support only G.711 a-law and G.711 mu-law. If you configure a profile as a hardware MTP and you want to change the codec to other than G.711, you must first remove the hardware MTP by using the **no maximum sessions hardware** command.

The **pass-through** keyword is supported for transcoding and MTP profiles only; the keyword is not supported for conferencing profiles. To support the Resource Reservation Protocol (RSVP) agent on a Skinny Client Control Protocol (SCCP) device, you must use the **codec pass-through** command. In the pass-through mode, the SCCP device processes the media stream by using a pure software MTP, regardless of the nature of the stream, which enables video and data streams to be processed in addition to audio streams. When the pass-through mode is set in a transcoding profile, no transcoding is done for the session; the transcoding device performs a pure software MTP function. The pass-through mode can be used for secure Real-Time Transport Protocol (RTP) sessions.

Examples

The following example shows how to set the call density and codec complexity to g729abr8:

```
Router(config)# dspfarm profile 123 transcode
Router(config-dspfarm-profile)# codec g729abr8
```

The following example shows how to set up a video conference with guaranteed-audio.

```
Router(config)# dspfarm profile 99 conference video guaranteed-audio
Router(config-dspfarm-profile)# codec h264 4cif
Router(config-dspfarm-profile)# codec h264 cif
Router(config-dspfarm-profile)# maximum conference-participants 8
```

Related Commands

Command	Description
associate application	Associates the SCCP protocol to the DSP farm profile.
dspfarm profile	Enters DSP farm profile configuration mode and defines a profile for DSP farm services.
maximum sessions (DSP Farm profile)	Specifies the maximum number of sessions that are supported by the profile.
rsvp	Enables RSVP support on a transcoding or MTP device.
maximum conference-participants (DSP Farm profile)	Specifies the maximum number of conference participants that are supported by this profile.
shutdown (DSP Farm profile)	Disables a DSP farm profile.

compatible rfc1583

To calculate the method used to calculate external route preferences per RFC 1583, use the **compatible rfc1583** command in router configuration mode. To disable RFC 1583 compatibility, use the **no** form of this command.

compatible rfc1583

no compatible rfc1583

Syntax Description

This command has no arguments or keywords.

Command Default

Compatible with RFC 1583.

Command Modes

OSPFv3 router configuration mode (config-router)

Command History

Release	Modification
15.1(3)S	This command was introduced.
Cisco IOS XE Release 3.4S	This command was integrated into Cisco IOS XE Release 3.4S.

Usage Guidelines

RFC 2328 describes a new method of calculating path preferences for AS external routes:

- Intra-area paths using non-backbone areas are always the most preferred.
- The other paths—intra-area backbone paths and inter-area paths—are of equal preference.

Use the **no compatible rfc1583** command to enable the calculation method used per RFC 2328. For more detailed information, see the "RFC 1583 Compatibility" section in RFC 2328.

Caution To minimize the chance of routing loops, all Open Shortest Path First (OSPF) routers in an OSPF routing domain should have RFC compatibility set identically.

Examples

The following example specifies that the router process is compatible with RFC 1583:

```
router ospf 1
 compatible rfc1583
 !
```

context

Note Effective with Cisco IOS Release 15.0(1)M, the **context** command is replaced by the **snmp context** command. See the **snmp context** command for more information.

To associate a Simple Network Management Protocol (SNMP) context with a particular VPN routing and forwarding (VRF) instance, use the **context** command in VRF configuration mode. To disassociate an SNMP context from a VPN, use the **no** form of this command.

context *context-name*

no context

Syntax Description

context-name	Name of the SNMP VPN context. The name can be up to 32 alphanumeric characters.

Command Default

No SNMP contexts are associated with VPNs.

Command Modes

VRF configuration (config-vrf)

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.3(2)T	This command was integrated into Cisco IOS Release 12.3(2)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB2.
12.2(33)SRB	This command was modified. Support for IPv6 was added.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
15.0(1)M	This command was replaced by the **snmp context** command.

Usage Guidelines

Before you use the **context** command to associate an SNMP context with a VPN, you must do the following:

- Issue the **snmp-server context** command to create an SNMP context.
- Associate a VPN with a context so that the specific MIB data for that VPN exists in the context.
- Associate a VPN group with the context of the VPN using the **context** *context-name* keyword argument pair of the **snmp-server group** command.

SNMP contexts provide VPN users with a secure way of accessing MIB data. When a VPN is associated with a context, MIB data for that VPN exists in that context. Associating a VPN with a context helps service providers to manage networks with multiple VPNs. Creating and associating a context with a VPN enables a provider to prevent the users of one VPN from accessing information about other VPN users on the same networking device.

A route distinguisher (RD) is required to configure an SNMP context. An RD creates routing and forwarding tables and specifies the default route distinguisher for a VPN. The RD is added to the beginning of an IPv4 prefix to make it globally unique. An RD is either an autonomous system number (ASN) relative, which means that it is composed of an autonomous system number and an arbitrary number, or an IP address relative and is composed of an IP address and an arbitrary number.

Examples

The following example shows how to create an SNMP context named context1 and associate the context with the VRF named vrf1:

```
Router(config)# snmp-server context context1
Router(config)# ip vrf vrf1
Router(config-vrf)# rd 100:120
Router(config-vrf)# context context1
```

Related Commands

Command	Description
ip vrf	Enters VRF configuration mode for the configuration of a VRF.
snmp mib community-map	Associates an SNMP community with an SNMP context, engine ID, or security name.
snmp mib target list	Creates a list of target VRFs and hosts to associate with an SNMP v1 or v2c community.
snmp-server context	Creates an SNMP context.
snmp-server group	Configures a new SNMP group or a table that maps SNMP users to SNMP views.
snmp-server trap authentication vrf	Controls VRF-specific SNMP authentication failure notifications.
snmp-server user	Configures a new user to an SNMP group.

crypto ipsec profile

To define the IP Security (IPsec) parameters that are to be used for IPsec encryption between two IPsec routers and to enter IPsec profile configuration mode, use the **crypto ipsec profile** command in global configuration mode. To delete an IPsec profile, use the **no** form of this command.

crypto ipsec profile *name*

no crypto ipsec profile *name*

Syntax Description

name	Profile name.

Command Default

An IPsec profile is not defined.

Command Modes

Global configuration

Command History

Release	Modification
12.2(13)T	This command was introduced.
12.2(18)SXE	This command was integrated into Cisco IOS Release 12.2(18)SXE.
12.4(4)T	Support for IPv6 was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

Usage Guidelines

An IPsec profile abstracts the IPsec policy settings into a single profile that can be used in other parts of the Cisco IOS configuration.

The IPsec profile shares most of the same commands with the crypto map configuration, but only a subset of the commands are valid in an IPsec profile. Only commands that pertain to an IPsec policy can be issued under an IPsec profile; you cannot specify the IPsec peer address or the access control list (ACL) to match the packets that are to be encrypted.

After this command has been enabled, the following commands can be configured under an IPsec profile:

- **default**—Lists the commands that can be configured under the **crypto ipsec profile** command.
- **description**—Describes the crypto map statement policy.
- **dialer**—Specifies dialer-related commands.
- **redundancy**—Specifies a redundancy group name.
- **set-identity**—Specifies identity restrictions.
- **set isakmp-profile**—Specifies an ISAKMP profile.
- **set pfs**—Specifies perfect forward secrecy (PFS) settings.
- **set security-association**—Defines security association parameters.

- **set-transform-set**—Specifies a list of transform sets in order of priority.

After enabling this command, the only parameter that *must* be defined under the profile is the transform set via the **set transform-set** command.

For more information on transform sets, refer to the section "Defining Transform Sets" in the chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*.

Examples

The following example shows how to configure a crypto map that uses an IPsec profile:

```
crypto ipsec transform-set cat-transforms esp-des esp-sha-hmac
 mode transport
!
crypto ipsec profile cat-profile
 set transform-set cat-transforms
 set pfs group2
!
interface Tunnel1
 ip address 192.168.1.1 255.255.255.252
 tunnel source FastEthernet2/0
 tunnel destination 10.13.7.67
 tunnel protection ipsec profile cat-profile
```

Related Commands

Command	Description
crypto ipsec transform-set	Defines a transform set.
set pfs	Specifies that IPsec should ask for PFS when requesting new security associations for a crypto map entry.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
tunnel protection	Associates a tunnel interface with an IPsec profile.

crypto isakmp identity

To define the ISAKMP identity used by the router when participating in the Internet Key Exchange (IKE) protocol, use the **crypto isakmp identity** command in global configuration mode. To reset the ISAKMP identity to the default value (address), use the **no** form of this command.

crypto isakmp identity {**address** | **dn** | **hostname**}

no crypto isakmp identity

Syntax Description

address	Sets the ISAKMP identity to the IP address of the interface that is used to communicate to the remote peer during IKE negotiations.
dn	Sets the ISAKMP identity to the distinguished name (DN) of the router certificate.
hostname	Sets the ISAKMP identity to the host name concatenated with the domain name (for example, myhost.example.com).

Command Default

The IP address is used for the ISAKMP identity.

Command Modes

Global configuration

Command History

Release	Modification
11.3T	This command was introduced.
12.4(4)T	Support for IPv6 was added.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command to specify an ISAKMP identity either by IP address, DN or host name. An ISAKMP identity is set whenever you specify preshared keys or RSA signature authentication.

The **address** keyword is typically used when only one interface (and therefore only one IP address) will be used by the peer for IKE negotiations, and the IP address is known.

The **dn** keyword should be used if the DN of a router certificate is to be specified and chosen as the ISAKMP identity during IKE processing. The **dn** keyword is used only for certificate-based authentication.

The **hostname** keyword should be used if more than one interface on the peer might be used for IKE negotiations, or if the interface's IP address is unknown (such as with dynamically assigned IP addresses).

As a general rule, you should set all peers' identities in the same way, either by IP address or by host name.

Examples

The following example uses preshared keys at two peers and sets both their ISAKMP identities to the IP address.

At the local peer (at 10.0.0.1) the ISAKMP identity is set and the preshared key is specified:

```
crypto isakmp identity address
crypto isakmp key sharedkeystring address 192.168.1.33
```

At the remote peer (at 192.168.1.33) the ISAKMP identity is set and the same preshared key is specified:

```
crypto isakmp identity address
crypto isakmp key sharedkeystring address 10.0.0.1
```

Note In the preceding example if the **crypto isakmp identity** command had not been performed, the ISAKMP identities would have still been set to IP address, the default identity.

The following example uses preshared keys at two peers and sets both their ISAKMP identities to the hostname.

At the local peer the ISAKMP identity is set and the preshared key is specified:

```
crypto isakmp identity hostname
crypto isakmp key sharedkeystring hostname RemoteRouter.example.com
ip host RemoteRouter.example.com 192.168.0.1
```

At the remote peer the ISAKMP identity is set and the same preshared key is specified:

```
crypto isakmp identity hostname
crypto isakmp key sharedkeystring hostname LocalRouter.example.com
ip host LocalRouter.example.com 10.0.0.1 10.0.0.2
```

In the example, hostnames are used for the peers' identities because the local peer has two interfaces that might be used during an IKE negotiation.

In the example the IP addresses are also mapped to the hostnames; this mapping is not necessary if the routers' hostnames are already mapped in DNS.

Related Commands

Command	Description
crypto ipsec security-association lifetime	Specifies the authentication method within an IKE policy.
crypto isakmp key	Configures a preshared authentication key.

crypto isakmp key

To configure a preshared authentication key, use the **crypto isakmp key** command in global configuration mode. To delete a preshared authentication key, use the **no** form of this command.

crypto isakmp key *enc-type-digit keystring* {**address** *peer-address* [*mask*] | **ipv6** *ipv6-address*/*ipv6-prefix* | **hostname** *hostname*} [**no-xauth**]

no crypto isakmp key *enc-type-digit keystring* {**address** *peer-address* [*mask*] | **ipv6** *ipv6-address*/*ipv6-prefix* | **hostname** *hostname*} [**no-xauth**]

Syntax Description

enc-type-digit	Specifies whether the password to be used is encrypted or unencrypted. • 0—Specifies that an unencrypted password follows. • 6—Specifies that an encrypted password follows.
keystring	Specifies the preshared key. Use any combination of alphanumeric or special characters up to 128 bytes. Special characters include the following: !?"#$%&'()*+,-./:;<=>@[\]^_`~. (Type "CTRL-V" before the "?" symbol to avoid invoking help.) This preshared key must be identical at both peers.
address	Use this keyword if the remote peer Internet Security Association Key Management Protocol (ISAKMP) identity was set with its IP or IPv6 address. The *peer-address* argument specifies the IP or IPv6 address of the remote peer.
peer-address	Specifies the IP address of the remote peer.
mask	(Optional) Specifies the subnet address of the remote peer. (The argument can be used only if the remote peer ISAKMP identity was set with its IP address.)
ipv6	Specifies that an IPv6 address of a remote peer will be used.
ipv6-address	IPv6 address of the remote peer. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
ipv6-prefix	IPv6 prefix of the remote peer.
hostname *hostname*	Fully qualified domain name (FQDN) of the peer. The **hostname** keyword and *hostname* argument are not supported by IPv6.
no-xauth	(Optional) Use this keyword if router-to-router IP Security (IPSec) is on the same crypto map as a Virtual Private Network (VPN)-client-to-Cisco-IOS IPSec. This keyword prevents the router from prompting the peer for extended authentication (Xauth) information (username and password).

Command Default

There is no default preshared authentication key.

Command Modes

Global configuration

Command History

Release	Modification
11.3T	This command was introduced.
12.1(1)T	The *mask* argument was added.
12.2(4)T	The **no-xauth** keyword was added.
12.3(2)T	This command was modified so that output shows that the preshared key is either encrypted or unencrypted.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.4(4)T	The **ipv6** keyword and the *ipv6-address* and *ipv6-prefix* arguments were added.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

Usage Guidelines

You must use this command to configure a key whenever you specify preshared keys in an Internet Key Exchange (IKE) policy; you must enable this command at both peers.

If an IKE policy includes preshared keys as the authentication method, these preshared keys must be configured at both peers—otherwise the policy cannot be used (the policy will not be submitted for matching by the IKE process). The **crypto isakmp key** command is the second task required to configure the preshared keys at the peers. (The first task is accomplished using the **crypto isakmp identity** command.)

Use the **address** keyword if the remote peer ISAKMP identity was set with its IP address.

With the **address** keyword, you can also use the *mask* argument to indicate the remote peer ISAKMP identity will be established using the preshared key only. If the *mask* argument is used, preshared keys are no longer restricted between two users.

Note If you specify *mask*, you must use a subnet address. (The subnet address 0.0.0.0 is not recommended because it encourages group preshared keys, which allow all peers to have the same group key, thereby reducing the security of your user authentication.)

When using IKE main mode, preshared keys are indexed by IP address only because the identity payload has not yet been received. This means that the hostname keyword in the identity statement is not used to look up a preshared key and will be used only when sending and processing the identity payloads later in the main mode exchange. The identity keyword can be used when preshared keys are used with IKE aggressive mode, and keys may be indexed by identity types other than IP address as the identity payload is received in the first IKE aggressive mode packet.

If **crypto isakmp identity hostname** is configured as identity, the preshared key *must* be configured with the peer's IP address for the process to work when using IKE in main mode.

Use the **no-xauth** keyword to prevent the router from prompting the peer for Xauth information (username and password). This keyword disables Xauth for static IPSec peers. The **no-xauth** keyword should be enabled when configuring the preshared key for router-to-router IPSec—not VPN-client-to-Cisco-IOS IPSec.

Output for the **crypto isakmp key** command will show that the preshared key is either encrypted or unencrypted. An output example for an unencrypted preshared key would be as follows:

```
crypto isakmp key test123 address 10.1.0.1
```

An output example for a type 6 encrypted preshared key would be as follows:

```
crypto isakmp key 6 RHZE[JACMUI\bcbTdELISAAB address 10.1.0.1
```

Examples

In the following example, the remote peer "RemoteRouter" specifies an ISAKMP identity by address:

```
crypto isakmp identity address
```

Now, the preshared key must be specified at each peer.

In the following example, the local peer specifies the preshared key and designates the remote peer by its IP address and a mask:

```
crypto isakmp key 0 sharedkeystring address 172.21.230.33  255.255.255.255
```

In the following example for IPv6, the peer specifies the preshared key and designates the remote peer with an IPv6 address:

```
crypto isakmp key 0 my-preshare-key-0 address ipv6 3ffe:1001::2/128
```

Related Commandsl

Command	Description
crypto ipsec security-association lifetime	Specifies the authentication method within an IKE policy.
crypto isakmp identity	Defines the identity the router uses when participating in the IKE protocol.
ip host	Defines a static host name-to-address mapping in the host cache.

crypto isakmp peer

To enable an IP Security (IPSec) peer for Internet Key Exchange (IKE) querying of authentication, authorization, and accounting (AAA) for tunnel attributes in aggressive mode, use the **crypto isakmp peer** command in global configuration mode. To disable this functionality, use the **no** form of this command.

crypto isakmp peer {**address** {*ipv4-address* | **ipv6** *ipv6-address*} | **hostname** *fqdn-hostname*}

no crypto isakmp peer {**address** {*ipv4-address* | **ipv6** *ipv6-address*} | **hostname** *fqdn-hostname*}

Syntax Description

address *ip-address*	Address of the peer router.
ipv4-address	IPv4 address of the peer router.
ipv6 *ipv6-address*	IPv6 address of the peer router.
hostname	Hostname of the peer router.
fqdn-hostname	Fully qualified domain name (FQDN) of the peer router.

Command Default

None

Command Modes

Global configuration

Command History

Release	Modification
12.2(8)T	This command was introduced.
12.2(15)T	The **vrf** keyword and *fvrf-name* argument were added.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.4(4)T	The **ipv6** keyword and *ipv6-address* argument were added.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

Usage Guidelines

After enabling this command, you can use the **set aggressive-mode client-endpoint** and **set aggressive-mode password** commands to specify RADIUS tunnel attributes in the Internet Security Association and Key Management Protocol (ISAKMP) peer policy for IPSec peers.

Instead of keeping your preshared keys on the hub router, you can scale your preshared keys by storing and retrieving them from an AAA server. The preshared keys are stored in the AAA server as Internet Engineering Task Force (IETF) RADIUS tunnel attributes and are retrieved when a user tries to "speak" to the hub router. The hub router retrieves the preshared key from the AAA server and the spokes (the users) initiate aggressive mode to the hub by using the preshared key that is specified in the ISAKMP peer policy as a RADIUS tunnel attribute.

Examples

The following example shows how to initiate aggressive mode using RADIUS tunnel attributes:

```
crypto isakmp peer ip-address 209.165.200.230 vrf vpn1
 set aggressive-mode client-endpoint user-fqdn user@cisco.com
 set aggressive-mode password cisco123
```

Related Commands

Command	Description
crypto map isakmp authorization list	Enables IKE querying of AAA for tunnel attributes in aggressive mode.
set aggressive-mode client-endpoint	Specifies the Tunnel-Client-Endpoint attribute within an ISAKMP peer configuration.
set aggressive-mode password	Specifies the Tunnel-Password attribute within an ISAKMP peer configuration.

crypto isakmp policy

To define an Internet Key Exchange (IKE) policy, use the **crypto isakmp policy** command in global configuration mode. To delete an IKE policy, use the **no** form of this command.

crypto isakmp policy *priority*

no crypto isakmp policy *priority*

Syntax Description

priority	Uniquely identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10,000, with 1 being the highest priority and 10,000 the lowest.

Command Default

Default IKE policies are in use.

Command Modes

Global configuration (config)

Command History

Release	Modification
11.3T	This command was introduced.
12.4(4)T	Support for IPv6 was added.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(20)T	The command default was modified. Support for eight default IKE (ISAKMP) policies was added.
Cisco IOS XE Release 2.4	This command was implemented on the Cisco ASR 1000 series routers.

Usage Guidelines

IKE policies define a set of parameters to be used during the IKE negotiation. Use this command to specify the parameters to be used during an IKE negotiation. (These parameters are used to create the IKE security association [SA].)

This command invokes the Internet Security Association Key Management Protocol (ISAKMP) policy configuration (config-isakmp) command mode. While in the ISAKMP policy configuration command mode, some of the commands for which you can specify parameters are as follows:

- **authentication**; default = RSA signatures
- **encryption (IKE policy)**; default = 56-bit DES-CBC
- **group (IKE policy)**; default = 768-bit Diffie-Hellman
- **hash (IKE policy)**; default = SHA-1
- **lifetime (IKE policy)**; default = 86,400 seconds (one day)

If you do not specify any given parameter, the default value will be used for that parameter.

To exit the config-isakmp command mode, type **exit**.

You can configure multiple IKE policies on each peer participating in IPsec. When the IKE negotiation begins, it tries to find a common policy configured on both peers, starting with the highest priority policies as specified on the remote peer.

Examples

The following example shows how to manually configure two policies for the peer:

```
crypto isakmp policy 15
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
```

The above configuration results in the following policies:

```
Router# show crypto isakmp policy

Protection suite priority 15
    encryption algorithm:DES - Data Encryption Standard (56 bit keys)
    hash algorithm:Message Digest 5
    authentication method:Rivest-Shamir-Adleman Signature
    Diffie-Hellman Group:#2 (1024 bit)
    lifetime:5000 seconds, no volume limit
Protection suite priority 20
    encryption algorithm:DES - Data Encryption Standard (56 bit keys)
    hash algorithm:Secure Hash Standard
    authentication method:preshared Key
    Diffie-Hellman Group:#1 (768 bit)
    lifetime:10000 seconds, no volume limit
Default protection suite
    encryption algorithm:DES - Data Encryption Standard (56 bit keys)
    hash algorithm:Secure Hash Standard
    authentication method:Rivest-Shamir-Adleman Signature
    Diffie-Hellman Group:#1 (768 bit)
    lifetime:86400 seconds, no volume limit
```

The following sample output from the **show crypto isakmp policy** command displays the default IKE policies when the manually configured IKE policies with priorities 15 and 20 have been removed.

```
Router(config)# no crypto isakmp policy 15
Router(config)# no crypto isakmp policy 20
Router(config)# exit
R1# show crypto isakmp policy

Default IKE policy
Protection suite of priority 65507
        encryption algorithm:   AES - Advanced Encryption Standard (128 bit key.
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
Protection suite of priority 65508
        encryption algorithm:   AES - Advanced Encryption Standard (128 bit key.
        hash algorithm:         Secure Hash Standard
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
Protection suite of priority 65509
        encryption algorithm:   AES - Advanced Encryption Standard (128 bit key.
        hash algorithm:         Message Digest 5
```

```
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
Protection suite of priority 65510
        encryption algorithm:   AES - Advanced Encryption Standard (128 bit key.
        hash algorithm:         Message Digest 5
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
Protection suite of priority 65511
        encryption algorithm:   Three key triple DES
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               86400 seconds, no volume limit
Protection suite of priority 65512
        encryption algorithm:   Three key triple DES
        hash algorithm:         Secure Hash Standard
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               86400 seconds, no volume limit
Protection suite of priority 65513
        encryption algorithm:   Three key triple DES
        hash algorithm:         Message Digest 5
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               86400 seconds, no volume limit
Protection suite of priority 65514
        encryption algorithm:   Three key triple DES
        hash algorithm:         Message Digest 5
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               86400 seconds, no volume limit
```

Related Commands

Command	Description
encryption (IKE policy)	Specifies the encryption algorithm within an IKE policy.
group (IKE policy)	Specifies the Diffie-Hellman group identifier within an IKE policy.
hash (IKE policy)	Specifies the hash algorithm within an IKE policy.
lifetime (IKE policy)	Specifies the lifetime of an IKE SA.
show crypto isakmp default policy	Displays the default IKE (ISAKMP) policies currently in use.
show crypto isakmp policy	Displays the parameters for each IKE policy.

crypto isakmp profile

To define an Internet Security Association and Key Management Protocol (ISAKMP) profile and to audit IP security (IPsec) user sessions, use the **crypto isakmp profile** command in global configuration mode. To delete a crypto ISAKMP profile, use the **no** form of this command.

crypto isakmp profile *profile-name* [**accounting** *aaa-list*]

no crypto isakmp profile *profile-name* [**accounting** *aaa-list*]

Syntax Description

profile-name	Name of the user profile. To associate a user profile with the RADIUS server, the user profile name must be identified.
accounting *aaa-list*	(Optional) Name of a client accounting list.

Command Defaults

No profile exists if the command is not used.

Command Modes

Global configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.4(2)T	Support for dynamic virtual tunnel interfaces was added.
12.4(4)T	Support for IPv6 was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

Usage Guidelines

Defining an ISAKMP Profile

An ISAKMP profile can be viewed as a repository of Phase 1 and Phase 1.5 commands for a set of peers. The Phase 1 configuration includes commands to configure such things as keepalive, identity matching, and the authorization list. The Phase 1.5 configuration includes commands to configure such things as extended authentication (Xauth) and mode configuration.

The peers are mapped to an ISAKMP profile when their identities are matched (as given in the identification [ID] payload of the Internet Key Exchange [IKE]) against the identities defined in the ISAKMP profile. To uniquely map to an ISAKMP profile, no two ISAKMP profiles should match the same identity. If the peer identity is matched in two ISAKMP profiles, the configuration is invalid. Also, there must be at least one **match identity** command defined in the ISAKMP profile for it to be complete.

After enabling this command and entering ISAKMP profile configuration mode, you can configure the following commands:

- **accounting**—Enables authentication, authorization, and accounting (AAA) accounting.
- **ca trust-point**—Specifies certificate authorities.
- **client**—Specifies client configuration settings.

- **default**—Lists subcommands for the **crypto isakmp profile** command.
- **description**—Specifies a description of this profile.
- **initiate mode**—Initiates a mode.
- **isakmp authorization**—ISAKMP authorization parameters.
- **keepalive**—Sets a keepalive interval.
- **keyring**—Specifies a keyring.
- **local-address**—Specifies the interface to use as the local address of this ISAKMP profile.
- **match**—Matches the values of the peer.
- **qos-group**—Applies a quality of service (QoS) policy class map for this profile.
- **self-identity**—Specifies the identity.
- **virtual-template**—Specifies the virtual template for the dynamic interface.
- **vrf**—Specifies the Virtual Private Network routing and forwarding (VRF) instance to which the profile is related.

Auditing IPSec User Sessions

Use this command to audit multiple user sessions that are terminating on the IPSec gateway.

Note The **crypto isakmp profile** command and the **crypto map (global IPSec)** command are mutually exclusive. If a profile is present (the **crypto isakmp profile** command has been used), with no accounting configured but with the global command present (the **crypto isakmp profile** command without the **accounting** keyword), accounting will occur using the attributes in the global command.

Dynamic Virtual Tunnel Interfaces

Support for dynamic virtual tunnel interfaces allows for the virtual profile to be mapped into a specified virtual template.

Examples

ISAKAMP Profile Matching Peer Identities Example

The following example shows how to define an ISAKMP profile and match the peer identities:

```
crypto isakmp profile vpnprofile
 match identity address 10.76.11.53
```

ISAKAMP Profile with Accounting Example

The following accounting example shows that an ISAKMP profile is configured:

```
aaa new-model
!
!
aaa authentication login cisco-client group radius
aaa authorization network cisco-client group radius
aaa accounting network acc start-stop broadcast group radius
aaa session-id common
!
crypto isakmp profile cisco
vrf cisco
match identity group cclient
 client authentication list cisco-client
```

```
 isakmp authorization list cisco-client
 client configuration address respond
 accounting acc
!
crypto dynamic-map dynamic 1
 set transform-set aswan
 set isakmp-profile cisco
 reverse-route
!
!
radius-server host 172.16.1.4 auth-port 1645 acct-port 1646
radius-server key nsite
```

Related Commands

Command	Description
crypto map (global IPsec)	Enters crypto map configuration mode and creates or modifies a crypto map entry, creates a crypto profile that provides a template for configuration of dynamically created crypto maps, or configures a client accounting list.
debug crypto isakmp	Displays messages about IKE events.
match identity	Matches an identity from a peer in an ISAKMP profile.
tunnel protection	Associates a tunnel interface with an IP Security (IPsec) profile.
virtual template	Specifies which virtual template to be used to clone virtual access interfaces.

crypto key generate rsa

To generate Rivest, Shamir, and Adelman (RSA) key pairs, use the **crypto key generate rsa** command in global configuration mode.

crypto key generate rsa [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename***:**][**redundancy**][**on** *devicename***:**]

Syntax Description

general-keys	(Optional) Specifies that a general-purpose key pair will be generated, which is the default.
usage-keys	(Optional) Specifies that two RSA special-usage key pairs, one encryption pair and one signature pair, will be generated.
signature	(Optional) Specifies that the RSA public key generated will be a signature special usage key.
encryption	(Optional) Specifies that the RSA public key generated will be an encryption special usage key.
label *key-label*	(Optional) Specifies the name that is used for an RSA key pair when they are being exported. If a key label is not specified, the fully qualified domain name (FQDN) of the router is used.
exportable	(Optional) Specifies that the RSA key pair can be exported to another Cisco device, such as a router.
modulus *modulus-size*	(Optional) Specifies the IP size of the key modulus. By default, the modulus of a certification authority (CA) key is 1024 bits. The recommended modulus for a CA key is 2048 bits. The range of a CA key modulus is from 350 to 4096 bits. **Note** Effective with Cisco IOS XE Release 2.4 and Cisco IOS Release 15.1(1)T, the maximum key size was expanded to 4096 bits for private key operations. The maximum for private key operations prior to these releases was 2048 bits.
storage *devicename***:**	(Optional) Specifies the key storage location. The name of the storage device is followed by a colon (:).
redundancy	(Optional) Specifies that the key should be synchronized to the standby CA.
on *devicename***:**	(Optional) Specifies that the RSA key pair will be created on the specified device, including a Universal Serial Bus (USB) token, local disk, or NVRAM. The name of the device is followed by a colon (:). Keys created on a USB token must be 2048 bits or less.

Command Default

RSA key pairs do not exist.

Command Modes

Global configuration

Command History

Release	Modification
11.3	This command was introduced.
12.2(8)T	The *key-label* argument was added.
12.2(15)T	The **exportable** keyword was added.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.4(4)T	The **storage** keyword and *devicename***:** argument were added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(11)T	The **storage** keyword and *devicename***:** argument were implemented on the Cisco 7200VXR NPE-G2 platform. The **signature**, **encryption** and **on** keywords and *devicename***:** argument were added.
12.4(24)T	Support for IPv6 Secure Neighbor Discovery (SeND) was added.
XE 2.4	The maximum RSA key size was expanded from 2048 to 4096 bits for private key operations.
15.0(1)M	This command was modified. The **redundancy** keyword was introduced.
15.1(1)T	This command was modified. The range value for the **modulus** keyword value is extended from 360 to 2048 bits to 360 to 4096 bits.

Usage Guidelines

Use this command to generate RSA key pairs for your Cisco device (such as a router).

RSA keys are generated in pairs—one public RSA key and one private RSA key.

If your router already has RSA keys when you issue this command, you will be warned and prompted to replace the existing keys with new keys.

Note Before issuing this command, ensure that your router has a hostname and IP domain name configured (with the **hostname** and **ip domain-name** commands). You will be unable to complete the **crypto key generate rsa** command without a hostname and IP domain name. (This situation is not true when you generate only a named key pair.)

Note Secure Shell (SSH) may generate an additional RSA key pair if you generate a key pair on a router having no RSA keys. The additional key pair is used only by SSH and will have a name such as {*router_FQDN*}.server. For example, if a router name is "router1.cisco.com," the key name is "router1.cisco.com.server."

This command is not saved in the router configuration; however, the RSA keys generated by this command are saved in the private configuration in NVRAM (which is never displayed to the user or backed up to another device) the next time the configuration is written to NVRAM.

Note If the configuration is not saved to NVRAM, the generated keys are lost on the next reload of the router.

There are two mutually exclusive types of RSA key pairs: special-usage keys and general-purpose keys. When you generate RSA key pairs, you will be prompted to select either special-usage keys or general-purpose keys.

Special-Usage Keys

If you generate special-usage keys, two pairs of RSA keys will be generated. One pair will be used with any Internet Key Exchange (IKE) policy that specifies RSA signatures as the authentication method, and the other pair will be used with any IKE policy that specifies RSA encrypted keys as the authentication method.

A CA is used only with IKE policies specifying RSA signatures, not with IKE policies specifying RSA-encrypted nonces. (However, you could specify more than one IKE policy and have RSA signatures specified in one policy and RSA-encrypted nonces in another policy.)

If you plan to have both types of RSA authentication methods in your IKE policies, you may prefer to generate special-usage keys. With special-usage keys, each key is not unnecessarily exposed. (Without special-usage keys, one key is used for both authentication methods, increasing the exposure of that key.)

General-Purpose Keys

If you generate general-purpose keys, only one pair of RSA keys will be generated. This pair will be used with IKE policies specifying either RSA signatures or RSA encrypted keys. Therefore, a general-purpose key pair might get used more frequently than a special-usage key pair.

Named Key Pairs

If you generate a named key pair using the *key-label* argument, you must also specify the **usage-keys** keyword or the **general-keys** keyword. Named key pairs allow you to have multiple RSA key pairs, enabling the Cisco IOS software to maintain a different key pair for each identity certificate.

Modulus Length

When you generate RSA keys, you will be prompted to enter a modulus length. The longer the modulus, the stronger the security. However a longer modules takes longer to generate (see Table 9 for sample times) and takes longer to use.

Table 9 Sample Times by Modulus Length to Generate RSA Keys

Router	360 bits	512 bits	1024 bits	2048 bits (maximum)
Cisco 2500	11 seconds	20 seconds	4 minutes, 38 seconds	More than 1 hour
Cisco 4700	Less than 1 second	1 second	4 seconds	50 seconds

Cisco IOS software does not support a modulus greater than 4096 bits. A length of less than 512 bits is normally not recommended. In certain situations, the shorter modulus may not function properly with IKE, so we recommend using a minimum modulus of 2048 bits.

Note As of Cisco IOS Release 12.4(11)T, peer *public* RSA key modulus values up to 4096 bits are automatically supported.

The largest private RSA key modulus is 4096 bits. Therefore, the largest RSA private key a router may generate or import is 4096 bits. However, RFC 2409 restricts the private key size to 2048 bits or less for RSA encryption.

The recommended modulus for a CA is 2048 bits; the recommended modulus for a client is 2048 bits.

Additional limitations may apply when RSA keys are generated by cryptographic hardware. For example, when RSA keys are generated by the Cisco VPN Services Port Adapter (VSPA), the RSA key modulus must be a minimum of 384 bits and must be a multiple of 64.

Specifying a Storage Location for RSA Keys

When you issue the **crypto key generate rsa** command with the **storage** *devicename***:** keyword and argument, the RSA keys will be stored on the specified device. This location will supersede any **crypto key storage** command settings.

Specifying a Device for RSA Key Generation

As of Cisco IOS Release 12.4(11)T and later releases, you may specify the device where RSA keys are generated. Devices supported include NVRAM, local disks, and USB tokens. If your router has a USB token configured and available, the USB token can be used as cryptographic device in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication of credentials to be performed on the token. The private key never leaves the USB token and is not exportable. The public key is exportable.

RSA keys may be generated on a configured and available USB token, by the use of the **on** *devicename***:** keyword and argument. Keys that reside on a USB token are saved to persistent token storage when they are generated. The number of keys that can be generated on a USB token is limited by the space available. If you attempt to generate keys on a USB token and it is full you will receive the following message:

```
% Error in generating keys:no available resources
```

Key deletion will remove the keys stored on the token from persistent storage immediately. (Keys that do not reside on a token are saved to or deleted from nontoken storage locations when the **copy** or similar command is issued.)

For information on configuring a USB token, see "Storing PKI Credentials" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.4T. For information on using on-token RSA credentials, see the "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.4T.

Specifying RSA Key Redundancy Generation on a Device

You can specify redundancy for existing keys only if they are exportable.

Examples

The following example generates a general-usage 1024-bit RSA key pair on a USB token with the label "ms2" with crypto engine debugging messages shown:

```
Router(config)# crypto key generate rsa label ms2 modulus 2048 on usbtoken0:

The name for the keys will be: ms2
% The key modulus size is 2048 bits
% Generating 1024 bit RSA keys, keys will be on-token, non-exportable...
Jan 7 02:41:40.895: crypto_engine: Generate public/private keypair [OK]
Jan 7 02:44:09.623: crypto_engine: Create signature
Jan 7 02:44:10.467: crypto_engine: Verify signature
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_ISA_RSA_CREATE_PUBKEY(hw)(ipsec)
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_ISA_RSA_PUB_DECRYPT(hw)(ipsec)
```

Now, the on-token keys labeled "ms2" may be used for enrollment.

The following example generates special-usage RSA keys:

```
Router(config)# crypto key generate rsa usage-keys

The name for the keys will be: myrouter.example.com

Choose the size of the key modulus in the range of 360 to 2048 for your Signature Keys.
Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus[512]? <return>
Generating RSA keys.... [OK].

Choose the size of the key modulus in the range of 360 to 2048 for your Encryption Keys.
Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus[512]? <return>
Generating RSA keys.... [OK].
```

The following example generates general-purpose RSA keys:

Note You cannot generate both special-usage and general-purpose keys; you can generate only one or the other.

```
Router(config)# crypto key generate rsa general-keys

The name for the keys will be: myrouter.example.com

Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
Keys. Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus[512]? <return>
Generating RSA keys.... [OK].
```

The following example generates the general-purpose RSA key pair "exampleCAkeys":

```
crypto key generate rsa general-keys label exampleCAkeys
crypto ca trustpoint exampleCAkeys
 enroll url http://exampleCAkeys/certsrv/mscep/mscep.dll
 rsakeypair exampleCAkeys 1024 1024
```

The following example specifies the RSA key storage location of "usbtoken0:" for "tokenkey1":

```
crypto key generate rsa general-keys label tokenkey1 storage usbtoken0:
```

The following example specifies the **redundancy** keyword:

```
Router(config)# crypto key generate rsa label MYKEYS redundancy

The name for the keys will be: MYKEYS
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.


How many bits in the modulus [512]:
% Generating 512 bit RSA keys, keys will be non-exportable with redundancy...[OK]
```

Related Commands

Command	Description
copy	Copies any file from a source to a destination, use the copy command in privileged EXEC mode.
crypto key storage	Sets the default storage location for RSA key pairs.
debug crypto engine	Displays debug messages about crypto engines.
hostname	Specifies or modifies the hostname for the network server.
ip domain-name	Defines a default domain name to complete unqualified hostnames (names without a dotted-decimal domain name).
show crypto key mypubkey rsa	Displays the RSA public keys of your router.
show crypto pki certificates	Displays information about your PKI certificate, certification authority, and any registration authority certificates.

crypto keyring

To define a crypto keyring to be used during Internet Key Exchange (IKE) authentication, use the **crypto keyring** command in global configuration mode. To remove the keyring, use the **no** form of this command.

crypto keyring *keyring-name* [**vrf** *fvrf-name*]

no crypto keyring *keyring-name* [**vrf** *fvrf-name*]

Syntax Description

keyring-name	Name of the crypto keyring.
vrf *fvrf-name*	(Optional) Front door virtual routing and forwarding (FVRF) name to which the keyring will be referenced. The *fvrf-name* must match the FVRF name that was defined during virtual routing and forwarding (VRF) configuration. The **vrf** keyword and *fvrf-name* argument are not supported by IPv6.

Command Default

All the Internet Security Association and Key Management Protocol (ISAKMP) keys that were defined in the global configuration are part of the default global keyring.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.4(4)T	Support for IPv6 was added.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

Usage Guidelines

A keyring is a repository of preshared and Rivest, Shamir, and Adelman (RSA) public keys. The keyring is used in the ISAKMP profile configuration mode. The ISAKMP profile successfully completes authentication of peers if the peer keys are defined in the keyring that is attached to this profile.

Examples

The following example shows that a keyring and its usage have been defined:

```
crypto keyring vpnkeys
  pre-shared-key address 10.72.23.11 key vpnsecret
crypto isakmp profile vpnprofile
  keyring vpnkeys
```

Related Commands

Command	Description
pre-shared-key	Defines a preshared key to be used for IKE authentication.

crypto map (global IPsec)

To enter crypto map configuration mode and create or modify a crypto map entry, to create a crypto profile that provides a template for configuration of dynamically created crypto maps, or to configure a client accounting list, use the **crypto map** command in global configuration mode. To delete a crypto map entry, profile, or set, use the **no** form of this command.

crypto map [**ipv6**] *map-name seq-num* [**ipsec-manual**]

crypto map [**ipv6**] *map-name seq-num* [**ipsec-isakmp** [**dynamic** *dynamic-map-name* | **discover** | **profile** *profile-name*]]

no crypto map [**ipv6**] *map-name* [*seq-num*]

crypto map [**ipv6**] *map-name* **client accounting list** *aaalist*

no crypto map [**ipv6**] *map-name* [**client accounting list**]

crypto map *map-name seq num* [**gdoi**]

no crypto map *map-name* [*seq-num*]

Syntax Description

ipv6	(Optional) Specifies an IPv6 crypto map. For IPv4 crypto maps, use the command without this keyword. **Note** IPv6 addresses are not supported on dynamic crypto maps.
map-name	Identifies the crypto map set.
seq-num	Sequence number you assign to the crypto map entry. See additional explanation for using this argument in the "Usage Guidelines" section.
ipsec-manual	(Optional) Indicates that Internet Key Exchange (IKE) will not be used to establish the IP Security (IPsec) security associations (SAs) for protecting the traffic specified by this crypto map entry. **Note** The **ipsec-manual** keyword is not supported by the virtual private network Shared Port Adapter (VPN SPA) beginning with Cisco IOS Release 12.2(33)SXH5 or 12.2(33)SXI1. If the **ipsec-manual** keyword is entered for images after those releases, the following error message appears beneath the keyword entry line: "Manually-keyed crypto map configuration is not supported by the current crypto engine."
ipsec-isakmp	(Optional) Indicates that IKE will be used to establish the IPsec for protecting the traffic specified by this crypto map entry.
dynamic	(Optional) Specifies that this crypto map entry must reference a preexisting dynamic crypto map. **Note** Dynamic crypto maps are policy templates used in processing negotiation requests from a peer IPsec device. If you use this keyword, none of the crypto map configuration commands will be available.
dynamic-map-name	(Optional) Name of the dynamic crypto map set that should be used as the policy template.
discover	(Optional) Enables peer discovery. By default, peer discovery is disabled.

profile	(Optional) Designates a crypto map as a configuration template. The security configurations of this crypto map will be cloned as new crypto maps are created dynamically on demand.
profile-name	(Optional) Name of the crypto profile being created.
client accounting list	Designates a client accounting list.
aaalist	(Optional) AAA list name.
gdoi	(Optional) Indicates that the key management mechanism is Group Domain of Interpretation (GDOI).

Command Default

No crypto maps exist.
Peer discovery is disabled.

Command Modes

Global configuration (config)

Command History

Release	Modification
11.2	This command was introduced.
11.3T	The following keywords and arguments were added: • **ipsec-manual** • **ipsec-isakmp** • **dynamic** • *dynamic-map-name*
12.0(5)T	The **discover** keyword was added to support Tunnel Endpoint Discovery (TED).
12.2(4)T	The **profile** *profile-name* keyword-argument pair was added to allow the generation of a crypto map profile that is cloned to create dynamically created crypto maps on demand.
12.2(11)T	This command was implemented on the Cisco 1760, Cisco AS5300, Cisco AS5400, and Cisco AS5800 platforms.
12.2(15)T	The **client accounting list** *aaalist* keyword-argument pair was added.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB without support for the **gdoi** keyword.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH5, 12.2(33)SXI1	The **ipsec-manual** keyword is not supported by the VPN SPA beginning with Cisco IOS Release 12.2(33)SXH5 or 12.2(33)SXI1.
12.4(6)T	The **gdoi** keyword was added.
Cisco IOS XE 2.1	This command was implemented on Cisco ASR 1000 series routers.
15.1(4) M	This command was modified. The **ipv6** keyword was added.

Usage Guidelines

Use this command to create a new crypto map entry or profile. Use the **crypto map ipv6** *map-name seq-num* command without any keyword to modify an existing IPv6 crypto map entry or profile. For IPv4 crypto maps, use the **crypto map** *map-name seq-num* command without any keyword to modify the existing crypto map entry or profile.

After a crypto map entry is created, you cannot change the parameters specified at the global configuration level because these parameters determine the configuration commands that are valid at the crypto map level. For example, after a map entry has been created using the **ipsec-isakmp** keyword, you cannot change it to the option specified by the **ipsec-manual** keyword; you must delete and reenter the map entry.

After you define crypto map entries, you can assign the crypto map set to interfaces using the **crypto map** (interface IPsec) command.

Crypto Map Functions

Crypto maps provide two functions: filtering and classifying the traffic to be protected and defining the policy to be applied to that traffic. The first affects the flow of traffic on an interface; the second affects the negotiation performed (via IKE) on behalf of that traffic.

IPsec crypto maps define the following:

- What traffic should be protected
- To which IPsec peers the protected traffic can be forwarded—these are the peers with which an SA can be established
- Which transform sets are acceptable for use with the protected traffic
- How keys and SAs should be used or managed (or what the keys are, if IKE is not used)

Multiple Crypto Map Entries with the Same Map Name Form a Crypto Map Set

A crypto map set is a collection of crypto map entries, each with a different *seq-num* argument but the same *map-name* argument. Therefore, for an interface, you could have certain traffic forwarded to one IPsec peer with specified security applied to that traffic and other traffic forwarded to the same or different IPsec peer with different IPsec security applied. To accomplish differential forwarding, you would create two crypto maps, each with the same *map-name* argument but different *seq-num* argument. Crypto profiles must have unique names within a crypto map set.

Note If a deny statement (which specifies the conditions under which a packet cannot pass the access control list) in an access control list belongs to a crypto map in a crypto map set, the IPsec logic causes a jump to the next crypto map in the crypto map set, hoping for a better possible match. VPN Service Adapter (VSA) hardware has a restriction of 14 jumps.

Sequence Numbers

The number you assign to the *seq-num* argument should not be arbitrary. This number is used to rank multiple crypto map entries within a crypto map set. Within a crypto map set, a crypto map entry with a lower *seq-num* is evaluated before a map entry with a higher *seq-num*; that is, the map entry with the lower number has a higher priority.

For example, assume that a crypto map set contains three crypto map entries: mymap 10, mymap 20, and mymap 30. The crypto map set named “mymap” is applied to serial interface 0. When traffic passes through serial interface 0, traffic is evaluated first for mymap 10. If the traffic matches any access list permit statement entry in the extended access list in mymap 10, the traffic will be processed according to the information defined in mymap 10 (which includes establishing IPsec SAs when necessary). If the

traffic does not match the mymap 10 access list, the traffic will be evaluated for mymap 20, and then mymap 30, until the traffic matches a permit entry in a map entry. (If the traffic does not match a permit entry in any crypto map entry, it will be forwarded without any IPsec security.)

Dynamic Crypto Maps

Refer to the "Usage Guidelines" section of the **crypto dynamic-map** command for a discussion on dynamic crypto maps.

Crypto map entries that reference dynamic map sets should be the lowest priority map entries, allowing inbound SA negotiation requests to try to match the static maps first. If the request does not match any of the static maps, it will be evaluated against the dynamic map set.

If a crypto map entry references a dynamic crypto map set, make it the lowest priority map entry by giving it the highest *seq-num* value of all the map entries in a crypto map set.

Create dynamic crypto map entries using the **crypto dynamic-map** command. After you create a dynamic crypto map set, add the dynamic crypto map set to a static crypto map set with the **crypto map** (global IPsec) command using the **dynamic** keyword.

Note IPv6 keywords are not supported on dynamic crypto maps.

TED

Tunnel Endpoint Discovery (TED) is an enhancement to the IPsec feature. Defining a dynamic crypto map allows you to dynamically determine an IPsec peer; however, only the receiving router has this ability. With TED, the initiating router can dynamically determine an IPsec peer for secure IPsec communications.

Dynamic TED helps to simplify the IPsec configuration on individual routers within a large network. Each node has a simple configuration that defines the local network that the router is protecting and the IPsec transforms that are required.

Note TED helps only in discovering peers; otherwise, TED does not function any differently from normal IPsec. Thus, TED does not improve the scalability of IPsec (in terms of performance or the number of peers or tunnels).

Crypto Map Profiles

Crypto map profiles are created using the **profile** *profile-name* keyword and argument combination. Crypto map profiles are used as configuration templates for dynamically creating crypto maps on demand for use with the L2TP Security feature. The relevant SAs in the crypto map profile will be cloned and used to protect IP traffic on the L2TP tunnel.

Note The **set peer** and **match address** commands are ignored by crypto profiles and should not be configured in the crypto map definition.

Examples

The following example shows the minimum required crypto map configuration when IKE will be used to establish the SAs:

```
crypto map mymap 10 ipsec-isakmp
 match address 101
 set transform-set my_t_set1
```

```
 set peer 10.0.0.1
```

The following example shows the minimum required IPv6 crypto map configuration when IKE will be used to establish the SAs:

```
crypto map ipv6 CM_V6 10 ipsec-isakmp
 match address ACL_IPV6_1
  set peer 2001:DB8:0:ABCD::1
```

The following example shows the minimum required crypto map configuration when the SAs are manually established:

```
crypto transform-set someset ah-md5-hmac esp-des
crypto map mymap 10 ipsec-manual
 match address 102
 set transform-set someset
 set peer 10.0.0.5
 set session-key inbound ah 256 9876543210987654987654321098765 4
 set session-key outbound ah 256 fedcbafedcbafedcfedcbafedcbafedc
 set session-key inbound esp 256 cipher 0123456789012345
 set session-key outbound esp 256 cipher abcdefabcdefabcd
```

The following example shows the minimum required IPv6 crypto map configuration when the SAs are manually established:

```
crypto map ipv6 CM_V6 ipsec-manual
 match address ACL_V6_2
 set transform-set someset
 set peer 2001:DB8:0:ABCD::1
 set session-key inbound ah 256 9876543210987654987654321098765 4
 set session-key outbound ah 256 fedcbafedcbafedcfedcbafedcbafedc
 set session-key inbound esp 256 cipher 0123456789012345
 set session-key outbound esp 256 cipher abcdefabcdefabcd
```

The following example shows how to configure an IPsec crypto map set that includes a reference to a dynamic crypto map set.

Crypto map “mymap 10” allows SAs to be established between the router and either or both the remote IPsec peers for traffic matching access list 101. Crypto map “mymap 20” allows either of the two transform sets to be negotiated with the remote peer for traffic matching access list 102.

Crypto map entry “mymap 30” references the dynamic crypto map set “mydynamicmap,” which can be used to process inbound SA negotiation requests that do not match “mymap” entries 10 or 20. In this case, if the peer specifies a transform set that matches one of the transform sets specified in “mydynamicmap,” for a flow permitted by the access list 103, IPsec will accept the request and set up SAs with the remote peer without previously knowing about the remote peer. If the request is accepted, the resulting SAs (and temporary crypto map entry) are established according to the settings specified by the remote peer.

The access list associated with “mydynamicmap 10” is also used as a filter. Inbound packets that match any access list permit statement in this list are dropped for not being IPsec protected. (The same is true for access lists associated with static crypto maps entries.) Outbound packets that match a permit statement without an existing corresponding IPsec SA are also dropped.

```
crypto map mymap 10 ipsec-isakmp
 match address 101
 set transform-set my_t_set1
 set peer 10.0.0.1
 set peer 10.0.0.2
crypto map mymap 20 ipsec-isakmp
 match address 102
 set transform-set my_t_set1 my_t_set2
```

```
 set peer 10.0.0.3
crypto map mymap 30 ipsec-isakmp dynamic mydynamicmap
!
crypto dynamic-map mydynamicmap 10
 match address 103
 set transform-set my_t_set1 my_t_set2 my_t_set3
```

The following example shows how to configure TED on a Cisco router:

```
crypto map testtag 10 ipsec-isakmp dynamic dmap discover
```

The following example shows how to configure a crypto profile to be used as a template for dynamically created crypto maps when IPsec is used to protect an L2TP tunnel:

```
crypto map l2tpsec 10 ipsec-isakmp profile l2tp
```

The following example shows how to configure a crypto map for a GDOI group member:

```
crypto map diffint 10 gdoi
 set group diffint
```

Related Commands

Command	Description
crypto dynamic-map	Creates a dynamic crypto map entry and enters crypto map configuration command mode.
crypto isakmp profile	Audits IPsec user sessions.
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.
crypto map local-address	Specifies and names an identifying interface to be used by the crypto map for IPsec traffic.
match address (IPSec)	Specifies an extended access list for a crypto map entry.
set peer (IPsec)	Specifies an IPsec peer in a crypto map entry.
set pfs	Specifies that IPsec should ask for PFS when requesting new SAs for this crypto map entry, or that IPsec requires PFS when receiving requests for new SAs.
set session-key	Specifies the IPsec session keys within a crypto map entry.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto map (IPsec)	Displays the crypto map configuration.

crypto map (isakmp)

To enable Internet Key Exchange (IKE) querying of authentication, authorization, and accounting (AAA) for tunnel attributes in aggressive mode, use the **crypto map** command in global configuration mode. To restore the default value, use the **no** form of this command.

crypto map [**ipv6**] *map-name* **isakmp authorization list** *list-name*

no crypto map [**ipv6**] *map-name* [**isakmp authorization list**]

Syntax Description

ipv6	(Optional) Specifies an IPv6 crypto map. For IPv4 crypto maps, use the command without this keyword.
map-name	Name you assign to the crypto map set.
isakmp authorization list	Specifies the Internet Security Association Key Management Protocol (ISAKMP) configuration settings and authorization parameters.
list-name	Character string used to name the list of authorization methods activated when a user logs in. The list name must match the list name defined during AAA configuration.

Defaults

No default behavior or values.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.1(4)M	This command was modified. The **ipv6** keyword was added.

Usage Guidelines

Use this command to enable key lookup from an AAA server.

Preshared keys deployed in a large-scale Virtual Private Network (VPN) without a certification authority, with dynamic IP addresses, are accessed during aggression mode of IKE negotiation through an AAA server. Thus, users have their own key, which is stored on an external AAA server. This allows for the central management of the user database, linking it to an existing database and allowing all users to have their own unique and secure pre-shared keys.

Before configuring this command, you should perform the following tasks:

- Set up an authorization list using AAA commands.
- Configure an IPsec transform.
- Configure a crypto map.

- Configure an ISAKMP policy using IPsec and IKE commands.

After enabling this command, you should apply the previously defined crypto map to the interface.

Examples

The following example shows how to configure the **crypto map** command for IPv4 crypto maps:

```
crypto map ikessaaamap isakmp authorization list ikessaaalist
crypto map ikessaaamap 10 ipsec-isakmp dynamic ikessaaadyn
```

The following example shows how to configure the **crypto map** command for IPv6 crypto maps:

```
crypto map ipv6 CM_V6 isakmp authorization list aaa
crypto map ipv6 CM_V6 10 ipsec-isakmp dynamic aaadyn
```

Related Commands

Command	Description
aaa authorization	Sets parameters that restrict a user's network access.
crypto ipsec transform-set	Defines a transform set, which is an acceptable combination of security protocols and algorithms, and enters crypto transform configuration mode.
crypto isakmp key	Configures a preshared authentication key.
crypto isakmp policy	Defines an IKE policy and enters ISAKMP policy configuration mode.
crypto map (global configuration)	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
interface	Enters interface configuration mode.

crypto map (Xauth)

To configure Internet Key Exchange (IKE) extended authentication (Xauth) on a router, use the **crypto map** command in global configuration mode. To restore the default value, use the **no** form of this command.

crypto map [**ipv6**] *map-name* **client authentication list** *list-name*

no crypto map [**ipv6**] *map-name* [**client authentication list**]

Syntax Description

ipv6	(Optional) Specifies an IPv6 crypto map. For IPv4 crypto maps, use the command without this keyword.
map-name	Name you assign to the crypto map set.
client authentication list	Designates an extended user authentication method.
list-name	Character string used to name the list of authentication methods activated when a user logs in. The list name must match the list name defined during AAA configuration.

Defaults

Xauth is disabled.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.1(4)M	This command was modified. The **ipv6** keyword was added.

Usage Guidelines

Before configuring Xauth, you should complete the following tasks:

- Set up an authentication list using AAA commands.
- Configure an IP Security transform.
- Configure a crypto map.
- Configure Internet Security Association Key Management Protocol (ISAKMP) policy.

After enabling Xauth, you should apply the crypto map on which Xauth is configured to the router interface.

Examples

The following example shows how to configure user authentication (a list of authentication methods called *xauthlist*) on an existing static crypto map called *xauthmap*:

```
crypto map xauthmap client authentication list xauthlist
```

The following example shows how to configure user authentication (a list of authentication methods called *CM_V6list)* on an existing static IPv6 crypto map called *CM_V6*:

```
crypto map ipv6 CM_V6 client authentication list CM_V6list
```

The following example shows how to configure user authentication (a list of authentication methods called *xauthlist*) on a dynamic crypto map called *xauthdynamic* that has been applied to a static crypto map called *xauthmap*:

```
crypto map xauthmap client authentication list xauthlist
crypto map xauthmap 10 ipsec-isakmp dynamic xauthdynamic
```

Related Commands

Command	Description
aaa authentication login	Sets AAA authentication at login.
crypto ipsec transform-set	Defines a transform set, which is an acceptable combination of security protocols and algorithms, and enters crypto transform configuration mode.
crypto isakmp key	Configures a preshared authentication key.
crypto isakmp policy	Defines an IKE policy, and enters ISAKMP policy configuration mode.
crypto map (global configuration)	Creates or modifies a crypto map entry, and enters the crypto map configuration mode.
interface	Enters the interface configuration mode.

crypto pki authenticate

To authenticate the certification authority (CA) (by getting the certificate of the CA), use the **crypto pki authenticate** command in global configuration mode.

crypto pki authenticate *name*

Syntax Description

name	The name of the CA. This is the same name used when the CA was declared with the **crypto ca identity** command.

Defaults

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modification
11.3T	The **crypto ca authenticate** command was introduced.
12.3(7)T	This command replaced the **crypto ca authenticate** command.
12.2(18)SXE	This command was integrated into Cisco IOS Release 12.2(18)SXE.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(24)T	Support for IPv6 Secure Neighbor Discovery (SeND) was added.

Usage Guidelines

This command is required when you initially configure CA support at your router.

This command authenticates the CA to your router by obtaining the self-signed certificate of the CA that contains the public key of the CA. Because the CA signs its own certificate, you should manually authenticate the public key of the CA by contacting the CA administrator when you enter this command.

If you are using Router Advertisements (RA) mode (using the **enrollment** command) when you issue the **crypto pki authenticate** command, then registration authority signing and encryption certificates will be returned from the CA and the CA certificate.

This command is not saved to the router configuration. However. the public keys embedded in the received CA (and RA) certificates are saved to the configuration as part of the Rivest, Shamir, and Adelman (RSA) public key record (called the "RSA public key chain").

Note If the CA does not respond by a timeout period after this command is issued, the terminal control will be returned so that it remains available. If this happens, you must reenter the command. Cisco IOS software will not recognize CA certificate expiration dates set for beyond the year 2049. If the validity period of the CA certificate is set to expire after the year 2049, the following error message will be displayed when authentication with the CA server is attempted:

```
error retrieving certificate :incomplete chain
```

If you receive an error message similar to this one, check the expiration date of your CA certificate. If the expiration date of your CA certificate is set after the year 2049, you must reduce the expiration date by a year or more.

Examples

In the following example, the router requests the certificate of the CA. The CA sends its certificate and the router prompts the administrator to verify the certificate of the CA by checking the CA certificate's fingerprint. The CA administrator can also view the CA certificate's fingerprint, so you should compare what the CA administrator sees to what the router displays on the screen. If the fingerprint on the router's screen matches the fingerprint viewed by the CA administrator, you should accept the certificate as valid.

```
Router(config)# crypto pki authenticate myca

Certificate has the following attributes:
Fingerprint: 0123 4567 89AB CDEF 0123
Do you accept this certificate? [yes/no] y#
```

Related Commands

Command	Description
debug crypto pki transactions	Displays debug messages for the trace of interaction (message type) between the CA and the router.
enrollment	Specifies the enrollment parameters of your CA.
show crypto pki certificates	Displays information about your certificate, the certificate of the CA, and any RA certificates.

crypto pki enroll

To obtain the certificates for your router from the certificate authority (CA), use the **crypto pki enroll** command in global configuration mode. To delete a current enrollment request, use the **no** form of this command.

crypto pki enroll *name*

no crypto pki enroll *name*

Syntax Description

name	The name of the CA. Use the same name as when you declared the CA using the **crypto pki trustpoint** command.

Defaults

No default behavior or values.

Command Modes

Global configuration (config)

Command History

Release	Modification
11.3T	The **crypto ca enroll** command was introduced.
12.3(7)T	This command replaced the **crypto ca enroll** command.
12.3(14)T	The command was modified to include self-signed certificate information.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(24)T	Support for IPv6 Secure Neighbor Discovery (SeND) was added.

Usage Guidelines

This command requests certificates from the CA for all of your router's Rivest, Shamir, and Adelmen (RSA) key pairs. This task is also known as enrolling with the CA. (Technically, enrolling and obtaining certificates are two separate events, but they both occur when this command is issued.)

Your router needs a signed certificate from the CA for each RSA key pairs of your router; if you previously generated general-purpose keys, this command obtains the one certificate corresponding to the one general-purpose RSA key pair. If you previously generated special-usage keys, this command obtains two certificates corresponding to each of the special-usage RSA key pairs.

If you already have a certificate for your keys you are prompted to remove the existing certificate first. (You can remove existing certificates with the **no certificate** command.)

The **crypto pki enroll** command is not saved in the router configuration.

Note If your router reboots after you issue the **crypto pki enroll** command but before you receive the certificates, you must reissue the command.

Note If you are using a Secure Shell (SSH) service, you should set up specific RSA key pairs (different private keys) for the trustpoint and the SSH service. (If the Public Key Infrastructure [PKI] and the SSH infrastructure share the same default RSA key pair, a temporary disruption of SSH service could occur. The RSA key pair could become invalid or change because of the CA system, in which case you would not be able to log in using SSH. You could receive the following error message: "key changed, possible security problem.")

Responding to Prompts

When you issue the **crypto pki enroll** command, you are prompted a number of times.

You are prompted to create a challenge password. This password can be up to 80 characters in length. This password is necessary in the event that you ever need to revoke your router's certificates. When you ask the CA administrator to revoke your certificate, you must supply this challenge password as a protection against fraudulent or mistaken revocation requests.

Note This password is not stored anywhere, so you need to remember this password.

If you lose the password, the CA administrator may still be able to revoke the router's certificate but will require further manual authentication of the router administrator identity.

You are also prompted to indicate whether your router's serial number should be included in the obtained certificate. The serial number is not used by IP Security (IPsec) or Internet Key Exchange, but may be used by the CA to either authenticate certificates or to later associate a certificate with a particular router. (Note that the serial number stored is the serial number of the internal board, not the one on the enclosure.) Ask your CA administrator if serial numbers should be included. If you are in doubt, include the serial number.

Normally, you would not include the IP address because the IP address binds the certificate more tightly to a specific entity. Also, if the router is moved, you would need to issue a new certificate. A router has multiple IP addresses, any of which might be used with IPsec.

If you indicate that the IP address should be included, you will then be prompted to specify the interface of the IP address. This interface should correspond to the interface that you apply your crypto map set to. If you apply crypto map sets to more than one interface, specify the interface that you name in the **crypto map local-address** command.

Examples

In the following example, a router with a general-purpose RSA key pair requests a certificate from the CA. When the router displays the certificate fingerprint, the administrator verifies this number by calling the CA administrator, which checks the number. The fingerprint is correct, so the router administrator accepts the certificate.

There can be a delay between when the router administrator sends the request and when the certificate is actually received by the router. The amount of delay depends on the CA method of operation.

```
Router(config)# crypto pki enroll myca
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.

Password: <mypassword>
```

```
Re-enter password: <mypassword>

% The subject name in the certificate will be: myrouter.example.com
% Include the router serial number in the subject name? [yes/no]: yes
% The serial number in the certificate will be: 03433678
% Include an IP address in the subject name [yes/no]? yes
Interface: ethernet0/0
Request certificate from CA [yes/no]? yes
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto pki certificates' command will also show the fingerprint.
```

Some time later, the router receives the certificate from the CA and displays the following confirmation message:

```
Router(config)#   Fingerprint: 01234567 89ABCDEF FEDCBA98 75543210

%CRYPTO-6-CERTRET: Certificate received from Certificate Authority

Router(config)#
```

If necessary, the router administrator can verify the displayed fingerprint with the CA administrator.

If there is a problem with the certificate request and the certificate is not granted, the following message is displayed on the console instead:

```
%CRYPTO-6-CERTREJ: Certificate enrollment request was rejected by Certificate Authority
```

The subject name in the certificate is automatically assigned to be the same as the RSA key pair's name. In the example, the RSA key pair was named "myrouter.example.com." (The router assigned this name.)

Requesting certificates for a router with special-usage keys would be the same as in the previous example, except that two certificates would have been returned by the CA. When the router received the two certificates, the router would have displayed the same confirmation message:

```
%CRYPTO-6-CERTRET: Certificate received from Certificate Authority
```

Related Commands

Command	Description
crypto map local address	Specifies and names an identifying interface to be used by the crypto map for IPsec traffic.
debug crypto pki messages	Displays debug messages for the details of the interaction (message dump) between the CA and the router.
debug crypto pki transactions	Displays debug messages for the trace of interaction (message type) between the CA and the router.
show crypto pki certificates	Displays information about your certificate, the certificate of the CA, and any RA certificates.

crypto pki import

To import a certificate manually via TFTP or as a cut-and-paste at the terminal, use the **crypto pki import** command in global configuration mode.

crypto pki import *name* **certificate**

Syntax Description

name **certificate**	Name of the certification authority (CA). This name is the same name used when the CA was declared with the **crypto pki trustpoint** command.

Defaults

No default behavior or values

Command Modes

Global configuration

Command History

Release	Modification
12.2(13)T	The **crypto ca import** command was introduced.
12.3(7)T	This command replaced the **crypto ca import** command.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(24)T	Support for IPv6 Secure Neighbor Discovery (SeND) was added.

Usage Guidelines

You must enter the **crypto pki import** command twice if usage keys (signature and encryption keys) are used. The first time the command is entered, one of the certificates is pasted into the router; the second time the command is entered, the other certificate is pasted into the router. (It does not matter which certificate is pasted first.)

Examples

The following example shows how to import a certificate via cut-and-paste. In this example, the CA trustpoint is "MS."

```
crypto pki trustpoint MS
 enroll terminal
 crypto pki authenticate MS
!
crypto pki enroll MS
crypto pki import MS certificate
```

Related Commands

Command	Description
crypto pki trustpoint	Declares the CA that your router should use.
enrollment	Specifies the enrollment parameters of your CA.
enrollment terminal	Specifies manual cut-and-paste certificate enrollment.

ctunnel mode

To transport IPv4 and IPv6 packets over Connectionless Network Service (CLNS) tunnel (CTunnel), use the **ctunnel mode** command in interface configuration mode. To return the ctunnel to the default **cisco** mode, use the **no** form of this command.

ctunnel mode [**gre** | **cisco**]

no ctunnel mode

Syntax Description

gre	(Optional) Sets the ctunnel mode to Generic Routing Encapsulation (GRE) for transporting IPv6 packets over the CLNS network.
cisco	(Optional) Returns the ctunnel mode to the default cisco.

Command Default

Cisco encapsulation tunnel mode is the default.

Command Modes

Interface configuration

Command History

Release	Modification
12.3(7)T	This command was introduced.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

GRE tunneling of IPv4 and IPv6 packets through CLNS-only networks enables Cisco ctunnels to interoperate with networking equipment from other vendors. This feature provides compliance with RFC 3147, *Generic Routing Encapsulation over CLNS Networks*, which should allow interoperation between Cisco equipment and that of other vendors. in which the same standard is implemented.

RFC 3147 specifies the use of GRE when tunneling packets. The implementation of this feature does not include support for GRE header fields such as those used to specify checksums, keys, or sequencing. Any packets received which specify the use of these features will be dropped.

The default ctunnel mode continues to use the standard Cisco encapsulation. Both ends of the tunnel must be configured with the same mode for it to work. If you want to tunnel ipv6 packets you must use the new gre mode.

Examples

The following example configures a CTunnel from one router to another and shows the CTunnel destination set to 49.0001.1111.1111.1111.00. The ctunnel mode is set to gre to transport IPv6 packets.

```
interface ctunnel 301
 ipv6 address 2001:0DB8:1111:2222::2/64
 ctunnel destination 49.0001.1111.1111.1111.00
```

```
ctunnel mode gre
```

Related Commands

Command	Description
clns routing	Enables routing of CLNS packets.
ctunnel destination	Specifies the destination for the CTunnel.
debug ctunnel	Displays debug messages for the IP over a CLNS Tunnel feature.
interface ctunnel	Creates a virtual interface to transport IP over a CLNS tunnel.
ip address	Sets a primary or secondary IP address for an interface.

debug adjacency

To enable the display of information about the adjacency database, use the **debug adjacency** command in privileged EXEC mode. To disable the display of these events, use the **no** form of this command.

debug adjacency [**epoch** | **ipc** | **state** | **table**] [*prefix*] [*interface*] [**connectionid** *id*] [**link** {**ipv4** | **ipv6** | **mpls**}]

no debug adjacency [**epoch** | **ipc** | **state** | **table**] [*prefix*] [*interface*] [**connectionid** *id*] [**link** {**ipv4** | **ipv6** | **mpls**}]

Syntax Description

epoch	(Optional) Displays adjacency epoch events.
ipc	(Optional) Displays interprocess communication (IPC) events for adjacencies.
state	(Optional) Displays adjacency system state machine events.
table	(Optional) Displays adjacency table operations.
prefix	(Optional) Displays debugging events for the specified IP address or IPv6 address. **Note** On the Cisco 10000 series routers, IPv6 is supported in Cisco IOS Release 12.2(28)SB and later releases.
interface	(Optional) Displays debugging events for the specified interface. For line cards, you must specify the line card if_number (interface number). Use the **show cef interface** command to obtain line card if_numbers.
connectionid *id*	(Optional) Displays debugging events for the specified client connection identification number.
link {**ipv4** | **ipv6** | **mpls**}	(Optional) Displays debugging events for the specified link type (IP, IPv6, or Multiprotocol Label Switching [MPLS] traffic). **Note** On the Cisco 10000 series routers, IPv6 is supported in Cisco IOS Release 12.2(28)SB and later releases.

Command Default

Debugging events are not displayed.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(7)XE	This command was introduced on the Cisco 7600 series routers.
12.1(1)E	This command was implemented on the Cisco 7600 series routers.
12.2(14)SX	This command was implemented on the Supervisor Engine 720.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S, and the *prefix*, *interface*, **connectionid** *id*, and **link** {**ipv4** | **ipv6** | **mpls**} keywords and arguments were added.

Release	Modification
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, you should use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Also, you should use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

You can use any combination of the *prefix*, *interface*, **connectionid** *id*, and **link** {**ipv4 | ipv6 | mpls**} keywords and arguments (in any order) as a filter to enable debugging for a specified subset of adjacencies.

Note On the Cisco 10000 series routers, IPv6 is supported in Cisco IOS Release 12.2(28)SB and later releases.

Examples

The following example shows how to display information on the adjacency database:

```
Router# debug adjacency

*Jan 27 06:22:50.543: ADJ-ios_mgr: repopulate adjs on up event for Ethernet3/0
*Jan 27 06:22:50.543: ADJ: IPV6 adj out of Ethernet3/0, addr FE80::20C:CFFF:FEDF:6854
(incomplete) no src set: init/update from interface
*Jan 27 06:22:50.543: ADJ: IPV6 adj out of Ethernet3/0, addr FE80::20C:CFFF:FEDF:6854
(incomplete) no src set: set bundle to IPv6 adjacency oce
*Jan 27 06:22:50.543: ADJ: IPV6 adj out of Ethernet3/0, addr FE80::20C:CFFF:FEDF:6854
(incomplete) no src set: allocated, setup and inserted OK
*Jan 27 06:22:50.543: ADJ: IPV6 adj out of Ethernet3/0, addr FE80::20C:CFFF:FEDF:6854
(incomplete) src IPv6 ND: source IPv6 ND added OK
*Jan 27 06:22:50.543: ADJ: IPV6 adj out of Ethernet3/0, addr FE80::20C:CFFF:FEDF:6854
(incomplete) src IPv6 ND: computed macstring (len 14): OK
*Jan 27 06:22:50.543: ADJ: IPV6 adj out of Ethernet3/0, addr FE80::20C:CFFF:FEDF:6854 src
IPv6 ND: made complete (macstring len 0 to 14/0 octets)
00:04:40: %LINK-3-UPDOWN: Interface Ethernet3/0, changed state to up
00:04:41: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet3/0, changed
```

Related Commands

Command	Description
clear adjacency	Clears the Cisco Express Forwarding adjacency table.
clear arp-cache	Deletes all dynamic entries from the ARP cache.
show adjacency	Displays Cisco Express Forwarding adjacency table information.
show mls cef adjacency	Displays information about the hardware Layer 3 switching adjacency node.

debug bfd

To display debugging messages about Bidirectional Forwarding Detection (BFD), use the **debug bfd** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

Cisco IOS Release 12.2(18)SXE, 12.4(4)T, and 12.2(33)SRA

debug bfd {**event** | **packet** [*ip-address* | *ipv6-address*]}

no debug bfd {**event** | **packet** [*ip-address* | *ipv6-address*]}

Cisco IOS Release 12.0(31)S

debug bfd {**event** | **packet** [*ip-address*] | **ipc-error** | **ipc-event** | **oir-error** | **oir-event**}

no debug bfd {**event** | **packet** [*ip-address*] | **ipc-error** | **ipc-event** | **oir-error** | **oir-event**}

Syntax Description

event	Displays debugging information about BFD state transitions.
packet	Displays debugging information about BFD control packets.
ip-address	(Optional) Displays debugging information about BFD only for the specified IP address.
ipv6-address	(Optional) Displays debugging information about BFD only for the specified IPv6 address.
ipc-error	(Optional) Displays debugging information with interprocess communication (IPC) errors on the Route Processor (RP) and line card (LC).
ipc-event	(Optional) Displays debugging information with IPC events on the RP and LC.
oir-error	(Optional) Displays debugging information with online insertion and removal (OIR) errors on the RP and LC.
oir-event	(Optional) Displays debugging information with OIR events on the RP and LC.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(18)SXE	This command was introduced.
12.0(31)S	This command was integrated into Cisco IOS Release 12.0(31)S.
12.4(4)T	This command was integrated into Cisco IOS Release 12.4(4)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SRE	This command was modified. Support for IPv6 was added.
15.1(2)T	This command was modified. Support for IPv6 was added to Cisco IOS Release 15.1(2)T.

Usage Guidelines

The **debug bfd** command can be used to troubleshoot the BFD feature.

Note Because BFD is designed to send and receive packets at a very high rate of speed, consider the potential effect on system resources before enabling this command, especially if there are a large number of BFD peers. The **debug bfd packet** command should be enabled only on a live network at the direction of Cisco Technical Assistance Center personnel.

Examples

The following example shows output from the **debug bfd packet** command. The IP address has been specified in order to limit the packet information to one interface:

```
Router# debug bfd packet 172.16.10.5

BFD packet debugging is on
*Jan 26 14:47:37.645: Tx*IP: dst 172.16.10.1, plen 24. BFD: diag 2, St/D/P/F (1/0/0/0),
mult 5, len 24, loc/rem discr 1 1, tx 1000000, rx 1000000 100000, timer 1000 ms, #103
*Jan 26 14:47:37.645: %OSPF-5-ADJCHG: Process 10, Nbr 172.16.10.12 on Ethernet1/4 from
FULL to DOWN, Neighbor Down: BFD node down
*Jan 26 14:47:50.685: %OSPF-5-ADJCHG: Process 10, Nbr 172.16.10.12 on Ethernet1/4 from
LOADING to FULL, Loading Done
*Jan 26 14:48:00.905: Rx  IP: src 172.16.10.1, plen 24. BFD: diag 0, St/D/P/F (1/0/0/0),
mult 4, len 24, loc/rem discr 2 1, tx 1000000, rx 1000000 100000, timer 4000 ms, #50
*Jan 26 14:48:00.905: Tx IP: dst 172.16.10.1, plen 24. BFD: diag 2, St/D/P/F (2/0/0/0),
mult 5, len 24, loc/rem discr 1 2, tx 1000000, rx 1000000 100000, timer 1000 ms, #131
*Jan 26 14:48:00.905: Rx  IP: src 172.16.10.1, plen 24. BFD: diag 0, St/D/P/F (3/0/0/0),
mult 4, len 24, loc/rem discr 2 1, tx 1000000, rx 1000000 100000, timer 4000 ms, #51
*Jan 26 14:48:00.905: Tx IP: dst 172.16.10.1, plen 24. BFD: diag 0, St/D/P/F (3/0/0/0),
mult 5, len 24, loc/rem discr 1 2, tx 1000000, rx 1000000 100000, timer 1000 ms, #132
```

The following example shows output from the **debug bfd event** command when an interface between two BFD neighbor routers fails and then comes back online:

```
Router# debug bfd event

22:53:48: BFD: bfd_neighbor - action:DESTROY, proc:1024, idb:FastEthernet0/1,
neighbor:172.16.10.2
22:53:48: BFD: bfd_neighbor - action:DESTROY, proc:512, idb:FastEthernet0/1,
neighbor:172.16.10.2
22:53:49: Session [172.16.10.1,172.16.10.2,Fa0/1,1], event DETECT TIMER EXPIRED, state UP
-> FAILING
.
.
.
22:56:35: BFD: bfd_neighbor - action:CREATE, proc:1024, idb:FastEthernet0/1,
neighbor:172.16.10.2
22:56:37: Session [172.16.10.1,172.16.10.2,Fa0/1,1], event RX IHY 0, state FAILING -> DOWN
22:56:37: Session [172.16.10.1,172.16.10.2,Fa0/1,1], event RX IHY 0, state DOWN -> INIT
22:56:37: Session [172.16.10.1,172.16.10.2,Fa0/1,1], event RX IHY 1, state INIT -> UP
```

Table 10 describes the significant fields shown in the display.

Table 10 debug bfd event Field Descriptions

Field	Description
bfd_neighbor - action:DESTROY	The BFD neighbor will tear down the BFD session.
Session [172.16.10.1, 172.16.10.2, Fa0/1,1]	IP addresses of the BFD neighbors holding this session that is carried over FastEthernet interface 0/1.
event DETECT TIMER EXPIRED	The BFD neighbor has not received BFD control packets within the negotiated interval and the detect timer has expired.
state UP -> FAILING	The BFD event state is changing from Up to Failing.
Session [172.16.10.1, 172.16.10.2, Fa0/1,1], event RX IHY 0	The BFD session between the neighbors indicated by the IP addresses that is carried over FastEthernet interface 0/1 is changing state from Failing to Down. The I Hear You (IHY) bit value is shown as 0 to indicate that the remote system is tearing down the BFD session.
event RX IHY 0, state DOWN -> INIT	The BFD session is still considered down, and the IHY bit value still is shown as 0, and the session state changes from DOWN to INIT to indicate that the BFD session is again initializing, as the interface comes back up.
event RX IHY 1, state INIT -> UP	The BFD session has been reestablished, and the IHY bit value changes to 1 to indicate that the session is live. The BFD session state changes from INIT to UP.

The following example shows output from the **debug bfd packet** command when an interface between two BFD neighbor routers fails and then comes back online. The diagnostic code changes from 0 (No Diagnostic) to 1 (Control Detection Time Expired) because no BFD control packets could be sent (and therefore detected by the BFD peer) after the interface fails. When the interface comes back online, the diagnostic code changes back to 0 to signify that BFD packets can be sent and received by the BFD peers.

```
Router# debug bfd packet

23:03:25: Rx  IP: src 172.16.10.2, plen 24. BFD: diag 0, H/D/P/F (0/0/0/0), mult 3, len
24, loc/rem discr 5 1, tx 1000000, rx 100007
23:03:25: Tx IP: dst 172.16.10.2, plen 24. BFD: diag 1, H/D/P/F (0/0/0/0), mult 5, len 24,
loc/rem discr 1 5, tx 1000000, rx 1000008
23:03:25: Tx IP: dst 172.16.10.2, plen 24. BFD: diag 1, H/D/P/F (1/0/0/0), mult 5, len 24,
loc/rem discr 1 5, tx 1000000, rx 1000009
```

Table 11 describes the significant fields shown in the display.

Table 11 debug bfd packet Field Descriptions

Field	Description
Rx IP: src 172.16.10.2	The router has received this BFD packet from the BFD router with source address 172.16.10.2.
plen 24	Length of the BFD control packet, in bytes.

Table 11 debug bfd packet Field Descriptions (continued)

Field	Description
diag 0	A diagnostic code specifying the local system's reason for the last transition of the session from Up to some other state. State values are as follows: • 0—No Diagnostic • 1—Control Detection Time Expired • 2—Echo Function Failed • 3—Neighbor Signaled Session Down • 4—Forwarding Plane Reset • 5—Path Down • 6—Concentrated Path Down • 7—Administratively Down
H/D/P/F (0/0/0/0)	H bit—Hear You bit. This bit is set to 0 if the transmitting system either is not receiving BFD packets from the remote system or is tearing down the BFD session. During normal operation the I Hear You bit is set to 1. D bit—Demand Mode bit. If the Demand Mode bit set, the transmitting system wants to operate in demand mode. BFS has two modes—asynchronous and demand. The Cisco implementation of BFD supports only asynchronous mode. P bit—Poll bit. If the Poll bit is set, the transmitting system is requesting verification of connectivity or of a parameter change. F bit—Final bit. If the Final bit is set, the transmitting system is responding to a received BFC control packet that had a Poll (P) bit set.
mult 3	Detect time multiplier. The negotiated transmit interval, multiplied by the detect time multiplier, determines the detection time for the transmitting system in BFD asynchronous mode. The detect time multiplier is similar to the hello multiplier in IS-IS, which is used to determine the hold timer: (hellointerval) * (hellomultiplier) = hold timer. If a hello packet is not received within the hold-timer interval, a failure has occurred. Similarly, for BFD: (transmit interval) * (detect multiplier) = detect timer. If a BFD control packet is not received from the remote system within the detect-timer interval, a failure has occurred.
len 24	The BFD packet length.

Table 11 debug bfd packet Field Descriptions (continued)

Field	Description
loc/rem discr 5 1	The values for My Discriminator (local) and Your Discriminator (remote) BFD neighbors. • My Discriminator—Unique, nonzero discriminator value generated by the transmitting system, used to demultiplex multiple BFD sessions between the same pair of systems. • Your Discriminator—The discriminator received from the corresponding remote system. This field reflects the received value of My Discriminator, or is zero if that value is unknown.
tx 1000000	Desired minimum transmit interval.
rx 100007	Required minimum receive interval.

debug bgp ipv6 dampening

To display debugging messages for IPv6 Border Gateway Protocol (BGP) dampening, use the **debug bgp ipv6 dampening** command in privileged EXEC mode. To disable debugging messages for IPv6 BGP dampening, use the **no** form of this command.

debug bgp ipv6 {**unicast** | **multicast**} **dampening** [**prefix-list** *prefix-list-name*]

no debug bgp ipv6 {**unicast** | **multicast**} **dampening** [**prefix-list** *prefix-list-name*]

Syntax Description

unicast	Specifies IPv6 unicast address prefixes.
multicast	Specifies IPv6 multicast address prefixes.
prefix-list *prefix-list-name*	(Optional) Name of an IPv6 prefix list.

Command Default

Debugging for IPv6 BGP dampening packets is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(13)T	The **prefix-list** keyword was added.
12.0(24)S	The **prefix-list** keyword was added.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

Usage Guidelines

The **debug bgp ipv6 dampening** command is similar to the **debug ip bgp dampening** command, except that it is IPv6-specific.

Use the **prefix-list** keyword and an argument to filter BGP IPv6 dampening debug information through an IPv6 prefix list.

Note By default, the network server sends the output from debug commands and system error messages to the console. To redirect debugging output, use the **logging** command options within global configuration mode. Destinations are the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server.

Examples

The following is sample output from the **debug bgp ipv6 dampening** command:

```
Router# debug bgp ipv6 dampening

00:13:28:BGP(1):charge penalty for 2000:0:0:1::/64 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:13:28:BGP(1):charge penalty for 2000:0:0:1:1::/80 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:13:28:BGP(1):charge penalty for 2000:0:0:5::/64 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:16:03:BGP(1):charge penalty for 2000:0:0:1::/64 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:16:03:BGP(1):flapped 2 times since 00:02:35. New penalty is 1892

00:18:28:BGP(1):suppress 2000:0:0:1:1::/80 path 2 1 for 00:27:30 (penalty 2671)
00:18:28:halflife-time 15, reuse/suppress 750/2000
00:18:28:BGP(1):suppress 2000:0:0:1::/64 path 2 1 for 00:27:20 (penalty 2664)
00:18:28:halflife-time 15, reuse/suppress 750/2000
```

The following example shows output for the **debug bgp ipv6 dampening** command filtered through the prefix list named marketing:

```
Router# debug bgp ipv6 dampening prefix-list marketing

00:16:08:BGP(1):charge penalty for 2001:0DB8::/64 path 30 with halflife-time 15
reuse/suppress 750/2000
00:16:08:BGP(1):flapped 1 times since 00:00:00. New penalty is 10
```

Table 12 describes the fields shown in the display.

Table 12 debug bgp ipv6 dampening Field Descriptions

Field	Description
penalty	Numerical value of 1000 assigned to a route by a router configured for route dampening in another autonomous system each time a route flaps. Penalties are cumulative. The penalty for the route is stored in the BGP routing table until the penalty exceeds the suppress limit. If the penalty exceeds the suppress limit, the route state changes from history to damp.
flapped	Number of times a route is available, then unavailable, or vice versa.
halflife-time	Amount of time (in minutes) by which the penalty is decreased after the route is assigned a penalty. The halflife-time value is half of the half-life period (which is 15 minutes by default). Penalty reduction happens every 5 seconds.

Table 12 debug bgp ipv6 dampening Field Descriptions (continued)

Field	Description
reuse	The limit by which a route is unsuppressed. If the penalty for a flapping route decreases and falls below this reuse limit, the route is unsuppressed. That is, the route is added back to the BGP table and once again used for forwarding. The default reuse limit is 750. Routes are unsuppressed at 10-second increments. Every 10 seconds, the router determines which routes are now unsuppressed and advertises them to the world.
suppress	Limit by which a route is suppressed. If the penalty exceeds this limit, the route is suppressed. The default value is 2000.
maximum suppress limit (not shown in sample output)	Maximum amount of time (in minutes) a route is suppressed. The default value is four times the half-life period.
damp state (not shown in sample output)	State in which the route has flapped so often that the router will not advertise this route to BGP neighbors.

Related Commands

Command	Description
debug bgp ipv6 updates	Displays debugging messages for IPv6 BGP update packets.

debug bgp ipv6 updates

To display debugging messages for IPv6 Border Gateway Protocol (BGP) update packets, use the **debug bgp ipv6 updates** command in privileged EXEC mode. To disable debugging messages for IPv6 BGP update packets, use the **no** form of this command.

debug bgp ipv6 {**unicast** | **multicast**} **updates** [*ipv6-address*] [**prefix-list** *prefix-list-name*] [**in** | **out**]

no debug bgp ipv6 {**unicast** | **multicast**} **updates** [*ipv6-address*] [**prefix-list** *prefix-list-name*] [**in** | **out**]

Syntax Description

unicast	Specifies IPv6 unicast address prefixes.
multicast	Specifies IPv6 multicast address prefixes.
ipv6-address	(Optional) The IPv6 address of a BGP neighbor. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
prefix-list *prefix-list-name*	(Optional) Name of an IPv6 prefix list.
in	(Optional) Indicates inbound updates.
out	(Optional) Indicates outbound updates.

Command Default

Debugging for IPv6 BGP update packets is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(13)T	The **prefix-list** keyword was added.
12.0(24)S	The **prefix-list** keyword was added.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

Usage Guidelines

The **debug bgp ipv6 updates** command is similar to the **debug ip bgp updates** command, except that it is IPv6-specific.

Use the **prefix-list** keyword to filter BGP IPv6 updates debugging information through an IPv6 prefix list.

Note By default, the network server sends the output from **debug** commands and system error messages to the console. To redirect debugging output, use the **logging** command options within global configuration mode. Destinations are the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server. For complete information on **debug** commands and redirecting debugging output, refer to the Release 12.2 *Cisco IOS Debug Command Reference*.

Examples

The following is sample output from the **debug bgp ipv6 updates** command:

```
Router# debug bgp ipv6 updates

14:04:17:BGP(1):2000:0:0:2::2 computing updates, afi 1, neighbor version 0, table version
1, starting at ::
14:04:17:BGP(1):2000:0:0:2::2 update run completed, afi 1, ran for 0ms, neighbor version
0, start version 1, throttled to 1
14:04:19:BGP(1):sourced route for 2000:0:0:2::1/64 path #0 changed (weight 32768)
14:04:19:BGP(1):2000:0:0:2::1/64 route sourced locally
14:04:19:BGP(1):2000:0:0:2:1::/80 route sourced locally
14:04:19:BGP(1):2000:0:0:3::2/64 route sourced locally
14:04:19:BGP(1):2000:0:0:4::2/64 route sourced locally
14:04:22:BGP(1):2000:0:0:2::2 computing updates, afi 1, neighbor version 1, table version
6, starting at ::
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (format) 2000:0:0:2::1/64, next 2000:0:0:2::1,
metric 0, path
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (format) 2000:0:0:2:1::/80, next 2000:0:0:2::1,
metric 0, path
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (prepend, chgflags:0x208) 2000:0:0:3::2/64, next
2000:0:0:2::1, metric 0, path
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (prepend, chgflags:0x208) 2000:0:0:4::2/64, next
2000:0:0:2::1, metric 0, path
```

The following is sample output from the **debug bgp ipv6 updates** command filtered through the prefix list named sales:

```
Router# debug bgp ipv6 updates prefix-list sales

00:18:26:BGP(1):2000:8493:1::2 send UPDATE (prepend, chgflags:0x208) 7878:7878::/64, next
2001:0DB8::36C, metric 0, path
```

Table 13 describes the significant fields shown in the display.

Table 13 debug bgp ipv6 updates Field Descriptions

Field	Description
BGP(1):	BGP debugging for address family index (afi) 1.
afi	Address family index.
neighbor version	Version of the BGP table on the neighbor from which the update was received.

Table 13 debug bgp ipv6 updates Field Descriptions (continued)

Field	Description
table version	Version of the BGP table on the router from which you entered the **debug bgp ipv6 updates** command.
starting at	Starting at the network layer reachability information (NLRI). BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address, community values, and other information.
route sourced locally	Indicates that a route is sourced locally and that updates are not sent for the route.
send UPDATE (format)	Indicates that an update message for a reachable network should be formatted. Addresses include prefix and next hop.
send UPDATE (prepend, chgflags:0x208)	Indicates that an update message about a path to a BGP peer should be written.

Related Commands

Command	Description
debug bgp ipv6 dampening	Displays debugging messages for IPv6 BGP dampening packets.

debug bgp vpnv6 unicast

To display Border Gateway Protocol (BGP) virtual private network (VPN) debugging output, use the **debug bgp vpnv6 unicast** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug bgp vpnv6 unicast

no debug bgp vpnv6

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRB	This command was introduced.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

Use the **debug bgp vpnv6 unicast** command to help troubleshoot the BGP VPN.

Note By default, the network server sends the output from debug commands and system error messages to the console. To redirect debugging output, use the logging command options within global configuration mode. Destinations are the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server. For complete information on debug commands and redirecting debugging output, refer to the *Cisco IOS Debug Command Reference*, Release 12.4.

Examples

The following example enables BGP debugging output for IPv6 VPN instances:

```
Router# debug bgp vpnv6 unicast
```

debug crypto condition

To define conditional debug filters, use the **debug crypto condition** command in privileged EXEC mode. To disable conditional debugging, use the **no** form of this command.

debug crypto condition [**connid** *integer* **engine-id** *integer*] [**flowid** *integer* **engine-id** *integer*] [**gdoi-group** *groupname*] [**isakmp profile** *profile-name*] [**fvrf** *string*] [**ivrf** *string*] [**local** {**ipv4** *ip-address* | **ipv6** *ip-address*}] [**peer** [**group** *string*] [**hostname** *string*] [**ipv4** *ip-address* | **ipv6** *ip-address*] [**subnet** *subnet mask* | *ipv6-prefix*] [**username** *string*]] [**spi** *integer*] [**reset**] [**unmatched** [**engine**] [**gdoi-group**] [**ipsec**] [**isakmp**] [**username** *string*]]

no debug crypto condition [**connid** *integer* **engine-id** *integer*] [**flowid** *integer* **engine-id** *integer*] [**gdoi-group** *groupname*] [**isakmp profile** *profile-name*] [**fvrf** *string*] [**ivrf** *string*] [**local** {**ipv4** *ip-address* | **ipv6** *ip-address*}] [**peer** [**group** *string*] [**hostname** *string*] [**ipv4** *ip-address* | **ipv6** *ip-address*] [**subnet** *subnet mask* | *ipv6-prefix*] [**username** *string*]] [**spi** *integer*] [**reset**] [**unmatched** [**engine**] [**gdoi-group**] [**ipsec**] [**isakmp**] [**username** *string*]]

Syntax Description

connid *integer*[1]	(Optional) Specifies the Internet Key Exchange (IKE) and IP Security (IPsec) connection ID filter. Valid values range from 1 to 32766.
engine-id *integer*	(Optional) Specifies the Crypto engine ID value, which can be retrieved via the **show crypto isakmp sa detail** command. Valid values are 1, which represents software engines, and 2, which represents hardware engines.
flowid *integer*	(Optional) Specifies the IPsec flow-ID filter. Valid values range from 1 to 32766.
gdoi-group *groupname*	(Optional) Specifies the Group Domain of Interpretation (GDOI) group filter. • The *groupname* value is the name of the GDOI group.
isakmp profile *profile-name*	(Optional) Specifies the filter for the Internet Security Association Key Management Protocol (ISAKMP) profile. • The *profile-name* value is the name of the ISAKMP profile to be filtered.
fvrf *string*[1]	(Optional) Specifies the Front-door Virtual Private Network (VPN) Routing and Forwarding (FVRF) filter. The *string* argument must be the name string of an FVRF instance.
ivrf *string*[1]	(Optional) Specifies the Inside VRF (IVRF) filter. The *string* argument must be the name string of an IVRF instance.
local {**ipv4** *ip-address* | **ipv6** *ip-address*}	(Optional) Specifies the IKE local address filter. • The *ip-address* value is the IP address of the local crypto endpoint.

peer[1]	(Optional) Specifies the IKE peer filter. At least one of the following keywords and arguments must be used: • **group** *string*—Unity group name filter of the IKE peer. • **hostname** *string*—Fully qualified domain name (FQDN) hostname filter of the IKE peer. • **ipv4** *ip-address* or **ipv6** *ip-address*—IP address filter of the IKE peer. • **subnet** *subnet mask* or **subnet** *ipv6-prefix*—Range of IKE peer IP addresses or prefix length. • **username** *string*—FQDN username filter of the IKE peer.
spi *integer*[1]	(Optional) Specifies the security policy index (SPI) filter. The integer must be a 32-bit unsigned integer.
reset	(Optional) Deletes all crypto debug filters. **Note** It is suggested that you turn off all crypto global debugging before using this keyword; otherwise, your system may be flooded with debug messages.
unmatched	(Optional) Filters all debug messages or only specified debug messages by choosing any of the following keywords: • **engine**—Output crypto engine debugs even if no context is available. • **gdoi-group**—Output GDOI group debugs even if no match occurs. • **ipsec**—Output IPsec debugs even if no context is available. • **isakmp**—Output IKE debugs even if no context is available.
username *string*	(Optional) Specifies the XAUTH or PKI-AAA username filter.

1. Additional conditional filters (IP address, subnet mask, username, hostname, group, connection-ID, flow-ID, SPI, FVRF, and IVRF) can be specified more than once by repeating the **debug crypto condition** command with any of the available filters.

Defaults

Crypto conditional debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(2)T	This command was introduced.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.4(11)T	The **gdoi-group** *groupname*, **isakmp profile** *profile-name*, **local ipv4** *ip-address*, **unmatched**, and **username** *string* keywords and arguments were added.

Release	Modification
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.1(4)M	This command was modified. The **ipv6** keyword was added to provide support for IPv6 addresses.

Usage Guidelines

Before enabling the **debug crypto condition** command, you must decide what debug condition types (also known as debug filters) and values will be used. The volume of debug messages is dependent on the number of conditions you define.

Note Specifying numerous debug conditions may consume CPU cycles and have a negative effect on router performance.

To begin crypto conditional debugging, you must also enable at least one global crypto debug command—**debug crypto isakmp**, **debug crypto ipsec**, and **debug crypto engine**; otherwise, conditional debugging will not occur. This requirement helps to ensure that the performance of the router will not be impacted when conditional debugging is not being used.

Note Debug message filtering for hardware crypto engines is not supported.

Examples

The following example shows how to display debug messages when the peer IP address is 10.1.1.1, 10.1.1.2, or 10.1.1.3 and when the connection-ID 2000 of crypto engine 0 is used. This example also shows how to enable global debug crypto CLIs and enable the **show crypto debug-condition** command to verify conditional settings.

```
Router# debug crypto condition connid 2000 engine-id 1
Router# debug crypto condition peer ipv4 10.1.1.1
Router# debug crypto condition peer ipv4 10.1.1.2
Router# debug crypto condition peer ipv4 10.1.1.3
Router# debug crypto condition unmatched
! Verify crypto conditional settings.
Router# show crypto debug-condition

Crypto conditional debug currently is turned ON
IKE debug context unmatched flag:ON
IPsec debug context unmatched flag:ON
Crypto Engine debug context unmatched flag:ON

IKE peer IP address filters:
10.1.1.1  10.1.1.2   10.1.1.3

Connection-id filters:[connid:engine_id]2000:1,
! Enable global crypto CLIs to start conditional debugging.
Router# debug crypto isakmp
Router# debug crypto ipsec
Router# debug crypto engine
```

The following example shows how to disable all crypto conditional settings via the **reset** keyword:

```
Router# no debug crypto isakmp
Router# no debug crypto ipsec
```

```
Router# no debug crypto engine
Router# debug crypto condition reset

! Verify that all crypto conditional settings have been disabled.

Router# show crypto debug-condition

Crypto conditional debug currently is turned OFF
IKE debug context unmatched flag:OFF
IPsec debug context unmatched flag:OFF
Crypto Engine debug context unmatched flag:OFF
```

Related Commands

Command	Description
show crypto debug-condition	Displays crypto debug conditions that have already been enabled in the router.
show crypto debug-condition unmatched	Displays crypto conditional debug messages when context information is unavailable to check against debug conditions.
show crypto ipsec sa	Displays the settings used by current SAs.
show crypto isakmp sa	Displays all current IKE SAs at a peer.