

debug ip msdp

To debug Multicast Source Discovery Protocol (MSDP) activity, use the **debug ip msdp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug ip msdp [vrf vrf-name] [peer-address | name] [detail] [routes]
```

```
no debug ip msdp [vrf vrf-name] [peer-address | name] [detail] [routes]
```

Syntax Description	
vrf	(Optional) Supports the Multicast Virtual Private Network (VPN) routing and forwarding (VRF) instance.
<i>vrf-name</i>	(Optional) Name assigned to the VRF.
<i>peer-address name</i>	(Optional) The peer for which debug events are logged.
detail	(Optional) Provides more detailed debugging information.
routes	(Optional) Displays the contents of Source-Active messages.

Defaults No default behavior or values.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.0(7)T	This command was introduced.
	12.0(23)S	The vrf keyword and <i>vrf-name</i> argument were added.
	12.2(13)T	The vrf keyword and <i>vrf-name</i> argument were added in Release 12.2T.

Examples The following is sample output from the **debug ip msdp** command:

```
Router# debug ip msdp

MSDP debugging is on
Router#
MSDP: 224.150.44.254: Received 1388-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1388, ec: 115, RP: 172.31.3.92
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.92, used EMBGP peer
MSDP: 224.150.44.250: Forward 1388-byte SA to peer
MSDP: 224.150.44.254: Received 1028-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1028, ec: 85, RP: 172.31.3.92
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.92, used EMBGP peer
MSDP: 224.150.44.250: Forward 1028-byte SA to peer
MSDP: 224.150.44.254: Received 1388-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1388, ec: 115, RP: 172.31.3.111
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.111, used EMBGP peer
MSDP: 224.150.44.250: Forward 1388-byte SA to peer
MSDP: 224.150.44.250: Received 56-byte message from peer
MSDP: 224.150.44.250: SA TLV, len: 56, ec: 4, RP: 205.167.76.241
MSDP: 224.150.44.250: Peer RPF check passed for 205.167.76.241, used EMBGP peer
MSDP: 224.150.44.254: Forward 56-byte SA to peer
```

```

MSDP: 224.150.44.254: Received 116-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 116, ec: 9, RP: 172.31.3.111
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.111, used EMBGP peer
MSDP: 224.150.44.250: Forward 116-byte SA to peer
MSDP: 224.150.44.254: Received 32-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 32, ec: 2, RP: 172.31.3.78
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.78, used EMBGP peer
MSDP: 224.150.44.250: Forward 32-byte SA to peer
    
```

Table 100 describes the significant fields shown in the display.

Table 100 *debug ip msdp Field Descriptions*

Field	Description
MSDP	Protocol being debugged.
224.150.44.254:	IP address of the MSDP peer.
Received 1388-byte message from peer	MSDP event.

debug ip msdp resets

To debug Multicast Source Discovery Protocol (MSDP) peer reset reasons, use the **debug ip msdp resets** command in privileged EXEC mode.

debug ip msdp [**vrf** *vrf-name*] **resets**

Syntax Description

vrf	(Optional) Supports the Multicast Virtual Private Network (VPN) routing and forwarding (VRF) instance.
<i>vrf-name</i>	(Optional) Name assigned to the VRF.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(7)T	This command was introduced.
12.0(23)S	The vrf keyword and <i>vrf-name</i> argument were added.
12.2(13)T	The vrf keyword and <i>vrf-name</i> argument were added in Release 12.2T.

debug ip nat

To display information about IP packets translated by the IP Network Address Translation (NAT) feature, use the **debug ip nat** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip nat [*access-list* | **detailed** | **h323** | **ipsec** | **pptp** | **sip** | **vrf**]

no debug ip nat [*access-list* | **detailed** | **h323** | **ipsec** | **pptp** | **sip** | **vrf**]

Syntax Description

<i>access-list</i>	(Optional) The standard IP access list number. If the datagram is not permitted by the specified access list, the related debugging output is suppressed.
detailed	(Optional) Displays debug information in a detailed format.
h323	(Optional) Displays H.225 and H.245 protocol information.
ipsec	(Optional) Displays IP Security (IPSec) packet information.
pptp	(Optional) Displays Point-to-Point Tunneling Protocol (PPTP) information.
sip	(Optional) Displays Session Initiation Protocol (SIP) information.
vrf	(Optional) Displays Virtual Private Network (VPN) routing and forwarding (VRF) traffic-related information.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.1(5)T	The h323 keyword was added.
12.2(8)T	The sip keyword was added.
12.2(13)T	The ipsec and vrf keywords were added.

Usage Guidelines

The NAT feature reduces the need for unique, registered IP addresses. It can also save private network administrators from needing to renumber hosts and routers that do not conform to global IP addressing.

Use the **debug ip nat** command to verify the operation of the NAT feature by displaying information about every packet that is translated by the router. The **debug ip nat detailed** command generates a description of each packet considered for translation. This command also outputs information about certain errors or exceptional conditions, such as the failure to allocate a global address. To display messages related to the processing of H.225 signaling and H.245 messages, use the **debug ip nat h323** command. To display messages related to the processing of SIP messages, use the **debug ip nat sip** command. To display messages related to the processing of VRF messages, use the **debug ip nat vrf** command.

**Caution**

Because the **debug ip nat** command generates a substantial amount of output, use it only when traffic on the IP network is low, so other activity on the system is not adversely affected.

Examples

The following is sample output from the **debug ip nat** command. In this example, the first two lines show the debugging output produced by a Domain Name System (DNS) request and reply. The remaining lines show the debugging output from a Telnet connection from a host on the inside of the network to a host on the outside of the network. All Telnet packets, except for the first packet, were translated in the fast path, as indicated by the asterisk (*).

```
Router# debug ip nat

NAT: s=192.168.1.95->172.31.233.209, d=172.31.2.132 [6825]
NAT: s=172.31.2.132, d=172.31.233.209->192.168.1.95 [21852]
NAT: s=192.168.1.95->172.31.233.209, d=172.31.1.161 [6826]
NAT*: s=172.31.1.161, d=172.31.233.209->192.168.1.95 [23311]
NAT*: s=192.168.1.95->172.31.233.209, d=172.31.1.161 [6827]
NAT*: s=192.168.1.95->172.31.233.209, d=172.31.1.161 [6828]
NAT*: s=172.31.1.161, d=172.31.233.209->192.168.1.95 [23313]
NAT*: s=172.31.1.161, d=172.31.233.209->192.168.1.95 [23325]
```

[Table 101](#) describes the significant fields shown in the display.

Table 101 *debug ip nat Field Descriptions*

Field	Description
NAT:	Indicates that the packet is being translated by the NAT feature. An asterisk (*) indicates that the translation is occurring in the fast path. The first packet in a conversation always goes through the slow path (that is, it is process switched). The remaining packets go through the fast path if a cache entry exists.
s=192.168.1.95->172.31.233.209	Source address of the packet and how it is being translated.
d=172.31.2.132	Destination address of the packet.
[6825]	IP identification number of the packet. Might be useful in the debugging process to correlate with other packet traces from protocol analyzers.

The following is sample output from the **debug ip nat detailed** command. In this example, the first two lines show the debugging output produced by a DNS request and reply. The remaining lines show the debugging output from a Telnet connection from a host on the inside of the network to a host on the outside of the network. In this example, the inside host 192.168.1.95 was assigned the global address 172.31.233.193.

```
Router# debug ip nat detailed

NAT: i: udp (192.168.1.95, 1493) -> (172.31.2.132, 53) [22399]
NAT: o: udp (172.31.2.132, 53) -> (172.31.233.193, 1493) [63671]
NAT*: i: tcp (192.168.1.95, 1135) -> (172.31.2.75, 23) [22400]
NAT*: o: tcp (172.31.2.75, 23) -> (172.31.233.193, 1135) [22002]
NAT*: i: tcp (192.168.1.95, 1135) -> (172.31.2.75, 23) [22401]
NAT*: i: tcp (192.168.1.95, 1135) -> (172.31.2.75, 23) [22402]
NAT*: o: tcp (172.31.2.75, 23) -> (172.31.233.193, 1135) [22060]
NAT*: o: tcp (172.31.2.75, 23) -> (172.31.233.193, 1135) [22071]
```

The following is sample output from the **debug ip nat h323** command. In this example, an H.323 call is established between two hosts, one host on the inside and the other one on the outside. The debug displays the H.323 messages names that NAT recognizes and the embedded IP addresses contained in those messages.

```
Router# debug ip nat h323

NAT:H225:[0] processing a Setup message
NAT:H225:[0] found Setup sourceCallSignalling
NAT:H225:[0] fix TransportAddress addr=192.168.122.50 port=11140
NAT:H225:[0] found Setup fastStart
NAT:H225:[0] Setup fastStart PDU length:18
NAT:H245:[0] processing OpenLogicalChannel message, forward channel
number 1
NAT:H245:[0] found OLC forward mediaControlChannel
NAT:H245:[0] fix TransportAddress addr=192.168.122.50 port=16517
NAT:H225:[0] Setup fastStart PDU length:29
NAT:H245:[0] processing OpenLogicalChannel message, forward channel
number 1
NAT:H245:[0] found OLC reverse mediaChannel
NAT:H245:[0] fix TransportAddress addr=192.168.122.50 port=16516
NAT:H245:[0] found OLC reverse mediaControlChannel
NAT:H245:[0] fix TransportAddress addr=192.168.122.50 port=16517
NAT:H225:[1] processing an Alerting message
NAT:H225:[1] found Alerting fastStart
NAT:H225:[1] Alerting fastStart PDU length:25
NAT:H245:[1] processing OpenLogicalChannel message, forward channel
```

Table 102 describes the significant fields shown in the display.

Table 102 debug ip nat h323 Field Descriptions

Field	Description
NAT:	Indicates that the packet is being translated by the NAT feature.
H.225 and H.245:	Protocol of the packet.
[1]	Indicates that the packet is moving from a host inside the network to one outside the network.
[0]	Indicates that the packet is moving from a host outside the network to one inside the network.

The following is sample output from the **debug ip nat ipsec** command:

```
Router# debug ip nat ipsec

5d21h:NAT:new IKE going In->Out, source addr 192.168.122.35, destination addr
192.168.22.20, initiator cookie
0x9C42065D
5d21h:NAT:IPSec:created In->Out ESP translation IL=192.168.122.35 SPI=0xAAE32A0A,
IG=192.168.22.40, OL=192.168.22.20,
OG=192.168.22.20
5d21h:NAT:IPSec:created Out->In ESP translation OG=192.168.22.20 SPI=0xA64B5BB6,
OL=192.168.22.20, IG=192.168.22.40,
IL=192.168.122.35

5d21h:NAT:new IKE going In->Out, source addr 192.168.122.20, destination addr
192.168.22.20, initiator cookie
0xC91738FF
5d21h:NAT:IPSec:created In->Out ESP translation IL=192.168.122.20 SPI=0x3E2E1B92,
IG=192.168.22.40, OL=192.168.22.20,
```

```

OG=192.168.22.20
5d21h:NAT:IPSec:Inside host (IL=192.168.122.20) trying to open an ESP connection to
Outside host (OG=192.168.22.20),
wait for Out->In reply
5d21h:NAT:IPSec:created Out->In ESP translation OG=192.168.22.20 SPI=0x1B201366,
OL=192.168.22.20, IG=192.168.22.40,
IL=192.168.122.20

```

The following is sample output from the **debug ip nat sip** command. In this example, one IP phone registers with a Cisco SIP proxy and then calls another IP phone. The debug output displays the SIP messages that NAT recognizes and the embedded IP addresses contained in those messages.

```

Router# debug ip nat sip

NAT:SIP:[0] processing REGISTER message
NAT:SIP:[0] translated embedded address
192.168.122.3->2.2.2.2
NAT:SIP:[0] translated embedded address
192.168.122.3->2.2.2.2
NAT:SIP:[0] message body found
NAT:SIP:[0] found address/port in SDP body:192.168.122.20
20332
NAT:SIP:[1] processing SIP/2.0 100 Trying reply message
NAT:SIP:[1] translated embedded address
2.2.2.2->192.168.122.3
NAT:SIP:[1] processing SIP/2.0 200 OK reply message
NAT:SIP:[1] translated embedded address
2.2.2.2->192.168.122.3
NAT:SIP:[1] translated embedded address
2.2.2.2->192.168.122.3
NAT:SIP:[1] processing INVITE message
NAT:SIP:[1] translated embedded address
2.2.2.2->192.168.122.3
NAT:SIP:[1] message body found
NAT:SIP:[1] found address/port in SDP body:192.168.22.20

```

[Table 103](#) describes the significant fields shown in the display.

Table 103 *debug ip nat sip Field Descriptions*

Field	Description
NAT:	Indicates that the packet is being translated by the NAT feature.
SIP:	Protocol of the packet.
[1]	Indicates that the packet is moving from a host inside the network to one outside the network.
[0]	Indicates that the packet is moving from a host outside the network to one inside the network.

The following is sample output from the **debug ip nat vrf** command:

```

Router# debug ip nat vrf

6d00h:NAT:address not stolen for 192.168.121.113, proto 1 port 7224
6d00h:NAT:creating portlist proto 1 globaladdr 2.2.2.10
6d00h:NAT:Allocated Port for 192.168.121.113 -> 2.2.2.10:wanted 7224 got 7224
6d00h:NAT:i:icmp (192.168.121.113, 7224) -> (168.58.88.2, 7224) [2460]
6d00h:NAT:s=192.168.121.113->2.2.2.10, d=168.58.88.2 [2460] vrf=> shop

```

```
6d00h:NAT*:o:icmp (168.58.88.2, 7224) -> (2.2.2.10, 7224) [2460] vrf=> shop
6d00h:NAT*:s=168.58.88.2, d=2.2.2.10->192.168.121.113 [2460] vrf=> shop
```

```
6d00h:NAT:Allocated Port for 192.168.121.113 -> 2.2.2.10:wanted 7225 got 7225
6d00h:NAT:i:icmp (192.168.121.113, 7225) -> (168.58.88.2, 7225) [2461]
6d00h:NAT:s=192.168.121.113->2.2.2.10, d=168.58.88.2 [2461] vrf=> shop
6d00h:NAT*:o:icmp (168.58.88.2, 7225) -> (2.2.2.10, 7225) [2461] vrf=> shop
6d00h:NAT*:s=168.58.88.2, d=2.2.2.10->192.168.121.113 [2461] vrf=> shop
6d00h:NAT:Allocated Port for 192.168.121.113 -> 2.2.2.10:wanted 7226 got 7226
6d00h:NAT:i:icmp (192.168.121.113, 7226) -> (168.58.88.2, 7226) [2462]
6d00h:NAT:s=192.168.121.113->2.2.2.10, d=168.58.88.2 [2462] vrf=> shop
```

Table 104 describes the significant fields shown in the display.

Table 104 debug ip nat vrf Field Descriptions

Field	Description
vrf=>	Indicates NAT is applied to a particular VPN.

debug ip ospf events

To display information on Open Shortest Path First (OSPF)-related events, such as adjacencies, flooding information, designated router selection, and shortest path first (SPF) calculation, use the **debug ip ospf events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip ospf events

no debug ip ospf events

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Examples The following is sample output from the **debug ip ospf events** command:

```
Router# debug ip ospf events

OSPF:hello with invalid timers on interface Ethernet0
hello interval received 10 configured 10
net mask received 255.255.255.0 configured 255.255.255.0
dead interval received 40 configured 30
```

The **debug ip ospf events** output shown might appear if any of the following situations occurs:

- The IP subnet masks for routers on the same network do not match.
- The OSPF hello interval for the router does not match that configured for a neighbor.
- The OSPF dead interval for the router does not match that configured for a neighbor.

If a router configured for OSPF routing is not seeing an OSPF neighbor on an attached network, perform the following tasks:

- Make sure that both routers have been configured with the same IP mask, OSPF hello interval, and OSPF dead interval.
- Make sure that both neighbors are part of the same area type.

In the following example line, the neighbor and this router are not part of a stub area (that is, one is a part of a transit area and the other is a part of a stub area, as explained in RFC 1247):

```
OSPF: hello packet with mismatched E bit
```

Related Commands

Command	Description
debug ip pgm host	Displays information about each OSPF packet received.

debug ip ospf mpls traffic-eng advertisements

To print information about traffic engineering advertisements in Open Shortest Path First (OSPF) link state advertisement (LSA) messages, use the **debug ip ospf mpls traffic-eng advertisements** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip ospf mpls traffic-eng advertisements

no debug ip ospf mpls traffic-eng advertisements

Syntax Description This command has no arguments or keywords

Defaults No default behavior or values.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.0(5)ST	This command was introduced.

Examples In the following example, information about traffic engineering advertisements is printed in OSPF LSA messages:

```
Router# debug ip ospf mpls traffic-eng advertisements

OSPF:IGP delete router node 10.106.0.6 fragment 0 with 0 links
      TE Router ID 10.106.0.6
OSPF:IGP update router node 10.110.0.10 fragment 0 with 0 links
      TE Router ID 10.110.0.10
OSPF:MPLS announce router node 10.106.0.6 fragment 0 with 1 links
      Link connected to Point-to-Point network
      Link ID :10.110.0.10
      Interface Address :10.1.0.6
      Neighbor Address :10.1.0.10
      Admin Metric :10
      Maximum bandwidth :1250000
      Maximum reservable bandwidth :625000
      Number of Priority :8
      Priority 0 :625000      Priority 1 :625000
      Priority 2 :625000      Priority 3 :625000
      Priority 4 :625000      Priority 5 :625000
      Priority 6 :625000      Priority 7 :625000
      Affinity Bit :0x0
```

Table 105 describes the significant fields shown in the display.

Table 105 *debug ip ospf mpls traffic-eng advertisements Field Descriptions*

Field	Description
Link ID	Index of the link being described.
Interface Address	Address of the interface.
Neighbor Address	Address of the neighbor.
Admin Metric	Administrative weight associated with this link.
Maximum bandwidth	Bandwidth capacity of the link (kbps).
Maximum reservable bandwidth	Amount of reservable bandwidth on this link.
Number of Priority	Number of priority levels for which bandwidth is advertised.
Priority	Bandwidth available at indicated priority level.
Affinity Bit	Attribute flags of the link that are being flooded.

debug ip ospf packet

To display information about each Open Shortest Path First (OSPF) packet received, use the **debug ip ospf packet** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip ospf packet

no debug ip ospf packet

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Examples The following is sample output from the **debug ip ospf packet** command:

```
Router# debug ip ospf packet

OSPF: rcv. v:2 t:1 l:48 rid:200.0.0.117
      aid:0.0.0.0 chk:6AB2 aut:0 auk:
```

The **debug ip ospf packet** command produces one set of information for each packet received. The output varies slightly depending on which authentication is used. The following is sample output from the **debug ip ospf packet** command when message digest algorithm 5 (MD5) authentication is used.

```
Router# debug ip ospf packet

OSPF: rcv. v:2 t:1 l:48 rid:200.0.0.116
      aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x0
```

[Table 106](#) describes the significant fields shown in the display.

Table 106 *debug ip ospf packet Field Descriptions*

Field	Description
v:	OSPF version.
t:	OSPF packet type. Possible packet types follow: <ul style="list-style-type: none"> • 1—Hello • 2—Data description • 3—Link state request • 4—Link state update • 5—Link state acknowledgment
l:	OSPF packet length in bytes.
rid:	OSPF router ID.
aid:	OSPF area ID.
chk:	OSPF checksum.

Table 106 *debug ip ospf packet Field Descriptions (continued)*

Field	Description
aut:	OSPF authentication type. Possible authentication types follow: <ul style="list-style-type: none"> • 0—No authentication • 1—Simple password • 2—MD5
keyid:	MD5 key ID.
seq:	Sequence number.

Related Commands

Command	Description
debug ip ospf events	Displays information on OSPF-related events, such as adjacencies, flooding information, designated router selection, and SPF calculation.

debug ip ospf spf statistic

To display statistical information while running the shortest path first (SPF) algorithm, use the **debug ip ospf spf statistic** command in privileged EXEC mode. To disable the debugging output, use the **no** form of this command.

debug ip ospf spf statistic

no debug ip ospf spf statistic

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.2(12)	This command was introduced.

Usage Guidelines The **debug ip ospf spf statistic** command displays the SPF calculation times in milliseconds, the node count, and a time stamp.

Examples The following is sample output from the **debug ip ospf spf statistic** command:

```
Router# debug ip ospf spf statistic

00:05:59:OSPF:Begin SPF at 359.216ms, process time 60ms
00:05:59:spf_time 00:05:59.216, wait_interval 0s
00:05:59:OSPF:End SPF at 359.216ms, Total elapsed time 0ms
00:05:59:Intra: 0ms, Inter: 0ms, External: 0ms
00:05:59:R: 4, N: 2, Stubs: 1
00:05:59:SN: 1, SA: 0, X5: 1, X7: 0
00:05:59:SPF suspends: 0 intra, 1 total
```

[Table 107](#) describes the significant fields shown in the display.

Table 107 *debug ip ospf spf statistic Field Descriptions*

Field	Description
Begin SPF at	Absolute time in milliseconds when SPF is started.
process time	Cumulative time since the process has been created.
spf_time	Last time SPF was run or an event has happened to run SPF.
wait_interval	Time waited to run SPF.
End SPF at	Absolute time in milliseconds when SPF had ended.
Total elapsed time	Total time take to run SPF.
Intra:	Time taken to process intra-area link-state advertisements (LSAs).

Table 107 *debug ip ospf spf statistic Field Descriptions (continued)*

Field	Description
Inter:	Time taken to process interarea LSAs.
External:	Time taken to process external LSAs.
R:	Number of router LSAs.
N:	Number of network LSAs.
Stubs:	Number of stub links.
SN:	Number of summary network LSAs.
SA:	Number of summary LSAs describing autonomous system boundary routers (ASBRs).
X5:	Number of external type 5 LSAs.
X7:	Number of external type 7 LSAs.
SPF suspends: intra	Number of times process is suspended during intra-area SPF run.
total	Total number of times process is suspended during SPF run.

debug ip packet

To display general IP debugging information and IP security option (IPSO) security transactions, use the **debug ip packet** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip packet [*access-list-number*] [**detail**] [**dump**]

no debug ip packet [*access-list-number*]

Syntax Description

<i>access-list-number</i>	(Optional) The IP access list number that you can specify. If the datagram is not permitted by that access list, the related debugging output is suppressed. Standard, extended, and expanded access lists are supported. The range of standard and extended access lists is from 1 to 199. The range of expanded access lists is from 1300 to 2699.
detail	(Optional) Displays detailed IP packet debugging information. This information includes the packet types and codes as well as source and destination port numbers.
dump	(Hidden) Displays IP packet debugging information along with raw packet data in hexadecimal and ASCII forms. This keyword can be enabled with individual access lists and also with the detail keyword. Note The dump keyword is not fully supported and should be used only in collaboration with Cisco Technical Support. See the caution notes below, in the usage guidelines, for more specific information.

Command Modes

Privileged EXEC

Usage Guidelines

If a communication session is closing when it should not be, an end-to-end connection problem can be the cause. The **debug ip packet** command is useful for analyzing the messages traveling between the local and remote hosts. IP packet debugging captures the packets that are process switched including received, generated and forwarded packets. IP packets that are switched in the fast path are not captured.

IPSO security transactions include messages that describe the cause of failure each time a datagram fails a security test in the system. This information is also sent to the sending host when the router configuration allows it.



Caution

Because the **debug ip packet** command generates a substantial amount of output and uses a substantial amount of system resources, this command should be used with caution in production networks. It should only be enabled when traffic on the IP network is low, so other activity on the system is not adversely affected. Enabling the **detail** and **dump** keywords use the highest level of system resources of the available configuration options for this command, so a high level of caution should be applied when enabling either of these keywords.

**Caution**

The **dump** keyword is not fully supported and should be used only in collaboration with Cisco Technical Support. Because of the risk of using significant CPU utilization, the **dump** keyword is hidden from the user and cannot be seen using the “?” prompt. The length of the displayed packet information may exceed the actual packet length and include additional padding bytes that do not belong to the IP packet. Also note that the beginning of a packet may start at different locations in the dump output depending on the specific router, interface type, and packet header processing that may have occurred before the output is displayed.

Examples

The following is sample output from the **debug ip packet** command:

debug ip packet

IP packet debugging is on

```
IP: s=172.69.13.44 (Fddi0), d=10.125.254.1 (Serial2), g=172.69.16.2, forward
IP: s=172.69.1.57 (Ethernet4), d=10.36.125.2 (Serial2), g=172.69.16.2, forward
IP: s=172.69.1.6 (Ethernet4), d=255.255.255.255, rcvd 2
IP: s=172.69.1.55 (Ethernet4), d=172.69.2.42 (Fddi0), g=172.69.13.6, forward
IP: s=172.69.89.33 (Ethernet2), d=10.130.2.156 (Serial2), g=172.69.16.2, forward
IP: s=172.69.1.27 (Ethernet4), d=172.69.43.126 (Fddi1), g=172.69.23.5, forward
IP: s=172.69.1.27 (Ethernet4), d=172.69.43.126 (Fddi0), g=172.69.13.6, forward
IP: s=172.69.20.32 (Ethernet2), d=255.255.255.255, rcvd 2
IP: s=172.69.1.57 (Ethernet4), d=10.36.125.2 (Serial2), g=172.69.16.2, access denied
```

The output shows two types of messages that the **debug ip packet** command can produce; the first line of output describes an IP packet that the router forwards, and the third line of output describes a packet that is destined for the router. In the third line of output, **rcvd 2** indicates that the router decided to receive the packet.

[Table 108](#) describes the significant fields shown in the output.

Table 108 *debug ip packet Field Descriptions*

Field	Description
IP:	Indicates that this is an IP packet.
s=172.69.13.44 (Fddi0)	Indicates the source address of the packet and the name of the interface that received the packet.
d=10.125.254.1 (Serial2)	Indicates the destination address of the packet and the name of the interface (in this case, S2) through which the packet is being sent out on the network.
g=172.69.16.2	Indicates the address of the next-hop gateway.
forward	Indicates that the router is forwarding the packet. If a filter denies a packet, “access denied” replaces “forward,” as shown in the last line of output.

The following is sample output from the **debug ip packet** command enabled with the **detail** keyword:

debug ip packet detail

IP packet debugging is on (detailed)

```
001556: 19:59:30: CEF: Try to CEF switch 10.4.9.151 from FastEthernet0/0
```

```

001557: 19:59:30: IP: s=10.4.9.6 (FastEthernet0/0), d=10.4.9.151 (FastEthernet03
001558: 19:59:30:     TCP src=179, dst=11001, seq=3736598846, ack=2885081910, wH
001559: 20:00:09: CEF: Try to CEF switch 10.4.9.151 from FastEthernet0/0
001560: 20:00:09: IP: s=10.4.9.4 (FastEthernet0/0), d=10.4.9.151 (FastEthernet03
001561: 20:00:09:     TCP src=179, dst=11000, seq=163035693, ack=2948141027, wiH
001562: 20:00:14: CEF: Try to CEF switch 10.4.9.151 from FastEthernet0/0
001563: 20:00:14: IP: s=10.4.9.6 (FastEthernet0/0), d=10.4.9.151 (FastEthernet03
001564: 20:00:14:     ICMP type=8, code=0
001565: 20:00:14: IP: s=10.4.9.151 (local), d=10.4.9.6 (FastEthernet0/0), len 1g
001566: 20:00:14:     ICMP type=0, code=0
    
```

The format of the output with **detail** keyword provides additional information, such as the packet type, code, some field values, and source and destination port numbers.

Table 109 describes the significant fields shown in the output.

Table 109 debug ip packet detail Field Descriptions

Field	Description
CEF:	Indicates that the IP packet is being processed by CEF.
IP:	Indicates that this is an IP packet.
s=10.4.9.6 (FastEthernet0/0)	Indicates the source address of the packet and the name of the interface that received the packet.
d=10.4.9.151 (FastEthernet03)	Indicates the destination address of the packet and the name of the interface through which the packet is being sent out on the network.
TCP src=	Indicates the source TCP port number.
dst=	Indicates the destination TCP port number.
seq=	Value from the TCP packet sequence number field./
ack=	Value from the TCP packet acknowledgement field.
ICMP type=	Indicates ICMP packet type.
code=	Indicates ICMP return code.

The following is sample output from the **debug ip packet** command enabled with the **dump** keyword:

```
debug ip packet dump
```

```

IP packet debugging is on (detailed) (dump)

21:02:42: IP: s=10.4.9.6 (FastEthernet0/0), d=10.4.9.4 (FastEthernet0/0), len 13
07003A00:                0005 00509C08                ...P..
07003A10: 0007855B 4DC00800 45000064 001E0000  ...[M@..E..d....
07003A20: FE019669 0A040906 0A040904 0800CF7C  ~..i.....|O|
07003A30: 0D052678 00000000 0A0B7145 ABCDABCD  ..&x.....qE+M+M
07003A40: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
07003A50: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
07003A60: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
07003A70: ABCDABCD ABCDABCD ABCDABCD                +M+M+M+M+M+M
21:02:42: IP: s=10.4.9.4 (local), d=10.4.9.6 (FastEthernet0/0), len 100, sending
07003A00:                0005 00509C08                ...P..
07003A10: 0007855B 4DC00800 45000064 001E0000  ...[M@..E..d....
07003A20: FF019569 0A040904 0A040906 0000D77C  ...i.....|W|
07003A30: 0D052678 00000000 0A0B7145 ABCDABCD  ..&x.....qE+M+M
07003A40: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
07003A50: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
07003A60: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
    
```

```

07003A70: ABCDABCD ABCDABCD ABCDABCD          +M+M+M+M+M+M
21:02:42: CEF: Try to CEF switch 10.4.9.4 from FastEthernet0/0
21:02:42: IP: s=10.4.9.6 (FastEthernet0/0), d=10.4.9.4 (FastEthernet0/0), len 13
07003380:                0005 00509C08          ...P..
07003390: 0007855B 4DC00800 45000064 001F0000  ...[M@..E..d....
070033A0: FE019668 0A040906 0A040904 0800CF77  ~..h.....Ow
070033B0: 0D062678 00000000 0A0B7149 ABCDABCD  ..&x.....qI+M+M
070033C0: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M
070033D0: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M
070033E0: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M
070033F0: ABCDABCD ABCDABCD ABCDABCD          +M+M+M+M+M+M

```

**Note**

The **dump** keyword is not fully supported and should be used only in collaboration with Cisco Technical Support. See the caution in the usage guidelines section of this command reference page for more specific information.

The output from the **debug ip packet** command, when the **dump** keyword is enabled, provides raw packet data in hexadecimal and ASCII forms. This additional output is displayed in addition to the standard output. The **dump** keyword can be used with all of the available configuration options of this command.

Table 110 describes the standard output fields shown.

Table 110 *debug ip packet dump Field Descriptions*

Field	Description
IP:	Indicates that this is an IP packet.
s=10.4.9.6 (FastEthernet0/0)	Indicates the source address of the packet and the name of the interface that received the packet.
d=10.4.9.4 (FastEthernet0/0) len 13	Indicates destination address and length of the packet and the name of the interface through which the packet is being sent out on the network.
sending	Indicates that the router is sending the packet.

The calculation on whether to send a security error message can be somewhat confusing. It depends upon both the security label in the datagram and the label of the incoming interface. First, the label contained in the datagram is examined for anything obviously wrong. If nothing is wrong, assume the datagram to be correct. If something is wrong, the datagram is treated as *unclassified genser*. Then the label is compared with the interface range, and the appropriate action is taken, as Table 111 describes.

Table 111 *Security Actions*

Classification	Authorities	Action Taken
Too low	Too low	No Response
	Good	No Response
	Too high	No Response

Table 111 Security Actions (continued)

Classification	Authorities	Action Taken
In range	Too low	No Response
	Good	Accept
	Too high	Send Error
Too high	Too low	No Response
	In range	Send Error
	Too high	Send Error

The security code can only generate a few types of Internet Control Message Protocol (ICMP) error messages. The only possible error messages and their meanings follow:

- ICMP Parameter problem, code 0—Error at pointer
- ICMP Parameter problem, code 1—Missing option
- ICMP Parameter problem, code 2—See Note that follows
- ICMP Unreachable, code 10—Administratively prohibited

**Note**

The message “ICMP Parameter problem, code 2” identifies a specific error that occurs in the processing of a datagram. This message indicates that the router received a datagram containing a maximum length IP header but no security option. After being processed and routed to another interface, it is discovered that the outgoing interface is marked with “add a security label.” Because the IP header is already full, the system cannot add a label and must drop the datagram and return an error message.

When an IP packet is rejected due to an IP security failure, an audit message is sent via Department of Defense Intelligence Information System Network Security for Information Exchange (DNSIX) Network Address Translation (NAT). Also, any **debug ip packet** output is appended to include a description of the reason for rejection. This description can be any of the following:

- No basic
- No basic, no response
- Reserved class
- Reserved class, no response
- Class too low, no response
- Class too high
- Class too high, bad authorities, no response
- Unrecognized class
- Unrecognized class, no response
- Multiple basic
- Multiple basic, no response
- Authority too low, no response
- Authority too high
- Compartment bits not dominated by maximum sensitivity level

- Compartment bits do not dominate minimum sensitivity level
- Security failure: extended security disallowed
- NLESO source appeared twice
- ESO source not found
- Postroute, failed xfc out
- No room to add IPSO

debug ip pgm host

To display debug messages for the Pragmatic General Multicast (PGM) Host feature, use the **debug ip pgm host** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug ip pgm host [data | nak | spm]
```

```
no debug ip pgm host [data | nak | spm]
```

Syntax Description

data	(Optional) Enables debugging for PGM sent (ODATA) and re-sent (RDATA) data packets.
nak	(Optional) Enables debugging for PGM negative acknowledgment (NAK) data packets, NAK confirmation (NCF) data packets, and Null NAK (NNAK) data packets.
spm	(Optional) Enables debugging for PGM source path messages (SPMs).

Defaults

Debugging for PGM Host is not enabled. If the **debug ip pgm host** command is used with no additional keywords, debugging is enabled for all PGM Host message types.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)T	This command was introduced.

Examples

The following is sample output from the **debug ip pgm host** command:

```
Router# debug ip pgm host

Host SPM debugging is on
Host NAK/NCF debugging is on
Host ODATA/RDATA debugging is on
```

The following is sample output from the **debug ip pgm host** command when the **data** keyword is used:

```
Router# debug ip pgm host data

02:50:23:PGM Host:Received ODATA from 10.0.30.2 to 224.3.3.3 (74 bytes)
02:50:23:      ODATA TSI 00000A001E02-0401 data-dport BBBB csum 9317 tlen 74
02:50:23:      tsqn          31 dsqn          39
```

The following example shows output of the **debug ip pgm host** command when the **nak** keyword is used. In the following example, the host sends a NAK to the source for a missing packet and the source returns an NCF to the host followed by an RDATA data packet.

```
Router# debug ip pgm host nak

02:50:24:PGM Host:Sending NAK from 10.0.32.2 to 10.0.32.1 (36 bytes)
02:50:24:    NAK TSI 0000A001E02-0401 data-dport BBBB csum 04EC tlen 36
02:50:24:    dsqn          38 data source 10.0.30.2 group 224.3.3.3

02:50:24:PGM Host:Received NCF from 10.0.30.2 to 224.3.3.3 (36 bytes)
02:50:24:    NCF TSI 0000A001E02-0401 data-dport BBBB csum 02EC tlen 36
02:50:24:    dsqn          38 data source 10.0.30.2 group 224.3.3.3

02:50:24:PGM Host:Received RDATA from 10.0.30.2 to 224.3.3.3 (74 bytes)
02:50:24:    RDATA TSI 0000A001E02-0401 data-dport BBBB csum 9218 tlen 74
02:50:24:    tsqn          31 dsqn          38
```

The following is sample output from the **debug ip pgm host** command with the **spm** keyword is used:

```
Router# debug ip pgm host spm

02:49:39:PGM Host:Received SPM from 10.0.30.2 to 224.3.3.3 (36 bytes)
02:49:39:    SPM TSI 0000A001E02-0401 data-dport BBBB csum EA08 tlen 36
02:49:39:    dsqn          980 tsqn          31 lsqn          31  NLA 10.0.32.1
```

Related Commands

Command	Description
clear ip pgm host	Resets PGM Host connections to their default values and clears traffic statistics.
ip pgm host	Enables the PGM Host feature.
show ip pgm host defaults	Displays the default values for PGM Host traffic.
show ip pgm host sessions	Displays open PGM Host traffic sessions.
show ip pgm host traffic	Displays PGM Host traffic statistics.

debug ip pgm router

To display debug messages for Pragmatic General Multicast (PGM), use the **debug ip pgm router** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip pgm router [**spm** | **nak** | **data**]

no debug ip pgm router [**spm** | **nak** | **data**]

Syntax Description

spm	(Optional) Enables debugging for Source Path Messages (SPMs).
nak	(Optional) Enables debugging for negative acknowledgments (NAKs), NAK confirmations (NCFs), and Null NAKs (NNAKs).
data	(Optional) Enables debugging for Retransmissions (RDATA).

Defaults

Debugging for PGM is not enabled. If the **debug ip pgm router** command is used with no additional keywords, debugging is enabled for all PGM message types.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.

Examples

The following shows sample output from the **debug ip pgm router** command:

```
Router# debug ip pgm router
```

```
SPM debugging is on
NAK/NNAK/NCF debugging is on
RDATA debugging is on
```

The following shows sample output from the **debug ip pgm router** command when the **spm** keyword is used:

```
Router# debug ip pgm router spm
```

```
PGM: Received SPM on Ethernet1/0/5 from 10.7.0.200 to 227.7.7.7 (52 bytes)
SPM TSI 0A0700C85555-1000 data-dport 1001 csum CCCC tlen 52
dsqn 3758096779 tsqn      1954 isqn      1979 lsqn      1990
NLA 10.7.0.200
SPM from source/RPF-neighbour 10.7.0.200 for 10.7.0.200 (SPT)
Forwarded SPM from 10.7.0.200 to 227.7.7.7
```

The following is a debugging message for a selective SPM:

```
Router# debug ip pgm router spm
```

```
PGM: Received SPM on Ethernet1/0/5 from 10.7.0.200 to 234.4.3.2 (52 bytes)
      SPM TSI 0A0700C85555-2000 data-dport 2001 csum CCCC tlen 52 Options P N O
      dsqn 3758096768 tsqn          1986 isqn          1994 lsqn          2006
      NLA 10.7.0.200
      SPM from source/RPF-neighbour 10.7.0.200 for 10.7.0.200 (SPT)
      Forwarded SPM from 10.7.0.200 to 227.7.7.7
```

The “P N O” flags indicate which options are present in this packet:

- P indicates that this is a parity packet.
- N indicates that options are network significant.
- O indicates that options are present.

The following shows sample output from the **debug ip pgm router** command when the **nak** keyword is used:

```
Router# debug ip pgm router nak
```

```
PGM: Received NAK on Ethernet1/0/0 from 10.1.0.4 to 10.1.0.2 (36 bytes)
      NAK TSI 0A0700C85555-1000 data-dport 1001 csum CCCC tlen 36
      dsqn          1990 data source 10.7.0.200 group 227.7.7.7
      NAK unicast routed to RPF neighbour 10.4.0.1
      Forwarding NAK from 10.1.0.4 to 10.4.0.1 for 10.7.0.200
PGM: Received NCF on Ethernet1/0/5 from 10.7.0.200 to 227.7.7.7 (36 bytes)
      NCF TSI 0A0700C85555-1000 data-dport 1001 csum CACC tlen 36
      dsqn          1990 data source 10.7.0.200 group 227.7.7.7
      NAK retx canceled for TSI 0A0700C85555-1000 dsqn          1990
      NAK elimination started for TSI 0A0700C85555-1000 dsqn          1990
PGM: Received NCF on Ethernet1/0/5 from 10.7.0.200 to 227.7.7.7 (36 bytes)
      NCF TSI 0A0700C85555-1000 data-dport 1001 csum CACC tlen 36
      dsqn          1991 data source 10.7.0.200 group 227.7.7.7
      No NAK retx outstanding for TSI 0A0700C85555-1000 dsqn          1991
      NAK anticipated for TSI 0A0700C85555-1000 dsqn          1991
```

The following example shows output of the **debug ip pgm router** command with the **data** keyword. The debugging message is for an RDATA packet for which the router has only anticipated state, sqn 1991. Because it did not actually get a NAK, this RDATA is not forwarded by the PGM router.

```
Router# debug ip pgm router data
```

```
PGM: Received RDATA on Ethernet1/0/5 from 10.7.0.200 to 227.7.7.7 (70 bytes)
      RDATA TSI 0A0700C85555-1000 data-dport 1001 csum CCCC tlen 32
      tsqn          1954 dsqn          1990
      Marking Ethernet1/0/0 for forwarding
      Marking Serial5/0 for skipping
      Forwarded RDATA from 10.7.0.200 to 227.7.7.7
```

Debug message for RDATA packet corresponding to a NAK for sqn 1990. Since the NAK was received on Ethernet1/0/0, RDATA is forwarded out only that interface and another interface in the multicast olist Serial5/0 is skipped.

```
PGM: Received RDATA on Ethernet1/0/5 from 10.7.0.200 to 227.7.7.7 (70 bytes)
      RDATA TSI 0A0700C85555-1000 data-dport 1001 csum CCCC tlen 32
      tsqn          1954 dsqn          1991
      Eliminated RDATA (null oif) from 10.7.0.200 to 227.7.7.7
```

Related Commands

Command	Description
ip pgm router	Enables the PGM Router Assist feature for the interface.
show ip pgm router	Displays PGM traffic statistics and TSI state.

debug ip pim

To display Protocol Independent Multicast (PIM) packets received and sent, and to display PIM-related events, use the **debug ip pim** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug ip pim [vrf vrf-name] [group-address | atm | auto-rp | bsr | df [rp-address] | hello | tag]
```

```
no debug ip pim [vrf vrf-name] [group-address | atm | auto-rp | bsr | df [rp-address] | hello | tag]
```

Syntax Description

vrf <i>vrf-name</i>	(Optional) Displays PIM-related events associated with the Multicast Virtual Private Network (MVPN) routing and forwarding (MVRP) instance specified for the <i>vrf-name</i> argument.
<i>group-address</i>	(Optional) IP address or Domain Name System (DNS) name of a multicast group. Entering a multicast group address restricts the output to display only PIM-related events associated with the multicast group address specified for the optional <i>group-address</i> argument.
atm	(Optional) Displays PIM ATM signaling activity.
auto-rp	(Optional) Displays the contents of each PIM packet used in the automatic discovery of group-to-rendezvous point (RP) mapping and the actions taken on the address-to-RP mapping database.
bsr	(Optional) Displays candidate-RPs and Bootstrap Router (BSR) activity.
df	(Optional) When bidirectional PIM is used, displays all designated forwarder (DF) election messages.
<i>rp-address</i>	(Optional) The rendezvous point IP address.
hello	(Optional) Displays events associated with PIM hello messages.
tag	(Optional) Displays tagswitching-related activity.

Defaults

All PIM packets are displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
10.2	This command was introduced.
11.1	The auto-rp keyword was added.
11.3	The atm and tag keywords were added.
12.1(2)T	The df keyword was added.
12.1(3)T	The bsr keyword was added.
12.0(22)S	The vrf keyword, <i>vrf-name</i> argument, and hello keyword were added.
12.2(13)T	The vrf keyword and <i>vrf-name</i> argument were added.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	The hello keyword was added.

Release	Modification
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.

Usage Guidelines

PIM uses Internet Group Management Protocol (IGMP) packets to communicate with routers and advertise reachability information.

Use this command with the **debug ip igmp** and **debug ip mrouting** commands to display additional multicast routing information.

Examples

The following is sample output from the **debug ip pim** command:

```
Router# debug ip pim 224.2.0.1

PIM: Received Join/Prune on Ethernet1 from 172.16.37.33
PIM: Received Join/Prune on Ethernet1 from 172.16.37.33
PIM: Received Join/Prune on Tunnel0 from 10.3.84.1
PIM: Received Join/Prune on Ethernet1 from 172.16.37.33
PIM: Received Join/Prune on Ethernet1 from 172.16.37.33
PIM: Received RP-Reachable on Ethernet1 from 172.16.20.31
PIM: Update RP expiration timer for 224.2.0.1
PIM: Forward RP-reachability packet for 224.2.0.1 on Tunnel0
PIM: Received Join/Prune on Ethernet1 from 172.16.37.33
PIM: Prune-list (10.221.196.51/32, 224.2.0.1)
PIM: Set join delay timer to 2 seconds for (10.221.0.0/16, 224.2.0.1) on Ethernet1
PIM: Received Join/Prune on Ethernet1 from 172.16.37.6
PIM: Received Join/Prune on Ethernet1 from 172.16.37.33
PIM: Received Join/Prune on Tunnel0 from 10.3.84.1
PIM: Join-list: (*, 224.2.0.1) RP 172.16.20.31
PIM: Add Tunnel0 to (*, 224.2.0.1), Forward state
PIM: Join-list: (10.0.0.0/8, 224.2.0.1)
PIM: Add Tunnel0 to (10.0.0.0/8, 224.2.0.1), Forward state
PIM: Join-list: (10.4.0.0/16, 224.2.0.1)
PIM: Prune-list (172.16.84.16/28, 224.2.0.1) RP-bit set RP 172.16.84.16
PIM: Send Prune on Ethernet1 to 172.16.37.6 for (172.16.84.16/28, 224.2.0.1), RP
PIM: For RP, Prune-list: 10.9.0.0/16
PIM: For RP, Prune-list: 10.16.0.0/16
PIM: For RP, Prune-list: 10.49.0.0/16
PIM: For RP, Prune-list: 10.84.0.0/16
PIM: For RP, Prune-list: 10.146.0.0/16
PIM: For 10.3.84.1, Join-list: 172.16.84.16/28
PIM: Send periodic Join/Prune to RP via 172.16.37.6 (Ethernet1)
```

The following lines appear periodically when PIM is running in sparse mode and indicate to this router the multicast groups and multicast sources in which other routers are interested:

```
PIM: Received Join/Prune on Ethernet1 from 172.16.37.33
PIM: Received Join/Prune on Ethernet1 from 172.16.37.33
```

The following lines appear when a rendezvous point (RP) message is received and the RP timer is reset. The expiration timer sets a checkpoint to make sure the RP still exists. Otherwise, a new RP must be discovered.

```
PIM: Received RP-Reachable on Ethernet1 from 172.16.20.31
PIM: Update RP expiration timer for 224.2.0.1
PIM: Forward RP-reachability packet for 224.2.0.1 on Tunnel0
```

The prune message in the following line states that this router is not interested in the Source-Active (SA) information. This message tells an upstream router to stop forwarding multicast packets from this source. The address 10.221.196.51/32 indicates a host route with 32 bits of mask.

```
PIM: Prune-list (10.221.196.51/32, 224.2.0.1)
```

In the following line, a second router on the network wants to override the prune message that the upstream router just received. The timer is set at a random value so that if additional routers on the network still want to receive multicast packets for the group, only one will actually send the message. The other routers will receive the join message and then suppress sending their own message.

```
PIM: Set join delay timer to 2 seconds for (10.221.0.0/16, 224.2.0.1) on Ethernet1
```

In the following line, a join message is sent toward the RP for all sources:

```
PIM: Join-list: (*, 224.2.0.1) RP 172.16.20.31
```

In the following lines, the interface is being added to the outgoing interface (OIF) of the (*, G) and (S, G) multicast route (mroute) table entry so that packets from the source will be forwarded out that particular interface:

```
PIM: Add Tunnel0 to (*, 224.2.0.1), Forward state
PIM: Add Tunnel0 to (10.0.0.0/8, 224.2.0.1), Forward state
```

The following line appears in sparse mode only. There are two trees on which data may be received: the RP tree and the source tree. In dense mode there is no RP. After the source and the receiver have discovered one another at the RP, the first-hop router for the receiver will usually join to the source tree rather than the RP tree.

```
PIM: Prune-list (172.16.84.16/28, 224.2.0.1) RP-bit set RP 172.16.84.16
```

The send prune message in the next line shows that a router is sending a message to a second router saying that the first router should no longer receive multicast packets for the (S, G). The RP at the end of the message indicates that the router is pruning the RP tree and is most likely joining the source tree, although the router may not have downstream members for the group or downstream routers with members of the group. The output shows the specific sources from which this router no longer wants to receive multicast messages.

```
PIM: Send Prune on Ethernet1 to 172.16.37.6 for (172.16.84.16/28, 224.2.0.1), RP
```

The following lines indicate that a prune message is sent toward the RP so that the router can join the source tree rather than the RP tree:

```
PIM: For RP, Prune-list: 10.9.0.0/16
PIM: For RP, Prune-list: 10.16.0.0/16
PIM: For RP, Prune-list: 10.49.0.0/16
```

In the following line, a periodic message is sent toward the RP. The default period is once per minute. Prune and join messages are sent toward the RP or source rather than directly to the RP or source. It is the responsibility of the next hop router to take proper action with this message, such as continuing to forward it to the next router in the tree.

```
PIM: Send periodic Join/Prune to RP via 172.16.37.6 (Ethernet1)
```

Related Commands

Command	Description
debug ip dvmrp	Displays information on DVMRP packets received and sent.
debug ip igmp	Displays IGMP packets received and sent, and displays IGMP host-related events.
debug ip igrp transactions	Displays transaction information on IGRP routing transactions.
debug ip mrouting	Displays changes to the IP multicast routing table.
debug ip sd	Displays all SD announcements received.

debug ip pim atm

To log Protocol Independent Multicast (PIM) ATM signalling activity, use the **debug ip pim atm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip pim atm

no debug ip pim atm

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Examples

The following sample output shows a new group being created and the router toward the rendezvous point (RP) opening a new virtual circuit (VC). Because there are now two groups on this router, there are two VCs open, as reflected by the “current count.”

The following is sample output from the **debug ip pim atm** command:

```
Router# debug ip pim atm

Jan 28 19:05:51: PIM-ATM: Max VCs 200, current count 1
Jan 28 19:05:51: PIM-ATM: Send SETUP on ATM2/0 for 239.254.254.253/171.69.214.43
Jan 28 19:05:51: PIM-ATM: Received CONNECT on ATM2/0 for 239.254.254.253, vcd 19
Jan 28 19:06:35: PIM-ATM: Max VCs 200, current count 2
```

[Table 112](#) describes the significant fields shown in the display.

Table 112 *debug ip pim atm Field Descriptions*

Field	Description
Jan 28 19:05:51	Current date and time (in hours:minutes:seconds).
PIM-ATM	Indicates what PIM is doing to set up or monitor an ATM connection (vc).
current count	Current number of open virtual circuits.

The resulting **show ip mroute** output follows:

```
Router# show ip mroute 239.254.254.253

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.254.254.253), 00:00:04/00:02:53, RP 171.69.214.50, flags: S
  Incoming interface: Ethernet1/1, RPF nbr 171.69.214.50
  Outgoing interface list:
    ATM2/0, VCD 19, Forward/Sparse-Dense, 00:00:04/00:02:52
```

debug ip pim auto-rp

To display the contents of each Protocol Independent Multicast (PIM) packet used in the automatic discovery of group-to- rendezvous point (RP) mapping and the actions taken on the address-to-RP mapping database, use the **debug ip pim auto-rp** command in privileged EXEC. To disable debugging output, use the **no** form of this command.

debug ip pim auto-rp [*vrf vrf-name*]

no debug ip pim auto-rp [*vrf vrf-name*]

Syntax Description

vrf	(Optional) Supports the Multicast Virtual Private Network (VPN) routing and forwarding (VRF) instance.
<i>vrf-name</i>	(Optional) Name assigned to the VRF.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3	This command was introduced.
12.0(23)S	The vrf keyword and <i>vrf-name</i> argument were added.
12.2(13)T	The vrf keyword and <i>vrf-name</i> argument were added to Release 12.2T.

Examples

The following is sample output from the **debug ip pim auto-rp** command:

```
Router# debug ip pim auto-rp

Auto-RP: Received RP-announce, from 172.16.214.66, RP_cnt 1, holdtime 180 secs
Auto-RP:  update (192.168.248.0/24, RP:172.16.214.66)
Auto-RP: Build RP-Discovery packet
Auto-RP:  Build mapping (192.168.248.0/24, RP:172.16.214.66),
Auto-RP:  Build mapping (192.168.250.0/24, RP:172.16.214.26).
Auto-RP:  Build mapping (192.168.254.0/24, RP:172.16.214.2).
Auto-RP: Send RP-discovery packet (3 RP entries)
Auto-RP: Build RP-Announce packet for 172.16.214.2
Auto-RP:  Build announce entry for (192.168.254.0/24)
Auto-RP: Send RP-Announce packet, IP source 172.16.214.2, ttl 8
```

The first two lines show a packet received from 172.16.214.66 announcing that it is the RP for the groups in 192.168.248.0/24. This announcement contains one RP address and is valid for 180 seconds. The RP-mapping agent then updates its mapping database to include the new information.

```
Auto-RP: Received RP-announce, from 172.16.214.66, RP_cnt 1, holdtime 180 secs
Auto-RP:  update (192.168.248.0/24, RP:172.16.214.66)
```

In the next five lines, the router creates an RP-discovery packet containing three RP mapping entries. The packet is sent to the well-known CISCO-RP-DISCOVERY group address (224.0.1.40).

```
Auto-RP: Build RP-Discovery packet
Auto-RP: Build mapping (192.168.248.0/24, RP:172.16.214.66),
Auto-RP: Build mapping (192.168.250.0/24, RP:172.16.214.26).
Auto-RP: Build mapping (192.168.254.0/24, RP:172.16.214.2).
Auto-RP: Send RP-discovery packet (3 RP entries)
```

The final three lines show the router announcing that it intends to be an RP for the groups in 192.168.254.0/24. Only routers inside the scope “ttl 8” receive the advertisement and use the RP for these groups.

```
Auto-RP: Build RP-Announce packet for 172.16.214.2
Auto-RP: Build announce entry for (192.168.254.0/24)
Auto-RP: Send RP-Announce packet, IP source 172.16.214.2, ttl 8
```

The following is sample output from the **debug ip pim auto-rp** command when a router receives an update. In this example, the packet contains three group-to-RP mappings, which are valid for 180 seconds. The RP-mapping agent then updates its mapping database to include the new information.

```
Router# debug ip pim auto-rp

Auto-RP: Received RP-discovery, from 172.16.214.17, RP_cnt 3, holdtime 180 secs
Auto-RP: update (192.168.248.0/24, RP:172.16.214.66)
Auto-RP: update (192.168.250.0/24, RP:172.16.214.26)
Auto-RP: update (192.168.254.0/24, RP:172.16.214.2)
```

debug ip policy

To display IP policy routing packet activity, use the **debug ip policy** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip policy [*access-list-name*]

no debug ip policy [*access-list-name*]

Syntax Description

access-list-name (Optional) The name of the access list. Displays packets permitted by the access list that are policy routed in process level, Cisco Express Forwarding (CEF), and distributed CEF (DCEF) with NetFlow enabled or disabled.

If no access list is specified, information about all policy-matched and policy-routed packets is displayed.

Command Modes

Privileged EXEC

Command History

Release	Command
12.0(3)T	This command was introduced.

Usage Guidelines

After you configure IP policy routing with the **ip policy** and **route-map** commands, use the **debug ip policy** command to ensure that the IP policy is configured correctly.

Policy routing looks at various parts of the packet and then routes the packet based on certain user-defined attributes in the packet.

The **debug ip policy** command helps you determine what policy routing is following. It displays information about whether a packet matches the criteria, and if so, the resulting routing information for the packet.



Caution

Because the **debug ip policy** command generates a substantial amount of output, use it only when traffic on the IP network is low, so other activity on the system is not adversely affected.

Examples

The following is sample output of the **debug ip policy** command:

```
Router# debug ip policy 3

IP: s=30.0.0.1 (Ethernet0/0/1), d=40.0.0.7, len 100,FIB flow policy match
IP: s=30.0.0.1 (Ethernet0/0/1), d=40.0.0.7, len 100,FIB PR flow accelerated!
IP: s=30.0.0.1 (Ethernet0/0/1), d=40.0.0.7, g=10.0.0.8, len 100, FIB policy routed
```

Table 113 describes the significant fields shown in the display.

Table 113 *debug ip policy Field Descriptions*

Field	Description
IP: s=	IP source address and interface of the packet being routed.
d=	IP destination address of the packet being routed.
len	Length of the packet.
g=	IP gateway address of the packet being routed.

debug ip rgmp

To log debugging messages sent by a Router-Port Group Management Protocol (RGMP)-enabled router, use the **debug ip rgmp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug ip rgmp [group-name | group-address]
```

```
no debug ip rgmp
```

Syntax Description

<i>group-name</i>	(Optional) The name of a specific IP multicast group.
<i>group-address</i>	(Optional) The IP address of a specific IP multicast group.

Defaults

Debugging for RGMP is not enabled. If the **debug ip rgmp** command is used without arguments, debugging is enabled for all RGMP message types.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(10)S	This command was introduced.
12.1(1)E	The command was integrated into Cisco IOS Release 12.1(1)E.
12.1(5)T	The command was integrated into Cisco IOS Release 12.1(5)T.

Examples

The following example shows output for the **debug ip rgmp** command:

```
Router# debug ip rgmp

RGMP: Sending a Hello packet on Ethernet1/0

RGMP: Sending a Join packet on Ethernet1/0 for group 224.1.2.3

RGMP: Sending a Leave packet on Ethernet1/0 for group 224.1.2.3

RGMP: Sending a Bye packet on Ethernet1/0
```

Related Commands

Command	Description
ip rgmp	Enables the RGMP on IEEE 802.3 Ethernet interfaces.
show ip igmp interface	Displays multicast-related information about an interface.

debug ip rip

To display information on Routing Information Protocol (RIP) routing transactions, use the **debug ip rip** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rip

no debug ip rip

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Examples The following is sample output from the **debug ip rip** command:

```

router# debug ip rip
Updates
received from this source address
-----
RIP: received update from 10.89.80.28 on Ethernet0
      10.89.95.0 in 1 hops
      10.89.81.0 in 1 hops
      10.89.66.0 in 2 hops
      172.31.0.0 in 16 hops (inaccessible)
      0.0.0.0 in 7 hop
Updates
sent to these two destination addresses
-----
RIP: sending update to 255.255.255.255 via Ethernet0 (10.89.64.31)
      subnet 10.89.94.0, metric 1
      172.31.0.0 in 16 hops (inaccessible)
RIP: sending update to 255.255.255.255 via Serial1 (10.89.94.31)
      subnet 10.89.64.0, metric 1
      subnet 10.89.66.0, metric 3
      172.31.0.0 in 16 hops (inaccessible)
      default 0.0.0.0, metric 8

```

S24550

The output shows that the router being debugged has received updates from one router at source address 160.89.80.28. That router sent information about five destinations in the routing table update. Notice that the fourth destination address in the update—131.108.0.0—is inaccessible because it is more than 15 hops away from the router sending the update. The router being debugged also sent updates, in both cases to broadcast address 255.255.255.255 as the destination.

The second line is an example of a routing table update. It shows how many hops a given Internet address is from the router.

The entries show that the router is sending updates that are similar, except that the number in parentheses is the source address encapsulated into the IP header.

Examples of additional output that the **debug ip rip** command can generate follow.

Entries such as the following appear at startup or when an event occurs such as an interface making a transition or a user manually clearing the routing table:

```

RIP: broadcasting general request on Ethernet0
RIP: broadcasting general request on Ethernet1

```

An entry such as the following is most likely caused by a malformed packet from the sender:

```

RIP: bad version 128 from 160.89.80.43

```

debug ip routing

To display information on Routing Information Protocol (RIP) routing table updates and route cache updates, use the **debug ip routing** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip routing

no debug ip routing

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.2(13) T	Support for Interior Gateway Routing Protocol (IGRP) was removed.

Examples The following is sample output from the **debug ip routing** command:

```
Router# debug ip routing

RT: add 172.25.168.0 255.255.255.0 via 172.24.76.30, igrp metric [100/3020]
RT: metric change to 172.25.168.0 via 172.24.76.30, igrp metric [100/3020]
    new metric [100/2930]
IP: cache invalidation from 0x115248 0x1378A, new version 5736
RT: add 172.26.219.0 255.255.255.0 via 172.24.76.30, igrp metric [100/16200]
RT: metric change to 172.26.219.0 via 172.24.76.30, igrp metric [100/16200]
    new metric [100/10816]
RT: delete route to 172.26.219.0 via 172.24.76.30, igrp metric [100/10816]
RT: no routes to 172.26.219.0, entering holddown
IP: cache invalidation from 0x115248 0x1378A, new version 5737
RT: 172.26.219.0 came out of holddown
RT: garbage collecting entry for 172.26.219.0
IP: cache invalidation from 0x115248 0x1378A, new version 5738
RT: add 172.26.219.0 255.255.255.0 via 172.24.76.30, igrp metric [100/10816]
RT: delete route to 172.26.219.0 via 172.24.76.30, igrp metric [100/10816]
RT: no routes to 172.26.219.0, entering holddown
IP: cache invalidation from 0x115248 0x1378A, new version 5739
RT: 172.26.219.0 came out of holddown
RT: garbage collecting entry for 172.26.219.0
IP: cache invalidation from 0x115248 0x1378A, new version 5740
RT: add 172.26.219.0 255.255.255.0 via 172.24.76.30, igrp metric [100/16200]
RT: metric change to 172.26.219.0 via 172.24.76.30, igrp metric [100/16200]
    new metric [100/10816]
RT: delete route to 172.26.219.0 via 172.24.76.30, igrp metric [100/10816]
RT: no routes to 172.26.219.0, entering holddown
IP: cache invalidation from 0x115248 0x1378A, new version 5741
```

In the following lines, a newly created entry has been added to the IP routing table. The “metric change” indicates that this entry existed previously, but its metric changed and the change was reported by means of IGRP. The metric could also be reported via RIP, OSPF, or another IP routing protocol. The numbers inside the brackets report the administrative distance and the actual metric.

```
RT: add 172.25.168.0 255.255.255.0 via 172.24.76.30, igrp metric [100/3020]
RT: metric change to 172.25.168.0 via 172.24.76.30, igrp metric [100/3020]
    new metric [100/2930]
IP: cache invalidation from 0x115248 0x1378A, new version 5736
```

“Cache invalidation” means that the fast-switching cache was invalidated due to a routing table change. “New version” is the version number of the routing table. When the routing table changes, this number is incremented. The hexadecimal numbers are internal numbers that vary from version to version and software load to software load.

In the following output, the “holddown” and “cache invalidation” lines are displayed. Most of the distance vector routing protocols use “holddown” to avoid typical problems like counting to infinity and routing loops. If you look at the output of the **show ip protocols** command you will see the timer values for “holddown” and “cache invalidation.” “Cache invalidation” corresponds to “came out of holddown.” “Delete route” is triggered when a better path appears. It removes the old inferior path.

```
RT: delete route to 172.26.219.0 via 172.24.76.30, igrp metric [100/10816]
RT: no routes to 172.26.219.0, entering holddown
IP: cache invalidation from 0x115248 0x1378A, new version 5737
RT: 172.26.219.0 came out of holddown
```

debug ip rsvp



Caution

Use this command with a small number of tunnels or Resource Reservation Protocol (RSVP) reservations. Too much data can overload the console.

To display debug messages for RSVP categories, use the **debug ip rsvp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug ip rsvp [all | api | data-pkts | database | dump-messages | events | fast-reroute | filter |
function | handles | messages | msg-mgr | path | policy | proxy | rate-limit | reliable-msg |
resv | routing | sbm | signalling | snmp | summary-refresh | svc | timer | traffic-control | wfq]
```

```
no debug ip rsvp
```

Syntax Description

all	(Optional) RSVP messages for all categories.
api	(Optional) RSVP application programming interface (API) events.
data-pkts	(Optional) RSVP data processing.
database	(Optional) RSVP Database debugging.
dump-messages	(Optional) Dump RSVP message contents.
events	(Optional) RSVP process events.
fast-reroute	(Optional) RSVP fast-reroute support for label-switched paths (LSPs).
filter	(Optional) RSVP debug message filter.
function	(Optional) RSVP function names.
handles	(Optional) RSVP database handles event.
messages	(Optional) Brief information about all RSVP messages that are sent and received via IP debugging.
msg-mgr	(Optional) RSVP message-manager events.
path	(Optional) RSVP Path messages.
policy	(Optional) RSVP policy information.
proxy	(Optional) Proxy API trace.
rate-limit	(Optional) RSVP rate-limiting events.
reliable-msg	(Optional) RSVP reliable messages events.
resv	(Optional) RSVP Resv messages.
routing	(Optional) RSVP routing messages.
sbm	(Optional) RSVP subnet bandwidth manager (SBM) messages.
signalling	(Optional) RSVP signalling (Path and Resv) messages.
snmp	(Optional) RSVP Simple Network Management Protocol (SNMP) events.
summary-refresh	(Optional) RSVP summary refresh and bundle messages events.
svc	(Optional) Switched virtual circuit (SVC) events.
timer	(Optional) RSVP timer events.
traffic-control	(Optional) RSVP traffic control events.
wfq	(Optional) RSVP weighted fair queueing (WFQ) events.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(13)T	The dump-messages , msg-mgr , proxy , rate-limit , reliable-msg , and summary-refresh keywords were added.

Examples

The following commands show how to enable debugging for RSVP categories, signalling and messages:

```
Router# debug ip rsvp signalling
```

```
RSVP signalling messages (Summary) debugging is on
```

```
Router# debug ip rsvp messages
```

```
RSVP messages (sent/received via IP) debugging is on
```

In the following display, RSVP signalling-related events that include sending and receiving Path and Resv messages, admitting new reservations, establishing sessions, sending and receiving acknowledgments (ACKS), and sending and receiving summary refresh messages appear:

```
01:14:56:RSVP 140.20.1.1_19->140.75.1.1_100[140.20.1.1]:Received Path message from
140.20.1.1 (on sender host)
01:14:56:RSVP:new path message passed parsing, continue...
01:14:56:RSVP 140.20.1.1_19->140.75.1.1_100[140.20.1.1]:Refresh Path psb = 61646BB0
refresh interval = 0mSec
01:14:56:RSVP 140.20.1.1_19->140.75.1.1_100[140.20.1.1]:Sending Path message to 140.4.4.2
01:14:56:RSVP session 140.75.1.1_100[140.20.1.1]:Path sent by IP to 140.4.4.2 length=216
checksum=B1E4 TOS=0xC0 prerouted=YES
router_alert=YES udp=NO (Ethernet1)
01:14:56:RSVP:Resv received from IP layer (IP HDR 140.4.4.2->140.4.4.1)
01:14:56:RSVP session 140.75.1.1_100[140.20.1.1]:Received RESV for 140.75.1.1 (Ethernet1)
from 140.4.4.2
01:14:56:RSVP 140.20.1.1_19->140.75.1.1_100[140.20.1.1]:reservation not found--new one
01:14:56:RSVP-RESV:Admitting new reservation:6165D0E4
01:14:56:RSVP 140.20.1.1_19->140.75.1.1_100[140.20.1.1]:RSVP bandwidth is available
01:14:56:RSVP-RESV:reservation was installed:6165D0E4
01:14:57:RSVP:Sending Unknown message to 140.4.4.2
01:14:57:RSVP:Ack sent by IP to 140.4.4.2 length=20 checksum=34A7 TOS=0x00 prerouted=NO
router_alert=NO udp=NO (Ethernet1)
01:14:57:RSVP 140.20.1.1_19->140.75.1.1_100[140.20.1.1]:Refresh Path psb = 61646BB0
refresh interval = 937mSec
01:14:58:%LINK-3-UPDOWN:Interface Tunnel100, changed state to up
01:14:59:%LINEPROTO-5-UPDOWN:Line protocol on Interface Tunnel100, changed state to up
01:15:26:RSVP 140.20.1.1_19->140.75.1.1_100[140.20.1.1]:Refresh Path psb = 61646BB0
refresh interval = 30000mSec
01:15:26:RSVP 140.20.1.1_19->140.75.1.1_100[140.20.1.1]:Sending Path message to 140.4.4.2
01:15:26:RSVP session 140.75.1.1_100[140.20.1.1]:Path sent by IP to 140.4.4.2 length=216
checksum=B1E4 TOS=0xC0 prerouted=YES
router_alert=YES udp=NO (Ethernet1)
01:15:26:RSVP:Resv received from IP layer (IP HDR 140.4.4.2->140.4.4.1)
```

```

01:15:26:RSVP session 140.75.1.1_100[140.20.1.1]:Received RESV for 140.75.1.1 (Ethernet1)
from 140.4.4.2
01:15:26:RSVP 140.20.1.1_19->140.75.1.1_100[140.20.1.1]:reservation found--processing
possible change:6165D0E4
01:15:26:RSVP 140.20.1.1_19->140.75.1.1_100[140.20.1.1]:No change in reservation
01:15:27:RSVP:Sending Ack message to 140.4.4.2
01:15:27:RSVP:Ack sent by IP to 140.4.4.2 length=20 checksum=34A7 TOS=0x00 prerouted=NO
router_alert=NO udp=NO (Ethernet1)
01:15:56:RSVP:Sending Srefresh message to 140.4.4.2
01:15:56:RSVP:Srefresh sent by IP to 140.4.4.2 length=32 checksum=CA0D TOS=0x00
prerouted=NO router_alert=NO udp=NO (Ethernet1)
01:15:56:RSVP:Ack received from IP layer (IP HDR 140.4.4.2->140.4.4.1)
01:15:56:RSVP:Srefresh received from IP layer (IP HDR 140.4.4.2->140.4.4.1)
01:15:56:RSVP-RESV:Resv state is being refreshed for 0x91
01:15:56:RSVP:Sending Ack message to 140.4.4.2
01:15:56:RSVP:Ack sent by IP to 140.4.4.2 length=20 checksum=34A5 TOS=0x00 prerouted=NO
router_alert=NO udp=NO (Ethernet1)
01:16:26:RSVP:Sending Srefresh message to 140.4.4.2
01:16:26:RSVP:Srefresh sent by IP to 140.4.4.2 length=32 checksum=CA0C TOS=0x00
prerouted=NO router_alert=NO udp=NO (Ethernet1)
01:16:26:RSVP:Ack received from IP layer (IP HDR 140.4.4.2->140.4.4.1)
01:16:26:RSVP:Srefresh received from IP layer (IP HDR 140.4.4.2->140.4.4.1)
01:16:26:RSVP-RESV:Resv state is being refreshed for 0x91
01:16:26:RSVP:Sending Ack message to 140.4.4.2
01:16:26:RSVP:Ack sent by IP to 140.4.4.2 length=20 checksum=34A3 TOS=0x00 prerouted=NO
router_alert=NO udp=NO (Ethernet1)

```

Related Commands

Command	Description
ip rsvp signalling refresh reduction	Enables refresh reduction.
show debug	Displays active debug output.

debug ip rsvp authentication

To display debugging output related to Resource Reservation Protocol (RSVP) authentication, use the **debug ip rsvp authentication** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rsvp authentication

no debug ip rsvp authentication

Syntax Description This command has no arguments or keywords.

Defaults This command is disabled.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.2(15)T	This command was introduced.

Usage Guidelines After you enable RSVP authentication, RSVP logs system error events whenever an authentication check fails. These events are logged instead of just being displayed when debugging is enabled because they may indicate potential security attacks. The events are generated when:

- RSVP receives a message that does not contain the correct cryptographic signature. This could be due to misconfiguration of the authentication key or algorithm on one or more RSVP neighbors, but it may also indicate an (unsuccessful) attack.
- RSVP receives a message with the correct cryptographic signature, but with a duplicate authentication sequence number. This may indicate an (unsuccessful) message replay attack.
- RSVP receives a message with the correct cryptographic signature, but with an authentication sequence number that is outside the receive window. This could be due to a reordered burst of valid RSVP messages, but it may also indicate an (unsuccessful) message replay attack.
- Failed challenges result from timeouts or bad challenge responses.

Examples The following example shows output from the **debug ip rsvp authentication** command in which the authentication type (digest) and the sequence number have been validated:

```
Router# debug ip rsvp authentication
```

```
RSVP authentication debugging is on
```

Router# **show debugging**

```
*Jan 30 08:10:46.335:RSVP_AUTH:Resv integrity digest from 192.168.101.2 valid
*Jan 30 08:10:46.335:RSVP_AUTH:Resv integrity sequence number 13971113505298841601 from
192.168.101.2 valid
*Jan 30 08:10:46.335:RSVP_AUTH:Resv from 192.168.101.2 passed all authentication checks
```



Note

Cisco routers using RSVP authentication on Cisco IOS ideally should have clocks that can be accurately restored to the correct time when the routers boot. This capability is available on certain Cisco routers that have clocks with battery backup. For those platforms that do not have battery backup, consider configuring the router to keep its clock synchronized with a Network Time Protocol (NTP) time server. Otherwise, if two adjacent routers have been operating with RSVP authentication enabled and one of them reboots such that its clock goes backward in time, it is possible (but unlikely) the router that did not reboot will log RSVP authentication sequence number errors.

Related Commands

Command	Description
ip rsvp authentication	Activates RSVP cryptographic authentication.
show debugging	Displays active debug output.

debug ip rsvp detail

To display detailed information about Resource Reservation Protocol (RSVP)-enabled and Subnetwork Bandwidth Manager (SBM) message processing, use the **debug ip rsvp detail** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rsvp detail

no debug ip rsvp detail

Syntax Description This command has no arguments or keywords.

Defaults This command is disabled by default.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.0(5)T	This command was introduced.

Examples The following example shows the detailed debug information about RSVP and SBM that is available when you enable debug mode through the **debug ip rsvp detail** command:

```
Router# debug ip rsvp detail

RSVP debugging is on
router2#u
*Dec 31 16:44:29.651: RSVP: send I_AM_DSBM message from 145.2.2.150
*Dec 31 16:44:29.651: RSVP: IP to 224.0.0.17 length=88 checksum=43AF
(Ethernet2)
*Dec 31 16:44:29.651: RSVP: version:1 flags:0000 type:I_AM_DSBM cksum:43AF
      ttl:254 reserved:0 length:88
*Dec 31 16:44:29.651: DSBM_IP_ADDR      type 1 length 8 : 91020296
*Dec 31 16:44:29.651: HOP_L2          type 1 length 12: 00E01ECE
*Dec 31 16:44:29.651:                   : 0F760000
*Dec 31 16:44:29.651: SBM_PRIORITY    type 1 length 8 : 00000064
*Dec 31 16:44:29.651: DSBM_TIMERS     type 1 length 8 : 00000F05
*Dec 31 16:44:29.651: SBM_INFO        type 1 length 44: 00000000
*Dec 31 16:44:29.651:                   : 00240C02 00000007
*Dec 31 16:44:29.651:                   : 01000006 7F000005
*Dec 31 16:44:29.651:                   : 00000000 00000000
*Dec 31 16:44:29.655:                   : 00000000 00000000
*Dec 31 16:44:29.655:                   : 00000000
```

Related Commands

Command	Description
debug ip rsvp	Displays information about SBM message processing, the DSBM election process, and RSVP message processing.
debug ip rsvp detail sbm	Displays detailed information about the contents of SMB messages only, and SBM and DSBM state transitions.
ip rsvp dsbm-candidate	Configures an interface as a DSBM candidate.
show ip rsvp sbm	Displays information about SBM configured for a specific RSVP-enabled interface or all RSVP-enabled interfaces on the router.

debug ip rsvp dump-messages



Caution

Use this command with a small number of tunnels or Resource Reservation Protocol (RSVP) reservations. Too much data can overload the console.

To display debugging messages for all RSVP events, use the **debug ip rsvp dump-messages** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug ip rsvp dump-messages [hex | path | resv | sbm | signalling]
```

```
no debug ip rsvp dump-messages
```

Syntax Description

hex	(Optional) Hex dump of packet contents.
path	(Optional) Contents of Path messages.
resv	(Optional) Contents of Resv messages.
sbm	(Optional) Contents of SBM messages.
signalling	(Optional) Contents of all signaling (Path and Resv) messages.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

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

Examples

The following command shows how to enable debugging for RSVP events:

```
Router# debug ip rsvp dump-messages
```

```
RSVP message dump debugging is on
```

In the following display, notice that a Path message is transmitted and an ACK_DESIRED flag is set for ID: 0x26 Epoch: 0x76798A. In response, a Resv message is sent and an acknowledgment (ACK) is issued for ID: 0x26 Epoch: 0x76798A indicating the RSVP state is established on the neighboring router:

```
00:37:15:RSVP:version:1 flags:0000 type:PROXY_PATH cksum:0000 ttl:255 reserved:0
length:212
00:37:15: SESSION                               type 7 length 16:
00:37:15:     Destination 140.75.1.1, TunnelId 100, Source 140.20.1.1, Protocol 0, Flags
0000
00:37:15: HOP                                   type 1 length 12:
00:37:15:     Neighbor 140.20.1.1, LIH 0x00000000
00:37:15: TIME_VALUES                               type 1 length 8 :
00:37:15:     Refresh period is 30000 msec
```

```

00:37:15: SENDER_TEMPLATE      type 7 length 12:
00:37:15:      Source 140.20.1.1, tunnel_id 9
00:37:15: SENDER_TSPEC                type 2 length 36:
00:37:15:      version=0, length in words=7
00:37:15:      Token bucket fragment (service_id=1, length=6 words
00:37:15:          parameter id=127, flags=0, parameter length=5
00:37:15:          average rate=1250 bytes/sec, burst depth=1000 bytes
00:37:15:          peak rate      =1250 bytes/sec
00:37:15:          min unit=0 bytes, max pkt size=4294967295 bytes
00:37:15: ADSPEC                      type 2 length 48:
00:37:15: version=0 length in words=10
00:37:15: General Parameters break bit=0 service length=8
00:37:15:                                IS Hops:0
00:37:15:                                Minimum Path Bandwidth (bytes/sec):2147483647
00:37:15:                                Path Latency (microseconds):0
00:37:15:                                Path MTU:-1
00:37:15: Controlled Load Service break bit=0 service length=0
00:37:15: LABEL_REQUEST               type 1 length 8 :
00:37:15:      Layer 3 protocol ID:2048
00:37:15: EXPLICIT_ROUTE              type 1 length 36:
00:37:15:      (#1) Strict IPv4 Prefix, 8 bytes, 140.20.1.1/32
00:37:15:      (#2) Strict IPv4 Prefix, 8 bytes, 140.4.4.2/32
00:37:15:      (#3) Strict IPv4 Prefix, 8 bytes, 140.70.1.1/32
00:37:15:      (#4) Strict IPv4 Prefix, 8 bytes, 140.70.1.2/32
00:37:15: SESSION_ATTRIBUTE           type 7 length 28:
00:37:15:      Session name:tagsw4500-21_t100
00:37:15:      Setup priority:7, reservation priority:7
00:37:15:      Status:May-Reroute
00:37:15:
00:37:15:RSVP:version:1 flags:0001 type:Path cksum:D61E ttl:255 reserved:0 length:216
00:37:15: MESSAGE_ID                  type 1 length 12:
00:37:15:      ID:0x26 Epoch:0x76798A
00:37:15:      Flags:ACK_DESIRED
00:37:15: SESSION                     type 7 length 16:
00:37:15:      Destination 140.75.1.1, TunnelId 100, Source 140.20.1.1, Protocol 0, Flags
00:37:15:      0000
00:37:15: HOP                          type 1 length 12:
00:37:15:      Neighbor 140.4.4.1, LIH 0x10000401
00:37:15: TIME_VALUES                  type 1 length 8 :
00:37:15:      Refresh period is 30000 msecs
00:37:15: EXPLICIT_ROUTE              type 1 length 28:
00:37:15:      (#1) Strict IPv4 Prefix, 8 bytes, 140.4.4.2/32
00:37:15:      (#2) Strict IPv4 Prefix, 8 bytes, 140.70.1.1/32
00:37:15:      (#3) Strict IPv4 Prefix, 8 bytes, 140.70.1.2/32
00:37:15: LABEL_REQUEST               type 1 length 8 :
00:37:15:      Layer 3 protocol ID:2048
00:37:15: SESSION_ATTRIBUTE           type 7 length 28:
00:37:15:      Session name:tagsw4500-21_t100
00:37:15:      Setup priority:7, reservation priority:7
00:37:15:      Status:May-Reroute
00:37:15: SENDER_TEMPLATE            type 7 length 12:
00:37:15:      Source 140.20.1.1, tunnel_id 9
00:37:15: SENDER_TSPEC                type 2 length 36:
00:37:15:      version=0, length in words=7
00:37:15:      Token bucket fragment (service_id=1, length=6 words
00:37:15:          parameter id=127, flags=0, parameter length=5
00:37:15:          average rate=1250 bytes/sec, burst depth=1000 bytes
00:37:15:          peak rate      =1250 bytes/sec
00:37:15:          min unit=0 bytes, max pkt size=4294967295 bytes
00:37:15: ADSPEC                      type 2 length 48:
00:37:15: version=0 length in words=10
00:37:15: General Parameters break bit=0 service length=8
00:37:15:                                IS Hops:1
00:37:15:                                Minimum Path Bandwidth (bytes/sec):1250000

```

```

00:37:15:                               Path Latency (microseconds):0
00:37:15:                               Path MTU:1500
00:37:15: Controlled Load Service break bit=0 service length=0
00:37:15:
00:37:15:RSVP:version:1 flags:0001 type:Resv cksum:DADF ttl:255 reserved:0 length:132
00:37:15: MESSAGE_ID_ACK type 1 length 12:
00:37:15:     Type:ACK
00:37:15:     ID:0x26 Epoch:0x76798A
00:37:15:     Flags:None
00:37:15: MESSAGE_ID type 1 length 12:
00:37:15:     ID:0x43 Epoch:0xE1A1B7
00:37:15:     Flags:ACK_DESIRED
00:37:15: SESSION type 7 length 16:
00:37:15:     Destination 140.75.1.1, TunnelId 100, Source 140.20.1.1, Protocol 0, Flags
0000
00:37:15: HOP type 1 length 12:
00:37:15:     Neighbor 140.4.4.2, LIH 0x10000401
00:37:15: TIME_VALUES type 1 length 8 :
00:37:15:     Refresh period is 30000 msec
00:37:15: STYLE type 1 length 8 :
00:37:15:     Shared-Explicit (SE)
00:37:15: FLOWSPEC type 2 length 36:
00:37:15:     version = 0 length in words = 7
00:37:15:     service id = 5, service length = 6
00:37:15:     tspec parameter id = 127, flags = 0, length = 5
00:37:15:     average rate = 1250 bytes/sec, burst depth = 1000 bytes
00:37:15:     peak rate = 1250 bytes/sec
00:37:15:     min unit = 0 bytes, max pkt size = 0 bytes
00:37:15: FILTER_SPEC type 7 length 12:
00:37:15:     Source 140.20.1.1, tunnel_id 9
00:37:15: LABEL type 1 length 8 :
00:37:15:     Labels:16
00:37:15:
00:37:15:RSVP:version:1 flags:0001 type:Ack cksum:34F5 ttl:255 reserved:0 length:20
00:37:15: MESSAGE_ID_ACK type 1 length 12:
00:37:15:     Type:ACK
00:37:15:     ID:0x43 Epoch:0xE1A1B7
00:37:15:     Flags:None
00:37:15:
00:37:17:%LINK-3-UPDOWN:Interface Tunnel100, changed state to up
00:37:18:%LINEPROTO-5-UPDOWN:Line protocol on Interface Tunnel100, changed state to up

```

Related Commands

Command	Description
ip rsvp signalling refresh reduction	Enables refresh reduction.
show debug	Displays active debug output.

debug ip rsvp policy

To display debugging messages for Resource Reservation Protocol (RSVP) policy processing, use the **debug ip rsvp policy** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rsvp policy

no debug ip rsvp policy

Syntax Description This command has no arguments or keywords.

Defaults Debugging for RSVP policy processing is not enabled.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.1(1)T	This command was introduced.

Usage Guidelines You might find it useful to enable the **debug cops** command when you are using the **debug ip rsvp policy** command. Together, these commands generate a complete record of the policy process.

Examples The following example uses only the **debug ip rsvp policy** command:

```
Router-1# debug ip rsvp policy

RSVP_POLICY debugging is on

02:02:14:RSVP-POLICY:Creating outbound policy IDB entry for Ethernet2/0 (61E6AB38)
02:02:14:RSVP-COPS:COPS query for Path message, 10.31.0.1_44->10.33.0.1_44
02:02:14:RSVP-POLICY:Building incoming Path context
02:02:14:RSVP-POLICY:Building outgoing Path context on Ethernet2/0
02:02:14:RSVP-POLICY:Build REQ message of 216 bytes
02:02:14:RSVP-POLICY:Message sent to PDP
02:02:14:RSVP-COPS:COPS engine called us with reason2, handle 6202A658
02:02:14:RSVP-COPS:Received decision message
02:02:14:RSVP-POLICY:Received decision for Path message
02:02:14:RSVP-POLICY:Accept incoming message
02:02:14:RSVP-POLICY:Send outgoing message to Ethernet2/0
02:02:14:RSVP-POLICY:Replacement policy object for path-in context
02:02:14:RSVP-POLICY:Replacement TSPEC object for path-in context
02:02:14:RSVP-COPS:COPS report for Path message, 10.31.0.1_44->10.33.0.1_44
02:02:14:RSVP-POLICY:Report sent to PDP
02:02:14:RSVP-COPS:COPS report for Path message, 10.31.0.1_44->10.33.0.1_44
```

The following example uses both the **debug ip rsvp policy** and the **debug cops** commands:

```
Router-1# debug ip rsvp policy

RSVP_POLICY debugging is on

Router-1# debug cops

COPS debugging is on

02:15:14:RSVP-POLICY:Creating outbound policy IDB entry for Ethernet2/0 (61E6AB38)
02:15:14:RSVP-COPS:COPS query for Path message, 10.31.0.1_44->10.33.0.1_44
02:15:14:RSVP-POLICY:Building incoming Path context
02:15:14:RSVP-POLICY:Building outgoing Path context on Ethernet2/0
02:15:14:RSVP-POLICY:Build REQ message of 216 bytes
02:15:14:COPS:** SENDING MESSAGE **
    COPS HEADER:Version 1, Flags 0, Opcode 1 (REQ), Client-type:1, Length:216
    HANDLE (1/1) object. Length:8.    00 00 22 01
    CONTEXT (2/1) object. Length:8.    R-type:5.    M-type:1
    IN_IF (3/1) object. Length:12.    Address:10.1.2.1.    If_index:4
    OUT_IF (4/1) object. Length:12.    Address:10.33.0.1.    If_index:3
    CLIENT SI (9/1) object. Length:168.    CSI data:
02:15:14: SESSION                type 1 length 12:
02:15:14: Destination 10.33.0.1, Protocol_Id 17, Don't Police , DstPort 44
02:15:14: HOP                    type 1 length 12:0A010201
02:15:14:                        :00000000
02:15:14: TIME_VALUES            type 1 length 8 :00007530
02:15:14: SENDER_TEMPLATE        type 1 length 12:
02:15:14: Source 10.31.0.1, udp_source_port 44
02:15:14: SENDER_TSPEC           type 2 length 36:
02:15:14: version=0, length in words=7
02:15:14: Token bucket fragment (service_id=1, length=6 words
02:15:14:     parameter id=127, flags=0, parameter length=5
02:15:14:     average rate=1250 bytes/sec, burst depth=10000 bytes
02:15:14:     peak rate =1250000 bytes/sec
02:15:14:     min unit=0 bytes, max unit=1514 bytes
02:15:14: ADSPEC                  type 2 length 84:
02:15:14: version=0 length in words=19
02:15:14: General Parameters break bit=0 service length=8
02:15:14:                        IS Hops:1
02:15:14: Minimum Path Bandwidth (bytes/sec):1250000
02:15:14: Path Latency (microseconds):0
02:15:14: Path MTU:1500
02:15:14: Guaranteed Service break bit=0 service length=8
02:15:14: Path Delay (microseconds):192000
02:15:14: Path Jitter (microseconds):1200
02:15:14: Path delay since shaping (microseconds):192000
02:15:14: Path Jitter since shaping (microseconds):1200
02:15:14: Controlled Load Service break bit=0 service length=0
02:15:14:COPS:Sent 216 bytes on socket,
02:15:14:RSVP-POLICY:Message sent to PDP
02:15:14:COPS:Message event!
02:15:14:COPS:State of TCP is 4
02:15:14:In read function
02:15:14:COPS:Read block of 96 bytes, num=104 (len=104)
02:15:14:COPS:** RECEIVED MESSAGE **
    COPS HEADER:Version 1, Flags 1, Opcode 2 (DEC), Client-type:1, Length:104
    HANDLE (1/1) object. Length:8.    00 00 22 01
    CONTEXT (2/1) object. Length:8.    R-type:1.    M-type:1
    DECISION (6/1) object. Length:8.    COMMAND cmd:1, flags:0
    DECISION (6/3) object. Length:56.    REPLACEMENT 00 10 0E 01 61 62 63 64 65 66 67
    68 69 6A 6B 6C 00 24 0C 02 00
    00 00 07 01 00 00 06 7F 00 00 05 44 9C 40 00 46 1C 40 00 49 98
    96 80 00 00 00 C8 00 00 01 C8
```

```

CONTEXT (2/1) object. Length:8.   R-type:4.   M-type:1
DECISION (6/1) object. Length:8.   COMMAND cmd:1, flags:0

02:15:14:Notifying client (callback code 2)
02:15:14:RSVP-COPS:COPS engine called us with reason2, handle 6202A104
02:15:14:RSVP-COPS:Received decision message
02:15:14:RSVP-POLICY:Received decision for Path message
02:15:14:RSVP-POLICY:Accept incoming message
02:15:14:RSVP-POLICY:Send outgoing message to Ethernet2/0
02:15:14:RSVP-POLICY:Replacement policy object for path-in context
02:15:14:RSVP-POLICY:Replacement TSPEC object for path-in context
02:15:14:RSVP-COPS:COPS report for Path message, 10.31.0.1_44->10.33.0.1_44
02:15:14:COPS:** SENDING MESSAGE **
    COPS HEADER:Version 1, Flags 1, Opcode 3 (RPT), Client-type:1, Length:24
    HANDLE (1/1) object. Length:8.   00 00 22 01
    REPORT (12/1) object. Length:8.   REPORT type COMMIT (1)

02:15:14:COPS:Sent 24 bytes on socket,
02:15:14:RSVP-POLICY:Report sent to PDP
02:15:14:Timer for connection entry is zero
02:15:14:RSVP-COPS:COPS report for Path message, 10.31.0.1_44->10.33.0.1_44

```

Related Commands

Command	Description
debug cops	Displays debugging messages for COPS processing.

debug ip rsvp rate-limit

To display debugging messages for Resource Reservation Protocol (RSVP) rate-limiting events, use the **debug ip rsvp rate-limit** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rsvp rate-limit

no debug ip rsvp rate-limit

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes Privileged EXEC

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

Examples The following command shows how to enable debugging for RSVP rate-limiting and message manager events:

```
Router# debug ip rsvp rate-limit
```

```
RSVP rate-limit debugging is on
```

```
Router# debug ip rsvp msg-mgr
```

```
RSVP msg-mgr debugging is on
```

In the following display, RSVP process information including messages, timers, neighbors IP addresses, and message IDs, appear:

```
01:00:19:RSVP-RATE-LIMIT:rsvp_msg_pacing_send_message
01:00:19:RSVP-MSG-MGR (140.4.4.2):Starting timer msg-pacing interval 20
01:00:19:RSVP-MSG-MGR (140.4.4.2):Enqueue element 27000405 of type 3 on msg-pacing TAIL
01:00:19:RSVP-RATE-LIMIT:rsvp_msg_pacing_timer - timer expired
01:00:19:RSVP-MSG-MGR (140.4.4.2):Dequeueing element 27000405 of type 3 from msg-pacing
01:00:19:RSVP-RATE-LIMIT:rsvp_msg_pacing_send_qe:sending psb (qe 27000405)
01:00:21:%LINK-3-UPDOWN:Interface Tunnel100, changed state to up
01:00:22:%LINEPROTO-5-UPDOWN:Line protocol on Interface Tunnel100, changed state to up
01:01:03:RSVP-RATE-LIMIT:rsvp_msg_pacing_send_message
01:01:03:RSVP-MSG-MGR (140.4.4.2):Starting timer msg-pacing interval 20
01:01:03:RSVP-MSG-MGR (140.4.4.2):Enqueue element 27000405 of type 3 on msg-pacing TAIL
01:01:03:RSVP-RATE-LIMIT:rsvp_msg_pacing_timer - timer expired
01:01:03:RSVP-MSG-MGR (140.4.4.2):Dequeueing element 27000405 of type 3 from msg-pacing
01:01:03:RSVP-RATE-LIMIT:rsvp_msg_pacing_send_qe:sending psb (qe 27000405)
01:01:42:RSVP-RATE-LIMIT:rsvp_msg_pacing_send_message
01:01:42:RSVP-MSG-MGR (140.4.4.2):Starting timer msg-pacing interval 20
```

```

01:01:42:RSVP-MSG-MGR (140.4.4.2):Enqueue element 27000405 of type 3 on msg-pacing TAIL
01:01:42:RSVP-RATE-LIMIT:rsvp_msg_pacing_timer - timer expired
01:01:42:RSVP-MSG-MGR (140.4.4.2):Dequeueing element 27000405 of type 3 from msg-pacing
01:01:42:RSVP-RATE-LIMIT:rsvp_msg_pacing_send_qe:sending psb (qe 27000405)
01:02:09:RSVP-RATE-LIMIT:rsvp_msg_pacing_send_message
01:02:09:RSVP-MSG-MGR (140.4.4.2):Starting timer msg-pacing interval 20
01:02:09:RSVP-MSG-MGR (140.4.4.2):Enqueue element 27000405 of type 3 on msg-pacing TAIL
01:02:09:RSVP-RATE-LIMIT:rsvp_msg_pacing_timer - timer expired
01:02:09:RSVP-MSG-MGR (140.4.4.2):Dequeueing element 27000405 of type 3 from msg-pacing
01:02:09:RSVP-RATE-LIMIT:rsvp_msg_pacing_send_qe:sending psb (qe 27000405)

```

Related Commands

Command	Description
ip rsvp signalling rate-limit	Controls the transmission rate for RSVP messages sent to a neighboring router during a specified interval.
show debug	Displays active debug output.

debug ip rsvp reliable-msg

To display debugging messages for Resource Reservation Protocol (RSVP) reliable messages events, use the **debug ip rsvp reliable-msg** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rsvp reliable-msg

no debug ip rsvp reliable-msg

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes Privileged EXEC

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

Examples The following command shows how to enable debugging for RSVP reliable messages events:

```
Router# debug ip rsvp reliable-msg
```

```
RSVP reliable-msg debugging is on
```

In the following display, message IDs, acknowledgments (ACKs), and message processes including retransmissions, appear:

```
01:07:37:RSVP-RMSG:Inserted msg id(0x46, 0x48000403) on local msgid db
01:07:37:RSVP-RMSG:rsvp_rmsg_process_acks, Handle:000C1701 neighbor:140.4.4.2
01:07:37:RSVP-RMSG:max_ids:1 q_sz:1 msg_sz:1500 ids_len:1432 num_objs:0 obj_len:0
nbr:140.4.4.2
01:07:39:%LINK-3-UPDOWN:Interface Tunnel100, changed state to up
01:07:40:%LINEPROTO-5-UPDOWN:Line protocol on Interface Tunnel100, changed state to up
01:08:07:RSVP-RMSG:rsvp_rmsg_process_acks, Handle:000C1701 neighbor:140.4.4.2
01:08:07:RSVP-RMSG:max_ids:1 q_sz:1 msg_sz:1500 ids_len:1432 num_objs:0 obj_len:0
nbr:140.4.4.2
01:08:37:RSVP-RMSG:max_ids:1 q_sz:1 msg_sz:1500 ids_len:1424 num_objs:1 obj_len:8
nbr:140.4.4.2
01:08:37:RSVP-RMSG:rsvp_rmsg_process_immediate_tmb, Handle:2D000404 neighbor:140.4.4.2
01:08:37:RSVP-RMSG:Inserted msg id(0x47, 0x2D000404) on local msgid db
01:08:37:RSVP-RMSG:current queue:immed next_queue:rxmt-1 (qe 2D000404s)
01:08:37:RSVP-RMSG:rsvp_rmsg_process_acks, Handle:000C1701 neighbor:140.4.4.2
01:08:37:RSVP-RMSG:max_ids:1 q_sz:1 msg_sz:1500 ids_len:1432 num_objs:0 obj_len:0
nbr:140.4.4.2
01:08:38:RSVP-RMSG:rsvp_rmsg_process_rxmt_tmb, Handle:2D000404 neighbor:140.4.4.2
01:08:38:RSVP-RMSG:An ack was received for tmb 2D000404 on neighbor 140.4.4.2
01:09:07:RSVP-RMSG:max_ids:1 q_sz:1 msg_sz:1500 ids_len:1424 num_objs:1 obj_len:8
nbr:140.4.4.2
```

```

01:09:07:RSVP-RMSG:rsvp_rmsg_process_immediate_tmb, Handle:2E000404 neighbor:140.4.4.2
01:09:07:RSVP-RMSG:Inserted msg id(0x48, 0x2E000404) on local msgid db
01:09:07:RSVP-RMSG:current queue:immed next_queue:rxmt-1 (qe 2E000404s)
01:09:07:RSVP-RMSG:rsvp_rmsg_process_acks, Handle:000C1701 neighbor:140.4.4.2
01:09:07:RSVP-RMSG:max_ids:1 q_sz:1 msg_sz:1500 ids_len:1432 num_objs:0 obj_len:0
nbr:140.4.4.2
01:09:08:RSVP-RMSG:rsvp_rmsg_process_rxmt_tmb, Handle:2E000404 neighbor:140.4.4.2
01:09:08:RSVP-RMSG:An ack was received for tmb 2E000404 on neighbor 140.4.4.2
01:09:37:RSVP-RMSG:max_ids:1 q_sz:1 msg_sz:1500 ids_len:1424 num_objs:1 obj_len:8
nbr:140.4.4.2
01:09:37:RSVP-RMSG:rsvp_rmsg_process_immediate_tmb, Handle:2F000404 neighbor:140.4.4.2
01:09:37:RSVP-RMSG:Inserted msg id(0x49, 0x2F000404) on local msgid db
01:09:37:RSVP-RMSG:current queue:immed next_queue:rxmt-1 (qe 2F000404s)
01:09:37:RSVP-RMSG:rsvp_rmsg_process_acks, Handle:000C1701 neighbor:140.4.4.2
01:09:37:RSVP-RMSG:max_ids:1 q_sz:1 msg_sz:1500 ids_len:1432 num_objs:0 obj_len:0
nbr:140.4.4.2
01:09:38:RSVP-RMSG:rsvp_rmsg_process_rxmt_tmb, Handle:2F000404 neighbor:140.4.4.2
01:09:38:RSVP-RMSG:An ack was received for tmb 2F000404 on neighbor 140.4.4.2

```

Related Commands

Command	Description
ip rsvp signalling refresh reduction	Enables refresh reduction.
show debug	Displays active debug output.

debug ip rsvp sbm

To display detailed information about Subnetwork Bandwidth Manager (SBM) messages only, and SBM and Designated Subnetwork Bandwidth Manager (DSBM) state transitions, use the **debug ip rsvp sbm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rsvp sbm

no debug ip rsvp sbm

Syntax Description This command has no arguments or keywords.

Defaults This command is disabled by default.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.0(5)T	This command was introduced.

Usage Guidelines The **debug ip rsvp sbm** command provides information about messages received, minimal detail about the content of these messages, and information about state transitions.

Examples The following example shows the detailed debug information about SBM and the SBM and DSBM state transitions that is available when you enable debug mode through the **debug ip rsvp sbm** command:

```
Router# debug ip rsvp sbm

RSVP debugging is on
router2#
*Dec 31 16:45:34.659: RSVP: send I_AM_DSBM message from 145.2.2.150
*Dec 31 16:45:34.659: RSVP: IP to 224.0.0.17 length=88 checksum=9385 (Ethernet2)
*Dec 31 16:45:34.659:  RSVP: version:1 flags:0000 type:I_AM_DSBM cksum:9385
                        ttl:254 reserved:0 length:88
*Dec 31 16:45:34.659:  DSBM_IP_ADDR      type 1 length 8 : 91020296
*Dec 31 16:45:34.659:  HOP_L2          type 1 length 12: 00E01ECE
*Dec 31 16:45:34.659:                               : 0F760000
*Dec 31 16:45:34.659:  SBM_PRIORITY    type 1 length 8 : 0029B064
*Dec 31 16:45:34.659:  DSBM_TIMERS     type 1 length 8 : 00000F05
*Dec 31 16:45:34.659:  SBM_INFO        type 1 length 44: 00000000
*Dec 31 16:45:34.659:                               : 00240C02 00000007
*Dec 31 16:45:34.659:                               : 01000006 7F000005
*Dec 31 16:45:34.659:                               : 00000000 00000000
*Dec 31 16:45:34.663:                               : 00000000 00000000
*Dec 31 16:45:34.663:                               : 00000000
*Dec 31 16:45:34.663:
```

Related Commands

Command	Description
debug ip rsvp	Displays information about SBM message processing, the DSBM election process, and RSVP message processing.
debug ip rsvp authentication	Displays detailed information about RSVP and SBM.
ip rsvp dsbm-candidate	Configures an interface as a DSBM candidate.

debug ip rsvp summary-refresh

To display debugging messages for Resource Reservation Protocol (RSVP) summary-refresh messages events, use the **debug ip rsvp summary-refresh** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rsvp summary-refresh

no debug ip rsvp summary-refresh

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes Privileged EXEC

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

Examples The following command shows how to enable debugging for RSVP summary-refresh messages events:

```
Router# debug ip rsvp summary-refresh
```

```
RSVP summary-refresh debugging is on
```

In the following output, the IP addresses, the interfaces, the types of RSVP messages (Path and Resv), message IDs, and epoch identifiers (for routers) for which RSVP summary-refresh events occur are shown:

```
01:11:00:RSVP-SREFRESH:Incoming message from nbr 140.4.4.2 with epoch:0xE1A1B7 msgid:0x84
on Ethernet1
01:11:00:RSVP-SREFRESH 140.20.1.1_18->140.75.1.1_100[140.20.1.1]:Created msgid 0x84 for
nbr 140.4.4.2
01:11:02:%LINK-3-UPDOWN:Interface Tunnel100, changed state to up
01:11:03:%LINEPROTO-5-UPDOWN:Line protocol on Interface Tunnel100, changed state to up
01:11:30:RSVP-SREFRESH:140.20.1.1_18->140.75.1.1_100[140.20.1.1]:Path, ID:0x4C :Start
using Srefresh to 140.4.4.2
01:11:31:RSVP-SREFRESH:Incoming message from nbr 140.4.4.2 with epoch:0xE1A1B7 msgid:0x84
on Ethernet1
01:11:31:RSVP-SREFRESH:State exists for nbr:140.4.4.2 epoch:0xE1A1B7 msgid:0x84
01:12:00:RSVP-SREFRESH:Preparing to Send Srefresh(es) to 140.4.4.2, 1 IDs Total
01:12:00:RSVP-SREFRESH:Sending 1 IDs in this Srefresh
01:12:00:RSVP-SREFRESH:140.20.1.1_18->140.75.1.1_100[140.20.1.1]:Path, ID:0x4C
01:12:01:RSVP-SREFRESH:Incoming message from nbr 140.4.4.2 with epoch:0xE1A1B7 msgid:0x86
on Ethernet1
01:12:01:RSVP-SREFRESH:Rec'd 1 IDs in Srefresh from 140.4.4.2 (on Ethernet1),
epoch:0xE1A1B7 msgid:0x86
01:12:01:RSVP-SREFRESH:140.20.1.1_18->140.75.1.1_100[140.20.1.1]:Resv, ID:0x84
01:12:30:RSVP-SREFRESH:Preparing to Send Srefresh(es) to 140.4.4.2, 1 IDs Total
```

```

01:12:30:RSVP-SREFRESH:Sending 1 IDs in this Srefresh
01:12:30:RSVP-SREFRESH:140.20.1.1_18->140.75.1.1_100[140.20.1.1]:Path, ID:0x4C
01:12:31:RSVP-SREFRESH:Incoming message from nbr 140.4.4.2 with epoch:0xE1A1B7 msgid:0x88
on Ethernet1
01:12:31:RSVP-SREFRESH:Rec'd 1 IDs in Srefresh from 140.4.4.2 (on Ethernet1),
epoch:0xE1A1B7 msgid:0x88
01:12:31:RSVP-SREFRESH:140.20.1.1_18->140.75.1.1_100[140.20.1.1]:Resv, ID:0x84
01:13:00:RSVP-SREFRESH:Preparing to Send Srefresh(es) to 140.4.4.2, 1 IDs Total
01:13:00:RSVP-SREFRESH:Sending 1 IDs in this Srefresh
01:13:00:RSVP-SREFRESH:140.20.1.1_18->140.75.1.1_100[140.20.1.1]:Path, ID:0x4C
01:13:01:RSVP-SREFRESH:Incoming message from nbr 140.4.4.2 with epoch:0xE1A1B7 msgid:0x8A
on Ethernet1
01:13:01:RSVP-SREFRESH:Rec'd 1 IDs in Srefresh from 140.4.4.2 (on Ethernet1),
epoch:0xE1A1B7 msgid:0x8A
01:13:01:RSVP-SREFRESH:140.20.1.1_18->140.75.1.1_100[140.20.1.1]:Resv, ID:0x84

```



Note

In the preceding output, notice the message IDs that correspond to Path or Resv state being refreshed. Because the entire message does not have to be transmitted, there is less data and network performance is improved.

Related Commands

Command	Description
ip rsvp signalling refresh reduction	Enables refresh reduction.
show debug	Displays active debug output.

debug ip rsvp traffic-control

To display debugging messages for compression-related events, use the **debug ip rsvp traffic-control** command in EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rsvp traffic-control

no debug ip rsvp traffic-control

Syntax Description This command has no arguments or keywords.

Defaults This command is disabled.

Command Modes EXEC

Command History	Release	Modification
	12.0	This command was introduced.
	12.2(15)T	The command output was modified to include compression-related events.

Usage Guidelines Use the **debug ip rsvp traffic-control** command to troubleshoot compression-related problems.

Examples

The following example from the **debug ip rsvp traffic-control** command shows that compression was successfully predicted:

```
Router# debug ip rsvp traffic-control

RSVP debugging is on

Router# show debugging

00:44:49: RSVP-TC: Attempting to install QoS for rsb 62CC66F0
00:44:49: RSVP-TC: Adding new tcsb 02000406 for rsb 62CC66F0
00:44:49: RSVP-TC: Assigning WFQ QoS (on FR VC 101) to tcsb 02000406
00:44:49: RSVP-TC: Predicted compression for TCSB 2000406:
00:44:49: RSVP-TC:   method       = rtp
00:44:49: RSVP-TC:   context ID = 2
00:44:49: RSVP-TC:   factor       = 82 percent
00:44:49: RSVP-TC:   bytes-saved = 36 bytes
00:44:49: RSVP-TC: Bandwidth check: requested bw=65600 old bw=0
00:44:49: RSVP-TC: RSVP bandwidth is available
00:44:49: RSVP-TC: Consulting policy for tcsb 02000406
00:44:49: RSVP-TC: Policy granted QoS for tcsb 02000406
00:44:49: RSVP-TC: Requesting QoS for tcsb 02000406
00:44:49: RSVP-TC:   ( r = 8200       bytes/s   M = 164       bytes
00:44:49: RSVP-TC:     b = 328       bytes     m = 164       bytes )
00:44:49: RSVP-TC:     p = 10000     bytes/s   Service Level = priority
00:44:49: RSVP-WFQ: Update for tcsb 02000406 on FR PVC dlci 101 on Se3/0
00:44:49: RSVP-WFQ: Admitted 66 kbps of bandwidth
00:44:49: RSVP-WFQ: Allocated PRIORITY queue 24
00:44:49: RSVP-TC: Allocation succeeded for tcsb 02000406
```

The following example from the **debug ip rsvp traffic-control** command shows that compression was unsuccessfully predicted because no compression context IDs were available:

```
Router# debug ip rsvp traffic-control

RSVP debugging is on

Router# show debugging

00:10:16:RSVP-TC:Attempting to install QoS for rsb 62CED62C
00:10:16:RSVP-TC:Adding new tcsb 01000421 for rsb 62CED62C
00:10:16:RSVP-TC:Assigning WFQ QoS (on FR VC 101) to tcsb 01000421
00:10:16:RSVP-TC:sender's flow is not rtp compressible for TCSB 1000421
00:10:16:   reason: no contexts available
00:10:16:RSVP-TC:sender's flow is not udp compressible for TCSB 1000421
00:10:16:   reason: no contexts available
00:10:16:RSVP-TC:Bandwidth check:requested bw=80000 old bw=0
00:10:16:RSVP-TC:RSVP bandwidth is available
00:10:16:RSVP-TC:Consulting policy for tcsb 01000421
00:10:16:RSVP-TC:Policy granted QoS for tcsb 01000421
00:10:16:RSVP-TC:Requesting QoS for tcsb 01000421
00:10:16:RSVP-TC:   ( r = 10000     bytes/s   M = 200       bytes
00:10:16:RSVP-TC:     b = 400       bytes     m = 200       bytes )
00:10:16:RSVP-TC:     p = 10000     bytes/s   Service Level = priority
00:10:16:RSVP-WFQ:Update for tcsb 01000421 on FR PVC dlci 101 on Se3/0
00:10:16:RSVP-WFQ:Admitted 80 kbps of bandwidth
00:10:16:RSVP-WFQ:Allocated PRIORITY queue 24
00:10:16:RSVP-TC:Allocation succeeded for tcsb 01000421
```

Related Commands

Command	Description
show debugging	Displays active debugging output.

debug ip rsvp wfq

To display debugging messages for the weighted fair queue (WFQ), use the **debug ip rsvp wfq** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rsvp wfq

no debug ip rsvp wfq

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.1(3)T	This command was introduced.

Examples The following is an example of output from the **debug ip rsvp wfq** command:

```
Router# debug ip rsvp wfq

RSVP debugging is on

Router# show debugging

IP RSVP debugging is on
IP RSVP debugging (Traffic Control events) is on
IP RSVP debugging (WFQ events) is on
Router#
03:03:23:RSVP-TC:Attempting to install QoS for rsb 6268A538
03:03:23:RSVP-TC:Adding new tcsb 00001A01 for rsb 6268A538
03:03:23:RSVP-TC:Assigning WFQ QoS to tcsb 00001A01
03:03:23:RSVP-TC:Consulting policy for tcsb 00001A01
03:03:23:RSVP-TC:Policy granted QoS for tcsb 00001A01
03:03:23:RSVP-TC:Requesting QoS for tcsb 00001A01
03:03:23:RSVP-TC: ( r = 12500      bytes/s  M = 1514      bytes
03:03:23:RSVP-TC:      b = 1000      bytes      m = 0      bytes )
03:03:23:RSVP-TC:      p = 12500      bytes/s  Service Level = non-priority
03:03:23:RSVP-WFQ:Requesting a RESERVED queue on Et0/1 for tcsb 00001A01
03:03:23:RSVP-WFQ:Queue 265 allocated for tcsb 00001A01
03:03:23:RSVP-TC:Allocation succeeded for tcsb 00001A01
Router#

Router# no debug ip rsvp

RSVP debugging is off
```

Related Commands

Command	Description
show debug	Displays active debugging output.

debug ip rtp header-compression

To display events specific to Real-Time Transport Protocol (RTP) header compression, use the **debug ip rtp header-compression** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rtp header-compression

no debug ip rtp header-compression

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Examples The following is sample output from the **debug ip rtp header-compression** command:

```
Router# debug ip rtp header-compression

RHC BRI0: rcv compressed rtp packet
RHC BRI0: context0: expected sequence 0, received sequence 0
RHC BRI0: rcv compressed rtp packet
RHC BRI0: context0: expected sequence 1, received sequence 1
RHC BRI0: rcv compressed rtp packet
RHC BRI0: context0: expected sequence 2, received sequence 2
RHC BRI0: rcv compressed rtp packet
RHC BRI0: context0: expected sequence 3, received sequence 3
```

[Table 114](#) describes the significant fields shown in the display.

Table 114 *debug ip rtp header-compression Field Descriptions*

Field	Description
context0	Compression state for a connection 0.
expected sequence	RTP header compression link sequence (expected).
received sequence	RTP header compression link sequence (actually received).

Related Commands	Command	Description
	debug ip rtp packets	Displays a detailed dump of packets specific to RTP header compression.

debug ip rtp packets

To display a detailed dump of packets specific to Real-Time Transport Protocol (RTP) header compression, use the **debug ip rtp packets** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rtp packets

no debug ip rtp packets

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Examples

The following is sample output from the **debug ip rtp packets** command:

```
Router# debug ip rtp packets

RTP packet dump:
  IP:  source: 171.68.8.10, destination: 224.2.197.169, id: 0x249B, ttl: 9,
      TOS: 0 prot: 17,
  UDP: source port: 1034, destination port: 27404, checksum: 0xB429, len: 152
  RTP: version: 2, padding: 0, extension: 0, marker: 0,
      payload: 3, ssrc 2369713968,
      sequence: 2468, timestamp: 85187180, csrc count: 0
```

[Table 115](#) describes the significant fields shown in the display.

Table 115 *debug ip rtp packets* Field Descriptions

Field	Description
id	IP identification.
ttl	IP time to live (TTL).
len	Total UDP length.

Related Commands

Command	Description
debug ip rtp header-compression	Displays events specific to RTP header compression.

debug ip scp

To troubleshoot secure copy (SCP) authentication problems, use the **debug ip scp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip scp

no debug ip scp

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.2(2)T	This command was introduced.
	12.0(21)S	This command was integrated into Cisco IOS Release 12.0(21)S and support for the Cisco 7500 series and Cisco 12000 series was added.

Examples The following example is sample output from the **debug ip scp** command. In this example, a copy of the file `scptest.cfg` from a UNIX host to the router's running configuration was successful.

```
Router# debug ip scp

4d06h:SCP: [22 -> 10.11.29.252:1018] send <OK>
4d06h:SCP: [22 <- 10.11.29.252:1018] recv C0644 20 scptest.cfg
4d06h:SCP: [22 -> 10.11.29.252:1018] send <OK>
4d06h:SCP: [22 <- 10.11.29.252:1018] recv 20 bytes
4d06h:SCP: [22 <- 10.11.29.252:1018] recv <OK>
4d06h:SCP: [22 -> 10.11.29.252:1018] send <OK>
4d06h:SCP: [22 <- 10.11.29.252:1018] recv <EOF>
```

The following example is also sample output from the **debug ip scp** command, but in this example, the user has privilege 0 and is therefore denied:

```
Router# debug ip scp

4d06h:SCP: [22 -> 10.11.29.252:1018] send Privilege denied.
```

Related Commands	Command	Description
	ip scp server enable	Enables SCP server-side functionality.