debug aaa sg-server selection through debug vrrp ha

debug aaa sg-server selection through debug vrrp ha

debug aaa sg-server selection

To obtain information about why the RADIUS and TACACS+ server group system in a router is choosing a particular server, use the **debugaaasg-serverselection**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug aaa sg-server selection

no debug aaa sg-server selection

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not turned on.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(1)	This command was introduced.
12.2(28)SB	This command was extended for RADIUS server load balancing to show which server is selected on the basis of a load balancing algorithm.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(33)SRC	This command was integrated into Cisco IOS Release 12.2(33)SRC.

Examples

The following example shows that debugging has been set to display information about server selection:

```
Router# debug aaa sg-server selection
```

The following two debug outputs display the behavior of RADIUS transactions within a server group with the server-reorder-on-failure feature configured.

Examples

In the following sample output, the RADIUS server-reorder-on-failure feature is configured. The server retransmits are set to 0 (so each server is attempted only once before failover to the next configured server), and the transmissions per transaction are set to 4 (the transmissions will stop on the third failover). The third server in the server group (192.0.2.118) has accepted the transaction on the third transmission (second failover).

```
00:38:35: %SYS-5-CONFIG-I: Configured from console by console
00:38:53: RADIUS/ENCODE(OOOOOOOF) : ask "Username: "
00:38:53: RADIUS/ENCODE (0000000F) : send packet; GET-USER
00:38:58: RADIUS/ENCODE (0000000F) : ask "Password: "
00:38:58: RADIUS/ENCODE(0000000F) : send packet; GET-PASSWORD
00:38:59: RADIUS: AAA Unsupported [152] 4
00:38:59: RADIUS: 7474 [tt]
00:38:59: RADIUS (0000000F) : Storing nasport 2 in rad-db
```

```
00:38:59: RADIUS/ENCODE(0000000F) : dropping service type, "radius-server
attribute 6 on-for-login-auth" is off
00:38:59: RADIUS (0000000F) : Config NAS IP: 192.0.2.4
00:38:59: RADIUS/ENCODE (0000000F) : acct-session-id: 15
00:38:59: RADIUS (0000000F) : sending
00:38:59: RADIUS/ENCODE: Best Local IP-Address 192.0.2.130 for Radius-Server 192.0.2.1
00:38:59: RAPIUS(0000000F) : Send Access-Request to 192.0.2.1:1645 id 21645/11, len 78
00:38:59: RADIUS:: authenticator 4481 E6 65 2D 5F 6F OA -lE F5 81 8F 4E 1478 9C
00:38:59: RADIUS: User-Name [1] 7 "username"
00:38:59: RADIUS: User-Password [2] 18 *
00:38:59: RADIUS: NAS-Port fSl 6 2
00:~8:59: RADIUS: NAS-Port-Type [61] 6 Virtual [5]
00:38:59: RADIUS: Calling-Station-Id [31] 15 "192.0.2.23"
00:39:00: RADIUS: NAS-IP-Address [4] 6 192.0.2.130
00:39:02: RADIUS: Fail-over to (192.0.2.2:1645,1646) for id 21645/11
00:39:02: RADIUS/ENCODE: Best Local IP-Address 192.0.2.130 for Radius-Server 192.0.2.2
00:39:04: RADIUS: Fail-over to (192.0.2.118:1645,1646) for id 21645/11
00:39:04: RADIUS/ENCODE: Best Local IP-Address 192.0.2.130 for Radius-Server
192.0.2.118
00:39:05: RADIUS: Received from id 21645/11 192.0.2.118:1645, Access-Accept, len 26
00:39:05: RADIUS: authenticator 5609 56 F9 64 4E DF 19- F3 A2 DD 73 EE 3F 9826
00:39:05: RADIUS: Service-Type [6] 6 Login [1]
```

Examples

In the following sample output, the RADIUS server-reorder-on-failure feature is configured. The server retransmits are set to 0, and the transmissions per transaction are set to 8. In this transaction, the transmission to server 192.0.2.1 has failed on the eighth transmission.

```
00:42:30: RADIUS(00000011): Received from id 21645/13
00:43:34: RADIUS/ENCODE(00000012) : ask "Username: "
00:43:34: RADIUS/ENCODE(00000012) : send packet; GET-USER
00:43:39: RADIUS/ENCODE(00000012) : ask "Password: "
00:43:39: RADIUS/ENCODE(00000012) : send packet; GET-PASSWORD
00:43:40: RADIUS: AAA Unsupported [152] 4
00:43:40: RADIUS: 7474 [tt]
00:43:40: RADIUS(00000012) : Storing nasport 2 in rad-db
00:43:40: RADIUS/ENCODE(00000012): dropping service type, "radius-server attribute 6
on-for-login-auth" is off
00:43:40: RADIUS(00000012) : Co~fig NAS IP: 192.0.2.4
00:43:40: RADIUS/ENCODE(00000012) : acct-session-id: 18
00:43:40: RADIUS(00000012) : sending
00:43:40: RADIUS/ENCODE: Best Local IP-Address 192.0.2.130 for Radius-Server 192.0.2.118
00:43:40: RADIUS(00000012) : Send Access-Request to 192.0.2.118:1645 id 21645/14, len 78
00:43:40: RADIUS: authenticator B8 OA 51 3A AF A6 0018 -B3 2E 94 5E 07 OB 2A
00:43:40: RADIUS: User-Name [1] 7 "username"
00:43:40: RADIUS: User-Password [2] 18 *
00:43:40: RADIUS: NAS-Port [5] 6 2
00:43:40: RADIUS: NAS-Port-Type [61] 6 Virtual [5]
00:43:40: RADIUS: Calling-Station-]d [31] 15 "192.0.2.23"
00:43:40: RADIUS: NAS-IP-Address [4] 6 192.0.2.130
00:43:42: RADIUS: Fail-over to (192.0.2,1:1645,1646) for id 21645/14
00:43:42: RADIUS/ENCODE: Best Local IP-Address 192.0.2.130 for Radius-Server 192.0.2.1
00:43:44: RADIUS: Fail-over to (192.0.2.2:1645,1646) for id 21645/14
00:43:44: RADIUS/ENCODE: Best Local IP-Address 192.0.2.130 for Radius-Server 192.0.2.2
00:43:46: RADIUS: Fail-over to (192.0.2.118:1645,1646) for id 21645/14
00:43:46: RADIUS/ENCODE: Best Local IP-Address 192.0.2.130 for Radius-Server 192.0.2.118
00:43:48: RADIUS: Fail-over to (192.0.2.1:1645,1646) for id 21645/14
00:43:48: RADIUS/ENCODE: Best Local IP-Address 192.0.2.130 for Radius-Server 192.0.2.1
00:43:50: RADIUS: Fail-over to (192.0.2.2:1645,1646) for id 21645/14
00:43:50: RADIUS/ENCODE: Best Local IP-Address 192.0.2.130 for Radius-Server 192.0.2.2
00:43:52: RADIUS: Fail-over to (192.0.2.118:1645,1646) for id 21645/14
00:43:52: RADIUS/ENCODE: Best Local IP-Address 192.0.2.130 for Radius-Server 192.0.2.118
00:43:54: RADIUS: Fail-over to (192.0.2.1:1645,1646) for id 21645/14
00:43:54: RADIUS/ENCODE: Best Local IP-Address 192.0.2.130 for Radius-Server 192.0.2.1
00:43:56: RADIUS: No response from (192.0.2.1:1645,1646) for id 21645/14
00:43:56:RADIUS/DECODE: parse response no app start; FAIL
00:43:56: RADIUS/DECODE: parse response;FAIL
```

The field descriptions are self-explanatory.

Examples

In the following sample output, the RADIUS server load balancing feature is enabled with a batch size of 3. The server selection, based on the load balancing algorithm, is shown as five access-requests that are being sent to the server group.

```
Router# debug aaa sg-server selection
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining least loaded server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: [3] transactions remaining in batch. Reusing server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining least loaded server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: [2] transactions remaining in batch. Reusing server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining least loaded server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: [1] transactions remaining in batch. Reusing server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining least loaded server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: No more transactions in batch. Obtaining a new server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining a new least loaded server.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Server[0] load: 3
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Server[1] load: 0
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Server[2] load: 0
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Selected Server[1] with load 0
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: [3] transactions remaining in batch.
Jul 16 03:15:05: AAA/SG/SERVER_SELECT: Obtaining least loaded server.
```

Jul 16 03:15:05: AAA/SG/SERVER_SELECT: [2] transactions remaining in batch. Reusing server.

The field descriptions are self-explanatory.

Related Commands

Command	Description
load-balance	Enables RADIUS server load balancing for named RADIUS server groups.
radius-server load-balance	Enables RADIUS server load balancing for the global RADIUS server group.
radius-server retry method reorder	Specifies the reordering of RADIUS traffic retries among a server group.
radius-server transaction max-tries	Specifies the maximum number of transmissions per transaction that may be retried on a RADIUS server.
test aaa group	Tests RADIUS load balancing server response manually.

debug aaa test

To show when the idle timer or dead timer has expired, when test packets are being sent, server response status, and the server state for RADIUS server load balancing, use the **debugaaatest**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug aaa test

no debug aaa test

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(28)SB	This command was introduced.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(33)SRC	This command was integrated into Cisco IOS Release 12.2(33)SRC.

Examples

In the following sample output, the RADIUS server load balancing feature is enabled. The idle timer has expired.

```
Router# debug aaa test
Router#
Jul 16 00:07:01: AAA/SG/TEST: Server (192.0.2.245:1700,1701) quarantined.
Jul 16 00:07:01: AAA/SG/TEST: Sending test request(s) to server (192.0.2.245:1700,1701)
Jul 16 00:07:01: AAA/SG/TEST: Sending 1 Access-Requests, 1 Accounting-Requests in current
batch.
Jul 16 00:07:01: AAA/SG/TEST(Req#: 1): Sending test AAA Access-Request.
Jul 16 00:07:01: AAA/SG/TEST(Req#: 1): Sending test AAA Accounting-Request.
Jul 16 00:07:01: AAA/SG/TEST: Obtained Test response from server (192.0.2.245:1700,1701)
Jul 16 00:07:01: AAA/SG/TEST: Obtained Test response from server (192.0.2.245:1700,1701)
Jul 16 00:07:01: AAA/SG/TEST: Necessary responses received from server (192.0.2.245:1700,1701)
Jul 16 00:07:01: AAA/SG/TEST: Server (192.0.2.245:1700,1701) marked ALIVE. Idle timer set
for 60 sec(s).
Jul 16 00:07:01: AAA/SG/TEST: Server (192.0.2.245:1700,1701) removed from quarantine.
```

Related Commands

Command	Description
load-balance	Enables RADIUS server load balancing for named RADIUS server groups.

Command	Description
radius-server host	Enables RADIUS automated testing for load balancing.
radius-server load-balance	Enables RADIUS server load balancing for the global RADIUS server group.
test aaa group	Tests RADIUS load balancing server response manually.

debug authentication

To display debugging information about the Authentication Manager, use the **debugauthentication** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug authentication {[**feature** *feature-name*]| {**all**| **detail**| **errors**| **events**| **sync**}}

no debug authentication {[**feature** *feature-name*]| {**all**| **detail**| **errors**| **events**| **sync**}}

Syntax Description

feature *feature-name*	Displays debugging information about specific features. To display the valid feature names, use the question mark (**?**) online help function.
all	Displays all debugging information about the Authentication Manager and all features.
detail	Displays detailed debugging information.
errors	Displays debugging information about Authentication Manager errors.
events	Displays debugging information about Authentication Manager events.
sync	Displays debugging information about Authentication Manager stateful switchovers (SSOs) or In Service Software Upgrades (ISSUs).

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SXI	This command was introduced.
15.2(2)T	This command was integrated into Cisco IOS Release 15.2(2)T.
Cisco IOS XE Release 3.2SE	This command was modified. The **detail** keyword was added.

Usage Guidelines

Use the **debug authentication** command to troubleshoot the Authentication Manager.

Examples

The following example shows sample output from the **debug authentication** command when the **feature** and **events** keywords are configured:

```
Device# debug authentication feature mda events
Auth Manager mda events debugging is on
```

Related Commands

Command	Description
debug access-session	Displays debugging information about Session Aware Networking sessions.
debug dot1x	Displays 802.1x debugging information.

debug eigrp address-family neighbor

To display debugging information about Enhanced Interior Gateway Routing Protocol (EIGRP) address family neighbors, use the **debug eigrp address-family neighbor** command in privileged EXEC mode. To disable debugging of EIGRP service-family neighbors, use the **no** form of this command.

debug eigrp address-family [**ipv4**| **ipv6**] **neighbor** [*ip-address*]

no debug eigrp address-family [**ipv4**| **ipv6**] **neighbor** [*ip-address*]

Syntax Description

ipv4	(Optional) Enables debugging for neighbors formed using the IPv4 protocol family.
ipv6	(Optional) Enables debugging for neighbors formed using the IPv6 protocol family.
ip-address	(Optional) IPv4 or IPv6 address of the neighbor. Specifying an address enables debugging for the service family at this address.

Command Default

Debugging of EIGRP service-family neighbors is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)M	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.
12.2(33)XNE	This command was integrated into Cisco IOS Release 12.2(33)XNE.
Cisco IOS XE Release 2.5	This command was integrated into Cisco IOS XE Release 2.5.

Usage Guidelines

Consult Cisco technical support before using this command.

Caution Use of **debug** commands can have severe performance penalties and should be used with extreme caution. For this reason, Cisco recommends that you contact Cisco technical support before enabling a **debug** command.

Examples

The following example shows how to enable debugging of an EIGRP address-family neighbor at 10.0.0.0:

```
Router# debug eigrp address-family ipv4 neighbor 10.0.0.0
Neighbor target enabled on AS 3 for 10.0.0.0
*Mar 17 15:50:53.244: EIGRP: 10.0.0.0/24 - do advertise out Serial1/2
*Mar 17 15:50:53.244: EIGRP: Int 10.0.0.0/24 metric 20512000 -20000000 512000
*Mar 17 15:50:53.244: EIGRP: 10.0.0.0/24 - do advertise out Serial1/2
*Mar 17 15:50:53.244: EIGRP: Int 10.0.0.0/24 metric 28160 - 256002560
*Mar 17 15:50:53.244: EIGRP: 10.0.0.0/24 - do advertise out Serial1/2
*Mar 17 15:50:53.244: EIGRP: 10.0.0.0/24 - do advertise out Serial1/2
*Mar 17 15:50:53.244: EIGRP: Int 10.0.0.0/24 metric 28160 - 25600256
*Mar 17 15:50:53.668: EIGRP: Processing incoming UPDATE packet
*Mar 17 15:50:54.544: EIGRP: 10.0.0.0/24 - do advertise out Serial1/1
```

Related Commands

Command	Description
debug eigrp address-family notifications	Displays debugging information about EIGRP event notifications.

debug eigrp address-family notifications

To display debugging information about Enhanced Interior Gateway Routing Protocol (EIGRP) address family event notifications, use the **debug eigrp address-family notifications** command in privileged EXEC mode. To disable EIGRP event notification debugging, use the **no** form of this command.

debug eigrp address-family {**ipv4** [*autonomous-system-number*| **vrf** [*vrf-name*]| *ip-address*]| **ipv6** [*autonomous-system-number*| *ip-address*]} **notifications**

no debug eigrp address-family {**ipv4** [*autonomous-system-number*| **vrf** [*vrf-name*]| *ip-address*]| **ipv6** [*autonomous-system-number*| *ip-address*]} **notifications**

Syntax Description

ipv4	Enables debugging for neighbors formed using the IPv4 protocol family.
ipv6	Enables debugging for neighbors formed using the IPv6 protocol family.
autonomous-system- number	(Optional) Autonomous system number of the EIGRP routing process. If no autonomous system number is specified, debugging information is displayed for all autonomous systems.
vrf	(Optional) Enables debugging for the specified VRF.
vrf-name	(Optional) Name of the VRF address family to which the command is applied.
ip-address	(Optional) IPv4 or IPv6 address of neighbor. Specifying an address enables debugging for all entries with this address.

Command Default

EIGRP event notification debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)M	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.
12.2(33)XNE	This command was integrated into Cisco IOS Release 12.2(33)XNE.

Release	Modification
Cisco IOS XE Release 2.5	This command was integrated into Cisco IOS XE Release 2.5.

Usage Guidelines

Consult Cisco technical support before using this command.

Caution Use of **debug** commands can have severe performance penalties and should be used with extreme caution. For this reason, Cisco recommends that you contact Cisco technical support before enabling a **debug** command.

Examples

The following example shows how to enable EIGRP event notification debugging:

```
Router# debug eigrp address-family ipv4 notifications
*Mar 17 15:58:07.144: IP-EIGRP: Callback: reload_iptable
*Mar 17 15:58:08.148: IP-EIGRP: iptable_redistribute into eigrp AS 1
*Mar 17 15:58:12.144: IP-EIGRP: Callback: redist frm static AS 0 10.0.0.0/24
*Mar 17 15:58:12.144: into: eigrp AS 1 event: 1
*Mar 17 15:58:12.144: IP-EIGRP: Callback: redist frm static AS 0 172.16.0.0/24
*Mar 17 15:58:12.144: into: eigrp AS 1 event: 1
```

Related Commands

Command	Description
debug eigrp address-family neighbor	Displays debugging information about EIGRP service family neighbors.

debug eigrp nsf

To display nonstop forwarding (NSF) events in the console of an NSF-aware or NSF-capable router, use the **debug eigrp nsf** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug eigrp nsf

no debug eigrp nsf

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
15.0(1)M	This command was integrated into Cisco IOS Release 15.0(1)M.
Cisco IOS XE Release 3.6S	This command was modified. Support for IPv6 and IPv6 VPN Routing and Forwarding (VRF) was added.
15.2(2)S	This command was modified. Support for IPv6 and IPv6 VRF was added.

Usage Guidelines

The output from the **debug eigrp nsf** command displays NSF-specific events. The **debug eigrp nsf** command can be issued on either an NSF-capable or an NSF-aware router.

Examples

The following example shows how to enable the Enhanced Interior Gateway Routing Protocol (EIGRP) NSF debugging and display information about neighbor devices:

```
Device# debug eigrp nsf

EIGRP NSF debugging is on
Device# show ip eigrp neighbors detail

EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.2.1                Et1/0             11 00:00:25   10   200  0  5
   Version 5.1/3.0, Retrans: 2, Retries: 0, Prefixes: 1
```

```
   Topology-ids from peer - 0
!
*Sep 23 18:57:19.423: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.1.2.1 (Ethernet1/0)
is resync: peer graceful-restart
*Sep 23 18:57:19.423: EIGRP: NSF: AS100, NSF or GR initiated by 10.1.2.1, flags 0x4:(RS)
*Sep 23 18:57:36.028: EIGRP: NSF: AS100, Receive EOT from 10.1.2.1, Flags 0x8:(EOT)
*Sep 23 18:57:36.028: EIGRP: NSF: route hold timer set to flush stale routes
*Sep 23 18:57:36.038: EIGRP: NSF: AS100. route hold timer expiry
*Sep 23 18:57:36.038: EIGRP: NSF: EIGRP-IPv4: Search for stale routes from 10.1.2.1
!
Device# show ip eigrp neighbors detail

EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.2.1                 Et1/0             11 00:02:31   12   200  0  6
   Time since Restart 00:01:34
   Version 5.1/3.0, Retrans: 2, Retries: 0, Prefixes: 1
   Topology-ids from peer - 0
```

The following sample output is displayed when a router is unable to handle an event with NSF-Awareness:

```
*Jan 23 18:59:56.040: EIGRP: NSF: AS100: Checking if Graceful Restart is possible with
neighbor 1.1.2.1, peer_down reason 'peer restarted'
*Jan 23 18:59:56.040: EIGRP: NSF: Not possible: 'peer_down was called with a HARD resync
flag'
*Jan 23 18:59:56.040: %DUAL-5-NBRCHANGE: EIGRP-IPv6 100: Neighbor 10.1.2.1 (Ethernet1/0)
is down: peer restarted
*Jan 23 19:00:00.170: %DUAL-5-NBRCHANGE: EIGRP-IPv6 100: Neighbor 10.1.2.1 (Ethernet1/0)
is up: new adjacency
*Jan 23 19:00:00.170: EIGRP: NSF: Enqueuing NULL update to 10.1.2.1, flags 0x1:(INIT)
```

debug events

To display events, use the **debug events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug events

no debug events

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

This command displays events that occur on the interface processor and is useful for diagnosing problems in an network. It provides an overall picture of the stability of the network. In a stable network, the **debug events** command does not return any information. If the command generates numerous messages, the messages can indicate the possible source of problems.

When configuring or making changes to a router or interface for, enable the **debug events** command. Doing so alerts you to the progress of the changes or to any errors that might result. Also use this command periodically when you suspect network problems.

Examples

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

```
Router# debug events
RESET(4/0): PLIM type is 1, Rate is 100Mbps
aip_disable(4/0): state=1
config(4/0)
aip_love_note(4/0): asr=0x201
aip_enable(4/0)
aip_love_note(4/0): asr=0x4000
aip_enable(4/0): restarting VCs: 7
aip_setup_vc(4/0): vc:1 vpi:1 vci:1
aip_love_note(4/0): asr=0x200
aip_setup_vc(4/0): vc:2 vpi:2 vci:2
aip_love_note(4/0): asr=0x200
aip_setup_vc(4/0): vc:3 vpi:3 vci:3
aip_love_note(4/0): asr=0x200
aip_setup_vc(4/0): vc:4 vpi:4 vci:4
aip_love_note(4/0): asr=0x200
aip_setup_vc(4/0): vc:6 vpi:6 vci:6
aip_love_note(4/0): asr=0x200
aip_setup_vc(4/0): vc:7 vpi:7 vci:7
aip_love_note(4/0): asr=0x200
aip_setup_vc(4/0): vc:11 vpi:11 vci:11
aip_love_note(4/0): asr=0x200
```

The below table describes the significant fields shown in the display.

Table 1: debug events Field Descriptions

Field	Description
PLIM type	Indicates the interface rate in Mbps. Possible values are: • 1 = TAXI(4B5B) 100 Mbps • 2 = SONET 155 Mbps • 3 = E3 34 Mbps
state	Indicates current state of the ATM Interface Processor (AIP). Possible values are: • 1 = An ENABLE will be issued soon. • 0 = The AIP will remain shut down.
asr	Defines a bitmask, which indicates actions or completions to commands. Valid bitmask values are: • 0x0800 = AIP crashed, reload may be required. • 0x0400 = AIP detected a carrier state change. • 0x0n00 = Command completion status. Command completion status codes are: • n = 8 Invalid physical layer interface module (PLIM) detected • n = 4 Command failed • n = 2 Command completed successfully • n = 1 CONFIG request failed • n = 0 Invalid value

The following line indicates that the AIP was reset. The PLIM detected was 1, so the maximum rate is set to 100 Mbps.

```
RESET(4/0): PLIM type is 1, Rate is 100Mbps
```

The following line indicates that the AIP was given a **shutdown** command, but the current configuration indicates that the AIP should be up:

```
aip_disable(4/0): state=1
```

The following line indicates that a configuration command has been completed by the AIP:

```
aip_love_note(4/0): asr=0x201
```

The following line indicates that the AIP was given a **no shutdown** command to take it out of the shutdown state:

```
aip_enable(4/0)
```

The following line indicates that the AIP detected a carrier state change. It does not indicate that the carrier is down or up, only that it has changed.

```
aip_love_note(4/0): asr=0x4000
```

The following line of output indicates that the AIP enable function is restarting all permanent virtual circuits (PVCs) automatically:

```
aip_enable(4/0): restarting VCs: 7
```

The following lines of output indicate that PVC 1 was set up and a successful completion code was returned:

```
aip_setup_vc(4/0): vc:1 vpi:1 vci:1
aip_love_note(4/0): asr=0x200
```

debug ip eigrp notifications

To display Enhanced Interior Gateway Routing Protocol (EIGRP) events and notifications in the console of the router, use the **debug ip eigrp notifications** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip eigrp notifications

no debug ip eigrp notifications

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
15.0(1)M	This command was integrated into Cisco IOS Release 15.0(1)M.

Usage Guidelines

The output of the debug ip eigrp notifications command displays EIGRP events and notifications.

Examples

The following example output shows that the NSF-aware router has received the restart notification. The NSF-aware router will now wait for end of transmission (EOT) to be sent from the restarting neighbor (NSF-capable).

```
Router# debug ip eigrp notifications
*Oct  4 11:39:18.092:EIGRP:NSF:AS2. Rec RS update from 135.100.10.1,
00:00:00. Wait for EOT.
*Oct  4 11:39:18.092:%DUAL-5-NBRCHANGE:IP-EIGRP(0) 2:Neighbor
135.100.10.1 (POS3/0) is up:peer NSF restarted
```

debug ip http all

To enable debugging output for all HTTP processes on the system, use the **debug ip http all**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip http all

no debug ip http all

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB2.
12.2(33)SRC	This command was integrated into Cisco IOS Release 12.2(33)SRC.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Usage Guidelines

Use this command to enable debugging messages for all HTTP processes and activity. Issuing this command is equivalent to issuing the following commands:

- **debug ip http authentication**
- **debug ip http ezsetup**
- **debug ip http ssi**
- **debug ip http token**
- **debug ip http transaction**
- **debug ip http url**

Examples

For sample output and field descriptions of this command, see the documentation of the commands listed in the "Usage Guidelines" section.

Related Commands

Command	Description
debug ip http authentication	Enables debugging output for all processes for HTTP server and client access.
debug ip http ezsetup	Displays the configuration changes that occur during the EZ Setup process.
debug ip http ssi	Displays SSI translations and SSI ECHO command execution.
debug ip http token	Displays individual tokens parsed by the HTTP server.
debug ip http transaction	Displays HTTP server transaction processing.
debug ip http url	Displays the URLs accessed from the router.

debug ip http client

To enable debugging output for the HTTP client, use the **debug ip http client** command in privileged EXEC mode. To disable debugging output for the HTTP client, use the **no** or **undebug** form of this command.

debug ip http client {**all**| **api**| **cache**| **error**| **main**| **msg**| **socket**}

no debug ip http client {**all**| **api**| **cache**| **error**| **main**| **msg**| **socket**}

undebug ip http client {**all**| **api**| **cache**| **error**| **main**| **msg**| **socket**}

Syntax Description

all	Enables debugging for all HTTP client elements.
api	Enables debugging output for the HTTP client application interface (API).
cache	Enables debugging output for the HTTP client cache.
error	Enables debugging output for HTTP communication errors.
main	Enables debugging output specific to the Voice XML (VXML) applications interacting with the HTTP client.
msg	Enables debugging output of HTTP client messages.
socket	Enables debugging output specific to the HTTP client socket.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(2)T	This command was introduced.
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB2.
12.2(33)SRC	This command was integrated into Cisco IOS Release 12.2(33)SRC.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Usage Guidelines

Use this command to display transactional information for the HTTP client for debugging purposes.

Examples

The following example shows sample debugging output for a failed **copy** transfer operation when the host name resolution fails:

```
Router# debug ip http client all
2w4d: Cache ager called
Router# copy http://www.example.com/index.html flash:index.html

Destination filename [index.html]?
Erase flash: before copying? [confirm] no

Translating "www.example.com"
% Bad IP address for host www.example.com
%Error opening http://www.example.com/index.html (I/O error)
Router#
2w4d: http_client_request:
2w4d: httpc_setup_request:
2w4d: http_client_process_request:
2w4d: HTTPC: Host name resolution failed for www.example.com
2w4d: http_transaction_free:
2w4d: http_transaction_free: freed httpc_transaction_t
```

The following example shows sample debugging output for a failed **copy** transfer operation when the source file is not available:

```
Router# copy http://example.com/hi/file.html flash:/file.html
Destination filename [file.html]?
%Error opening http://example.com/hi/file.html (No such file or directory)
Router#
2w4d: http_client_request:
2w4d: httpc_setup_request:
2w4d: http_client_process_request:
2w4d: httpc_request:Dont have the credentials
Thu, 17 Jul 2003 07:05:25 GMT  http://209.168.200.225/hi/file.html ok
        Protocol = HTTP/1.1
        Content-Type = text/html; charset=iso-8859-1
        Date = Thu, 17 Jul 2003 14:24:29 GMT
2w4d: http_transaction_free:
2w4d: http_transaction_free:freed httpc_transaction_t
2w4d: http_client_abort_request:
2w4d: http_client_abort_request:Bad Transaction Id
Router#
```

The table below describes the significant fields shown in the display.

Table 2: debug ip http client Field Descriptions

Field	Description
2w4d:	In the examples shown, the string “2w4d” is the timestamp configured on the system. Indicates two weeks and four days since the last system reboot. • The time-stamp format is configured using the **service timestamps debug** global configuration mode command.

Field	Description
HTTPC: or httpc	Indicates the HTTP client in Cisco IOS software.
httpc_request:Dont have the credentials	Indicates that this HTTP client request did not supply any authentication information to the server. The authentication information consists of a username and password combination. The message is applicable to both HTTP and HTTPS.
Thu, 17 Jul 2003 07:05:25 GMT http://209.168.200.225/hi/file.html ok	The “ok” in this line indicates that there were no internal errors relating to processing this HTTP client transaction by the HTTP client. In other words, the HTTP client was able to send the request and receive some response. **Note** The “ok” value in this line does not indicate file availability (“200: OK” message or “404: File Not Found” message).

Related Commands

Command	Description
copy	Copies a file from any supported remote location to a local file system, or from a local file system to a remote location, or from a local file system to a local file system.
ip http client connection	Configures the HTTP client connection.
ip http client password	Configures a password for all HTTP client connections.
ip http client proxy-server	Configures an HTTP proxy server.
ip http client source-interface	Configures a source interface for the HTTP client.
ip http client username	Configures a login name for all HTTP client connections.
service timestamps	Configures the time-stamping format for debugging or system logging messages.
show ip http client connection	Displays a report about HTTP client active connections.

Command	Description
show ip http client history	Displays the URLs accessed by the HTTP client.
show ip http client session-module	Displays a report about sessions that have registered with the HTTP client.

debug ip http ssl error

To enable debugging messages for the secure HTTP (HTTPS) web server and client, use the **debug ip http ssl error** command in privileged EXEC mode. To disable debugging messages for the HTTPS web server and client, use the **no** form of this command.

debug ip http ssl error

no debug ip http ssl error

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging message output is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Usage Guidelines

This command displays output for debugging purposes related to the HTTPS server and HTTPS client. HTTPS services use the Secure Socket Layer (SSL) protocol, version 3.0, for encryption.

Examples

The following is sample debugging output from the **debug ip http ssl error** command:

```
Router# 000030:00:08:01:%HTTPS:Key pair generation failed
Router# 000030:00:08:10:%HTTPS:Failed to generate self-signed cert
Router# 000030:00:08:15:%HTTPS:SSL handshake fail
Router# 000030:00:08:21:%HTTPS:SSL read fail, uninitialized hndshk ctxt
Router# 000030:00:08:25:%HTTPS:SSL write fail, uninitialized hndshk ctxt
```

The table below describes the debug messages shown above.

Table 3: debug ip http ssl error Field Descriptions

Field	Description
%HTTPS:Key pair generation failed	The RSA key pair generation failed.
%HTTPS:Failed to generate self-signed cert	The HTTPS server or client failed to generate a self-signed certificate.
%HTTPS:SSL handshake fail	SSL connection handshake failed.
%HTTPS:SSL read fail, uninitialized hndshk ctxt	A read operation failed for SSL with an unitialized handshake context

Related Commands

Command	Description
ip http secure-server	Enables the secure HTTP (HTTPS) server.

debug ip mrouting

To display information about activity in the multicast route (mroute) table, use the **debug ip mrouting** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip mrouting [**vrf** *vrf-name*] [**rpf-events**| **timers**] [*group-address*]

no debug ip mrouting [**vrf** *vrf-name*] [**rpf-events**| **timers**] [*group-address*]

Command Syntax in Cisco IOS 12.2(33)SXH and Subsequent 12.2SX Releases

debug ip mrouting [**vrf** *vrf-name*] [**high-availability**| **rpf-events** [*group-address*]| **timers** *group-address*]

no debug ip mrouting [**vrf** *vrf-name*] [**high-availability**| **rpf-events** [*group-address*]| **timers** *group-address*]

Syntax Description

vrf *vrf-name*	(Optional) Displays debugging information related to mroute activity associated with the Multicast Virtual Private Network (MVPN) routing and forwarding (MVRF) instance specified for the *vrf-name* argument.
high-availability	(Optional) Displays high availability (HA) events associated with supervisor engine switchovers on Catalyst 6500 series switches, in Cisco IOS Release 12.2(33)SXH and subsequent 12.2SX releases.
rpf-events	(Optional) Displays Reverse Path Forwarding (RPF) events associated with mroutes in the mroute table.
timers	(Optional) Displays timer-related events associated with mroutes in the mroute table.
group-address	(Optional) IP address or Domain Name System (DNS) name of a multicast group. Entering a multicast group address restricts the output to only display mroute activity associated with the multicast group address specified for the optional *group-address* argument.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
10.2	This command was introduced.
12.0(22)S	The **rpf-events** keyword was added.

Release	Modification
12.2(13)T	The **timers** keyword, **vrf** keyword, and *vrf-name* argument were added.
12.2(14)S	The **timers** keyword, **vrf** keyword, and *vrf-name* argument were added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH. The **high-availability** keyword was added in support of the PIM Triggered Joins feature.

Usage Guidelines

This command indicates when the router has made changes to the mroute table. Use the **debug ip pim** and **debug ip mrouting** commands consecutively to obtain additional multicast routing information. In addition, use the **debug ip igmp** command to learn why an mroute message is being displayed.

This command generates a substantial amount of output. Use the optional *group-address* argument to limit the output to a single multicast group.

In Cisco IOS 12.2(33)SXH and subsequent 12.2SX releases, the **high-availability** keyword was added in support of the PIM Triggered Joins feature to monitor HA events in the event of a supervisor engine switchover on a Catalyst 6500 series switch. The PIM Triggered Joins feature is an HA multicast enhancement that improves the reconvergence of mroutes after a supervisor engine switchover on a Catalyst 6500 series switch. After a service engine switchover, all instances of PIM running on the newly active supervisor engine will modify the value of the Generation ID (GenID) that is included in PIM hello messages sent to adjacent PIM neighbors. When an adjacent PIM neighbor receives a PIM hello message on an interface with a new GenID, the PIM neighbor will interpret the modified GenID as an indication that all mroutes states on that interface have been lost. A modified GenID, thus, is utilized as a mechanism to alert all adjacent PIM neighbors that PIM forwarding on that interface has been lost, which then triggers adjacent PIM neighbors to send PIM joins for all (*, G) and (S, G) mroute states that use that interface as an RPF interface.

Examples

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

```
Router# debug ip mrouting 224.2.0.1

MRT: Delete (10.0.0.0/8, 224.2.0.1)
MRT: Delete (10.4.0.0/16, 224.2.0.1)
MRT: Delete (10.6.0.0/16, 224.2.0.1)
MRT: Delete (10.9.0.0/16, 224.2.0.1)
MRT: Delete (10.16.0.0/16, 224.2.0.1)
MRT: Create (*, 224.2.0.1), if_input NULL
MRT: Create (224.69.15.0/24, 225.2.2.4), if_input Ethernet0, RPF nbr 224.69.61.15
MRT: Create (224.69.39.0/24, 225.2.2.4), if_input Ethernet1, RPF nbr 0.0.0.0
MRT: Create (10.0.0.0/8, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0
MRT: Create (10.4.0.0/16, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0
MRT: Create (10.6.0.0/16, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0
MRT: Create (10.9.0.0/16, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0
MRT: Create (10.16.0.0/16, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0
```

The following lines show that multicast IP routes were deleted from the routing table:

```
MRT: Delete (10.0.0.0/8, 224.2.0.1)
```

```
MRT: Delete (10.4.0.0/16, 224.2.0.1)
MRT: Delete (10.6.0.0/16, 224.2.0.1)
```

The (*, G) entries are generally created by receipt of an Internet Group Management Protocol (IGMP) host report from a group member on the directly connected LAN or by a Protocol Independent Multicast (PIM) join message (in sparse mode) that this router receives from a router that is sending joins toward the Route Processor (RP). This router will in turn send a join toward the RP that creates the shared tree (or RP tree).

```
MRT: Create (*, 224.2.0.1), if_input NULL
```

The following lines are an example of creating an (S, G) entry that shows that an IP multicast packet (mpacket) was received on Ethernet interface 0. The second line shows a route being created for a source that is on a directly connected LAN. The RPF means “Reverse Path Forwarding,” whereby the router looks up the source address of the multicast packet in the unicast routing table and determines which interface will be used to send a packet to that source.

```
MRT: Create (224.69.15.0/24, 225.2.2.4), if_input Ethernet0, RPF nbr 224.69.61.15
MRT: Create (224.69.39.0/24, 225.2.2.4), if_input Ethernet1, RPF nbr 0.0.0.0
```

The following lines show that multicast IP routes were added to the routing table. Note the 224.0.0.0 as the RPF, which means the route was created by a source that is directly connected to this router.

```
MRT: Create (10.9.0.0/16, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0
MRT: Create (10.16.0.0/16, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0
```

If the source is not directly connected, the neighbor address shown in these lines will be the address of the router that forwarded the packet to this router.

The shortest path tree state maintained in routers consists of source (S), multicast address (G), outgoing interface (OIF), and incoming interface (IIF). The forwarding information is referred to as the multicast forwarding entry for (S, G).

An entry for a shared tree can match packets from any source for its associated group if the packets come through the proper incoming interface as determined by the RPF lookup. Such an entry is denoted as (*, G). A (*, G) entry keeps the same information a (S, G) entry keeps, except that it saves the rendezvous point address in place of the source address in sparse mode or as 24.0.0.0 in dense mode.

The table below describes the significant fields shown in the display.

Table 4: debug ip mrouting Field Descriptions

Field	Description
MRT	Multicast route table.
RPF	Reverse Path Forwarding.
nbr	Neighbor.

Related Commands

Command	Description
debug ip dvmrp	Displays information on DVMRP packets received and sent.

Command	Description
debug ip igmp	Displays IGMP packets received and sent, and IGMP host-related events.
debug ip packet	Displays general IP debugging information and IPSO security transactions.
debug ip pim	Displays all PIM announcements received.
debug ip sd	Displays all SD announcements received.

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.
vrf-name	(Optional) Name assigned to the VRF.
peer-address | *name*	(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.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(7)T	This command was introduced.
12.0(23)S	The **vrf** keyword and *vrf-name* argument were added.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

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
```

The table below describes the significant fields shown in the display.

Table 5: 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.
vrf-name	(Optional) Name assigned to the VRF.

Command Default

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 *vrf-name* argument were added.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

debug ip ospf nsf

To display debugging messages about Open Shortest Path First (OSPF) during a Cisco nonstop forwarding (NSF) restart, use the **debug ip ospf nsf** command in privileged EXEC mode. To disable the display of debugging output, use the **no** form of this command.

debug ip ospf nsf [detail]

no debug ip ospf nsf [detail]

Syntax Description

detail	(Optional) Displays detailed debug messages.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(22)S	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.2(20)S	Support for the Cisco 7304 router was added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

Use the **debug ip ospf nsf** command to diagnose problems with OSPF link-state database (LSDB) resynchronization and NSF operations.

Examples

The following example shows that OSPF NSF events debugging is enabled:

```
Router# debug ip ospf nsf
```

Related Commands

Command	Description
nsf (OSPF)	Configures NSF operations for OSPF.
show ip ospf	Displays general information about OSPF routing processes.

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

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 *vrf-name*	(Optional) Displays PIM-related events associated with the Multicast Virtual Private Network (MVPN) routing and forwarding (MVRF) instance specified for the *vrf-name* argument.
group-address	(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 *group-address* 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.
rp-address	(Optional) The rendezvous point IP address.
hello	(Optional) Displays events associated with PIM hello messages.
tag	(Optional) Displays tag-switching-related activity.

Command Default

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, *vrf-name* argument, and **hello** keyword were added.
12.2(13)T	The **vrf** keyword and *vrf-name* 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.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

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 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 outut, use the **no** form of this command.

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

no debug ip rgmp

Syntax Description

group-name	(Optional) The name of a specific IP multicast group.
group-address	(Optional) The IP address of a specific IP multicast group.

Command Default

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.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following shows sample output from 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.

Command	Description
show ip igmp interface	Displays multicast-related information about an interface.

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.
12.2(22)S	This command was integrated into Cisco IOS Release 12.2(22)S.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.

Examples

The following example is output from the **debug ip scp** command. In this example, a copy of the file scptest.cfg from a UNIX host running configuration of the router 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 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.

debug ip ssh

To display debugging messages for Secure Shell (SSH), use the **debug ip ssh** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip ssh [**detail**| **packet**]

no debug ip ssh

Syntax Description

detail	(Optional) Specifies SSH protocol, channel requests and information state changes.
packet	(Optional) Specifies information regarding the SSH packet.

Command Default

Debugging for SSH is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(1)T	This command was integrated into Cisco IOS Release 12.1T.
12.4(20)T	The detail and packet keywords were added.
Cisco IOS XE Release 2.4	This command was implemented on the Cisco ASR 1000 series routers.

Usage Guidelines

Use the **debug ip ssh** command to ensure normal operation of the SSH server.

Examples

The following example shows the SSH debugging output:

```
Router# debug ip ssh
00:53:46: SSH0: starting SSH control process
00:53:46: SSH0: Exchanging versions - SSH-1.5-Cisco-1.25
00:53:46: SSH0: client version is - SSH-1.5-1.2.25
00:53:46: SSH0: SSH_SMSG_PUBLIC_KEY message sent
00:53:46: SSH0: SSH_CMSG_SESSION_KEY message received
00:53:47: SSH0: keys exchanged and encryption on
00:53:47: SSH0: authentication request for userid guest
00:53:47: SSH0: authentication successful for jcisco
00:53:47: SSH0: starting exec shell
```

The following example shows the SSH detail output:

```
Router# debug ip ssh detail
00:04:22: SSH0: starting SSH control process
00:04:22: SSH0: sent protocol version id SSH-1.99-Cisco-1.25
00:04:22: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
00:04:22: SSH2 0: SSH2_MSG_KEXINIT sent
00:04:22: SSH2 0: SSH2_MSG_KEXINIT received
00:04:22: SSH2:kex: client->server enc:aes128-cbc mac:hmac-sha1
00:04:22: SSH2:kex: server->client enc:aes128-cbc mac:hmac-sha1
00:04:22: SSH2 0: expecting SSH2_MSG_KEXDH_INIT
00:04:22: SSH2 0: SSH2_MSG_KEXDH_INIT received
00:04:22: SSH2: kex_derive_keys complete
00:04:22: SSH2 0: SSH2_MSG_NEWKEYS sent
00:04:22: SSH2 0: waiting for SSH2_MSG_NEWKEYS
00:04:22: SSH2 0: SSH2_MSG_NEWKEYS received
00:04:24: SSH2 0: authentication successful for lab
00:04:24: SSH2 0: channel open request
00:04:24: SSH2 0: pty-req request
00:04:24: SSH2 0: setting TTY - requested: height 24, width 80; set: height 24, width 80
00:04:24: SSH2 0: shell request
00:04:24: SSH2 0: shell message received
00:04:24: SSH2 0: starting shell for vty
00:04:38: SSH0: Session terminated normally
```

The following example shows the SSH packet output:

```
Router# debug ip ssh packet
00:05:43: SSH2 0: send:packet of  length 280 (length also includes padlen of 4)
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: input: total packet length of 280 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 24 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 4 bytes
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: input: total packet length of 144 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 16 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 6 bytes
00:05:43: SSH2 0: signature length 143
00:05:43: SSH2 0: send:packet of  length 448 (length also includes padlen of 7)
00:05:43: SSH2 0: send:packet of  length 16 (length also includes padlen of 10)
00:05:43: SSH2 0: newkeys: mode 1
00:05:43: SSH2 0: ssh_receive: 16 bytes received
00:05:43: SSH2 0: input: total packet length of 16 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 8 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 10 bytes
00:05:43: SSH2 0: newkeys: mode 0
00:05:43: SSH2 0: ssh_receive: 52 bytes received
00:05:43: SSH2 0: input: total packet length of 32 bytes
00:05:43: SSH2 0: partial packet length(block size)16 bytes,needed 16 bytes, maclen 20
00:05:43: SSH2 0: MAC compared for #3 :ok
```

debug ipv6 snooping

To enable debugging for security snooping information in IPv6, use the **debug ipv6 snooping**command in privileged EXEC mode.

debug ipv6 snooping [**binding-table**| **classifier**| **errors**| **feature-manager**| **filter** *acl*| **ha**| **hw-api**| **interface** *interface*| **memory**| **ndp-inspection**| **policy**| **vlan** *vlanid*| **switcher**| **filter** *acl*| **interface** *interface*| *vlanid*]

no debug ipv6 snooping

Syntax Description

binding-table	(Optional) Displays information about the neighbor binding table.
classifier	(Optional) Displays information about the classifier.
errors	(Optional) Displays information about snooping security errors.
feature-manager	(Optional) Displays feature manager information.
filter *acl*	(Optional) Allows users to configure an access list to filter debugged traffic.
ha	(Optional) Displays information about high availability (HA) and stateful switchover (SSO).
hw-api	(Optional) Displays information about the hardware API.
interface *interface*	(Optional) Provides debugging information on a specified interface.
memory	(Optional) Displays information about security snooping memory.
ndp-inspection	(Optional) Displays information about Neighbor Discovery inspection.
policy	(Optional)
switcher	(Optional) Displays packets handled by the switcher.
vlanid	(Optional) Provides debugging information about a specified VLAN ID.

Command Modes

Privileged EXEC (#)

Command History

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

Usage Guidelines

The **debug ipv6 snooping**command provides debugging output for IPv6 snooping information.

Because debugging output is assigned high priority in the CPU process, you should use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff.

Examples

The following example enables debugging for all IPv6 snooping information:

```
Router# debug ipv6 snooping
```

debug ipv6 snooping raguard

To enable debugging for security snooping information in the IPv6 router advertisement (RA) guard feature, use the **debug ipv6 snooping raguard** command in privileged EXEC mode.

debug ipv6 snooping raguard [*filter*| *interface*| *vlanid*]

no debug ipv6 snooping raguard

Syntax Description

filter	(Optional) Allows users to configure an access list to filter debugged traffic.
interface	(Optional) Provides debugging information about a specified interface configured with the IPv6 RA guard feature.
vlanid	(Optional) Provides debugging information about a specified VLAN ID configured with the IPv6 RA guard feature.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(54)SG	This command was introduced.
12.2(50)SY	This command was integrated into Cisco IOS Release 12.2(50)SY.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

The **debug ipv6 snooping raguard** command provides debugging output for IPv6 RA guard events and errors that may occur.

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

Examples

The following example shows the command enabling debugging for the IPv6 RA guard feature:

```
Router# debug ipv6 snooping raguard
```

Related Commands

Command	Description
ipv6 nd raguard	Applies the IPv6 RA guard feature.

debug lacp

To enable debugging of all Link Aggregation Control Protocol (LACP) activity, use the **debug lacp**command in privileged EXEC mode. To disable LACP debugging, use the **no** form of this command.

debug lacp [**all**| **event**| **fsm**| **misc**| **multi-chassis** [**all**| **database**| **lacp-mgr**| **redundancy-group**| **user-interface**]| **packet**]

no debug lacp [**all**| **event**| **fsm**| **misc**| **multi-chassis** [**all**| **database**| **lacp-mgr**| **redundancy-group**| **user-interface**]| **packet**]

Syntax Description

all	(Optional) Activates debugging for all LACP operations.
event	(Optional) Activates debugging of events that occur within LACP.
fsm	(Optional) Activates debugging for changes within the LACP finite state machine.
misc	(Optional) Activates debugging for various operations that may be useful for monitoring the status of LACP.
multi-chassis	(Optional) Activates multi-chassis LACP (mLACP) debugging.
all	(Optional) Activates all mLACP debugging.
database	(Optional) Activates mLACP database debugging.
lacp-mgr	(Optional) Activates mLACP interface debugging.
redundancy-group	(Optional) Activates mLACP interchassis redundancy group debugging.
user-interface	(Optional) Activates mLACP interchassis user interface debugging.
packet	(Optional) Displays the receiving and transmitting LACP control packets.

Command Default

LACP debugging activity is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.1(13)EW	Support for this command was introduced on the Cisco Catalyst 4500 series switch.
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB.
12.2(33)SRB	Support for this command on the Cisco 7600 router was integrated into Cisco IOS Release 12.2(33)SRB.
Cisco IOS XE Release 2.4	This command was integrated into Cisco IOS XE Release 2.4.
12.2(33)SRE	This command was modified. The following keywords were added: **multi-chassis**, **all**, **database**, **lacp-mgr**, **redundancy-group**, and **user-interface**.

Usage Guidelines

This command is useful for troubleshooting problems with LACP.

Examples

The following sample output from the **debug lacp all** command shows LACP activity on a port-channel member link Gigabit Ethernet 5/0/0:

```
Router# debug lacp all
Link Aggregation Control Protocol all debugging is on
Router1#
*Aug 20 17:21:51.685: LACP :lacp_bugpak: Receive LACP-PDU packet via Gi5/0/0
*Aug 20 17:21:51.685: LACP : packet size: 124
*Aug 20 17:21:51.685: LACP: pdu: subtype: 1, version: 1
*Aug 20 17:21:51.685: LACP: Act: tlv:1, tlv-len:20, key:0x1, p-pri:0x8000, p:0x14,
p-state:0x3C,
s-pri:0xFFFF, s-mac:0011.2026.7300
*Aug 20 17:21:51.685: LACP: Part: tlv:2, tlv-len:20, key:0x5, p-pri:0x8000, p:0x42,
p-state:0x3D,
s-pri:0x8000, s-mac:0014.a93d.4a00
*Aug 20 17:21:51.685: LACP: col-tlv:3, col-tlv-len:16, col-max-d:0x8000
*Aug 20 17:21:51.685: LACP: term-tlv:0 termr-tlv-len:0
*Aug 20 17:21:51.685: LACP: Gi5/0/0 LACP packet received, processing
*Aug 20 17:21:51.685:     lacp_rx Gi5: during state CURRENT, got event 5(recv_lacpdu)
*Aug 20 17:21:59.869: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:21:59.869: LACP: lacp_p(Gi5/0/0) expired
*Aug 20 17:21:59.869:     lacp_ptx Gi5: during state SLOW_PERIODIC, got event 3(pt_expired)
*Aug 20 17:21:59.869: @@@ lacp_ptx Gi5: SLOW_PERIODIC -> PERIODIC_TX
*Aug 20 17:21:59.869: LACP: Gi5/0/0 lacp_action_ptx_slow_periodic_exit entered
*Aug 20 17:21:59.869: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:00.869: LACP: lacp_t(Gi5/0/0) timer stopped
*Aug 20 17:22:00.869: LACP: lacp_t(Gi5/0/0) expired
*Aug 20 17:22:19.089: LACP :lacp_bugpak: Receive LACP-PDU packet via Gi5/0/0
*Aug 20 17:22:19.089: LACP : packet size: 124
*Aug 20 17:22:19.089: LACP: pdu: subtype: 1, version: 1
*Aug 20 17:22:19.089: LACP: Act: tlv:1, tlv-len:20, key:0x1, p-pri:0x8000, p:0x14,
p-state:0x4,
s-pri:0xFFFF, s-mac:0011.2026.7300
*Aug 20 17:22:19.089: LACP: Part: tlv:2, tlv-len:20, key:0x5, p-pri:0x8000, p:0x42,
p-state:0x34,
s-pri:0x8000, s-mac:0014.a93d.4a00
*Aug 20 17:22:19.089: LACP: col-tlv:3, col-tlv-len:16, col-max-d:0x8000
*Aug 20 17:22:19.089: LACP: term-tlv:0 termr-tlv-len:0
*Aug 20 17:22:19.089: LACP: Gi5/0/0 LACP packet received, processing
```

```
*Aug 20 17:22:19.089:     lacp_rx Gi5: during state CURRENT, got event 5(recv_lacpdu)
*Aug 20 17:22:19.989: LACP: lacp_t(Gi5/0/0) timer stopped
*Aug 20 17:22:19.989: LACP: lacp_t(Gi5/0/0) expired
*Aug 20 17:22:19.989: LACP: timer lacp_t(Gi5/0/0) started with interval 1000.
*Aug 20 17:22:19.989: LACP: lacp_send_lacpdu: (Gi5/0/0) About to send the 110 LACPDU
*Aug 20 17:22:19.989: LACP :lacp_bugpak: Send LACP-PDU packet via Gi5/0/0
*Aug 20 17:22:19.989: LACP : packet size: 124
*Aug 20 17:22:20.957: LACP: lacp_t(Gi5/0/0) timer stopped
*Aug 20 17:22:20.957: LACP: lacp_t(Gi5/0/0) expired
*Aug 20 17:22:21.205: %LINK-3-UPDOWN: Interface GigabitEthernet5/0/0, changed state to down
*Aug 20 17:22:21.205: LACP: lacp_hw_off: Gi5/0/0 is going down
*Aug 20 17:22:21.205: LACP: if_down: Gi5/0/0
*Aug 20 17:22:21.205:    lacp_ptx Gi5: during state SLOW_PERIODIC, got event 0(no_periodic)
*Aug 20 17:22:22.089: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel5, changed
 state to down
*Aug 20 17:22:22.153: %C10K_ALARM-6-INFO: CLEAR CRITICAL GigE 5/0/0 Physical Port Link Down

*Aug 20 17:22:23.413: LACP: Gi5/0/0 oper-key: 0x0
*Aug 20 17:22:23.413: LACP: lacp_hw_on: Gi5/0/0 is coming up
*Aug 20 17:22:23.413:     lacp_ptx Gi5: during state NO_PERIODIC, got event 0(no_periodic)
*Aug 20 17:22:23.413: @@@ lacp_ptx Gi5: NO_PERIODIC -> NO_PERIODIC
*Aug 20 17:22:23.413: LACP: Gi5/0/0 lacp_action_ptx_no_periodic entered
*Aug 20 17:22:23.413: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:24.153: %LINK-3-UPDOWN: Interface GigabitEthernet5/0/0, changed state to up
*Aug 20 17:22:24.153: LACP: lacp_hw_on: Gi5/0/0 is coming up
*Aug 20 17:22:24.153:    lacp_ptx Gi5: during state FAST_PERIODIC, got event 0(no_periodic)
*Aug 20 17:22:24.153: @@@ lacp_ptx Gi5: FAST_PERIODIC -> NO_PERIODIC
*Aug 20 17:22:24.153: LACP: Gi5/0/0 lacp_action_ptx_fast_periodic_exit entered
*Aug 20 17:22:24.153: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:24.153: LACP:
*Aug 20 17:22:25.021: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:25.021: LACP: lacp_p(Gi5/0/0) expired
*Aug 20 17:22:25.021:    lacp_ptx Gi5: during state FAST_PERIODIC, got event 3(pt_expired)
*Aug 20 17:22:25.021: @@@ lacp_ptx Gi5: FAST_PERIODIC -> PERIODIC_TX
*Aug 20 17:22:25.021: LACP: Gi5/0/0 lacp_action_ptx_fast_periodic_exit entered
*Aug 20 17:22:25.021: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:25.917: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:25.917: LACP: lacp_p(Gi5/0/0) expired
*Aug 20 17:22:25.917:    lacp_ptx Gi5: during state FAST_PERIODIC, got event 3(pt_expired)
*Aug 20 17:22:25.917: @@@ lacp_ptx Gi5: FAST_PERIODIC -> PERIODIC_TX
*Aug 20 17:22:25.917: LACP: Gi5/0/0 lacp_action_ptx_fast_periodic_exit entered
*Aug 20 17:22:25.917: LACP: lacp_p(Gi5/0/0) timer stopped
Router1#
```

debug ntp

To display debugging messages for Network Time Protocol (NTP) features, use the **debug ntp**command in prvileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ntp {**adjust**| **all**| **authentication**| **core**| **events**| **loopfilter**| **packet**| **params**| **refclock**| **select**| **sync**| **validity**}

no debug ntp {**adjust**| **all**| **authentication**| **core**| **events**| **loopfilter**| **packet**| **params**| **refclock**| **select**| **sync**| **validity**}

Syntax Description

adjust	Displays debugging information on NTP clock adjustments.
all	Displays all debugging information on NTP.
authentication	Displays debugging information on NTP authentication.
core	Displays debugging information on NTP core messages.
events	Displays debugging information on NTP events.
loopfilter	Displays debugging information on NTP loop filters.
packet	Displays debugging information on NTP packets.
params	Displays debugging information on NTP clock parameters.
refclock	Displays debugging information on NTP reference clocks.
select	Displays debugging information on NTP clock selection.
sync	Displays debugging information on NTP clock synchronization.
validity	Displays debugging information on NTP peer clock validity.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.1	This command was introduced in a release prior to Cisco IOS Release 12.1.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(20)T	Support for IPv6 and NTP version 4 was added. The **all** and **core** keywords were added. The **authentication**, **loopfilter**, **params**, **select**, **sync** and **validity** keywords were removed. The **packets** keyword was modified as **packet**.
Cisco IOS XE Release 3.5S	This command was integrated into Cisco IOS XE Release 3.5S.
Cisco IOS Release 15.2(1)S	This command was integrated into Cisco IOS Release 15.2(1)S.

Usage Guidelines

Starting from Cisco IOS Release 12.4(20)T, NTP version 4 is supported. In NTP version 4 the debugging options available are **adjust**, **all**, **core**, **events**, **packet**, and **refclock**. In NTP version 3 the debugging options available were **events**, **authentication**, **loopfilter**, **packets**, **params**, **select**, **sync** and **validity**.

Examples

The following example shows how to enable all debugging options for NTP:

```
Router# debug ntp all
NTP events debugging is on
NTP core messages debugging is on
NTP clock adjustments debugging is on
NTP reference clocks debugging is on
NTP packets debugging is on
```

Related Commands

Command	Description
ntp refclock	Configures an external clock source for use with NTP services.

debug radius

To enable debugging for Remote Authentication Dial-In User Service (RADIUS) configuration, use the **debug radius** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug radius [**accounting**| **authentication**| **brief**| **elog**| **failover**| **retransmit**| **verbose**]

no debug radius [**accounting**| **authentication**| **brief**| **elog**| **failover**| **retransmit**| **verbose**]

Syntax Description

accounting	(Optional) Enables debugging of RADIUS accounting collection.
authentication	(Optional) Enables debugging of RADIUS authentication packets.
brief	(Optional) Displays abbreviated debug output.
elog	(Optional) Enables RADIUS event logging.
failover	(Optional) Enables debugging of packets sent upon failover.
retransmit	(Optional) Enables retransmission of packets.
verbose	(Optional) Displays detailed debug output.

Command Default

RADIUS event logging and debugging output in ASCII format are enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.2(1)T	This command was introduced.
12.0(2)T	The **brief** keyword was added. The default output format became ASCII from hexadecimal.
12.2(11)T	The **verbose** keyword was added.
12.3(2)T	The **elog** keyword was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Release	Modification
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

RADIUS is a distributed security system that secures networks against unauthorized access. Cisco supports RADIUS under the authentication, authorization, and accounting (AAA) security system. When RADIUS is used on the router, you can use the **debug radius** command to display debugging and troubleshooting information in ASCII format. Use the **debug radius brief** command for abbreviated output displaying client/server interaction and minimum packet information. Only the input and output transactions are recorded. Use the **debug radius verbose** command to include non-essential RADIUS debugs.

Examples

The following is sample output from the **debug radius** command:

```
Router# debug radius
Radius protocol debugging is on
Radius packet hex dump debugging is off
Router# show debug
00:19:20: RADIUS/ENCODE(00000015):Orig. component type = AUTH_PROXY
00:19:20: RADIUS(00000015): Config NAS IP: 0.0.0.0
00:19:20: RADIUS/ENCODE(00000015): acct_session_id: 21
00:19:20: RADIUS(00000015): sending
00:19:20: RADIUS/ENCODE: Best Local IP-Address 33.0.0.2 for Radius-Server 33.2.0.1
00:19:20: RADIUS(00000015): Send Access-Request to 33.2.0.1:1645 id 1645/21, len 159
00:19:20: RADIUS:  authenticator 2D 03 E5 A6 A5 30 1A 32 - F2 C5 EE E2 AC 5E 5D 22
00:19:20: RADIUS:  User-Name           [1]   11  "authproxy"
00:19:20: RADIUS:  User-Password       [2]   18  *
00:19:20: RADIUS:  Service-Type        [6]   6   Outbound                  [5]
00:19:20: RADIUS:  Message-Authenticato[80]  18
00:19:20: RADIUS:   85 EF E8 43 03 88 58 63 78 D2 7B E7 26 61 D3 3C          [ CXcx{&a<]
00:19:20: RADIUS:  Vendor, Cisco       [26]  49
00:19:20: RADIUS:   Cisco AVpair      [1]   43  "audit-session-id=0D000002000000130011112FD"
00:19:20: RADIUS:  NAS-Port-Type       [61]  6   Ethernet                  [15]
00:19:20: RADIUS:  NAS-Port            [5]   6   16480
00:19:20: RADIUS:  NAS-Port-Id         [87]  19  "FastEthernet1/0/3"
00:19:20: RADIUS:  NAS-IP-Address      [4]   6   33.0.0.2
00:19:20: RADIUS(00000015): Started 5 sec timeout
00:19:20: RADIUS: Received from id 1645/21 33.2.0.1:1645, Access-Accept, len 313
00:19:20: RADIUS:  authenticator E6 6E 1D 64 5A 15 FD AE - C9 60 C0 68 F5 10 E9 B7
00:19:20: RADIUS:  Filter-Id           [11]  8
00:19:20: RADIUS:   31 30 30 2E 69 6E           [ 100.in]
00:19:20: RADIUS:  Vendor, Cisco       [26]  19
00:19:20: RADIUS:   Cisco AVpair       [1]   13  "priv-lvl=15"
00:19:20: RADIUS:  Termination-Action  [29]  6   1
00:19:20: RADIUS:  Vendor, Cisco       [26]  45
00:19:20: RADIUS:   Cisco AVpair       [1]   39  "supplicant-name=Port-description test"
00:19:20: RADIUS:  Vendor, Cisco       [26]  38
00:19:20: RADIUS:   Cisco AVpair       [1]   32  "security-group-tag=2468-C0FFEE"
00:19:20: RADIUS:  Vendor, Cisco       [26]  33
00:19:20: RADIUS:   Cisco AVpair       [1]   27  "supplicant-group=engineer"
00:19:20: RADIUS:  Vendor, Cisco       [26]  36
00:19:20: RADIUS:   Cisco AVpair       [1]   30  "supplicant-group=idf_testing"
00:19:20: RADIUS:  Vendor, Cisco       [26]  28
00:19:20: RADIUS:   Cisco AVpair       [1]   22  "authz-directive=open"
00:19:20: RADIUS:  Vendor, Cisco       [26]  32
00:19:20: RADIUS:   Cisco AVpair       [1]   26  "supplicant-group=group-9"
00:19:20: RADIUS:  Class               [25]  30
00:19:20: RADIUS:   43 41 43 53 3A 63 2F 61 37 31 38 38 61 2F 32 31  [CACS:c/a7188a/21]
00:19:20: RADIUS:   30 30 30 30 30 32 2F 31 36 34 38 30      [ 000002/16480]
00:19:20: RADIUS:  Message-Authenticato[80]  18
```

```
00:19:20: RADIUS:   24 13 29 95 A1 5E 9F D3 CB ED 78 F1 F6 62 2B E3           [ $)^xb+]
00:19:20: RADIUS(00000015): Received from id 1645/21
00:19:20: RADIUS/DECODE: parse unknown cisco vsa "supplicant-group" - IGNORE
00:19:20: RADIUS/DECODE: parse unknown cisco vsa "supplicant-group" - IGNORE
00:19:20: RADIUS/DECODE: parse unknown cisco vsa "authz-directive" - IGNORE
00:19:20: RADIUS/DECODE: parse unknown cisco vsa "supplicant-group" - IGNORE
00:19:20: RADIUS/ENCODE(00000015):Orig. component type = AUTH_PROXY
00:19:20: RADIUS(00000015): Config NAS IP: 0.0.0.0
00:19:20: RADIUS(00000015): sending
00:19:20: RADIUS/ENCODE: Best Local IP-Address 33.0.0.2 for Radius-Server 33.2.0.1
00:19:20: RADIUS(00000015): Send Accounting-Request to 33.2.0.1:1646 id 1646/1, len 204
00:19:20: RADIUS:  authenticator A7 6B A0 94 F4 63 30 51 - 8A CE 8C F4 8A 8E 0B CC
00:19:20: RADIUS:  Acct-Session-Id     [44]  10  "00000015"
00:19:20: RADIUS:  Calling-Station-Id  [31]  10  "13.1.0.1"
00:19:20: RADIUS:  Vendor, Cisco       [26]  49
00:19:20: RADIUS:   Cisco AVpair      [1]   43  "audit-session-id=0D000002000000013001112FD"
```

The following is sample output from the **debug radius brief** command:

```
Router# debug radius brief
Radius protocol debugging is on
Radius packet hex dump debugging is off
Radius protocol in brief format debugging is on
00:05:21: RADIUS: Initial Transmit ISDN 0:D:23 id 6 10.0.0.1:1824, Accounting-Request, len
 358
00:05:21: %ISDN-6-CONNECT: Interface Serial0:22 is now connected to 4085274206
00:05:26: RADIUS: Retransmit id 6
00:05:31: RADIUS: Tried all servers.
00:05:31: RADIUS: No valid server found. Trying any viable server
00:05:31: RADIUS: Tried all servers.
00:05:31: RADIUS: No response for id 7
00:05:31: RADIUS: Initial Transmit ISDN 0:D:23 id 8 10.0.0.0:1823, Access-Request, len 171
00:05:36: RADIUS: Retransmit id 8
00:05:36: RADIUS: Received from id 8 1.7.157.1:1823, Access-Accept, len 115
00:05:47: %ISDN-6-DISCONNECT: Interface Serial0:22 disconnected from 4085274206, call lasted
 26 seconds
00:05:47: RADIUS: Initial Transmit ISDN 0:D:23 id 9 10.0.0.1:1824, Accounting-Request, len
 775
00:05:47: RADIUS: Received from id 9 1.7.157.1:1824, Accounting-response, len 20
```

The following example shows how to enable debugging of RADIUS accounting collection:

```
Router# debug radius accounting
Radius protocol debugging is on
Radius protocol brief debugging is off
Radius protocol verbose debugging is off
Radius packet hex dump debugging is off
Radius packet protocol (authentication) debugging is off
Radius packet protocol (accounting) debugging is on
Radius packet retransmission debugging is off
Radius server fail-over debugging is off
Radius elog debugging is off
```

Related Commands

Command	Description
debug aaa accounting	Displays information on accountable events as they occur.
debug aaa authentication	Displays information on AAA/TACACS+ authentication.

debug snmp packet

To display information about every Simple Network Management Protocol (SNMP) packet sent or received by the router, use the **debug snmp packet** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug snmp packet

no debug snmp packet

Syntax Description

This command has no arguments or keywords.

Command Default

The command is disabled by default.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.3(2)T	This command was integrated into Cisco IOS Release 12.3(2)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
Cisco IOS XE Release 2.5	This command was implemented on Cisco ASR 1000 series routers.

Examples

The following is sample output from the **debug snmp packet**command. In this example, the router receives a get-next request from the host at 192.10.2.10 and responds with the requested information.

```
Router# debug snmp packet
SNMP: Packet received via UDP from 192.10.2.10 on Ethernet0
SNMP: Get-next request, reqid 23584, errstat 0, erridx 0
 sysUpTime = NULL TYPE/VALUE
 system.1 = NULL TYPE/VALUE
 system.6 = NULL TYPE/VALUE
SNMP: Response, reqid 23584, errstat 0, erridx 0
 sysUpTime.0 = 2217027
 system.1.0 = Cisco Internetwork Operating System Software
 system.6.0 =
SNMP: Packet sent via UDP to 192.10.2.10
```

Based on the kind of packet sent or received, the output may vary. For get-bulk requests, a line similar to the following is displayed:

```
SNMP: Get-bulk request, reqid 23584, nonrptr 10, maxreps 20
```

For traps, a line similar to the following is displayed:

```
SNMP: V1 Trap, ent 1.3.6.1.4.1.9.1.13, gentrap 3, spectrap 0
```

The table below describes the significant fields shown in the display.

Table 6: debug snmp packet Field Descriptions

Field	Description
Get-next request	Indicates what type of SNMP protocol data unit (PDU) the packet is. Possible types are as follows: • Get request • Get-next request • Response • Set request • V1 Trap • Get-bulk request • Inform request • V2 Trap Depending on the type of PDU, the rest of this line displays different fields. The indented lines following this line list the MIB object names and corresponding values.
reqid	Request identification number. This number is used by the SNMP manager to match responses with requests.
errstat	Error status. All PDU types other than response will have an errstat of 0. If the agent encounters an error while processing the request, it will set errstat in the response PDU to indicate the type of error.
erridx	Error index. This value will always be 0 in all PDUs other than responses. If the agent encounters an error, the erridx will be set to indicate which varbind in the request caused the error. For example, if the agent had an error on the second varbind in the request PDU, the response PDU will have an erridx equal to 2.
nonrptr	Nonrepeater value. This value and the maximum repetition value are used to determine how many varbinds are returned. Refer to RFC 1905 for details.

Field	Description
maxreps	Maximum repetition value. This value and the nonrepeater value are used to determine how many varbinds are returned. Refer to RFC 1905 for details.
ent	Enterprise object identifier. Refer to RFC 1215 for details.
gentrap	Generic trap value. Refer to RFC 1215 for details.
spectrap	Specific trap value. Refer to RFC 1215 for details.

debug track

To display tracking activity for tracked objects, use the **debug track** command in privileged EXEC mode. To turn off output, use the **no** form of this command.

debug track

no debug track

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.3(8)T	The output was enhanced to include the track-list objects.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS XE Release 2.1.

Usage Guidelines

Use this command to display activity for objects being tracked by the tracking process. These objects can be the state of IP routing, the line-protocol state of an interface, the IP-route reachability, and the IP-route threshold metric.

Examples

The following example shows that object number 100 is being tracked and that the state of IP routing on Ethernet interface 0/2 is down:

```
Router# debug track
Feb 26 19:56:23.247:Track:100 Adding interface object
Feb 26 19:56:23.247:Track:Initialise
Feb 26 19:56:23.247:Track:100 New interface Et0/2, ip routing Down
Feb 26 19:56:23.247:Track:Starting process
```

The following example shows that object number 100 is being tracked and that the state of IP routing on Ethernet interface 0/2 has changed and is back up:

```
Router# debug track
Feb 26 19:56:41.247:Track:100 Change #2 interface Et0/2, ip routing Down->Up
00:15:07:%LINK-3-UPDOWN:Interface Ethernet0/2, changed state to up
00:15:08:%LINEPROTO-5-UPDOWN:Line protocol on Interface Ethernet0/2, changed state to up
```

Related Commands

Command	Description
show track	Displays tracking information.

debug vrrp ha

To display debugging messages for Virtual Router Redundancy Protocol (VRRP) high availability, use the **debug vrrp ha**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug vrrp ha

no debug vrrp ha

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRC	This command was introduced.
12.2(33)SB2	This command was integrated into Cisco IOS Release 12.2(33)SB2.
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS XE Release 2.1.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Examples

The following examples for the **debug vrrp ha**command display the syncing of VRRP state information from the Active RP to the Standby RP.

The following sample output displays two VRRP state changes on the Active RP:

```
Router# debug vrrp ha
.
.
.
*Nov 14 11:36:50.272 UTC: VRRP: Gi3/2 Grp 42 RF Encode state Backup into sync buffer
*Nov 14 11:36:50.272 UTC: %VRRP-6-STATECHANGE: Gi3/2 Grp 42 state Init -> Backup
*Nov 14 11:36:53.884 UTC: VRRP: Gi3/2 Grp 42 RF Encode state Master into sync buffer
*Nov 14 11:36:53.884 UTC: %VRRP-6-STATECHANGE: Gi3/2 Grp 42 state Backup -> Master
```

The following sample output displays two VRRP state changes on the Standby RP:

```
Router# debug vrrp ha
.
.
.
*Nov 14 11:36:50.392 UTC: STDBY: VRRP: Gi3/2 Grp 42 RF sync state Init -> Backup
*Nov 14 11:36:53.984 UTC: STDBY: VRRP: Gi3/2 Grp 42 RF sync state Backup -> Master
```

Related Commands

Command	Description
debug vrrp error	Displays debugging messages about VRRP error conditions.
debug vrrp events	Displays debugging messages about VRRP events.
debug vrrp state	Displays debugging messages about the VRRP state transitions.