



debug parser cache through debug xml Commands

debug parser cache

To display CLI parser debugging information, use the **debug parser cache** command in privileged EXEC mode. To disable the display of CLI parser debugging information, use the **no** form of this command.

debug parser cache [*level*]

no debug parser cache

Syntax Description	<i>level</i>	(Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.
--------------------	--------------	---

Defaults	The default value for the debugging level is 1.
----------	---

Command Modes	The following table shows the modes in which you can enter the command:
---------------	---

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History	Release	Modification
	7.0(1)	This command was introduced.

Usage Guidelines	Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to 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 enables CLI parser debugging messages. The show debug command indicates the current debugging configuration. The CLI parser debugging messages appear before and after the output of the show debug command. <pre> hostname# debug parser cache debug parser cache enabled at level 1 hostname# show debug parser cache: try to match 'show debug' in exec mode debug parser cache enabled at level 1 parser cache: hit at index 8 hostname# </pre>
----------	--

Related Commands

Command	Description
show debug	Displays the current debugging configuration.

debug phone-proxy

To show debugging messages for the Phone Proxy instance, use the **debug phone-proxy** command in privileged EXEC mode. To stop displaying Phone Proxy messages, use the **no** form of this command.

debug phone-proxy [**media** | **signaling** | **tftp** [**errors** | **events**]]

no debug phone-proxy [**media** | **signaling** | **tftp** [**errors** | **events**]]

Syntax Description

errors	(Optional) Show debugging messages of phone-proxy errors.
events	(Optional) Show debugging messages of phone-proxy events.
media	(Optional) Show debugging messages of media sessions for SIP and Skinny inspections.
signaling	(Optional) Show debugging messages of signaling sessions for SIP and Skinny inspections.
tftp	(Optional) Show debugging messages of TFTP inspection, including creation of the CTL file and configuration file parsing.

Defaults

If no options are specified with the **debug phone-proxy** command, all phone-proxy debugging messages are displayed.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	—	•	—	—

Command History

Release	Modification
8.0(4)	The command was introduced.

Usage Guidelines

The **debug phone-proxy** command displays detailed information about Phone Proxy activity. The **no debug phone-proxy** commands turn off all enabled debugging.

Examples

The following example shows successful TFTP transactions for the configuration file request for the Phone Proxy:

```
hostname(config)# debug phone-proxy tftp
PP: 98.208.49.30/1028 requesting SEP00070E364804.cnf.xml.sgn
PP: opened 0x33952aa2
PP: Received data from 192.168.200.101 to outside:98.208.49.30/1028
    Received Block 1
```

```
PP: Acked Block #1 from 98.208.49.30/1028 to 192.168.200.101/39514
    .... [snip].....
PP: Received data from 192.168.200.101 to outside:98.208.49.30/1028
    Received Block 10
PP: Acked Block #10 from 98.208.49.30/1028 to 192.168.200.101/39514
PP: Installed application redirect rule from 98.208.49.30 to 192.168.200.101 using
redirect port 2000 and secure port 2443
PP: Modifying to TLS as the transport layer protocol.
PP: Modifying to encrypted mode.
PP: Data Block 1 forwarded from 192.168.200.101/39514 to 98.208.49.30/1028
PP: Received ACK Block 1 from outside:98.208.49.30/1028 to inside:192.168.200.101
    ..... [snip] ....
PP: Data Block 11 forwarded to 98.208.49.30/1028
PP: Received ACK Block 11 from outside:98.208.49.30/1028 to inside:192.168.200.101
PP: TFTP session complete, all data sent
```

Related Commands

Command	Description
phone-proxy	Configures the Phone Proxy instance.
show running-config	Displays Phone Proxy-specific information.
phone-proxy	

debug pim

To display PIM debugging information, use the **debug pim** command in privileged EXEC mode. To disable the display of debugging information, use the **no** form of this command.

```
debug pim [df-election [interface if_name | rp rp] | group group | interface if_name | neighbor]

no debug pim [df-election [interface if_name | rp rp] | group group | interface if_name | neighbor]
```

Syntax Description

df-election	(Optional) Displays debugging messages for PIM bidirectional DF-election message processing.
group <i>group</i>	(Optional) Displays debugging information for the specified group. The value for <i>group</i> can be one of the following: - Name of the multicast group, as defined in the DNS hosts table or with the domain ipv4 host command. - IP address of the multicast group, which is a multicast IP address in four-part, dotted-decimal notation.
interface <i>if_name</i>	(Optional) When used with the df-election keyword, limits the DF election debugging display to information for the specified interface. When used without the df-election keyword, displays PIM error messages for the specified interface. Note The debug pim interface command does not display PIM protocol activity messages; it only displays error messages. To see debugging information for PIM protocol activity, use the debug pim command without the interface keyword. You can use the group keyword to limit the display to the specified multicast group.
neighbor	(Optional) Displays only the sent and received PIM hello messages.
rp <i>rp</i>	(Optional) Can be either one of the following: - Name of the RP, as defined in the DNS hosts table or with the domain ipv4 host command. - IP address of the RP, which is a multicast IP address in four-part, dotted-decimal notation.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	—	•	—	—

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

This command logs PIM packets received and transmitted and PIM-related events.

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to 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 is sample output from the **debug pim** command:

```
hostname# debug pim
PIM: Received Join/Prune on Ethernet1 from 172.24.37.33
PIM: Received Join/Prune on Ethernet1 from 172.24.37.33
PIM: Received Join/Prune on Tunnel0 from 10.3.84.1
PIM: Received Join/Prune on Ethernet1 from 172.24.37.33
PIM: Received Join/Prune on Ethernet1 from 172.24.37.33
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 172.24.37.33
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 172.24.37.6
PIM: Received Join/Prune on Ethernet1 from 172.24.37.33
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 (172.24.84.16/28, 224.2.0.1) RP-bit set RP 172.24.84.16
PIM: Send Prune on Ethernet1 to 172.24.37.6 for (172.24.84.16/28, 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: 172.24.84.16/28
PIM: Send periodic Join/Prune to RP via 172.24.37.6 (Ethernet1)
```

Related Commands

Command	Description
show pim group-map	Displays the group-to-protocol mapping table.
show pim interface	Displays interface-specific information for PIM.
show pim neighbor	Displays entries in the PIM neighbor table.

debug pix acl

To show PIX ACL debugging messages, use the **debug pix acl** command in privileged EXEC mode. To stop showing debugging messages, use the **no** form of this command.

debug pix acl

no debug pix acl

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 enables debugging messages for PIX ACLs:

```
hostname# debug pix acl
```

Related Commands

Command	Description
debug pix process	Shows debugging messages for xlate and secondary connections processing.
show debug	Shows all enabled debuggers.

debug pix cls

To show PIX CLS debugging messages, use the **debug pix cls** command in privileged EXEC mode. To stop showing debugging messages, use the **no** form of this command.

debug pix cls

no debug pix cls

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 enables debugging messages for PIX CLS:

```
hostname# debug pix cls
```

Related Commands

Command	Description
debug pix process	Shows debug messages for xlate and secondary connections processing.
show debug	Shows all enabled debuggers.

debug pix pkt2pc

To show debugging messages that trace packets sent to the uauth code and that trace the event where the uauth proxy session is cut through to the data path, use the **debug pix pkt2pc** command in privileged EXEC mode. To stop showing debugging messages, use the **no** form of this command.

debug pix pkt2pc

no debug pix pkt2pc

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines Using **debug** commands might slow down traffic on busy networks.

Examples The following example enables debugging messages that trace packets sent to the uauth code and that trace the event where the uauth proxy session is cut through to the data path:

```
hostname# debug pix pkt2pc
```

Command	Description
debug pix process	Shows debugging messages for xlate and secondary connections processing.
show debug	Shows all enabled debuggers.

debug pix process

To show debugging messages for xlate and secondary connections processing, use the **debug pix process** command in privileged EXEC mode. To stop showing debugging messages, use the **no** form of this command.

debug pix process

no debug pix process

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

Using **debug** commands might slow down traffic on busy networks.

Examples

The following example enables debugging messages for xlate and secondary connections processing:

```
hostname# debug pix process
```

Related Commands

Command	Description
debug pix pkt2pc	Shows debugging messages that trace packets sent to the uauth code and that trace the event where the uauth proxy session is cut through to the data path.
show debug	Shows all enabled debuggers.

debug pix uauth

To show PIX uauth debugging messages, use the **debug pix uauth** command in privileged EXEC mode. To stop showing debugging messages, use the **no** form of this command.

```
debug pix uauth
no debug pix uauth
```

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 enables debugging messages for PIX uauth:

```
hostname# debug pix uauth
```

Command	Description
debug pix process	Shows debugging messages for xlate and secondary connections processing.
show debug	Shows all enabled debuggers.

debug pptp

To show debugging messages for PPTP application inspection, use the **debug pptp** command in privileged EXEC mode. To stop showing debugging messages for PPTP, use the **no** form of this command.

debug pptp [*level*]

no debug pptp [*level*]

Syntax Description

level (Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.

Defaults

The default value for the debugging level is 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	—

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

To see the current debugging command settings, enter the **show debug** command. To stop the debugging output, enter the **no debug** command. To stop all debugging messages from being displayed, enter the **no debug all** command.



Note

Enabling the **debug pptp** command may slow down traffic on busy networks.

Examples

The following example enables debugging messages at the default level (1) for PPTP application inspection:

```
hostname# debug pptp
```

Related Commands	Command	Description
	class-map	Defines the traffic class to which to apply security actions.
	inspect pptp	Enables PPTP application inspection.
	policy-map	Associates a class map with specific security actions.
	service-policy	Applies a policy map to one or more interfaces.

debug radius

To show RADIUS messages between the ASA and a RADIUS AAA server, use the **debug radius** command in privileged EXEC mode. To stop showing RADIUS messages, use the **no** form of this command.

debug radius [**all** | **decode** | **session** | **user** *username*]

no debug radius

Syntax Description

all	(Optional) Show RADIUS debugging messages for all users and sessions, including decoded RADIUS messages.
decode	(Optional) Show decoded content of RADIUS messages. Content of all RADIUS packets display, including hexadecimal values and the decoded, eye-readable versions of these values.
session	(Optional) Show session-related RADIUS messages. Packet types for sent and received RADIUS messages appear, but not the packet content.
user	(Optional) Show RADIUS debugging messages for a specific user.
<i>username</i>	Specifies the user whose messages you want to see. Valid with the user keyword only.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

The **debug radius** command displays detailed information about RADIUS messaging between the ASA and a RADIUS AAA server. The **no debug all** or **undebug all** command turns off all enabled debugging.

Examples

The following example shows decoded RADIUS messages, which happen to be accounting packets:

```
hostname(config)# debug radius decode
hostname(config)# RADIUS packet decode (accounting request)
```

```
-----
```

```

Raw packet data (length = 216).....
i
Parsed packet data.....
Radius: Code = 4 (0x04)
Radius: Identifier = 105 (0x69)
Radius: Length = 216 (0x00D8)
Radius: Vector: 842E0E99F44C00C05A0A19AB88A81312
Radius: Type = 40 (0x28) Acct-Status-Type
Radius: Length = 6 (0x06)
Radius: Value (Hex) = 0x2
Radius: Type = 5 (0x05) NAS-Port
Radius: Length = 6 (0x06)
Radius: Value (Hex) = 0x1
Radius: Type = 4 (0x04) NAS-IP-Address
Radius: Length = 6 (0x06)
Radius: Value (IP Address) = 10.1.1.1 (0x0A010101)
Radius: Type = 14 (0x0E) Login-IP-Host
Radius: Length = 6 (0x06)
Radius: Value (IP Address) = 10.2.0.50 (0xD0FE1291)
Radius: Type = 16 (0x10) Login-TCP-Port
Radius: Length = 6 (0x06)
Radius: Value (Hex) = 0x50
Radius: Type = 44 (0x2C) Acct-Session-Id
Radius: Length = 12 (0x0C)
Radius: Value (String) =
30 78 31 33 30 31 32 39 66 65 | 0x130129fe
Radius: Type = 1 (0x01) User-Name
Radius: Length = 9 (0x09)
Radius: Value (String) =
62 72 6f 77 73 65 72 | browser
Radius: Type = 46 (0x2E) Acct-Session-Time
Radius: Length = 6 (0x06)
Radius: Value (Hex) = 0x0
Radius: Type = 42 (0x2A) Acct-Input-Octets
Radius: Length = 6 (0x06)
Radius: Value (Hex) = 0x256D
Radius: Type = 43 (0x2B) Acct-Output-Octets
Radius: Length = 6 (0x06)
Radius: Value (Hex) = 0x3E1
Radius: Type = 26 (0x1A) Vendor-Specific
Radius: Length = 30 (0x1E)
Radius: Vendor ID = 9 (0x00000009)
Radius: Type = 1 (0x01) Cisco-AV-pair
Radius: Length = 24 (0x18)
Radius: Value (String) =
69 70 3a 73 6f 75 72 63 65 2d 69 70 3d 31 30 2e | ip:source-ip=10.
31 2e 31 2e 31 30 | 1.1.10
Radius: Type = 26 (0x1A) Vendor-Specific
Radius: Length = 27 (0x1B)
Radius: Vendor ID = 9 (0x00000009)
Radius: Type = 1 (0x01) Cisco-AV-pair
Radius: Length = 21 (0x15)
Radius: Value (String) =
69 70 3a 73 6f 75 72 63 65 2d 70 6f 72 74 3d 33 | ip:source-port=3
34 31 33 | 413
Radius: Type = 26 (0x1A) Vendor-Specific
Radius: Length = 40 (0x28)
Radius: Vendor ID = 9 (0x00000009)
Radius: Type = 1 (0x01) Cisco-AV-pair
Radius: Length = 34 (0x22)
Radius: Value (String) =
69 70 3a 64 65 73 74 69 6e 61 74 69 6f 6e 2d 69 | ip:destination-i
70 3d 32 30 38 2e 32 35 34 2e 31 38 2e 31 34 35 | p=10.2.0.50
Radius: Type = 26 (0x1A) Vendor-Specific

```



```
Radius: Length = 30 (0x1E)
Radius: Vendor ID = 9 (0x00000009)
Radius: Type = 1 (0x01) Cisco-AV-pair
Radius: Length = 24 (0x18)
Radius: Value (String) =
69 70 3a 64 65 73 74 69 6e 61 74 69 6f 6e 2d 70 | ip:destination-p
6f 72 74 3d 38 30 | ort=80
```

Related Commands

Command	Description
show running-config	Displays the configuration that is running on the ASA.

debug redundant-interface

To show debugging messages about redundant interfaces, use the **debug redundant-interface** command in privileged EXEC mode. To stop showing debugging messages for redundant interfaces, use the **no** form of this command.

- debug redundant-interface [level]

no debug redundant-interfac [level]

Syntax Description	level	(Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.
--------------------	-------	---

Defaults	The default level is 1.
----------	-------------------------

Command Modes	The following table shows the modes in which you can enter the command:
---------------	---

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	—	•

Command History	Release	Modification
	8.0(2)	This command was introduced.

Usage Guidelines	Using debug commands might slow down traffic on busy networks.
------------------	---

Examples	The following example enables debugging messages for redundant interfaces: hostname# debug redundant-interface
----------	--

Related Commands	Command	Description
	interface redundant	Creates a redundant interface.
	member-interface	Assigns a physical interface to a redundant interface.
	redundant-interface	Changes the active interface in a redundant interface pair.
	show debug	Shows all enabled debuggers.

debug rip

To display debugging information for RIP, use the **debug rip** command in privileged EXEC mode. To disable the debugging information display, use the **no** form of this command.

debug rip [database | events]

no debug rip [database | events]

Syntax Description

database	Displays RIP database events.
events	Displays RIP processing events.

Defaults

All RIP events are shown in the debugging output.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	—	•	—	—

Command History

Release	Modification
7.0(1)	This command was introduced.
7.2(1)	The database and events keywords were added.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 is sample output from the **debug rip** command:

```
hostname# debug rip

RIP: broadcasting general request on GigabitEthernet0/1
RIP: broadcasting general request on GigabitEthernet0/2
RIP: Received update from 10.89.80.28 on GigabitEthernet0/1
      10.89.95.0 in 1 hops
      10.89.81.0 in 1 hops
      10.89.66.0 in 2 hops
      172.31.0.0 in 16 hops (inaccessible)
      0.0.0.0 in 7 hops
RIP: Sending update to 255.255.255.255 via GigabitEthernet0/1 (10.89.64.31)
```

```

    subnet 10.89.94.0, metric 1
    172.31.0.0 in 16 hops (inaccessible)
RIP: Sending update to 255.255.255.255 via GigabitEthernet0/2 (10.89.94.31)
    subnet 10.89.64.0, metric 1
    subnet 10.89.66.0, metric 3
    172.31.0.0 in 16 hops (inaccessible)
    default 0.0.0.0, metric 8
RIP: bad version 128 from 192.168.80.43

```

Related Commands

Command	Description
router rip	Configures a RIP process.
show running-config rip	Displays the RIP commands in the running configuration.

debug route

To display debugging information for general routing and failover debugging messages for routing, use the **debug route** command in privileged EXEC mode. To disable the debugging information display, use the **no** form of this command.

debug route [**ha** | **events**]

no debug route [**ha** | **events**]

Syntax Description

events	Displays general routing-related debugging messages.
ha	Displays Stateful Failover-related debugging messages for dynamic routing protocols.

Defaults

All general routing events and failover events for routing are shown in the debugging output.

Command Modes

The following table shows the modes in which you can enter the command.

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	—	•	—	—

Command History

Release	Modification
8.4(1)	This command was introduced.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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.

To display general routing debugging information, such as adding or deleting routes in the CP routing table, use the **debug route events** command.

Examples

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

```
hostname# debug route events
add 10.0.3.0 255.255.255.0 via 10.1.1.10, ospf metric [110/40]
```

The following is sample output from the **debug route ha** command on the active unit:

```
ROUTE HA: Route HA start bulk sync
ROUTE HA: Sending Message Version: 1 Action: add Object: route Address: 10.0.0.0 Mask:
255.0.0.0
ROUTE HA: Sending Message Version: 1 Action: add Object: route Address: 10.0.1.0 Mask:
255.0.0.0
ROUTE HA: Sending Message Version: 1 Action: add Object: route Address: 10.0.3.0 Mask:
255.255.255.0
```

The following is sample output from the **debug route ha** command on the standby unit:

```
ROUTE HA: Processing rcvd msg with address: 10.0.3.0 mask: 255.255.255.0 gateway:
10.10.0.10
ROUTE HA: Received Msg ADD address: 10.0.3.0 mask: 255.255.255.0 gateway: 10.0.1.10
metric: 40
ROUTE HA: RIB Epoch number 0 assigned to NDB: 10.0.0.0
ROUTE HA: RIB epoch number 0 assigned to SDB: 10.0.3.0
```

Related Commands

Command	Description
debug rip	Displays RIP debugging information.
show debug route	Displays the general routing debugging configuration.

debug route cluster

To display debugging information for RIB table replication and dynamic updates through trace messages to determine whether or not the RIB table is correctly synchronized to slave units, use the **debug route cluster** command in privileged EXEC mode. To disable the debugging information display, use the **no** form of this command.

debug route cluster

no debug route cluster

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command.

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	—	•	—	—

Command History	Release	Modification
	9.0(1)	This command was introduced. Applies only to the ASA 5580 and 5585-X.

Usage Guidelines Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 is sample output from the **debug route cluster** command:

```
hostname# debug route cluster
ROUTE CLUSTER: old ndb_epoch=1, RIB Epoch number 1 assigned to NDB: 192.168.33.0
ROUTE CLUSTER: old ndb_epoch=1, RIB Epoch number 1 assigned to NDB: 172.16.0.0
ROUTE CLUSTER: old ndb_epoch=1, RIB Epoch number 1 assigned to NDB: 192.168.32.0
ROUTE CLUSTER ip_route_delete: del 172.31.32.1 255.255.255.255 via 172.16.32.4, ospf
metric [110/13]
ROUTE CLUSTER ip_route_delete: delete subnet route to 172.31.32.1 255.255.255.255
ROUTE CLUSTER ip_route_delete: delete network route to 172.31.0.0
ROUTE CLUSTER: old ndb_epoch=1, RIB Epoch number 1 assigned to NDB: 172.16.0.0
ROUTE CLUSTER: old ndb_epoch=1, RIB Epoch number 1 assigned to NDB: 172.17.0.0
ROUTE CLUSTER: old ndb_epoch=1, RIB Epoch number 1 assigned to NDB: 172.18.0.0
```

debug route cluster

```
ROUTE CLUSTER: old ndb_epoch=1, RIB Epoch number 1 assigned to NDB: 172.20.0.0
ROUTE CLUSTER: old ndb_epoch=1, RIB Epoch number 1 assigned to NDB: 172.30.0.0
ROUTE CLUSTER: old ndb_epoch=1, RIB Epoch number 1 assigned to NDB: 172.31.0.0
ROUTE CLUSTER: old ndb_epoch=1, RIB Epoch number 1 assigned to NDB: 192.168.32.0
```

Related Commands

Command	Description
debug route	Displays general routing and failover debugging messages for routing.
show debug route	Displays the general routing debugging configuration.

debug rtp

To display debugging information and error messages for RTP packets associated with H.323 and SIP inspection, use the **debug rtp** command in privileged EXEC mode. To disable the display of debugging information, use the **no** form of this command.

debug rtp [*level*]

no debug rtp [*level*]

Syntax Description

level (Optional) Specifies the level of debugging.

Defaults

The default level is 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	—

Command History

Release	Modification
7.2(1)	This command was introduced.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to 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 how to enable debugging for RTP packets using the **debug rtp** command:

```
hostname# debug rtp 255
debug rtp enabled at level 255
```

Related Commands

Command	Description
policy-map	Creates a Layer 3/4 policy map.

Command	Description
rtp-conformance	Checks RTP packets flowing on the pinholes for protocol conformance in H.323 and SIP.
show running-config policy-map	Displays all current policy map configurations.

debug rtsp

To show debugging messages for RTSP application inspection, use the **debug rtsp** command in privileged EXEC mode. To stop showing debugging messages for RTSP application inspection, use the **no** form of this command.

debug rtsp [*level*]

no debug rtsp [*level*]

Syntax Description

level (Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.

Defaults

The default value for the debugging level is 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	—

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

To see the current debugging command settings, enter the **show debug** command. To stop the debugging output, enter the **no debug** command. To stop all debugging messages from being displayed, enter the **no debug all** command.



Note

Enabling the **debug rtsp** command may slow down traffic on busy networks.

Examples

The following example enables debugging messages at the default level (1) for RTSP application inspection:

```
hostname# debug rtsp
```

Related Commands

Command	Description
class-map	Defines the traffic class to which to apply security actions.
inspect rtsp	Enables RTSP application inspection.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.

debug sdi

To display SDI authentication debugging information, use the **debug sdi** command in privileged EXEC mode. To disable the display of SDI debugging information, use the **no** form of this command.

debug sdi [*level*]

no debug sdi

Syntax Description

level (Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.

Defaults

The default value for the debugging level is 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to 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 enables SDI debugging messages. The **show debug** command indicates that SDI debugging messages are enabled.

```
hostname# debug sdi
debug sdi  enabled at level 1
hostname# show debug
debug sdi  enabled at level 1
hostname#
```

Related Commands

Command	Description
show debug	Displays the current debugging configuration.

debug sequence

To add a sequence number to the beginning of all debugging messages, use the **debug sequence** command in privileged EXEC mode. To disable the use of debugging sequence numbers, use the **no** form of this command.

debug sequence [*level*]

no debug sequence

Syntax Description

level (Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.

Defaults

The defaults are as follows:

- Debugging message sequence numbers are disabled.
- The default value for the debugging level is 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to 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 enables sequence numbers in debugging messages. The **debug parser cache** command enables CLI parser debugging messages. The **show debug** command indicates the current debugging configuration. The CLI parser debugging messages shown include sequence numbers before each message.

```
hostname# debug sequence
debug sequence enabled at level 1
hostname# debug parser cache
```

debug sequence

```
debug parser cache enabled at level 1
hostname# show debug
0: parser cache: try to match 'show debug' in exec mode
debug parser cache enabled at level 1
debug sequence  enabled at level 1
1: parser cache: hit at index 8
hostname#
```

Related Commands

Command	Description
show debug	Displays the current debugging configuration.

debug session-command

To show debugging messages for a session to an SSM, use the **debug session-command** command in privileged EXEC mode. To disable the display of debugging messages for sessions, use the **no** form of this command.

debug session-command [*level*]

no debug session-command [*level*]

Syntax Description

level (Optional) Sets the level to display debugging messages. The range of values is between 1 and 255. To display additional messages at higher levels, set the level to a higher number.

Defaults

The default level is 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	—	•

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

Using **debug** commands might slow down traffic on busy networks.

Examples

The following example enables debugging messages for sessions:

```
hostname# debug session-command
```

Related Commands

Command	Description
session	Sessions to an SSM.

debug sip

To show debugging messages for SIP application inspection, use the **debug sip** command in privileged EXEC mode. To stop showing debugging messages for SIP application inspection, use the **no** form of this command.

- debug sip [ha]**
- no debug sip [ha]**

Syntax Description

ha	(Optional) Displays SIP Stateful Failover messages. When this keyword is used with the debug sip command on the active unit, debugging messages are displayed when SIP state information is sent to the standby unit. When this keyword is used with the debug sip command on the standby unit, debugging messages are displayed with state updates that are received from the active unit.
-----------	--

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	—

Command History

Release	Modification
7.0(1)	This command was introduced.
8.0(2)	The ha keyword was added.

Usage Guidelines

To see the current **debug** command settings, enter the **show debug** command. To stop the debugging output, enter the **no debug** command. To stop all debugging messages from being displayed, enter the **no debug all** command.

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 is sample output from the **debug sip** command for the active unit or failover group in a failover pair:

```
hostname# debug sip ha
SIP HA:      Sending      update SESSION message from faddr 10.132.80.120/5060 laddr
10.130.80.4/50295 Call-id: 001201e8-8a36000d-196df7f1-17cfef14@10.130.80.4 From:
sip:1004@10.132.80.120:001201e88a3600124a7fad61-640406c0 To: sip:1009@10.132.80.120:
State:1

SIP HA:      msg sent to peer successful  Version: 1 Action: update Object: session

SIP HA:      Sending      update TX message from faddr 10.132.80.120/5060laddr
10.130.80.4/50295CSeq 101 INVITEState Transaction Calling
```

The following is sample output from the **debug sip** command for the standby unit or failover group in a failover pair:

```
hostname# debug sip ha
SIP HA:      Message      received from peer, Version: 1 Action: add Object: session

SIP HA:      Created      SIP session for faddr 10.132.80.120/5060 laddr 10.130.80.4/50295
Call-id: 001201e8-8a36000d-196df7f1-17cfef14@10.130.80.4 From:
sip:1004@10.132.80.120:001201e88a3600124a7fad61-640406c0 To: sip:1009@10.132.80.120: 1
total

SIP HA:      Message      received from peer, Version: 1 Action: add Object: tx

SIP HA:      Found an existing session faddr 10.132.80.120/5060 laddr 10.130.80.4/50295
Call-id: 001201e8-8a36000d-196df7f1-17cfef14@10.130.80.4 From:
sip:1004@10.132.80.120:001201e88a3600124a7fad61-640406c0 To: sip:1009@10.132.80.120:

SIP HA:      Created      SIP Transaction      for faddr 10.132.80.120/5060 to      laddr
10.130.80.4/50295CSeq 101 INVITEState Transaction Calling
```

Related Commands

Command	Description
class-map	Defines the traffic class to which to apply security actions.
inspect sip	Enables SIP application inspection.
show conn	Displays the connection state for different connection types.
show sip	Displays information about SIP sessions established through the ASA.
timeout	Sets the maximum idle time duration for different protocols and session types.

debug skinny

To show debugging messages for SCCP (Skinny) application inspection, use the **debug skinny** command in privileged EXEC mode. To stop showing debugging messages for SCCP application inspection, use the **no** form of this command.

- debug skinny** [*level*]
- no debug skinny** [*level*]

Syntax Description	<i>level</i>	(Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.
--------------------	--------------	---

Defaults	The default value for the debugging level is 1.
----------	---

Command Modes	The following table shows the modes in which you can enter the command:
---------------	---

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	—

Command History	Release	Modification
	7.0(1)	This command was introduced.

Usage Guidelines	To see the current debug command settings, enter the show debug command. To stop the debugging output, enter the no debug command. To stop all debugging messages from being displayed, enter the no debug all command.
------------------	---



Note	Enabling the debug skinny command may slow down traffic on busy networks.
------	--

Examples	The following example enables debugging messages at the default level (1) for SCCP application inspection: <pre>hostname# debug skinny</pre>
----------	---

Related Commands

Command	Description
class-map	Defines the traffic class to which to apply security actions.
inspect skinny	Enables SCCP application inspection.
show skinny	Displays information about SCCP sessions established through the ASA.
show conn	Displays the connection state for different connection types.
timeout	Sets the maximum idle time duration for different protocols and session types.

debug sla monitor

To display debugging messages for the SLA monitor operation, use the **debug sla monitor** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

```
debug sla monitor [error | trace] [sla-id]

no debug sla monitor [sla-id]
```

Syntax Description

error	(Optional) Shows the output of IP SLA monitor error messages.
<i>sla-id</i>	(Optional) Shows the ID of the SLA to debug.
trace	(Optional) Shows the output of IP SLA monitor trace messages.

Defaults

Both error and trace messages are shown by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	—	•	—	—

Command History

Release	Modification
7.2(1)	This command was introduced.

Usage Guidelines

Only 32 SLA operations can be debugged at one time.

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 enables SLA operation error debugging:

```
hostname(config)# debug sla monitor error
```

The following example shows how to display SLA operation trace messages for the specified SLA operation:

```
hostname(config)# debug sla monitor trace 123
```

Related Commands	Command	Description
	clear configure route	Removes statically configured route commands.
	clear route	Removes routes learned through dynamic routing protocols such as RIP.
	show route	Displays route information.
	show running-config route	Displays configured routes.

debug sqlnet

To show debugging messages for SQL*Net application inspection, use the **debug sqlnet** command in privileged EXEC mode. To stop showing debugging messages for SQL*Net application inspection, use the **no** form of this command.

```
debug sqlnet [level]
no debug sqlnet [level]
```

Syntax Description	level	(Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.
--------------------	-------	---

Defaults	The default value for the debugging level is 1.
----------	---

Command Modes	The following table shows the modes in which you can enter the command:
---------------	---

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	—

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines	To see the current debug command settings, enter the show debug command. To stop the debugging output, enter the no debug command. To stop all debugging messages from being displayed, enter the no debug all command.
------------------	---



Note	Enabling the debug sqlnet command may slow down traffic on busy networks.
------	--

Examples	The following example enables debugging messages at the default level (1) for SQL*Net application inspection: hostname# debug sqlnet
----------	--

Related Commands

Command	Description
class-map	Defines the traffic class to which to apply security actions.
inspect sqlnet	Enables SQL*Net application inspection.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.
show conn	Displays the connection state for different connection types, including SQL*Net.

debug ssh

To display debugging information and error messages associated with SSH, use the **debug ssh** command in privileged EXEC mode. To disable the display of debugging information, use the **no** form of this command.

```
debug ssh [level]

no debug ssh [level]
```

Syntax Description	level (Optional) Specifies the level of debugging.
--------------------	--

Defaults	The default level is 1.
----------	-------------------------

Command Modes	The following table shows the modes in which you can enter the command:
---------------	---

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	—

Command History	Release	Modification
	7.0(1)	This command was introduced.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to 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 is sample output from the **debug ssh 255** command:

```
hostname# debug ssh 255
debug ssh enabled at level 255
SSH2 0: send: len 64 (includes padlen 17)
SSH2 0: done calc MAC out #239
SSH2 0: send: len 32 (includes padlen 7)
SSH2 0: done calc MAC out #240
SSH2 0: send: len 64 (includes padlen 15)
SSH2 0: done calc MAC out #241
SSH2 0: send: len 32 (includes padlen 16)
SSH2 0: done calc MAC out #242
SSH2 0: send: len 64 (includes padlen 7)
SSH2 0: done calc MAC out #243
```

```

SSH2 0: send: len 64 (includes padlen 18)
SSH2 0: done calc MAC out #244
SSH2 0: send: len 64 (includes padlen 8)
SSH2 0: done calc MAC out #245
SSH2 0: send: len 64 (includes padlen 18)
SSH2 0: done calc MAC out #246
SSH2 0: send: len 64 (includes padlen 7)
SSH2 0: done calc MAC out #247
SSH2 0: send: len 64 (includes padlen 18)
SSH2 0: done calc MAC out #248
SSH2 0: send: len 64 (includes padlen 7)
SSH2 0: done calc MAC out #249
SSH2 0: send: len 64 (includes padlen 18)
SSH2 0: done calc MAC out #250
SSH2 0: send: len 64 (includes padlen 8)
SSH2 0: done calc MAC out #251
SSH2 0: send: len 64 (includes padlen 18)
SSH2 0: done calc MAC out #252
SSH2 0: send: len 64 (includes padlen 7)
SSH2 0: done calc MAC out #253
SSH2 0: send: len 64 (includes padlen 18)
SSH2 0: done calc MAC out #254
SSH2 0: send: len 64 (includes padlen 8)
SSH2 0: done calc MAC out #255
SSH2 0: send: len 64 (includes padlen 18)
SSH2 0: done calc MAC out #256
SSH2 0: send: len 64 (includes padlen 7)
SSH2 0: done calc MAC out #257
SSH2 0: send: len 64 (includes padlen 18)
SSH2 0: done calc MAC out #258

```

Related Commands

Command	Description
clear configure ssh	Clears all SSH commands from the running configuration.
show running-config ssh	Displays the current SSH commands in the running configuration.
show ssh sessions	Displays information about active SSH sessions to the ASA.
ssh	Allows SSH connectivity to the ASA from the specified client or network.

debug sunrpc

To show debugging messages for RPC application inspection, use the **debug sunrpc** command in privileged EXEC mode. To stop showing debugging messages for RPC application inspection, use the **no** form of this command.

- debug sunrpc

[level]
- no debug sunrpc

[level]

Syntax Description	level	(Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.
--------------------	-------	---

Defaults	The default value for the debugging level is 1.
----------	---

Command Modes	The following table shows the modes in which you can enter the command:
---------------	---

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	—

Command History	Release	Modification
	7.0(1)	This command was introduced.

Usage Guidelines	To see the current debugging command settings, enter the show debug command. To stop the debugging output, enter the no debug command. To stop all debugging messages from being displayed, enter the no debug all command.
------------------	--

Note	
	Enabling the debug sunrpc command may slow down traffic on busy networks.

Examples	The following example enables debugging messages at the default level (1) for RPC application inspection: <pre>hostname# debug sunrpc</pre>
----------	--

Related Commands

Command	Description
class-map	Defines the traffic class to which to apply security actions.
inspect sunrpc	Enables Sun RPC application inspection.
policy-map	Associates a class map with specific security actions.
show conn	Displays the connection state for different connection types, including RPC.
timeout	Sets the maximum idle time duration for different protocols and session types.

debug switch ilpm

To show debugging messages for models with a built-in switch, such as the ASA 5505, show debugging messages for PoE, use the **debug switch ilpm** command in privileged EXEC mode. To stop showing debugging messages for PoE, use the **no** form of this command.

```
debug switch ilpm [events | errors] [level]

no debug switch ilpm [events | errors] [level]
```

Syntax Description

errors	(Optional) Shows troubleshooting information when there is an error.
events	(Optional) Shows PoE events.
<i>level</i>	(Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.

Defaults

By default, both events and errors are shown if you do not specify a keyword. The default level is 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	—	—

Command History

Release	Modification
7.2(1)	This command was introduced.

Usage Guidelines

Using **debug** commands might slow down traffic on busy networks.

Examples

The following example enables debugging messages for PoE ports:

```
hostname# debug switch ilpm
```

Related Commands

Command	Description
interface vlan	Adds a VLAN interface.
debug switch manager	Shows debugging messages for VLAN assignment and switchport command-caused events and errors.
show debug	Shows all enabled debuggers.

debug switch manager

To show debugging messages for switch port models with a built-in switch, such as the ASA 5505, show debugging messages for VLAN assignment, and **switchport** command-caused events and errors, use the **debug switch manager** command in privileged EXEC mode. To stop showing debugging messages for switch ports, use the **no** form of this command.

debug switch manager [events | errors] [level]

no debug switch manager [events | errors] [level]

Syntax Description

errors	(Optional) Shows troubleshooting information when there is an error.
events	(Optional) Shows the switch manager events.
<i>level</i>	(Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.

Defaults

By default, both events and errors are shown if you do not specify a keyword. The default level is 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	—	—

Command History

Release	Modification
7.2(1)	This command was introduced.

Usage Guidelines

Using **debug** commands might slow down traffic on busy networks.

Examples

The following example enables debugging messages for switch ports:

```
hostname# debug switch manager
```

Related Commands

Command	Description
interface vlan	Adds a VLAN interface.
debug switch ilpm	Shows debugging messages for PoE.
show debug	Shows all enabled debuggers.

debug tacacs

To display TACACS+ debugging information, use the **debug tacacs** command in privileged EXEC mode. To disable the display of TACACS+ debugging information, use the **no** form of this command.

debug tacacs [session | user *username*]

no debug tacacs [session | user *username*]

Syntax Description

session	Displays session-related TACACS+ debugging messages.
user	Displays user-specific TACACS+ debugging messages. You can display TACACS+ debugging messages for only one user at a time.
<i>username</i>	Specifies the user whose TACACS+ debugging messages you want to view.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to 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 enables TACACS+ debugging messages and provides sample output from the **show debug** command:

```
hostname# debug tacacs user admin342
hostname# show debug
debug tacacs user admin342
hostname#
```


Related Commands

Command	Description
show debug	Displays the current debugging configuration.

debug tcp-map

To show debugging messages for TCP application inspection maps, use the **debug tcp-map** command in privileged EXEC mode. To stop showing debugging messages for TCP application inspection, use the **no** form of this command.

```
debug tcp-map
no debug tcp-map
```

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 enables debugging messages for TCP application inspection maps. The **show debug** command indicates that debugging messages for TCP application inspection maps are enabled.

```
hostname# debug tcp-map
debug tcp-map enabled at level 1.
hostname# show debug
debug tcp-map enabled at level 1.
hostname#
```

Related Commands	Command	Description
	show debug	Displays the current debugging configuration.

debug timestamps

To add timestamp information to the beginning of all debugging messages, use the **debug timestamps** command in privileged EXEC mode. To disable the use of debugging timestamps, use the **no** form of this command.

debug timestamps [*level*]

no debug timestamps

Syntax Description	<i>level</i>	(Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.
--------------------	--------------	---

Defaults	The defaults are as follows: - Debugging timestamp information is disabled. - The default value for the debugging level is 1.				
----------	--	--	--	--	--

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History	Release	Modification
	7.0(1)	This command was introduced.

Usage Guidelines	Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 enables timestamps in debugging messages. The debug parser cache command enables CLI parser debugging messages. The show debug command indicates the current debugging configuration. The CLI parser debugging messages shown include timestamps before each message.
----------	---

```

hostname# debug timestamps
debug timestamps enabled at level 1
hostname# debug parser cache
debug parser cache enabled at level 1
    
```

```
hostname# show debug
1982769.770000000: parser cache: try to match 'show debug' in exec mode
1982769.770000000: parser cache: hit at index 8
hostname#
```

Related Commands

Command	Description
show debug	Displays the current debugging configuration.

debug user-identity

To debug the Identity Firewall, use the **debug user-identity** command in privileged EXEC mode. To disable the use of debug command, use the **no** form of this command.

debug user-identity { **acl** | **ad-agent** | **all** | **debug** | **error** | **fqdn** | **ha** | **ldap** | **logout-probe** | **process** | **tmatch** | **user** *user_name* | **user-group** *user_group_name* }

no debug user-identity { **acl** | **ad-agent** | **all** | **debug** | **error** | **fqdn** | **ha** | **ldap** | **logout-probe** | **process** | **tmatch** | **user** *user_name* | **user-group** *user_group_name* }

Syntax Description

acl	Enables debugging messages related to access list changes.
ad-agent	Enables debugging messages for the connection between the ASA and the Windows server on which the AD Agent is installed.
all	Enables all debugging messages for all aspects of the Identity Firewall.
debug	Enables debugging information about the debugging-level messages for the Identity Firewall.
error	Enables debugging information about errors in the user identity module or the Identity Firewall.
fqdn	Enables debugging messages about FQDN to IP address updates.
ha	Enables debugging messages for your high availability deployment of the Identity Firewall. See the CLI configuration guide for information about the types of Identity Firewall deployments.
ldap	Enables debugging messages for the LDAP query that the ASA sends to Microsoft Active Directory for the user groups configured on the AD Server and for the reply that the ASA receives from Active Directory.
logout-probe	Enables debugging messages for logout probing.
process	Enables debugging messages for the user identity process of the Identity Firewall.
tmatch	Enables debugging messages related to tmatch changes.
user <i>user_name</i>	Enables debugging messages for the specified user in all domains.
user-group <i>user_group_name</i>	Enables debugging messages for the specified user group in all domains.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
8.4(2)	This command was introduced.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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.

debug user-identity logout-probe

Using the **debug user-identity logout-probe** command enables debugging messages for logout probing, including NetBIOS handling and inactive user check. The client logs onto the network through Microsoft Active Directory. The AD Server authenticates users and generates user login security logs. The ASA probes the NetBIOS of the client to pass inactive and no-response users. Enabling NetBIOS probing configures how often the ASA probes the user client IP address to determine whether the client is still active.

To minimize the NetBIOS packets, the ASA only sends a NetBIOS probe to a client when the user has been idle for more than the specified number of minutes. By default, NetBIOS probing is disabled.

debug user-identity tmatch

When the ASA resolves an FQDN network object, it populates the resolved IP addresses and other fields, such as the IP address of a user or user-group of an access-list in the tmatch lookup table.

Depending on the Identity Firewall configuration, the ASA updates IP addresses from the DNS server periodically or when the TTL of DNS entries expire, whichever comes first. When the ASA finds new IP addresses, it adds them to the tmatch lookup table. When existing IP addresses expire, the ASA removes them from the tmatch lookup table.

You can configure a longer DNS expire-entry timer to balance degradation of system performance because of tmatch recompilation; you must balance system performance with the security risk caused by the gap between DNS entries TTL expiration and the actual time when expired entries are removed from tmatch table.

Even when you configure the ASA with reasonable expire-DNS-entry value, the ASA can still recompile the tmatch lookup table periodically when DNS load balancing is configured; however, this does not guarantee that all valid IP addresses will be refreshed within any defined time period.

Some FQDN hosts can have very short TTL, which leads to frequent recompilation of the tmatch lookup table and a performance impact.

When the Identity Firewall process is disabled (no identity-firewall enable), the tmatch lookup table is not updated for new or changed IP-user mappings, and user-identity rules have any effect at all on security policies.

Examples

The following example shows how to turn off debugging for all aspects of the Identity Firewall, and then turn on debugging of the Identity Firewall process:

```
hostname# debug user-identity ?
acluser-identity ACL message
ad-agent          user-identity ad-agent message
all               All user-identity messages
debug            user-identity debug message
error            user-identity error message
```

debug user-identity

```

fqdn          user-identity fqdn message
ha            user-identity HA message
ldap          user-identity ldap message
logout-probe  user-identity logout-probe message
process       user-identity process message
tmatch        user-identity tmatch message
user          user-identity user message
user-group    user-identity user-group message
hostname# no debug user-identity all
debug user-identity process enabled
hostname# debug user-identity process

```

Related Commands

Command	Description
show debug	Displays the current debugging configuration.

debug vpn-sessiondb

To display VPN-session database debugging information, use the **debug vpn-sessiondb** command in privileged EXEC mode. To disable the display of VPN-session database debugging information, use the **no** form of this command.

debug vpn-sessiondb [*level*]

no debug vpn-sessiondb

Syntax Description

level (Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.

Defaults

The default value for the debugging level is 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
7.0(1)	This command was introduced.
9.0(1)	Support for multiple context mode was added.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 enables VPN-session database debugging messages. The **show debug** command indicates that VPN-session database debugging messages are enabled.

```
hostname# debug vpn-sessiondb
debug vpn-sessiondb enabled at level 1
hostname# show debug
debug vpn-sessiondb enabled at level 1
hostname#
```

 debug vpn-sessiondb**Related Commands**

Command	Description
show debug	Displays the current debugging configuration.

debug wccp

To enable logging of WCCP events, use the **debug wccp** command in privileged EXEC mode. To disable the logging of WCCP debugging messages, use the **no** form of this command.

debug wccp {events | packets | subblocks}

no debug wccp {events | packets | subblocks}

Syntax Description

events	Enables logging of WCCP session events.
packets	Enables logging of debugging messages about WCCP packet information.
subblocks	Enables logging of debugging messages about WCCP subblocks.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	—	•

Command History

Release	Modification
7.2(1)	This command was introduced.

Usage Guidelines

The high priority assigned to debugging output can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 enables the logging of all WCCP session events:

```
hostname# debug wccp events
hostname#
```

The following example enables the logging of WCCP packet debugging messages:

```
hostname# debug wccp packets
hostname#
```

The following example disables the logging of WCCP debugging messages:

```
hostname# no debug wccp
```

debug wccp

hostname#

Related Commands

Command	Description
wccp	Enables support of WCCP.
show debug	Displays the current debugging configuration.

debug webvpn

To log WebVPN debugging messages, use the **debug webvpn** command in privileged EXEC mode. To disable the logging of WebVPN debugging messages, use the **no** form of this command.

debug webvpn [**chunk** | **cifs** | **citrix** | **failover** | **html** | **javascript** | **request** | **response** | **svc** | **transformation** | **url** | **util** | **xml**] [*level*]

no debug webvpn [**chunk** | **cifs** | **citrix** | **failover** | **html** | **javascript** | **request** | **response** | **svc** | **transformation** | **url** | **util** | **xml**] [*level*]

Syntax Description		
chunk		Displays debugging messages about memory blocks used to support WebVPN connections.
cifs		Displays debugging messages about connections between CIFS servers and WebVPN users.
citrix		Displays debug messages about connections between Citrix Metaframe Servers and Citrix ICA clients over WebVPN.
failover		Displays debugging messages about equipment failovers affecting WebVPN connections.
html		Displays debugging messages about HTML pages sent over WebVPN connections.
javascript		Displays debugging messages about JavaScript sent over WebVPN connections.
<i>level</i>		(Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.
request		Displays debugging messages about requests issued over WebVPN connections.
response		Displays debugging messages about responses issued over WebVPN connections.
svc		Displays debugging messages about connections to SSL VPN clients over WebVPN.
transformation		Displays debugging messages about WebVPN content transformation.
url		Displays debugging messages about website requests issued over WebVPN connections.
util		Displays debugging messages about CPU utilization dedicated to support connections to WebVPN remote users.
xml		Displays debugging messages about JavaScript sent over WebVPN connections.

Defaults

The default value for the debugging level is 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
7.0(1)	This command was introduced.
9.0(1)	Support for multiple context mode was added.

Usage Guidelines

The high priority assigned to debugging output can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to 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 enables WebVPN debugging messages for CIFS. The **show debug** command indicates that CIFS debugging messages are enabled.

```
hostname# debug webvpn cifs
INFO: debug webvpn cifs enabled at level 1.
hostname# show debug
debug webvpn cifs enabled at level 1
hostname#
```

Related Commands

Command	Description
show debug	Displays the current debugging configuration.

debug xdmcp

To show debugging messages for XDMCP application inspection, use the **debug xdmcp** command in privileged EXEC mode. To stop showing debugging messages for XDMCP application inspection, use the **no** form of this command.

debug xdmcp [*level*]

no debug xdmcp [*level*]

Syntax Description

level (Optional) Sets the debugging message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.

Defaults

The default value for the debugging level is 1.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	—

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

To see the current debugging command settings, enter the **show debug** command. To stop the debugging output, enter the **no debug** command. To stop all debugging messages from being displayed, enter the **no debug all** command.



Note

Enabling the **debug xdmcp** command may slow down traffic on busy networks.

Examples

The following example enables debugging messages at the default level (1) for XDMCP application inspection:

```
hostname# debug xdmcp
```

Related Commands

Command	Description
class-map	Defines the traffic class to which to apply security actions.
inspect xdmcp	Enables XDMCP application inspection.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.

debug xml

To display debugging information for the XML parser, use the **debug xml** command in privileged EXEC mode. To disable the display of debugging information, use the **no** form of this command.

debug xml [element | event]

no debug xml [element | event]

Syntax Description

element	(Optional) Displays debugging events related to processing individual XML elements.
event	(Optional) Displays XML parsing or error events.

Defaults

If no keywords are specified, all XML parser debugging messages are shown.

Command Modes

The following table shows the modes in which you can enter the command:

	Firewall Mode		Security Context		
				Multiple	
Command Mode	Routed	Transparent	Single	Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
8.0(2)	This command was introduced.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to 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 is sample output from the **debug xml element** command:

```
hostname# debug xml element
debug xml element enabled at level 1

XML Executes cmd: hostname hostname
XML Executes cmd: domain-name example.com
XML Executes cmd: names
XML Executes cmd: dns-guard
XML Executes cmd: !
XML Executes cmd: interface Ethernet0
XML Executes cmd: nameif outside
XML Executes cmd: security-level 0
```

```

XML Executes cmd: ip address 192.168.5.151 255.255.255.0 standby 192.168.5.152
XML Executes cmd: interface Ethernet1
XML Executes cmd: nameif inside
XML Executes cmd: security-level 100
XML Executes cmd: ip address 192.168.0.151 255.255.255.0 standby 192.168.0.152
XML Executes cmd: !
XML Executes cmd: boot system flash:/f
XML Executes cmd: ftp mode passive
XML Executes cmd: clock timezone jst 9
XML Executes cmd: dns server-group DefaultDNS
XML Executes cmd: domain-name cisco.com
_tcp_listen: could not query index for interface 65535 port 23
XML Executes cmd: pager lines 24
XML Executes cmd: logging console debugging
XML Executes cmd: logging buffered debugging
XML Executes cmd: mtu outside 1500
XML Executes cmd: mtu inside 1500
XML Executes cmd: failover
XML Executes cmd: no asdm history enable
XML Executes cmd: arp timeout 14000
XML Executes cmd: route outside 0.0.0.0 0.0.0.0 192.168.5.1 1
XML Executes cmd: timeout xlate 3:00:00
XML Executes cmd: timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
XML Executes cmd: timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat
0:05:00
XML Executes cmd: timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect
0:02:00
XML Executes cmd: timeout uauth 0:05:00 absolute
XML Executes cmd: username user1 password mb02jYs13AXlIAGa encrypted
XML Executes cmd: username sugi password EB30P7Hu2hSu6x/7 encrypted
XML Executes cmd: http server enable
XML Executes cmd: http 0.0.0.0 0.0.0.0 outside
XML Executes cmd: no snmp-server location
XML Executes cmd: no snmp-server contact
XML Executes cmd: snmp-server enable traps snmp authentication linkup linkdown coldstart
XML Executes cmd: telnet timeout 5
XML Executes cmd: ssh timeout 5
XML Executes cmd: console timeout 0
XML Executes cmd: !
XML Executes cmd: class-map inspection_default
XML Executes cmd: match default-inspection-traffic
XML Executes cmd: !
XML Executes cmd: !
XML Executes cmd: policy-map type inspect dns migrated_dns_map_1
XML Executes cmd: parameters
XML Executes cmd: message-length maximum 512
XML Executes cmd: policy-map global_policy
XML Executes cmd: class inspection_default
XML Executes cmd: inspect ftp
XML Executes cmd: inspect h323 h225
XML Executes cmd: inspect h323 ras
XML Executes cmd: inspect netbios
XML Executes cmd: inspect rsh
XML Executes cmd: inspect rtsp
XML Executes cmd: inspect skinny
XML Executes cmd: inspect esmtp
XML Executes cmd: inspect sqlnet
XML Executes cmd: inspect sunrpc
XML Executes cmd: inspect tftp
XML Executes cmd: inspect sip
XML Executes cmd: inspect xdmcp
XML Executes cmd: !
XML Executes cmd: service-policy global_policy global
XML error info: cmd-id 87 type info

```

```
XML Executes cmd: prompt hostname context
XML Executes cmd: crashinfo save disable
```

The following is sample output from the **debug xml event** command:

```
hostname# debug xml event
debug xml event enabled at level 1

XML parsing: data = <con... len = 3176
Exit XML parser, ret code = 0
```

Related Commands

Command	Description
show debug	Displays the debugging status for the various debug commands.

