d - r

debug

To show debugging messages for a given feature, use the **debug** command. To disable the display of debug messages, use the **no** form of this command. Use **no debug all** to turn off all debugging commands.

debug *feature* [*subfeature*] [*level*]
no debug *feature* [*subfeature*]

Syntax Description

feature	Specifies the feature for which you want to enable debugging. To see available features, use the **debug ?** command for CLI help.
subfeature	(Optional) Depending on the feature, you can enable debug messages for one or more subfeatures. Use ? to see the available subfeatures.
level	(Optional) Specifies the debugging level. The level might not be available for all features. Use ? to see the available levels.

Command Default

The default debugging level is 1.

Command History

Release	Modification
6.1	This command was introduced.
7.2	This command was modified to include the debug for path monitoring.

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 the Cisco Technical Assistance Center (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.

You can view debug output in a CLI session only. Output is directly available when connected to the Console port, or when in the diagnostic CLI (enter **system support diagnostic-cli**). You can also view output from the regular threat defense CLI using the **show console-output** command.

Example

The following example enables DNS debugging and performs an action that generates messages in the diagnostic CLI. The debug messages start after the “ERROR: % Invalid Hostname” message. Press enter to get to the prompt. The example then shows what these debug messages would look like in the **show console-output** display.

```
> debug dns
debug dns enabled at level 1.

> system support diagnostic-cli
Attaching to Diagnostic CLI ... Press 'Ctrl+a then d' to detach.
Type help or '?' for a list of available commands.

firepower# ping www.example.com
```

```
                ^
ERROR: % Invalid Hostname
firepower# DNS: get global group DefaultDNS handle 1fa0b047
DNS: Resolve request for 'www.example.com' group DefaultDNS
DNS: No interfaces enabled
DNS: get global group DefaultDNS handle 1fa0b047
DNS: Resolve request for 'www.example.com' group DefaultDNS
DNS: No interfaces enabled

firepower# (press Ctrl+a, then d, to return to the regular CLI.)

Console connection detached.
> show console-output
... (output redacted)...
Message #75 : DNS: get global group DefaultDNS handle 1fa0b047
Message #76 : DNS: Resolve request for 'www.cisco.com' group DefaultDNS
Message #77 : DNS: No interfaces enabled
Message #78 : DNS: get global group DefaultDNS handle 1fa0b047
Message #79 : DNS: Resolve request for 'www.cisco.com' group DefaultDNS
Message #80 : DNS: No interfaces enabled
```

Related Commands

Command	Description
show debug	Shows the currently active debug settings.
undebug	Disables debugging for a feature. This command is a synonym for **no debug**.

debug packet-condition

To apply the filters on the flows that must be debugged, use the **debug packet-condition** command. To remove the filters on the flows, use the **no** form of this command. Use **no debug packet-condition** to turn off all the filters on the flows.

debug packet-condition [**position** *<line>*] **match** *<proto> {any|any4|any6|host <ip>|<ipv4>|<ipv4_mask>|<ipv6>/<prefixlen>}* [*<src_operator> <ports> {any|any4|any6|host <ip>|<ipv4>|<ipv4_mask>|<ipv6>/<prefixlen>}*] [*<dest_operator> <ports>*] [*<icmp_type>* | *<icmp6_type>*] [*connection <connection-id>*] [*unidirectional*]

Syntax Description

position *<line>*	Specifies the position at which the filter should be placed in the list of existing filters. *<line>* indicates the number.
match*<proto> {any|any4|any6|host <ip>|<ipv4>|<ipv4_mask>| <ipv6>/<prefixlen>}*	Specifies the matching condition for the filter. *<proto>* indicates the protocol. *{any|any4|any6|host <ip>|<ipv4>|<ipv4_mask>|<ipv6>/<prefixlen>}* indicates the IP address options.
<src_operator><port> {any|any4|any6|host <ip>|<ipv4>|<ipv4_mask>| <ipv6>/<prefixlen>}	(Optional) Specifies the port or IP address details of the source.
<dest_operator><port> {any|any4|any6|host <ip>|<ipv4>|<ipv4_mask>| <ipv6>/<prefixlen>}	(Optional) Specifies the port or IP address details of the destination.
<icmp_type>|<icmp6_type>	(Optional) Specifies the ICMP type of the connection.
connection <connection-id>	(Optional) Specifies the connection ID of an ongoing connection.
unidirectional	(Optional) Specifies that the debugging should be performed only on packets in the specified direction. If the variable is not provided, then the default behavior is bi-directional, wherein the traffic will be matched with both the forward and the reverse flows of the connection.

Command Default

Command History

Release	Modification
6.4	This command was introduced.
6.5	The command was changed from **debug packet condition** to **debug packet-condition**.

Release	Modification
6.6	The command **debug packet-condition** was enhanced to provide support for ongoing connections.

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 the Cisco Technical Assistance Center (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.

You can view debug output in a CLI session only. Output is directly available when connected to the Console port, or when in the diagnostic CLI (enter **system support diagnostic-cli**). You can also view output from the regular threat defense CLI using the **show console-output** command.

Example

The following examples show how you can set filters to the flows that must be debugged.

```
> debug packet-condition position 7 match tcp 1.2.3.0 255.255.255.0 any4
> debug packet-condition match tcp 1.2.3.0 255.255.255.0 eq www any4 unidirectional
> debug packet-condition match connection 70856531
> no debug packet-condition match tcp 1.2.3.0 255.255.255 eq www unidirectional
```

Related Commands

Command	Description
debug packet-start	Open the connection to debug logs database and start writing debug logs to the database.
debug packet-stop	Closes the connection to debug logs database and stops writing debug logs to the database.

debug packet-module

To set the level for each module to send debug messages, use the **debug packet-module** command. The level can be set from 0 (emergencies) to 7 (debug). After a level is set, all the messages with equal or higher severity are logged.

debug packet-module [**acl** | **all** | **appid** | **daq** | **pdts** | **snort-engine** | **snort-fileprocessor** | **snort-firewall**] < *0-7* >

Syntax Description

acl	Selects the access control policies in the packet processing path.
all	Selects all the modules in the packet processing path.
daq	Selects the DAQ information in the packet processing path.
pdts	Selects the PDTS (data plane transmit/receive queues to snort) communication in the packet processing path.
snort-engine	Selects the Snort information in the packet processing path.
snort-fileprocessor	Selects the Snort file processor information in the packet processing path.
snort-firewall	Selects the Snort firewall information in the packet processing path.

Command History

Release	Modification
6.4	This command was introduced.
6.5	The command was changed from **debug packet** to **debug packet-module**.

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 the Cisco Technical Assistance Center (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.

You can view debug output in a CLI session only. Output is directly available when connected to the Console port, or when in the diagnostic CLI (enter **system support diagnostic-cli**). You can also view output from the regular threat defense CLI using the **show console-output** command.

Examples

The following example shows how you can set a level to the DAQ information in the packet processing path.

```
> debug packet daq 6
```

Related Commands

Command	Description
debug packet-start	Open the connection to debug logs database and start writing the debug logs to the database.
debug packet-stop	Closes the connection to debug logs database and stops writing the debug logs to the database.

debug packet-module trace

To enable module level packet tracing, use the **debug packet-module trace** command.

debug packet-module trace

Command History

Release	Modification
6.6	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 the Cisco Technical Assistance Center (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.

You can view debug output in a CLI session only. Output is directly available when connected to the Console port, or when in the diagnostic CLI (enter **system support diagnostic-cli**). You can also view output from the regular threat defense CLI using the **show console-output** command.

Example

The following example shows how you can enable module level packet tracing.

```
> debug packet-module trace
```

The following is sample output from the **debug packet-module trace** command:

```
ID         | Details                                                                | Time (ns)

----------- ------------------------------------------------------------------------
------------------------------
6525759    | TCP        74.125.24.156      : 443   -> 192.168.0.31      : 58280 | 19-02-2020
 06:48:43.050675868
```

Further, details of the packet can be derived by using the following command.

```
> show packet debugs module trace packet-id 6525759
```

```
Module: tcp-normalizer
Entry Time: 19-02-2020 06:48:43.050675868(ns)
*****************************************
Module: translate
Entry Time: 19-02-2020 06:48:43.050684452(ns)
*****************************************
Module: inspect_snort
Entry Time: 19-02-2020 06:48:43.050688028(ns)
*****************************************
Module: pdts
Entry Time: 19-02-2020 06:48:43.050691843(ns)
*****************************************
Module: pdts
Entry Time: 19-02-2020 06:48:43.051417112(ns)
*****************************************
Module: pdts
Entry Time: 19-02-2020 06:48:43.051421642(ns)
*****************************************
Module: tcp-normalizer
```

```
Entry Time: 19-02-2020 06:48:43.051424980(ns)
*******************************************
Module: adjacency
Entry Time: 19-02-2020 06:48:43.051438331(ns)
*******************************************
Module: fragment
Entry Time: 19-02-2020 06:48:43.051442861(ns)
*******************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.750763893(ns)
*******************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.750815391(ns)
*******************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.750831365(ns)
*******************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.750843286(ns)
*******************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.750889778(ns)
*******************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.750911474(ns)
*******************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.750942230(ns)
*******************************************
Module: snort_engine
Entry Time: 19-02-2020 06:48:43.750986576(ns)
*******************************************
Module: snort_engine
Entry Time: 19-02-2020 06:48:43.750999689(ns)
*******************************************
Module: snort_engine
Entry Time: 19-02-2020 06:48:43.751020193(ns)
*******************************************
Module: snort_engine
Entry Time: 19-02-2020 06:48:43.751051425(ns)
*******************************************
Module: snort_firewall
Entry Time: 19-02-2020 06:48:43.751075029(ns)
*******************************************
Module: snort_firewall
Entry Time: 19-02-2020 06:48:43.751084804(ns)
*******************************************
Module: snort_engine
Entry Time: 19-02-2020 06:48:43.751099348(ns)
*******************************************
Module: snort_engine
Entry Time: 19-02-2020 06:48:43.751118421(ns)
*******************************************
Module: snort_engine
Entry Time: 19-02-2020 06:48:43.751137018(ns)
*******************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.751152753(ns)
*******************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.751164197(ns)
*******************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.751177072(ns)
```

```
*****************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.751186609(ns)
*****************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.751203775(ns)
*****************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.751224517(ns)
*****************************************
Module: daq
Entry Time: 19-02-2020 06:48:43.751236677(ns)
*****************************************
```

Related Commands

Command	Description
show packet debugs module trace	Displays the list of all the debug traces collected from each module.
debug packet-start	Open the connection to debug logs database and start writing the debug logs to the database.
debug packet-stop	Closes the connection to debug logs database and stops writing the debug logs to the database.

debug packet-start

To start debugging of packets and to start writing debug logs to the debug log database, use the **debug packet-start** command.

debug packet-start

Command History

Release	Modification
6.4	This command was introduced.
6.5	This command was changed from **debug packet start** to **debug packet-start**.

Usage Guidelines

The **debug packet-start** opens the connection to the debug log database. Debug logs are not written to the database unless this command is invoked.

Example

The following example shows how to start debugging packets:

```
> debug packet-start
```

Related Commands

Command	Description
debug packet-stop	Closes the connection to debug logs database and stops writing debug logs to the database.

debug packet-stop

To stop debugging of packets and to stop writing debug logs to the debug log database, use the **debug packet-stop** command.

debug packet-stop

Command History

Release	Modification
6.4	This command was introduced.
6.5	This command was changed from **debug packet stop** to **debug packet-stop**.

Usage Guidelines

The **debug packet-stop** closes the connection to the debug log database.

Example

The following example shows how to stop debugging packets:

```
> debug packet-stop
```

Related Commands

Command	Description
debug packet-start	Open the connection to debug logs database and start writing debug logs to the database.

delete

To delete a file from flash memory, use the **delete** command.

delete /noconfirm [**/recursive**] [**/replicate**] [**disk0:** | **disk***n***:** | **flash:**] [*path/*] *filename*

Syntax Description

/noconfirm	Does not prompt for confirmation.
/recursive	(Optional) Deletes the specified file recursively in all subdirectories.
/replicate	(Optional) Deletes the specified file on the standby unit.
disk0:	(Optional) Specifies the internal flash memory.
disk*n***:**	(Optional) Indicates optional external flash drive, where n specifies the drive number. This is typically disk1:
filename	Specifies the name of the file to delete.
flash:	(Optional) Specifies the internal flash memory. This keyword is the same as **disk0**.
path/	(Optional) Specifies to the path to the file.

Command Default

If you do not specify a directory, the directory is the current working directory by default.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The file is deleted from the current working directory if a path is not specified. Wildcards are supported when deleting files.

Examples

The following example shows how to delete a file named test.cfg in the current working directory:

```
> delete /noconfirm test.cfg
```

Related Commands

Command	Description
cd	Changes the current working directory to the one specified.
dir	List the files in the current directory.
rmdir	Removes a file or directory.

dig

To look up the IP address for a fully-qualified domain name, use the **dig** command.

dig *hostname*

Syntax Description

hostname	The fully-qualified domain name of a host whose IP address you are looking up. For example, www.example.com.

Command History

Release	Modification
7.1	This command was introduced. It replaced the **nslookup** command.

Usage Guidelines

Some commands that allow fully-qualified domain names cannot use the DNS servers configured for the management interface to look up the IP address for the name. If you do not have DNS servers configured for commands that go through the data interfaces, use the **dig** command to determine the IP address, then use the IP address in the command.

The **dig** command works through the management interface only, and returns information from the DNS servers configured for the management interface. If you configure different servers for the data interfaces, using an FQDN on a command that goes through a data interface might return a different IP address, or no IP address at all if those DNS servers cannot resolve the name.

Example

The following example looks up the IP address of the FQDN www.example.com. The address is highlighted in the ANSWER section of the output. The SERVER indication near the end of the output shows the IP address of the DNS server that returned the resolution (the IP address in this example has been sanitized).

The NOERROR status in the header indicates the request was successful; any other value represents an error. For example, NXDOMAIN means the domain name does not exist in the responding DNS server. You can search the internet for more details about reading the output of the Linux dig command.

```
> dig www.example.com
; <<>> DiG 9.11.4 <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14008
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
; COOKIE: 88335c9f3dc2ca124e36b5eb60db9067b6cae4de2ea5bffb (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        0       IN      A       93.184.216.34
;; AUTHORITY SECTION:
example.com.            58911   IN      NS      a.iana-servers.net.
```

```
example.com.            58911   IN      NS      b.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     0       IN      A       199.43.135.53

;; Query time: 12 msec
;; SERVER: 10.163.47.11#53(10.163.47.11)
;; WHEN: Tue Jun 29 21:28:07 UTC 2021
;; MSG SIZE  rcvd: 152
```

dir

To display the directory contents, use the dir command.

dir [**/all**] [**all-filesystems**] [**/recursive**] [**disk0:** | **disk***n***:** | **flash:** | **system:**] [*path*] [*filename*]

Syntax Description

/all	(Optional) Displays all files.
/recursive	(Optional) Displays the directory contents recursively.
all-filesystems	(Optional) Displays the files of all filesystems.
disk0:	(Optional) Specifies the internal Flash memory, followed by a colon.
disk*n***:**	(Optional) Indicates optional external flash drive, where *n* specifies the drive number. This is typically disk1:
flash:	(Optional) Displays the directory contents of the default flash partition.
path	(Optional) Specifies a specific path.
filename	(Optional) Specifies the name of a file.
system:	(Optional) Displays the directory contents of the file system.

Command Default

If you do not specify a directory, the directory is the current working directory by default.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows how to display the directory contents:

```
> dir
Directory of disk0:/
1      -rw-  1519        10:03:50 Jul 14 2003    my_context.cfg
2      -rw-  1516        10:04:02 Jul 14 2003    my_context.cfg
3      -rw-  1516        10:01:34 Jul 14 2003    admin.cfg
60985344 bytes total (60973056 bytes free)
```

Related Commands

Command	Description
cd	Changes the current working directory to the one specified.
pwd	Displays the current working directory.

Command	Description
mkdir	Creates a directory.
rmdir	Removes a directory.

dns update

To start DNS lookup to resolve the designated hostnames without waiting for the expiration of the DNS poll timer, use the **dns update** command.

dns update [**host** *fqdn_name*] [**timeout seconds** *number*]

Syntax Description

host *fqdn_name*	Specifies the fully qualified domain name of the host on which to run DNS updates.
timeout seconds *number*	Specifies the timeout for the lookup operation, in seconds, from 3-30. The default is 30.

Command History

Release	Modification
6.3	This command was introduced.

Usage Guidelines

This command immediately starts a DNS lookup to resolve the designated hostnames without waiting for the expiration of the DNS poll timer. When you run DNS update without specifying a hostname, all names that are used in access control rules, which is known as being activated, are resolved. When the command finishes running, the system displays [Done] at the command prompt and generates a syslog message.

Examples

The following example performs a DNS update for all FQDNs used in access control rules.

```
> dns update
INFO: update dns process started
> [Done]
```

Related Commands

Command	Description
clear dns	Removes FQDN network object DNS resolutions.
show dns	Displays FQDN network object DNS resolutions.

eotool commands

Only use **eotool** commands under the direction of the Cisco Technical Assistance Center.

exit

To exit from the CLI, use the **exit** command.

exit

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

In the regular CLI, the **exit** and **logout** commands do the same thing, closing the SSH session with the device.

When you are in expert mode, **exit** leaves expert mode and returns you to the regular CLI.

When you are in the Diagnostic CLI (**system support diagnostic-cli**), the **exit** command also moves you from Privileged EXEC mode back to User EXEC mode.

Examples

The following example shows how to use the **exit** command to close the SSH connection to the CLI.

```
> exit
```

The following example shows how to use the **exit** command go from Privileged EXEC mode in the Diagnostic CLI (represented by the # sign in the prompt) back to User EXEC mode. You can ignore the Logoff message, your CLI session remains active.

```
firepower# exit
Logoff
Type help or '?' for a list of available commands.
firepower>
```

Related Commands

Command	Description
logout	Logs off from the CLI session.

expert

To enter expert mode, which is required for some procedures, use the **expert** command.

expert

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use expert mode only if a documented procedure tells you to enter it, or if the Cisco Technical Assistance Center tells you to use it. The use of expert mode is unsupported under any other circumstances.

Caution You might be able to execute commands in expert mode whose results are not reflected in device manager. Use documented commands only in expert mode, or commands as directed by Cisco Technical Support, to avoid unintended results.

Examples

The following example shows how to enter and exit expert mode. The expert mode prompt shows the username@hostname information.

```
> expert
admin@firepower:~$
admin@firepower:~$ exit
logout
>
```

Related Commands

Command	Description
exit	Exit from expert mode.

failover active

To switch a standby device to the active state, use the **failover active** command. To switch an active device to standby, use the **no** form of this command.

failover active
no failover active

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use the **failover active** command to initiate a failover switch from the standby unit, or use the **no failover active** command from the active unit to initiate a failover switch. You can use this feature to return a failed unit to service, or to force an active unit offline for maintenance. If you are not using Stateful Failover, all active connections are dropped and must be reestablished by the clients after the failover occurs.

Examples

The following example switches the standby unit to active:

```
> failover active
```

Related Commands

Command	Description
failover reset	Moves a device from a failed state to standby.

failover exec

To execute a command on a specific unit in a failover pair, use the **failover exec** command.

failover exec {**active** | **standby** | **mate**} *cmd_string*

Syntax Description

active	Specifies that the command is executed on the active unit in the failover pair.
cmd_string	The command to be executed. See the CLI help for supported commands.
mate	Specifies that the command is executed on the failover peer.
standby	Specifies that the command is executed on the standby unit in the failover pair.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

You can use the **failover exec** command to send commands to a specific unit in a failover pair.

Output from the commands is displayed in the current terminal session, so you can use the **failover exec** command to issue **show** commands on a peer unit and view the results in the current terminal.

You must have sufficient privileges to execute a command on the local unit to execute the command on the peer unit.

Limitations

- Command completion and context help are not available for the commands in the *cmd_string* argument.
- You cannot use the **debug** (**undebug**) command with the **failover exec** command.
- If the standby unit is in the failed state, it can still receive commands from the **failover exec** command if the failure is due to a service card failure; otherwise, the remote command execution will fail.
- You cannot enter recursive **failover exec** commands, such as the **failover exec mate failover exec mate** *command*.
- Commands that require user input or confirmation must use the **/nonconfirm** option.

Examples

The following example uses the **failover exec** command to display the failover configuration of the failover peer. The command is executed on the primary unit, which is the active unit, so the information displayed is from the secondary, standby unit.

```
> failover exec mate show running-config failover
failover
failover lan interface failover GigabitEthernet0/3
failover polltime unit 1 holdtime 3
failover polltime interface 3 holdtime 15
failover link failover GigabitEthernet0/3
```

```
failover interface ip failover 10.0.5.1 255.255.255.0 standby 10.0.5.2
```

The following example uses the **failover exec** command to send the **show interface** command to the standby unit:

```
> failover exec standby show interface
Interface GigabitEthernet0/0 "outside", is up, line protocol is up
  Hardware is i82546GB rev03, BW 1000 Mbps
      Auto-Duplex(Half-duplex), Auto-Speed(100 Mbps)
      MAC address 000b.fcf8.c290, MTU 1500
      IP address 192.168.5.111, subnet mask 255.255.255.0
      216 packets input, 27030 bytes, 0 no buffer
      Received 2 broadcasts, 0 runts, 0 giants
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
      0 L2 decode drops
      284 packets output, 32124 bytes, 0 underruns
      0 output errors, 0 collisions
      0 late collisions, 0 deferred
      input queue (curr/max blocks): hardware (0/0) software (0/0)
      output queue (curr/max blocks): hardware (0/1) software (0/0)
  Traffic Statistics for "outside":
      215 packets input, 23096 bytes
      284 packets output, 26976 bytes
      0 packets dropped
      1 minute input rate 0 pkts/sec,  21 bytes/sec
      1 minute output rate 0 pkts/sec,  23 bytes/sec
      1 minute drop rate, 0 pkts/sec
      5 minute input rate 0 pkts/sec,  21 bytes/sec
      5 minute output rate 0 pkts/sec,  24 bytes/sec
      5 minute drop rate, 0 pkts/sec
Interface GigabitEthernet0/1 "inside", is up, line protocol is up
  Hardware is i82546GB rev03, BW 1000 Mbps
      Auto-Duplex(Half-duplex), Auto-Speed(10 Mbps)
      MAC address 000b.fcf8.c291, MTU 1500
      IP address 192.168.0.11, subnet mask 255.255.255.0
      214 packets input, 26902 bytes, 0 no buffer
      Received 1 broadcasts, 0 runts, 0 giants
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
      0 L2 decode drops
      215 packets output, 27028 bytes, 0 underruns
      0 output errors, 0 collisions
      0 late collisions, 0 deferred
      input queue (curr/max blocks): hardware (0/0) software (0/0)
      output queue (curr/max blocks): hardware (0/1) software (0/0)
  Traffic Statistics for "inside":
      214 packets input, 23050 bytes
      215 packets output, 23140 bytes
      0 packets dropped
      1 minute input rate 0 pkts/sec,  21 bytes/sec
      1 minute output rate 0 pkts/sec,  21 bytes/sec
      1 minute drop rate, 0 pkts/sec
      5 minute input rate 0 pkts/sec,  21 bytes/sec
      5 minute output rate 0 pkts/sec,  21 bytes/sec
      5 minute drop rate, 0 pkts/sec
Interface GigabitEthernet0/2 "failover", is up, line protocol is up
  Hardware is i82546GB rev03, BW 1000 Mbps
      Auto-Duplex(Full-duplex), Auto-Speed(100 Mbps)
      Description: LAN/STATE Failover Interface
      MAC address 000b.fcf8.c293, MTU 1500
      IP address 10.0.5.2, subnet mask 255.255.255.0
      1991 packets input, 408734 bytes, 0 no buffer
      Received 1 broadcasts, 0 runts, 0 giants
```

```
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
      0 L2 decode drops
      1835 packets output, 254114 bytes, 0 underruns
      0 output errors, 0 collisions
      0 late collisions, 0 deferred
      input queue (curr/max blocks): hardware (0/0) software (0/0)
      output queue (curr/max blocks): hardware (0/2) software (0/0)
  Traffic Statistics for "failover":
      1913 packets input, 345310 bytes
      1755 packets output, 212452 bytes
      0 packets dropped
      1 minute input rate 1 pkts/sec,  319 bytes/sec
      1 minute output rate 1 pkts/sec,  194 bytes/sec
      1 minute drop rate, 0 pkts/sec
      5 minute input rate 1 pkts/sec,  318 bytes/sec
      5 minute output rate 1 pkts/sec,  192 bytes/sec
      5 minute drop rate, 0 pkts/sec
...
```

The following example shows the error message returned when issuing an illegal command to the peer unit:

```
> failover exec mate bad command
bad command
  ^
ERROR: % Invalid input detected at '^' marker.
```

The following example shows the error message that is returned when you use the **failover exec** command when failover is disabled:

```
> failover exec mate show failover
ERROR: Cannot execute command on mate because failover is disabled
```

Related Commands

Command	Description
debug fover	Displays failover-related debugging messages.
debug xml	Displays debugging messages for the XML parser used by the **failover exec** command.
show failover exec	Displays the **failover exec** command mode.

failover reload-standby

To force the standby unit to reboot, use the **failover reload-standby** command.

failover reload-standby

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use this command when your failover units do not synchronize. The standby unit restarts and resynchronizes to the active unit after it finishes booting.

Examples

The following example shows how to use the **failover reload-standby** command on the active unit to force the standby unit to reboot:

```
> failover reload-standby
```

failover reset

To restore a failed device to an unfailed state, use the **failover reset** command.

failover reset

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **failover reset** command allows you to change the failed unit to an unfailed state. The **failover reset** command can be entered on either unit, but we recommend that you always enter the command on the active unit. Entering the **failover reset** command at the active unit will “unfail” the standby unit.

You can display the failover status of the unit with the **show failover** command.

Examples

The following example shows how to change a failed unit to an unfailed state:

```
> failover reset
```

Related Commands

Command	Description
show failover	Displays information about the failover status of the unit.

file copy

To transfer files from the common directory to a remote host via FTP, use the **file copy** command.

file copy *host_name* *user_id path* *filename_1* [*filename_2* **...** *filename_n*]

Syntax Description

host_name	Specifies the name or IP address of the target remote host.
user_id	Specifies the user on the remote host.
path	Specifies the destination path on the remote host.
filename_1 through *filename_n*	Specifies the names of the files to transfer from the common directory. If multiple file names are specified, they must be separated with blanks. This argument supports wildcards.

Command Default

This command transfers files only from the common directory where the system writes troubleshooting files.

Command History

Release	Modification
6.0.1	This command was introduced.

Examples

This example transfers all files in the common directory to the **/pub** directory on the remote host **sentinel** accessed via user **jdoe**:

```
> file copy sentinel jdoe /pub *
```

Related Commands

Command	Description
file list	List files in the common directory.
file delete	Delete files from the common directory.
file secure-copy	Transfer files in the common directory via SCP.

file delete

To erase files from the common directory, use the **file delete** command.

file delete *filename_1* [*filename_2* . . . *filename_n*]

Syntax Description

filename_1 through *filename_n*	Specifies the names of the files to delete from the common directory. If multiple file names are specified, they must be separated with blanks. This argument supports wildcards.

Command Default

This command operates only on files in the common directory where the system writes troubleshooting files.

Command History

Release	Modification
6.0.1	This command was introduced.

Examples

This example deletes a single file:

```
> file delete 10.83.170.31-43235986-2363-11e6-b278-aff0a43948fe-troubleshoot.tar.gz
```

Related Commands

Command	Description
file list	List files in the common directory.
file copy	Transfer files in the common directory via FTP.
file secure-copy	Transfer files in the common directory via SCP.

file list

To list the files in the common directory, use the **file list** command.

file list [*filename_1* **...** *filename_n*]

Syntax Description

filename_1 through *filename_n*	Specifies the names of the files to list from the common directory. If multiple file names are specified, they must be separated with blanks. This argument supports wildcards.

Command History

Release	Modification
6.0.1	This command was introduced.

Usage Guidelines

This command lists only files in the common directory where the system writes troubleshooting files. If no file names are specified, all files in the common directory are listed.

Examples

This example lists the contents of the common directory:

```
> file list
May 26 17:46        137474048 /core_1464284811_rackham-sfr.cisco.com_diskmanager_11.21145
Jun 27 20:36       1464696832 /core_1467059810_rackham-sfr.cisco.com_lina_6.21293
```

Related Commands

Command	Description
file copy	Transfer files in the common directory via FTP.
file delete	Delete files from the common directory.
file secure-copy	Transfer files in the common directory via SCP.

file secure-copy

To transfer files from the common directory to a remote host via SCP, use the **file secure-copy** command.

file secure-copy *host_name* *user_id path* *filename_1* [*filename_2* . . . *filename_n*]

Syntax Description

host_name	Specifies the name or IP address of the target remote host.
user_id	Specifies the user on the remote host.
path	Specifies the destination path on the remote host.
filename_1 through *filename_n*	Specifies the names of the files to transfer from the common directory. If multiple file names are specified, they must be separated with blanks. This argument supports wildcards.

Command Default

This command transfers files only from the common directory where the system writes troubleshooting files.

Command History

Release	Modification
6.0.1	This command was introduced.

Examples

This example transfers all files in the common directory to the **/tmp** directory on the remote host **101.123.31.1** accessed via user **jdoe**:

```
> file secure-copy 101.123.31.1 jdoe /tmp *
```

Related Commands

Command	Description
file copy	Transfer files in the common directory via FTP.
file delete	Delete files from the common directory.
file list	List files in the common directory.

fsck

To perform a file system check and to repair corruptions, use the **fsck** command.

fsck /noconfirm disk*n*:

Syntax Description

disk*n*:	Specifies the flash memory drive, where *n* is the drive number.
/noconfirm	Specifies that the command runs without prompting. This keyword is required.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **fsck** command checks and tries to repair corrupt file systems. Use this command before trying more permanent procedures.

If the FSCK utility fixes an instance of disk corruption (due to power failure or abnormal shutdown, for example), it creates recovery files named FSCKxxx.REC. These files can contain a fraction of a file or a whole file that was recovered while FSCK was running. In rare circumstances, you might need to inspect these files to recover data; generally, these files are not needed, and can be safely deleted.

Note The FSCK utility runs automatically at startup, so you may see these recovery files even if you did not manually enter the **fsck** command.

Examples

The following example shows how to check the file system of the flash memory:

```
> fsck /noconfirm disk0:
```

Related Commands

Command	Description
delete	Removes all user-visible files.
erase	Deletes all files and formats the flash memory.
format	Formats the file system.

help

To display help information for a specified command, use the **help** command.

help {*command* | **?**}

Syntax Description

?	Displays all commands for which help is available.
command	Specifies the command for which to display the CLI help.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **help** command displays help information about some commands. You can see help for an individual command by entering the **help** command followed by the command name. If you do not specify a command name and enter **?** instead, all commands for which there is help are listed.

You can also get help by entering ? after entering a partial command. This shows you the valid parameters at that location in the command string.

Examples

The following example shows how to display help for the **traceroute** command:

```
> help traceroute
USAGE:
        traceroute <destination> [source <src_address|src_intf>]
                   [numeric] [timeout <time>] [ttl <min-ttl> <max-ttl>]
                   [probe <probes>] [port <port-value>] [use-icmp]
DESCRIPTION:
traceroute      Print the route packets take to a network host
SYNTAX:
destination     Address or hostname of destination
src_address     Source address used in the outgoing probe packets
src_intf        Interface through which the destination is accessible
numeric         Do not resolve addresses to hostnames
time            The time in seconds to wait for a response to a probe
min-ttl         Minimum time-to-live value used in probe packets
max-ttl         Maximum time-to-live value used in probe packets
probes          The number of probes to send for each TTL value
port-value      Base UDP destination port used in probes
use-icmp        Use ICMP probes instead of UDP probes
```

history

To display the command line history for the current session, use the **history** command.

history *limit*

Syntax Description

limit	The size of the history list in number of entries. To set the size to unlimited, that is, to see the full history, enter 0.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

You can also use the up arrow to scroll through past commands.

The history view includes sequence numbers for the order in which the commands were entered.

Examples

The following example shows the command history.

```
> history 0
   48  show environment
   49  show network-static-routes
   50  show network
   51  show running-config
   52  show service-policy
   53  show ntp
   54  show cpu
   55  show memory
   56  history 0
>
```

local-base-url

(Optional) Configures the local base URL of the SAML service provider for VPN authentication. In a DNS load balancing cluster, when you configure SAML authentication on threat defenses, you can specify this URL to uniquely resolve to the device on which the configuration is applied.

To disable this feature, use the **no** form of this command

local base-url { **url** }
no local base-url

Syntax Description

url Local base URL of the SAML service provider for VPN authentication.

Command Default

None.

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
Parameters configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.4.0	This command was added.
7.2.4	This command was added.

Usage Guidelines

You must use this command in conjunction with the **base-url** command. Versions 7.3.0 and 7.3.1 and versions lower than 7.2.4 do not support this option.

Examples

The following example sets up a local base-url:

```
ciscoasa(config)# webvpn
ciscoasa(config-webvpn)# saml idp https://idp.com/<app-specific>
ciscoasa(config-webvpn-saml-idp)# base url https://ftd-dns-group.vpn.customer.com
ciscoasa(config-webvpn-saml-idp)# local-base-url https://this-ftd.vpn.customer.com
```

logging savelog

To save the log buffer to flash memory, use the **logging savelog** command.

logging savelog [*savefile*]

Syntax Description

savefile	(Optional) The file name for the saved log. If you do not specify the file name, the system saves the log file using a default time-stamp format, as follows: `LOG-YYYY-MM-DD-HHMMSS.TXT` where *YYYY* is the year, *MM* is the month, *DD* is the day of the month, and *HHMMSS* is the time in hours, minutes, and seconds.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Before you can save the log buffer to flash memory, you must enable logging to the buffer; otherwise, the log buffer never has data to be saved to flash memory. However, if the configured logging buffer size is more than 2MB, the internal log buffer will not be written to flash memory. Configure buffer logging using management center (remote) or device manager (local).

Note The **logging savelog** command does not clear the buffer. To clear the buffer, use the **clear logging buffer** command.

Examples

The following example saves the log buffer to flash memory using the file name, latest-logfile.txt:

```
> logging savelog latest-logfile.txt
>
```

Related Commands

Command	Description
clear logging buffer	Clears the log buffer of all syslog messages that it contains.
copy	Copies a file from one location to another, including to a TFTP or FTP server.
delete	Deletes a file from the disk partition, such as saved log files.

logout

To exit from the CLI, use the **logout** command.

logout

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **logout** command lets you log out of the device and end your CLI session. You can also use the **exit** command.

Examples

The following example shows how to log out of the device:

```
> logout
```

memory caller-address

To configure a specific range of program memory for the call tracing, or caller PC, to help isolate memory problems, use the **memory caller-address** command. The caller PC is the address of the program that called a memory allocation primitive. To remove an address range, use the no form of this command.

memory caller-address *startPC endPC*
no memory caller-address

Syntax Description

endPC	Specifies the end address range of the memory block.
startPC	Specifies the start address range of the memory block.

Command Default

The actual caller PC is recorded for memory tracing.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use the **memory caller-address** command to isolate memory problems to a specific block of memory.

In certain cases the actual caller PC of the memory allocation primitive is a known library function that is used at many places in the program. To isolate individual places in the program, configure the start and end program address of the library function, thereby recording the program address of the caller of the library function.

Note The device might experience a temporary reduction in performance when caller-address tracing is enabled.

Examples

The following examples show the address ranges configured with the **memory caller-address** commands, and the resulting display of the **show memory caller-address** command:

```
> memory caller-address 0x00109d5c 0x00109e08
> memory caller-address 0x009b0ef0 0x009b0f14
> memory caller-address 0x00cf211c 0x00cf4464
> show memory caller-address
Move down stack frame for the addresses:
pc = 0x00109d5c-0x00109e08
pc = 0x009b0ef0-0x009b0f14
pc = 0x00cf211c-0x00cf4464
```

Related Commands

Command	Description
memory profile enable	Enables the monitoring of memory usage (memory profiling).

Command	Description
memory profile text	Configures a text range of memory to profile.
show memory	Displays a summary of the maximum physical memory and current free memory available to the operating system.
show memory binsize	Displays summary information about the chunks allocated for a specific bin size.
show memory profile	Displays information about the memory usage (profiling) of the device.
show memory caller-address	Displays the address ranges configured on the device.

memory delayed-free-poisoner

Use the **memory delayed-free-poisoner** command to set parameters for the delayed free-memory poisoner tool. To enable the delayed free-memory poisoner tool, use the **memory delayed-free-poisoner enable** command. To disable the delayed free-memory poisoner tool, use the **no** form of this command. The delayed free-memory poisoner tool lets you monitor freed memory for changes after it has been released by an application.

memory delayed-free-poisoner {**enable** | **desired-fragment-count** *frag_count* | **desired-fragment-size** *frag-size* | **threshold** *heap_use_percent* | **validate** | **watchdog-percent** *watchdog_limit*}
no memory delayed-free-poisoner enable

Syntax Description

enable	Start operation of the delayed free-memory poisoner tool.
desired-fragment-count *frag_count*	Set the number of memory fragments to keep in the poisoner's queue. Legal values range from 0 to 8192; the default is 16
desired-fragment-size *frag-size*	Set the size in bytes of the contiguous free memory fragments to keep in the poisoner's queue. Legal values range from 0 to 268435456; the default is 102400.
threshold *heap_use_percent*	Set the percentage threshold of system memory use at which the system will release memory from the poisoner's queue, ranging from 0 to 100. The default is 100.
validate	Forces validation of all elements in the delayed-free-poisoner queue.
watchdog-percent *watchdog_limit*	Set the watchdog limit as a percentage of the watchdog threshold, which is 15 seconds. Values range from 10 to 100. The default is 50.

Command Default

The **memory delayed-free-poisoner enable** command is disabled by default.

The default **desired-fragment-count** is 16.

The default **desire-fragment-size** is 102400.

The default **watchdog-percent** is 50.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Enabling the delayed free-memory poisoner tool has a significant impact on memory usage and system performance. The command should be used only under the supervision of the Cisco Technical Assistance Center. It should not be run in a production environment during heavy system usage.

When you enable this tool, requests to free memory by the applications running on the device are written to a FIFO queue. As each request is written to the poisoner's queue, each associated byte of memory that is not required by lower-level memory management is "poisoned" by being written with the value 0xcc.

The freed memory requests remain in the queue until more memory is required by an application than is in the system free memory pool. When more memory is needed, the poisoner seeks at least

desired-fragment-count memory buffers of **desired-fragment-size** bytes in its queue, pulls that memory from the queue, and validates it. You can tune the time it takes the poisoner to satisfy large memory requests by changing the values for **desired-fragment-size** and **desired-fragment-count**.

If the memory is unmodified, it is returned to the system free memory pool and the poisoner reissues the memory request from the application that made the initial request. The process repeats until enough memory for the requesting application is freed.

If the poisoned memory has been modified, then the system forces a crash and produces diagnostic output which can be used to determine the cause of the crash.

The delayed free poisoner includes a watchdog mechanism to prevent processes from excessive resource usage. The watchdog threshold is 15 seconds, and when a process executes continuously for that time without relinquishing the CPU, the poisoner forces a system crash.

You can tune the watchdog behavior by setting the watchdog limit, which indicates a percentage of the 15 second watchdog threshold; the default is 50%. Therefore when the delayed free poisoner is active, by default if a process executes continuously for 7.5 seconds without relinquishing the CPU, further memory allocation requests from that process fail until the process is rescheduled. You can tune this behavior by changing the value of the watchdog limit.

To guard against excessive memory fragmentation and reduce system CPU load, you can set a percentage **threshold** of free memory usage at which the poisoner automatically releases memory from its queue to the system memory pool. (By default, the poisoner does not release memory from its queue until system memory has been exhausted.)

The delayed free-memory poisoner tool periodically performs validation on all of the elements of the queue automatically. You can also start validation manually using the **memory delayed-free-poisoner validate** command. If an element contains unexpected values, then the system forces a crash and produces diagnostic output to determine the cause of the crash. If no unexpected values are encountered, the elements remain in the queue and are processed normally by the tool; the **memory delayed-free-poisoner validate** command does not cause the memory in the queue to be returned to the system memory pool.

The **no** form of the command causes all of the memory referenced by the requests in the queue to be returned to the free memory pool without validation and any statistical counters to be cleared.

Examples

The following example enables the delayed free-memory poisoner tool:

```
> memory delayed-free-poisoner enable
```

The following is sample output when the delayed free-memory poisoner tool detects illegal memory reuse:

```
delayed-free-poisoner validate failed because a
        data signature is invalid at delayfree.c:328.
    heap region:    0x025b1cac-0x025b1d63 (184 bytes)
    memory address: 0x025b1cb4
    byte offset:    8
    allocated by:   0x0060b812
    freed by:       0x0060ae15
Dumping 80 bytes of memory from 0x025b1c88 to 0x025b1cd7
025b1c80:                         ef cd 1c a1 e1 00 00 00  |          ........
025b1c90: 23 01 1c a1 b8 00 00 00 15 ae 60 00 68 ba 5e 02  |  #.........`.h.^.
025b1ca0: 88 1f 5b 02 12 b8 60 00 00 00 00 00 6c 26 5b 02  |  ..[...`.....l&[.
```

```
025b1cb0: 8e a5 ea 10 ff ff ff ff cc cc cc cc cc cc cc cc  |  ................
025b1cc0: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc  |  ................
025b1cd0: cc cc cc cc cc cc cc cc                          |  ........
An internal error occurred.  Specifically, a programming assertion was
violated.  Copy the error message exactly as it appears, and get the
output of the show version command and the contents of the configuration
file.  Then call your technical support representative.
assertion "0" failed: file "delayfree.c", line 191
```

The following table describes the significant portion of the output.

Table 1: Illegal Memory Usage Output Description

Field	Description
heap region	The address region and size of the region of memory available for use by the requesting application. This is not the same as the requested size, which may be smaller given the manner in which the system may parcel out memory at the time the memory request was made.
memory address	The location in memory where the fault was detected.
byte offset	The byte offset is relative to the beginning of the heap region and can be used to find the field that was modified if the result was used to hold a data structure starting at this address. A value of 0 or that is larger than the heap region byte count may indicate that the problem is an unexpected value in the lower level heap package.
allocated by/freed by	Instruction addresses where the last malloc/calloc/realloc and free calls where made involving this particular region of memory.
Dumping...	A dump of one or two regions of memory, depending upon how close the detected fault was to the beginning of the region of heap memory. The next eight bytes after any system heap header is the memory used by this tool to hold a hash of various system header values plus the queue linkage. All other bytes in the region until any system heap trailer is encountered should be set to 0xcc.

Related Commands

Command	Description
clear memory delayed-free-poisoner	Clears the delayed free-memory poisoner tool queue and statistics.
show memory delayed-free-poisoner	Displays a summary of the delayed free-memory poisoner tool queue usage.

memory logging

To enable memory logging, use the **memory logging** command. To disable memory logging, use the **no** form of this command.

memory logging 1024-4194304 [**wrap** [**size** [**1-2147483647**] | **process** *process-name*]
no memory logging

Syntax Description

1024-4194304	Specifies the number of logging entries in the memory logging buffer. This is the only required argument to specify.
process *process-name*	Specifies the process to monitor. **Note** The Checkheaps process is completely ignored as a process because it uses the memory allocator in a non-standard way.
size 1-2147483647	Specifies the size and number of entries to monitor.
wrap	Save the buffer when it wraps. It can only be saved once. If it wraps multiple times, it can be overwritten. When the buffer wraps, a trigger is sent to the event manager to enable saving of the data.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

To change memory logging parameters, you must disable it, then reenable it. Use the **show memory logging** command to view the log.

Examples

The following example enables memory logging:

```
> memory logging 202980
```

Related Commands

Command	Description
show memory logging	Displays memory logging results.

memory profile enable

To enable the monitoring of memory usage (memory profiling), use the **memory profile enable** command. To disable memory profiling, use the **no** form of this command.

memory profile enable [**peak** *peak_value*]
no memory profile enable [**peak** *peak_value*]

Syntax Description

peak *peak_value*	Specifies the memory usage threshold at which a snapshot of the memory usage is saved to the peak usage buffer. The contents of this buffer could be analyzed at a later time to determine the peak memory needs of the system.

Command Default

Memory profiling is disabled by default.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Before enabling memory profiling, you must first configure a memory text range to profile with the **memory profile text** command.

Some memory is held by the profiling system until you enter the **clear memory profile** command. See the output of the **show memory profile status** command.

Note The device might experience a temporary reduction in performance when memory profiling is enabled.

Examples

The following example enables memory profiling:

```
> memory profile enable
```

Related Commands

Command	Description
memory profile text	Configures a text range of memory to profile.
show memory profile	Displays information about the memory usage (profiling) of the device.

memory profile text

To configure a program text range of memory to profile, use the **memory profile text** command. To disable, use the no form of this command.

memory profile text {*startPC endPC* | **all**} *resolution*
no memory profile text {*startPC endPC* | **all**} *resolution*

Syntax Description

all	Specifies the entire text range of the memory block.
endPC	Specifies the end text range of the memory block.
resolution	You must set the resolution of tracing for the source text region, from 1-44582263.
startPC	Specifies the start text range of the memory block.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

For a small text range, a resolution of "4" normally traces the call to an instruction. For a larger text range, a coarse resolution is probably enough for the first pass and the range could be narrowed down to a set of smaller regions in the next pass.

After entering the text range with the **memory profile text** command, you must then enter the **memory profile enable** command to begin memory profiling. Memory profiling is disabled by default.

Note The device might experience a temporary reduction in performance when memory profiling is enabled.

Examples

The following example shows how to configure a text range of memory to profile, with a resolution of 100.

```
> memory profile text all 100
```

The following example displays the configuration of the text range and the status of memory profiling (OFF):

```
> show memory profile status
InUse profiling: OFF
Peak profiling: OFF
Memory used by profile buffers: 0 bytes
Profile:
0x00007efc3e0227a8-0x00007efc40aa1f8e(00000100)
```

Note To begin memory profiling, you must enter the **memory profile enable** command. Memory profiling is disabled by default.

Related Commands

Command	Description
clear memory profile	Clears the buffers held by the memory profiling function.
memory profile enable	Enables the monitoring of memory usage (memory profiling).
show memory profile	Displays information about the memory usage (profiling) of the device.

memory tracking

To enable the tracking of heap memory request, use the **memory tracking** command. To disable memory tracking, use the **no** form of this command.

memory tracking {**enable** | **allocates-by-threshold** *min_allocates* | **bytes-threshold** *min_bytes* | **filter-from-address-pool** *address*}
no memory tracking enable

Syntax Description

enable	Enable memory tracking.
allocates-by-threshold *min_allocates*	Address pool entries for callers must make at least this many allocation calls to be included, from 0-4294967295.
bytes-threshold *min_bytes*	Address pool entries for callers must consume at least this many bytes of memory to be included, from 0-4294967295.
filter-from-address-pool *address*	Exclude address pool entries at this address. To determine an address, first enable tracking, then use show memory tracking address. Look for the "allocated by" address in the "memory tracking address pool" listing. For example, if you see the following: ...allocated by 0x00007efc3f80e508 You can exclude it using: **filter-from-address-pool 0x00007efc3f80e508**

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example enables tracking heap memory requests:

```
> memory tracking enable
```

Related Commands

Command	Description
clear memory tracking	Clears all currently gathered information.
show memory tracking	Shows memory tracking results.

more

To display the contents of a file, use the **more** command.

more [**/ascii** | **/binary** | **/ebcdic** | **disk0:** | **disk1:** | **flash:** | **ftp:** | **http:** | **https:** | **tftp:**] *filename*

Syntax Description

/ascii	(Optional) Displays a binary file in binary mode and an ASCII file in binary mode.
/binary	(Optional) Displays any file in binary mode.
/ebcdic	(Optional) Displays binary files in EBCDIC.
disk0:	(Optional) Displays a file on the internal Flash memory.
disk1:	(Optional) Displays a file on the external Flash memory card.
filename	Specifies the name of the file to display.
flash:	(Optional) Specifies the internal Flash memory, followed by a colon. In the ASA 5500 series adaptive security appliance, the **flash** keyword is aliased to **disk0**.
ftp:	(Optional) Displays a file on an FTP server.
http:	(Optional) Displays a file on a website.
https:	(Optional) Displays a file on a secure website.
tftp:	(Optional) Displays a file on a TFTP server.

Command Default

ASCII mode.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **system support view-files** command is a better option for finding and viewing log files.

Examples

The following example shows how to display the contents of a local file named “test.cfg”:

```
> more test.cfg
: Saved
: Written by enable_15 at 10:04:01 Apr 14 2005
XXX Version X.X(X)
nameif vlan300 outside security10
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
```

```
ciscoasa test
fixup protocol ftp 21
fixup protocol h323 H225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
access-list deny-flow-max 4096
access-list alert-interval 300
access-list 100 extended permit icmp any any
access-list 100 extended permit ip any any
pager lines 24
icmp permit any outside
mtu outside 1500
ip address outside 172.29.145.35 255.255.0.0
no asdm history enable
arp timeout 14400
access-group 100 in interface outside
!
interface outside
!
route outside 0.0.0.0 0.0.0.0 172.29.145.1 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02 rpc 0:10:00 h3
23 0:05:00 h225 1:00:00 mgcp 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
snmp-server host outside 128.107.128.179
snmp-server location my_context, USA
snmp-server contact admin@example.com
snmp-server community public
no snmp-server enable traps
floodguard enable
fragment size 200 outside
no sysopt route dnat
telnet timeout 5
ssh timeout 5
terminal width 511
gdb enable
mgcp command-queue 0
Cryptochecksum:00000000000000000000000000000000
: end
```

Related Commands

Command	Description
cd	Changes to the specified directory.
pwd	Displays the current working directory.
system support view-files	Find and view the contents of log files.

nslookup (deprecated)

To look up the IP address for a fully-qualified domain name, or the reverse, use the **nslookup** command.

nslookup {*hostname* | *ip_address*}

Syntax Description

hostname	The fully-qualified domain name of a host whose IP address you are looking up. For example, www.example.com.
ip_address	The IP address of a host whose fully-qualified domain name you are looking up.

Command History

Release	Modification
6.1	This command was introduced.
6.6	This command no longer works and is deprecated.
7.1	This command was removed and replaced by **dig**.

Usage Guidelines

Some commands that allow fully-qualified domain names cannot use the DNS servers configured for the management interface to look up the IP address for the name. If you do not have DNS servers configured for commands that go through the data interfaces, use the **nslookup** command to determine the IP address, then use the IP address in the command.

The **nslookup** command is also useful in determining the fully-qualified domain name for a given IP address.

Examples

The following example looks up the IP address for www.cisco.com. The initial Server and Address information shows the DNS server (which could be a fully-qualified domain name), IP address, and port. (The addresses in this example are faked.) The following information shows the canonical (real) host name and IP address for the name you entered.

```
> nslookup www.cisco.com
Server:         10.102.6.247
Address:        10.102.6.247#53

www.cisco.com   canonical name = origin-www.cisco.com.
Name:   origin-www.cisco.com
Address: 173.37.145.84
```

The following example shows how to do a reverse lookup and determine a hostname for an IP address. The initial information is for the DNS server used. The mapped hostname is indicated by the **name =** field.

```
> nslookup 173.37.145.84
Server:         10.102.6.247
Address:        10.102.6.247#53

84.145.37.173.in-addr.arpa      name = www2.cisco.com.
```

packet-tracer

To enable packet-tracing capabilities for troubleshooting by specifying the 5-tuple to test firewall rules, use the **packet-tracer** command. (For clarity, the syntax is shown separately for ICMP, TCP/UDP, and IP packet modeling. You can replay multiple packets and trace a complete workflow using the **pcap** keyword.)

packet-tracer input *ifc_name* **icmp** { *sip* | **user** *username* } *type code* [*ident*] { *dip* | **fqdn** *fqdn-string* } [**detailed**] [**xml**]
packet-tracer input *ifc_name* { **tcp** | **udp** } { *sip* | **user** *username* } *sport* { *dip* | **fqdn** *fqdn-string* } *dport* [**detailed**] [**xml**]
packet-tracer input *ifc_name* **rawip** { *sip* | **user** *username* } *protocol* { *dip* | **fqdn** *fqdn-string* } [**detailed**] [**xml**]
packet-tracer input *ifc_name* **pcap** *pcap_filename* [**bypass-checks** | **decrypted** | **detailed** | **persist** | **transmit** | **xml** | **json** | **force**]

Syntax Description

bypass-checks	(Optional) Bypasses the security checks for simulated packets.
decrypted	(Optional) Considers simulated packet as IPsec/SSL VPN decrypted.
code	The ICMP code for an ICMP packet trace.
detailed	(Optional) Provides detailed trace results information.
dip	The destination IPv4 or IPv6 address for the packet trace.
dport	The destination port for a TCP/UDP/SCTP packet trace.
fqdn *fqdn-string*	The fully qualified domain name of the host. Supports the FQDN for IPv4 only.
force	Removes existing pcap trace and executes a new pcap file.
icmp	Specifies the protocol to use is ICMP.
ident	(Optional.) The ICMP identifier for an ICMP packet trace.
inline-tag *tag*	The security group tag value being embedded in the Layer 2 CMD header. Valid values range from 0 - 65533.
input *ifc_name*	The name of the source interface on which to trace the packets.
json	(Optional) Displays the trace results in JSON format.
pcap	Specifies pcap as input.
pcap_filename	The pcap filename that contain the packet for tracing.
protocol	The protocol number for raw IP packet tracing, 0-255.
persist	(Optional) Enables tracing for a long term and also tracing in cluster.
rawip	Specifies the protocol to use is raw IP.
sip	The source IPv4 or IPv6 address for the packet trace.

sport	The source port for a TCP/UDP/SCTP packet trace.
tcp	Specifies the protocol to use is TCP.
transmit	(Optional) Allows simulated packet to transmit from device.
type	The ICMP type for an ICMP packet trace.
udp	Specifies the protocol to use is UDP.
user *username*	The user identity in the format of domain\user if you want to specify the user as the source IP address. The most recently mapped address for the user (if any) is used in the trace.
xml	(Optional) Displays the trace results in XML format.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The output was enhanced to provide specific reasons for packet allow and drop while routing the packets.
7.1	The **packet-tracer** command is enhanced to allow a PCAP file as input for tracing.

Usage Guidelines

In addition to capturing packets, it is possible to trace the lifespan of a packet through the threat defense device to see if it is behaving as expected. The **packet-tracer** command enables you to do the following:

- Debug all packet drops in production network.
- Verify the configuration is working as intended.
- Show all rules applicable to a packet along with the CLI lines that caused the rule addition.
- Show a time line of packet changes in a data path.
- Inject tracer packets into the data path.

The **packet-tracer** command provides detailed information about the packets and how they are processed by the threat defense device. If a command from the configuration did not cause the packet to drop, the **packet-tracer** command provides information about the cause in an easily readable format. For example if a packet was dropped because of an invalid header validation, the following message appears: “packet dropped due to bad ip header (reason).”

While the **packet-tracer** injects and traces a single packet, the **pcap** keyword enables the packet-tracer to replay multiple packets (maximum of 100 packets) and to trace an entire flow. You can provide the pcap file as input and obtain the results in XML or JSON format for further analysis. To clear the trace output, use the **pcap trace** sub command of **clear packet-tracer**. You cannot use the trace output while the trace is in progress.

Examples

The following example shows how to run packet-tracer with a pcap file as input:

```
> packet-tracer input inside pcap http_get.pcap detailed xml
```

The following example shows how to run packet-tracer by clearing existing pcap trace buffer and giving a pcap file as input:

```
> packet-tracer input inside pcap http_get.pcap force
```

The following example traces a ICMP packet from the inside interface. The result indicates that the packet will be dropped for the reverse-path verification failure (RPF). The reason for the failure could be that the traffic entered the outside interface from an address that is known to the routing table, but is associated with the inside interface, then the device drops the packet. Similarly, if traffic enters the inside interface from an unknown source address, the device drops the packet because the matching route (the default route) indicates the outside interface.

```
> packet-tracer input inside icmp 10.15.200.2 8 0$

Phase: 1
Type: CAPTURE
Subtype:
Result: ALLOW
Config:
Additional Information:
 Forward Flow based lookup yields rule:
 in  id=0xd793b4a0, priority=12, domain=capture, deny=false
        hits=621531641, user_data=0xd7bbe720, cs_id=0x0, l3_type=0x0
        src mac=0000.0000.0000, mask=0000.0000.0000
        dst mac=0000.0000.0000, mask=0000.0000.0000

Phase: 2
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Config:
Implicit Rule
Additional Information:
 Forward Flow based lookup yields rule:
 in  id=0xd7dc31d8, priority=1, domain=permit, deny=false
        hits=23451445222, user_data=0x0, cs_id=0x0, l3_type=0x8
        src mac=0000.0000.0000, mask=0000.0000.0000
        dst mac=0000.0000.0000, mask=0100.0000.0000

Phase: 3
Type: ROUTE-LOOKUP
Subtype: input
Result: ALLOW
Config:
Additional Information:
in   10.15.216.0     255.255.252.0   inside

Phase: 4
Type: ROUTE-LOOKUP
Subtype: input
Result: ALLOW
Config:
Additional Information:
in   0.0.0.0         0.0.0.0         outside


Result:
input-interface: inside
input-status: up
input-line-status: up
output-interface: inside
output-status: up
```

```
output-line-status: up
Action: drop
Drop-reason: (rpf-violated) Reverse-path verify failed
```

The following example traces a TCP packet for the HTTP port from 10.100.10.10 to 10.100.11.11. The result indicates that the packet will be dropped by the implicit deny access rule:

```
> packet-tracer input outside tcp 10.100.10.10 80 10.100.11.11 80
Phase: 1
Type: ROUTE-LOOKUP
Subtype: Resolve Egress Interface
Result: ALLOW
Config:
Additional Information:
found next-hop 10.86.116.1 using egress ifc  outside
Phase: 2
Type: ACCESS-LIST
Subtype:
Result: DROP
Config:
Implicit Rule
Additional Information:
Result:
input-interface: outside
input-status: up
input-line-status: up
output-interface: NP Identity Ifc
output-status: up
output-line-status: up
Action: drop
Drop-reason: (acl-drop) Flow is denied by configured rule
```

The following example traces a TCP packet in a directly connected hosts having the ARP entry for nexthop:

```
firepower(config)# packet-tracer input inside tcp 192.168.100.100 12345 192.168.102.102 80
 detailed
Phase: 1
Type: ROUTE-LOOKUP
Subtype: No ECMP load balancing
Result: ALLOW
Config:
Additional Information:
Destination is locally connected. No ECMP load balancing.
Found next-hop 192.168.102.102 using egress ifc  outside(vrfid:0)

Phase: 2
Type: ACCESS-LIST
Subtype: log
Result: ALLOW
Config:
access-group TEST global
access-list TEST advanced trust ip any any
Additional Information:
Forward Flow based lookup yields rule:
in  id=0x2ae2a8aa5e90, priority=12, domain=permit, trust
hits=17, user_data=0x2ae29aabc100, cs_id=0x0, use_real_addr, flags=0x0, protocol=0
src ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, ifc=any
dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, ifc=any, vlan=0, dscp=0x0
input_ifc=any, output_ifc=any

Phase: 3
```

```
Type: NAT
Subtype: per-session
Result: ALLOW
Config:
Additional Information:
Forward Flow based lookup yields rule:
in  id=0x2ae2a69a7240, priority=0, domain=nat-per-session, deny=false
hits=34, user_data=0x0, cs_id=0x0, reverse, use_real_addr, flags=0x0, protocol=6
src ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any
dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, dscp=0x0
input_ifc=any, output_ifc=any

Phase: 4
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:
Forward Flow based lookup yields rule:
in  id=0x2ae2a8488800, priority=0, domain=inspect-ip-options, deny=true
hits=22, user_data=0x0, cs_id=0x0, reverse, flags=0x0, protocol=0
src ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any
dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, dscp=0x0
input_ifc=inside(vrfid:0), output_ifc=any

Phase: 5
Type: NAT
Subtype: per-session
Result: ALLOW
Config:
Additional Information:
Reverse Flow based lookup yields rule:
in  id=0x2ae2a69a7240, priority=0, domain=nat-per-session, deny=false
hits=36, user_data=0x0, cs_id=0x0, reverse, use_real_addr, flags=0x0, protocol=6
src ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any
dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, dscp=0x0
input_ifc=any, output_ifc=any

Phase: 6
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:
Reverse Flow based lookup yields rule:
in  id=0x2ae2a893e230, priority=0, domain=inspect-ip-options, deny=true
hits=10, user_data=0x0, cs_id=0x0, reverse, flags=0x0, protocol=0
src ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any
dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, dscp=0x0
input_ifc=outside(vrfid:0), output_ifc=any

Phase: 7
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 21, packet dispatched to next module
Module information for forward flow ...
snp_fp_inspect_ip_options
snp_fp_tcp_normalizer
snp_fp_translate
snp_fp_adjacency
snp_fp_fragment
```

```
snp_fp_tracer_drop
snp_ifc_stat

Module information for reverse flow ...
snp_fp_inspect_ip_options
snp_fp_translate
snp_fp_tcp_normalizer
snp_fp_adjacency
snp_fp_fragment
snp_fp_tracer_drop
snp_ifc_stat

Phase: 8
Type: INPUT-ROUTE-LOOKUP-FROM-OUTPUT-ROUTE-LOOKUP
Subtype: Resolve Preferred Egress interface
Result: ALLOW
Config:
Additional Information:
Found next-hop 192.168.102.102 using egress ifc  outside(vrfid:0)

Phase: 9
Type: ADJACENCY-LOOKUP
Subtype: Resolve Nexthop IP address to MAC
Result: ALLOW
Config:
Additional Information:
found adjacency entry for next-hop 192.168.102.102 on interface  outside
Adjacency :Active
mac address 0aaa.0bbb.00cc hits 5 reference 1

Result:
input-interface: inside(vrfid:0)
input-status: up
input-line-status: up
output-interface: outside(vrfid:0)
output-status: up
output-line-status: up
Action: allow
```

The following example traces a TCP packet that is dropped due to absence of a valid ARP entry for nexthop. Note that the drop reason provides the tip to check the ARP table.

```
<Displays same phases as in the previous example till Phase 8>

Result:
input-interface: inside(vrfid:0)
input-status: up
input-line-status: up
output-interface: outside(vrfid:0)
output-status: up
output-line-status: up
Action: drop
Drop-reason: (no-v4-adjacency) No valid V4 adjacency. Check ARP table (show arp) has entry
 for nexthop., Drop-location: frame snp_fp_adj_process_cb:200 flow (NA)/NA
```

The following example depicts packet tracer for sub-optimal routing with NAT and a reachable nexthop:

```
firepower(config)# sh run route
route inside 0.0.0.0 0.0.0.0 192.168.100.100 1
route outside 0.0.0.0 0.0.0.0 192.168.102.102 10

firepower(config)# sh nat detail
Manual NAT Policies (Section 1)
1 (outside) to (dmz) source static src_real src_mapped  destination static dest_real
```

```
dest_mapped
translate_hits = 3, untranslate_hits = 3
Source - Origin: 9.9.9.0/24, Translated: 10.10.10.0/24
Destination - Origin: 192.168.104.0/24, Translated: 192.168.104.0/24
firepower(config)# packet-tracer input dmz tcp 192.168.104.104 12345 10.10.10.10 80 detailed


Phase: 1
Type: UN-NAT
Subtype: static
Result: ALLOW
Config:
nat (outside,dmz) source static src_real src_mapped destination static dest_real dest_mapped
Additional Information:
NAT divert to egress interface outside(vrfid:0)
Untranslate 10.10.10.10/80 to 9.9.9.10/80

Phase: 2
Type: ACCESS-LIST
Subtype: log
Result: ALLOW
Config:
access-group TEST global
access-list TEST advanced trust ip any any
Additional Information:
Forward Flow based lookup yields rule:
in  id=0x2ae2a8aa5e90, priority=12, domain=permit, trust
hits=20, user_data=0x2ae29aabc100, cs_id=0x0, use_real_addr, flags=0x0, protocol=0
src ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, ifc=any
dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, ifc=any, vlan=0, dscp=0x0
input_ifc=any, output_ifc=any

Phase: 3
Type: NAT
Subtype:
Result: ALLOW
Config:
nat (outside,dmz) source static src_real src_mapped destination static dest_real dest_mapped
Additional Information:
Static translate 192.168.104.104/12345 to 192.168.104.104/12345
Forward Flow based lookup yields rule:
in  id=0x2ae2a8aa4ff0, priority=6, domain=nat, deny=false
hits=4, user_data=0x2ae2a8a9d690, cs_id=0x0, flags=0x0, protocol=0
src ip/id=192.168.104.0, mask=255.255.255.0, port=0, tag=any
dst ip/id=10.10.10.0, mask=255.255.255.0, port=0, tag=any, dscp=0x0
input_ifc=dmz(vrfid:0), output_ifc=outside(vrfid:0)

Phase: 4
Type: NAT
Subtype: per-session
Result: ALLOW
Config:
Additional Information:
Forward Flow based lookup yields rule:
in  id=0x2ae2a69a7240, priority=0, domain=nat-per-session, deny=false
hits=40, user_data=0x0, cs_id=0x0, reverse, use_real_addr, flags=0x0, protocol=6
src ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any
dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, dscp=0x0
input_ifc=any, output_ifc=any

Phase: 5
Type: IP-OPTIONS
Subtype:
Result: ALLOW
```

```
Config:
Additional Information:
Forward Flow based lookup yields rule:
in  id=0x2ae2a89de1b0, priority=0, domain=inspect-ip-options, deny=true
hits=4, user_data=0x0, cs_id=0x0, reverse, flags=0x0, protocol=0
src ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any
dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, dscp=0x0
input_ifc=dmz(vrfid:0), output_ifc=any

Phase: 6
Type: NAT
Subtype: rpf-check
Result: ALLOW
Config:
nat (outside,dmz) source static src_real src_mapped destination static dest_real dest_mapped
Additional Information:
Forward Flow based lookup yields rule:
out id=0x2ae2a8aa53d0, priority=6, domain=nat-reverse, deny=false
hits=5, user_data=0x2ae2a8a9d580, cs_id=0x0, use_real_addr, flags=0x0, protocol=0
src ip/id=192.168.104.0, mask=255.255.255.0, port=0, tag=any
dst ip/id=9.9.9.0, mask=255.255.255.0, port=0, tag=any, dscp=0x0
input_ifc=dmz(vrfid:0), output_ifc=outside(vrfid:0)

Phase: 7
Type: NAT
Subtype: per-session
Result: ALLOW
Config:
Additional Information:
Reverse Flow based lookup yields rule:
in  id=0x2ae2a69a7240, priority=0, domain=nat-per-session, deny=false
hits=42, user_data=0x0, cs_id=0x0, reverse, use_real_addr, flags=0x0, protocol=6
src ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any
dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, dscp=0x0
input_ifc=any, output_ifc=any

Phase: 8
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:
Reverse Flow based lookup yields rule:
in  id=0x2ae2a893e230, priority=0, domain=inspect-ip-options, deny=true
hits=13, user_data=0x0, cs_id=0x0, reverse, flags=0x0, protocol=0
src ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any
dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, tag=any, dscp=0x0
input_ifc=outside(vrfid:0), output_ifc=any

Phase: 9
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 24, packet dispatched to next module
Module information for forward flow ...
snp_fp_inspect_ip_options
snp_fp_tcp_normalizer
snp_fp_translate
snp_fp_adjacency
snp_fp_fragment
snp_fp_tracer_drop
snp_ifc_stat
```

```
Module information for reverse flow ...
snp_fp_inspect_ip_options
snp_fp_translate
snp_fp_tcp_normalizer
snp_fp_adjacency
snp_fp_fragment
snp_fp_tracer_drop
snp_ifc_stat

Phase: 10
Type: INPUT-ROUTE-LOOKUP-FROM-OUTPUT-ROUTE-LOOKUP
Subtype: Resolve Preferred Egress interface
Result: ALLOW
Config:
Additional Information:
Found next-hop 192.168.100.100 using egress ifc  inside(vrfid:0)

Phase: 11
Type: SUBOPTIMAL-LOOKUP
Subtype: suboptimal next-hop
Result: ALLOW
Config:
Additional Information:
Input route lookup returned ifc  inside is not same as existing ifc  outside
Doing adjacency lookup lookup on existing ifc outside

Phase: 12
Type: NEXTHOP-LOOKUP-FROM-OUTPUT-ROUTE-LOOKUP
Subtype: Lookup Nexthop on interface
Result: ALLOW
Config:
Additional Information:
Found next-hop 192.168.102.102 using egress ifc  outside(vrfid:0)

Phase: 13
Type: ADJACENCY-LOOKUP
Subtype: Resolve Nexthop IP address to MAC
Result: ALLOW
Config:
Additional Information:
found adjacency entry for Next-hop 192.168.102.102 on interface  outside
Adjacency :Active
mac address 0aaa.0bbb.00cc hits 5 reference 1

Result:
input-interface: dmz(vrfid:0)
input-status: up
input-line-status: up
output-interface: outside(vrfid:0)
output-status: up
output-line-status: up
Action: allow
```

The following example depicts packet tracer for sub-optimal routing with NAT, where, the packet is dropped due to non-reachable nexthop:

```
firepower(config)# sh run route
route inside 0.0.0.0 0.0.0.0 192.168.100.100 1

firepower(config)# sh nat detail
Manual NAT Policies (Section 1)
1 (outside) to (dmz) source static src_real src_mapped  destination static dest_real
dest_mapped
```

```
translate_hits = 3, untranslate_hits = 3
Source - Origin: 9.9.9.0/24, Translated: 10.10.10.0/24
Destination - Origin: 192.168.104.0/24, Translated: 192.168.104.0/24


<Displays same phases as in the previous example till Phase 11>

Result:
input-interface: dmz(vrfid:0)
input-status: up
input-line-status: up
output-interface: outside(vrfid:0)
output-status: up
output-line-status: up
Action: drop
Drop-reason: (no-adjacency) No valid adjacency, Drop-location: frame
snp_fp_adjacency_internal:5890 flow (NA)/NA
```

Related Commands

Command	Description
capture	Captures packet information, including trace packets.
show capture	Displays the capture configuration when no options are specified.
show packet-tracer	Displays the trace buffer output of the most recently run packet-tracer on a PCAP file.

perfmon

To display performance information at the console, use the **perfmon** command.

perfmon {**verbose** | **interval***seconds* | **settings**}

Syntax Description

verbose	Displays performance monitor information at the console. The default is to not display information, which is shown as "quiet" in the perfmon settings. You must be in the Diagnostic CLI to turn off **perfmon verbose**.
interval *seconds*	Specifies the number of seconds before the performance display is refreshed on the console.
settings	Displays the interval and whether perfmon is quiet or verbose.

Command Default

The default interval is 120 seconds.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **perfmon** command allows you to monitor the performance of the device. Use the **show perfmon** command to display the information immediately.

Use the **perfmon verbose** command to display the information on the console each interval.

The information appears automatically only if you are actually connected to the CLI on the Console port, or if you are in the Diagnostic CLI (**system support diagnostic-cli**). If you are in the CLI on a different port, including the management interface, use the **show console-output** command to see the automatically-generated information. Alternatively, do not use this command, and simply use the **show perfmon** command directly.

We recommend you use this command in the Diagnostic CLI only.

Note You cannot turn off **verbose** from the regular CLI. Instead, you must turn it off from Privileged EXEC mode in the Diagnostic CLI. See the examples section.

Examples

This example shows how to display the performance monitor statistics every 120 seconds on the console. In the output, the "Fixup" statistics refer to the related protocol inspection engine.

```
> perfmon verbose
> perfmon settings
interval: 120 (seconds)
verbose
> show console-output
...
Message #109 :
```

```
Message #110 : PERFMON STATS:                     Current      Average
Message #111 : Xlates                                0/s          0/s
Message #112 : Connections                           0/s          0/s
Message #113 : TCP Conns                             0/s          0/s
Message #114 : UDP Conns                             0/s          0/s
Message #115 : URL Access                            0/s          0/s
Message #116 : URL Server Req                        0/s          0/s
Message #117 : TCP Fixup                             0/s          0/s
Message #118 : TCP Intercept Established Conns       0/s          0/s
Message #119 : TCP Intercept Attempts                0/s          0/s
Message #120 : TCP Embryonic Conns Timeout           0/s          0/s
Message #121 : FTP Fixup                             0/s          0/s
Message #122 : AAA Authen                            0/s          0/s
Message #123 : AAA Author                            0/s          0/s
Message #124 : AAA Account                           0/s          0/s
Message #125 : HTTP Fixup                            0/s          0/s
Message #126 :
...
```

The following example shows how to turn off verbose mode. You must do so from the Diagnostic CLI.

```
> system support diagnostic-cli
Attaching to Diagnostic CLI ... Press 'Ctrl+a then d' to detach.
Type help or '?' for a list of available commands.

firepower> enable
Password: <Press return, do not enter a password>

firepower# perfmon quiet
firepower# perfmon settings
interval: 120 (seconds)
quiet
firepower# <Press Ctrl+a, d>

Console connection detached.
> perfmon settings
interval: 120 (seconds)
quiet
```

Related Commands

Command	Description
show perfmon	Displays performance information.

pigtail commands

Only use **pigtail** commands under the direction of the Cisco Technical Assistance Center.

If you want to view logs as they are written, use the **tail-logs** command instead of **pigtail**.

Caution Do not leave the pigtail process running as it can cause high disk usage. This process may also interfere with policy deployment if it is running during deployment. For information on how to stop the pigtail process, contact the Cisco Technical Assistance Center.

ping

To test connectivity from a specified interface to an IP address, use the **ping** command. The parameters available differ for regular ICMP-based ping, TCP ping, and a "system" ping. Also, system pings are from the management interface, whereas the other types of ping go through the data interfaces. Be sure to use the correct type of ping for your tests.

ping [**interface** *if_name* | **vrf** *name*] *host* [**repeat** *count*] [**timeout** *seconds*] [**data** *pattern*] [**size** *bytes*] [**validate**]
ping tcp [**interface** *if_name* | **vrf** *name*] *host port* [**repeat** *count*] [**timeout** *seconds*] [**source** *host port*]
ping system *host*

Syntax Description

data *pattern*	(Optional, ICMP only.) Specifies the 16-bit data pattern in hexadecimal format, from 0 to FFFF. The default is 0xabcd.
host	Specifies the IPv4 address or name of the host to ping. For ICMP pings, you can also specify an IPv6 address. IPv6 is not supported for TCP or system pings. Whether a ping can use a fully-qualified domain name, such as www.example.com, depends on the availability of a DNS server to resolve the name. The system pings use the DNS servers for the management interface, but other types of ping do not use the management DNS servers. You must configure DNS for the data interfaces for non-system hostname pings to work. If **ping** cannot resolve a hostname, use **nslookup** to determine the IP address associated with the name, and then ping the IP address.
interface *if_name*	(Optional) For ICMP, this is the name of the interface through which the host is accessible. If not supplied, then the host is resolved to an IP address and the routing table is consulted to determine the destination interface. For TCP, this is the input interface through which the source sends SYN packets. If you specify the **interface** keyword when virtual routing and forwarding (VRF) is enabled, the ping uses the virtual routing table for the specified interface.
port	(TCP only.) Specifies the TCP port number for the host you are pinging, 1-65535.
repeat *count*	(Optional) Specifies the number of times to repeat the ping request. The default is 5.
size *bytes*	(Optional, ICMP only.) Specifies the datagram size in bytes. The default is 100.
source *host port*	(Optional, TCP only.) Specifies a certain IP address and port to send the ping from (Use port = 0 for a random port).
system	Ping the host through the management interface. Unlike pings through the data interfaces, there is no default count for system pings. The ping continues until you stop it using Ctrl+c.

tcp	(Optional) Tests a connection over TCP (the default is ICMP). A TCP ping sends SYN packets and considers the ping successful if the destination sends a SYN-ACK packet. You can also have at most 2 concurrent TCP pings running at a time.
timeout *seconds*	(Optional) Specifies the number of seconds of the timeout interval. The default is 2 seconds.
validate	(Optional, ICMP only.) Validates reply data.
vrf *name*	(Optional.) If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can choose which virtual routing table to use by specifying the name of the virtual router. This keyword is exclusive with the **interface** keyword. If you specify the **interface** keyword when virtual routing and forwarding (VRF) is enabled, the ping uses the virtual routing table for the specified interface.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The **vrf** keyword was added.

Usage Guidelines

The **ping** command allows you to determine if the device has connectivity or if a host is available on the network.

When using regular ICMP-based ping, ensure that you do not have ICMP rules that prohibit these packets (if you do not use ICMP rules, all ICMP traffic is allowed).

When using TCP ping, you must ensure that access policies allow TCP traffic on the ports you specify.

This configuration is required to allow the device to respond and accept messages generated from the **ping** command. The **ping** command output shows if the response was received. If a host is not responding after you enter the **ping** command, a message similar to the following appears:

```
> ping 10.1.1.1
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
?????
Success rate is 0 percent (0/5)
```

Use the **show interface** command to ensure that the device is connected to the network and is passing traffic. The address of the specified interface name is used as the source address of the ping.

Examples

The following example shows how to determine if an IP address is accessible through a data interface. Because no interface is specified, the routing table is used to determine how to get to the address.

```
> ping 171.69.38.1
Sending 5, 100-byte ICMP Echos to 171.69.38.1, timeout is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
```

The following examples use TCP ping to determine if a host is accessible through a data interface.

```
> ping tcp 10.0.0.1 21
Type escape sequence to abort.
No source specified. Pinging from identity interface.
Sending 5 TCP SYN requests to 10.0.0.1 port 21
from 10.0.0.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

> ping tcp 10.0.0.1 21 source 192.168.1.1 2002 repeat 10
Type escape sequence to abort.
Sending 10 TCP SYN requests to 10.0.0.1 port 21
from 192.168.1.1 starting port 2002, timeout is 2 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 1/2/2 ms
```

The following example does a system ping to determine if www.cisco.com is accessible through the management interface. You must use Ctrl+c to stop the ping (indicated by ^C in the output).

```
> ping system www.cisco.com
PING origin-www.cisco.COM (72.163.4.161) 56(84) bytes of data.
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=1 ttl=242 time=10.6 ms
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=2 ttl=242 time=8.13 ms
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=3 ttl=242 time=8.51 ms
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=4 ttl=242 time=8.40 ms
^C
--- origin-www.cisco.COM ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 8.139/8.927/10.650/1.003 ms
>
```

The following example pings an address using the routing table of the virtual router named red.

```
> ping vrf red 2002::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2002::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/20 ms
```

Related Commands

Command	Description
nslookup	Perform a DNS lookup for a hostname or IP address.
show interface	Displays information about the interface configuration.

pmtool commands

Only use **pmtool** commands under the direction of the Cisco Technical Assistance Center.

reboot

To reboot the device, use the **reboot** command.

reboot

Command History

Release	Modification
6.1	This command was introduced.

Examples

```
> reboot
This command will reboot the system. Continue?
Please enter 'YES' or 'NO': yes

Broadcast message from root@firepower

The system is going down for reboot NOW!
...
```

redundant-interface

To set which member interface of a redundant interface is active, use the **redundant-interface** command.

redundant-interface redundant *number* **active-member** *physical_interface*

Syntax Description

active-member *physical_interface*	Sets the active member. Use the show interface command to see available physical interface names, such as GigabitEthernet0/0. Both member interfaces must be the same physical type.
redundant *number*	Specifies the redundant interface ID, such as **redundant 1**. Numbers are 1-8.

Command Default

By default, the active interface is the first member interface listed in the configuration, if it is available.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Create redundant interfaces in the device manager. When you create the redundant interface, you specify which is primary. Use this command to change which interface is active during run time.

To view which interface is active, enter the following command:

show interface redundant*number* **detail** | **grep Member**

For example:

```
> show interface redundant1 detail | grep Member
      Members GigabitEthernet0/3(Active), GigabitEthernet0/2
```

Examples

The following example changes the active interface for the redundant1 interface.

```
> show interface redundant1 detail | grep Member
      Members GigabitEthernet0/3(Active), GigabitEthernet0/2

> redundant-interface redundant 1 active-member gigabitethernet0/2
```

Related Commands

Command	Description
clear interface	Clears counters for the **show interface** command.
show interface	Displays the runtime status and statistics of interfaces.

restore

To restore configuration backed up locally from a Secure Firewall Threat Defense device being managed by a Secure Firewall Management Center, use the **restore** command. To restore a backup saved to a remote location, specify additional parameters for location of the backup file and username.

restore remote-manager-backup [*backup tar-file* | **location** [*scp-hostname username filepath backup tar-file*]]

Syntax Description

remote-manager-backup *backup tar-file*	Restore a local backup created by the Secure Firewall Management Center. The local backup file is saved on the Secure Firewall Threat Defense device.
remote-manager-backup location *scp-hostname username filepath backup tar-file*	Restore a remote backup created by the Secure Firewall Management Center. The remote backup is saved at a user-configured location, accessible by an SCP server and identified by the hostname, username and file path.

Command History

Release	Modification
6.3	This command was introduced.

Usage Guidelines

The **restore** command restores the Secure Firewall Threat Defense system files, Snort DB tables and the LINA running configuration on the new/ replacement Secure Firewall Threat Defense. The **restore** command also ensures that the existing LINA running configuration on the Secure Firewall Threat Defense device is deleted before the actual restore operation proceeds. This ensures that the Secure Firewall Threat Defense device carries only the configurations present at the time the backup was taken. All device configurations except the serial number of the replacement device, will be replaced after a successful restore operation.

The restore operation ensures that the connection between the replacement /new Secure Firewall Threat Defense device and the original Secure Firewall Management Center is re-established using the universally unique identifier (UUID), assigned to the original device. After successful restore, the Secure Firewall Management Center marks all the policies of the device as out-of-date so that any configuration changes on the Secure Firewall Management Center that may affect the replacement Secure Firewall Threat Defense are deployed to it, when the device replacement procedure is complete. This ensures that the new Secure Firewall Threat Defense and Secure Firewall Management Centerconfigurations are in sync.

Examples

The following example shows a restore operation from a local backup file:

```
> restore remote-manager-backup 10.10.1.168_PRIMARY_20180614055906.tar
```

The following example shows a restore operation from a remote backup file:

```
>restore remote-manager-backup location 10.106.140.100 admin /Volume/home/admin
10.10.1.168_PRIMARY_20180614055906.tar
```