t - z

tail-logs

To open a system log to view messages as they are written when working with the Cisco Technical Assistance Center (TAC) to resolve a problem, use the **tail-logs** command.

tail-logs

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **tail-logs** command opens a system log so that you can see messages as they are written. Use this command while working with the Cisco Technical Assistance Center (TAC) so that they can help you interpret the output and to select the appropriate log to view.

The command presents a menu listing all available logs. Follow the command prompts to select the log. If the log is long, you will see a More line; press Enter to progress a line at a time, Space to go a page at a time. Press Ctrl+C to return to the command prompt when you are finished viewing the log.

Examples

The following example shows how tail the ngfw.log file. The file listing starts with directories at the top, then a list of files in the current directory.

```
> tail-logs
===Tail Logs===
============================
Directory: /ngfw/var/log
----------sub-dirs----------
cisco
mojo
removed_packages
setup
connector
sf
scripts
packages
removed_scripts
httpd
-----------files------------
2016-10-14 18:12:04.514783 | 5371       | SMART_STATUS_sda.log
2016-10-14 18:12:04.524783 | 353        | SMART_STATUS_sdb.log
2016-10-11 21:32:23.848733 | 326517     | action_queue.log
2016-10-06 16:00:56.620019 | 1018       | br1.down.log

<list abbreviated>

2016-10-06 15:38:22.630001 | 9194       | ngfw.log

<list abbreviated>

([b] to go back or [s] to select a file to view, [Ctrl+C] to exit)
Type a sub-dir name to list its contents: s

Type the name of the file to view ([b] to go back, [Ctrl+C] to exit)
```

```
> ngfw.log
2016-10-06 15:38:22 Running [rm -rf /etc/logrotate-dmesg.conf /etc/logrotate.conf
/etc/logrotate.d
/etc/logrotate_ssp.conf /etc/logrotate_ssp.d] ... success
2016-10-06 15:38:22 Running [ln -sf /ngfw/etc/logrotate-size.conf /etc/] ... success
2016-10-06 15:38:22 Running [ln -sf /ngfw/etc/logrotate-size.d /etc/] ... success
2016-10-06 15:38:22 Running [ln -sf /ngfw/etc/logrotate.conf /etc/] ... success
2016-10-06 15:38:22 Running [ln -sf /ngfw/etc/logrotate.d /etc/] ... success
2016-10-06 15:38:22 Running [rm -f /usr/sbin/ntpd] ... success
```

Related Commands

Command	Description
system support view-files	Opens a log file.

test aaa-server

To check whether the device can authenticate or authorize users with a particular AAA server, use the **test aaa-server** command.

test aaa-server {**authentication** *groupname* [**host** *ip_address*] [**username** *username*] [**password** *password*] | **authorization** *groupname* [**host** *ip_address*] [**username** *username*}

Syntax Description

groupname	Specifies the AAA server group or realm name.
host *ip-address*	Specifies the server IP address. If you do not specify the IP address in the command, you are prompted for it.
password *password*	Specifies the user password. If you do not specify the password in the command, you are prompted for it.
username *username*	Specifies the username of the account used to test the AAA server settings. Make sure the username exists on the AAA server; otherwise, the test will fail. If you do not specify the username in the command, you are prompted for it.

Command History

Release	Modification
6.2.1	This command was introduced.

Usage Guidelines

This command lets you verify that the system can authenticate or authorize users with a particular AAA server. This command lets you test the AAA server without having an actual user attempt to authenticate. It also helps you isolate whether AAA failures are due to misconfiguration of AAA server parameters, a connection problem to the AAA server, or other configuration errors.

Examples

The following is an example of a successful authentication:

```
> test aaa-server authentication svrgrp1 host 192.168.3.4 username bogus password mypassword
INFO: Attempting Authentication test to IP address <10.77.152.85> (timeout: 12 seconds)
INFO: Authentication Successful
```

The following is an unsuccessful authentication attempt:

```
> test aaa-server authentication svrgrp1
Server IP Address or name: 192.168.3.4
Username: bogus
Password: mypassword
INFO: Attempting Authentication test to IP address <192.168.3.4> (timeout: 10
          seconds)
ERROR: Authentication Rejected: Unspecified
```

Related Commands

Commands	Description
aaa-server active **aaa-server fail**	Reactivate a AAA server that is marked failed or fail an active AAA server.
clear aaa-server statistics	Clears AAA server statistics.
show aaa-server	Displays AAA server statistics.

traceroute

To determine the route packets will take to their destination through data interfaces, use the **traceroute** command. To determine the route packets will take to their destination when going through the management IP address, use the **traceroute system** command.

traceroute *destination* [**source** {*source_ip* | *source-interface*}] [**numeric**] [**timeout** *timeout_value*] [**probe** *probe_num*] [**ttl** *min_ttl max_ttl*] [**port** *port_value*] [**use-icmp**]
traceroute system *destination*

Syntax Description

destination	The IPv4 or IPv6 address, or hostname, of the host to which the route is to be traced. For example, 10.100.10.10 or www.example.com. You must configure a DNS server to resolve a hostname. Traces that use the **system** keyword use the DNS servers configured for the management interface. Other traces use the DNS servers configured for the data interfaces. If you do not have DNS defined for the data interfaces, first use the **nslookup** command to determine the host's IP address, and then use the IP address instead of the FQDN.
numeric	Specifies the output print only the IP addresses of the intermediate gateways. If this keyword is not specified the traceroute attempts to look up the hostnames of the gateways reached during the trace.
port *port_value*	The destination port used by the User Datagram Protocol (UDP) probe messages. The default is 33434.
probe *probe_num*	The number of probes to be sent at each TTL level. The default count is 3.
source {*source_ip* | *source_interface*}	Specifies an IP address or interface to be used as the source for the trace packets. This IP address must be the IP address of one of the data interfaces. In transparent mode, it must be the management IP address. If you specify an interface name, the IP address of the interface is used.
system	Indicates the traceroute should be through the management interface, not a data interface.
timeout *timeout_value*	Specifies the amount of time in seconds to wait for a response before the connection times out. The default is three seconds.
ttl *min_ttl max_ttl*	Specifies the range of Time To Live values to use in the probes. • *min_ttl*—The TTL value for the first probes. The default is 1, but it can be set to a higher value to suppress the display of known hops. • *max-ttl*—The largest TTL value that can be used. The default is 30. The command terminates when the traceroute packet reaches the destination or when the value is reached.
use-icmp	Specifies the use of ICMP probe packets instead of UDP probe packets.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **traceroute** command prints the result of each probe sent. Every line of output corresponds to a TTL value in increasing order. The following are the output symbols printed by the **traceroute** command:

Output Symbol	Description
*	No response was received for the probe within the timeout period.
nn msec	For each node, the round-trip time (in milliseconds) for the specified number of probes.
!N.	ICMP network unreachable.
!H	ICMP host unreachable.
!P	ICMP protocol unreachable.
!A	ICMP administratively prohibited.
?	Unknown ICMP error.

Examples

The following example shows traceroute output that results when a destination IP address has been specified:

```
> traceroute 209.165.200.225
Tracing the route to 209.165.200.225
 1  10.83.194.1 0 msec 10 msec 0 msec
 2  10.83.193.65 0 msec 0 msec 0 msec
 3  10.88.193.101 0 msec 10 msec 0 msec
 4  10.88.193.97 0 msec 0 msec 10 msec
 5  10.88.239.9 0 msec 10 msec 0 msec
 6  10.88.238.65 10 msec 10 msec 0 msec
 7 172.16.7.221 70 msec 70 msec 80 msec
 8 209.165.200.225 70 msec 70 msec 70 msec
```

The following example shows a traceroute through the management interface to a hostname.

```
> traceroute system www.example.com
traceroute to www.example.com (172.163.4.161), 30 hops max, 60 byte packets
 1  192.168.0.254 (192.168.0.254)  0.213 ms  0.310 ms  0.328 ms
 2  10.88.127.1 (10.88.127.1)  0.677 ms  0.739 ms  0.899 ms
 3  lab-gw1.example.com (10.89.128.25)  0.638 ms  0.856 ms  0.864 ms
 4  04-bb-gw1.example.com (10.152.240.65)  1.169 ms  1.355 ms  1.409 ms
 5  wan-gw1.example.com (10.152.240.33)  0.712 ms  0.722 ms  0.790 ms
 6  wag-gw1.example.com (10.152.240.73)  13.868 ms  10.760 ms  11.187 ms
 7  rbb-gw2.example.com (172.30.4.85)  7.202 ms  7.301 ms  7.101 ms
 8  rbb-gw1.example.com (172.30.4.77)  8.162 ms  8.225 ms  8.373 ms
 9  sbb-gw1.example.com (172.16.16.210)  7.396 ms  7.548 ms  7.653 ms
10  corp-gw2.example.com (172.16.16.58)  7.413 ms  7.310 ms  7.431 ms
11  dmzbb-gw2.example.com (172.16.0.78)  7.835 ms  7.705 ms  7.702 ms
```

```
12  dmzdcc-gw2.example.com (172.16.0.190)  8.126 ms  8.193 ms  11.559 ms
13  dcz05n-gw1.example.com (172.16.2.106)  11.729 ms  11.728 ms  11.939 ms
14  www1.example.com (172.16.4.161)  11.645 ms  7.958 ms  7.936 ms
```

Related Commands

Command	Description
capture	Captures packet information, including trace packets.
show capture	Displays the capture configuration when no options are specified.
packet-tracer	Enables packet tracing capabilities.

undebug

To disable debugging for a given feature, use the **undebug** command. This command is a synonym for the **no debug** command.

undebug {*feature* [*subfeature*] [*level*] | **all**}

Syntax Description

all	Disables debugging for all features.
feature	Specifies the feature for which you want to disable debugging. To see available features, use the **undebug ?** command for CLI help.
subfeature	(Optional) Depending on the feature, you can disable 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 History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with 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 disables debugging for all enabled debugs.

```
> undebug all
>
```

Related Commands

Command	Description
debug	Enables debugging for a feature.
show debug	Shows the currently active debug settings.

upgrade

To retry, cancel, or revert a system software upgrade, use the **upgrade** command. Note that this command is supported only for major and maintence upgrades.

upgrade { **cancel** | **cleanup-revert** | **revert** | **retry** }

Syntax Description

cancel	Cancel a failed or in-progress upgrade. If an upgrade fails, but the system believes it is still in progress, you must cancel it to change the job status to one where you can retry the upgrade. The system should be able to automatically cancel failed upgrades in most cases.
cleanup-revert	Permanently remove the revert snapshot to free up disk space. If you clean up the revertible version, you cannot use the **revert** keyword to return to it.
revert	Undo a system software upgrade by returning to the previous version, if a revertible one is available. First use the **show upgrade revert-info** command to verify there is a revertible version, and which version it is. If that version is acceptable, you can use this command to revert to that version. In high availability/scalability deployments, revert is more successful when all units are reverted simultaneously. When reverting with the CLI, open sessions with all units, verify that revert is possible on each, then start the processes at the same time. After you revert, you must re-register the device with the Smart Software Manager. In Versions 6.7 through 7.1, **upgrade revert** is available for a locally managed system only. You cannot use this command on a system managed by management center. In Version 7.2+, this command is supported in management center deployments *if* communications between the management center and device are disrupted. **Caution** Reverting from the CLI can cause configurations between the device and the management center to go out of sync, depending on what you changed post-upgrade. This can cause further communication and deployment issues.
retry	Retry a failed upgrade. The upgrade must be considered failed by the system, and not in progress. You might need to enter **upgrade cancel** before you can retry the upgrade.

Command History

Release	Modification
6.7	This command was introduced.
7.0	The **upgrade revert** command now automatically unregisters the device from the Smart Software Manager. You must re-register the device after reverting an upgrade.

Release	Modification
7.2	The **upgrade revert** command is now supported in management center deployments if communications between the management center and device are disrupted.

Examples

The following example shows how to cancel a system software update that is in progress. After an upgrade cancel completes successfully, the device will be rebooted automatically.

```
> upgrade cancel
Warning: Upgrade in progress (11%,  8 mins remaining).
Are you sure you want to cancel it(yes/no)? yes
```

The following example shows how to retry a failed upgrade. You need to first correct the issues that made the upgrade fail, as indicated by failure messages. You might need to use **upgrade cancel** before you can retry the upgrade. Not all failed upgrades can be retried.

```
> upgrade retry
Tue Dec  3 23:50:31 UTC 2020: Resuming upgrade for
Cisco_FTD_Upgrade-6.7.0-32.sh.REL.tar
```

The following example shows how to revert to the previous version on a locally-managed system. Use the **show upgrade revert-info** command to determine if there is a version available for reversion.

```
> upgrade revert
Current version is 6.7.0.50
Detected previous version 6.6.1.20
Are you sure you want to revert (Yes/No)? Yes
```

The following example shows how to remove the previous version to clear up disk space. After using this command, you will not be able to revert to the previous version.

```
> upgrade cleanup-revert
Version 6.6 was cleaned up successfully.
```

Related Commands

Command	Description
show last-upgrade status	Shows information on the last system software upgrade.
show upgrade	Shows information on the current system software upgrade.

verify

To verify the checksum of a file, use the **verify** command.

verify [**sha-512** | **/signature**] *path*
verify/md5 *path* [*md5-value*]

Syntax Description

/md5	(Optional) Calculates and displays the MD5 value for the specified software image. Compare this value with the value available on Cisco.com for this image.
sha-512	(Optional) Calculates and displays the SHA-512 value for the specified software image. Compare this value with the value available on Cisco.com for this image.
/signature	(Optional) Verifies the signature of an image stored in flash.
md5-value	(Optional) The known MD5 value for the specified image. When an MD5 value is specified in the command, the system will calculate the MD5 value for the specified image and display a message verifying that the MD5 values match or that there is a mismatch.

path	• *filename* The name of a file in the current directory. Use **dir** to see directory contents, **cd** to change directories. • **disk0:**/[*path*/]*filename* This option indicates the internal Flash memory. You can also use **flash:** instead of **disk0**; they are aliased. • **disk1:**/[*path*/]*filename* This option indicates the external Flash memory card. • **flash:**/[*path*/]*filename* This option indicates the internal Flash card. For the ASA 5500 series, **flash** is an alias for **disk0:**. • **ftp:**//[*user*[**:***password*]**@**]*server*[: *port*]/[*path*/]*filename*[**;type=***xx*] The **type** can be one of the following keywords: • **ap**—ASCII passive mode • **an**—ASCII normal mode • **ip**—(Default) Binary passive mode • **in**—Binary normal mode • **http**[**s**]**:**//[*user*[:*password*] **@**]*server*[: *port*]/[*path*/]*filename* • **tftp:**//[*user*[:*password*]@]*server*[: *port*]/[*path*/]*filename*[**;int=***interface_name*] Specify the interface name if you want to override the route to the server address. The pathname cannot contain spaces.

Command Default

The current flash device is the default file system.

Note When you specify the **/md5** option, you can use a network file, such as ftp, http and tftp as the source. The **verify** command without the **/md5** option only lets you verify local images in Flash.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Use the **verify** command to verify the checksum of a file before using it.

Each software image that is distributed on disk uses a single checksum for the entire image. This checksum is displayed only when the image is copied into Flash memory; it is not displayed when the image file is copied from one disk to another.

Before loading or duplicating a new image, record the checksum and MD5 information for the image so that you can verify the checksum when you copy the image into Flash memory or onto a server. A variety of image information is available on Cisco.com.

To display the contents of Flash memory, use the **show flash:** command. The Flash contents listing does not include the checksum of individual files. To recompute and verify the image checksum after the image has been copied into Flash memory, use the **verify** command. Note, however, that the **verify** command only performs a check on the integrity of the file after it has been saved in the file system. It is possible for a corrupt image to be transferred to the device and saved in the file system without detection. If a corrupt image is transferred successfully to the device, the software will be unable to tell that the image is corrupted and the file will verify successfully.

To use the message-digest5 (MD5) hash algorithm to ensure file validation, use the **verify** command with the **/md5** option. MD5 is an algorithm (defined in RFC 1321) that is used to verify data integrity through the creation of a unique 128-bit message digest. The **/md5** option of the **verify** command allows you to check the integrity of the security appliance software image by comparing its MD5 checksum value against a known MD5 checksum value for the image. MD5 values are now made available on Cisco.com for all security appliance software images for comparison against local system image values.

To perform the MD5 integrity check, issue the **verify** command using the **/md5** keyword. For example, issuing the **verify /md5 flash:cdisk.bin** command will calculate and display the MD5 value for the software image. Compare this value with the value available on Cisco.com for this image.

Alternatively, you can get the MD5 value from Cisco.com first, then specify this value in the command syntax. For example, issuing the **verify /md5 flash:cdisk.bin 8b5f3062c4cacdbae72571440e962233** command will display a message verifying that the MD5 values match or that there is a mismatch. A mismatch in MD5 values means that either the image is corrupt or the wrong MD5 value was entered.

Examples

The following example verifies an image file. This is the same result you would see if you included the **/signature** keyword.

```
> verify os.img
Verifying file integrity of disk0:/os.img
Computed Hash   SHA2: 4916c9b70ad368feb02a0597fbef798e
                      ca360037fc0bb596c78e7ef916c6c398
                      e238e2597eab213d5c48161df3e6f4a7
                      66e4ec15a7b327ee26963b2fd6e2b347
Embedded Hash   SHA2: 4916c9b70ad368feb02a0597fbef798e
                      ca360037fc0bb596c78e7ef916c6c398
                      e238e2597eab213d5c48161df3e6f4a7
                      66e4ec15a7b327ee26963b2fd6e2b347
Digital signature successfully validated
```

The following example calculates an MD5 value for the image. Most exclamation points have been removed for brevity.

```
> verify /md5 os.img
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!Done!
verify /MD5 (disk0:/os.img) = 0940c6c71d3d43b3ba495f7290f4f276
>
```

The following example calculates an MD5 value and compares it to the expected value. The decision in this case is Verified, the calculated and expected values match.

```
> verify /md5 os.img 0940c6c71d3d43b3ba495f7290f4f276
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!Done!
Verified (disk0:/os.img) = 0940c6c71d3d43b3ba495f7290f4f276
>
```

The following example computes the SHA-512 value for the image.

```
> verify /sha-512 os.img
!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!Done!
verify /SHA-512 (disk0:/os.img) = 77421c0f6498976fbe5300e62bd8b7e8140b52a851f055265080
a392299848a77227d6047827192f34d969d36944abf2bddd215ec4127f9503173f82a2d6c7e2
```

Related Commands

Command	Description
copy	Copies files.
dir	Lists the files in the system.

vpn-sessiondb logoff

To log off all or selected VPN sessions, use the **vpn-sessiondb logoff** command.

vpn-sessiondb logoff {**all** | **index** *index_number* | **ipaddress** *IPaddr* | **l2l** | **name** *username* | **protocol** *protocol-name* | **tunnel-group** *groupname*} **noconfirm**

Syntax Description

all	Logs off all VPN sessions.
index *index_number*	Logs off a single session by index number. You can view index numbers for each session with the **show vpn-sessiondb detail** command.
ipaddress *IPaddr*	Logs off sessions for the IP address that you specify.
l2l	Logs off all LAN-to-LAN sessions.
name *username*	Logs off sessions for the username that you specify.
protocol *protocol-name*	Logs off sessions for protocols that you specify. The protocols include: • **ikev1**—Internet Key Exchange version 1 (IKEv1) sessions. • **ikev2**—Internet Key Exchange version 2 (IKEv2) sessions. • **ipsec**—IPsec sessions using either IKEv1 or IKEv2. • **ipseclan2lan**—IPsec LAN-to-LAN sessions. • **ipseclan2lanovernatt**—IPsec LAN-to-LAN over NAT-T sessions.
tunnel-group *groupname*	Logs off sessions for the tunnel group (connection profile) that you specify.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows how to log off sessions for the Corporate tunnel group (connection profile).

```
> vpn-sessiondb logoff tunnel-group Corporate noconfirm
INFO: Number of sessions from TunnelGroup "Corporate" logged off : 1
```

write net

To save the running configuration to a TFTP server, use the **write net** command.

write net [**interface** *if_name*] *server***:**[*filename*]

Syntax Description

:*filename*	Specifies the path and filename.
interface *if_name*	The name of the interface through which the TFTP server can be reached.
*server***:**	Sets the TFTP server IP address or name.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The running configuration is the configuration currently running in memory.

Examples

The following example copies the running configuration to a TFTP server through the inside interface.

```
> write net interface inside 10.1.1.1:/configs/contextbackup.cfg
```

Related Commands

Command	Description
show running-config	Shows the running configuration.

write terminal

To show the running configuration on the terminal, use the **write terminal** command.

write terminal

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

This command is equivalent to the **show running-config** command.

Examples

The following example writes the running configuration to the terminal:

```
> write terminal
: Saved
:
: Serial Number: XXXXXXXXXXX
: Hardware:   ASA5516, 8192 MB RAM, CPU Atom C2000 series 2416 MHz, 1 CPU (8 cores)
:
NGFW Version 6.2.0
!
hostname firepower
(...remaining output deleted...)
```