

# Network Diagnostics and Configuration Commands


- ping, on page 2
- flowctl, on page 3
- route, on page 6
- ssh, on page 7
- tcpdump, on page 8
- ip address, on page 9
- iptables, on page 10
- nslookup, on page 11
- ntpq, on page 12

# ping

Use the **ping** command to check if a host is reachable.

**ping** [ *options* ] [ *destination* ]

This example checks if the host 1.1.1.1 is reachable:

```
root@center100:~# ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1): 56 data bytes
64 bytes from 1.1.1.1: icmp_seq=0 ttl=54 time=21.439 ms
64 bytes from 1.1.1.1: icmp_seq=1 ttl=54 time=12.201 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=54 time=19.945 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=54 time=19.250 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=54 time=20.691 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=54 time=13.313 ms
64 bytes from 1.1.1.1: icmp_seq=6 ttl=54 time=19.154 ms
64 bytes from 1.1.1.1: icmp_seq=7 ttl=54 time=19.491 ms
64 bytes from 1.1.1.1: icmp_seq=8 ttl=54 time=13.291 ms
```

For more information, see ping.

# flowctl

Use the **flowctl** command to manage and troubleshoot deep packet inspection (DPI) on Cisco Cyber Vision sensors.

**flowctl** [ *options* ] *command* [ *args..* ]

## Syntax Description

| | |
|---|---|
| **Options**<br>`--port` | Specifies the flow HTTP port.<br>Default: `6666` |
| `--syncd-sock` | Specifies the sensorsyncd HTTP socket.<br>Default: `/tmp/sensorsyncd/sensorsyncd.sock` |
| `--scand-port` | Specifies the scan HTTP port.<br>Default: `6668` |
| **Commands**<br>`add-dpad-variable-export-periods` | Adds variable export periods for DPAD. |
| `buffer-ratio` | Sets the flow buffer ratio (RAM percentage). |
| `count-files` | Counts the number of files in the flow directory. |
| `disk` | Displays disk usage. |
| `dump-pcap` | Writes a PCAP file containing the latest received packets. |
| `environment` | Displays IOX environment variables. |
| `flushtcp`, | Asks flow to flush all TCP connections (use with caution). |
| `forget`, | Forgets pending data and reloads services.<br>Default: `False` |
| `list-dpad-variable-export-periods` | Lists current variable export periods for DPAD. |
| `meminfo` | Displays memory information. |
| `network-interfaces` | Displays network interface information. |
| `pause` | Pauses flow processing. |
| `pids` | Returns flow PIDs. |
| `ping` | Checks flow status. |
| `print-conf` | Prints the flow configuration file. |
| `processes` | Displays information about running processes. |
| `read-capture-file`, | Uploads, and analyzes a PCAP file (modified timestamps). |

| | |
|---|---|
| `read-capture-file-raw`, | Uploads, and analyzes a PCAP file (original timestamps). |
| `reload` | Asks flow to reload the configuration file. |
| `remove-filter` | Removes BPF filter from flow configuration. |
| `set-dpad-variable-export-periods` | Replaces variable export periods for DPAD. |
| `set-sensor-id` | Updates the sensor ID in the flow configuration. |
| `since-last-captured-packet` | Displays the duration since the last captured packet (in milliseconds). |
| `start-recording` | Starts recording packets. |
| `stats` | Prints flow run time statistics. |
| `stop-recording` | Stops recording packets. |
| `sub-dpad-variable-export-periods` | Removes variable export periods for DPAD. |
| `syncd-stats` | Prints sensorsyncd run-time statistics. |
| `unpause` | Unpauses flow processing. |
| `update-filter` | Updates BPF filter in flow configuration. |

## Command History

| Release | Modification |
|---|---|
| 4.0 | This command was introduced. |

This example displays how to print the flow run-time statistics:

```
sh-5.0# flowctl stats --human
{
  "flow_dumper_active": 0,
  "flow_internal_buffer_length": 0,
  "flow_internal_buffer_memory": 12582912,
  "flow_internal_buffer_use_percent": 0,
  "flow_nb_afpacket_captured_packets_eth1": 572724,
  "flow_nb_afpacket_dropped_packets_eth1": 0,
  "flow_nb_erspan_decapsulation_error": 0,
  "flow_nb_erspan_fragemented_packets": 0,
  "flow_nb_erspan_ip4": 0,
  "flow_nb_flows_in_flowtable": 4,
  "flow_nb_hsrp_lru_errors": 0,
  "flow_nb_iface_dropped_packets_eth1": 0,
  "flow_nb_ipv4_defrag_errors": 0,
  "flow_nb_no_flow_cleaned_up": 0,
  "flow_nb_packets_deduplicated": 497837,
  "flow_nb_packets_per_interface_eth1": 572724,
  "flow_nb_panics_recovered": 0,
  "flow_nb_s7plus_subscriptions": 0,
  "flow_nb_s7plus_subscriptions_dropped": 0,
  "flow_nb_scan_detected_sources": 0,
  "flow_nb_too_many_flows": 0,
  "flow_nb_tracked_packets": 74887,
```

```
  "flow_nb_tracked_packets_per_layer_icmp": 10,
  "flow_nb_tracked_packets_per_layer_tcp": 354,
  "flow_nb_tracked_packets_per_protocol_arp": 4,
  "flow_nb_tracked_packets_per_protocol_cisco_discovery": 26396,
  "flow_nb_tracked_packets_per_protocol_icmp": 10,
  "flow_nb_tracked_packets_per_protocol_lldp": 95152,
  "flow_nb_tracked_packets_per_protocol_smb": 222,
  "flow_nb_tracked_packets_per_protocol_snap": 26396,
  "flow_paused": 0,
  "flow_since_last_captured_packet_ms": 461,
  "flow_sum_capture_length_eth1": 82221916,
  "flow_sum_length_eth1": 82221916,
  "flow_sum_tracked_capture_length": 26207377,
  "flow_sum_tracked_length": 26207377
}
```

# route

Use the **route** command to view the routing table.

**route -n**

This example displays how to check the routing table:

```
root@center100:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.2.3.254      0.0.0.0         UG    0      0        0 eth0
10.2.0.0        0.0.0.0         255.255.252.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.255.248 U     0      0        0 brrsyslogd
169.254.0.8     0.0.0.0         255.255.255.252 U     0      0        0 brntpd
169.254.0.16    0.0.0.0         255.255.255.248 U     0      0        0 brburrow
169.254.0.32    0.0.0.0         255.255.255.248 U     0      0        0 brbackend
169.254.0.40    0.0.0.0         255.255.255.252 U     0      0        0 brhaproxyadmin
169.254.0.48    0.0.0.0         255.255.255.252 U     0      0        0 brhaproxyacq
169.254.0.56    0.0.0.0         255.255.255.248 U     0      0        0 brhaproxylog
169.254.0.64    0.0.0.0         255.255.255.248 U     0      0        0 bralfred
169.254.0.72    0.0.0.0         255.255.255.248 U     0      0        0 brsysinfodh
169.254.0.80    0.0.0.0         255.255.255.248 U     0      0        0 brsensorinputd
169.254.0.88    0.0.0.0         255.255.255.248 U     0      0        0 brpxgridagent
169.254.0.96    0.0.0.0         255.255.255.248 U     0      0        0 brext-apid
169.254.0.120   0.0.0.0         255.255.255.248 U     0      0        0 brsyncd
169.254.0.128   0.0.0.0         255.255.255.248 U     0      0        0 braspic
169.254.0.136   0.0.0.0         255.255.255.248 U     0      0        0 brnodeexporter
169.254.0.144   0.0.0.0         255.255.255.248 U     0      0        0 brpgexporter
169.254.0.152   0.0.0.0         255.255.255.248 U     0      0        0 brmarmotd
169.254.0.160   0.0.0.0         255.255.255.248 U     0      0        0 brrefreshviews
169.254.0.168   0.0.0.0         255.255.255.248 U     0      0        0 brsnmp
169.254.0.224   0.0.0.0         255.255.255.224 U     0      0        0 brrmq
192.168.69.0    0.0.0.0         255.255.255.0   U     0      0        0 eth1
```

For more information, see route.

# ssh

Use the **ssh** command to securely log in to a remote machine in a network.

**ssh**

This example displays how to log in to a remote host.

```
Device# ssh cv-admin@center100
```

For more information, see ssh.

# tcpdump

Use the **tcpdump** command on the sensor application CLI to create PCAP files, which may be required for troubleshooting or issue reporting.

**tcpdump** *--options*

## Syntax Description

| | |
|---|---|
| **-i** *interface* | Creates dumps for the specified interface. |
| **-G** *rotate_seconds* | Rotates the dump file after the specified duration. |
| **-z** *postrotate_command* | Used with **-G** option. Creates a zip file using the gunzip utility. |
| **--W***filecount* | Used with **-G** option. Limits the number of rotated dump files to the specified value. |
| **-w***file* | Writes the PCAP data to the specified file. |

## Command History

| Release | Modification |
|---|---|
| Release 4.0.0 | This command was introduced. |

This example creates 5 zipped PCAP files for 120 seconds of traffic and saves the files to the `/iox_data/appdata/` folder

```
sh-5.0# tcpdump -i eth1 -G 120 -z gzip -W 5 -w /iox_data/appdata/capture-%H-%M-%S.pcaptcpdump:
 listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
Maximum file limit reached: 5
648161 packets captured
861891 packets received by filter
212901 packets dropped by kernel
```

To verify the creation of the files at the specified location, run this command:

```
sh-5.0# ls -lh /iox_data/appdata/-rw-r--r--    1 root     root        3.8M Jun 21 11:40
capture-11-37-51.pcap.gz
-rw-r--r--    1 root     root        4.6M Jun 21 11:42 capture-11-39-51.pcap.gz
-rw-r--r--    1 root     root        3.8M Jun 21 11:44 capture-11-41-58.pcap.gz
-rw-r--r--    1 root     root        4.4M Jun 21 11:46 capture-11-44-15.pcap.gz
-rw-r--r--    1 root     root        5.4M Jun 21 11:49 capture-11-46-15.pcap.gz
```

This example checks if any data is received or sent on the **eth0** interface on the Cisco Cyber Vision Center:

```
root@center100:~# tcpdump -i eth0 -wroot@center100:~# tcpdump -i eth0 -w
/data/tmp/tcpdumpeth0.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C262 packets captured
264 packets received by filter
0 packets dropped by kernel
```

For more information, see tcpdump.

# ip address

Use the **ip address** command to check the network interface IP address and the status of the interface. You can use this command for troubleshooting when the Cisco Cyber Vision Center is not reachable.

**ip address** [ **show** [ **dev** *IFNAME* ] ]

## Syntax Description

| | |
|---|---|
| IFNAME | Interface name such as eth0, eth1, and so on. |

## Command History

| Release | Modification |
|---|---|
| 3.0.0 | This command was introduced. |

This example displays how to check the network interface IP address and the status of the "eth0" interface:

```
root@center100:~# ip a show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
qlen 1000
    link/ether 00:50:56:8f:9d:16 brd ff:ff:ff:ff:ff:ff
    inet 10.2.3.102/22 brd 10.2.3.255 scope global eth0
       valid_lft forever preferred_lft forever
```

For more information, see ip address.

# iptables

Use the **iptables** command to list the packet filter rules.

**iptables -L** [ *chain* ]

## Syntax Description

| | |
|---|---|
| chain | To list rules in the specified chain. |

## Command History

| Release | Modification |
|---|---|
| 3.0.0 | This command was introduced. |

This example displays how to check the rules of the IP table:

```
root@center100:~# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
DROP       tcp  --  anywhere            anywhere             tcp flags:!FIN,SYN,RST,ACK/SYN
 state NEW
DROP       all  --  anywhere             anywhere               state INVALID
ACCEPT     all  --  anywhere             anywhere               state RELATED,ESTABLISHED
NFLOG      all  --  anywhere             anywhere               ! match-set
center_admin_networks src nflog-prefix  "DropUnAuthNetwork:" nflog-group 1
DROP       all  --  anywhere             anywhere               ! match-set
center_admin_networks src
ACCEPT     icmp --  anywhere             anywhere               icmp destination-unreachable
ACCEPT     icmp --  anywhere             anywhere               icmp source-quench
ACCEPT     icmp --  anywhere             anywhere               icmp time-exceeded
ACCEPT     icmp --  anywhere             anywhere               icmp parameter-problem
ACCEPT     icmp --  anywhere             anywhere               icmp echo-request
ACCEPT     tcp  --  anywhere             anywhere               tcp dpt:ssh state NEW
ACCEPT     udp  --  anywhere             anywhere               udp dpt:bootps state NEW
ACCEPT     udp  --  anywhere             anywhere               udp dpt:bootpc state NEW
```

For more information, see iptables.

# nslookup

Use the **nslookup** command to query the name servers for information about various hosts and domains.

**nslookup** [ *DNS_server* ]

## Syntax Description

| | |
|---|---|
| *DNS_server* | FQDN or IP address of the DNS server. |

## Command History

| Release | Modification |
|---|---|
| 3.0 | This command was introduced. |

To check the name resolution of the server, "iseccv002.lab-autom-ccv.local", run this command:

```
root@center100:~# nslookup iseccv00x.lab-ccv.local

Server:    10.2.3.254
Address 1: 10.2.3.254 _gateway

Name:      iseccv00x.lab-ccv.local
Address 1: 10.2.2.131 iseccv00x.lab-ccv.local
```

To check the name resolution of the server, "8.8.8.8", run this command:

```
root@center100:~# nslookup 8.8.8.8

Server:    208.67.220.220
Address 1: 208.67.220.220 dns.sse.cisco.com

Name:      8.8.8.8
Address 1: 8.8.8.8 dns.google
```

# ntpq

Use the **ntpq** command to check the NTP server communication details.

**Syntax**

**ntpq -c peer** *IP_address*

**Syntax Description**

| | |
|---|---|
| **IP_address** | IP address of the NTP server |

**Command History**

| Release | Modification |
|---|---|
| 3.0 | This command was introduced. |

This example displays the NTP server communication details:

```
root@center100:~# ntpq -c peer 169.254.0.10
remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 LOCAL(0)        .LOCL.          10 l 159m   64    0    0.000   +0.000   0.000
*aer01-r4d20-dc- .GNSS.           1 u   10  256  377   22.445   -3.473   0.491
```