CHAPTER 2

Tools Used in Troubleshooting

This chapter describes the troubleshooting tools available for the Cisco Nexus 1000V and includes the following topics:

Commands

You use the CLI from a local console or remotely using a Telnet or SSH session. The CLI provides a command structure similar to Cisco NX-OS software, with context-sensitive help, **show** commands, multi-user support, and role-based access control.

Each feature has show commands that provide information about the feature configuration, status, and performance. Additionally, you can use the following commands for more information:

- **show system—**Provides information on system-level components, including cores, errors, and exceptions. Use the show system error-id command to find details on error codes:

```
n1000v# copy running-config startup-config
[########################################] 100%
2008 Jan 16 09:59:29 zoom %$ VDC-1 %$ %BOOTVAR-2-AUTOCOPY_FAILED: Autocopy of file
/bootflash/n1000-s1-dk9.4.0.0.837.bin.S8 to standby failed, error=0x401e0008

n1000v# show system error-id 0x401e0008
Error Facility: sysmgr
Error Description: request was aborted, standby disk may be full
```

Ping

The ping utility generates a series of *echo* packets to a destination across a TCP/IP internetwork. When the echo packets arrive at the destination, they are rerouted and sent back to the source. Using ping, you can verify connectivity and latency to a particular destination across an IP routed network.

The ping allows you to ping a port or end device. By specifying the IPv4 address, you can send a series of frames to a target destination. Once these frames reach the target, they are looped back to the source and a time-stamp is taken. Ping helps you to verify the connectivity and latency to destination.

Traceroute

Use traceroute to:

- Trace the route followed by data traffic.
- Compute inter-switch (hop-to-hop) latency.

Traceroute identifies the path taken on a hop-by-hop basis and includes a timestamp at each hop in both directions. You can use traceroute to test the connectivity of ports along the path between the generating switch and the switch closest to the destination.

Use the **traceroute** CLI command to access this feature.

If the destination cannot be reached, the path discovery starts, which traces the path up to the point of failure.

Monitoring Processes and CPUs

There are features in the CLI for monitoring switch processes and CPU status and utilization.

This section contains the following topics:

Identifying the Processes Running and their States

Use the **show processes command** to identify the processes that are running and the status of each process. (See Example 2-1.) The command output includes:

- PID = process ID.
- State = process state.
- PC = current program counter in hex format.
- Start_cnt = how many times a process has been started (or restarted).
- TTY = terminal that controls the process. A “-” usually means a daemon not running on any particular TTY.
- Process = name of the process.

Process states are:

- D = uninterruptible sleep (usually I/O).
- R = runnable (on run queue).
- S = sleeping.
- T = traced or stopped.

- Z = defunct (“zombie”) process.
- NR = not-running.
- ER = should be running but currently not-running.

Note The ER state typically designates a process that has been restarted too many times, causing the system to classify it as faulty and disable it.

Example 2-1 show processes Command

```
n1000v# show processes ?
  cpu     Show processes CPU Info
  log     Show information about process logs
  memory  Show processes Memory Info

n1000v# show processes

PID    State  PC        Start_cnt    TTY   Process
-----  -----  --------  -----------  ----  -------------
    1      S  b7f9e468            1     -  init
    2      S         0            1     -  migration/0
    3      S         0            1     -  ksoftirqd/0
    4      S         0            1     -  desched/0
    5      S         0            1     -  migration/1
    6      S         0            1     -  ksoftirqd/1
    7      S         0            1     -  desched/1
    8      S         0            1     -  events/0
    9      S         0            1     -  events/1
   10      S         0            1     -  khelper
   15      S         0            1     -  kthread
   24      S         0            1     -  kacpid
  101      S         0            1     -  kblockd/0
  102      S         0            1     -  kblockd/1
  115      S         0            1     -  khubd
  191      S         0            1     -  pdflush
  192      S         0            1     -  pdflushn
...
```

Displaying CPU Utilization

Use the **show processes cpu** command to display CPU utilization. The command output includes:

- Runtime(ms) = CPU time the process has used, expressed in milliseconds.
- Invoked = number of times the process has been invoked.
- uSecs = microseconds of CPU time in average for each process invocation.
- 1Sec = CPU utilization in percentage for the last one second.

Example 2-2 show processes cpu Command

```
n1000v# show processes cpu

PID    Runtime(ms)  Invoked   uSecs  1Sec   Process
-----  -----------  --------  -----  -----  -----------
    1          922  4294967295      0      0  init
    2          580    377810      1      0  migration/0
```

```
    3          889   3156260       0       0  ksoftirqd/0
    4         1648    532020       3       0  desched/0
    5          400    150060       2       0  migration/1
    6         1929   2882820       0       0  ksoftirqd/1
    7         1269    183010       6       0  desched/1
    8         2520  47589180       0       0  events/0
    9         1730   2874470       0       0  events/1
   10           64    158960       0       0  khelper
   15            0    106970       0       0  kthread
   24            0     12870       0       0  kacpid
  101           62   3737520       0       0  kblockd/0
  102           82   3806840       0       0  kblockd/1
  115            0     67290       0       0  khubd
  191            0      5810       0       0  pdflush
  192          983   4141020       0       0  pdflush
  194            0      5700       0       0  aio/0
  193            0      8890       0       0  kswapd0
  195            0      5750       0       0  aio/1
...
```

Displaying CPU and Memory Information

Use the **show system resources** command to display system-related CPU and memory statistics. The output includes the following:

- Load is defined as number of running processes. The average reflects the system load over the past 1, 5, and 15 minutes.
- Processes displays the number of processes in the system, and how many are actually running when the command is issued.
- CPU states shows the CPU usage percentage in user mode, kernel mode, and idle time in the last one second.
- Memory usage provides the total memory, used memory, free memory, memory used for buffers, and memory used for cache in KB. Buffers and cache are also included in the used memory statistics.

Example 2-3 show system resources Command

```
n1000v# show system resources
Load average:   1 minute: 0.30   5 minutes: 0.34   15 minutes: 0.28
Processes   :   606 total, 2 running
CPU states  :   0.0% user,   0.0% kernel,   100.0% idle
Memory usage:   2063268K total,   1725944K used,    337324K free
                   2420K buffers,  857644K cache
```

RADIUS

RADIUS is a protocol used for the exchange of attributes or credentials between a head-end RADIUS server and a client device. These attributes relate to three classes of services:

- Authentication
- Authorization
- Accounting

Authentication refers to the authentication of users for access to a specific device. You can use RADIUS to manage user accounts for access to an Cisco Nexus 1000V device. When you try to log into a device, Cisco Nexus 1000V validates you with information from a central RADIUS server.

Authorization refers to the scope of access that you have once you have been authenticated. Assigned roles for users can be stored in a RADIUS server along with a list of actual devices that the user should have access to. Once the user has been authenticated, then switch can then refer to the RADIUS server to determine the extent of access the user will have within the switch network.

Accounting refers to the log information that is kept for each management session in a switch. This information may be used to generate reports for troubleshooting purposes and user accountability. Accounting can be implemented locally or remotely (using RADIUS).

The following is an example of an accounting log entries.

```
n1000v# show accounting log
Sun Dec 15 04:02:27 2002:start:/dev/pts/0_1039924947:admin
Sun Dec 15 04:02:28 2002:stop:/dev/pts/0_1039924947:admin:vsh exited normally
Sun Dec 15 04:02:33 2002:start:/dev/pts/0_1039924953:admin
Sun Dec 15 04:02:34 2002:stop:/dev/pts/0_1039924953:admin:vsh exited normally
Sun Dec 15 05:02:08 2002:start:snmp_1039928528_172.22.95.167:public
Sun Dec 15 05:02:08 2002:update:snmp_1039928528_172.22.95.167:public:Switchname
```

Note The accounting log only shows the beginning and ending (start and stop) for each session.

Syslog

The system message logging software saves messages in a log file or directs the messages to other devices. This feature provides the following capabilities:

- Logging information for monitoring and troubleshooting.
- Selection of the types of logging information to be captured.
- Selection of the destination of the captured logging information.

Syslog lets you store a chronological log of system messages locally or sent to a central Syslog server. Syslog messages can also be sent to the console for immediate use. These messages can vary in detail depending on the configuration that you choose.

Syslog messages are categorized into 7 severity levels from *debug to critical* events. You can limit the severity levels that are reported for specific services within the switch.

Log messages are not saved across system reboots. However, a maximum of 100 log messages with a severity level of critical and below (levels 0, 1, and 2) can logged to a local file or server.

Logging Levels

Cisco Nexus 1000V supports the following logging levels:

- 0-emergency
- 1-alert
- 2-critical
- 3-error
- 4-warning

- 5-notification
- 6-informational
- 7-debugging

By default, the switch logs normal but significant system messages to a log file and sends these messages to the system console. Users can specify which system messages should be saved based on the type of facility and the severity level. Messages are time-stamped to enhance real-time debugging and management.

Enabling Logging for Telnet or SSH

System logging messages are sent to the console based on the default or configured logging facility and severity values.

Users can disable logging to the console or enable logging to a given Telnet or SSH session.

- To disable console logging, use the **no logging console** command in CONFIG mode.
- To enable logging for telnet or SSH, use the **terminal monitor** command in EXEC mode.

Note Note: When logging to a console session is disabled or enabled, that state is applied to all future console sessions. If a user exits and logs in again to a new session, the state is preserved. However, when logging to a Telnet or SSH session is enabled or disabled, that state is applied only to that session. The state is not preserved after the user exits the session.

The **no logging console** command shown in Example 2-4:

- Disables console logging
- Enabled by default

Example 2-4 no logging console Command

```
n1000v(config)# no logging console
```

The **terminal monitor** command shown in Example 2-5:

- Enables logging for telnet or SSH
- Disabled by default

Example 2-5 terminal monitor Command

```
n1000v# terminal monitor
```

For more information about configuring syslog, see the *Cisco Nexus 1000V System Management Configuration Guide, Release 4.0(4)SV1(3)*.