CCNA Exam Prep: Back to Networking Basics with Hank Preston and Patrick Gargano -- Season 2

The Syslog Detective: Unraveling Network Mysteries

Ever been overwhelmed by console messages when you were trying to configure a Cisco router? Ever wonder what all those console messages mean, where they come from and how to control where their displayed? Maybe it's an interface going down or an OSPF neighbor coming up. These are all useful and important messages but too many of them can cause system interruptions and distract you.

You can use a protocol called syslog to send these console messages to a server to collect them for easy access and analysis.

In this lab you will explore how log messages work and how to configure and verify syslog.

Related CCNA v1.1 exam topic:

- 4.5 Describe the use of syslog features including facilities and levels

In this lab, we will explore how to:

- Verify the default logging configuration
- Explore where log messages can be sent
- Configure and verify logging to a syslog server

Setup and Scenario

In this set of lab-based demonstrations, you are a network engineer tasked with exploring and testing how log messages can be displayed in real-time or stored for later review.

You've been asked to:

- Configure console, terminal monitor, buffer and server logging.

*Be sure to **START** the lab before continuing to the demo lab.*

> Note: The credentials for all devices are **cisco / cisco**

Part 1: Reviewing the current logging configuration on the rtr01 router

Before we jump into configuring logging, let's explore what its default settings are.

Step 1

Open a console connection to the `rtr01` router and verify its default logging configuration. Use the `show run all` command.

```
rtr01# show run all | include logging
no service pt-vty-logging
no logging discriminator
logging exception 4096
no logging count
no logging message-counter log
no logging message-counter debug
logging message-counter syslog
```

```
no logging snmp-authfail
no logging userinfo
logging buginf
logging queue-limit 100
logging queue-limit esm 0
logging queue-limit trap 100
logging buffered 4096 debugging
logging reload message-limit 1000 notifications
no logging persistent
logging rate-limit console 40 except errors
logging console guaranteed
logging console debugging
logging monitor debugging
logging cns-events informational
logging on
ethernet cfm logging alarm ieee
ethernet cfm logging alarm cisco
ethernet cfm logging ais
ethernet cfm logging lck
ip dhcp conflict logging
no device-tracking logging packet drop
no device-tracking logging theft
no device-tracking logging resolution-veto
mpls ldp logging neighbor-changes
mpls ldp logging password configuration
mpls ldp logging password rollover
no cts logging verbose
 logging event link-status
 logging event link-status
 logging event link-status
 logging event link-status
 logging event link-status
ip ssh logging events
logging dmvpn rate-limit 600
logging history size 1
logging history warnings
logging trap informational
logging delimiter tcp
no logging origin-id
logging facility local7
no logging source-interface
logging server-arp
snmp-server enable logging setop
 logging synchronous
no device-tracking binding logging
```

Focusing on the lines in the output that are highlighted, we see that the router is configured to send debugging (or level 7) messages to the console, monitor, and buffer. We further see that logging is turned on, and that the trap logging level is informational (or level 6), and that the logging facility is number 7. The logging trap command has to deal with what messages are sent to an external server, while the facility is how the server routes the log messages when they are received. A facility can be a hardware device, a protocol, or a module of the system software. It denotes the source or the cause of the system message. (Log facility 7 indicates one of several custom logging facilities, which are typically tied to a specific file). We will look at the facility concept in more detail later in the lab.

Step 2

Verify how the `rtr01` router is configured to display time stamps when producing log messages.

```
rtr01# show run | section timestamp
service timestamps debug datetime msec
service timestamps log datetime msec
```

By default, the router is configured to time stamp all debug and logging messages that are produced at the console or elsewhere, down to the millisecond. These time stamps are displayed in UTC time by default but it is possible to display them in the local time zone of the device. Also, you can configure the router to display sequence numbers for all log messages instead of the date and time.

Step 3

Check the current time on the `rtr01` router. Although the router is configured to get its time from the NTP server, you know from the previous CCNA Prep episode on NTP that it can take a long time to synchronize its clock. If the time on the router is not accurate, go ahead and manually set it for the correct UTC time.

```
rtr01# show clock
22:57:50.203 UTC Thu Jan 14 1993

rtr01# show ntp status
Clock is unsynchronized, stratum 2, reference is 192.168.100.100
nominal freq is 250.0000 Hz, actual freq is 249.8750 Hz, precision is 2**10
ntp uptime is 820200 (1/100 of seconds), resolution is 4016
reference time is AF006F7A.AB854306 (23:02:18.670 UTC Thu Jan 14 1993)
clock offset is -825084165.0000 msec, root delay is 1.94 msec
root dispersion is -825076224.00 msec, peer dispersion is 7938.96 msec
loopfilter state is 'SPIK' (Spike), drift is 0.000499999 s/s
system poll interval is 64, last update was 45 sec ago.

rtr01# clock set 17:55:00 8 April 2025
rtr01#
Apr  8 17:55:00.000: %SYS-6-CLOCKUPDATE: System clock has been updated from 23:04:58 UTC Thu Jan 14
1993 to 17:55:00 UTC Tue Apr 8 2025, configured from console by console.
```

It is critical to note that prior to configuring any device to send log information, the date and time of the clock must be properly configured for accurate time. If it isn't, the time stamps on all the logging messages will not reflect the appropriate and accurate time, which will make troubleshooting much more difficult because you will not be able to correlate issues with the logs by using the time stamps generated.

Notice the log message that is generated at the console when you manually set the clock. Let's break it down:

First, you are shown a time stamp down to the millisecond: `Apr 8 17:55:00.000`

Second, you are shown the facility on the router that generated the message: `%SYS`

Third, you are shown the severity level of the message: `6` (informational)

Fourth, an abbreviation for the message: `CLOCKUPDATE`

Finally, a description of the message itself: `System clock has been updated from 23:04:58 UTC Thu Jan 14 1993 to 17:55:00 UTC Tue Apr 8 2025, configured from console by console`

All log messages that are generated will follow this general format.

Part 2: Configure and verify console and terminal monitor logging

Let's look again at the current logging settings for messages displayed at the console and for terminal connections, and modify their parameters.

Step 1

Use the `show logging` command on the `rtr01` router and verify the console and monitor settings.

```
rtr01# show logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0 overruns, xml
disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

    Console logging: level debugging, 60 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 61 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled
    Trap logging: level informational, 64 message lines logged
        Logging Source-Interface:       VRF Name:

Log Buffer (4096 bytes):
Jan  1 00:00:00.771: %LINK-3-UPDOWN: Interface Ethernet0/1, changed state to up
Jan  1 00:00:00.784: %LINK-3-UPDOWN: Interface Ethernet0/2, changed state to up
<...output omitted...>
Apr  8 17:55:00.000: %SYS-6-CLOCKUPDATE: System clock has been updated from 23:04:58 UTC Thu Jan 14
1993 to 17:55:00 UTC Tue Apr 8 2025, configured from console by console.
rtr01#
```

Focussing once again on the highlighted lines, you can see that logging is enabled (it can be disabled globally with the `no logging on` configuration command or more specifically with the `no logging console` or `no logging monitor` commands). For both console and monitor logging, the severity level is set to debugging (level 7).

Network devices should log levels 0â€“6 under normal operation. Level 7 should be used for console troubleshooting only.

Note that monitor logging messages refer to messages sent to other logged-in users, for example Telnet and SSH users connected to the VTY terminal lines.

Step 2

Configure monitor logging for level 3 and below (error). Then `telnet` from the `netadmin` PC to the `rtr01` router. When configuring a severity level for logging, the level chosen always includes more severe levels. Since you are configuring monitor logging for level 3, this will also include levels 0, 1, and 2 as well.

```
rtr01(config)# logging monitor ?
  <0-7>          Logging severity level
  alerts         Immediate action needed           (severity=1)
  critical       Critical conditions               (severity=2)
  debugging      Debugging messages                (severity=7)
  discriminator  Establish MD-Console association
  emergencies    System is unusable                (severity=0)
  errors         Error conditions                  (severity=3)
  informational  Informational messages            (severity=6)
  notifications  Normal but significant conditions (severity=5)
  warnings       Warning conditions                (severity=4)
  xml            Enable logging in XML

rtr01(config)# logging monitor 3


netadmin:~$ telnet rtr01
Connected to rtr01

Entering character mode
Escape character is '^]'.


User Access Verification

Password: telnetpass
rtr01#
```

Step 3

From the telnet session, enable `terminal monitoring`. Then disable and re-enable the Ethernet0/0 interface. Compare the output from the telnet session to the output at the router's actual console.

```
FROM THE TELNET SESSION
rtr01# terminal monitor
rtr01# config t
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
rtr01(config)# interface Ethernet0/0
rtr01(config-if)# shutdown
rtr01(config-if)# no shutdown
rtr01(config-if)#
Apr  8 18:58:57.425: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up


FROM THE ROUTER'S CONSOLE
rtr01#
Apr  8 18:58:53.745: %LINK-5-CHANGED: Interface Ethernet0/0, changed state to administratively down
Apr  8 18:58:54.745: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state to
down
rtr01#
Apr  8 18:58:57.425: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
rtr01#
Apr  8 18:58:58.425: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state to
up
rtr01#
Apr  8 18:59:02.567: %DHCP-6-ADDRESS_ASSIGN: Interface Ethernet0/0 assigned DHCP address
192.168.255.81, mask 255.255.255.0, hostname rtr01
```

The `terminal monitor` command is necessary to allow your Telnet session to receive log messages otherwise they will not be displayed, while the previous `logging monitor 3` command set the severity level of the messages sent to any terminal user.

Notice that the output from the Telnet session only displays the severity level 3 "interface up" message, while the router's console displays all up/down messages, as well as the DHCP message, since the console is still configured to display messages up to severity level 7.

Step 4

Use the `show logging` command again to verify the current logging configuration.

```
rtr01# show logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0 overruns, xml
disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

    Console logging: level debugging, 73 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level errors, 1 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 87 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled
    Trap logging: level informational, 90 message lines logged
        Logging Source-Interface:       VRF Name:

Log Buffer (4096 bytes):
<...output omitted...>>
```

Console logging is still set to the debugging level, while monitor logging is now set to the errors level. Notice the number of messages generated for each. Your results will perhaps differ from what is show above.

Part 3: Configure and verify buffer and trap logging

Besides sending log messages to the console and to the terminal monitor, it is also possible to save log messages to a local buffer on the router, as well as send the log messages to an external server. RFC 5424 defines the syslog protocol which allows network devices to use UDP to send log messages to a server for storage. In this part of the lab we will configure and verify both buffer logging and trap logging to a server.

Step 1

Return to the rtr01 router and consult the `show logging` output you generated in the last step of Part 2.

```
rtr01# show logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0 overruns, xml
disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

    Console logging: level debugging, 73 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level errors, 1 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 87 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled
    Trap logging: level informational, 90 message lines logged
        Logging Source-Interface:       VRF Name:

Log Buffer (4096 bytes):
<...output omitted...>>
```

Focussing on the highlighted lines, you see that the buffer is currently set for severity level 7 (debugging), that the buffer size is set to 4096 bytes, and that trap logging to a syslog server is set to severity level 6 (informational)

Since the syslog server IP address has not yet been configured on the router, the trap logging messages are not currently being sent anywhere. We will correct that shortly.

The buffer logs are automatically displayed when you use the `show logging` command and the output can be rather overwhelming as you scroll through multiple pages of logs. Let's minimize what is sent to the buffer but also increase the buffer size to hold more messages.

Step 2

Configure the logging buffer for severity level 4 (warnings) and set the buffer size to 8192 bytes so that it can hold more messages.

```
rtr01(config)# logging buffer 8192 4
rtr01(config)#
Apr 10 14:50:09.445: %SYS-5-LOG_CONFIG_CHANGE: Buffer logging: level warnings, xml disabled,
filtering disabled, size (8192)
```

The `logging buffer` configuration command allows you to set both the size of the buffer and the severity level.

Notice that the log message that is displayed at the console in response is a severity level 5 message. If you do a `show logging` command you will see that the buffer has been erased and will now only show severity level 4 and lower messages.

To generate a message that will be sent to the buffer, you can disable and re-enable the E0/0 interface as we did in an earlier step. This will generate a severity level 3 message which will be sent to the buffer.

```
rtr01(config)# interface e0/0
rtr01(config-if)# shutdown
rtr01(config-if)#
Apr 10 14:55:35.060: %LINK-5-CHANGED: Interface Ethernet0/0, changed state to administratively down
Apr 10 14:55:36.060: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state to
down
rtr01(config-if)# no shutdown
rtr01(config-if)#
Apr 10 14:55:42.327: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
rtr01(config-if)#
Apr 10 14:55:43.327: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state to
up
```

```
Apr 10 14:55:47.718: %DHCP-6-ADDRESS_ASSIGN: Interface Ethernet0/0 assigned DHCP address
192.168.255.81, mask 255.255.255.0, hostname rtr01

rtr01(config-if)# do show logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0 overruns, xml
disabled, filtering disabled)

No Active Message Discriminator.



No Inactive Message Discriminator.


    Console logging: level debugging, 83 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level errors, 2 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level warnings, 1 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled
    Trap logging: level informational, 100 message lines logged
        Logging Source-Interface:       VRF Name:

Log Buffer (8192 bytes):

Apr 10 14:55:42.327: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
```

The buffer logs are now set to severity level 3 (and lower) and the buffer size is set to 8192 bytes.

Step 3

Configure trap logging to the syslog server at address 192.168.100.100. This is the same address as the NTP server.

```
rtr01(config)# logging host 192.168.100.100
rtr01(config)#
Apr 10 15:07:00.430: %SYS-6-LOGGINGHOST_STARTSTOP: Logging to host 192.168.100.100 port 514 started
- CLI initiated
```

According to the `show logging` output in Step 2, trap logging is currently set to severity level 6 (informational). You could change this with the `logging trap` configuration command but we will leave it at 6 for this lab.

Notice that after entering the `logging host` command, the console displayed a severity level 6 message confirming that logging messages are now being sent to the 192.168.100.100 device on UDP port 514. It is possible to change the transport protocol to TCP and also change the port used. Finally, it is also possible to sent trap logging messages as XML instead of regular text.

Step 4

Use the CML Packet Capture feature to view syslog messages being sent from the rtr01 router to the syslog/NTP server at 192.168.100.100. Disable and re-enable the rtr01 E0/0 interface to generate messages for the packet capture.

Right-click the link between the rtr01 router and the syslog/NTP server. Select **Packet Capture** from the contextual menu.

Click the **Settings** tab and enter `udp port 514` in the BPF field. Click **Apply**.

Create another pane and drag the Packet Capture window to it so that you can view the capture live as it happens.

Click **Start** to launch the packet capture.

Return to the rtr01 router and shut/no shut the E0/0 interface to generate some log messages. Wait for the interface to come back up and for the DHCP message to be displayed.

Stop the Packet Capture.

You should see a total of **5 syslog messages** captured (two for the interface being disabled, two for the interface being re-enabled, and one for DHCP).

Click any line in the Packet Capture window and expand the Syslog message information. You will see:

- The facility used by the router to generate the message: LOCAL7
- The type of message generated: either NOTICE (5), ERR (3), or INFO (6)
- The actual syslog message that was displayed at the console.

Step 5

Use the `show logging` command to confirm the trap logging configuration.

```
rtr01# show logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0 overruns, xml
disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

    Console logging: level debugging, 105 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level errors, 2 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level warnings, 4 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled
    Trap logging: level informational, 123 message lines logged
        Logging to 192.168.100.100  (udp port 514, audit disabled,
              link up),
              20 message lines logged,
              0 message lines rate-limited,
              0 message lines dropped-by-MD,
              xml disabled, sequence number disabled
              filtering disabled
        Logging Source-Interface:       VRF Name:

Log Buffer (8192 bytes):
<...output omitted...>>
```

The highlighted lines show that trap logging is configured to 192.168.100 100 on UDP port 514 for severity level 6 messages and lower.

To wrap up, let's return to the concept of logging facility. The definition of "facility" in a log message on Cisco IOS/IOS XE is not the same as the RFC definition of "facility" (such as local7). Cisco facilities are a free-form method of identifying the source message type such as SYS, IP, LDP, L2, MEM, FILESYS, DOT11, LINEPROTO, and so on. The complete list of Cisco facilities can be found here.

Recall that at the start of the lab and in the CML Packet Capture, you saw that the default facility was set to `local7`. This is one of the many possible RFC values. It is possible to change this to something else to allow you to filter and more quickly find certain types of messages sent to the syslog server.

To change the default facility value on a Cisco router, use the following command:

```
rtr01(config)# logging facility ?
  auth    Authorization system
  cron    Cron/at facility
  daemon  System daemons
  kern    Kernel
  local0  Local use
  local1  Local use
  local2  Local use
  local3  Local use
  local4  Local use
  local5  Local use
  local6  Local use
  local7  Local use
  lpr     Line printer system
  mail    Mail system
  news    USENET news
  sys10   System use
  sys11   System use
  sys12   System use
  sys13   System use
  sys14   System use
  sys9    System use
  syslog  Syslog itself
  user    User process
  uucp    Unix-to-Unix copy system
```

The local use facilities are not reserved; the processes and applications that do not have preassigned facility values can choose any of the eight local use facilities. As such, Cisco devices use one of the local use facilities for sending syslog messages (local7).

Congratulations! You have completed the lab. You learned how to configure and verify syslog for console, terminal monitor, buffer, and trap logging.