

VxLEARN Networks

Networking & Cybersecurity Track
Simulated Employment Program

Lab Report: Reading Server Logs

Prepared by:
Kudzaishe Majeza
Junior Network Engineer – VxLEARN Networks

Mentor:
Titus Majeza
Senior Network Engineer

Date: 12 January 2026

Contents

Objective.....	3
Tools and Environment.....	3
Part 1: Reading Log Files	3
Step 1: Opening Log Files.	4
Step 2: Actively Following Logs.	8
Part 2: Syslog	10
Part 3: Journald.....	12
Step 1: Running journalctl with no options.	13
Step 2: Journalctl and a few options.	14
Reflection.....	16

Objective

The objective of this lab is to understand how to read, monitor, and analyze server log files using common Linux command-line tools. The lab also explores centralized logging using Syslog and Journald.

Tools and Environment

- Security Workstation Virtual Machine
- Linux Terminal
- Log analysis commands: cat, more, less, tail, journalctl

Part 1: Reading Log Files

In this section, various commands were used to read log files.

Cat Command:

The cat command displays the entire content of a file at once. A drawback of using cat with large files is that the output scrolls quickly, making it difficult to review specific information.

More Command:

The more command displays files page by page. A drawback of more is that it only allows forward navigation and does not support scrolling backward.

Less Command:

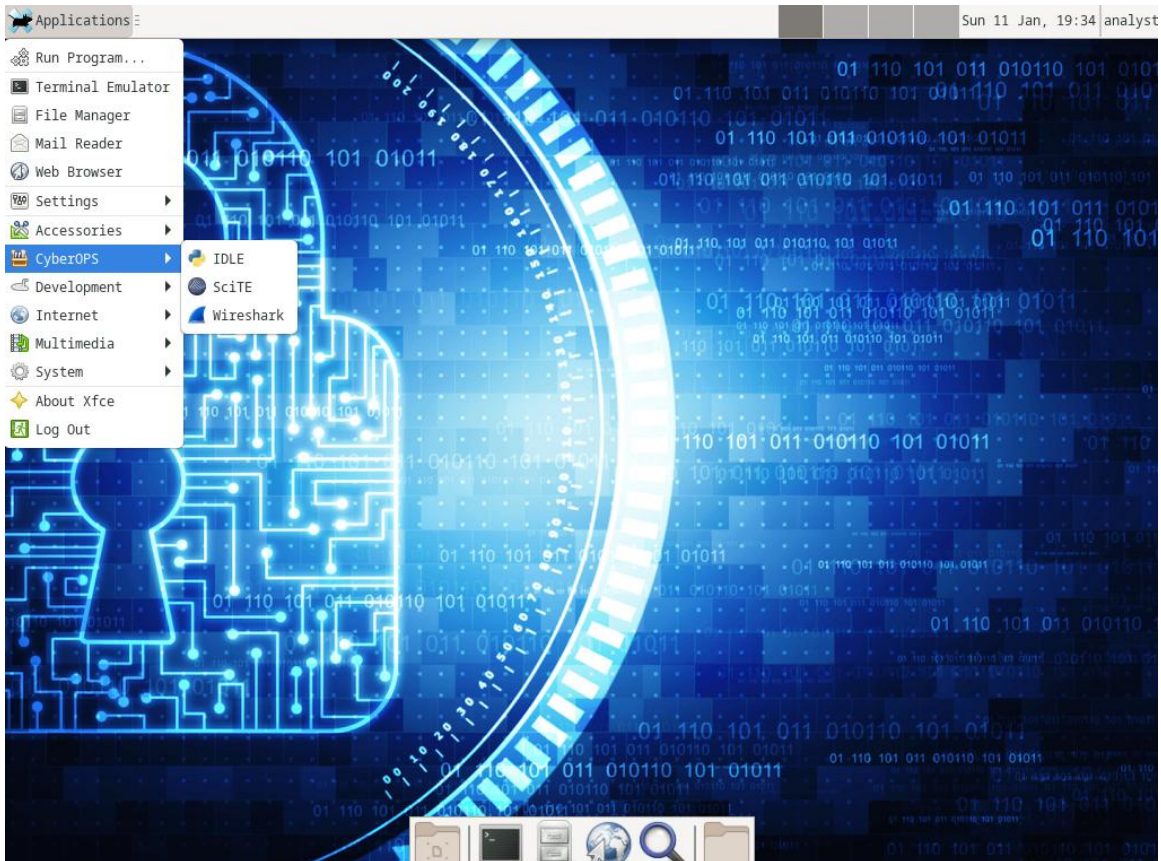
The less command improves usability by allowing both forward and backward navigation. It is ideal for analyzing large log files.

Tail and Tail -f:

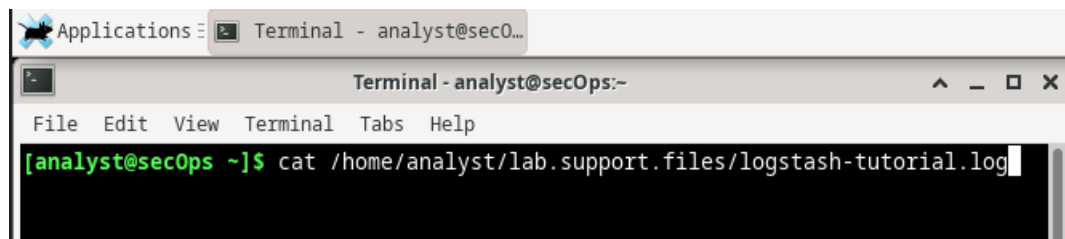
The tail command shows the last lines of a file, while tail -f continuously updates the display as new entries are added. Tail -f is useful for real-time log monitoring.

Step 1: Opening Log Files.

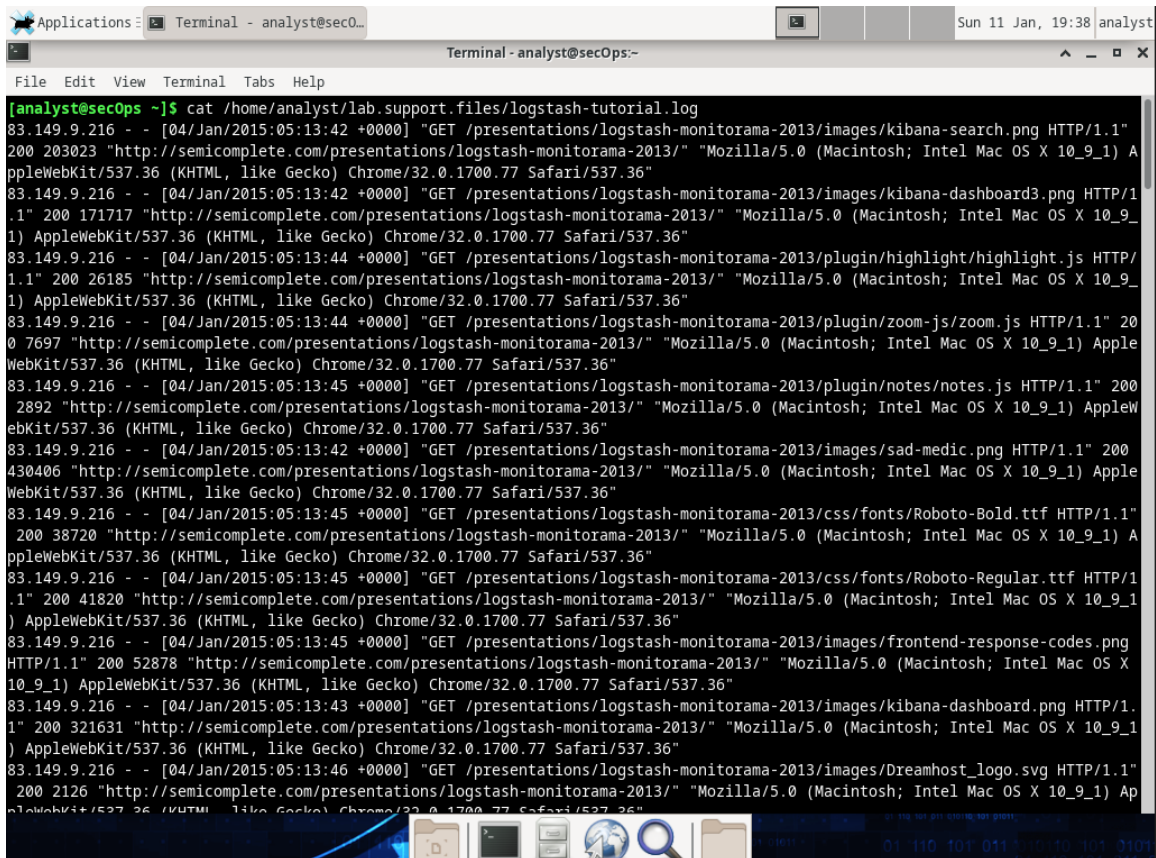
- a. Start the **CyberOps Workstation VM** and open a terminal window.



- b. From the terminal window, issue the command below to display the contents of the **logstash-tutorial.log** file, located in the **/home/analyst/lab.support.files/** folder:

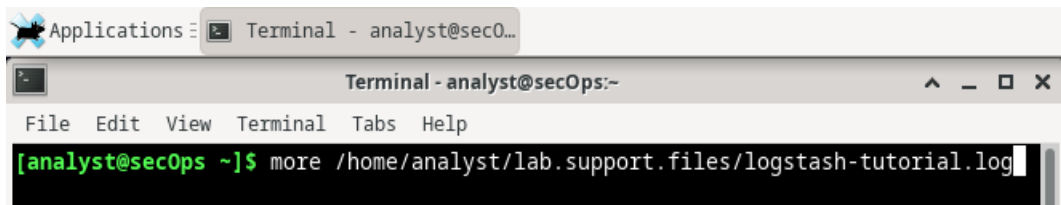


The contents of the file should scroll through the terminal window until the all contents have been displayed.

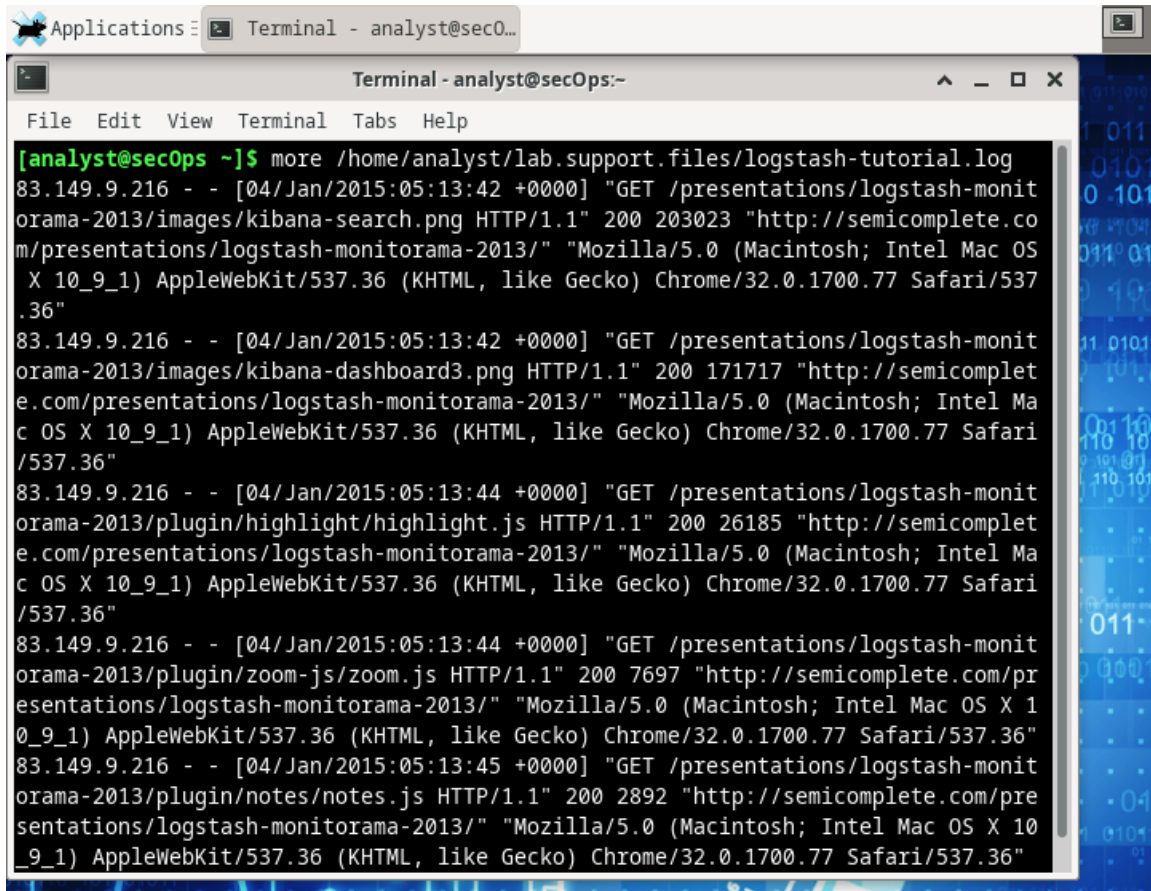
A terminal window titled 'Terminal - analyst@secOps:~' showing the output of the command 'cat /home/analyst/lab.support.files/logstash-tutorial.log'. The output is a long list of HTTP log entries, including timestamps, IP addresses, and details of GET requests to various resources like images and JavaScript files from semicomplete.com. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The status bar at the bottom shows a file manager icon and a search icon.

Another popular tool for visualizing log files is **more**. Similar to **cat**, **more** is also a UNIX command-line-based tool that can open a text-based file and display the file contents on the screen. The main difference between **cat** and **more** is that **more** supports page breaks, allowing the user to view the contents of a file, one page at a time. This can be done using the space bar to display the next page.

- c. From the same terminal window, use the command below to display the contents of the **logstash-tutorial.log** file again. This time using **more**:

A terminal window titled 'Terminal - analyst@secOps:~' showing the command 'more /home/analyst/lab.support.files/logstash-tutorial.log' being entered at the prompt. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The status bar at the bottom shows a file manager icon and a search icon.

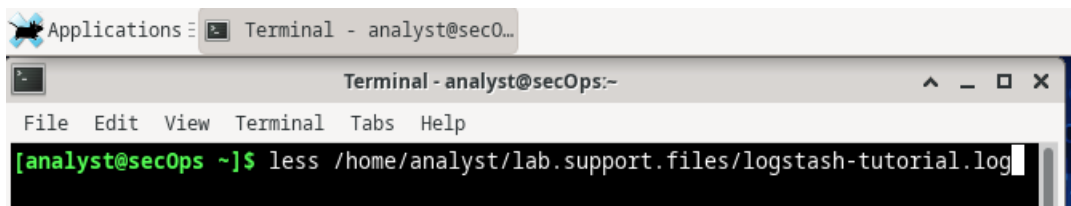
The contents of the file should scroll through the terminal window and stop when one page is displayed. Press the space bar to advance to the next page. Press enter to display the next line of text.

A screenshot of a macOS Terminal window titled "Terminal - analyst@secOps~". The window shows the output of the command `more /home/analyst/lab.support.files/logstash-tutorial.log`. The output consists of several lines of HTTP log entries, each starting with an IP address, a timestamp, and a GET request. The entries are truncated by the `more` command, showing only a portion of each line. The background of the terminal window has a blue and black pattern with binary code.

```
[analyst@secOps ~]$ more /home/analyst/lab.support.files/logstash-tutorial.log
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monit
orama-2013/images/kibana-search.png HTTP/1.1" 200 203023 "http://semicomplete.co
m/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS
X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537
.36"
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monit
orama-2013/images/kibana-dashboard3.png HTTP/1.1" 200 171717 "http://semicomplet
e.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Ma
c OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari
/537.36"
83.149.9.216 - - [04/Jan/2015:05:13:44 +0000] "GET /presentations/logstash-monit
orama-2013/plugin/highlight/highlight.js HTTP/1.1" 200 26185 "http://semicomplet
e.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Ma
c OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari
/537.36"
83.149.9.216 - - [04/Jan/2015:05:13:44 +0000] "GET /presentations/logstash-monit
orama-2013/plugin/zoom-js/zoom.js HTTP/1.1" 200 7697 "http://semicomplete.com/pr
esentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 1
0_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
83.149.9.216 - - [04/Jan/2015:05:13:45 +0000] "GET /presentations/logstash-monit
orama-2013/plugin/notes/notes.js HTTP/1.1" 200 2892 "http://semicomplete.com/pre
sentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10
_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
```

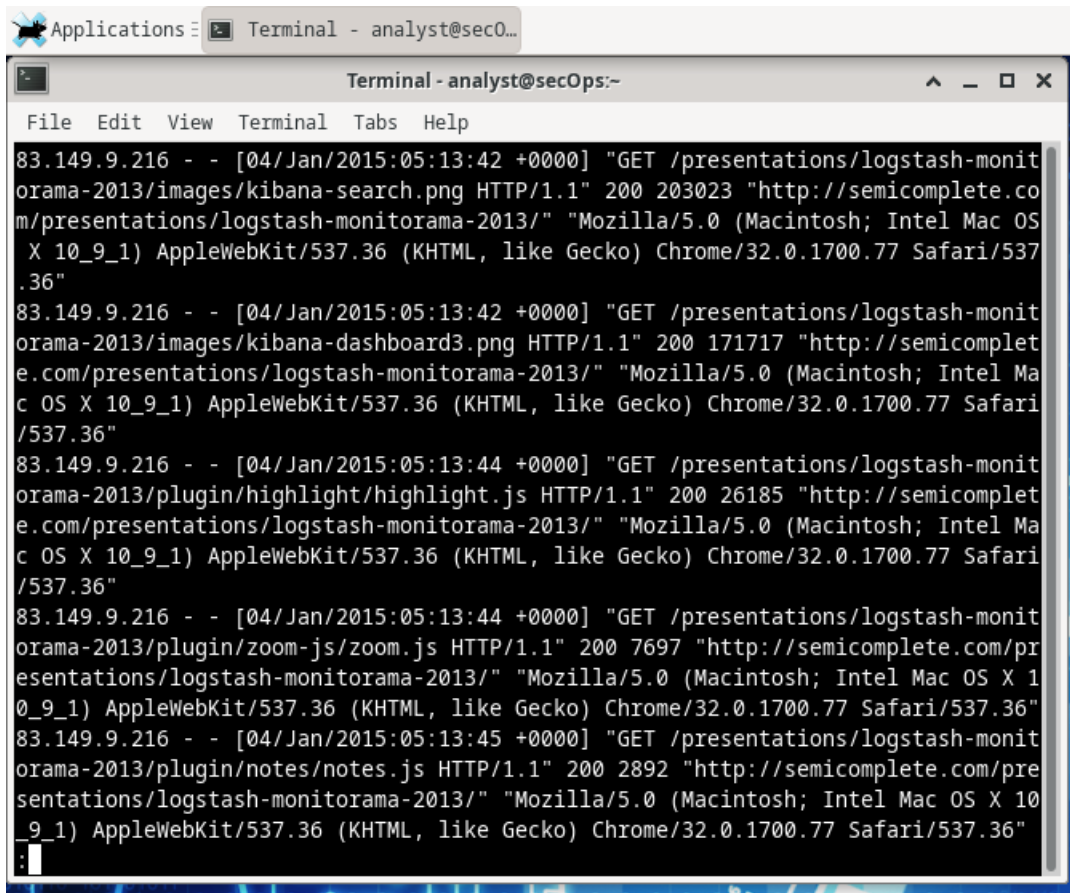
Building on the functionality of **cat** and **more**, the **less** tool allows the contents of a file to be displayed page by page, while also allowing the user the choice of viewing previously displayed pages.

- d. From the same terminal window, use **less** to display the contents the **logstash-tutorial.log** file again:

A screenshot of a macOS Terminal window titled "Terminal - analyst@secOps~". The window shows the command `less /home/analyst/lab.support.files/logstash-tutorial.log` being entered at the prompt. The terminal window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The background of the terminal window has a blue and black pattern with binary code.

```
[analyst@secOps ~]$ less /home/analyst/lab.support.files/logstash-tutorial.log
```

The contents of the file should scroll through the terminal window and stop when one page is displayed. Press the space bar to advance to the next page. Press enter to display the next line of text. Use the up and down arrow keys to move back and forth through the text file.



The screenshot shows a macOS desktop with a 'Terminal' window open. The window title is 'Terminal - analyst@secOps:~'. The terminal displays a series of HTTP GET requests from the IP address 83.149.9.216 to a Logstash monitor. The requests are for various assets including images, a dashboard, a highlight script, a zoom script, and a notes script. Each request is followed by its status (200), size, and the user agent string.

```
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/images/kibana-search.png HTTP/1.1" 200 203023 "http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/images/kibana-dashboard3.png HTTP/1.1" 200 171717 "http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
83.149.9.216 - - [04/Jan/2015:05:13:44 +0000] "GET /presentations/logstash-monitorama-2013/plugin/highlight/highlight.js HTTP/1.1" 200 26185 "http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
83.149.9.216 - - [04/Jan/2015:05:13:44 +0000] "GET /presentations/logstash-monitorama-2013/plugin/zoom-js/zoom.js HTTP/1.1" 200 7697 "http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
83.149.9.216 - - [04/Jan/2015:05:13:45 +0000] "GET /presentations/logstash-monitorama-2013/plugin/notes/notes.js HTTP/1.1" 200 2892 "http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
:
```

- e. The **tail** command displays the end of a text file. By default, **tail** displays the last ten lines of the file.

Use tail to display the last ten lines of the **/home/analyst/lab.support.files/logstash-tutorial.log** file.

```
Applications: Terminal - analyst@sec0... Sun 11 Jan, 20:01 analyst
Terminal - analyst@secOps:-
File Edit View Terminal Tabs Help
[analyst@secOps ~]$ less /home/analyst/lab.support.files/logstash-tutorial.log
[analyst@secOps ~]$ tail /home/analyst/lab.support.files/logstash-tutorial.log
218.30.103.62 - - [04/Jan/2015:05:28:43 +0000] "GET /blog/geekery/xvfb-firefox.html HTTP/1.1" 200 10975 "-" "Sogou web spider/
4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"
218.30.103.62 - - [04/Jan/2015:05:29:06 +0000] "GET /blog/geekery/puppet-facts-into-mcollective.html HTTP/1.1" 200 9872 "-" "S
ogou web spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"
198.46.149.143 - - [04/Jan/2015:05:29:13 +0000] "GET /blog/geekery/disabling-battery-in-ubuntu-vms.html?utm_source=feedburner&
utm_medium=feed&utm_campaign=Feed%3A+semicomplete%2Fmain+%28semicomplete.com+-+Jordan+Sissel%29 HTTP/1.1" 200 9316 "-" "Tiny T
iny RSS/1.11 (http://tt-rss.org/)"
198.46.149.143 - - [04/Jan/2015:05:29:13 +0000] "GET /blog/geekery/solving-good-or-bad-problems.html?utm_source=feedburner&utm
_medium=feed&utm_campaign=Feed%3A+semicomplete%2Fmain+%28semicomplete.com+-+Jordan+Sissel%29 HTTP/1.1" 200 10756 "-" "Tiny Tin
y RSS/1.11 (http://tt-rss.org/)"
218.30.103.62 - - [04/Jan/2015:05:29:26 +0000] "GET /blog/geekery/jquery-interface-puffer.html%20target= HTTP/1.1" 200 202 "-"
"Sogou web spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"
218.30.103.62 - - [04/Jan/2015:05:29:48 +0000] "GET /blog/geekery/ec2-reserved-vs-ondemand.html HTTP/1.1" 200 11834 "-" "Sogou
web spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"
66.249.73.135 - - [04/Jan/2015:05:30:06 +0000] "GET /blog/web/firefox-scrolling-fix.html HTTP/1.1" 200 8956 "-" "Mozilla/5.0 (
iPhone; CPU iPhone OS 6_0 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A5376e Safari/8536.25 (co
mpatible; Googlebot/2.1; +http://www.google.com/bot.html)"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /projects/xdotool/ HTTP/1.1" 200 12292 "http://www.haskell.org/haskellwiki/Xm
onad/Frequently_asked_questions" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /reset.css HTTP/1.1" 200 1015 "http://www.semicomplete.com/projects/xdotool/"
"Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /style2.css HTTP/1.1" 200 4877 "http://www.semicomplete.com/projects/xdotool/"
"Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"
[analyst@secOps ~]$
```

Step 2: Actively Following Logs.

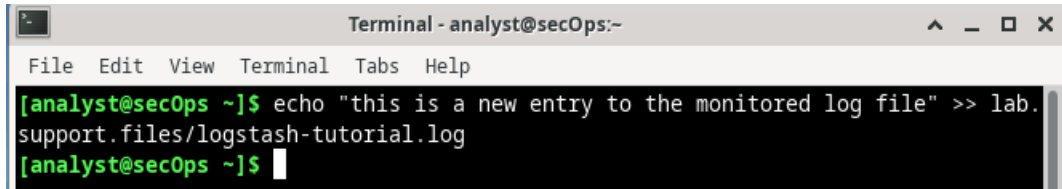
In some situations, it is desirable to monitor log files as log entries are written to the log files. For those cases, the **tail -f** command is very helpful.

- Use **tail -f** to actively monitor the contents of the **/var/log/syslog** file:

```
[analyst@secOps ~]$ sudo tail -f /home/analyst/lab.support.files/logstash-tutorial.log
218.30.103.62 - - [04/Jan/2015:05:28:43 +0000] "GET /blog/geekery/xvfb-firefox.html HTTP/1.1" 200 10975 "-" "Sogou web spider/
4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"
218.30.103.62 - - [04/Jan/2015:05:29:06 +0000] "GET /blog/geekery/puppet-facts-into-mcollective.html HTTP/1.1" 200 9872 "-" "S
ogou web spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"
198.46.149.143 - - [04/Jan/2015:05:29:13 +0000] "GET /blog/geekery/disabling-battery-in-ubuntu-vms.html?utm_source=feedburner&
utm_medium=feed&utm_campaign=Feed%3A+semicomplete%2Fmain+%28semicomplete.com+-+Jordan+Sissel%29 HTTP/1.1" 200 9316 "-" "Tiny T
iny RSS/1.11 (http://tt-rss.org/)"
198.46.149.143 - - [04/Jan/2015:05:29:13 +0000] "GET /blog/geekery/solving-good-or-bad-problems.html?utm_source=feedburner&utm
_medium=feed&utm_campaign=Feed%3A+semicomplete%2Fmain+%28semicomplete.com+-+Jordan+Sissel%29 HTTP/1.1" 200 10756 "-" "Tiny Tin
y RSS/1.11 (http://tt-rss.org/)"
218.30.103.62 - - [04/Jan/2015:05:29:26 +0000] "GET /blog/geekery/jquery-interface-puffer.html%20target= HTTP/1.1" 200 202 "-"
"Sogou web spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"
218.30.103.62 - - [04/Jan/2015:05:29:48 +0000] "GET /blog/geekery/ec2-reserved-vs-ondemand.html HTTP/1.1" 200 11834 "-" "Sogou
web spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"
66.249.73.135 - - [04/Jan/2015:05:30:06 +0000] "GET /blog/web/firefox-scrolling-fix.html HTTP/1.1" 200 8956 "-" "Mozilla/5.0 (
iPhone; CPU iPhone OS 6_0 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A5376e Safari/8536.25 (co
mpatible; Googlebot/2.1; +http://www.google.com/bot.html)"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /projects/xdotool/ HTTP/1.1" 200 12292 "http://www.haskell.org/haskellwiki/Xm
onad/Frequently_asked_questions" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /reset.css HTTP/1.1" 200 1015 "http://www.semicomplete.com/projects/xdotool/"
"Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /style2.css HTTP/1.1" 200 4877 "http://www.semicomplete.com/projects/xdotool/"
"Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"
```

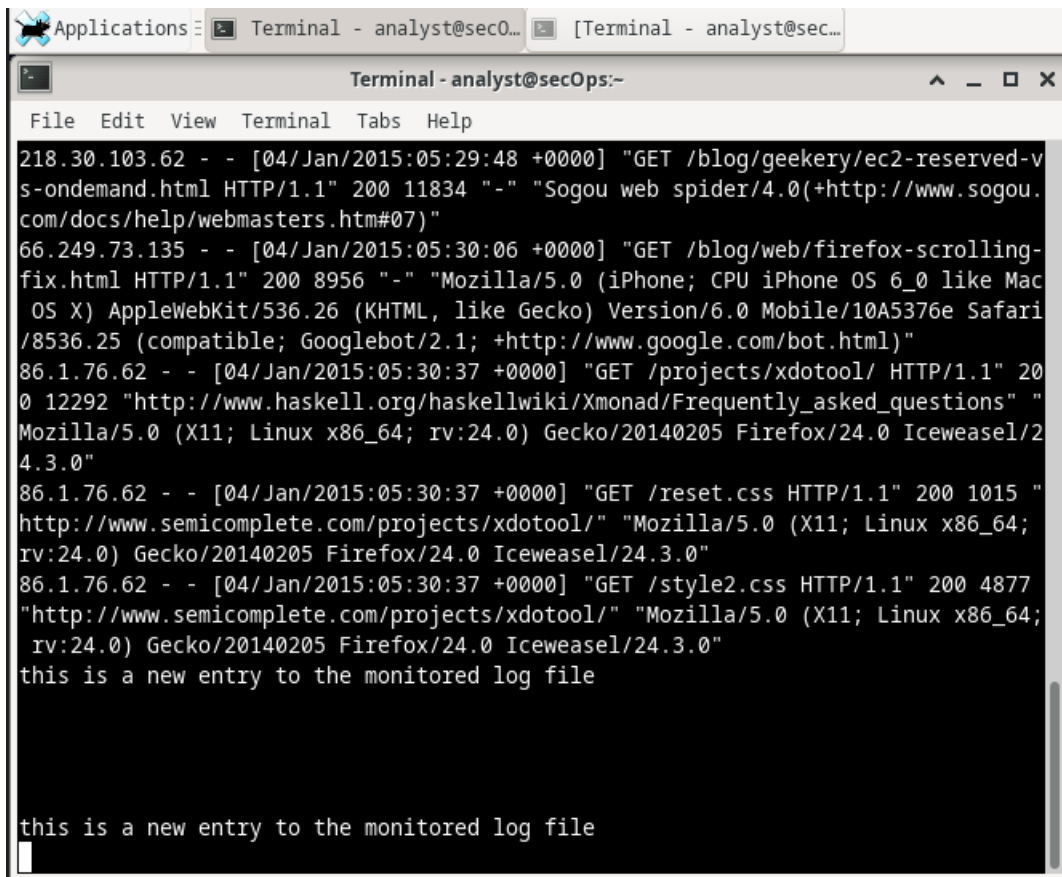

- b. To watch `tail -f` in action, open a second terminal window. Arrange your display so you can see both terminal windows. Re-size the windows so you can see them both at the same, as shown in the image below:

The terminal window on the top is running `tail -f` to monitor the `/home/analyst/lab.support.files/logstash-tutorial.log` file. Use the terminal window on the bottom to add information to the monitored file.



```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
[analyst@secOps ~]$ echo "this is a new entry to the monitored log file" >> lab.support.files/logstash-tutorial.log
[analyst@secOps ~]$
```

To make it easier to visualize, select the top terminal window (the one running `tail -f`) and press enter a few times. This will add a few lines between the current contents of the file and the new information to be added.



```
Applications: Terminal - analyst@sec0... [Terminal - analyst@sec...
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
218.30.103.62 - - [04/Jan/2015:05:29:48 +0000] "GET /blog/geekery/ec2-reserved-vs-on-demand.html HTTP/1.1" 200 11834 "-" "Sogou web spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"
66.249.73.135 - - [04/Jan/2015:05:30:06 +0000] "GET /blog/web/firefox-scrolling-fix.html HTTP/1.1" 200 8956 "-" "Mozilla/5.0 (iPhone; CPU iPhone OS 6_0 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A5376e Safari/8536.25 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /projects/xdotool/ HTTP/1.1" 200 12292 "http://www.haskell.org/haskellwiki/Xmonad/Frequently_asked_questions" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /reset.css HTTP/1.1" 200 1015 "http://www.semicomplete.com/projects/xdotool/" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /style2.css HTTP/1.1" 200 4877 "http://www.semicomplete.com/projects/xdotool/" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"
this is a new entry to the monitored log file

this is a new entry to the monitored log file
```

c. Select the bottom terminal window and enter the following command:

```
[analyst@secOps ~]$ echo "this is a new entry to the monitored log file" >> lab.support.files/logstash-tutorial.log  
[analyst@secOps ~]$
```

The command above appends the #this is a new entry to the monitored log file# message to the **/home/analyst/lab.support.files/logstash-tutorial.log** file. Because **tail -f** is monitoring the file at the moment a line is added to the file. The top window should display the new line in real-time.

- d. Press CTRL + C to stop the execution of **tail -f** and return to the shell prompt.
- e. Close one of the two terminal windows.

Part 2: Syslog

Syslog is a centralized logging system that collects operating system and application logs.

Root Access:

The cat command had to be executed as root because syslog files contain sensitive system information and are protected to prevent unauthorized access.

Log Rotation:

Older syslog files are rotated and renamed (syslog.1, syslog.2, etc.) to prevent files from becoming too large.

Time Synchronization:

Accurate system time is essential for log analysis, troubleshooting, and security investigations. Incorrect timestamps can lead to misinterpretation of events.

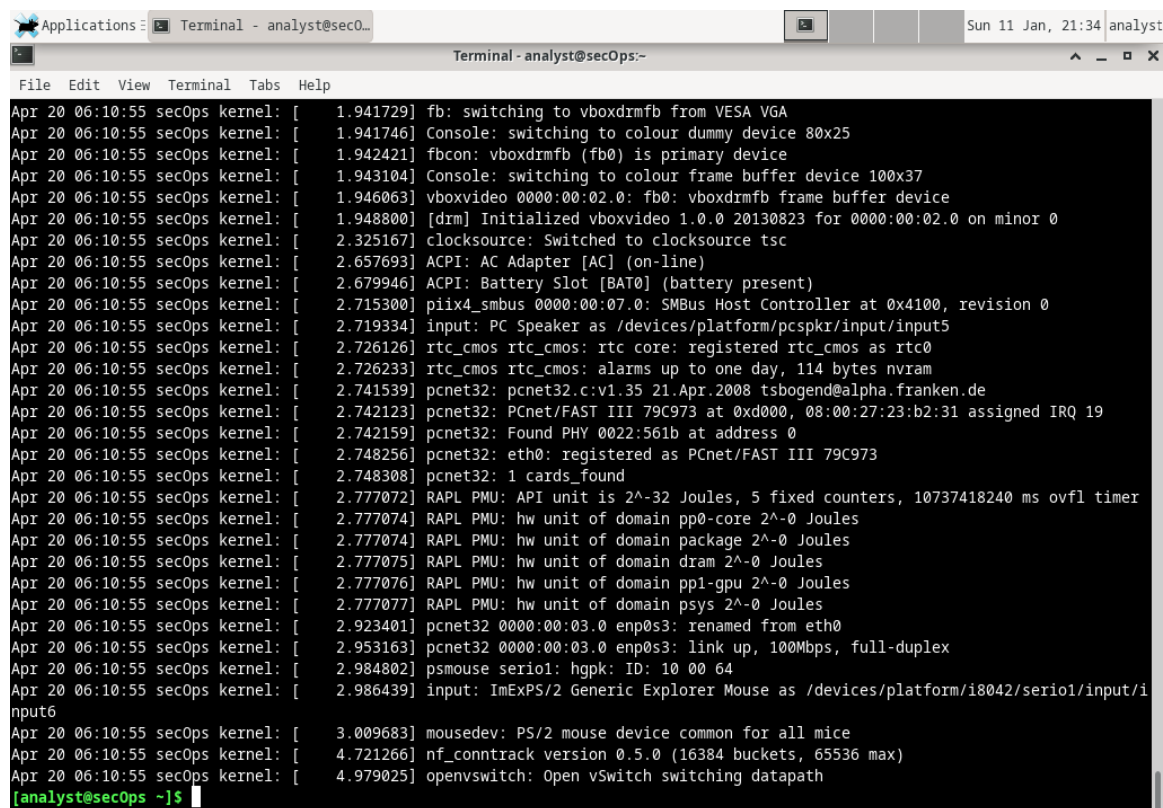
Because of their importance, it is common practice to concentrate log files in one monitoring computer. **Syslog** is a system designed to allow devices to send their log files to a centralized server, known as **syslog** server. Clients

communicate to a syslog server using the **syslog** protocol. **Syslog** is commonly deployed and supports practically all computer platforms.

The CyberOps Workstation VM generates operating system level log files and hands them over to **syslog**.

- a. Use the **cat** command as **root** to list the contents of the **/var/log/syslog.1** file. This file holds the log entries that are generated by the CyberOps Workstation VM operating system and sent to the **syslog** service.

```
analyst@secOps ~$ sudo cat /var/log/syslog.1  
[sudo] password for analyst:
```



```
Applications: Terminal - analyst@sec0... Sun 11 Jan, 21:34 analyst  
Terminal - analyst@secOps:~  
File Edit View Terminal Tabs Help  
Apr 20 06:10:55 secOps kernel: [ 1.941729] fb: switching to vboxdrmfb from VESA VGA  
Apr 20 06:10:55 secOps kernel: [ 1.941746] Console: switching to colour dummy device 80x25  
Apr 20 06:10:55 secOps kernel: [ 1.942421] fbcon: vboxdrmfb (fb0) is primary device  
Apr 20 06:10:55 secOps kernel: [ 1.943104] Console: switching to colour frame buffer device 100x37  
Apr 20 06:10:55 secOps kernel: [ 1.946063] vboxvideo 0000:00:02.0: fb0: vboxdrmfb frame buffer device  
Apr 20 06:10:55 secOps kernel: [ 1.948800] [drm] Initialized vboxvideo 1.0.0 20130823 for 0000:00:02.0 on minor 0  
Apr 20 06:10:55 secOps kernel: [ 2.325167] clocksource: Switched to clocksource tsc  
Apr 20 06:10:55 secOps kernel: [ 2.657693] ACPI: AC Adapter [AC] (on-line)  
Apr 20 06:10:55 secOps kernel: [ 2.679946] ACPI: Battery Slot [BAT0] (battery present)  
Apr 20 06:10:55 secOps kernel: [ 2.715300] piix4_smbus 0000:00:07.0: SMBus Host Controller at 0x4100, revision 0  
Apr 20 06:10:55 secOps kernel: [ 2.719334] input: PC Speaker as /devices/platform/pcspkr/input/input5  
Apr 20 06:10:55 secOps kernel: [ 2.726126] rtc_cmos rtc_cmos: rtc core: registered rtc_cmos as rtc0  
Apr 20 06:10:55 secOps kernel: [ 2.726233] rtc_cmos rtc_cmos: alarms up to one day, 114 bytes nvram  
Apr 20 06:10:55 secOps kernel: [ 2.741539] pcnet32: pcnet32.c:v1.35 21.Apr.2008 tsbogend@alpha.franken.de  
Apr 20 06:10:55 secOps kernel: [ 2.742123] pcnet32: PCnet/FAST III 79C973 at 0xd000, 08:00:27:23:b2:31 assigned IRQ 19  
Apr 20 06:10:55 secOps kernel: [ 2.742159] pcnet32: Found PHY 0022:561b at address 0  
Apr 20 06:10:55 secOps kernel: [ 2.748256] pcnet32: eth0: registered as PCnet/FAST III 79C973  
Apr 20 06:10:55 secOps kernel: [ 2.748308] pcnet32: 1 cards_found  
Apr 20 06:10:55 secOps kernel: [ 2.777072] RAPL PMU: API unit is 2^-32 Joules, 5 fixed counters, 10737418240 ms ovfl timer  
Apr 20 06:10:55 secOps kernel: [ 2.777074] RAPL PMU: hw unit of domain pp0-core 2^-0 Joules  
Apr 20 06:10:55 secOps kernel: [ 2.777074] RAPL PMU: hw unit of domain package 2^-0 Joules  
Apr 20 06:10:55 secOps kernel: [ 2.777075] RAPL PMU: hw unit of domain dram 2^-0 Joules  
Apr 20 06:10:55 secOps kernel: [ 2.777076] RAPL PMU: hw unit of domain ppl-gpu 2^-0 Joules  
Apr 20 06:10:55 secOps kernel: [ 2.777077] RAPL PMU: hw unit of domain psys 2^-0 Joules  
Apr 20 06:10:55 secOps kernel: [ 2.923401] pcnet32 0000:00:03.0 enp0s3: renamed from eth0  
Apr 20 06:10:55 secOps kernel: [ 2.953163] pcnet32 0000:00:03.0 enp0s3: link up, 100Mbps, full-duplex  
Apr 20 06:10:55 secOps kernel: [ 2.984802] psmouse serio1: hgpk: ID: 10 00 64  
Apr 20 06:10:55 secOps kernel: [ 2.986439] input: ImEXPS/2 Generic Explorer Mouse as /devices/platform/i8042/serio1/input/input6  
Apr 20 06:10:55 secOps kernel: [ 3.009683] mousedev: PS/2 mouse device common for all mice  
Apr 20 06:10:55 secOps kernel: [ 4.721266] nf_conntrack version 0.5.0 (16384 buckets, 65536 max)  
Apr 20 06:10:55 secOps kernel: [ 4.979025] openvswitch: Open vSwitch switching datapath  
[analyst@secOps ~]$
```

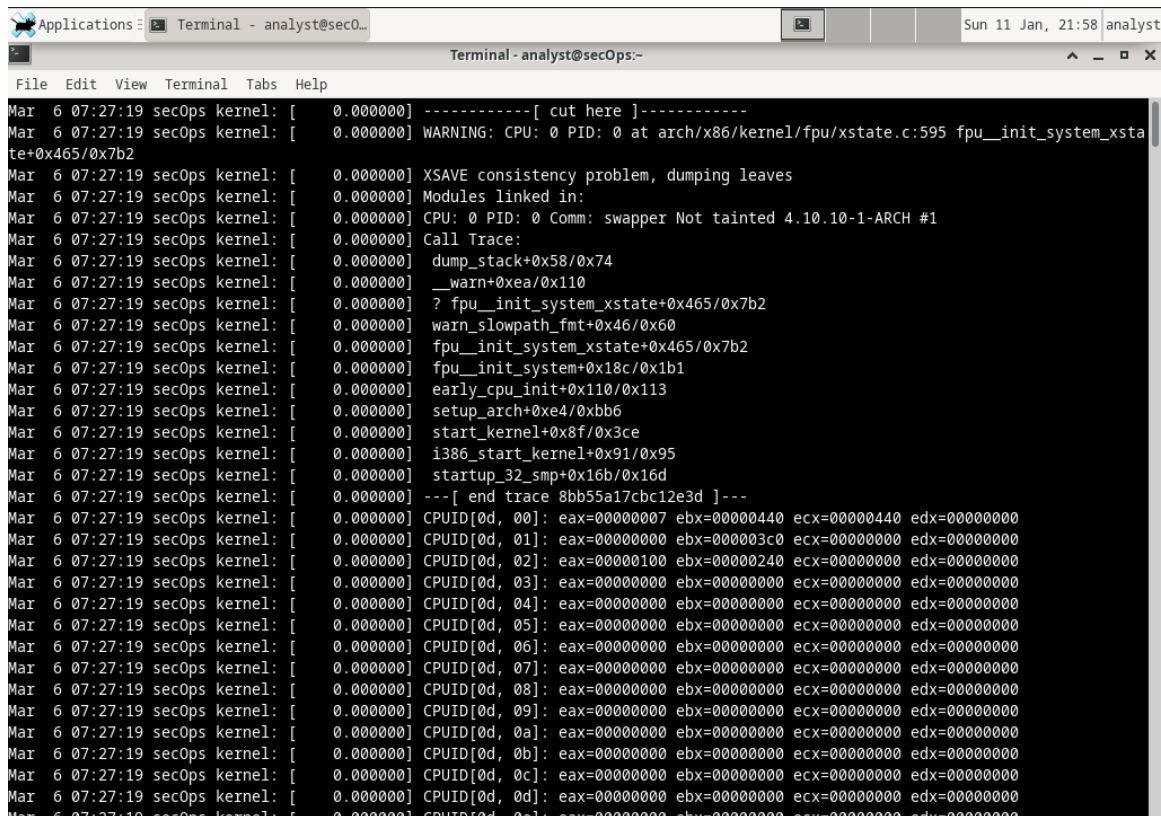
- b. Notice that the **/var/log/syslog** file only stores the most recent log entries. To keep the syslog file small, the operating system periodically rotates the log files, renaming older log files as **syslog.1**, **syslog.2**, and so on.

Use the **cat** command to list older **syslog** files:

analyst@secOps ~\$ sudo cat /var/log/syslog.2

analyst@secOps ~\$ sudo cat /var/log/syslog.3

analyst@secOps ~\$ sudo cat /var/log/syslog.4



```
Applications: Terminal - analyst@secOps~ Sun 11 Jan, 21:58 analyst
Terminal - analyst@secOps~
File Edit View Terminal Tabs Help
Mar 6 07:27:19 secOps kernel: [ 0.000000] -----[ cut here ]-----
Mar 6 07:27:19 secOps kernel: [ 0.000000] WARNING: CPU: 0 PID: 0 at arch/x86/kernel/fpu/xstate.c:595 fpu__init_system_xstate+0x465/0x7b2
Mar 6 07:27:19 secOps kernel: [ 0.000000] XSAVE consistency problem, dumping leaves
Mar 6 07:27:19 secOps kernel: [ 0.000000] Modules linked in:
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPU: 0 PID: 0 Comm: swapper Not tainted 4.10.10-1-ARCH #1
Mar 6 07:27:19 secOps kernel: [ 0.000000] Call Trace:
Mar 6 07:27:19 secOps kernel: [ 0.000000] dump_stack+0x58/0x74
Mar 6 07:27:19 secOps kernel: [ 0.000000] __warn+0xea/0x110
Mar 6 07:27:19 secOps kernel: [ 0.000000] ? fpu__init_system_xstate+0x465/0x7b2
Mar 6 07:27:19 secOps kernel: [ 0.000000] warn_slowpath_fmt+0x46/0x60
Mar 6 07:27:19 secOps kernel: [ 0.000000] fpu__init_system_xstate+0x465/0x7b2
Mar 6 07:27:19 secOps kernel: [ 0.000000] fpu__init_system+0x18c/0x1b1
Mar 6 07:27:19 secOps kernel: [ 0.000000] early_cpu_init+0x110/0x113
Mar 6 07:27:19 secOps kernel: [ 0.000000] setup_arch+0xe4/0xbb6
Mar 6 07:27:19 secOps kernel: [ 0.000000] start_kernel+0x8f/0x3ce
Mar 6 07:27:19 secOps kernel: [ 0.000000] i386_start_kernel+0x91/0x95
Mar 6 07:27:19 secOps kernel: [ 0.000000] startup_32_smp+0x16b/0x16d
Mar 6 07:27:19 secOps kernel: [ 0.000000] ---[ end trace 8bb55a17cbc12e3d ]---
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 00]: eax=00000007 ebx=00000440 ecx=00000440 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 01]: eax=00000000 ebx=000003c0 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 02]: eax=00000100 ebx=00000240 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 03]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 04]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 05]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 06]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 07]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 08]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 09]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 0a]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 0b]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 0c]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 0d]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
Mar 6 07:27:19 secOps kernel: [ 0.000000] CPUID[0d, 0e]: eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
```

Part 3: Journald

Journald manages logs using binary files and centralizes log data from multiple sources.

Journalctl:

The journalctl command allows filtering by boot session, service, kernel messages, and time range.

Advantages:

- Structured logging
- Advanced filtering
- Tamper resistance

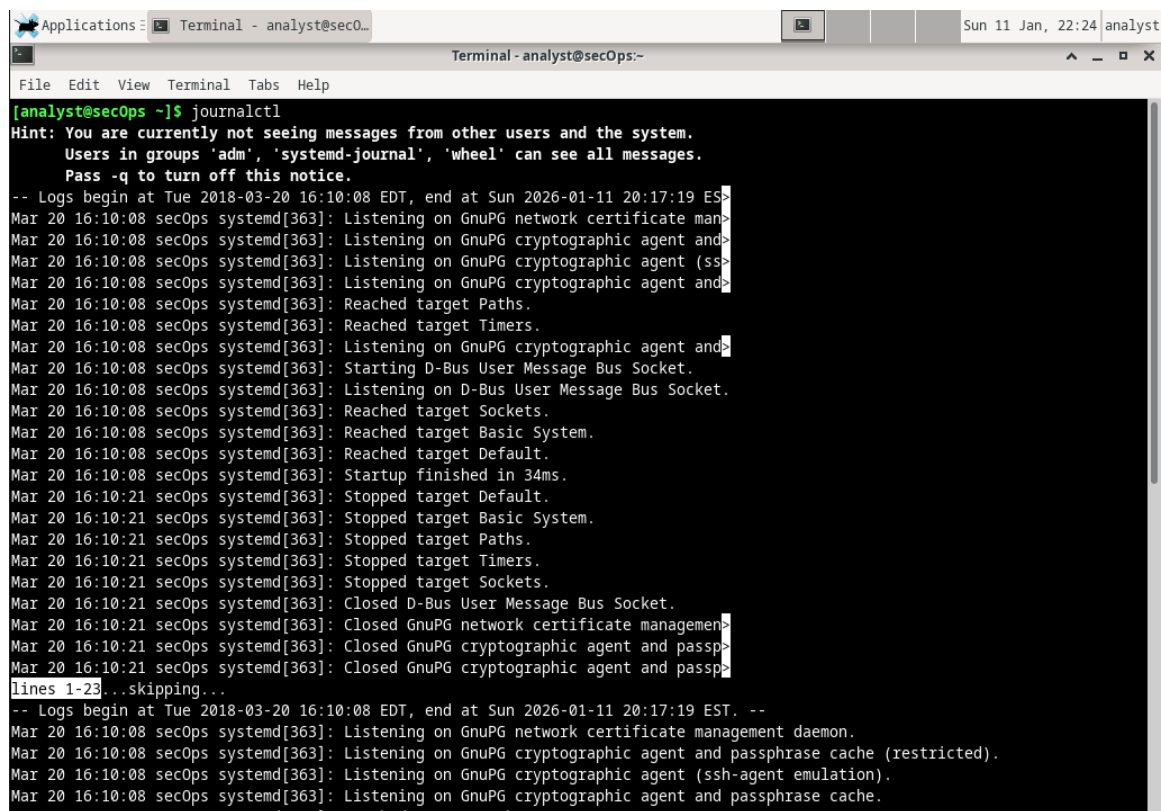
Disadvantages:

- Binary format is not human-readable without `journalctl`
- Slightly higher resource usage

In the context of this lab, the most evident feature of the **journal** system daemon is the use of append-only binary files serving as its **log files**.

Step 1: Running `journalctl` with no options.

- a. To look at the **journald** logs, use the **journalctl** command.
The **journalctl** tool interprets and displays the log entries previously stored in the **journal** binary log files.



```
[analyst@secOps ~]$ journalctl
Hint: You are currently not seeing messages from other users and the system.
      Users in groups 'adm', 'systemd-journal', 'wheel' can see all messages.
      Pass -q to turn off this notice.
-- Logs begin at Tue 2018-03-20 16:10:08 EDT, end at Sun 2026-01-11 20:17:19 EST. --
Mar 20 16:10:08 secOps systemd[363]: Listening on GnuPG network certificate manag
Mar 20 16:10:08 secOps systemd[363]: Listening on GnuPG cryptographic agent and
Mar 20 16:10:08 secOps systemd[363]: Listening on GnuPG cryptographic agent (ss
Mar 20 16:10:08 secOps systemd[363]: Listening on GnuPG cryptographic agent and
Mar 20 16:10:08 secOps systemd[363]: Reached target Paths.
Mar 20 16:10:08 secOps systemd[363]: Reached target Timers.
Mar 20 16:10:08 secOps systemd[363]: Listening on GnuPG cryptographic agent and
Mar 20 16:10:08 secOps systemd[363]: Starting D-Bus User Message Bus Socket.
Mar 20 16:10:08 secOps systemd[363]: Listening on D-Bus User Message Bus Socket.
Mar 20 16:10:08 secOps systemd[363]: Reached target Sockets.
Mar 20 16:10:08 secOps systemd[363]: Reached target Basic System.
Mar 20 16:10:08 secOps systemd[363]: Reached target Default.
Mar 20 16:10:08 secOps systemd[363]: Startup finished in 34ms.
Mar 20 16:10:21 secOps systemd[363]: Stopped target Default.
Mar 20 16:10:21 secOps systemd[363]: Stopped target Basic System.
Mar 20 16:10:21 secOps systemd[363]: Stopped target Paths.
Mar 20 16:10:21 secOps systemd[363]: Stopped target Timers.
Mar 20 16:10:21 secOps systemd[363]: Stopped target Sockets.
Mar 20 16:10:21 secOps systemd[363]: Closed D-Bus User Message Bus Socket.
Mar 20 16:10:21 secOps systemd[363]: Closed GnuPG network certificate managemen
Mar 20 16:10:21 secOps systemd[363]: Closed GnuPG cryptographic agent and passp
Mar 20 16:10:21 secOps systemd[363]: Closed GnuPG cryptographic agent and passp
lines 1-23... skipping...
-- Logs begin at Tue 2018-03-20 16:10:08 EDT, end at Sun 2026-01-11 20:17:19 EST. --
Mar 20 16:10:08 secOps systemd[363]: Listening on GnuPG network certificate management daemon.
Mar 20 16:10:08 secOps systemd[363]: Listening on GnuPG cryptographic agent and passphrase cache (restricted).
Mar 20 16:10:08 secOps systemd[363]: Listening on GnuPG cryptographic agent (ssh-agent emulation).
Mar 20 16:10:08 secOps systemd[363]: Listening on GnuPG cryptographic agent and passphrase cache.
```

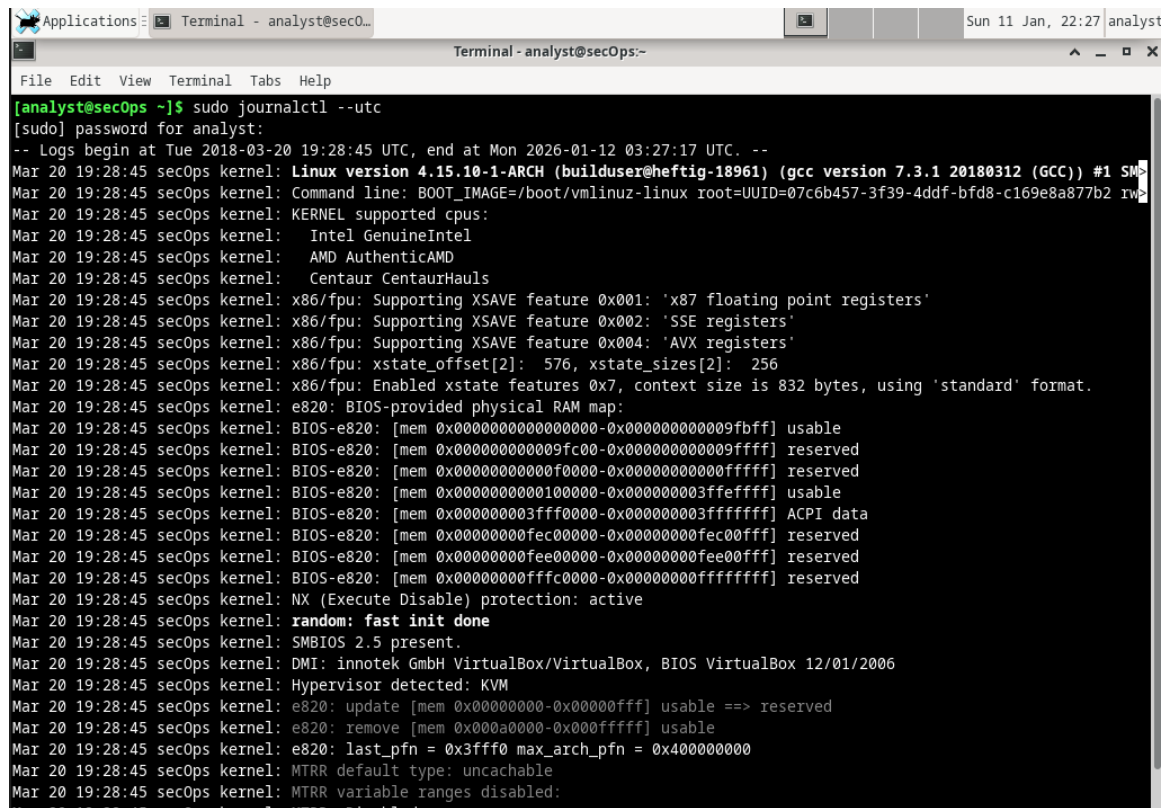
Note: Running `journalctl` as root will display more detailed information.

- b. Use CTRL+C to exit the display.

Step 2: Journalctl and a few options.

Part of the power of using journalctl lies in its options. For the following commands, use CTRL+C to exit the display.

- a. Use **journalctl --utc** to display all timestamps in UTC time:



```
[analyst@secOps ~]$ sudo journalctl --utc
[sudo] password for analyst:
-- Logs begin at Tue 2018-03-20 19:28:45 UTC, end at Mon 2026-01-12 03:27:17 UTC. --
Mar 20 19:28:45 secOps kernel: Linux version 4.15.10-1-ARCH (builduser@heftig-18961) (gcc version 7.3.1 20180312 (GCC)) #1 SMP
Mar 20 19:28:45 secOps kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-linux root=UUID=07c6b457-3f39-4ddf-bfd8-c169e8a877b2 rw
Mar 20 19:28:45 secOps kernel: KERNEL supported cpus:
Mar 20 19:28:45 secOps kernel: Intel GenuineIntel
Mar 20 19:28:45 secOps kernel: AMD AuthenticAMD
Mar 20 19:28:45 secOps kernel: Centaur CentaurHauls
Mar 20 19:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
Mar 20 19:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
Mar 20 19:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
Mar 20 19:28:45 secOps kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
Mar 20 19:28:45 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
Mar 20 19:28:45 secOps kernel: e820: BIOS-provided physical RAM map:
Mar 20 19:28:45 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbff] usable
Mar 20 19:28:45 secOps kernel: BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
Mar 20 19:28:45 secOps kernel: BIOS-e820: [mem 0x000000000000f0000-0x000000000000ffffff] reserved
Mar 20 19:28:45 secOps kernel: BIOS-e820: [mem 0x00000000000100000-0x000000000003ffff] usable
Mar 20 19:28:45 secOps kernel: BIOS-e820: [mem 0x000000000003ffff000-0x000000000003ffffff] ACPI data
Mar 20 19:28:45 secOps kernel: BIOS-e820: [mem 0x0000000000fec0000-0x0000000000fec0fff] reserved
Mar 20 19:28:45 secOps kernel: BIOS-e820: [mem 0x0000000000fee0000-0x0000000000fee0fff] reserved
Mar 20 19:28:45 secOps kernel: BIOS-e820: [mem 0x0000000000fffc000-0x0000000000ffffff] reserved
Mar 20 19:28:45 secOps kernel: NX (Execute Disable) protection: active
Mar 20 19:28:45 secOps kernel: random: fast init done
Mar 20 19:28:45 secOps kernel: SMBIOS 2.5 present.
Mar 20 19:28:45 secOps kernel: DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Mar 20 19:28:45 secOps kernel: Hypervisor detected: KVM
Mar 20 19:28:45 secOps kernel: e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
Mar 20 19:28:45 secOps kernel: e820: remove [mem 0x0000a0000-0x0000ffff] usable
Mar 20 19:28:45 secOps kernel: e820: last_pfn = 0x3ffff max_arch_pfn = 0x400000000
Mar 20 19:28:45 secOps kernel: MTRR default type: uncachable
Mar 20 19:28:45 secOps kernel: MTRR variable ranges disabled:
Mar 20 19:28:45 secOps kernel: MTRR: Disabled
```

- b. Use **journalctl -b** to display log entries recorded during the last boot:


```
Applications: Terminal - analyst@sec0... Sun 11 Jan, 22:29 analyst
Terminal - analyst@secOps:-
File Edit View Terminal Tabs Help

[analyst@secOps ~]$ sudo journalctl -b
-- Logs begin at Tue 2018-03-20 15:28:45 EDT, end at Sun 2026-01-11 22:29:04 EST. --
Jan 11 18:17:40 secOps kernel: Linux version 5.6.3-arch1-1 (linux@archlinux) (gcc version 9.3.0 (Arch Linux 9.3.0-1)) #1 SMP >
Jan 11 18:17:40 secOps kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-linux root=UUID=07c6b457-3f39-4ddf-bfd8-c169e8a877b2 rw
Jan 11 18:17:40 secOps kernel: KERNEL supported cpus:
Jan 11 18:17:40 secOps kernel: Intel GenuineIntel
Jan 11 18:17:40 secOps kernel: AMD AuthenticAMD
Jan 11 18:17:40 secOps kernel: Hygon HygonGenuine
Jan 11 18:17:40 secOps kernel: Centaur CentaurHauls
Jan 11 18:17:40 secOps kernel: zhaoxin Shanghai
Jan 11 18:17:40 secOps kernel: [Firmware Bug]: TSC doesn't count with P0 frequency!
Jan 11 18:17:40 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
Jan 11 18:17:40 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
Jan 11 18:17:40 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
Jan 11 18:17:40 secOps kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
Jan 11 18:17:40 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
Jan 11 18:17:40 secOps kernel: BIOS-provided physical RAM map:
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbf] usable
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x0000000000009fc0-0x0000000000009fff] reserved
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x000000000000f000-0x000000000000ffff] reserved
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x0000000000100000-0x00000000003fffff] usable
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x00000000003fff0000-0x00000000003fffffff] ACPI data
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
Jan 11 18:17:40 secOps kernel: NX (Execute Disable) protection: active
Jan 11 18:17:40 secOps kernel: SMBIOS 2.5 present.
Jan 11 18:17:40 secOps kernel: DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Jan 11 18:17:40 secOps kernel: Hypervisor detected: KVM
Jan 11 18:17:40 secOps kernel: kvm-clock: Using msrs 4b564d01 and 4b564d00
Jan 11 18:17:40 secOps kernel: kvm-clock: cpu 0, msr 13001001, primary cpu clock
Jan 11 18:17:40 secOps kernel: kvm-clock: using sched offset of 11692516626 cycles
Jan 11 18:17:40 secOps kernel: clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles: 0x1cd42e4dffb, max_idle_ns: 881598
```

- c. Use **journalctl** to specify the service and timeframe for log entries. The command below shows all **nginx** service logs recorded today:

```
[analyst@secOps ~]$ sudo journalctl -u nginx.service --since today
-- Logs begin at Tue 2018-03-20 15:28:45 EDT, end at Sun 2026-01-11 22:33:53 EST. --
-- No entries --
```

- d. Use the **-k** switch to display only messages generated by the kernel:

```

[analyst@secOps ~]$ sudo journalctl -k
[sudo] password for analyst:
Sorry, try again.
[sudo] password for analyst:
-- Logs begin at Tue 2018-03-20 15:28:45 EDT, end at Sun 2026-01-11 22:37:28 EST. --
Jan 11 18:17:40 secOps kernel: Linux version 5.6.3-arch1-1 (linux@archlinux) (gcc version 9.3.0 (Arch Linux 9.3.0-1)) #1 SMP >
Jan 11 18:17:40 secOps kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-linux root=UUID=07c6b457-3f39-4ddf-bfd8-c169e8a877b2 rw
Jan 11 18:17:40 secOps kernel: KERNEL supported cpus:
Jan 11 18:17:40 secOps kernel: Intel GenuineIntel
Jan 11 18:17:40 secOps kernel: AMD AuthenticAMD
Jan 11 18:17:40 secOps kernel: Hygon HygonGenuine
Jan 11 18:17:40 secOps kernel: Centaur CentaurHauls
Jan 11 18:17:40 secOps kernel: zhaoxin Shanghai
Jan 11 18:17:40 secOps kernel: [Firmware Bug]: TSC doesn't count with P0 frequency!
Jan 11 18:17:40 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
Jan 11 18:17:40 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
Jan 11 18:17:40 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
Jan 11 18:17:40 secOps kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
Jan 11 18:17:40 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
Jan 11 18:17:40 secOps kernel: BIOS-provided physical RAM map:
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbff] usable
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x000000000000f0000-0x000000000000ffffff] reserved
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x0000000000100000-0x00000000003ffff] usable
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x00000000003ffff0000-0x00000000003ffffff] ACPI data
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
Jan 11 18:17:40 secOps kernel: BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
Jan 11 18:17:40 secOps kernel: NX (Execute Disable) protection: active
Jan 11 18:17:40 secOps kernel: SMBIOS 2.5 present.
Jan 11 18:17:40 secOps kernel: DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Jan 11 18:17:40 secOps kernel: Hypervisor detected: KVM
Jan 11 18:17:40 secOps kernel: kvm-clock: Using msrs 4b564d01 and 4b564d00

```

- e. Similar to `tail -f` described above, use the `-f` switch to actively follow the logs as they are being written:

```

[analyst@secOps ~]$ sudo journalctl -f
-- Logs begin at Tue 2018-03-20 15:28:45 EDT. --
Jan 11 22:38:53 secOps audit[1410]: USER_END pid=1410 uid=0 auid=1000 ses=2 msg='op=PAM:session_close grantors=pam_limits,pam_unix,pam_permit acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/2 res=success'
Jan 11 22:38:53 secOps audit[1410]: CRED_DISP pid=1410 uid=0 auid=1000 ses=2 msg='op=PAM:setcred grantors=pam_unix,pam_permit acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/2 res=success'
Jan 11 22:39:15 secOps audit[1423]: USER_ACCT pid=1423 uid=1000 auid=1000 ses=2 msg='op=PAM:accounting grantors=pam_unix,pam_permit,pam_time acct="analyst" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/2 res=success'
Jan 11 22:39:15 secOps sudo[1423]: analyst : TTY=pts/2 ; PWD=/home/analyst ; USER=root ; COMMAND=/usr/bin/journalctl -f
Jan 11 22:39:15 secOps kernel: audit: type=1101 audit(1768189155.039:120): pid=1423 uid=1000 auid=1000 ses=2 msg='op=PAM:accounting grantors=pam_unix,pam_permit,pam_time acct="analyst" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/2 res=success'
Jan 11 22:39:15 secOps audit[1423]: CRED_REFR pid=1423 uid=0 auid=1000 ses=2 msg='op=PAM:setcred grantors=pam_unix,pam_permit,pam_env acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/2 res=success'
Jan 11 22:39:15 secOps sudo[1423]: pam_unix(sudo:session): session opened for user root by (uid=0)
Jan 11 22:39:15 secOps audit[1423]: USER_START pid=1423 uid=0 auid=1000 ses=2 msg='op=PAM:session_open grantors=pam_limits,pam_unix,pam_permit acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/2 res=success'
Jan 11 22:39:15 secOps kernel: audit: type=1110 audit(1768189155.046:121): pid=1423 uid=0 auid=1000 ses=2 msg='op=PAM:setcred grantors=pam_unix,pam_permit,pam_env acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/2 res=success'
Jan 11 22:39:15 secOps kernel: audit: type=1105 audit(1768189155.046:122): pid=1423 uid=0 auid=1000 ses=2 msg='op=PAM:session_open grantors=pam_limits,pam_unix,pam_permit acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/2 res=success'

```

Reflection

Syslog is simple, widely supported, and uses plain-text logs, making it easy to integrate with other tools. However, it lacks advanced filtering. Journald provides powerful querying and structured logging but relies on binary files, which require specific tools to read. Both systems are effective depending on use case.