Introduction

Linux is a free, open-source operating system, which means that the source code is available for anyone who wants to modify, repackage, and distribute it. Although it is available for free, it is a robust, high-performance system that is quite popular in certain areas of the computing community, particularly for infrastructure and web servers. For forensic investigators, this means that you are just as likely to encounter Linux-based evidence while examining servers and network infrastructure as you are likely to encounter Windows-based evidence when dealing with desktops and laptops. It is also worth noting that Linux is the underlying operating system used for Android, the most common mobile operating system worldwide with more than 72 percent of the market share. Although mobile forensics is outside the scope of this particular lab, many of the skills you will learn can also be applied to Android devices.

One of the most important components of Linux is the interactive interface, or shell. The shell is what takes the commands entered into the keyboard and delivers them to the operating system. There are two main shells for Linux: a graphical user interface (GUI) and a command line interface (CLI). Although the GUI may initially feel more familiar to users with a Windows or Mac OS background, the command line interface offers tremendous flexibility and power for interacting with the operating system. As a forensics investigator, understanding common Linux commands is essential to performing forensic analysis on a Linux machine.

In this lab, you will explore the Linux file system and practice some basic commands, which you will use to retrieve log files – a common source of forensic evidence. In Section 2, you will shift your attention to an existing Linux drive image and conduct forensic analysis on the file system.

Lab Overview

SECTION 1 of this lab has three parts, which should be completed in the order specified.

1. In the first part of the lab, you will explore the Linux filesystem using the GUI and CLI.

2. In the second part of the lab, you will learn some Linux shell commands that are commonly used in forensic investigations.

3. In the third part of the lab, you will retrieve information from Linux log files that can be used in a forensic investigation.

SECTION 2 of this lab allows you to apply what you learned in **SECTION 1** with less guidance and different deliverables, as well as some expanded tasks and alternative methods. You will explore a Linux drive image for evidence of a potential security breach.

Finally, you will explore the virtual environment on your own in **SECTION 3** of this lab to answer a set of questions and challenges that allow you to use the skills you learned in the lab to conduct independent, unguided work - similar to what you will encounter in a real-world situation.

Learning Objectives

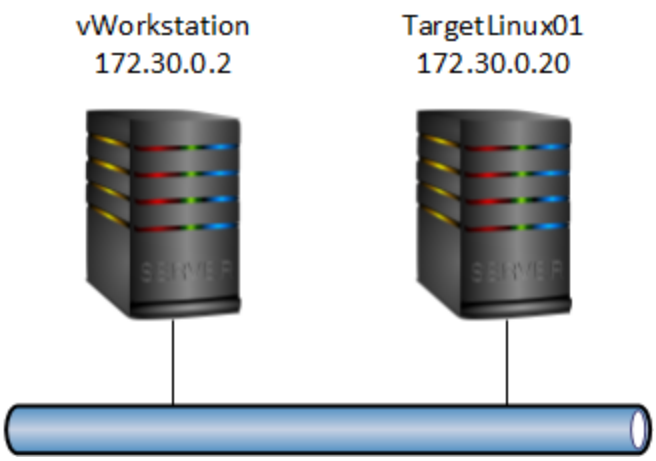
Upon completing this lab, you will be able to:

1. Identify key directories in the Linux file system and their purpose.
2. Use basic shell commands for forensic investigations.
3. Use the terminal to retrieve log files on a live Linux system.
4. Retrieve evidence from a Linux disk image using forensic analysis tools.
5. Identify the locations of critical Linux log files on a Linux disk image.

Topology

This lab contains the following virtual machines. Please refer to the network topology diagram below.

- vWorkstation (Windows: Server 2019)
- TargetLinux01 (Linux: Ubuntu)



Tools and Software

The following software and/or utilities are required to complete this lab. Students are encouraged to explore the Internet to learn more about the products and tools used in this lab.

- Files
- Terminal
- Paraben's E3

Deliverables

Upon completion of this lab, you are required to provide the following deliverables to your instructor:

SECTION 1

1. Lab Report file, including screen captures of the following:

- Contents of the /bin directory
- Contents of the /etc directory
- Contents of the /var directory
- Contents of the /proc directory

- Results of the dmesg command
- Results of the fsck command
- Results of the history command
- Running processes
- Results of the file command
- Records in the kern.log file
- Records in the auth.log file

2. Any additional information as directed by the lab:

- None

SECTION 2

1. Lab Report file, including screen captures of the following:

- None

2. Any additional information as directed by the lab:

- Document the name(s) of the user(s) that attempted to login, the number of attempts detected, the date/time range of the attempts, the source IP address for the login attempts, and the port.
- Document the date and time the most recent successful login for the user(s) that you identified in step 15.
- Document the applications that were installed using apt-get, then use the Internet to identify the ones that might be considered suspicious.
- Document when the USB storage device was connected and its serial number.

SECTION 3

1. Lab Report file, including screen captures of the following:

- Contents of the printer log file.
- Record of the dd command in the Text View

2. Any additional information as directed by the lab:

- None

Section 1: Hands-On Demonstration

Note: In this section of the lab, you will follow a step-by-step walk-through of the objectives for this lab to produce the expected deliverables.

1. **Review** the **Tutorial**.

   Frequently performed tasks, such as making screen captures and downloading your Lab Report, are explained in the Cloud Lab Tutorial. The Cloud Lab Tutorial is available from the User menu in the upper-right corner of the Student Dashboard. You should review these tasks before starting the lab.

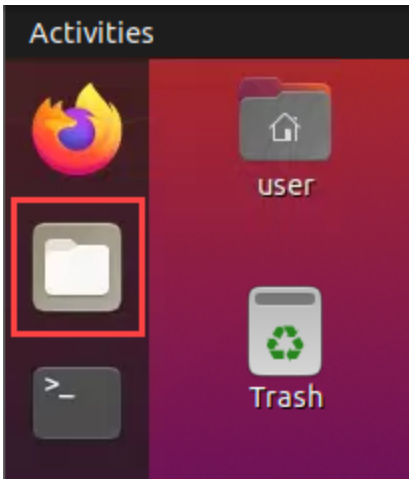
2. **Proceed** with **Part 1**.

Part 1: Explore a Live Linux System

Note: In this part of lab, you will learn to navigate a Linux system using both the graphical user interface (GUI) and command line interface (CLI) and identify important directories. You will begin by exploring the Linux system using the familiar Graphical User Interface (GUI). User credentials are provided below for reference:

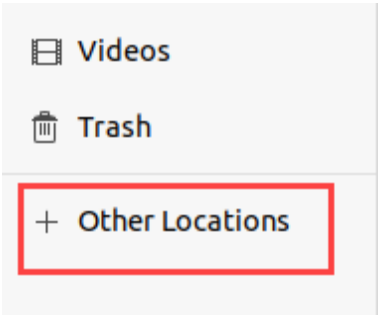
Username: **`user`**
Password: **`password`**

1. From the TargetLinux01 sidebar, **click** the **Files icon** to open the Files application.



Files icon

2. In the File window, **click Other Locations** to display locations outside the current user's Home directory.



Other Locations

3. On the Other Locations page, **click Computer** to navigate to the root directory (represented as /).

Computer

Note: Everything in a Linux system begins in the root directory. All the folders, hard drives, USB drivers – everything rolls up to the root directory. For Windows users, it may be useful to think of the root directory as similar to the C: drive (although the limits of this comparison quickly become clear when you begin to consider other disks).

As you look over the root directory, you may notice a directory within it that is explicitly named *root*. This directory is actually the root user folder, which contains files that are specific to the root user. In Linux systems, the root user is equivalent to the Administrator user in Windows systems.

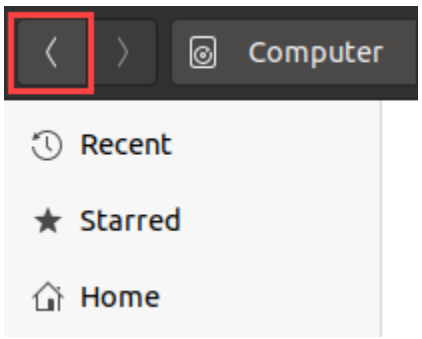
4. In the root directory, **double-click** the **bin directory** to open it.

Bin directory

Note: The /bin directory contains programs that are essential for the system to boot and run. These programs are stored in a binary format. In other words, they are not in text format, which means you cannot open and read the content of these programs. The advantage of this format is that a computer can read and execute these programs very quickly.

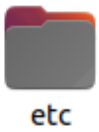
For a forensic investigator, the /bin directory is an important location to review when attempting to identify evidence of specific programs being installed or tampered with on the target system, including malware.

5. On the Files toolbar, **click** the **Back button** to return to the root directory.



Back button

6. In the root directory, **double-click** the **etc directory** to open it.



Etc directory

Note: The /etc folder contains all the system configuration files in Linux. You should think of this folder as the nerve center for your Linux machine. The /etc folder contains system-wide configuration files, while user configuration files are stored in each user's home directory.

Because configuration files determine how an application behaves, this makes them a valuable target for hackers. For forensic investigators, the /etc directory can be a valuable source of evidence that an application has been tampered with.

7. On the Files toolbar, **click** the **Back button** to return to the root directory.

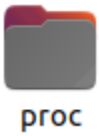
8. In the root directory, **double-click** the **var directory** to open it.

Var directory

Note: The /var directory is the editable version of files found in the /usr directory. The /usr directory contains the read-only applications and files used by users. A key component to the /var directory is log files, which you will explore later in this lab. It is worth noting that the contents of this directory change each time the system is booted up, which means this directory will only be useful to forensic investigators on a live system.

9. On the Files toolbar, **click** the **Back button** to return to the root directory.

10. In the /root folder, **double-click** the **proc directory** to open it.

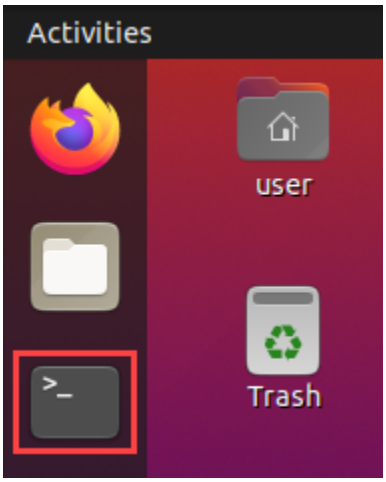


Proc directory

Note: The /proc directory contains files that represent system and process information. These can yield valuable information when conducting an investigation to reveal the state of the system at a specific time or to reveal that a parameter was altered by malware or an intruder. Unlike other directories, the contents of the /proc directory are stored in the system memory, rather than the hard disk.

11. **Close** the **Files application**.

12. From the TargetLinux01 sidebar, **click** the **Terminal icon** to open the Terminal application.



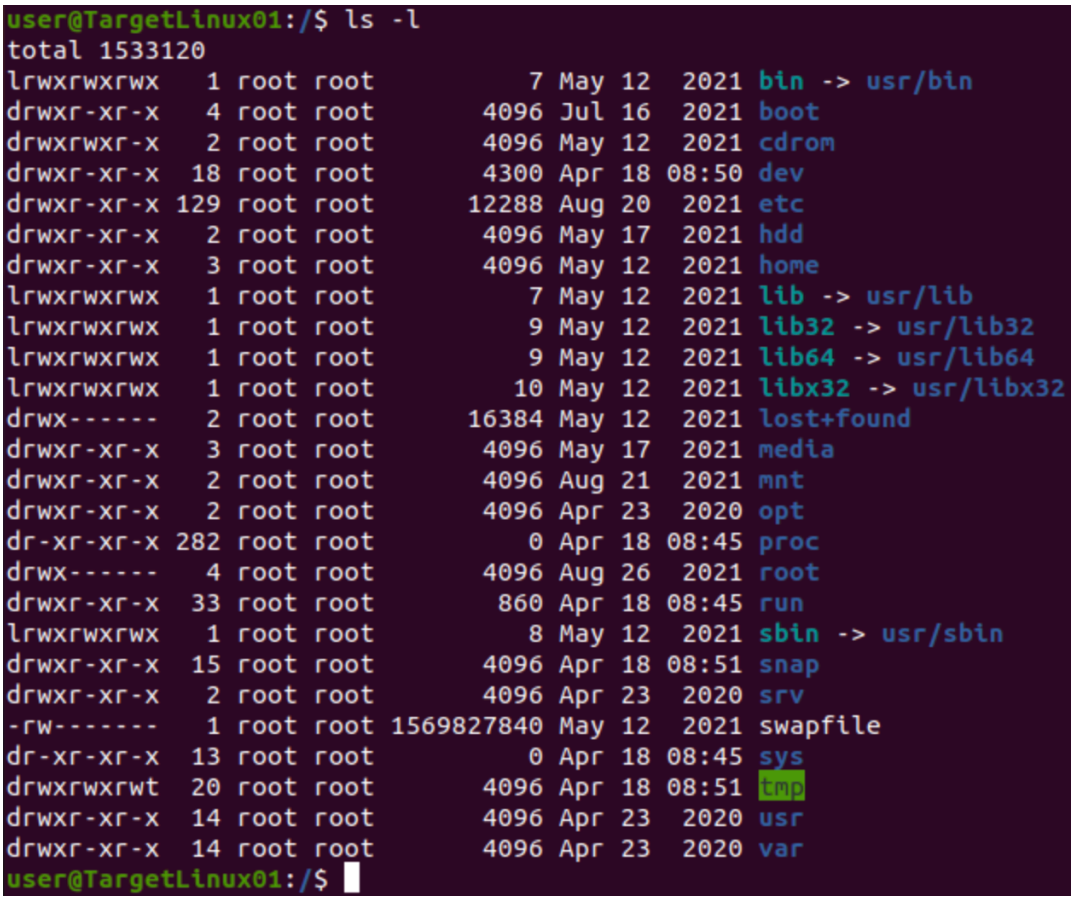
Terminal icon

13. At the command prompt, **type** `cd /` and **press Enter** to navigate to the root directory.

```
user@TargetLinux01:~$ cd /
user@TargetLinux01:/$
```

Navigate to the root directory

14. At the command prompt, **type** `ls -l` and **press Enter** to list the files in the root directory.



List files in the root directory

Note: The **`ls`** command lists files and directories within the current directory. The **`-l`** (lowercase L) option tells **`ls`** to print files in a long listing format so you can view all the file attributes easily, including size, modified date and time, file name, owner, and permissions. Adding the **`-l`** option is especially useful in a forensic investigation, as it can quickly provide a considerable amount of information about the files.

At a glance, the results of the **`ls -l`** command should match the graphical folder icons you observed earlier in the root directory.

15. At the command prompt, **type** **`cd /bin`** and **press Enter** to navigate to the /bin directory.

16. In the command prompt, **type** **`ls -l`** and **press Enter** to list the files in /bin directory

17. **Make a screen capture** showing the **contents of the /bin directory**.

18. At the command prompt, **type** `cd /` and **press Enter** to return to the root directory.

19. **Repeat steps 15-18** for the /etc, /var, and /proc directories.

20. **Make a screen capture** showing the **contents of the /etc directory**.

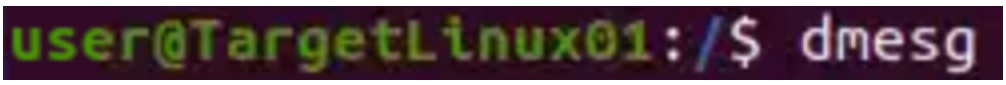
21. **Make a screen capture** showing the **contents of the /var directory**.

22. **Make a screen capture** showing the **contents of the /proc directory**.

Part 2: Use Linux Shell Commands for Forensic Investigations

Note: In this part of the lab, you use several common Linux commands that can be useful during a forensic investigation. As you saw in Part 1, while Linux offers a graphical user interface for navigating the file system, Linux is really designed to be used from the command line interface. While potentially challenging for new users, the command line interface allows seasoned professionals to quickly and efficiently retrieve critical information from a Linux system.

1. At the command prompt, **type** `dmesg` and **press Enter** to display the messages that were displayed during the boot process.



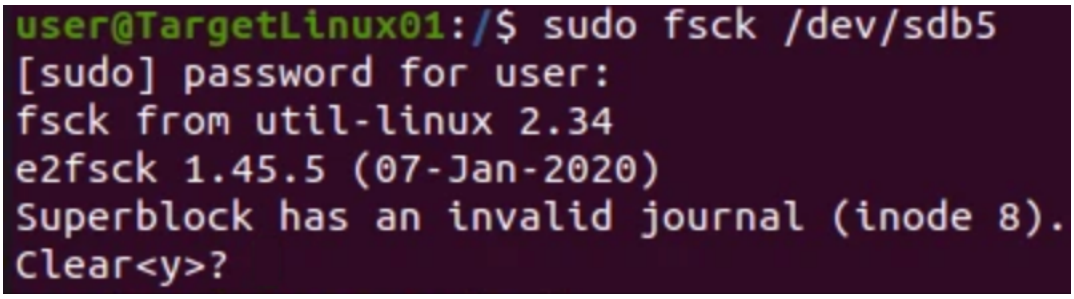
Dmesg

Note: The **`dmesg`** command allows you to view all the messages that were displayed during the boot process. When a Linux system boots up, you will see information about which processes are starting, which processes failed, what hardware is being initialized, and more. For forensic investigators, this command can be useful for determining if a specific process (for example, malware) was initiated during the boot process or if another process (for example, system logging) failed during boot. Due to the size of the output, it is typically advisable to export the output to a text file using the following syntax: **`dmesg > <filepath>`**.

2. **Make a screen capture** showing the **results of the dmesg command**.

3. At the command prompt, **type** `sudo fsck /dev/sdb5` and **press Enter** to check the file system.

   When prompted, **type** `password` and **press Enter** to escalate your privileges.



File system check

Note: The `fsck` command stands for *file system check*. This utility is used for checking and, if necessary, repairing the file system. In this case, you are checking the health of an ext4 filesystem on an attached disk, because running fsck on a mounted filesystem (one currently accessible from your system) is likely to lead to corruption.
The attached disk is called sdb, and is represented as a file — as everything is in Linux — located at /dev/sdb. On this disk is a partition, sdb5, which contains your target ext4 filesystem, and is represented by the file /dev/sdb5. These files are in fact interfaces that provide raw access to the attached device, with no concept of files or a filesystem structure. You would first be required to mount this drive to a location (a *mount point*) on your current filesystem before you could traverse its directory structure and interact with it at the file level. That is, assuming the superblock is undamaged, which contains information about the filesystem metadata that is crucial for facilitating these interactions. If the superblock is damaged, a mount is unlikely, and recovery is necessary.

In your case, it appears the journal superblock is corrupt. The journal contains changes to the filesystem that have not gone through yet (changes go through the journal first before they are written to disk), which can be used to restore a system from a power failure or similar with a lower likelihood of corruption. In the next steps, you are going to reissue the fsck command, this time specifying that it should operate in *preen* mode, which is to say: “don’t ask me any questions, just repair anything that can be safely fixed.”

4. At the command prompt, **hold ctrl** and **press c** to terminate the fsck program.

5. At the command prompt, **type** `sudo fsck /dev/sdb5 -p` and **press Enter** to repair the filesystem using fsck's preen (**-p**) option.

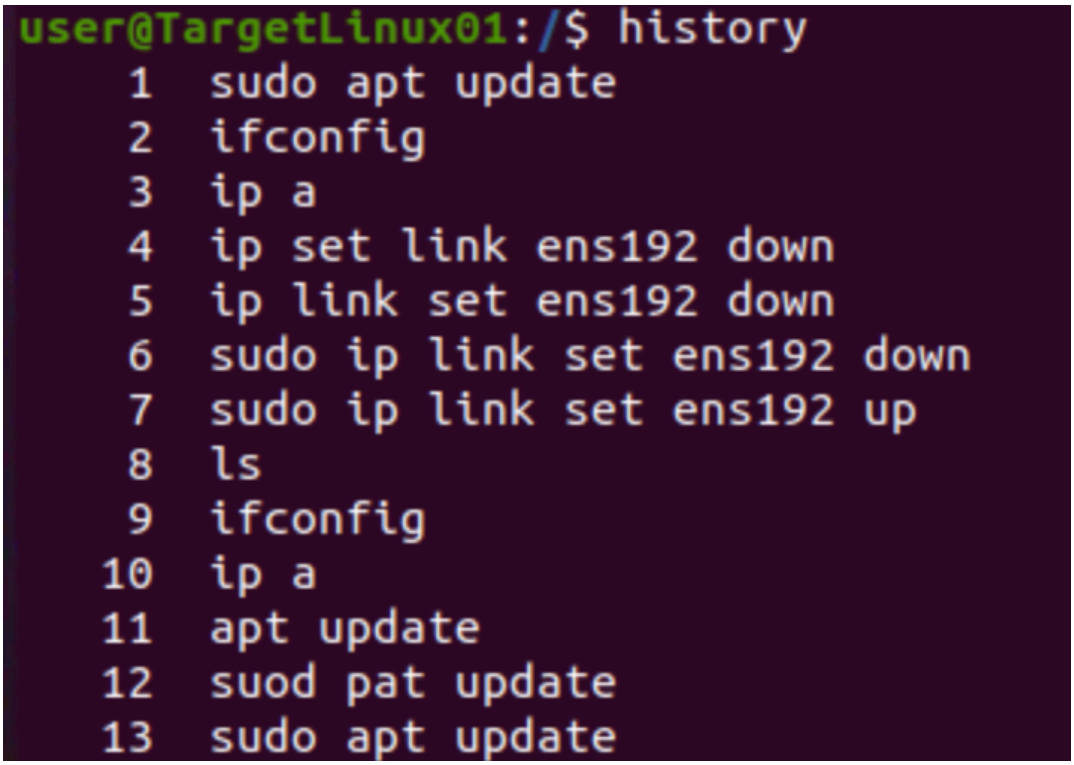
```
user@TargetLinux01:~$ sudo fsck /dev/sdb5 -p
fsck from util-linux 2.34
/dev/sdb5: Superblock has an invalid journal (inode 8).
CLEARED.
*** journal has been deleted ***

/dev/sdb5: Journal inode is not in use, but contains data.  CLEARED.
/dev/sdb5: Recreate journal.
Creating journal (16384 blocks):  Done.

*** journal has been regenerated ***
/dev/sdb5: 163682/622592 files (0.1% non-contiguous), 1357738/2489600 blocks
user@TargetLinux01:~$
```

Repair the filesystem

6. **Repeat step 3** to re-check the health of the ext4 filesystem on /dev/sdb5.

7. **Make a screen capture** showing the **results of the fsck command.**

8. At the command prompt, **type** `history` and **press Enter** to view the history of commands that were executed on the machine.



Command history

Note: For forensic investigators, viewing the history of commands that have been run on the operating system can tell you a great deal about what the system was doing prior to, possibly during, and after an event. For example, if **`mv`** (move) or **`cp`** (copy) commands were run, this could tell you that certain files were moved or copied. The command history can also tell you if a specific program had been run. For example, if you encountered a reference to “bleachbit”, a little research would tell you that it is a program designed to delete cookies, clear Internet history, shred temporary files, delete logs, and so on.

9. **Make a screen capture** showing the **results of the history command**.

10. At the command prompt, **type** **`ps -aux`** and **press Enter** to display the processes that are running for the current user.

```
user@TargetLinux01:/$ ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.5  0.6 168940 12448 ?        Ss   08:45   0:06 /sbin/init s
root           2  0.0  0.0      0     0 ?        S    08:45   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   08:45   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   08:45   0:00 [rcu_par_gp]
root           6  0.0  0.0      0     0 ?        I<   08:45   0:00 [kworker/0:0
root           9  0.0  0.0      0     0 ?        I<   08:45   0:00 [mm_percpu_w
root          10  0.0  0.0      0     0 ?        S    08:45   0:00 [ksoftirqd/0
root          11  0.0  0.0      0     0 ?        I    08:45   0:00 [rcu_sched]
root          12  0.0  0.0      0     0 ?        S    08:45   0:00 [migration/0
root          13  0.0  0.0      0     0 ?        S    08:45   0:00 [idle_inject
```

Active processes

Note: For forensic investigators, this command can reveal what is happening on a live system in real time. The **`ps -aux`** command displays the following information:

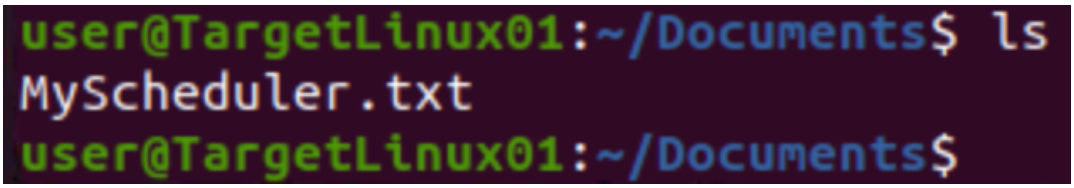
- User: The user account that initiated the process.
- PID: The process ID.
- %CPU: What percentage of CPU the process is using.
- %MEM: What percentage of memory the process is using.
- VSZ: Virtual memory size.
- RSS: Resident set size.
- TTY: Terminal type (a legacy of older Unix systems).
- Stat: The process state.
- Start: When the process was started
- Time: How long the process has been running.
- CMD: The name of the command that launched the process.

This command can also help detect Linux kernel process masquerading, which is sometimes used by malware to hide when it is running. Appending the “pipe” character (|) to your command will allow you to review the results, page by page, at your own pace.

11. **Make a screen capture** showing the **running processes**.

12. At the command prompt, **type** **`cd home/user/Documents`** and **press Enter** to navigate to the Documents directory.

13. At the command prompt, **type** **`ls`** and **press Enter** to display the contents of the Documents directory.



Documents directory

14. At the command prompt, **type** `file MyScheduler.txt` and **press Enter** to determine if the file was renamed or its extension was changed.

Note: The file command can tell you exactly what a file is, regardless of whether it has been renamed or had its extension changed. One of most common tricks for distributing malware is to rename an executable file to a text file or some other non-executing format in order to hide the file's true nature.

15. **Make a screen capture** showing the **results of the file command**.

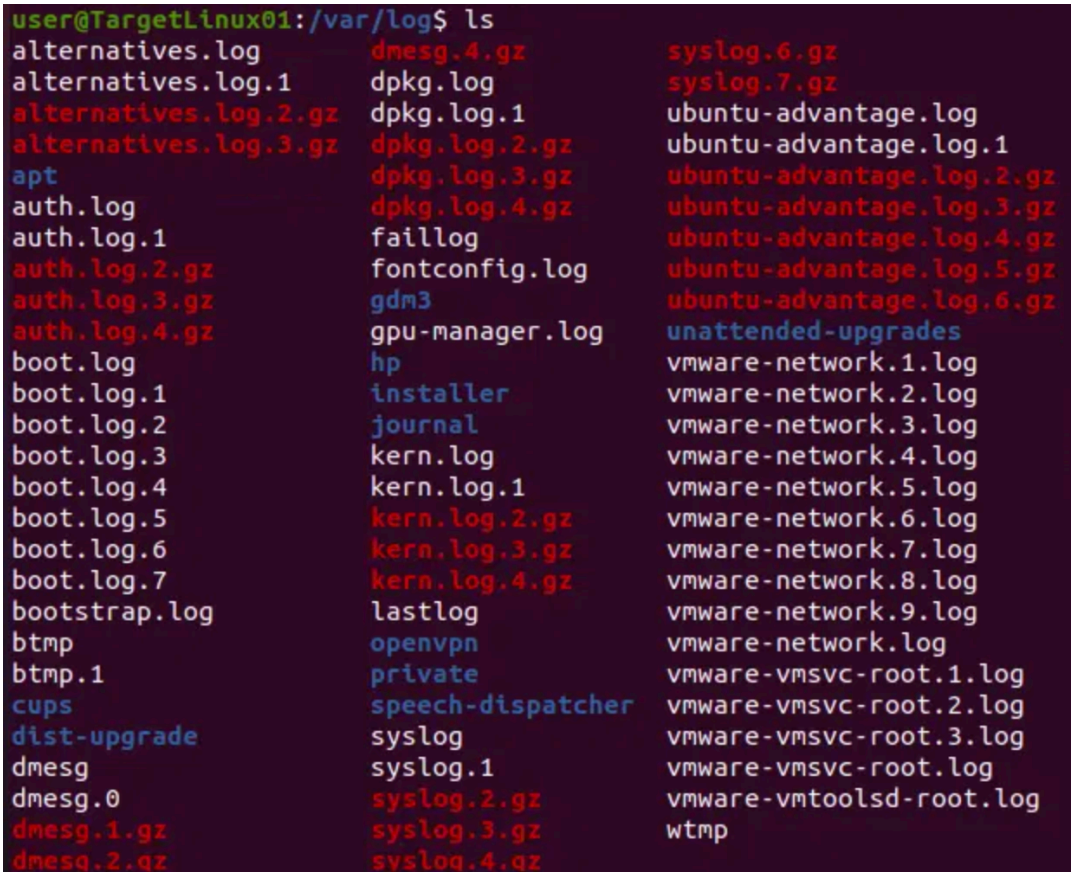
16. At the command prompt, **type** `cd /` and **press Enter** to return to the root directory.

Part 3: Retrieve Logs Files on a Live Linux System

Note: In this part of the lab, you will navigate to the /var/log directory and review the log files. For forensic investigators, log files are an invaluable source of information. By analyzing system logs, you can determine the timeline of activities and events that occurred during an incident. For example, under this directory you can view the message logs, which display a general message for everything that happened to the system, as well as authentication logs, kernel logs, boot logs, and the login records file.

1. At the command prompt, **type** `cd /var/log` and **press Enter** to navigate to the /var/log directory.

2. At the command prompt, **type** `ls` and **press Enter** to display the contents of the /var/log directory.



Display the contents of the /var/log directory

Note: Among the results, you should see a file titled kern.log. The kern.log file contains records related to the kernel and events on the system. For forensic investigators, the kern.log file can help with identifying the cause of system performance issues that might otherwise be mistaken for malware. The kern.log file is also useful for investigating physical security incidents involving USB flash drives, for which the kern.log keeps attachment records.

3. At the command prompt, **type** `sudo tail -f kern.log` and **press Enter** to display the last ten records in the kern.log file.

   If prompted, **type** `password` and **press Enter** to escalate your privileges.

```
user@TargetLinux01:/var/log$ sudo tail -f kern.log
Apr 18 08:50:37 TargetLinux01 kernel: [  326.191228] audit: type=1400 audit(174
4980637.424:65): apparmor="STATUS" operation="profile_replace" info="same as cu
rrent profile, skipping" profile="unconfined" name="/snap/snapd/23771/usr/lib/s
napd/snap-confine//mount-namespace-capture-helper" pid=2521 comm="apparmor_pars
er"
Apr 18 08:50:37 TargetLinux01 kernel: [  326.212631] audit: type=1400 audit(174
4980637.444:66): apparmor="STATUS" operation="profile_replace" profile="unconfi
ned" name="snap-update-ns.core" pid=2523 comm="apparmor_parser"
Apr 18 08:50:37 TargetLinux01 kernel: [  326.339246] audit: type=1400 audit(174
4980637.572:67): apparmor="STATUS" operation="profile_replace" profile="unconfi
ned" name="snap.core.hook.configure" pid=2524 comm="apparmor_parser"
Apr 18 08:50:39 TargetLinux01 kernel: [  327.936171] audit: type=1400 audit(174
4980639.168:68): apparmor="STATUS" operation="profile_replace" profile="unconfi
ned" name="snap-update-ns.snap-store" pid=2525 comm="apparmor_parser"
```

Display the records in the kern.log file

Note: To increase security within Linux, the sudo command is used to assign temporary privileges to users performing administrative tasks. This allows the root account to remain locked and prevents inexperienced users from making dangerous system changes.

While viewing kernel logs will likely not be part of every investigation, they can show you information about hardware drivers, kernel information and status during bootup, and more.

4. **Make a screen capture** showing the **records in the kern.log file.**

5. **Press Ctrl+c** to return to the command prompt.

6. At the command prompt, **type** `sudo more -f auth.log` and **press Enter** to display all of the records in the auth.log file, then **press Enter** to scroll through the logs until the command prompt reappears.

```
user@TargetLinux01:/var/log$ sudo more -f auth.log
Aug 11 22:34:56 TargetLinux01 gdm-autologin]: gkr-pam: no password is available
 for user
Aug 11 22:34:56 TargetLinux01 gdm-autologin]: pam_unix(gdm-autologin:session):
session opened for user user by (uid=0)
Aug 11 22:34:56 TargetLinux01 systemd-logind[625]: New session 1 of user user.
Aug 11 22:34:56 TargetLinux01 systemd: pam_unix(systemd-user:session): session
opened for user user by (uid=0)
Aug 11 22:34:57 TargetLinux01 gdm-autologin]: gkr-pam: gnome-keyring-daemon sta
rted properly
Aug 11 22:35:00 TargetLinux01 gnome-keyring-daemon[826]: The PKCS#11 component
was already initialized
Aug 11 22:35:00 TargetLinux01 gnome-keyring-daemon[826]: The Secret Service was
 already initialized
```

Display all of the records in the auth.log file

Note: The auth.log file contains records of all authorization-related events. For forensic investigators, this is an important file to review when searching for evidence of failed login attempts. The auth.log can show you the user that initiated an authentication request, as well as the path and command that was authorized and executed. During an investigation, this information can be cross-referenced with other logs to provide a clearer picture of what was happening on the system during a specific period.

7. **Make a screen capture** showing the **records in the auth.log file**.

8. **Close** the **Terminal window**.

Note: This concludes Section 1 of the lab.

Section 2: Applied Learning

Note: SECTION 2 of this lab allows you to apply what you learned in **SECTION 1** with less guidance and different deliverables, as well as some expanded tasks and alternative methods. You will explore a Linux drive image for evidence of a potential security breach.

Part 1: Identify Login Attempts on a Linux Drive Image

Note: For this exercise, you will assume the role of a junior forensic investigator at a consulting firm that assists government agencies, cryptocurrency businesses, and financial institutions in investigating illicit cyber activity. You have been assigned to a new client, Harper and Associates.

Harper and Associates is a large financial investment firm that manages billions of dollars in investments. The security team at Harper and Associates recently notified leadership that suspicious activity has been detected on a critical Linux server. The team believes that an intruder attempted to access the server using a legitimate user's credentials. To address the issue, Harper and Associates has hired your firm to locate evidence of the intrusion. To complete this task, you have been provided with a copy of a disk image taken from the Linux system after the intrusion attempt.

In the next steps, you will use Paraben's E3 to import a copy of the Linux drive image and access the system's log files.

1. On the Lab View toolbar, **select** the **vWorkstation** from the Virtual Machine menu to open a connection to the vWorkstation.

2. From the vWorkstation desktop, **launch** the **Electronic Evidence Examiner (E3) application**.

Note: E3 may take several minutes to load. The E3 welcome screen opens with shortcuts to the most common activities that forensic investigators will perform within the tool.

3. On the Welcome screen, **click** the **Add Evidence button** to open the New Case dialog box.

4. In the New Case dialog box, **type** ***yourname*** **`Linux Forensics`** in the Case name field, replacing *yourname* with your own name, then **click Continue** to create your new case file and open the Add New Evidence dialog box.

5. In the Add New Evidence dialog box, **click** the **Image File category**, then **select** the **Auto-detect image Source type** and **click OK** to continue.

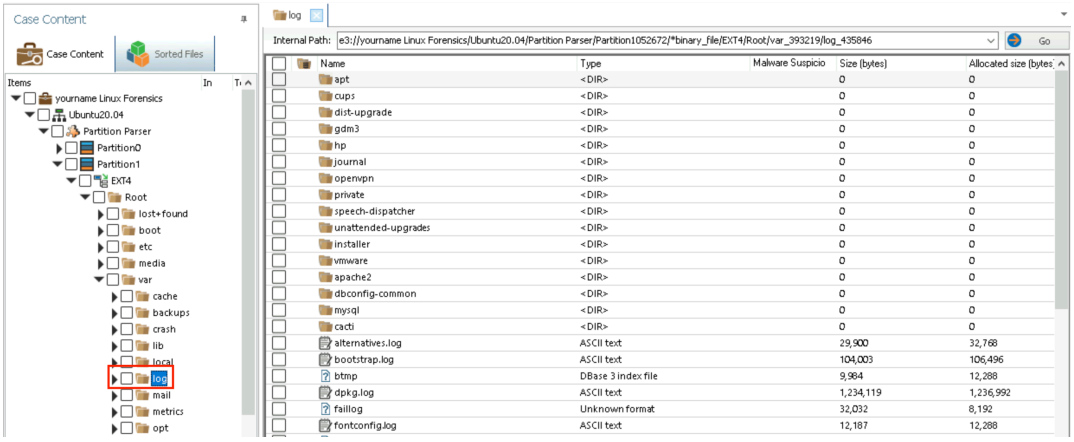
6. In the Open dialog box, **navigate** to **This PC > Local Disk (C:) > Linux Forensics** and **double-click** the **Ubuntu20.04.image** file to select the drive image for this lab.

7. When prompted, **click** the **OK button** to accept the default name for the drive image and add the data from the drive image to your case file.

Note: The *yourname* Linux Forensics case will appear in the Case Content pane. The Case Content pane is the primary display and navigation area for all evidence in the open case file. Within the Case Content pane's tree-view structure, you can access case nodes, evidence nodes, evidence type nodes, folder nodes, and data grids/streams. Technically, the Case Content pane does not store the actual evidence, but provides a series of links to elements within the physical evidence.

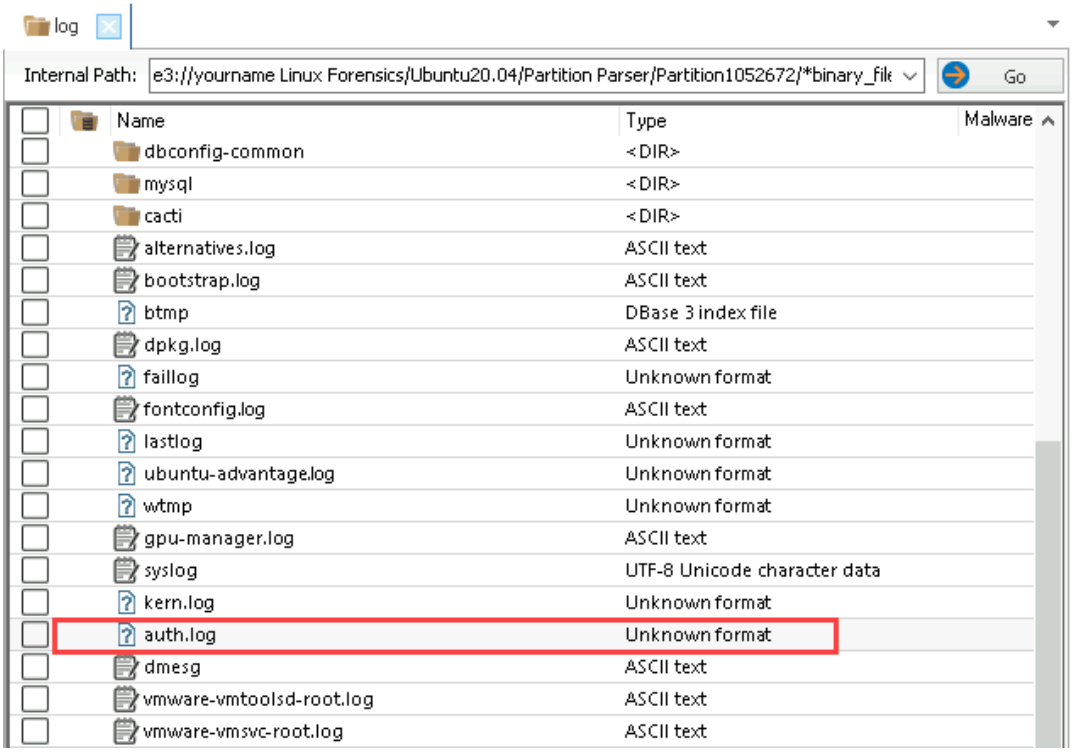
When you select a specific node or grid within the Case Content pane, the contents of the enclosing folder or database will be displayed in the Data Viewer in the center pane. Within the Data Viewer, you can select individual files, folders, and records. Additional information about the currently selected node, grid, file, folder, or record will be displayed in the Viewers pane on the right, which provides multiple ways to view your current selection.

8. In the Case Content pane, **navigate** to **Linux Forensics\Ubuntu20.04\Partition Parser\Partition1\EXT4\Root\var\log** to display the contents in the Data Viewer.



Log folder

9. In the Data Viewer, **select** the **auth.log file** to display information about the file in the Properties pane.

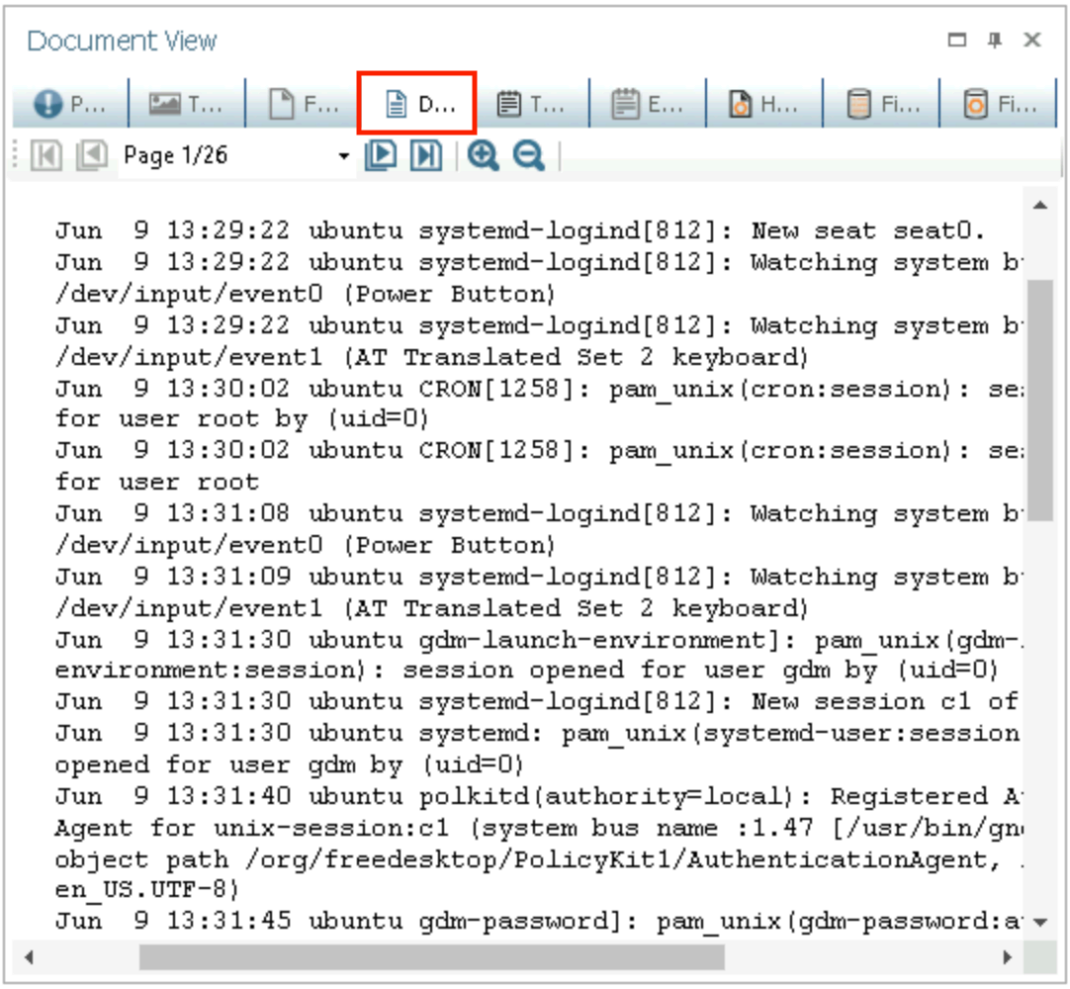


Auth.log

Note: As you may recall from Section 1, the auth.log file (or authorization log) tracks usage of a variety of authorization systems, such as password entry and failed entries, the sudo command, remote logins and so on. As a forensic investigator attempting to locate evidence of recent login attempts, the auth.log is a great place to begin your search.

10. In the Properties pane, **click** the **Document View tab** to display the contents of the file.

    You may need adjust the frame of the right pane to see the full contents of the file.



Document View

Note: As is often the case with log files, this one appears to be quite long – 26 pages to be exact. While you could attempt to manually read the entire thing, you would be unlikely to get very far before your eyes begin to glaze over.

To simplify your analysis, you will use E3's Advanced Search functionality to search the auth.log for the phrase “failed password”. This is a common technique when analyzing Linux logs. While different Linux distributions may phrase the event slightly differently, the words “failed” and “password” are both typically used in some capacity when recording failed login attempts in the auth.log.

11. In the Data Viewer, **right-click** the **auth.log file** and **select Advanced Search** from the context menu to open the Advanced Search pane in the center console.

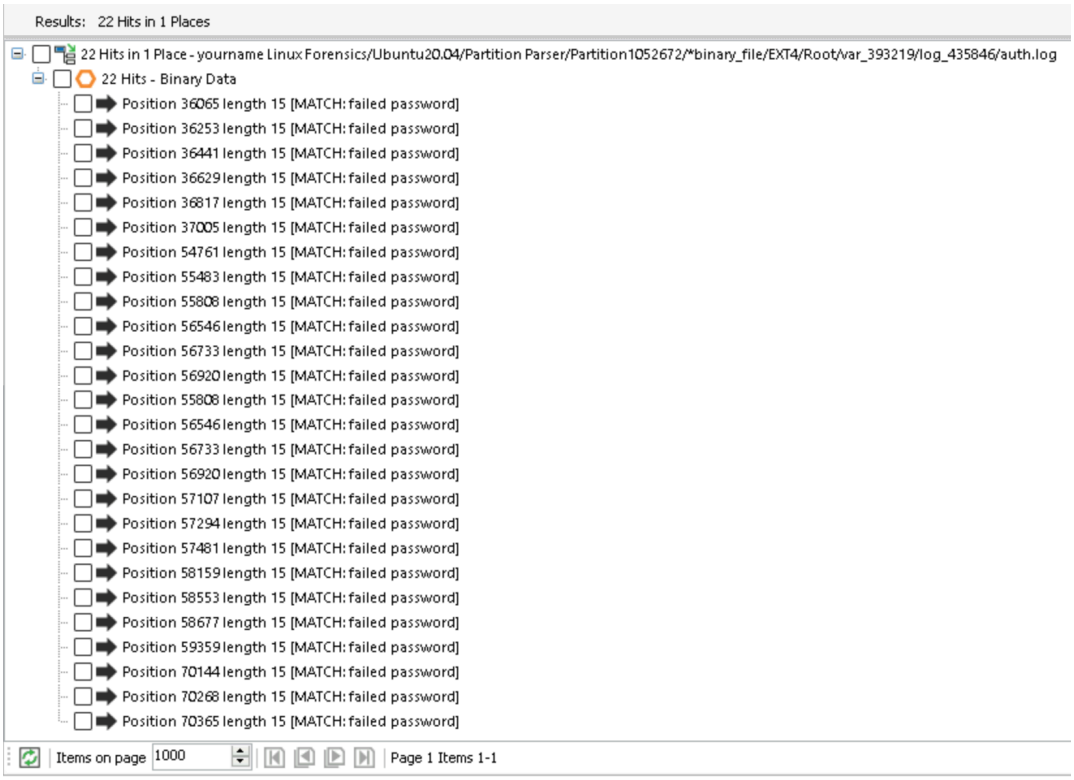
12. In the Advanced Search pane, **type** **`“failed password”`** in the Search what field, **select**

Simple search from the Use menu, and then **click** the **Start button** to begin the search.

You may need to scroll to the right to see the Start button.

13. When prompted, **click OK** to close the Task Status Notification.

14. When the search is completed, **expand** the **22 Hits in 1 Place** and **22 Hits – Binary Data headers**, then **double-click each result** to jump to the corresponding instance of the search term within the Text View.



Search results

Note: Among the results, you should identify multiple unsuccessful login attempts outside of typical work hours (from 7:00 PM to 9:00 AM) on June 11. As a general rule, when a legitimate user account repeatedly attempts to log in outside of typical work hours using invalid credentials, this is a strong sign of malicious activity.

15. **Document** the names of the two non-root users that attempted to log in, the number of

attempts detected, the date/time range of the attempts, the source IP address for the login attempts, and the port.

Note: Based on the variable time intervals between failed attempts, you can reasonably conclude that the login attempts were generated by a human being, rather than a machine conducting a brute force attack.

In the next steps, you will conduct a similar search for successful logins to determine if the intruder was successful. The keyword that is associated with successful logins in the auth.log is “session opened for user”. A session is a temporary and interactive information interchange between a computer and user. Each session is unique to the user and allows the user to interact with the server. In this case, an open session means that the user has successfully accessed the server.

16. **Repeat steps 12-14** using the search term **`"session opened for user`*`username`*`"`**, where *username* is the name of one of the users you identified in step 15.

    You will need to repeat this step with each user.

17. **Document** the date and time the most recent successful login for the user(s) that you previously identified in step 15.

x

Part 2: Identify Software Installations on a Linux Drive Image

Note: At this point, you have found evidence that a legitimate user account made repeated unsuccessful login attempts outside of work hours, but eventually logged in successfully. This series of attempts all took place on June 11. If you rule out the possibility that the legitimate account owner was just trying to access a production file server in the early hours of the morning to get a jump on the day, but had not had their coffee yet and repeatedly entered the wrong password, then it seems likely that the suspected intruder was using a compromised account. If that is the case, then it seems unlikely that the trail ends there. You will need to investigate further to determine how they obtained the valid user credentials in the first place.

1. In the Advanced Search pane, **type `“apt-get install”`** in the Search what field and **click** the **Start** button to begin the search.

Note: Advanced Package Tool, or APT, is a free software user interface that works with core libraries to handle the installation and removal of software on Linux distributions. The **`apt-get install`**

command installs a new package on a Linux system. As a forensic investigator, searching for records of this command in the auth.log can tell you what software was recently installed on the system.

2. When the search is completed, **review** the results and **identify** any unusual installation commands.

3. **Document** the applications that were installed using apt-get, then use the Internet to identify the ones that might be considered suspicious.

Note: Among the applications installed, you should see at least one that is related to keylogging that appears to have been installed on June 10. You may also notice that the apt-get command was run using the **sudo** command, which indicates that whoever installed the software had access to an account that was authorized to run commands with root privileges.

Part 3: Identify External Drive Attachments on a Linux Drive Image

Note: Unfortunately, the Harper and Associates security team's suspicions have been proven correct. Not only did an intruder gain access to the Linux server, but they were likely assisted by an internal collaborator who installed network monitoring and keylogging software. This could explain how the intruder obtained the compromised account credentials that they used on June 11, and why they were accompanied by several failed attempts.

After reporting your initial findings to the security team, they inform you that the server in question is actually an older system that runs directly within a server closet at the company's Boston office. Your evidence suggesting an internal collaborator prompts them to review the security footage from June 10, and they confirm that a hooded individual did indeed access the server closet about two hours before the auth.log records indicate the keylogger was installed. This revelation suggests that the server may have been physically compromised as well, and the security team asks you to review the drive image for evidence of any physical drive attachments. Thanks to some recent Linux training you received, you know that the kern.log file should have records of any recent drive attachments.

1. In the center console, **click** the **log tab**, then **right-click** the **kern.log file** and **select Advanced Search** from the context menu to open a new Advanced Search tab.

2. In the center pane, **type** **`"USB mass storage device detected"`** in the Search what field, **select Simple search** from the Use menu, and then **click** the **Start button** to begin the search.

3. When the search is completed, **review** the results.

4. **Document** when the USB storage device was connected and its serial number.

Note: This concludes Section 2 of the lab.

Section 3: Challenge and Analysis

Note: The following exercises are provided to allow independent, unguided work using the skills you learned earlier in this lab - similar to what you would encounter in a real-world situation.

Part 1: Identify Recently Printed Files on a Linux Drive Image

In Section 2 of this lab, you explored various log files and located evidence of unauthorized physical access to a Linux server by an internal user at the company. According to the evidence you located, this person accessed the server over a two-hour period, which means they likely performed other actions during that window.

Using the Internet, research the location of the log file that Linux uses to keep track of printing records. Next, use E3 to access that log file and display the contents in the Document View.

Make a screen capture showing the **contents of the printer log file**.

Part 2: Identify Disk Imaging on a Linux Drive Image

The security team has come back to you with additional evidence that the insider threat who accessed the server closet may have created a copy of the server's hard drive and used the USB drive to exfiltrate it. They have asked you to review the evidence drive again and attempt to identify evidence in the log folder that a disk copy command was run.

Using the Internet, research the syntax and arguments used for the **dd** command. Next, use E3 to run an Advanced Search on the /var/log folder for any records of the **dd** command being run.

Make a screen capture showing the **record of the dd command in the Text View**.

Note: This concludes Section 3 of the lab.