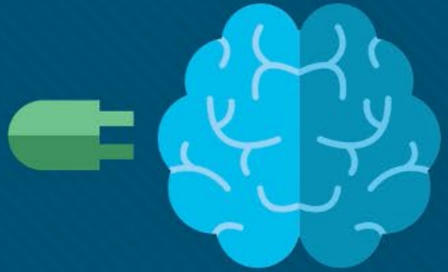


Module 8: Linux Overview

Cybersecurity Essentials 3.0

Module Objectives

Module Title: Linux Overview

Module Objective: Implement basic Linux security.

Topic Title	Topic Objective
Linux Basics	Explain why Linux skills are essential for network security monitoring and investigation.
Working in the Linux Shell	Use the Linux shell to manipulate text files.
Linux Servers and Clients	Use the Linux command line to identity servers that are running on a computer.
Basic Server Administration	Use commands to locate and monitor log files.
The Linux File System	Use commands to manage the Linux file system and permissions.
Working in the Linux GUI	Explain the basic components of the Linux GUI.
Working on a Linux host	Use tools to detect malware on a Linux host.

8.1 Linux Basics

What is Linux?

- Linux is an operating system created in 1991 that is open source, fast, reliable, and small.
- It requires very little hardware resources to run and is highly customizable.
- Unlike Windows and Mac OS X, it was created and currently maintained by a community of programmers.
- It is designed to be connected to the network, which makes it much simpler to write and use network-based applications.
- Any person or company can get the kernel’s source code, inspect it, modify it, and re-compile it at will.
- They are also allowed to redistribute the program with or without charges.
- Linux distribution (distro) is the term used to describe packages created by different organizations and includes the Linux kernel with customized tools and software packages.
- Examples of distros include: Debian, Red Hat, Ubuntu, CentOS, and SUSE.

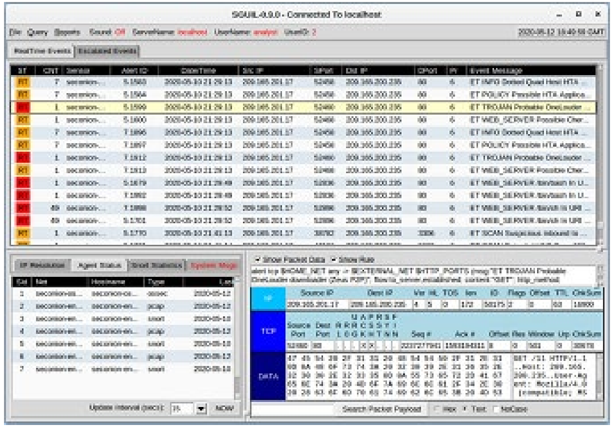
The Value of Linux

Linux is often the operating system of choice in the Security Operations Center (SOC). Some of the reasons to choose Linux:

- Linux is open source - Anyone can acquire Linux at no charge and modify it to fit specific needs.
- The Linux CLI is very powerful - GUI makes many tasks easier to perform but adds complexity and requires more computer resources to run. CLI enables analysts to perform tasks directly on a terminal and remotely.
- The user has more control over the OS - The administrator user in Linux (root user or superuser) has absolute power over the computer. The root user can modify any aspect of the computer with a few keystrokes.
- It allows for better network communication control - Because the OS can be adjusted in practically every aspect, it is a great platform for creating network applications.

Linux in the SOC

- The entire operating system can be tailored to become the perfect security analysis platform.
- Administrators can add only the necessary packages to the OS, making it lean and efficient.
- Specific software tools can be installed and configured to work in conjunction, allowing administrators to build a customized computer that fits perfectly in the workflow of a SOC.
- The figure shows Sguil, which is the cybersecurity analyst console in a special version of Linux called Security Onion.
- Security Onion is an open-source suite of tools that work together for network security analysis.



SGUIL-0.9.0 - Connected To localhost

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	7	seconion-...	5.1563	2020-05-10 21:29:13	209.165.201.17	52456	209.165.200.235	80	6	ET INFO Dotted Quad Host HTA ...
RT	7	seconion-...	5.1564	2020-05-10 21:29:13	209.165.201.17	52456	209.165.200.235	80	6	ET POLICY Possible HTA Applica...
RT	1	seconion-...	5.1599	2020-05-10 21:29:13	209.165.201.17	52460	209.165.200.235	80	6	ET TROJAN Probable OneLouder ...
RT	1	seconion-...	5.1600	2020-05-10 21:29:13	209.165.201.17	52460	209.165.200.235	80	6	ET WEB_SERVER Possible Cher...
RT	7	seconion-...	7.1896	2020-05-10 21:29:13	209.165.201.17	52456	209.165.200.235	80	6	ET INFO Dotted Quad Host HTA ...
RT	7	seconion-...	7.1897	2020-05-10 21:29:13	209.165.201.17	52456	209.165.200.235	80	6	ET POLICY Possible HTA Applica...
RT	1	seconion-...	7.1912	2020-05-10 21:29:13	209.165.201.17	52460	209.165.200.235	80	6	ET TROJAN Probable OneLouder ...
RT	1	seconion-...	7.1913	2020-05-10 21:29:13	209.165.201.17	52460	209.165.200.235	80	6	ET WEB_SERVER Possible Cher...
RT	1	seconion-...	5.1679	2020-05-10 21:29:49	209.165.201.17	53006	209.165.200.235	80	6	ET WEB_SERVER /bin/bash In U...
RT	1	seconion-...	7.1992	2020-05-10 21:29:49	209.165.201.17	53006	209.165.200.235	80	6	ET WEB_SERVER /bin/bash In U...
RT	49	seconion-...	7.1998	2020-05-10 21:29:52	209.165.201.17	53006	209.165.200.235	80	6	ET WEB_SERVER /bin/sh In URI ...
RT	49	seconion-...	5.1701	2020-05-10 21:29:52	209.165.201.17	53006	209.165.200.235	80	6	ET WEB_SERVER /bin/sh In URI ...
RT	1	seconion-...	5.1770	2020-05-10 21:41:13	209.165.201.17	38792	209.165.200.235	3306	6	ET SCAN Suspicious inbound to ...

Sid	Net	Hostname	Type	Last
1	seconion-os...	seconion-os...	ossec	2020-05-12
2	seconion-eth...	seconion-eth...	pcap	2020-05-12
3	seconion-eth...	seconion-eth...	snort	2020-05-10
4	seconion-eth...	seconion-eth...	pcap	2020-05-12
5	seconion-eth...	seconion-eth...	snort	2020-05-10
6	seconion-eth...	seconion-eth...	pcap	2020-05-12
7	seconion-eth...	seconion-eth...	snort	2020-05-10

Update Interval (secs): 15 NOW

☑ Show Packet Data ☑ Show Rule

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"ET TROJAN Probable
OneLouder downloader (Zeus P2P)"; flow:to_server,established; content:"GET"; http_method;
```

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
	209.165.201.17	209.165.200.235	4	5	0	172	50175	2	0	63	16900

TCP	Source Port	Dest Port	R 1	R 0	U R G	A C K	P S H	R S T	S Y N	F I N	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum
	52460	80	.	.	.	X	X	.	.	.	2237277901	1583184311	8	0	502	0	30878

```
DATA
47 45 54 20 2F 31 31 20 48 54 54 50 2F 31 2E 31   GET /11 HTTP/1.1
0D 0A 48 6F 73 74 3A 20 32 30 39 2E 31 36 35 2E   ..Host: 209.165.
32 30 30 2E 32 33 35 0D 0A 55 73 65 72 2D 41 67   200.235..User-Ag
65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 34 2E 30   ent: Mozilla/4.0
20 28 63 6F 6D 70 61 74 69 62 6C 65 3B 20 4D 53    (compatible; MS
```

Search Packet Payload ○ Hex ● Text □ NoCase

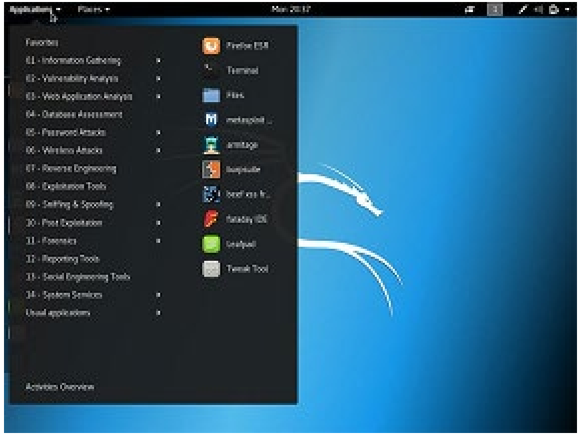
Linux in the SOC (Cont.)

Tools that are often found in a SOC are:

Tool	Explanation
Network packet capture software	This is a crucial tool to observe and understand every detail of a network transaction. Wireshark is a popular packet capture tool.
Malware analysis tools	They allow analysts to safely run and observe malware execution without the risk of compromising the underlying system.
Intrusion detection systems (IDSs)	They are used for real-time traffic monitoring and inspection. If any aspect of the currently flowing traffic matches any of the established rules, a pre-defined action is taken.
Firewalls	This software is used to specify, based on pre-defined rules, whether traffic is allowed to enter or leave a network or device.
Log managers	Because a network can generate a very large number of log entries, log manager software is employed to facilitate log monitoring.
Security information and event management (SIEM)	SIEMs provide real-time analysis of alerts and log entries generated by network appliances such as IDSs and firewalls.
Ticketing systems	Task ticket assignment, editing, and recording is done through a ticket management system.

Linux Tools

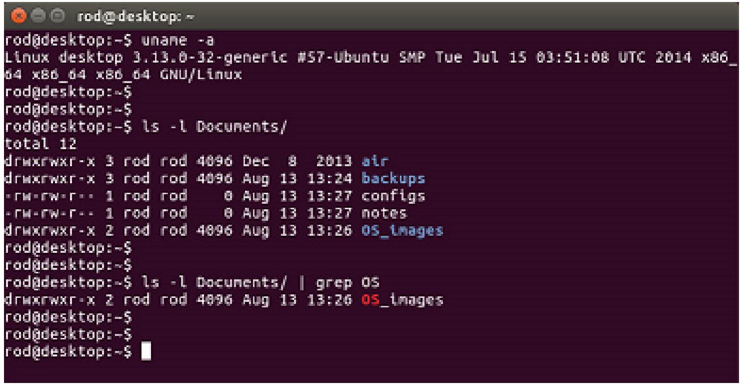
- In addition to SOC-specific tools, Linux computers that are used in the SOC often contain penetration testing tools.
- A penetration test (PenTesting) is the process of looking for vulnerabilities in a network or computer by attacking it.
- Packet generators, port scanners, and proof-of-concept exploits are examples of PenTesting tools.
- Kali Linux is a Linux distribution that groups many penetration tools together in a single Linux distribution. It contains a great selection of tools.
- The figure shows a screenshot of Kali Linux.



8.2 Working in the Linux Shell

The Linux Shell

- In Linux, the user communicates with the OS by using the CLI or the GUI (default).
- This hides the CLI from the user. One way to access the CLI from the GUI is through a terminal emulator application. These applications provide user access to the CLI.
- In Linux, popular terminal emulators are Terminator, eterm, xterm, konsole, and gnome-terminal.
- The figure shows gnome-terminal, a popular Linux terminal emulator.
- The terms shell, console, console window, CLI terminal, and terminal window are often used interchangeably.



```
rod@desktop:~$ uname -a
Linux desktop 3.13.0-32-generic #57-Ubuntu SMP Tue Jul 15 03:51:08 UTC 2014 x86_
64 x86_64 x86_64 GNU/Linux
rod@desktop:~$
rod@desktop:~$
rod@desktop:~$ ls -l Documents/
total 12
drwxrwxr-x 3 rod rod 4096 Dec  8  2013 air
drwxrwxr-x 3 rod rod 4096 Aug 13 13:24 backups
-rw-rw-r-- 1 rod rod    0 Aug 13 13:27 configs
-rw-rw-r-- 1 rod rod    0 Aug 13 13:27 notes
drwxrwxr-x 2 rod rod 4096 Aug 13 13:26 OS_images
rod@desktop:~$
rod@desktop:~$
rod@desktop:~$ ls -l Documents/ | grep OS
drwxrwxr-x 2 rod rod 4096 Aug 13 13:26 OS_images
rod@desktop:~$
rod@desktop:~$
rod@desktop:~$
```

Basic Commands

Linux commands are programs stored on the disk created to perform a specific task. When a user types a command, the shell must find it on the disk before it can be executed.

The table lists basic Linux commands and their functions.

Command	Description
mv	Moves or renames files and directories
chmod	Modifies file permissions
chown	Changes the ownership of a file
dd	Copies data from an input to an output
pwd	Displays the name of the current directory
ps	Lists the processes currently running in the system
su	Simulates a login as another user or to become a superuser
sudo	Runs a command as a super user, by default, or another named user

Working in the Linux Shell

Basic Commands (Cont.)

Command	Description
grep	Used to search for specific strings of characters within a file or other command outputs.
ifconfig	Used to display or configure network card related information. If issued without parameters, ifconfig will display the current network card(s) configuration.
apt-get	Used to install, configure, and remove packages on Debian and its derivatives.
iwconfig	Used to display or configure wireless network card related information.
shutdown	Used to shut down the system, shutdown can be instructed to perform a number of shut-down-related tasks.
passwd	Used to change the password.
cat	Used to list the contents of a file and expects the file name as the parameter.
man	Used to display the documentation for a specific command.

File and Directory Commands

Many command line tools are included in Linux by default.

The table lists a few of the most common commands related to files and directories.

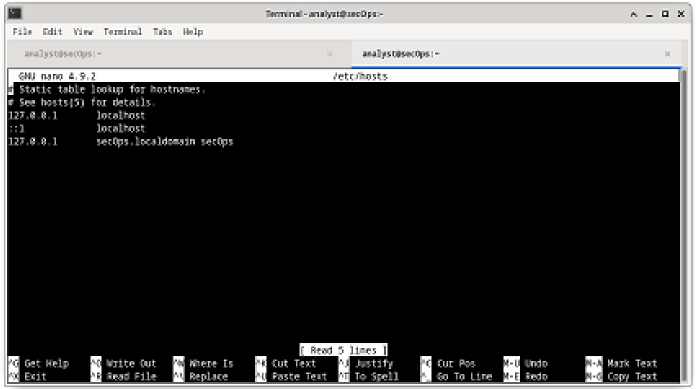
Command	Description
ls	Displays the files inside a directory
cd	Changes the current directory
mkdir	Creates a directory under the current directory
cp	Copies files from source to destination
mv	Moves or renames files and directories
rm	Removes files
grep	Searches for specific strings of characters within a file or other commands outputs
cat	Lists the contents of a file and expects the file name as the parameter

Working with Text Files

- Linux has many different text editors, with various features and functions. Some text editors include graphical interfaces while others are command-line only tools.
- Each text editor includes a feature set designed to support a specific type of task.
- Some text editors focus on the programmer and include features such as syntax highlighting, brackets and parenthesis check, and other programming-focused features.
- While graphical text editors are convenient and easy to use, command line-based text editors are very important for Linux users.
- The main benefit of command-line-based text editors is that they allow for text file editing from a remote computer.
- Consider the following scenario: a user must perform administrative tasks on a Linux computer but is not sitting in front of it. Using SSH, the user starts a remote shell to the remote computer and launches a text-based tool to perform the tasks.

The Importance of Text Files in Linux

- In Linux, everything is treated as a file (memory, the disks, the monitor, and the directories).
- Practically everything in Linux relies on configuration files to work. Some services have not one, but several configuration files.
- Users with proper permission levels can use text editors to change the contents of configuration files.
- After the changes are made, the file is saved and can be used by the related service or application.
- In the figure, the administrator opened the host configuration file in nano for editing with the **sudo nano /etc/hosts** command.
- The superuser or a user with the superuser privilege can change the host file.


Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
analyst@secOps:~
analyst@secOps:~

```
  GNU nano 4.9.2                                  /etc/hosts
# Static table lookup for hostnames.
# See hosts(5) for details.
127.0.0.1       localhost
::1             localhost
127.0.0.1       secOps.localdomain secOps

                                  [ Read 5 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text
```

Lab – Working with Text Files in the CLI

In this lab, you will become familiar with Linux command line text editors and configuration files, and you complete the following objectives:

- Part 1: Graphical Text Editors
- Part 2: Command Line Text Editors
- Part 3: Working with Configuration Files

Lab – Getting Familiar with the Linux Shell

In this lab, you will use the Linux command line to manage files and folders and perform some basic administrative tasks:

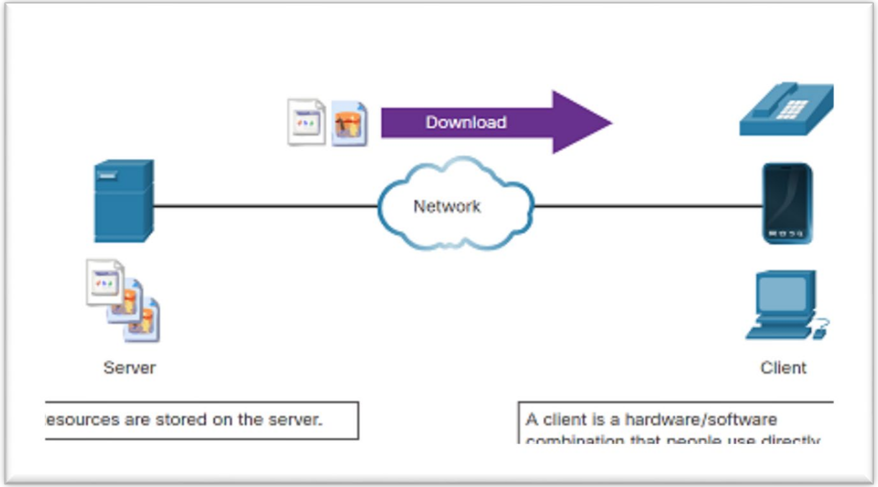
- Part 1: Shell Basics
- Part 2: Copying, Deleting, and Moving Files

8.3 Linux Servers and Clients

Linux Servers and Clients

An Introduction to Client-Server Communications

- Servers are computers with software installed that enables them to provide services to clients across the network.
- Some services provide external resources (files, email messages, web pages) to clients and others run maintenance tasks (log management, memory management, disk scanning).
- Each service requires separate server software. For example, the server in the figure uses file server software to provide clients with the ability to retrieve and submit files.



esources are stored on the server.

A client is a hardware/software combination that people use directly

Servers, Services, and Their Ports

- In order that a computer can be the server for multiple services, ports are used.
- A port is a reserved network resource used by a service. A service is said to be “listening” on a port when it has associated itself to that port.
- While the administrator can decide which port to use with any given service, many clients are configured to use a specific port by default.
- It is common practice to leave the service running in its default port.

The table lists a few commonly used ports (“well-known ports”) and their services.

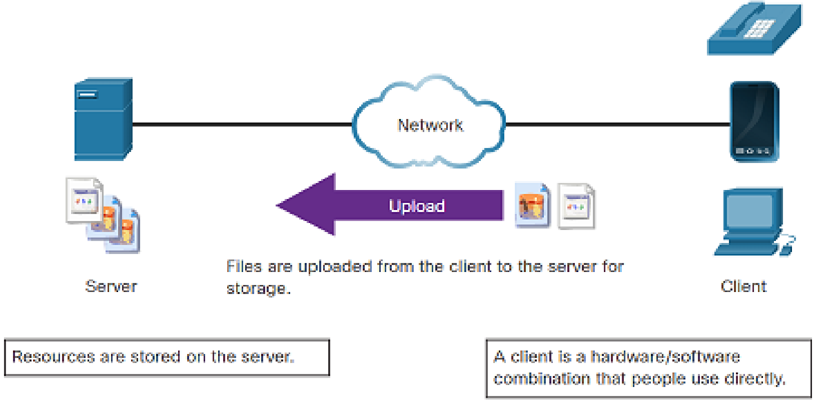
Port	Description
20/21	File Transfer Protocol (FTP)
22	Secure Shell (SSH)
23	Telnet remote login service
25	Simple Mail Transfer Protocol (SMTP)
53	Domain Name System (DNS)

Servers, Services, and Their Ports (Cont.)

Port	Description
67/68	Dynamic Host Configuration Protocol (DHCP)
69	Trivial File Transfer Protocol (TFTP)
80	Hypertext Transfer Protocol (HTTP)
110	Post Office Protocol version 3 (POP3)
123	Network Time Protocol (NTP)
143	Internet Message Access Protocol (IMAP)
161/162	Simple Network Management Protocol (SNMP)
443	HTTP Secure (HTTPS)

Clients

- Clients are programs or applications designed to communicate with a specific type of service.
- Clients (client applications) use a well-defined protocol to communicate with the server.
- Web browsers are web clients that are used to communicate with web servers through HTTP on port 80.
- The FTP client is software used to communicate with an FTP server.
- The figure shows a client uploading files to a server.



Files are uploaded from the client to the server for storage.

Resources are stored on the server.

A client is a hardware/software combination that people use directly.

Lab Video - Use a Port Scanner to Detect Open Ports

In this video, the use of a port scanner and network mapping tool, Nmap, to detect open ports is discussed and demonstrated:

- Step 1: Open a terminal window in the CSE-LABVM (PC with the CSE-LABVM installed in VirtualBox).
- Step 2: Run Nmap.
- Step 3: Use administrative privileges with Nmap.
- Step 4: Capture SSH keys.

Lab - Use a Port Scanner to Detect Open Ports

In this lab, you will use Nmap, a port scanner and network mapping tool, to detect open ports.

- Step 1: Open a terminal window in the CSE-LABVM (PC with the CSE-LABVM installed in VirtualBox).
- Step 2: Run Nmap.
- Step 3: Use administrative privileges with Nmap.
- Step 4: Capture SSH keys.

Lab - Linux Servers

In this lab, you will use the Linux command line to identify servers running on a given computer.

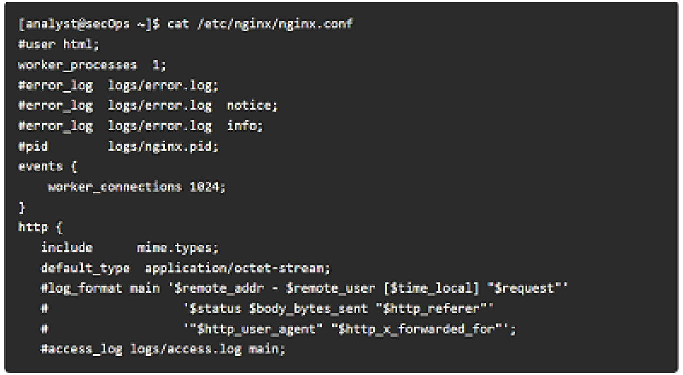
- Part 1: Servers
- Part 2: Using Telnet to Test TCP Services

8.4 Basic Server Administration

Basic Server Administration

Service Configuration Files

- In Linux, services are managed using configuration files.
- Common options in configuration files are port number, location of the hosted resources, and client authorization details.
- When the service starts, it looks for its configuration files, loads them into memory, and adjusts itself according to the configuration settings in the files.
- Configuration file modifications often require restarting the service before the changes take effect.
- Because services often require superuser privileges to run, service configuration files often require superuser privileges to edit.
- The command output shows a portion of the configuration file for Nginx, which is a lightweight web server for Linux.



```
[analyst@secOps ~]$ cat /etc/nginx/nginx.conf
#user html;
worker_processes  1;
#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;
#pid        logs/nginx.pid;
events {
    worker_connections 1024;
}
http {
   include      mime.types;
   default_type  application/octet-stream;
   #log_format main '$remote_addr - $remote_user [$time_local] "$request"'
   #                  '$status $body_bytes_sent "$http_referer"'
   #                  '"$http_user_agent" "$http_x_forwarded_for"';
   #access_log logs/access.log main;
```

Hardening Devices

- Device hardening involves implementing proven methods of securing the device and protecting its administrative access.
- Defining administrative roles in terms of access is another important aspect of securing infrastructure devices.
- OS updates are also extremely important to maintaining a hardened device. New vulnerabilities are discovered every day. OS developers create and issue fixes and patches regularly.

The basic best practices for device hardening are:

- Ensure physical security
- Minimize installed packages
- Disable unused services
- Use SSH and disable the root account login over SSH
- Keep the system updated
- Disable USB auto-detection
- Enforce strong passwords
- Force periodic password changes
- Keep users from re-using old passwords

Monitoring Service Logs

- Log files are the records that a computer stores to keep track of important events. Kernel, services, and application events are all recorded in log files.
- It is very important for an administrator to periodically review the logs of a computer to keep it healthy.
- By monitoring Linux log files, an administrator gains a clear picture of the computer’s performance, security status, and any underlying issues.
- Log file analysis allows an administrator to guard against upcoming issues before they occur.

In Linux, log files can be categorized as:

- Application logs
- Event logs
- Service logs
- System logs

- Some logs contain information about daemons that are running in the Linux system.
- A daemon is a background process that runs without the need for user interaction.

Monitoring Service Logs (Cont.)

The table lists a few popular Linux log files and their functions.

Linux Log File	Description
/var/log/messages	This directory contains generic computer activity logs. It is mainly used to store informational and non-critical system messages.
/var/log/auth.log	This file stores all authentication-related events in Debian and Ubuntu computers. Anything involving the user authorization mechanism can be found in this file.
/var/log/secure	This directory is used by RedHat and CentOS computers instead of /var/log/auth.log. It also tracks sudo logins, SSH logins, and other errors logged by SSSD.
/var/log/boot.log	This file stores boot-related information and messages logged during the computer startup process.
/var/log/dmesg	This directory contains kernel ring buffer messages. Information related to hardware devices and their drivers is recorded here.

Basic Server Administration

Monitoring Service Logs (Cont.)

The table lists a few more popular Linux log files and their functions.

Linux Log File	Description
/var/log/kern.log	This file contains information logged by the kernel.
/var/log/cron	Cron is a service to schedule automated tasks in Linux and this directory stores its events.
/var/log/mysqld.log or /var/log/mysql.log	This is the MySQL log file. All debug, failure, and success messages related to the mysqld process and mysqld_safe daemon are logged here.

Lab – Locating Log Files

In this lab, you will get familiar with locating and manipulating Linux log files.

- Part 1: Log File Overview
- Part 2: Locating Log Files in Unknown Systems
- Part 3: Monitoring Log Files in Real Time

8.5 The Linux File System

The File System Types in Linux

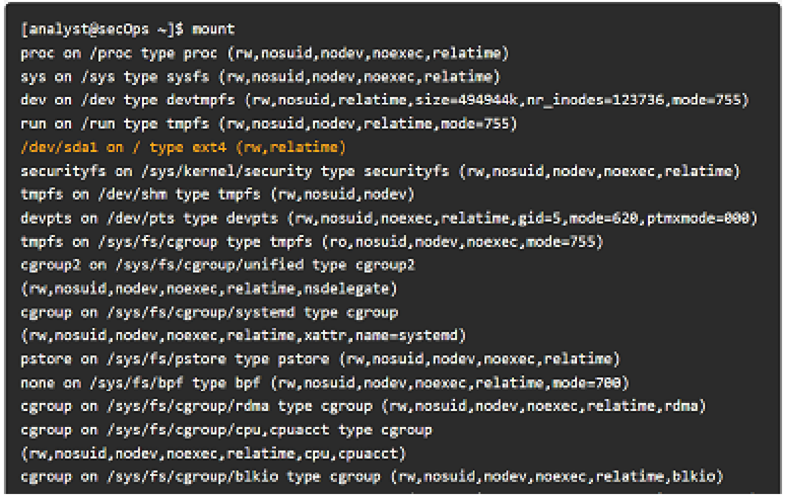
Linux File System	Description
ext2 (second extended file system)	It was the default file system in several major Linux distributions until supplanted by ext3.
ext3 (third extended file system)	It is a journaled file system designed to improve the existing ext2 file system.
ext4 (fourth extended file system)	Designed as a successor of ext3, it was created based on a series of extensions to ext3, but the ext3 project was split in two; one kept as ext3 (normal development) and other, ext4 (the mentioned extensions).
NFS (Network File System)	NFS is a network-based file system, allowing file access over the network.
CDFS (Compact Disc File System)	CDFS was created specifically for optical disk media.
Swap File System	The swap file system is used by Linux when it runs out of RAM.
HFS Plus or HFS+ (Hierarchical File System Plus)	A file system used by Apple in its Macintosh computers. The Linux kernel includes a module for mounting HFS+ for read-write operations.
APFS (Apple File System)	An updated file system that is used by Apple devices.
Master Boot Record (MBR)	Located in the first sector of a partitioned computer, it stores all the information about the way in which the file system is organized.

The File System Types in Linux (Cont.)

- Mounting is the term used for the process of assigning a directory to a partition.
- After a successful mount operation, the file system contained on the partition is accessible through the specified directory.

The command output shows part of the output of the **mount** command issued in the Cisco CyberOPS VM.

- When issued with no options, **mount** returns the list of file systems currently mounted in a Linux computer.
- Notice that the root file system (highlighted) is represented by the “/” symbol and holds all files in the computer by default.
- The output also shows that the root file system was formatted as ext4 and occupies the first partition of the first drive (/dev/sda1).

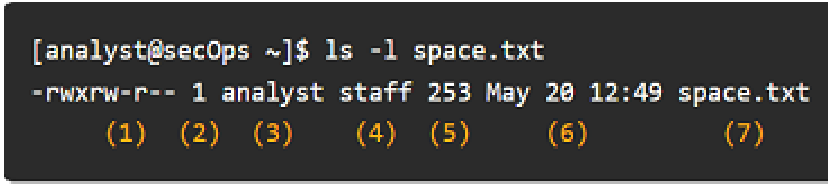


```
[analyst@secOps ~]$ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sys on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
dev on /dev type devtmpfs (rw,nosuid,relatime,size=494944k,nr_inodes=123736,mode=755)
run on /run type tmpfs (rw,nosuid,nodev,relatime,mode=755)
/dev/sda1 on / type ext4 (rw,relatime)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup2 on /sys/fs/cgroup/unified type cgroup2
(rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup
(rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup
(rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
```

The Linux File System

Linux Roles and File Permissions

Consider the output of the **ls -l** command.



```
[analyst@secOps ~]$ ls -l space.txt
-rwxrw-r-- 1 analyst staff 253 May 20 12:49 space.txt
     (1)   (2)   (3)     (4)   (5)     (6)          (7)
```

Field	Description
1	It displays the permissions associated with the file. The dash (-) means that this is a file. For directories, the first dash would be a “d”. The first set of characters is for user permission (rwx) who owns the file. The user (analyst) can Read, Write, and eXecute the file. The second set of characters is for group permissions (rw-) who owns the file. The group (staff) can Read and Write to the file. The third set of characters is for any other user or group permissions (r--). They can only Read the file.
2	It defines the number of hard links to the file (the number 1 after the permissions).
3	It displays the user (analyst).
4	It displays the group (staff).
5	It displays the file size in bytes (253 bytes).
6	It displays the date and time of the last modification (May 20 12:49).
7	It displays the file name (space.txt).

Linux Roles and File Permissions (Cont.)

Use octal values to define permissions.

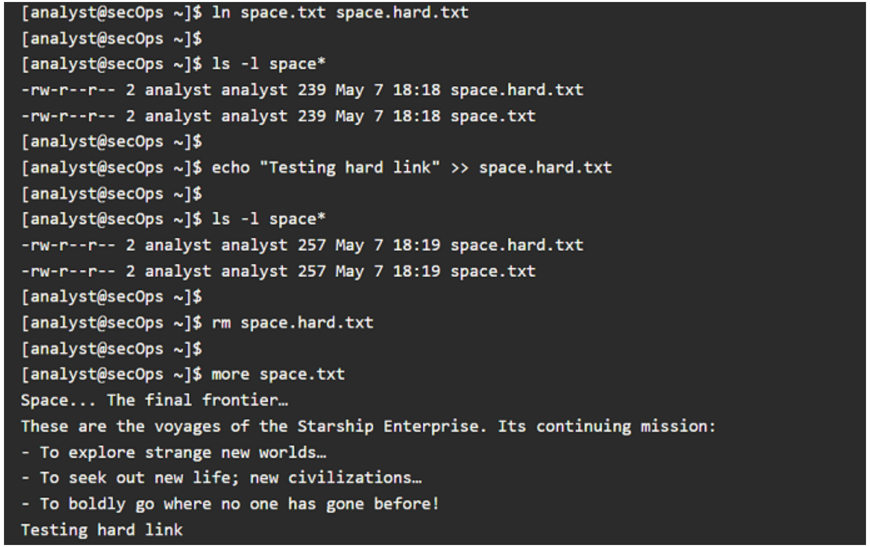
Binary	Octal	Permission	Description
000	0	---	No access
001	1	--x	Execute only
010	2	-w-	Write only
011	3	-wx	Write and Execute
100	4	r--	Read only
101	5	r-x	Read and Execute
110	6	rw-	Read and Write
111	7	rwx	Read, Write, and Execute

File permissions are a fundamental part of Linux and cannot be broken. A user has only the rights to a file that the file permissions allow. The only user that can override file permission on a Linux computer is the root user.

The Linux File System

Hard Links and Symbolic Links

- A hard link is another file that points to the same location as the original file.
- The **ln** command creates a hard link. The first argument is the existing file, and the second one is the new file.
- The command output shows that the file space.txt is linked to space.hard.txt and the link field now shows 2.
- Both files point to the same location in the file system.
- If you change one file, the other is changed as well.
- The **echo** command is used to add some text to space.txt (the file size for both files increased).
- If you delete the new file with the **rm** command, the original one still exists (**more space.txt** command).



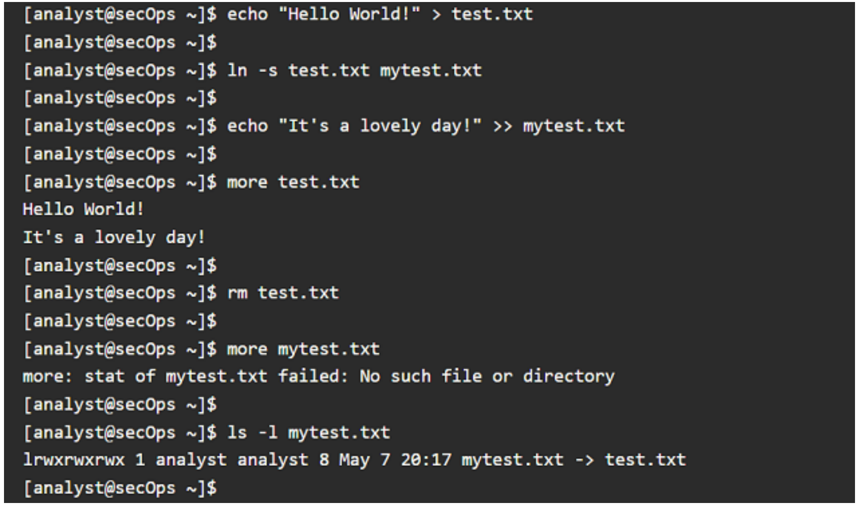
```
[analyst@secOps ~]$ ln space.txt space.hard.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ ls -l space*
-rw-r--r-- 2 analyst analyst 239 May 7 18:18 space.hard.txt
-rw-r--r-- 2 analyst analyst 239 May 7 18:18 space.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ echo "Testing hard link" >> space.hard.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ ls -l space*
-rw-r--r-- 2 analyst analyst 257 May 7 18:19 space.hard.txt
-rw-r--r-- 2 analyst analyst 257 May 7 18:19 space.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ rm space.hard.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ more space.txt
Space... The final frontier…
These are the voyages of the Starship Enterprise. Its continuing mission:
- To explore strange new worlds…
- To seek out new life; new civilizations…
- To boldly go where no one has gone before!
Testing hard link
```

The Linux File System

Hard Links and Symbolic Links (Cont.)

- A symbolic link is like a hard link in that applying changes to the symbolic link will also change the original file.
- The **ln -s** command creates a symbolic link.

- Symbolic links have a single point of failure (the underlying file) but have several benefits over hard links:
    - Locating hard links is more difficult. Symbolic links show the location of the original file **(ls -l** command).
    - Symbolic links can link to a file in another file system.
    - Symbolic links can link to directories.



```
[analyst@secOps ~]$ echo "Hello World!" > test.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ ln -s test.txt mytest.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ echo "It's a lovely day!" >> mytest.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ more test.txt
Hello World!
It's a lovely day!
[analyst@secOps ~]$
[analyst@secOps ~]$ rm test.txt
[analyst@secOps ~]$
[analyst@secOps ~]$ more mytest.txt
more: stat of mytest.txt failed: No such file or directory
[analyst@secOps ~]$
[analyst@secOps ~]$ ls -l mytest.txt
lrwxrwxrwx 1 analyst analyst 8 May 7 20:17 mytest.txt -> test.txt
[analyst@secOps ~]$
```

Lab - Navigating the Linux Filesystem and Permission Settings

In this lab, you will familiarize yourself with Linux filesystems.

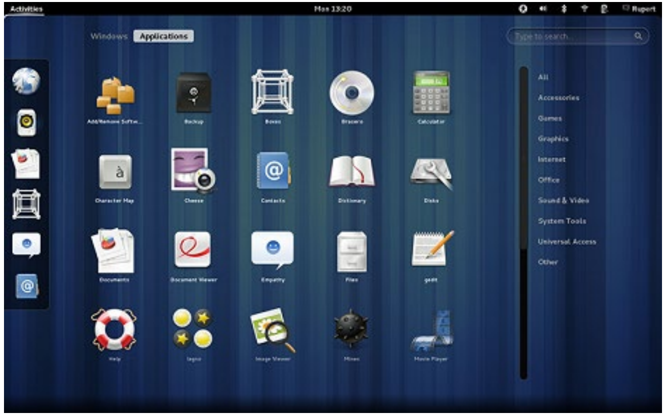
- Part 1: Exploring Filesystems in Linux
- Part 2: File Permissions
- Part 3: Symbolic Links and other Special File Types

8.6 Working with the Linux GUI

Working with the Linux GUI

X Window System

- The graphical interface present in most Linux computers (X or X11) is based on the X Window System.
- X Window is designed to provide the basic framework for a GUI.
- It includes functions for drawing and moving windows on the display device and interacting with a mouse and keyboard.
- X works as a server which allows a remote user to use the network to connect, start a graphical application, and have the graphical window open on the remote terminal.
- While the application itself runs on the server, the graphical aspect of it is sent by X over the network and displayed on the remote computer.
- X does not specify the user interface, leaving it to other programs, such as window managers, to define all the graphical components.



The Linux GUI

- GUIs are considered more user-friendly than the CLI. This module focuses on Ubuntu when covering Linux because it is a very popular and user-friendly distribution.
- Ubuntu Linux uses Gnome 3 as its default GUI. The goal of Gnome 3 is to make Ubuntu even more user-friendly. The table lists the main UI components of Unity.

UI Component	Description
Apps Menu	It shows icons for the apps that are installed on the system. A right-click menu provides shortcuts that allow starting or configuring the apps.
Ubuntu Dock	This is a dock on the left side of the screen that serves as an application launcher and switcher for app favorites. Click to launch an application and when the application is running, click again to switch between running applications.
Top Bar	This multipurpose menu bar contains a menu for the application that currently has the focus. It displays the current time and indicates whether there are new system messages.
Calendar and System Message Tray	Click the day and time to see the full appointment calendar and any current system messages. Access the appointment calendar from here to create new appointments.
Activities	Switch to application view to switch to or close running applications.
Status Menu	Allows configuration of the network adaptor and other running devices.

8.7 Working on a Linux Host

Installing and Running Applications on a Linux Host

- Many end-user applications are complex programs written in compiled languages. To aid in the installation process, Linux often includes programs called package managers.

- A package is the term used to refer to a program and all its supporting files. By using a package manager to install a package, all the necessary files are placed in the correct file system location.

- Package managers vary depending on Linux distributions. Pacman is used by Arch Linux while dpkg (Debian package) and apt (Advanced Packaging Tool) are used in Debian and Ubuntu Linux distributions.

- The **apt-get update** command is used to get the package list from the package repository and update the local package database.

- The **apt-get upgrade** command is used to update all currently installed packages to their latest versions.

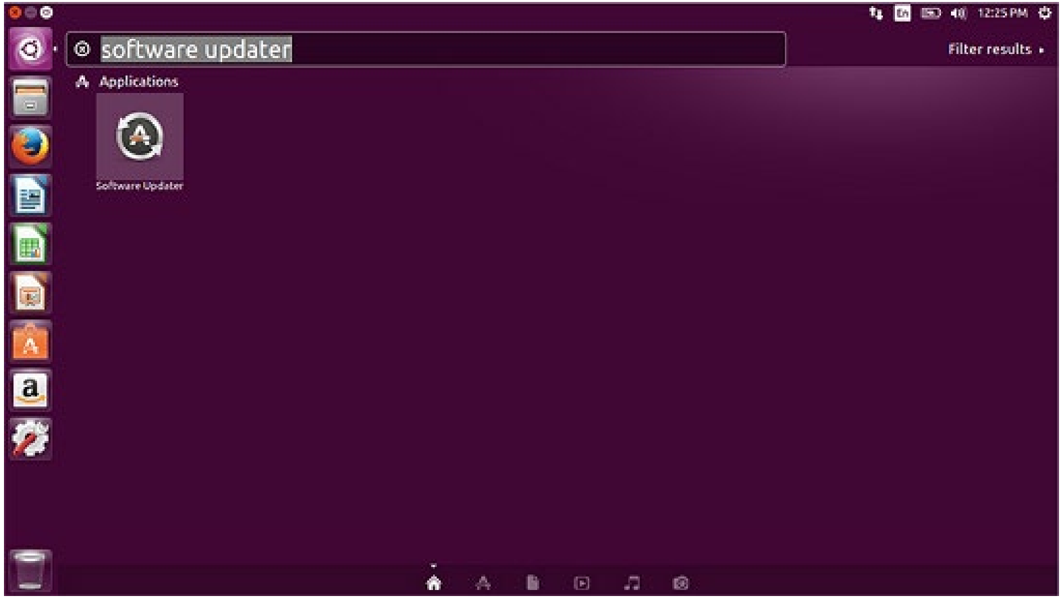
Keeping the System Up to Date

- Also known as patches, OS updates are released periodically by OS companies to address any known vulnerabilities in their operating systems.
- While companies have update schedules, the release of unscheduled OS updates can happen when a major vulnerability is found in the OS code.
- Modern operating systems will alert the user when updates are available for download and installation, but the user can check for updates at any time.
- The table compares Arch Linux and Debian / Ubuntu Linux distribution commands to perform package system basic operations.

Task	Arch	Debian / Ubuntu
Install a package by name	pacman -S	apt install
Remove a package by name	pacman -Rs	apt remove
Update a local package	pacman -Syy	apt-get update
Upgrade all currently installed packages	pacman -Syu	apt-get upgrade

Keeping the System Up to Date (Cont.)

- A Linux GUI can also be used to manually check and install updates.
- In Ubuntu for example, to install updates you would click Dash Search Box, type software updater, and then click the Software Updater icon, as shown in the figure.



software updater

Working on a Linux Host

Processes and Forks

- Multitasking operating systems can execute many processes at the same time.
- Processes need a way to create new processes in multitasking operating systems. The fork operation is the only way of doing so in Linux.
- When a process calls a fork, the caller process becomes the parent process, with the newly created process referred to as its child.
- After the fork, the processes are, to some extent, independent processes; they have different process IDs but run the same program code. The table lists three commands that are used to manage processes.

Command	Description
task manager ps	Used to list the processes running on the computer at the time it is invoked. It can be instructed to display running processes that belong to the current user or other users.
top	Used to list running processes, but unlike ps, top keeps displaying running processes dynamically. Press q to exit top.
kill	Used to modify the behavior of a specific process. Depending on the parameters, kill will remove, restart, or pause a process.

Malware on a Linux Host

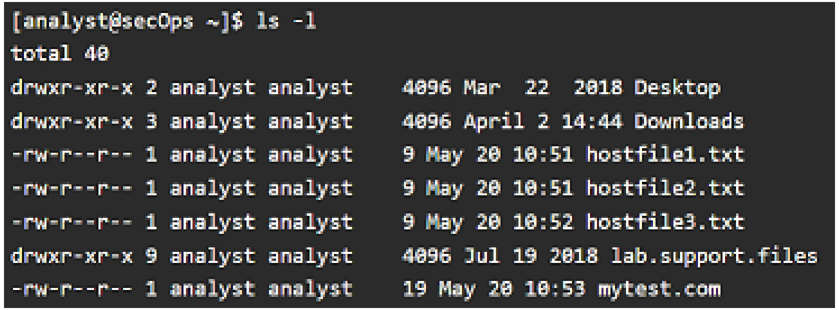
- Linux malware includes viruses, Trojan horses, worms, and other types of malware that can affect the operating system.
- Due to some design components such as file system structure, file permissions, and user account restrictions, Linux operating systems are generally regarded as better protected against malware.
- Linux is not immune to malware. Many vulnerabilities have been found and exploited in Linux.
- Because Linux is open source, fixes and patches are often made available within hours of the discovery of such problems.
- If a malicious program is executed, it will cause damage, regardless of the platform. A common Linux attack vector is its services and processes.
- As with most vulnerabilities, keeping the computer updated and closing any unused services and ports is a good way to reduce the opportunities for attack in a Linux computer.

Rootkit Check

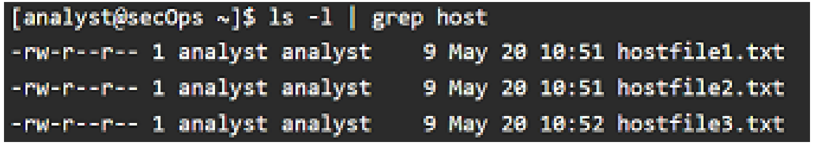
- A rootkit is a type of malware that is designed to increase an unauthorized user’s privileges or grant access to portions of the software that should not normally be allowed.
- Its installation can be automated (done as part of an infection) or manually.
- It changes kernel code and its modules, changing the most fundamental operations of the OS itself.
- Most of the rootkit compromises require root or administrator access.
- Because the very nature of the computer is compromised, rootkit detection can be very difficult.
- Inspection methods include behavioral-based methods, signature scanning, difference scanning, and memory dump analysis.
- Rootkit removal can be complicated and often impossible, especially in cases where the rootkit resides in the kernel; re-installation of the operating system is usually the only real solution.
- Firmware rootkits usually require hardware replacement.
- **chkrootkit** is a popular Linux-based program designed to check the computer for known rootkits.
- While helpful, keep in mind that programs to check for rootkits are not 100% reliable.

Piping Commands

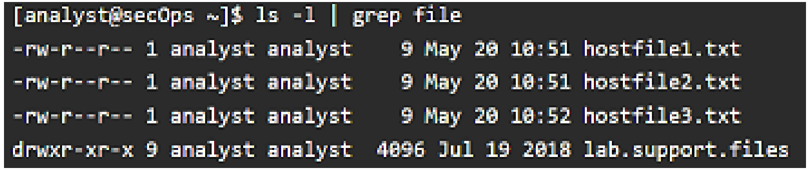
- Many commands can be combined to perform more complex tasks by a technique known as piping.
- Piping consists of chaining commands together, feeding the output of one command into the input of another using the character "|" (pipe).
- For example, the **ls** command is used to display all the files and directories of a given directory. The **grep** command compares searches through a file or text looking for the specified string.
- The two commands, **ls** and **grep**, can be piped together to filter out the output of **ls**. This is shown in the output of the **ls -l | grep host** command and the **ls -l | grep file** command.



```
[analyst@secOps ~]$ ls -l
total 40
drwxr-xr-x 2 analyst analyst    4096 Mar   22   2018 Desktop
drwxr-xr-x 3 analyst analyst    4096 April 2 14:44 Downloads
-rw-r--r-- 1 analyst analyst    9 May  20 10:51 hostfile1.txt
-rw-r--r-- 1 analyst analyst    9 May  20 10:51 hostfile2.txt
-rw-r--r-- 1 analyst analyst    9 May  20 10:52 hostfile3.txt
drwxr-xr-x 9 analyst analyst    4096 Jul 19 2018 lab.support.files
-rw-r--r-- 1 analyst analyst    19 May  20 10:53 mytest.com
```



```
[analyst@secOps ~]$ ls -l | grep host
-rw-r--r-- 1 analyst analyst    9 May 20 10:51 hostfile1.txt
-rw-r--r-- 1 analyst analyst    9 May 20 10:51 hostfile2.txt
-rw-r--r-- 1 analyst analyst    9 May 20 10:52 hostfile3.txt
```



```
[analyst@secOps ~]$ ls -l | grep file
-rw-r--r-- 1 analyst analyst    9 May 20 10:51 hostfile1.txt
-rw-r--r-- 1 analyst analyst    9 May 20 10:51 hostfile2.txt
-rw-r--r-- 1 analyst analyst    9 May 20 10:52 hostfile3.txt
drwxr-xr-x 9 analyst analyst  4096 Jul 19 2018 lab.support.files
```

Working on a Linux Host

Video - Applications, Rootkits, and Piping Commands

This video will cover the following:

- A demonstration of installing and updating applications.
- Checking for a rootkit.
- Use of piping commands.

Lab - Configure Security Features in Windows and Linux

In this lab, you will create restore points and backups for use in Windows and Linux systems. Furthermore, you will configure Windows Defender Firewall to allow desired traffic. In a Linux system, you will configure the Uncomplicated Firewall (UFW) to block unsecure Telnet traffic. You will also disable Telnet services.

- Part 1: Create Backups and Restore Points
- Part 2: Configure Firewall Rules
- Part 3: Disable Services
- Part 4: Restore

8.8 Linux Basics Summary

What Did I Learn in this Module?

- Linux is an open-source operating system that is fast, powerful, and highly customizable.
- It was created and is currently maintained by a community of programmers.
- It is designed to be connected to the network, which makes it much simpler to write and use network-based applications.
- A Linux distribution (distro) describes packages created by different organizations. It includes the Linux kernel with customized tools and software packages.
- The flexibility provided by Linux is a great feature for the SOC. The entire operating system can be tailored to become the perfect security analysis platform.
- In Linux, the user communicates with the OS by using the CLI or the GUI (default).
- Many commands line tools are included in Linux by default. To adjust the command operation, user can pass parameters and switches along with the command.
- Linux has many different text editors, with various features and functions.
- In Linux, everything is treated as a file (memory, disks, monitor, and the directories).
- In Linux, services are managed using configuration files. When the service starts, it looks for its configuration files, loads them into memory, and adjust itself according to the settings in the files.
- Log files are the records that a computer stores to keep track of important events. In Linux, log files can be categorized as application logs, event logs, service logs, and system logs.