



# Chapter 3: Linux Operating System

Cybersecurity Operations v1.1



# Chapter 3 - Sections & Objectives

## ■ 3.1 Linux Overview

- Perform basic operations in the Linux shell.
  - Explain why Linux skills are essential for network security monitoring and investigation.
  - Use the Linux shell to manipulate text files.
  - Explain how client-server networks function.

## ■ 3.2 Linux Administration

- Perform basic Linux administration tasks.
  - Explain how a Linux administrator locates and manipulates security log files..
  - Manage the Linux file system and permissions.

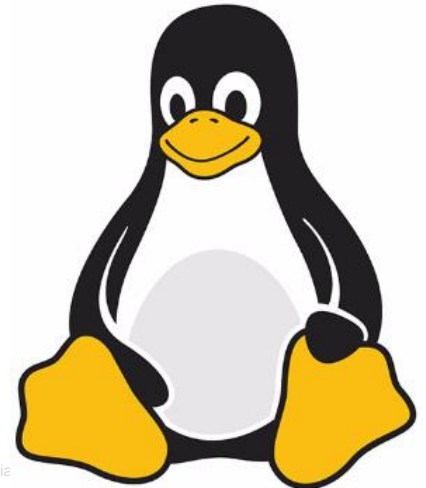
## ■ 3.3 Linux Hosts

- Perform basic security-related tasks on a Linux host.
  - Explain the basic components of the Linux GUI.
  - Use tools to detect malware on a Linux host.

# 3.1 Linux Overview

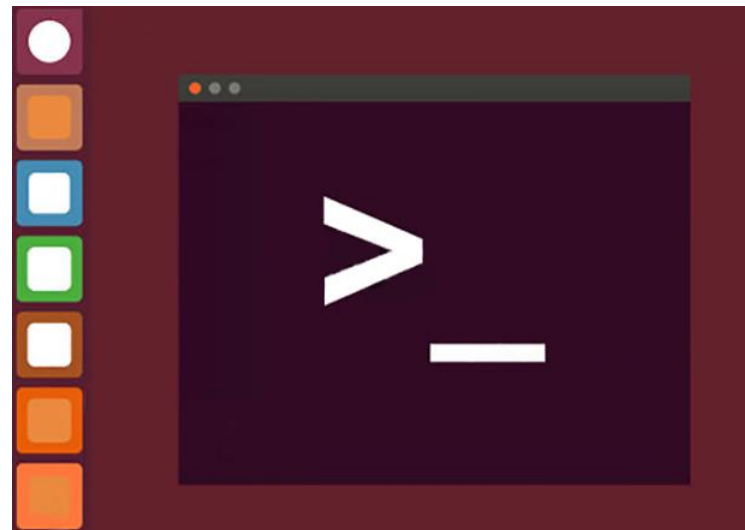
# What is Linux?

- Linux is an Open Source operating system created in 1991 and maintained by a community of programmers.
- Open Source means the source programming files, including the kernel, shell, and applications are available for downloading, viewing and modification.
- Linux was designed as a network operating system and is widely used on different platforms including embedded systems.
- There are many different versions or distributions of Linux.  
A distribution is defined by its kernel, as well as its programs and software packaging.
- Some Linux distributions are free, like CentOS and Fedora.  
Others like RedHat Enterprise Server, cost money, but include support services.



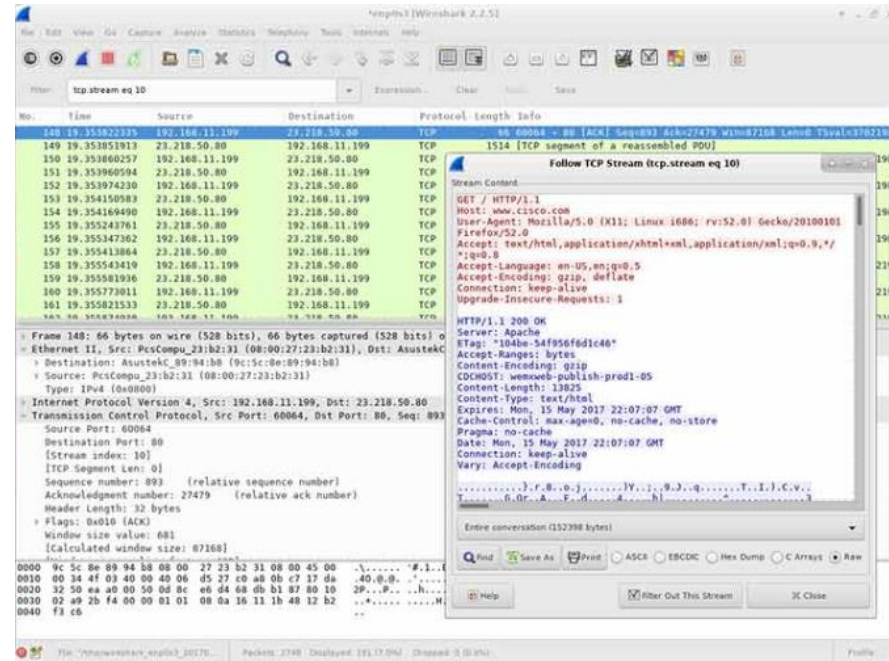
# The Value of Linux

- Linux is an operating system of choice in Security Operations Center (SOC).
  - Open source
    - Allows analysts and administrators to tailor-build the OS.
  - Command Line Interface (CLI) is very powerful
    - Enables analysts to perform tasks directly or remotely on a terminal.
  - More user control over the OS
    - Root user or superuser has absolute power over the computer.
      - Modify any aspect of the computer.
      - Precise control over the functions of the computer.
  - Better network communication control
    - Great platform for creating network application.



# Linux in the SOC

- A custom security distribution of Linux can be created for the SOC with just the tools needed for the job.
- Packet Capture (Wireshark)
- Malware Analysis Tools
- Intrusion Detection Systems (IDSs)
- Firewalls
- Log Managers
- Security Information and Event Management (SIEM)
- Ticketing Systems



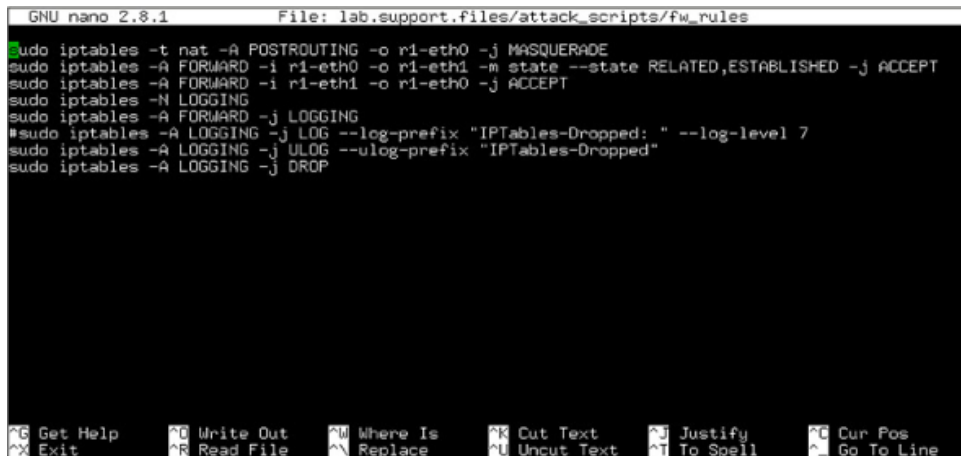
# Linux Tools

- Penetration testing tools
  - Process of looking for vulnerabilities.
  - Tool examples:
    - Packet generators
    - Port scanners
    - Proof-of-concept exploits
- Kali Linux distribution groups many penetration tools.



# Working with Text Files

- There are many text editors available in Linux.
- Some text editors are for the CLI only, like vi, vim, and nano.
- Other text editors, like gedit, are GUI-based.
- CLI text editors allow system management remotely, such as via SSH.



```
GNU nano 2.8.1 File: lab.support.files/attack_scripts/fw_rules
sudo iptables -t nat -A POSTROUTING -o r1-eth0 -j MASQUERADE
sudo iptables -A FORWARD -i r1-eth0 -o r1-eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i r1-eth1 -o r1-eth0 -j ACCEPT
sudo iptables -N LOGGING
sudo iptables -A FORWARD -j LOGGING
#sudo iptables -A LOGGING -j LOG --log-prefix "IPTables-Dropped: " --log-level 7
sudo iptables -A LOGGING -j ULOG --ulog-prefix "IPTables-Dropped"
sudo iptables -A LOGGING -j DROP
```

The screenshot shows the GNU nano 2.8.1 text editor interface. The title bar indicates the file being edited is 'lab.support.files/attack\_scripts/fw\_rules'. The editor contains several lines of iptables commands. The first line is 'sudo iptables -t nat -A POSTROUTING -o r1-eth0 -j MASQUERADE'. The second line is 'sudo iptables -A FORWARD -i r1-eth0 -o r1-eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT'. The third line is 'sudo iptables -A FORWARD -i r1-eth1 -o r1-eth0 -j ACCEPT'. The fourth line is 'sudo iptables -N LOGGING'. The fifth line is 'sudo iptables -A FORWARD -j LOGGING'. The sixth line is a commented-out line: '#sudo iptables -A LOGGING -j LOG --log-prefix "IPTables-Dropped: " --log-level 7'. The seventh line is 'sudo iptables -A LOGGING -j ULOG --ulog-prefix "IPTables-Dropped"'. The eighth line is 'sudo iptables -A LOGGING -j DROP'. The bottom status bar shows various nano editor shortcuts: 'Get Help', 'Exit', 'Write Out', 'Read File', 'Where Is', 'Replace', 'Cut Text', 'Uncut Text', 'Justify', 'To Spell', 'Cur Pos', and 'Go To Line'.



# The Importance of Text Files in Linux

- In Linux, everything is treated as a file, this includes the memory, the disks, the monitor, the files, and the directories.
- The operating system as well as most programs are configured by editing the configuration files which are text files.
- Editing system or application configuration files requires super user (root) privileges. This can be accomplished with the sudo command.

A screenshot of the nano 2.7.4 text editor in a terminal window. The title bar shows 'GNU nano 2.7.4' and 'File: /etc/hosts'. The editor content shows the /etc/hosts file with entries for 127.0.0.1 (localhost) and 127.0.1.1 (kali), followed by a comment and IPv6 addresses. A large, faint Kali Linux dragon logo is visible in the background. The bottom status bar shows various nano editor commands like 'Get Help', 'Write Out', 'Where Is', 'Cut Text', 'Justify', 'Cur Pos', 'Exit', 'Read File', 'Replace', 'Uncut Text', 'To Spell', and 'Go To Line'. A small box above the status bar indicates '[ Read 7 lines ]'.

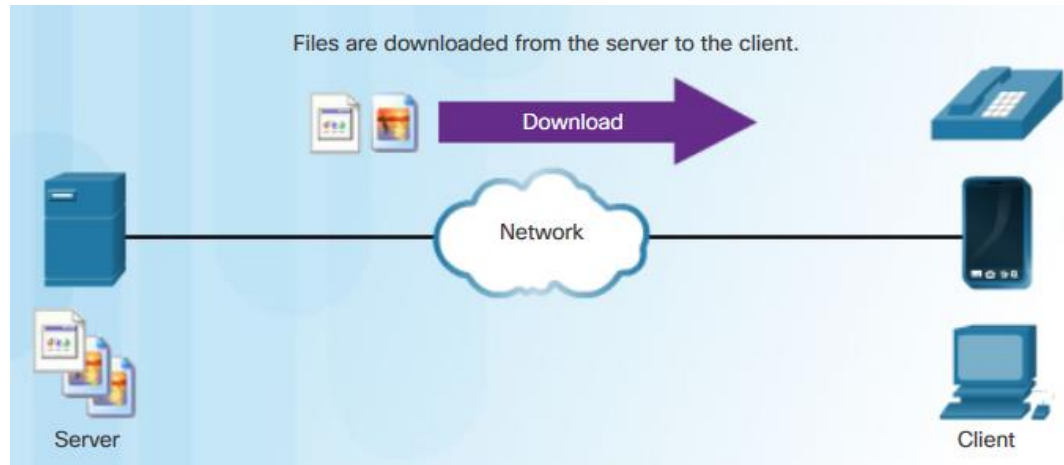
```
GNU nano 2.7.4                                File: /etc/hosts
127.0.0.1    localhost
127.0.1.1    kali

# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters

```

# An Introduction to Client-Server Communications

- Servers are computers with software installed that enable them to provide services to clients.
- Resources, such as files, email messages, or web pages, are stored on the server.
- Servers can also provide services, such as log management, memory management, and disk scanning.
- The client software is designed to communicate with the server.



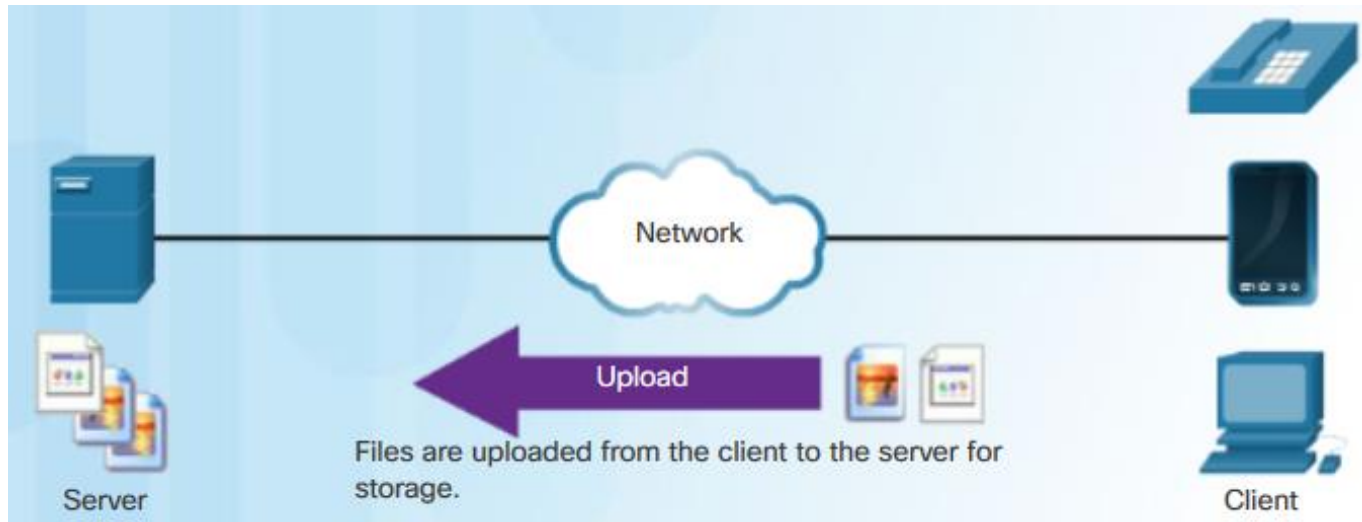
# Servers, Services, and Their Ports

- A port is a reserved network resource used by a service.
- An administrator can assign a port to a specific service or use the default port number.

Default Port Number	Service
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)
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 server.
- Client applications are used for a well-defined protocol:
  - File Transfer Protocol (FTP)
  - Hyper Text Transfer Protocol (HTTP)



## 3.2 Linux Administration

# Service Configuration Files

- Linux Servers are typically configured with text-based configuration files.
- The configuration file defines options for the service, such as port number, location of hosted resources, and client authorization details.
- A server configuration file often consists of important server settings in the form of variables in key=value pairs.
- A server configuration file usually has instructions that begin with a comment like a hash #. Comments are ignored by the software.

```
[analyst@secops ~]$ cat /etc/ntp.conf
cat: /etc/ntp.conf: No such file or directory
# Please consider joining the pool:
#
#   http://www.pool.ntp.org/join.html
#
# For additional information see:
# - https://wiki.archlinux.org/index.php/Network_Time_Protocol_daemon
# - http://support.ntp.org/bin/view/Support/GettingStarted
# - the ntp.conf man page

# Associate to Arch's NTP pool
server 0.arch.pool.ntp.org
server 1.arch.pool.ntp.org
server 2.arch.pool.ntp.org
server 3.arch.pool.ntp.org

# By default, the server allows:
# - all queries from the local host
# - only time queries from remote hosts, protected by rate limiting and kod
restrict default kod limited nomodify nopeer noquery notrap
restrict 127.0.0.1
restrict ::1

# Location of drift file
driftfile /var/lib/ntp/ntp.drift
[analyst@secops ~]$
```

## Basic Server Administration

# Hardening Devices

- 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
- Review logs regularly

# Monitoring Service Logs

- Log files are records to keep track of important computer events.
- Linux has the following types of logs:
  - Application Logs
  - Event Logs
  - Service Logs
  - System Logs

Log	Purpose
/var/log/messages	Used to store informational and non-critical system messages
/var/log/auth.log	Stores all authentication-related events
/var/log/secure	Used by RedHat and CentOS and tracks sudo logins, SSH logins, and errors logged by SSSD
/var/log/boot.log	Stores boot related messages during startup
/var/log/dmesg	Contains kernel ring bugger messages
/var/log/kern.log	Contains information logged by the kernel
/var/log/cron	A service used for scheduling automated tasks in Linux
/var/log/mysqld.log or /var/log/mysql.log	Logs all debug, failure and success messages related to the mysql process and mysqld_safe daemon



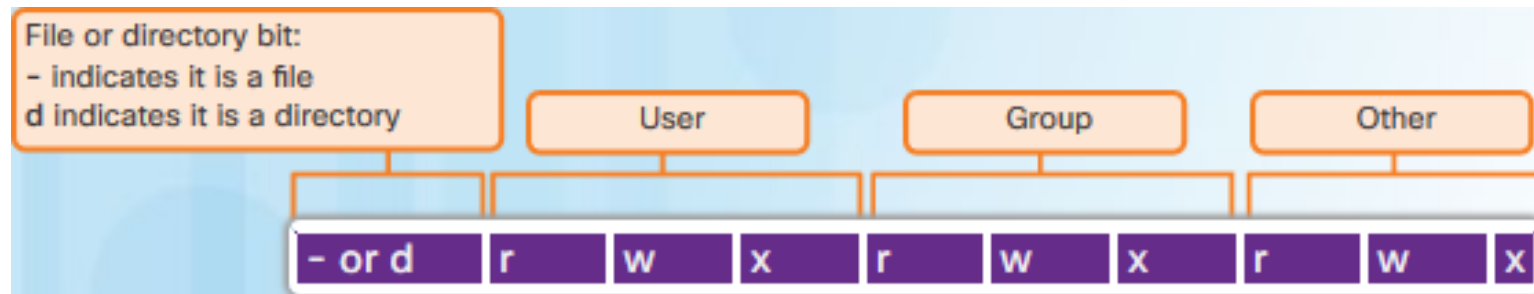
# The File System Types in Linux

File System Type	Description
ext2 (second extended file system)	Is the file system of choice for flash-based storage media.
ext3 (third extended file system)	Is an improved successor to ext2 with the additional feature of journaling of all the file system changes.
ext4 (fourth extended file system)	Is designed as a successor to ex3 with increased support file sizes and better performance than ext3.
NFS (Network File System)	Is a network-based file system, allowing file access over the network.
CDFS (Compact Disc File System)	Was created specifically for optical disk media.
Swap File System	Is used when the system runs out of RAM.
HFS+ (Hierarchical File System Plus)	Is the primary file system used by Apple in its Macintosh computers.
APFS (Apple File System)	An updated file system used by Apple devices that provides strong encryption and is optimized for flash and solid state drives.
Master Boot Record (MBR)	Is located in the first sector of a partitioned computer and stores all the information about the way the file system is organized.

# Linux Roles and File Permissions

In octal (3bits), per permission (i.e. 111 is a 7 for read, write and execute)

- **User** - the file owner's permission
- **Group** - a group's permission to a file
- **Other** – any user, non-owner's permission to a file
- **Read** – the ability to look at a file's contents
- **Write** – the ability to change a files contents
- **Execute** – the ability to run or launch a file (scripts and programs)



# Hard Links and Symbolic Links

- The *ln* command make links between files.
- Hard Links:
  - Points to the same location as the original file.
  - Changes one file, the other one also changes.
- Symbolic or Soft Links:
  - Uses the -s option in the command to create the symbolic link.
  - Delete the original file, the soft link is link to the original file that no longer exists.
- Advantages to symbolic link:
  - Locating hard links is more difficult.
  - Hard links are limited to the file system in which they are created. Symbolic links can link to a file in another file system.
  - Hard links cannot link to a directory, but symbolic links can.

# 3.3 Linux Hosts

# Working with the Linux GUI

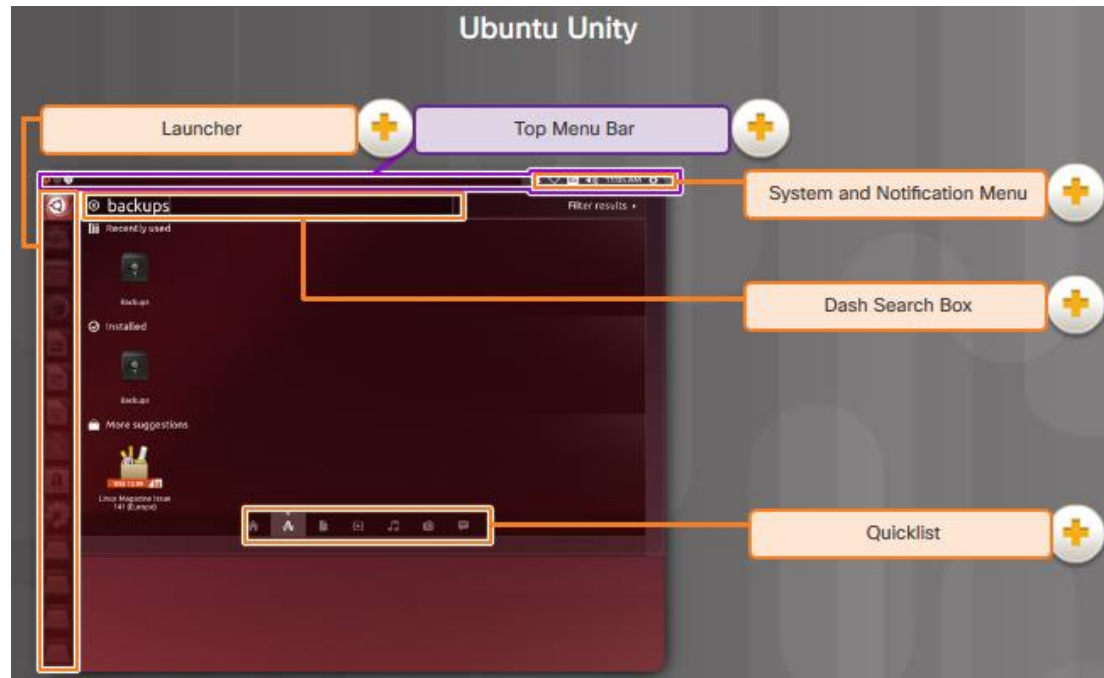
## X Windows System

- X Window System is the framework for the Linux GUI also known as X and X11.
- Functions for drawing and moving the window, as well as interacting with the mouse and keyboard.
- X works as a server and can send the graphical window over a network to a remote computer.
- X does not specify the user interface or desktop. That is left to a window manager to define the look and feel of the GUI.
- Gnome and KDE are examples of popular Linux window managers.



# Working with the Linux GUI

## The Linux GUI



- **Top Menu Bar** – currently running application
- **Launcher** – serves as the application launcher and switcher
- **Quicklist** - Right-click any application hosted on the Launcher to access a short list of tasks the application can perform.
- **Dash Search Box** - holds the Search tool and a list of recently used applications.
- **System and Notification Menu** – Can be used to switch users, shut down your computer, control the volume level, or change network settings.

# Installing and Running Applications on a Linux Host

- The Installation and removal of programs in Linux is simplified by using a package manager.
- Linux package managers maintain lists of available software and their dynamic library dependencies and requirements.
- Popular package managers are APT for Debian packages (dpkg) and Yum for RedHat packages (rpm).



## Keeping the System Up to Date

- **apt-get update** – downloads the list of available software from the distribution repository and updates the local package database.
- **apt-get upgrade** – downloads and upgrades all of the installed software applications on the system.

```
analyst@cuckoo:~$ sudo apt-get update
[sudo] password for analyst:
Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Get:3 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages
[534 kB]
```



# Processes and Forks

- A process is a running instance of a computer program. Multitasking operating systems can execute many processes at the same time.
- Forking is a method that the kernel uses to allow a process to create a copy of itself to provide process scalability.
- Some commands to manage processes:
  - **ps** – list processes running on the system
  - **top** – list running processes dynamically
  - **kill** – modify the behavior of a specific process, such as remove, restart or pause a process

# Malware on a Linux Host

- Linux is generally considered more resistant to malware than other operating systems but it is still not immune.
- A Linux server is only as secure as the programming behind its services and applications.

```
[analyst@secOps ~]$ telnet 209.165.200.224 80
Trying 209.165.200.224...
Connected to 209.165.200.224.
Escape character is '^]'.
type anything to force an HTTP error response
HTTP/1.1 400 Bad Request
Server: nginx/1.12.0
Date: Wed, 17 May 2017 14:27:30 GMT
Content-Type: text/html
Content-Length: 173
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.12.0</center>
</body>
</html>
Connection closed by foreign host.
[analyst@secOps ~]$
```

# Rootkit Check

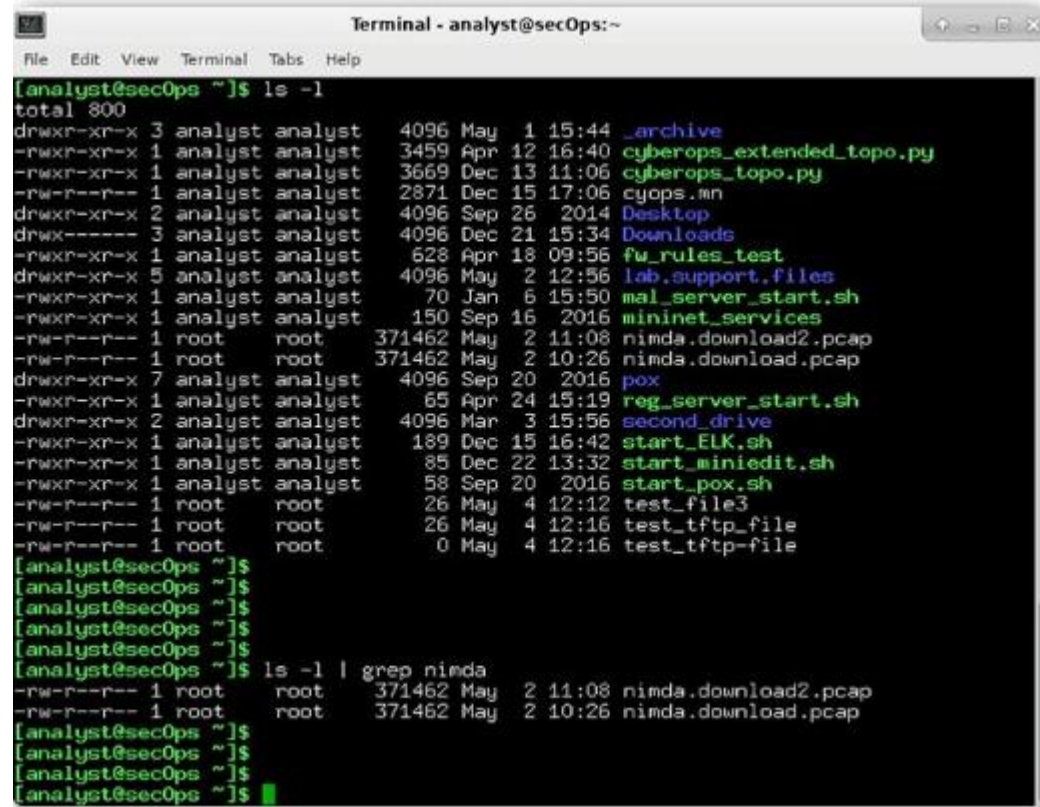
- Rootkits are installed into the operating system kernel and are often used to establish hidden backdoors.
- chkrootkit is a program that will check for rootkits and remove them.
- Rootkit removal can be complicated and often impossible, especially in cases where the rootkit resides in the kernel; re-installation of the operating system is usually the only real solution to the problem.

```
analyst@cuckoo:~$ sudo ./chkrootkit
[sudo] password for analyst:
ROOTDIR is '/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `crontab'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not found
Checking `gpm'... not found
Checking `grep'... not infected
Checking `hdparm'... not infected
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'... not tested
Checking `inetdconf'... not found
Checking `identd'... not found
Checking `init'... not infected
Checking `killall'... not infected
```

# Working on a Linux Host

## Piping Commands

- Many commands can be combined to perform more complex tasks by a technique known as piping.
- the pipe (|)
- Piping consists of chaining commands together, feeding the output of one command into the input of another.



```
Terminal - analyst@secOps:~  
File Edit View Terminal Tabs Help  
[analyst@secOps ~]$ ls -l  
total 800  
drwxr-xr-x 3 analyst analyst 4096 May 1 15:44 _archive  
-rwxr-xr-x 1 analyst analyst 3459 Apr 12 16:40 cyberops_extended_topo.py  
-rwxr-xr-x 1 analyst analyst 3669 Dec 13 11:06 cyberops_topo.py  
-rw-r--r-- 1 analyst analyst 2871 Dec 15 17:06 cyops.mn  
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop  
drwx----- 3 analyst analyst 4096 Dec 21 15:34 Downloads  
-rwxr-xr-x 1 analyst analyst 628 Apr 18 09:56 fw_rules_test  
drwxr-xr-x 5 analyst analyst 4096 May 2 12:56 lab.support.files  
-rwxr-xr-x 1 analyst analyst 70 Jan 6 15:50 mal_server_start.sh  
-rwxr-xr-x 1 analyst analyst 150 Sep 16 2016 mininet_services  
-rw-r--r-- 1 root root 371462 May 2 11:08 nimda.download2.pcap  
-rw-r--r-- 1 root root 371462 May 2 10:26 nimda.download.pcap  
drwxr-xr-x 7 analyst analyst 4096 Sep 20 2016 pox  
-rwxr-xr-x 1 analyst analyst 65 Apr 24 15:19 reg_server_start.sh  
drwxr-xr-x 2 analyst analyst 4096 Mar 3 15:56 second_drive  
-rwxr-xr-x 1 analyst analyst 189 Dec 15 16:42 start_ELK.sh  
-rwxr-xr-x 1 analyst analyst 85 Dec 22 13:32 start_miniedit.sh  
-rwxr-xr-x 1 analyst analyst 58 Sep 20 2016 start_pox.sh  
-rw-r--r-- 1 root root 26 May 4 12:12 test_file3  
-rw-r--r-- 1 root root 26 May 4 12:16 test_tftp_file  
-rw-r--r-- 1 root root 0 May 4 12:16 test_tftp_file  
[analyst@secOps ~]$  
[analyst@secOps ~]$  
[analyst@secOps ~]$  
[analyst@secOps ~]$  
[analyst@secOps ~]$  
[analyst@secOps ~]$ ls -l | grep nimda  
-rw-r--r-- 1 root root 371462 May 2 11:08 nimda.download2.pcap  
-rw-r--r-- 1 root root 371462 May 2 10:26 nimda.download.pcap  
[analyst@secOps ~]$  
[analyst@secOps ~]$  
[analyst@secOps ~]$  
[analyst@secOps ~]$
```

