Chapter 2: Linux Operating System

Information Security

Dr. Ayman Aljarbouh

Chapter 2 - Sections & Objectives

- 2.1 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.
- 2.2 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.
- 2.3 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.

Chapter 2 - Sections & Objectives

- 2.1 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.
- 2.2 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.
- 2.3 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.

Today

2.3 Linux Hosts

Module Objectives

Module Title: Linux Hosts

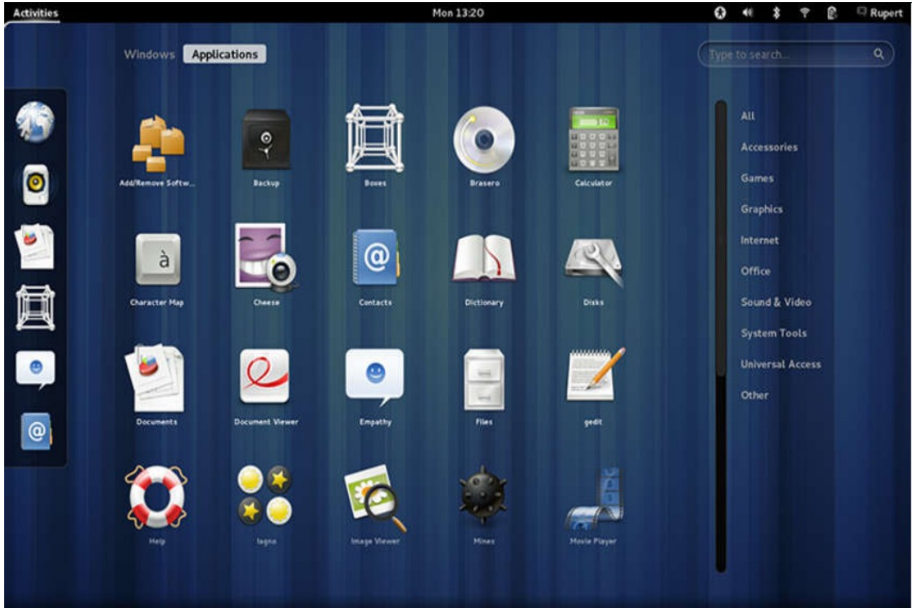
Module Objective: Perform basic security-related tasks on a Linux host.

Topic Title	Topic Objective
Working with 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.

Working with the Linux GUI

X Window System

- The graphical interface present in most Linux computers is based on the X Window System.
- X includes functions for drawing and moving windows on the display device and interacting with a 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, leaving it to other programs, such as window managers, to define all the graphical components.

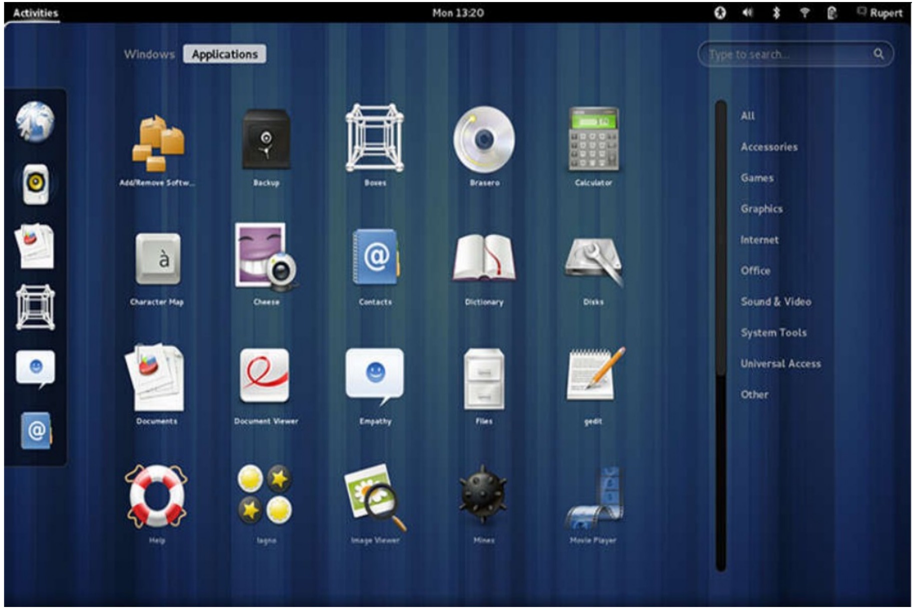


The Gnome Window Manager

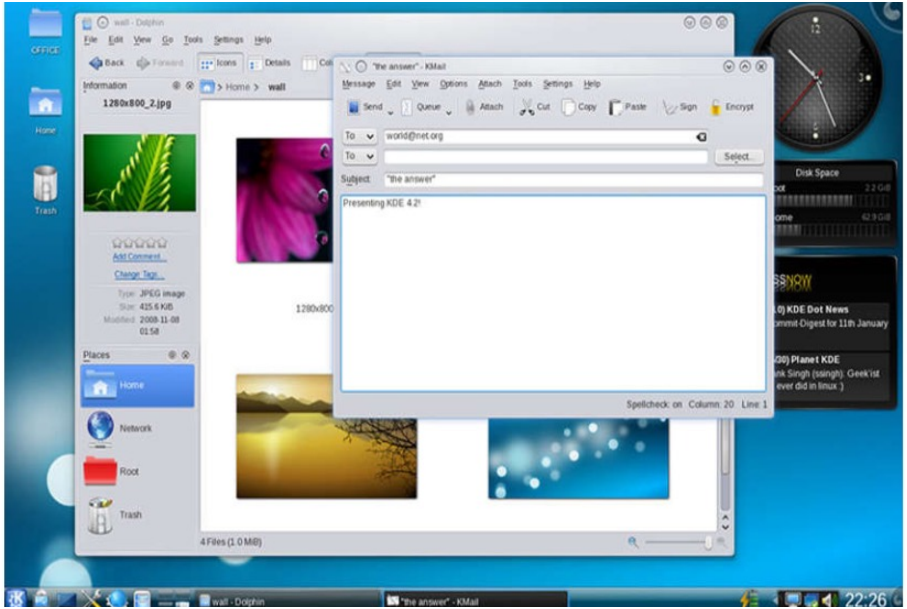
Working with the Linux GUI

X Window System (Contd.)

Examples of window managers are Gnome and KDE.



The Gnome Window Manager



To: world@net.org

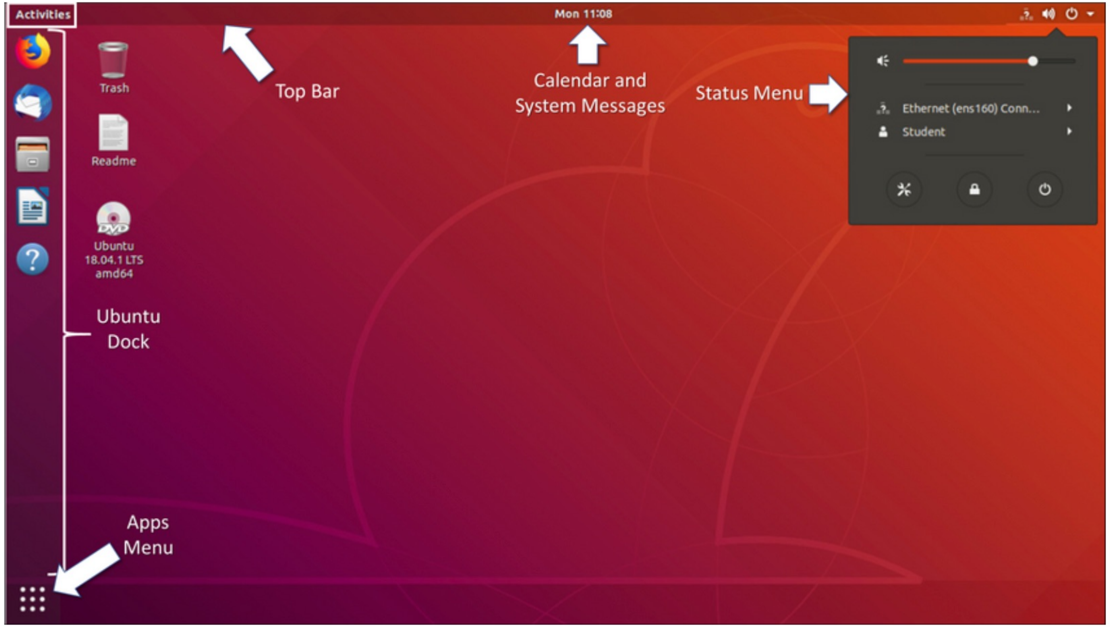
Subject: "the answer"

Presenting KDE 4.2!

The KDE Window Manager

Working with the Linux GUI

The Linux GUI

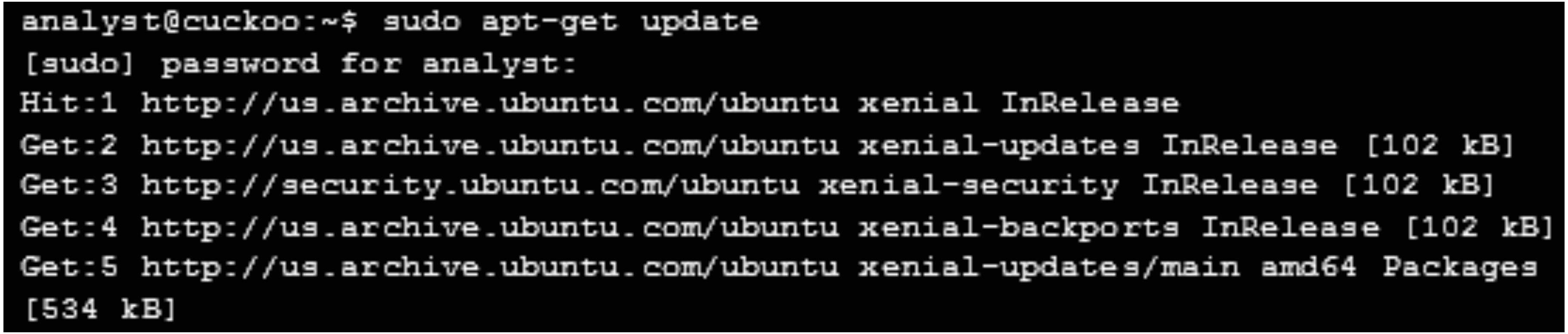


- **Apps Menu** – the apps that are installed on the system.
- **Ubuntu Dock** – serves as the application launcher and switcher.
- **Top Bar -** contains a menu for the application that currently has the focus.
- **Calendar and System Message Tray -** Access the appointment calendar from here to create 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.

Working on a Linux Host

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).



```
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]
```

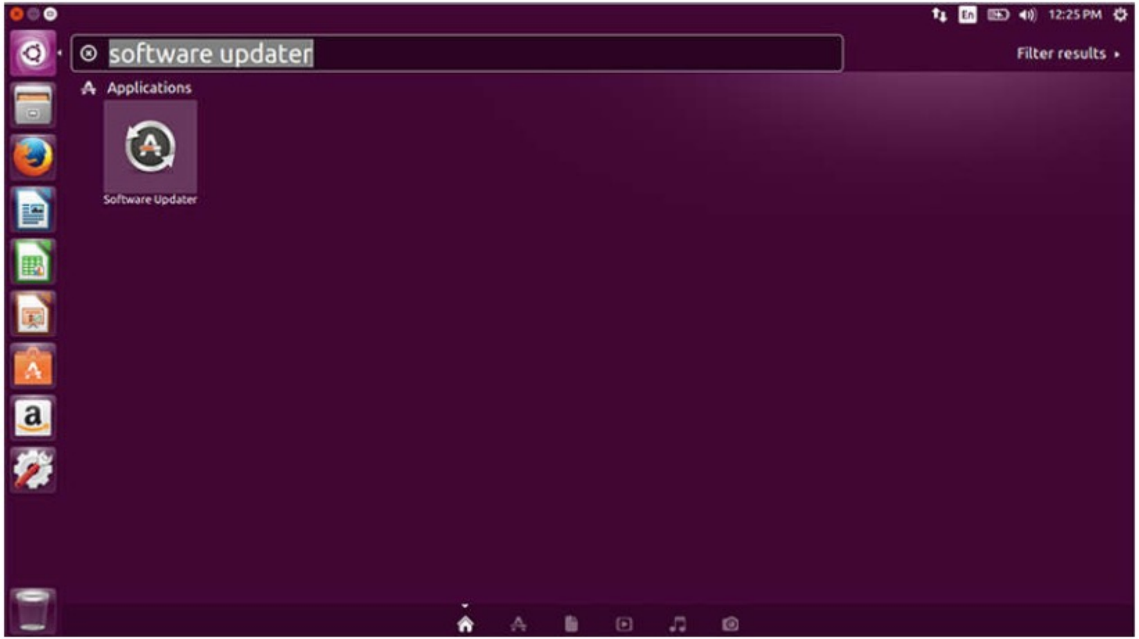
Keeping the System Up to Date

- The following 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 (Contd.)

- 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.



software updater

Working on a Linux Host

Processes and Forks

- A process is a running instance of a computer program. Multitasking operating systems can execute many processes at the same time.

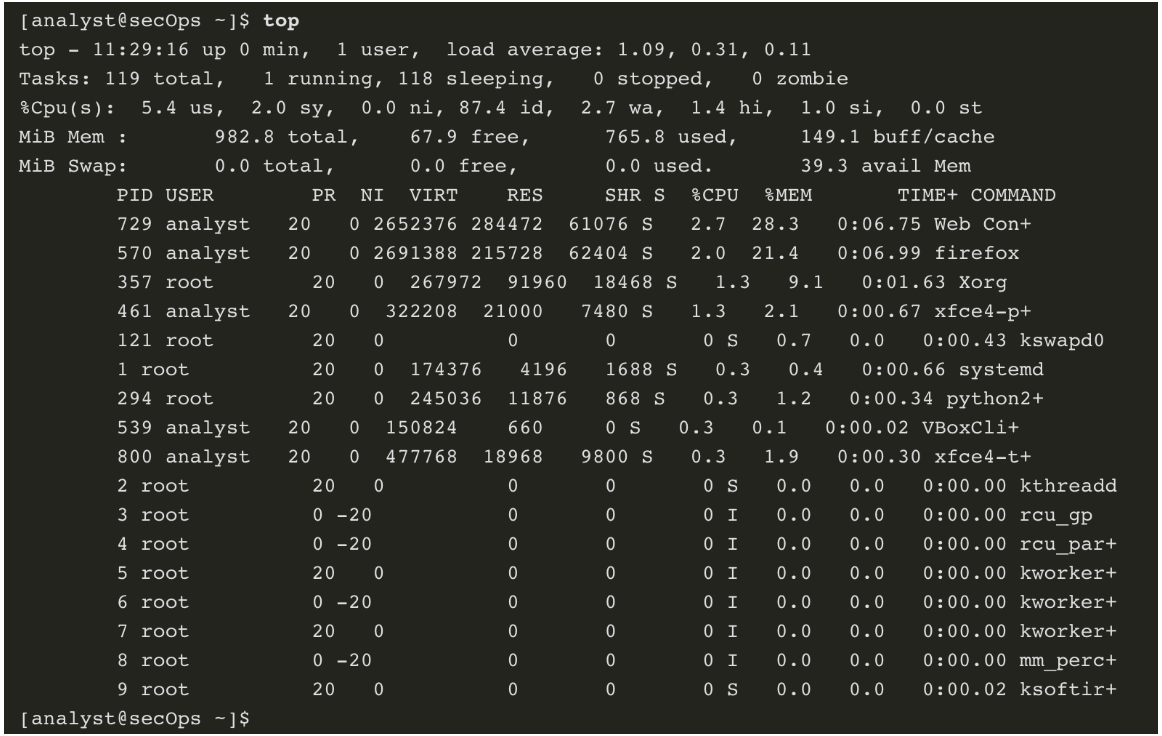
- Forking is a method that the kernel uses to allow a process to create a copy of itself to provide process scalability.

- 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

Working on a Linux Host

Processes and Forks (Contd.)

The command output shows the output of the **top** command on a Linux computer.

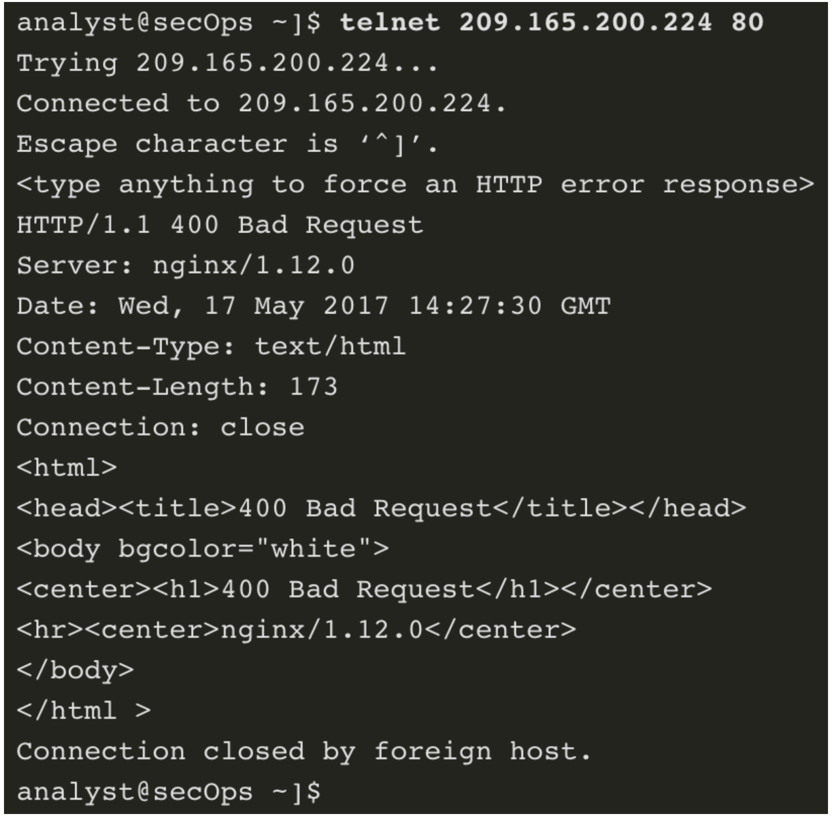


```
[analyst@secOps ~]$ top
top - 11:29:16 up 0 min,  1 user,  load average: 1.09, 0.31, 0.11
Tasks: 119 total,   1 running, 118 sleeping,   0 stopped,   0 zombie
%Cpu(s):  5.4 us,  2.0 sy,  0.0 ni, 87.4 id,  2.7 wa,  1.4 hi,  1.0 si,  0.0 st
MiB Mem :    982.8 total,     67.9 free,    765.8 used,    149.1 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.     39.3 avail Mem
        PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
        729 analyst   20   0 2652376 284472  61076 S   2.7  28.3   0:06.75 Web Con+
        570 analyst   20   0 2691388 215728  62404 S   2.0  21.4   0:06.99 firefox
        357 root      20   0  267972  91960  18468 S   1.3   9.1   0:01.63 Xorg
        461 analyst   20   0  322208  21000   7480 S   1.3   2.1   0:00.67 xfce4-p+
        121 root      20   0       0      0      0 S   0.7   0.0   0:00.43 kswapd0
        1 root        20   0  174376   4196   1688 S   0.3   0.4   0:00.66 systemd
        294 root      20   0  245036  11876    868 S   0.3   1.2   0:00.34 python2+
        539 analyst   20   0  150824    660      0 S   0.3   0.1   0:00.02 VBoxCli+
        800 analyst   20   0  477768  18968   9800 S   0.3   1.9   0:00.30 xfce4-t+
        2 root        20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
        3 root         0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
        4 root         0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
        5 root        20   0       0      0      0 I   0.0   0.0   0:00.00 kworker+
        6 root         0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
        7 root        20   0       0      0      0 I   0.0   0.0   0:00.00 kworker+
        8 root         0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
        9 root        20   0       0      0      0 S   0.0   0.0   0:00.02 ksoftir+
[analyst@secOps ~]$
```

Working on a Linux Host

Malware on a Linux Host

- Linux malware includes viruses, Trojan horses, worms, and other types of malware that can affect the operating system.
- A common Linux attack vector is its services and processes.
- The command output shows an attacker using the Telnet command to probe the nature and version of a web server (port 80).
- The attacker has learned that the server is running nginx version 1.12.0. The next step would be to research known vulnerabilities in the nginx 1.12.0 code.

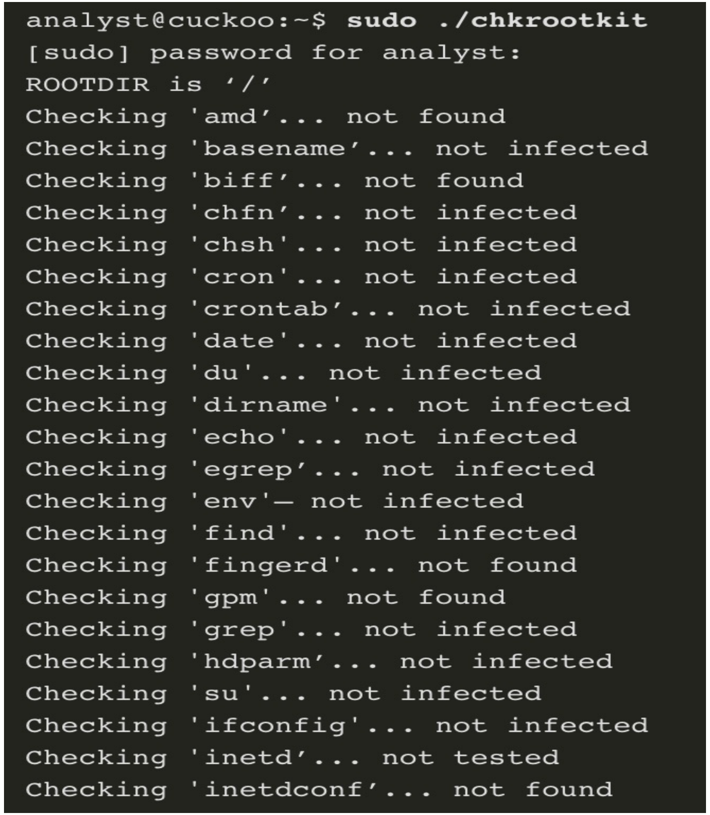


```
analyst@secOps ~]$ telnet 209.165.200.224 80
Trying 209.165.200.224...
Connected to 209.165.200.224.
Escape character is ‘^]’.
<type anything to force an HTTP error response>
HTTP/1.1 400 Bad Request
Server: nginx/1.12.0
Date: Wed, 17 May 2017 14:27:30 GMT
Content-Type: text/html
Content-Length: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

Working on a Linux Host

Rootkit Check

- A rootkit is a type of malware designed to increase an unauthorized user’s privileges or grant access to portions of the software that should not normally be allowed.
- A rootkit is destructive as it changes kernel code and its modules, changing the most fundamental operations of the OS itself.
- Rootkit detection methods include booting the computer from a trusted media.
- Rootkit removal can be complicated. Re-installation of the operating system is the only real solution to the problem.
- **chkrootkit** is a popular Linux-based program designed to check the computer for known rootkits.
- The command output shows the output of **chkrootkit** on an Ubuntu Linux.

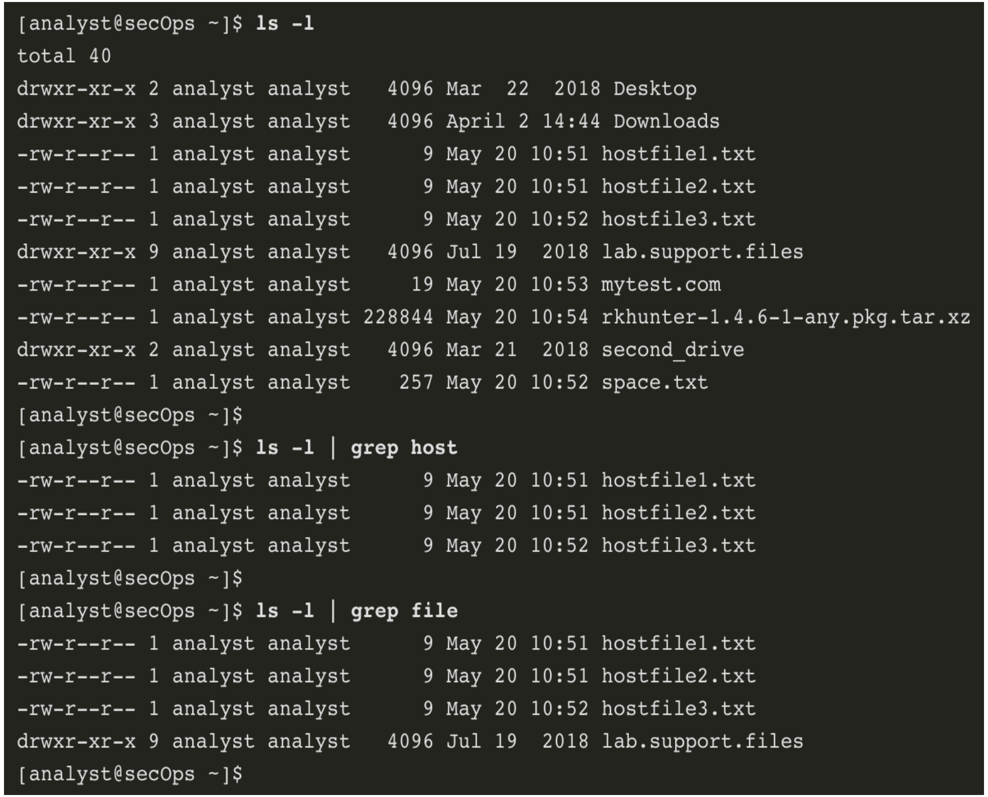


```
analyst@cuckoo:~$ sudo ./chkrootkit
[sudo] password for analyst:
ROOTDIR is ‘/’
Checking 'amd’... not found
Checking 'basename’... not infected
Checking 'biff’... not found
Checking 'chfn’... not infected
Checking 'chsh'... not infected
Checking 'cron'... not infected
Checking 'crontab’... not infected
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
```

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.
- 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
-rw-r--r-- 1 analyst analyst 228844 May 20 10:54 rkhunter-1.4.6-1-any.pkg.tar.xz
drwxr-xr-x 2 analyst analyst   4096 Mar 21  2018 second_drive
-rw-r--r-- 1 analyst analyst    257 May 20 10:52 space.txt
[analyst@secOps ~]$
[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 ~]$
[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
[analyst@secOps ~]$
```

Linux Operating System

New Terms and Commands

- ps, top, kill
- ls, grep
- Forking

- piping
- rookit
- X Window System

Lab 7 – 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.

Linux Operating System

Lab 8 – Linux Servers

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