

Information Security Operations Center (Info SOC)

Lecture 2

The Features and Characteristics of the Linux Operating System

Prepared by
Dr. Ibrahim Gomaa

Contents

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	Explain how client-server networks function.
Basic Server Administration	Explain how a Linux administrator locates and manipulates security log files.
The Linux File System	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.

The features and characteristics of the Linux Operating System

What is Linux?

- Linux is an **operating system** that was created in **1991**.
- Linux is **open source**, **fast**, **reliable**, and **small**. It requires **very little hardware resources** to run and is highly customizable.
- Linux is part of several platforms and can be found on devices anywhere from **wristwatches** to **supercomputers**.
- Linux is designed to be connected to the network, which makes it much simpler to write and use network-based applications.
- A **Linux distribution** is the term used to describe packages created by different organizations and include the Linux kernel with customized tools and software packages.



The features and characteristics of the Linux Operating System

The Value of Linux

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

- **Linux is open source** - Any person can acquire Linux at no charge and modify it to fit specific needs.
- **The Linux CLI is very powerful** - The Linux Command Line Interface (CLI) is extremely powerful and enables analysts to perform tasks not only directly on a terminal, but also remotely.
- **The user has more control over the OS** - The administrator user in Linux, known as the **root user**, or superuser, can modify any aspect of the computer with a few keystrokes.
- **It allows for better network communication control** - Control is an inherent part of Linux.

The features and characteristics of the Linux Operating System

Linux in the SOC

- 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.
- Sguil 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.

The screenshot displays the Sguil-0.9.0 interface, which is a cybersecurity analyst console. The top section shows a list of real-time events with columns for ST, CNT, Sensor, Alert ID, Date/Time, Src IP, SPort, Dst IP, DPort, Pr, and Event Message. The events are filtered by 'RealTime Events' and 'Escalated Events'. The bottom section shows a detailed packet analysis view for a specific event, including a table of IP Resolution, Agent Status, Snort Statistics, and System Msgs. The packet analysis view includes a table of IP Resolution, Agent Status, Snort Statistics, and System Msgs, and a detailed packet analysis view for a specific event, including a table of IP Resolution, Agent Status, Snort Statistics, and System Msgs.

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	7	seconion...	5.1583	2020-05-10 21:29:13	209.165.201.17	52458	209.165.200.235	80	6	ET INFO Dotted Quad Host HTA ...
RT	7	seconion...	5.1584	2020-05-10 21:29:13	209.165.201.17	52458	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 OneLoader ...
RT	1	seconion...	5.1600	2020-05-10 21:29:13	209.165.201.17	52468	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	52458	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	52458	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 OneLoader ...
RT	1	seconion...	7.1913	2020-05-10 21:29:13	209.165.201.17	52468	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	52836	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	52836	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	52896	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	52896	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	38782	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-en...	seconion-en...	pcap	2020-05-12
3	seconion-en...	seconion-en...	snort	2020-05-10
4	seconion-en...	seconion-en...	pcap	2020-05-12
5	seconion-en...	seconion-en...	snort	2020-05-10
6	seconion-en...	seconion-en...	pcap	2020-05-12
7	seconion-en...	seconion-en...	snort	2020-05-10

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

TCP	Source Port	Dest Port	RRR	CSS	YI	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum	
TCP	52460	80	...	X	X	...	2237277941	1593194311	8	0	501	0	30678

DATA	47	54	20	2F	31	31	20	48	54	50	2F	31	2E	31	...	
DATA	00	0A	48	6F	73	74	3A	20	32	30	39	2E	31	36	35	2E
DATA	32	30	30	2E	32	33	35	00	0A	55	73	65	72	20	41	67
DATA	65	6E	74	3A	20	40	6F	7A	69	6C	6C	61	2F	34	2E	30
DATA	20	28	63	6F	60	70	61	74	69	62	6C	65	38	20	40	53

Linux in the SOC (Contd.)

The following table lists a few tools that are often found in a SOC:

SOC Tool	Description
Network packet capture software	• A crucial tool for a SOC analyst as it makes it possible to observe and understand every detail of a network transaction. • Wireshark is a popular packet capture tool.
Malware analysis tools	• These tools allow analysts to safely run and observe malware execution without the risk of compromising the underlying system.
Intrusion detection systems (IDSs)	• These tools 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.

The features and characteristics of the Linux Operating System

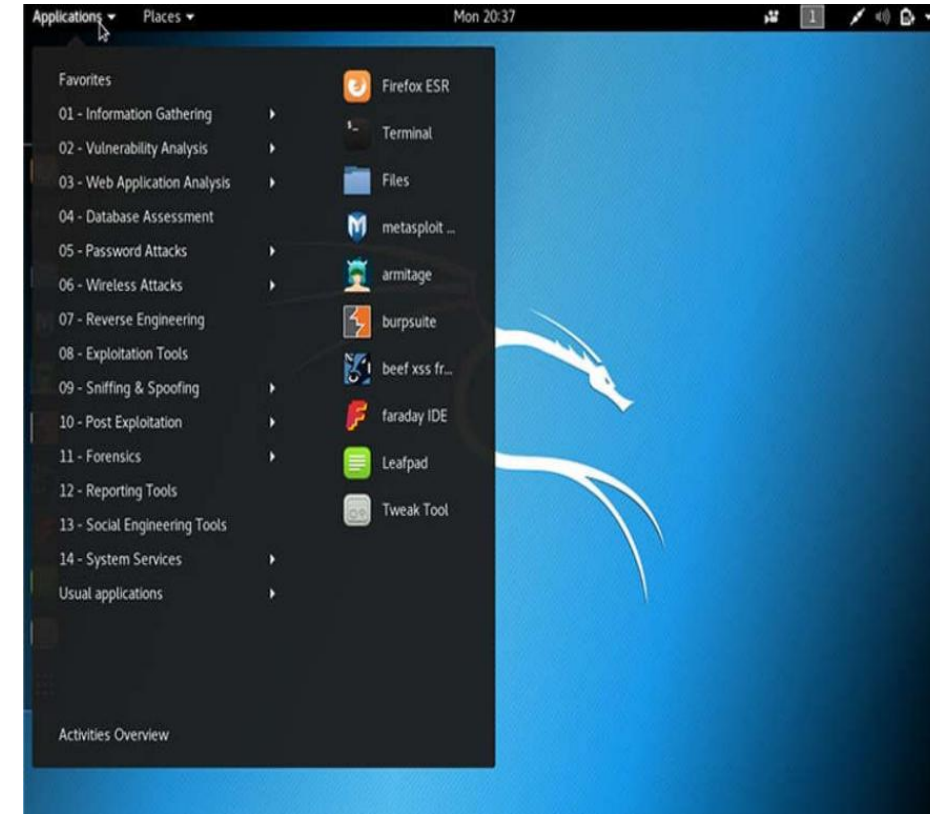
Linux in the SOC (Contd.)

SOC Tool	Description
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	• Log files are used to record events. • 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. Security alerts are often assigned to analysts through a ticketing system.

The features and characteristics of the Linux Operating System

Linux Tools

- Linux computers that are used in the SOC often contain **penetration testing tools**.
- A penetration test, also known as **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 which contains many penetration tools together in a single Linux distribution.
- Notice all the major categories of penetration testing tools of Kali Linux.



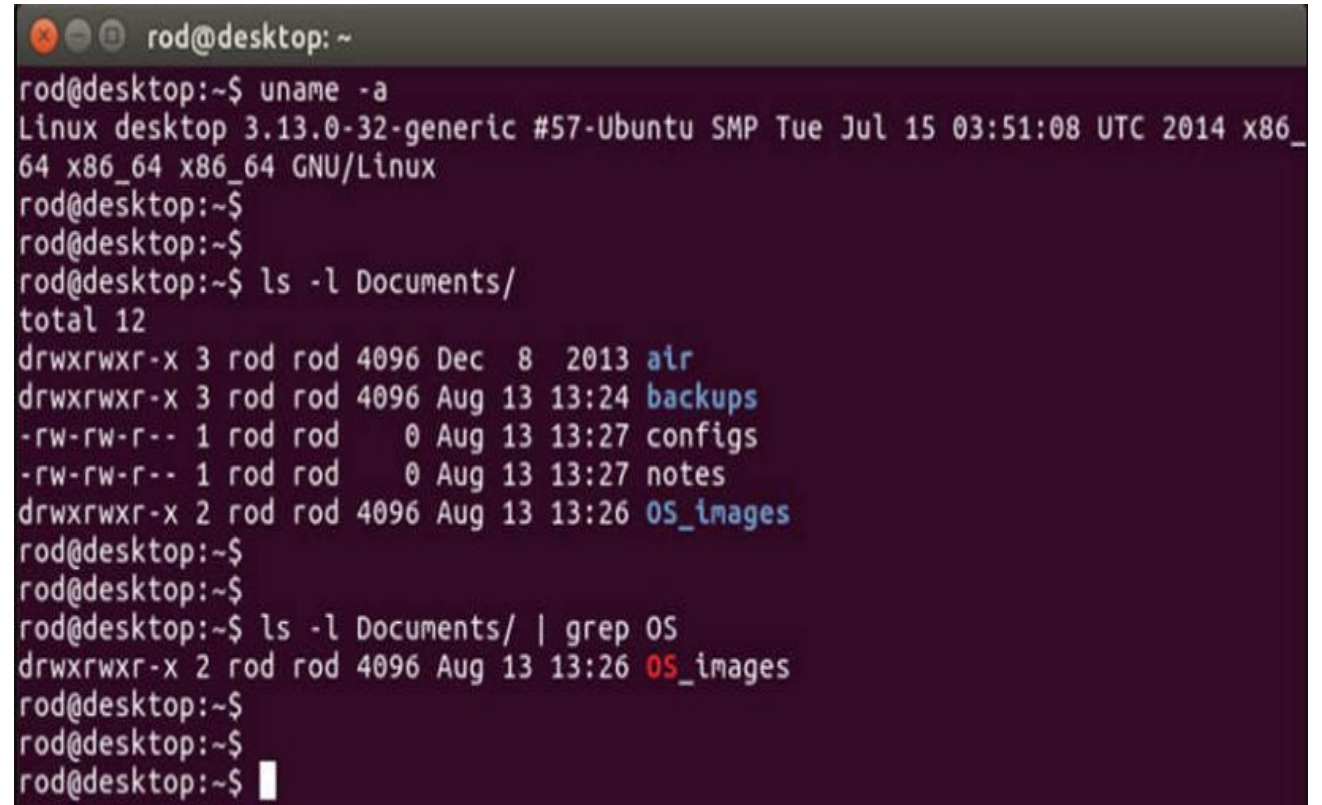
The Linux Shell

- In Linux, the user communicates with the OS by using the **CLI** or the **GUI**.
- Linux often starts in the GUI by 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 and are named as some variation of the word terminal.
- In Linux, popular terminal emulators are **Terminator**, **eterm**, **xterm**, **konsole**, and **gnome-terminal**.
- **Fabrice Bellard** has created **JSLinux** which allows an emulated version of Linux to run in a browser.

Note: *The terms shell, console, console window, CLI terminal, and terminal window are often used interchangeably.*

The Linux Shell (Contd.)

The figure shows **gnome-terminal**, a popular Linux terminal emulator.



```
rod@desktop: ~  
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 created to perform a specific task.
- As the commands are programs stored on the disk, when a user types a command, the shell must find it on the disk before it can be executed.
- The following 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 that are currently running in the system.
su	Simulates a login as another user or to become a superuser.

Basic Commands (Contd.)

Command	Description
sudo	Runs a command as a super user, by default, or another named user.
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.
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	Shuts down the system and performs shut down related tasks including restart, halt, put to sleep or kick out all currently connected users.
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 following table lists a few of the most common commands related to files and directories:

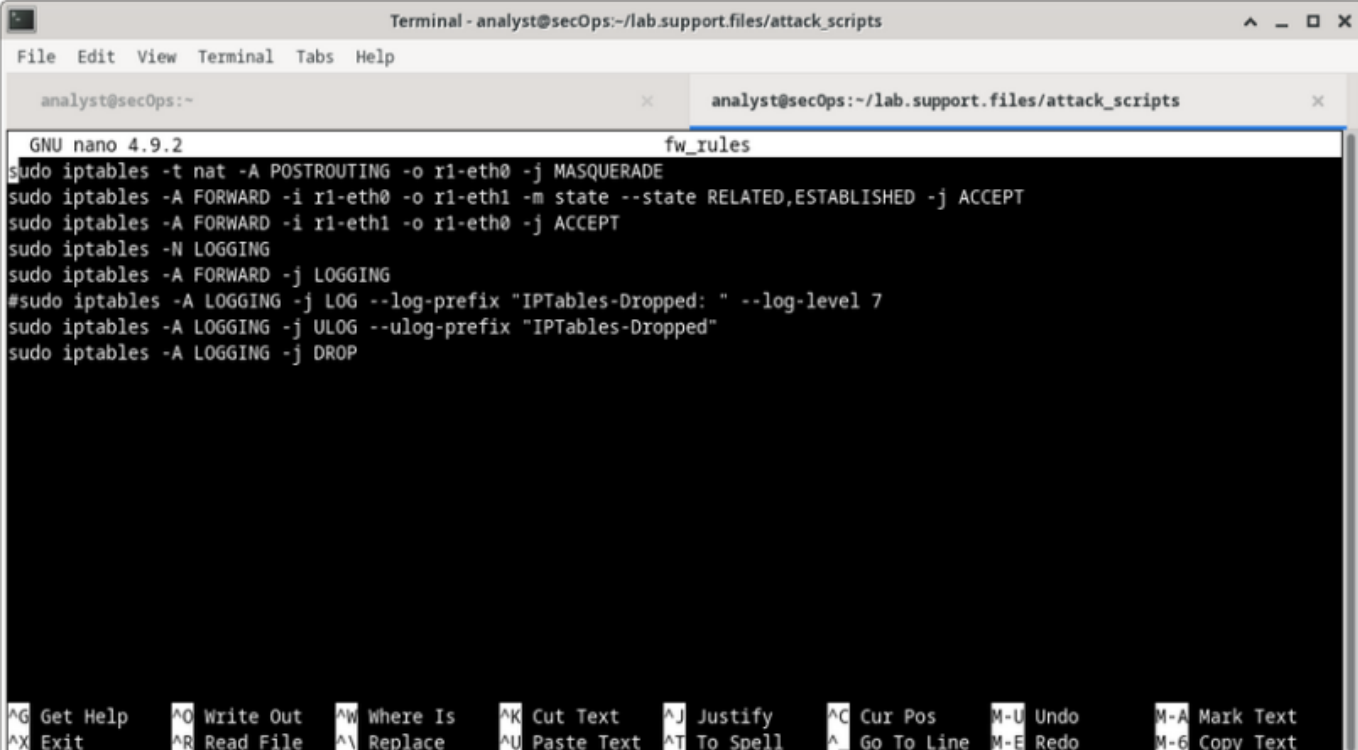
Command	Description
ls	Displays the files inside a directory.
cd	Changes the current directory.
mkdir	Creates a directory under the current directory.
cp	Copies files from source to destination.
mv	Moves files to a different directory.
rm	Removes files.
grep	Searches for specific strings of characters 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, 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.

Working with text files (Contd.)

- The figure shows **nano**, a popular command-line text editor.
- The administrator is editing firewall rules. Text editors are often used for system configuration and maintenance in Linux.
- Due to the lack of graphical support, nano (or **GNU nano**) can only be controlled with the keyboard.



```
Terminal - analyst@secOps:~/lab.support.files/attack_scripts
File Edit View Terminal Tabs Help
analyst@secOps:~
GNU nano 4.9.2 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

^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-C Copy Text
```

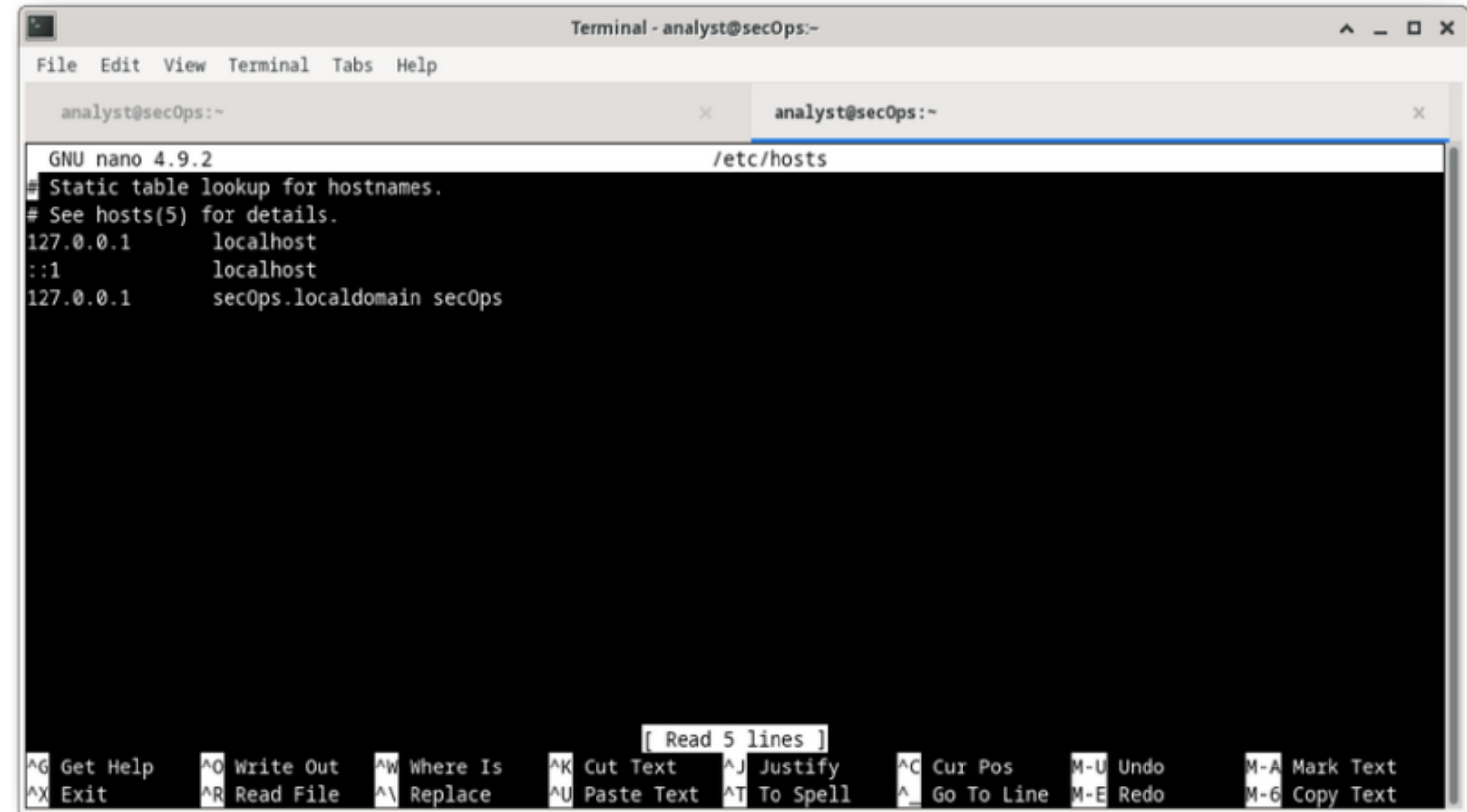
The importance of text files in Linux

- In Linux, **everything is treated as a file**. This includes the memory, the disks, the monitor, and the directories.
- **Configuration files** are text files which are used to store adjustments and settings for specific applications or services.
- 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. Users are able to specify exactly how they want any given application or service to behave. When launched, services and applications check the contents of specific configuration files to adjust their behavior accordingly.

Note: The administrator used the command `sudo nano /etc/hosts` to open the file. The command **sudo** (short for “**superuser do**”) invokes the superuser privilege to use the nano text editor to open the host file.

The importance of text files in Linux (Contd.)

- In the figure, the administrator opened the host configuration file in **nano** for editing.
- The host file contains static mappings of host IP addresses to names.
- The names serve as shortcuts that allow connecting to other devices by using a name instead of an IP address. **Only the superuser** can change the host file.

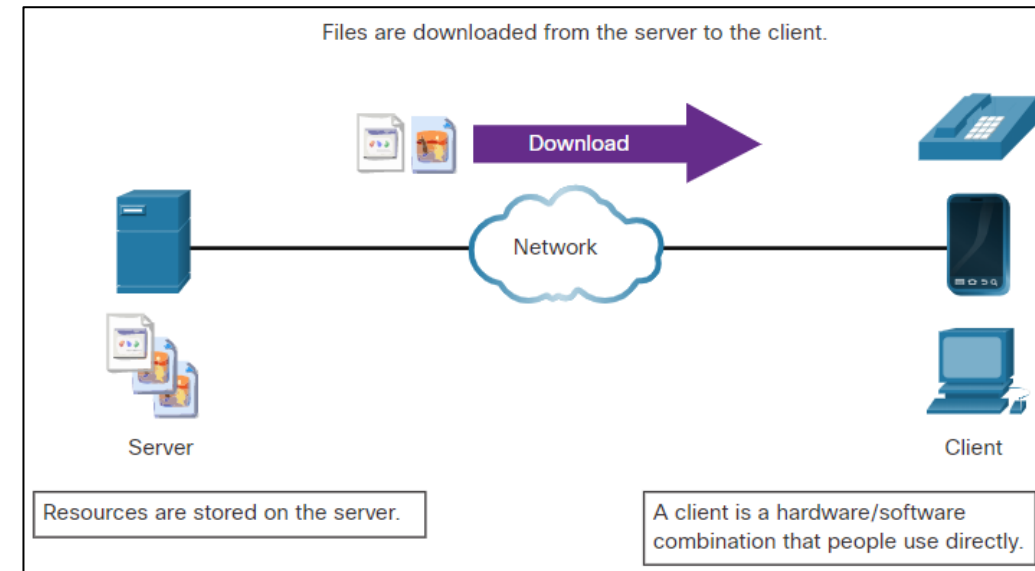


```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
analyst@secOps:~
GNU nano 4.9.2 /etc/hosts
# Static table lookup for hostnames.
# See hosts(5) for details.
127.0.0.1    localhost
::1         localhost
127.0.0.1    secOps.localdomain secOps

[ 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 ^N Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo    M-6 Copy Text
```

An Introduction to Client-Server Communications

- **Servers** are computers with software installed that enables them to provide services to clients across the network.
- Some provide external resources such as files, email messages, or web pages to clients upon request.
- Other services run maintenance tasks such as log management, disk scanning and so on.
- Each service requires separate server software.
- The server in the figure uses file server software to provide clients with the ability to retrieve and submit files.



Servers, Services, and Their Ports

- A **port** is a reserved network resource used by a service.
- While the administrator can decide which port to use with any given service, many clients are configured to use a specific port by default.
- The following table lists a few commonly used ports and their services. These are also called as **well-known ports**.

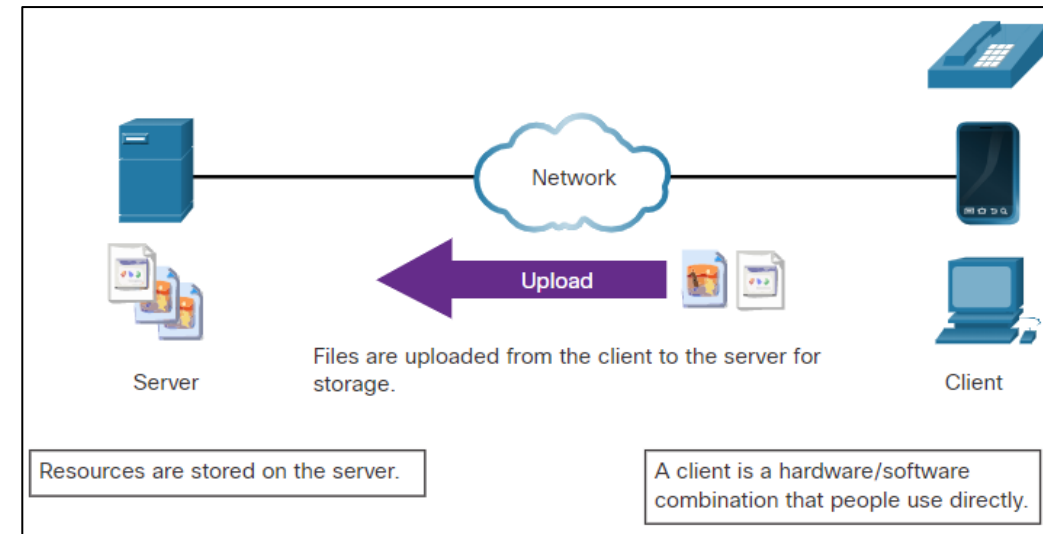
Port	Description
20/21	File Transfer Protocol (FTP)
22	Secure Shell (SSH)
23	Telnet remote login service
25	Simple Mail Transfer Protocol (SMTP)
53	Domain Name System (DNS)
67/68	Dynamic Host Configuration Protocol (DHCP)

Servers, Services, and Their Ports (Contd.)

Port	Description
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 server.
- Clients use a **well-defined protocol** to communicate with the server.
- **Web browsers** are web clients that are used to communicate with web servers through the Hyper Text Transfer Protocol on port 80.
- The File Transfer Protocol client is software used to communicate with an FTP server.
- The figure shows a client uploading files to a server.



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 settings in the files.
- 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;
```

Service Configuration Files (Contd.)

The command output shows the configuration file for the network time protocol (**NTP**).

```
[analyst@secOps ~]$ cat /etc/ntp.conf
# 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
[analyst@secOps ~]$
```

Service Configuration Files (Contd.)

- The command output shows the configuration file for Snort, a Linux-based intrusion detection system (IDS).
- There is no rule for a configuration file format. It is the choice of the service's developer. However, the **option = value** format is often used.

```
[analyst@secOps ~]$ cat /etc/snort/snort.conf
#-----
#   VRT Rule Packages Snort.conf
#
#   For more information visit us at:
#       http://www.snort.org           Snort Website
#       http://vrt-blog.snort.org/     Sourcefire VRT Blog
#
#   Mailing list Contact:  snort-sigs@lists.sourceforge.net
#   False Positive reports: fp@sourcefire.com
#   Snort bugs:           bugs@snort.org
#
#   Compatible with Snort Versions:
#   VERSIONS : 2.9.9.0
#
#   Snort build options:
#   OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --
enable-zlib --enable-active-response --enable-normalizer --enable-reload --enable-react --enable-
flexresp3
<output omitted>
#####
# Step #1: Set the network variables.  For more information, see README.variables
#####
# Setup the network addresses you are protecting
##ipvar HOME_NET any
##ipvar HOME_NET [192.168.0.0/24,192.168.1.0/24]
ipvar HOME_NET [209.165.200.224/27]
# Set up the external network addresses.  Leave as "any" in most situations
ipvar EXTERNAL_NET any
```

Hardening Devices

- **Device hardening** involves implementing proven methods of securing the device and protecting its administrative access.
- Some of these methods involve maintaining passwords, configuring enhanced remote login features, and implementing secure login with SSH.
- Depending on the **Linux distribution**, many services are enabled by default. Stopping such services and ensuring they do not automatically start at boot time is another device hardening technique.
- OS updates are extremely important to maintaining a hardened device. OS developers create and issue fixes and patches regularly.

Hardening Devices (Contd.)

The following are **basic best practices** for device hardening:

- 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

Linux Roles and File Permissions

- Linux uses file permissions to organize the system and enforce boundaries within the computer.
- Every file in Linux carries its **file permissions**, which define the actions that the **owner**, the **group**, and **others** can perform with the file.
- The possible permission rights are **Read**, **Write**, and **Execute**.
- The **ls** command with the **-l** parameter lists additional information about the file.

Linux Roles and File Permissions (Contd.)

The output of the **ls -l** command provides a lot of information about the file **space.txt**:

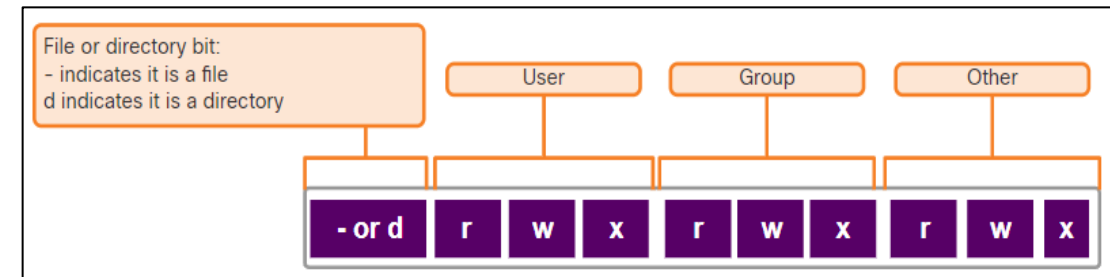
- The first field displays the permissions with **space.txt** (**-rwxrw-r--**).
- The second field defines the number of **hard links** to the file (number **1** after the permissions).
- The third and fourth field display the **user** (**analyst**) and **group** (**staff**) who own the file, respectively.
- The fifth field displays the **file size in bytes**. The **space.txt** file has 253 bytes.
- The sixth field displays the **date and time of the last modification**.
- The seventh field displays **the file name**.

```
[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)
[analyst@secOps ~]$
```

Linux Roles and File Permissions (Contd.)

The figure here shows a breakdown of file permissions in Linux. The file **space.txt** has the following permissions:

- The **dash** (-) means that this is a file.
- The first set of characters (**rw****x**) is for **user permission**. The user (**analyst**) who owns the file can **Read**, **Write** and **eXecute** the file.
- The second set of characters is for **group permissions** (**rw**-). The group (**staff**) who owns the file can **Read** and **Write** to the file.
- The third set of characters is for any **other user or group permissions** (**r--**) who can only **Read** the file.



The Linux File System

Linux Roles and File Permissions (Contd.)

- Octal values are used to define permissions.
- File permissions are a fundamental part of Linux and cannot be broken.
- The **only user** that can override file permission on a Linux computer is the **root user**.

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

X Window System

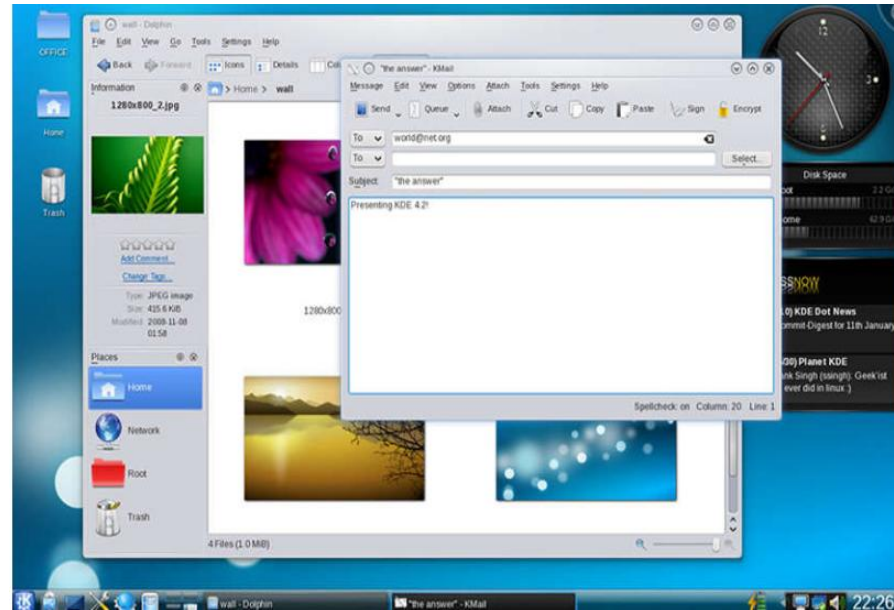
- The graphical interface present in most Linux computers is based on the **X Window System**.
- **X Window**, also known as **X** or **X11**, is a windowing system designed to provide the basic framework for a GUI.
- X includes functions for drawing and moving windows on the display device and interacting 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.
- X does not specify the user interface, leaving it to other programs, such as window managers, to define all the graphical components.

X Window System (Contd.)

Examples of window managers are **Gnome** and **KDE**.



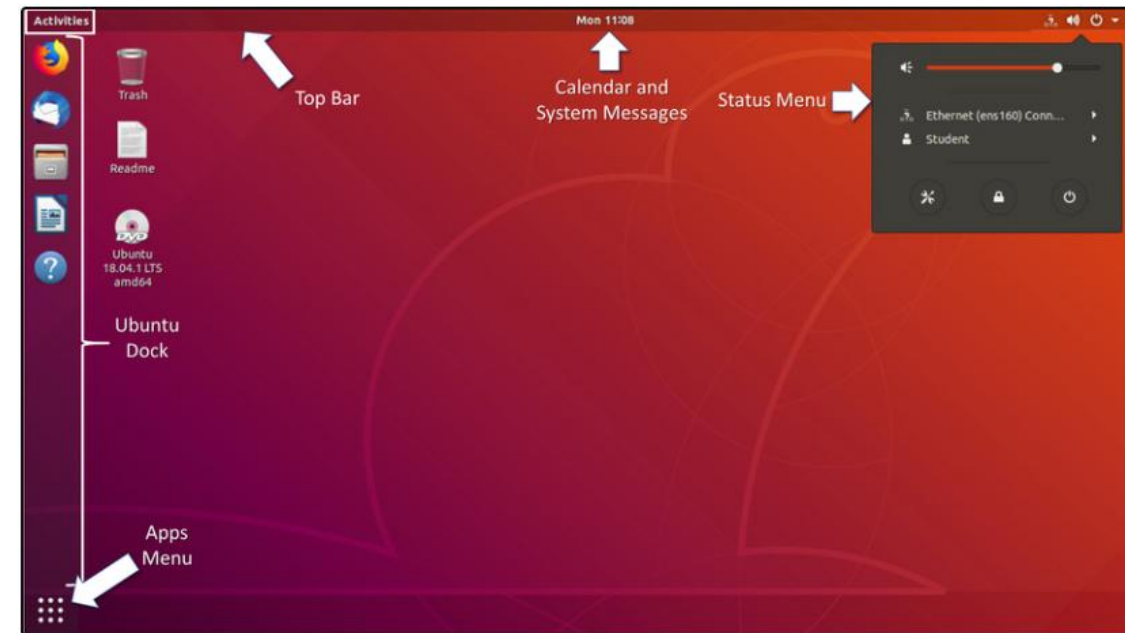
The Gnome Window Manager



The KDE Window Manager

The Linux GUI

- While an operating system does not require a GUI to function, GUIs are considered more user-friendly than the CLI. The Linux GUI can be easily replaced by the user.
- **Ubuntu** is a very popular and user-friendly Linux distribution.
- Ubuntu Linux uses **Gnome 3** as its default GUI.
- The figure shows the location of some of the features of the Ubuntu Gnome 3 Desktop.



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 includes programs called **package managers**.
- By using a **package manager** to install a package, all the necessary files are placed in the correct file system location.
- A package is the term used to refer to a program and all its supporting files.
- The command output shows the output of a few **apt-get** commands used in Debian distributions.

```
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]
<output omitted>
Fetched 4,613 kB in 4s (1,003 kB/s)
Reading package lists... Done
analyst@cuckoo:~$
analyst@cuckoo:~$ sudo apt-get upgrade
Reading package lists Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
linux-generic-hwe-16.04 linux-headers-generic-hwe-16.04
linux-image-generic-hwe-16.04
The following packages will be upgraded:
firefox firefox-locale-en gir1.2-javascriptcoregtk-4.0 gir1.2-webkit2-4.0 libjavascriptcoregtk-4.0-18
libwebkit2gtk-4.0-37 libwebkit2gtk-4.0-37-gtk2 libxen-4.6 libxenstore3.0 linux-libc-dev logrotate
openssh-client
qemu-block-extra qerau-kvm qemu-system-common qemu-system-x86 qemu-utils
```

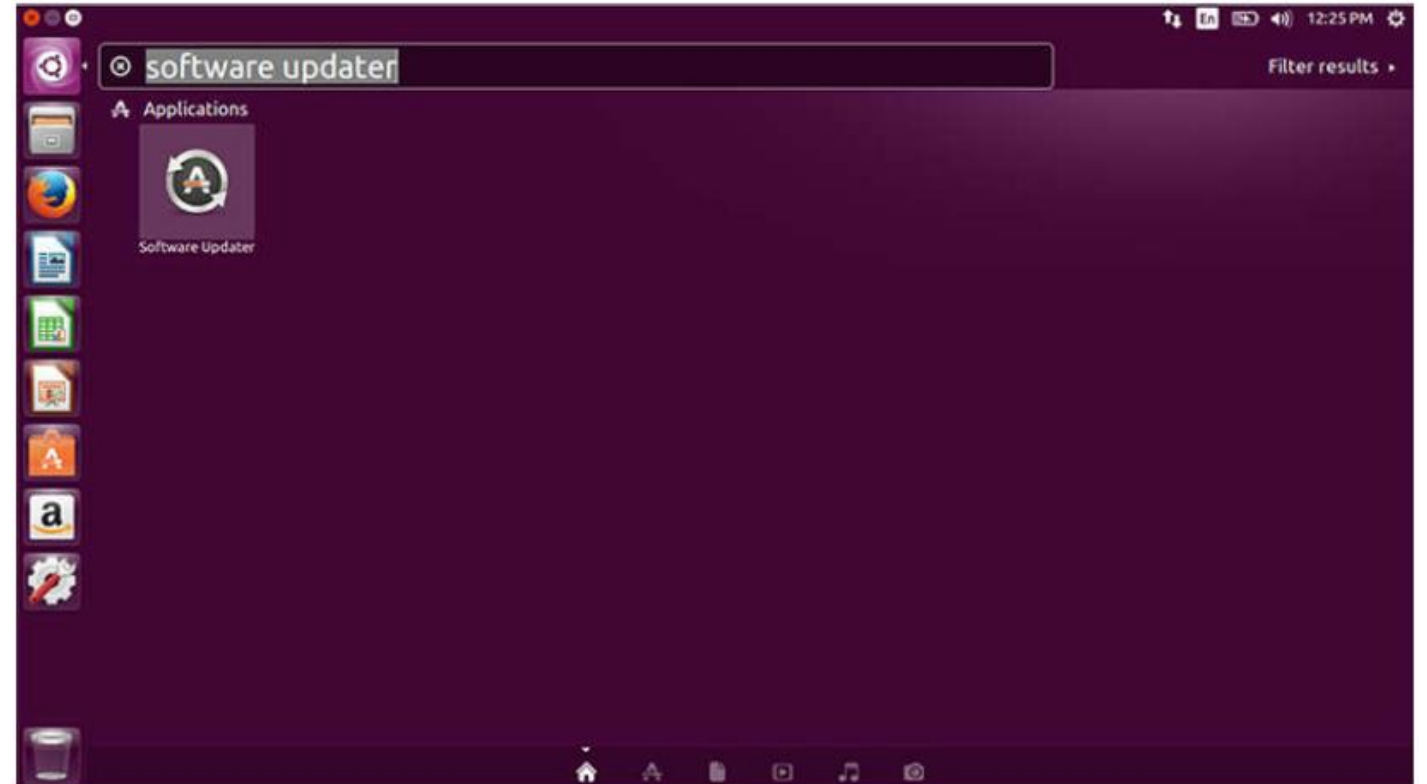
Keeping the System Up to Date

- **OS updates**, also known as **patches**, are released periodically by OS companies to address any known vulnerabilities in their operating systems.
- 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 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.



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 ~]$
```

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
```

Thanks