Cyber Security Internship Report

Cyber Security Internship – Task 2

Operating System Hardening and Security Best Practices

Submitted by: Hetal Antala

Date : 16 / 01 / 2026

Section 1: Introduction

What is Operating System Security

Operating System (OS) security refers to the measures and controls implemented to protect an operating system from unauthorized access, misuse, data theft, and attacks. The operating system acts as an interface between hardware and users, so compromising it can give attackers full control over the system. OS security ensures confidentiality, integrity, and availability of system resources and data.

Why OS Hardening is Important

OS hardening is the process of securing an operating system by reducing its vulnerabilities and attack surface. A default OS installation contains many unnecessary services and settings that can be exploited by attackers. Hardening helps prevent malware infections, unauthorized access, privilege escalation, and data breaches.

Section 2: OS Setup

OS Used

For this task, **Ubuntu Linux** was used as the operating system.

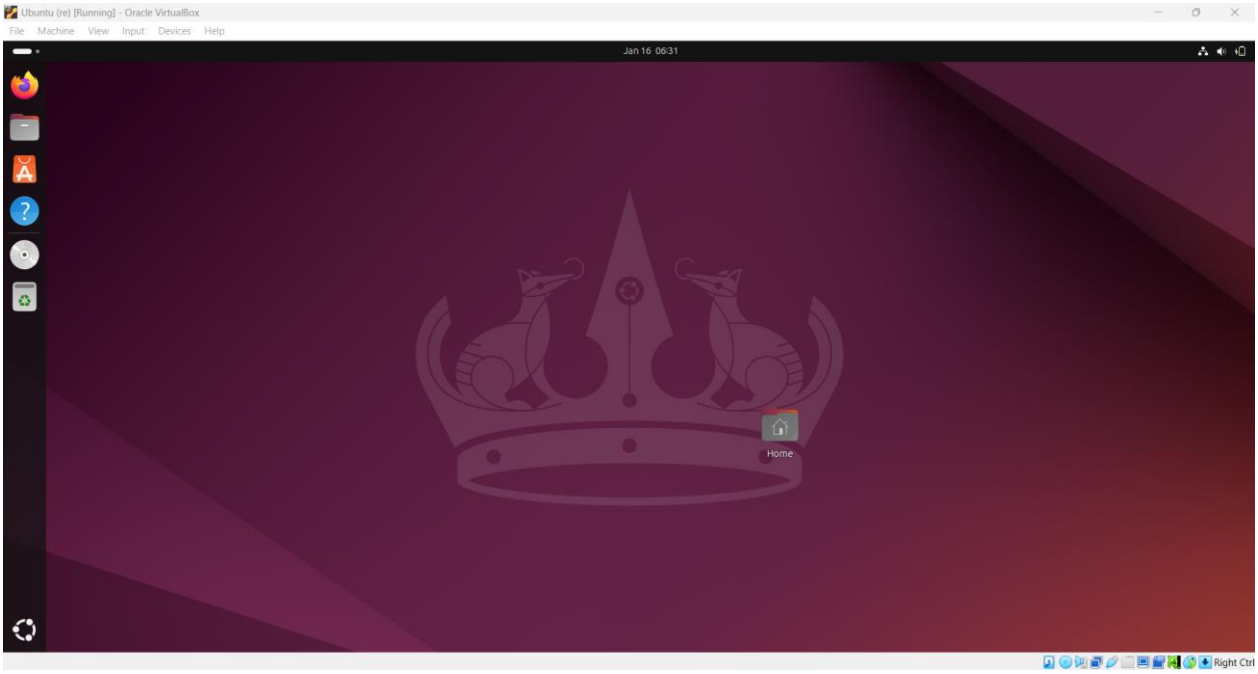
Installation Type

Ubuntu was installed inside a **Virtual Machine (VM)** using **VirtualBox**, which provides a safe environment to test security configurations without affecting the host system.

Tools Used

- **VirtualBox** – For creating and managing the Linux virtual machine
- **Ubuntu Linux** – Secure open-source operating system
- **Terminal** – To execute system and security-related commands

- Successfully Installed



- **Section 3: User Accounts & Access Control**

Root vs Normal User

- **Root user** has complete control over the system, including system files and configurations.

```
vboxuser@Ubuntu:~$ sudo whoami
[sudo] password for vboxuser:
root
vboxuser@Ubuntu:~$
```

- **Normal users** have limited permissions and cannot modify critical system settings.

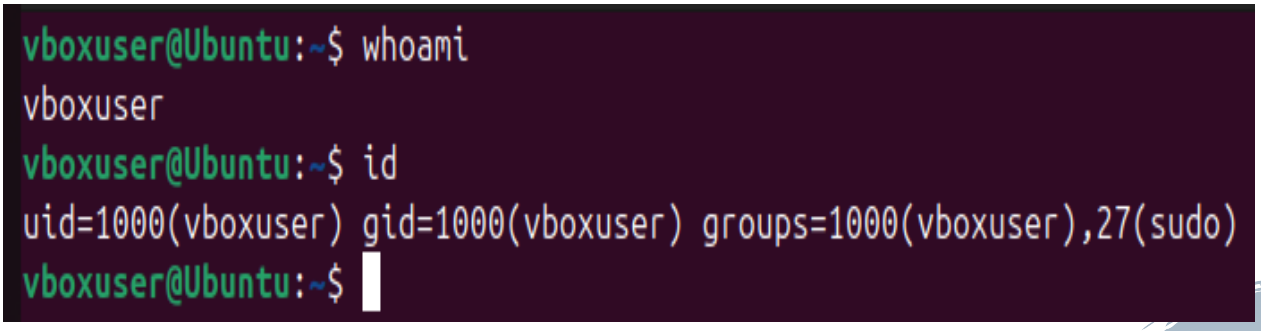
Using a normal user reduces the risk of accidental or malicious system damage.

sudo Usage

The sudo command allows a normal user to temporarily execute commands with administrative privileges. This prevents constant root usage and increases accountability by logging privileged actions.

Least Privilege Principle

The principle of least privilege states that users and programs should be given only the minimum permissions required to perform their tasks. This limits the damage if an account is compromised.



Section 4: File Permissions (Linux)

Explanation of r, w, x

- **r (read)** – Permission to read file contents
- **w (write)** – Permission to modify the file
- **x (execute)** – Permission to execute the file

Owner, Group, and Others

Linux assigns permissions to:

- **Owner** – File creator
- **Group** – Users in the same group
- **Others** – All other users

Commands Used

ls -l

Displays file permissions and ownership.

```
vboxuser@Ubuntu:~$ ls -l
total 36
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Desktop
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Documents
drwxr-xr-x 2 vboxuser vboxuser 4096 Jan 14 15:45 Downloads
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Music
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Pictures
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Public
drwx------ 5 vboxuser vboxuser 4096 Nov 19 10:24 snap
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Templates
-rw-rw-r-- 1 vboxuser vboxuser    0 Dec  6 18:33 test.txt
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Videos
```

chmod 700 test.txt

Changes file permissions.

```
vboxuser@Ubuntu:~$ chmod 700 test.txt
vboxuser@Ubuntu:~$ ls -l
total 36
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Desktop
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Documents
drwxr-xr-x 2 vboxuser vboxuser 4096 Jan 14 15:45 Downloads
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Music
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Pictures
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Public
drwx------ 5 vboxuser vboxuser 4096 Nov 19 10:24 snap
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Templates
-rwx------ 1 vboxuser vboxuser    0 Dec  6 18:33 test.txt
drwxr-xr-x 2 vboxuser vboxuser 4096 Nov  8 06:11 Videos
```

chown user:user file.txt

Changes file ownership.

◆ Section 5: Administrator vs Standard User Differences

- **Administrator (root)** can install software, modify system files, and manage users.

- **Standard user** has restricted access to prevent system-level changes.

Security Risks of Admin Misuse

Using an administrator account for daily tasks increases the risk of malware infections and accidental system changes. Attackers gaining admin access can fully compromise the system.

◆ Section 6: Firewall Configuration

Firewall Used

Ubuntu uses **UFW (Uncomplicated Firewall)** to manage network traffic.

Commands Used

sudo ufw enable

sudo ufw status

```
vboxuser@Ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
vboxuser@Ubuntu:~$ sudo ufw status
Status: active
vboxuser@Ubuntu:~$
```

Why Firewall is Important

A firewall blocks unauthorized network access, allowing only trusted traffic. It protects the system from network-based attacks such as port scanning and unauthorized connections.

- **Section 7: Running Processes & Services**

Commands Used

ps aux

top

```
vboxuser@Ubuntu:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.3  0.4  23488 14784 ?        Ss   06:30   0:02 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    06:30   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    06:30   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   06:30   0:00 [kworker/R-rcu_gp]
root           5  0.0  0.0      0     0 ?        I<   06:30   0:00 [kworker/R-sync_wq]
root           6  0.0  0.0      0     0 ?        I<   06:30   0:00 [kworker/R-kvfree_rcu_reclaim]
root           7  0.0  0.0      0     0 ?        I<   06:30   0:00 [kworker/R-slub_flushwq]
root           8  0.0  0.0      0     0 ?        I<   06:30   0:00 [kworker/R-netns]
root           9  0.2  0.0      0     0 ?        I    06:30   0:01 [kworker/0:0-events]
root          11  0.0  0.0      0     0 ?        I<   06:30   0:00 [kworker/0:0H-events_highpri]
root          12  0.0  0.0      0     0 ?        I    06:30   0:00 [kworker/u8:0-ipv6_addrconf]
root          13  0.0  0.0      0     0 ?        I<   06:30   0:00 [kworker/R-mm_percpu_wq]
root          14  0.0  0.0      0     0 ?        I    06:30   0:00 [rcu_tasks_kthread]
root          15  0.0  0.0      0     0 ?        I    06:30   0:00 [rcu_tasks_rude_kthread]
root          16  0.0  0.0      0     0 ?        I    06:30   0:00 [rcu_tasks_trace_kthread]
root          17  0.0  0.0      0     0 ?        S    06:30   0:00 [ksoftirqd/0]
root          18  0.0  0.0      0     0 ?        I    06:30   0:00 [rcu_preempt]
root          19  0.0  0.0      0     0 ?        S    06:30   0:00 [rcu_exp_par_gp_kthread_worker/0]
root          20  0.0  0.0      0     0 ?        S    06:30   0:00 [rcu_exp_gp_kthread_worker]
root          21  0.0  0.0      0     0 ?        S    06:30   0:00 [migration/0]
root          22  0.0  0.0      0     0 ?        S    06:30   0:00 [idle_inject/0]
root          23  0.0  0.0      0     0 ?        S    06:30   0:00 [cpuhp/0]
root          24  0.0  0.0      0     0 ?        S    06:30   0:00 [cpuhp/1]
root          25  0.0  0.0      0     0 ?        S    06:30   0:00 [idle_inject/1]
root          26  0.0  0.0      0     0 ?        S    06:30   0:00 [migration/1]
root          27  0.0  0.0      0     0 ?        S    06:30   0:00 [ksoftirqd/1]
root          32  0.0  0.0      0     0 ?        S    06:30   0:00 [kdevtmpfs]
root          33  0.0  0.0      0     0 ?        I<   06:30   0:00 [kworker/R-inet_frag_wq]
root          34  0.0  0.0      0     0 ?        S    06:30   0:00 [kauditd]
root          35  0.0  0.0      0     0 ?        S    06:30   0:00 [khungtaskd]
root          37  0.0  0.0      0     0 ?        S    06:30   0:00 [oom_reaper]
root          39  0.0  0.0      0     0 ?        I<   06:30   0:00 [kworker/R-writeback]
root          40  0.0  0.0      0     0 ?        S    06:30   0:00 [kcompactd0]
root          41  0.0  0.0      0     0 ?        SN   06:30   0:00 [ksmd]
root          42  0.0  0.0      0     0 ?        SN   06:30   0:00 [khugepaged]
```

systemctl list-units --type=service

```
vboxuser@Ubuntu:~$ systemctl list-units --type=service
  UNIT                                LOAD   ACTIVE SUB     DESCRIPTION
  accounts-daemon.service             loaded active running Accounts Service
  alsa-restore.service                loaded active exited  Save/Restore Sound Card State
  apache2.service                     loaded active running The Apache HTTP Server
  apparmor.service                    loaded active exited  Load AppArmor profiles
  apport.service                      loaded active exited  automatic crash report generation
  avahi-daemon.service                loaded active running Avahi mDNS/DNS-SD Stack
  colord.service                      loaded active running Manage, Install and Generate Color Profiles
  console-setup.service               loaded active exited  Set console font and keymap
  cron.service                        loaded active running Regular background program processing daemon
  cups-browsed.service                loaded active running Make remote CUPS printers available locally
  cups.service                        loaded active running CUPS Scheduler
  dbus.service                        loaded active running D-Bus System Message Bus
  gdm.service                         loaded active running GNOME Display Manager
  gnome-remote-desktop.service        loaded active running GNOME Remote Desktop
```

Why Monitoring Services Matters

Running services consume system resources and may expose vulnerabilities. Monitoring helps identify suspicious or unnecessary services that attackers could exploit.

- **Section 8: Disabling Unnecessary Services**

Example Service Disabled

sudo systemctl stop bluetooth

sudo systemctl disable Bluetooth

```
vboxuser@Ubuntu:~$ sudo systemctl stop bluetooth
sudo systemctl disable bluetooth
Synchronizing state of bluetooth.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install disable bluetooth
Removed "/etc/systemd/system/dbus-org.bluez.service".
Removed "/etc/systemd/system/bluetooth.target.wants/bluetooth.service".
```

Why It Reduces Attack Surface

Disabling unused services minimizes potential entry points for attackers, reduces resource usage, and improves overall system security.

Section 9: OS Hardening Best Practices

Password Policy

- Use strong, complex passwords
- Avoid password reuse

```
vboxuser@Ubuntu:~$ chage -l vboxuser
Last password change                                    : Nov 08, 2025
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

Updates

- Regular system updates patch known vulnerabilities

```
vboxuser@Ubuntu:~$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1,404 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [9,724 B]
```

Firewall

- Always keep firewall enabled

```
vboxuser@Ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
vboxuser@Ubuntu:~$ sudo ufw status
Status: active
```

Service Minimization

- Disable unused services and ports

```
vboxuser@Ubuntu:~$ systemctl list-units --type=service
  UNIT                                LOAD   ACTIVE SUB     DESCRIPTION
  accounts-daemon.service             loaded active running Accounts Service
  alsa-restore.service                loaded active exited  Save/Restore Sound Card State
  apache2.service                     loaded active running The Apache HTTP Server
  apparmor.service                    loaded active exited  Load AppArmor profiles
  apport.service                      loaded active exited  automatic crash report generation
  avahi-daemon.service                loaded active running Avahi mDNS/DNS-SD Stack
  colord.service                      loaded active running Manage, Install and Generate Color Profiles
  console-setup.service               loaded active exited  Set console font and keymap
  cron.service                        loaded active running Regular background program processing daemon
  cups-browsed.service                loaded active running Make remote CUPS printers available locally
  cups.service                        loaded active running CUPS Scheduler
  dbus.service                        loaded active running D-Bus System Message Bus
  gdm.service                         loaded active running GNOME Display Manager
  gnome-remote-desktop.service        loaded active running GNOME Remote Desktop
```

Logging

- Monitor system logs to detect suspicious activities

```
vboxuser@Ubuntu:~$ sudo tail -f /var/log/syslog
2026-01-16T06:47:25.982529+00:00 Ubuntu systemd[1]: Finished esm-cache.service - Update the local
2026-01-16T06:47:53.491174+00:00 Ubuntu dbus-daemon[682]: [system] Activating via systemd: servic
d=0 pid=4489 comm="/usr/bin/gdbus call --system --dest org.freedeskto" label="unconfined")
2026-01-16T06:47:53.506295+00:00 Ubuntu systemd[1]: Starting packagekit.service - PackageKit Daem
2026-01-16T06:47:53.520069+00:00 Ubuntu PackageKit: daemon start
2026-01-16T06:47:53.549207+00:00 Ubuntu dbus-daemon[682]: [system] Successfully activated service
2026-01-16T06:47:53.550571+00:00 Ubuntu systemd[1]: Started packagekit.service - PackageKit Daemo
2026-01-16T06:48:05.821206+00:00 Ubuntu kernel: workqueue: e1000_watchdog [e1000] hogged CPU for
2026-01-16T06:48:28.870468+00:00 Ubuntu kernel: workqueue: blk_mq_run_work_fn hogged CPU for >100
2026-01-16T06:48:51.421950+00:00 Ubuntu systemd[1]: apt-news.service: Deactivated successfully.
2026-01-16T06:48:51.422212+00:00 Ubuntu systemd[1]: Finished apt-news.service - Update APT News.
```

Antivirus

- Use Windows Defender or Linux security tools for malware detection

```
vboxuser@Ubuntu:~$ clamscan -r /home
LibClamAV Error: cli_loaddbdir: No supported database files found in /var/lib/clamav
ERROR: Can't open file or directory

----------- SCAN SUMMARY -----------
Known viruses: 0
Engine version: 1.4.3
Scanned directories: 0
Scanned files: 0
Infected files: 0
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 0.011 sec (0 m 0 s)
Start Date: 2026:01:16 06:50:28
End Date:   2026:01:16 06:50:28
```

Section 10 : Conclusion

What I Learned

Through this task, I learned how operating systems can be secured by managing users, permissions, services, and network access.

Why OS Hardening is Important

OS hardening reduces vulnerabilities, limits attacker capabilities, and strengthens the overall security posture of a system, making it a critical practice in cybersecurity.