

Operating System Security Fundamentals

1 Introduction

Operating System (OS) hardening is the process of securing an operating system by reducing vulnerabilities, minimizing the attack surface, and applying security best practices. This practical demonstrates essential OS hardening techniques using Linux and Windows systems.

Installation of Linux Virtual Machine

A Linux virtual machine can be installed using VirtualBox, VMware, or Windows Subsystem for Linux (WSL). Virtualization allows safe experimentation without affecting the host operating system.

Steps

- Install VirtualBox
- Create a new virtual machine
- Attach Linux ISO file
- Complete installation process

Screenshot



Figure 1: Linux Virtual Machine

User Accounts and Access Control

Linux supports multiple users with different privilege levels. Access control ensures that users only perform authorized actions.

1.1 Types of Users

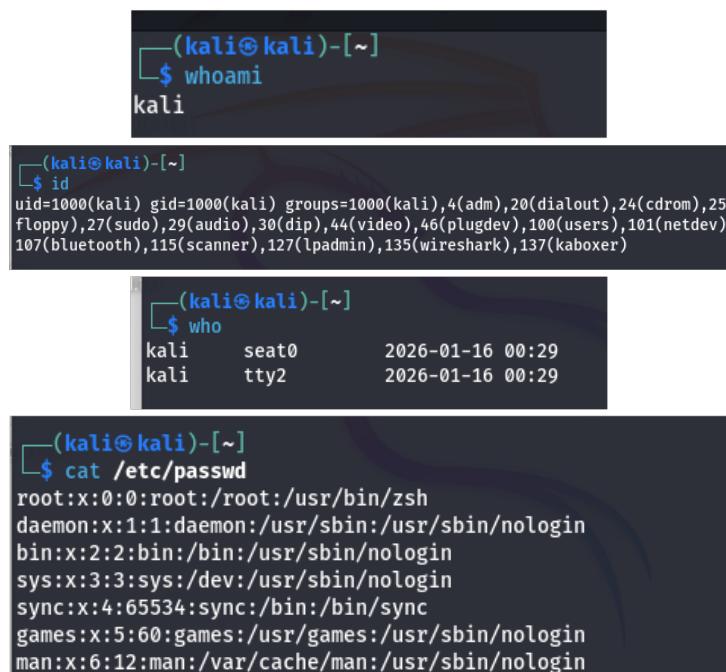
1.1.1 Root User

- Username: `root`
- User ID (UID): 0
- Has full administrative control over the system
- Can read, write, and execute any file
- Dangerous if misused due to unrestricted privileges

1.1.2 Normal Users

- UID: 1000 and above
- Limited privileges compared to the root user
- Used for daily and routine operations

Screenshot



The figure consists of four vertically stacked terminal windows from a Kali Linux environment. The top window shows the command `$ whoami` with the output "kali". The second window shows the command `$ id` with the output detailing the user's UID (1000), GID (1000), and group memberships (kali, adm, dialout, cdrom, floppy, sudo, audio, dip, video, plugdev, users, netdev, bluetooth, scanner, lpadmin, wireshark, kaboxer). The third window shows the command `$ who` with the output showing two entries: "kali seat0 2026-01-16 00:29" and "kali tty2 2026-01-16 00:29". The bottom window shows the command `$ cat /etc/passwd` with the output listing various system accounts and their details.

```
(kali㉿kali)-[~]
$ whoami
kali

(kali㉿kali)-[~]
$ id
uid=1000(kali) gid=1000(kali) groups=1000(kali),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),101(netdev),107(bluetooth),115(scanner),127(lpadmin),135(wireshark),137(kaboxer)

(kali㉿kali)-[~]
$ who
kali      seat0      2026-01-16 00:29
kali      tty2       2026-01-16 00:29

(kali㉿kali)-[~]
$ cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

Figure 2: User Accounts and Group Memberships

File Permissions Management

Linux file permissions determine who can read, write, or execute a file.

Breakdown of File Permissions

Part	Meaning
-	File type (- = file, d = directory)
rwx	Owner permissions
r-x	Group permissions
r-	Others permissions

Permission Meanings

- **r** → Read
- **w** → Write
- **x** → Execute

Commands

```
ls -l  
chmod 755 filename  
chown user:group filename
```

Screenshot

```
(kali㉿kali)-[~]
└─$ chmod 755 newfile.txt

(kali㉿kali)-[~]
└─$ ls -l newfile.txt
-rwxr-xr-x 1 kali root 0 Oct 31 02:28 newfile.txt

(kali㉿kali)-[~]
└─$ sudo chown root newfile.txt

(kali㉿kali)-[~]
└─$ ls -l newfile.txt
-rwxr-xr-x 1 root root 0 Oct 31 02:28 newfile.txt
```

Symbolic	Numeric	Permission
---	0	None
--x	1	Execute
-w-	2	Write
-wx	3	Write + Execute
r--	4	Read
r-x	5	Read + Execute
rw-	6	Read + Write
rwx	7	Read + Write + Execute

Figure 3: File Permission Management using chmod and chown

Administrator vs Standard User Privileges

Administrator users have full control over the system, while standard users have limited access.

1. Administrator (Root) Privileges

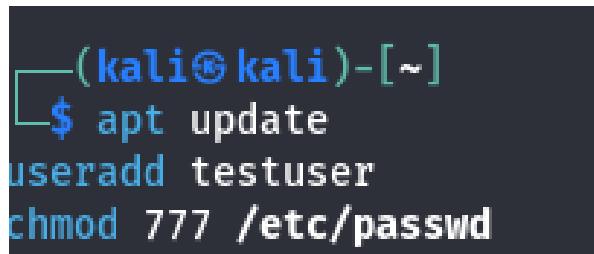
Who is the Administrator?

- The **root user** is the system administrator in Linux.
- The root user has a **UID = 0**.
- Root has **unrestricted access** to the entire system.

Capabilities of Root User

- Install and remove software packages.
- Modify critical system files such as `/etc`, `/boot`, and `/usr`.
- Create, modify, and delete users and groups.
- Start, stop, and manage system services.
- Change file permissions and ownership.
- Access and manage all user data on the system.

Screenshot



A terminal window showing a root shell. The prompt is `(kali㉿kali)-[~]`. The user has run the command `apt update`, then used `useradd testuser` to create a new user, and finally used `chmod 777 /etc/passwd` to change the permissions of the password file.

```
(kali㉿kali)-[~]
$ apt update
useradd testuser
chmod 777 /etc/passwd
```

Figure 4: Administrator Privilege using sudo Command

2. Standard User Privileges

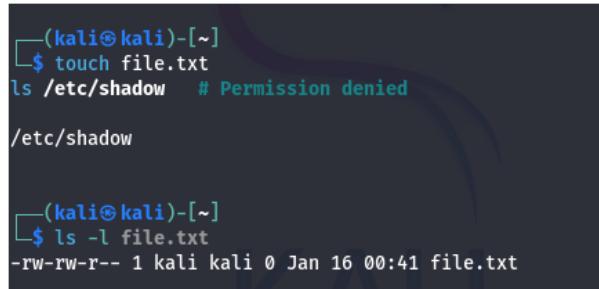
Who is a Standard User?

- A standard user is a normal user account (e.g., `kali`).
- Standard users typically have a **UID ≥ 1000** .
- They have **limited permissions** by default for system security.

Capabilities of Standard User

- Access their own home directory.
- Run user-level applications.
- Create and modify personal files.
- Cannot modify system files.
- Cannot install system-wide software.
- Cannot manage other users.

Screenshot



```
(kali㉿kali)-[~]
└─$ touch file.txt
ls /etc/shadow  # Permission denied

/etc/shadow

(kali㉿kali)-[~]
└─$ ls -l file.txt
-rw-rw-r-- 1 kali kali 0 Jan 16 00:41 file.txt
```

Figure 5: Standard user using sudo Command

Firewall Configuration

Firewalls control incoming and outgoing network traffic.

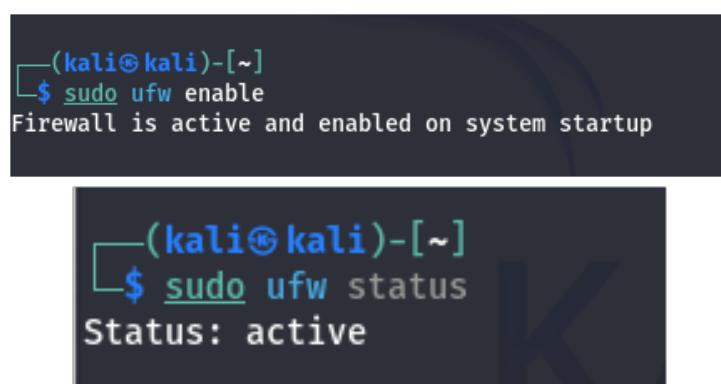
2 Importance of UFW (Security Perspective)

- Blocks unauthorized network access to the system.
- Reduces the overall attack surface by limiting open ports.
- Protects the system against port scanning and brute-force attempts.
- Essential for securing servers and Kali Linux laboratory environments.

Linux (UFW)

```
sudo ufw enable
sudo ufw status
```

Screenshot



```
(kali㉿kali)-[~]
└─$ sudo ufw enable
Firewall is active and enabled on system startup

(kali㉿kali)-[~]
└─$ sudo ufw status
Status: active
```

Figure 6: Firewall Enabled using UFW

3 Identifying Running Processes and Services

Monitoring active processes helps detect malicious or unnecessary applications.

Commands

```
ps aux  
top  
systemctl list-units --type=service
```

Screenshot

The screenshot shows a terminal session on a Kali Linux system. The user runs several commands to view system activity:

- `ps`: Shows a list of running processes. The output includes columns for PID, TTY, TIME, and CMD. It shows two processes: zsh (PID 2763) and ps (PID 10830).
- `systemctl list-units --type=service`: Shows a list of active system services. The output includes columns for UNIT, LOAD, ACTIVE, SUB, and DESCRIPTION. Services listed include accounts-daemon.service, colord.service, cron.service, dbus.service, gdm.service, haveged.service, ifupdown-pre.service, keyboard-setup.service, kmod-static-nodes.service, ModemManager.service, networking.service, NetworkManager-wait-online.service, and NetworkManager.service.
- `ps aux`: Shows a detailed list of all processes. The output includes columns for USER, PID, %CPU, %MEM, VSZ, RSS, TTY, STAT, START, TIME, and COMMAND. Most processes are owned by root and are in the S state.
- `top`: Shows a live update of system performance metrics and a list of processes. The output includes columns for PID, USER, PR, NI, VIRT, RES, SHR, S, %CPU, %MEM, and TIME+. It shows five processes: kali, root, and four other root processes. The system load average is 0.01, 0.05, 0.07.

Figure 7: Viewing Running Processes and Services

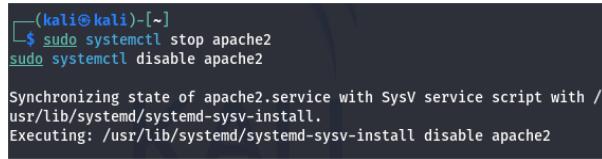
4 Disabling Unnecessary Services

Unused services increase the attack surface and should be disabled.

Commands

```
sudo systemctl stop servicename  
sudo systemctl disable servicename
```

Screenshot



A terminal window showing the command to disable the apache2 service. The command is \$ sudo systemctl stop apache2 followed by \$ sudo systemctl disable apache2. The output shows the service is being synchronized with a SysV service script and then executing the disable command.

```
(kali㉿kali)-[~]  
$ sudo systemctl stop apache2  
$ sudo systemctl disable apache2  
  
Synchronizing state of apache2.service with SysV service script with /  
usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install disable apache2
```

Figure 8: Disabling Unnecessary Services

5 Best Practices for OS Hardening

Applying best practices improves overall system security.

Best Practices

- Keep the system updated
- Use strong passwords
- Disable unused services
- Configure firewalls
- Apply least privilege principle

6 Summary

Operating system hardening is a crucial security practice that helps protect systems from cyber threats. By implementing user control, file permissions, firewall rules, and service management, system security can be significantly improved.