

# Operating System Security Fundamentals (Linux & Windows)

## 1] Windows OS Security

Administrator user and Standard user

- In Windows, an Administrator account has full control over the PC, allowing software installation, system changes, and access to all files.
- A Standard User account has limited permissions for everyday tasks like browsing and email, requiring administrator approval (password prompt) for critical actions, making it safer against malware.

Least Privilege Principle

- The least privilege principle is a security concept that states that a user, application, or system process should be given only the minimum level of access or permissions required to perform its intended task, and nothing more.
- This principle helps reduce security risks by limiting the potential damage caused by accidents, misuse, or cyber attacks.



Summary and Observation:

- Checked user accounts and understood the difference between administrator and standard user privileges, observing the use of UAC for controlled access.
- Verified that Windows Defender real-time and cloud-based protection were enabled to protect against malware and threats.
- Explored Windows Firewall settings and confirmed that firewall protection was active for all network profiles.
- Reviewed running processes and startup applications using Task Manager, noting that unnecessary services increase the attack surface.

- Checked Windows Update status and observed the importance of regular patching for OS security.
- Gained an overall understanding of Windows OS hardening practices and least privilege principle.

## 2] Linux OS Security

```

Session Actions Edit View Help
[(kali㉿kali)-~]
$ whoami
kali

[(kali㉿kali)-~]
$ sudo whoami
[sudo] password for kali:
root

[(kali㉿kali)-~]
$ touch testfile.txt

[(kali㉿kali)-~]
$ ls -l
total 32
drwxr-xr-x 2 kali kali 4096 Jan 16 13:38 Desktop
drwxr-xr-x 2 kali kali 4096 Jan 16 13:38 Documents
drwxr-xr-x 2 kali kali 4096 Jan 16 13:38 Downloads
drwxr-xr-x 2 kali kali 4096 Jan 16 13:38 Music
drwxr-xr-x 2 kali kali 4096 Jan 16 13:38 Pictures
drwxr-xr-x 2 kali kali 4096 Jan 16 13:38 Public
drwxr-xr-x 2 kali kali 4096 Jan 16 13:38 Templates
-rw-rw-r-- 1 kali kali 0 Jan 16 16:24 testfile.txt
drwxr-xr-x 2 kali kali 4096 Jan 16 13:38 Videos

[(kali㉿kali)-~]
$ chmod 0-r testfile.txt

[(kali㉿kali)-~]
$ ls -l testfile.txt
-rw-rw-r-- 1 kali kali 0 Jan 16 16:24 testfile.txt

[(kali㉿kali)-~]
$ 

```

**touch** → command to create or update a file

**ls -l** → list files or directories in long format (shows detailed information)

**ls -l** is used to view file permissions, ownership, and access control details in Linux.

Output : -rw-r--r-- 1 kali kali 0 Jan 16 12:24 testfile.txt

**-** → regular file

**rw-** : read, write for owner

**r-** : read only for groups

**r-** : read only for others

**1** = one reference to this file

**kali** : File is owned by user kali

**kali** : File belongs to group kali

**0** : Size is in bytes since **touch** created an empty file, size is **0**.

**Jan 16 12:24** : Last modification time

```
chmod o-r testfile.txt
```

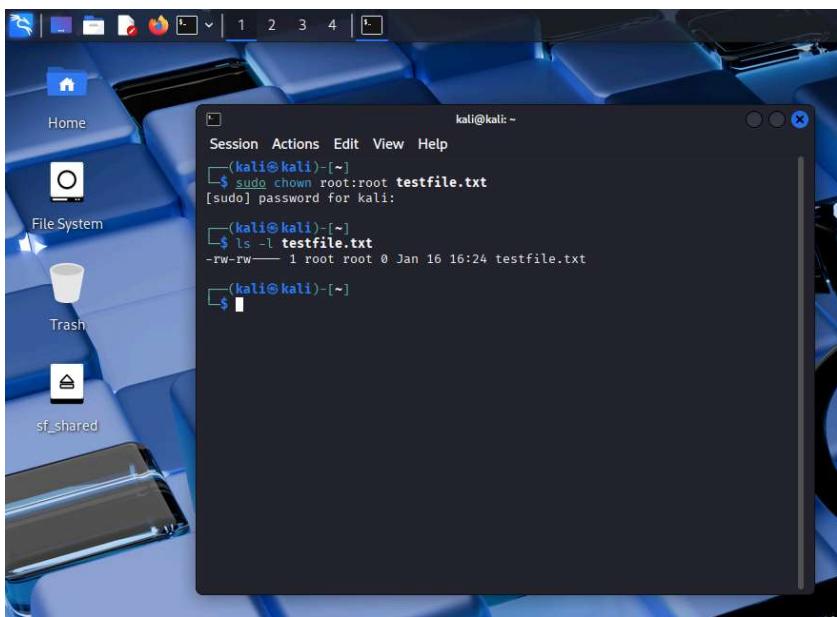
**chmod** = change mode; Used to change file permissions in Linux

o : others

- : remove permissions

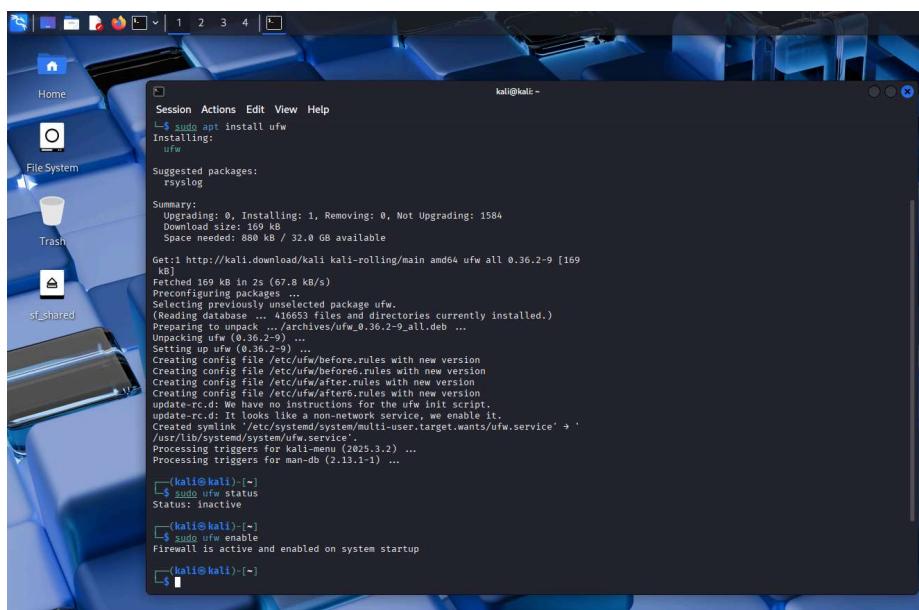
r : read permission

This command **removes read permission from others** for the file **testfile.txt**.



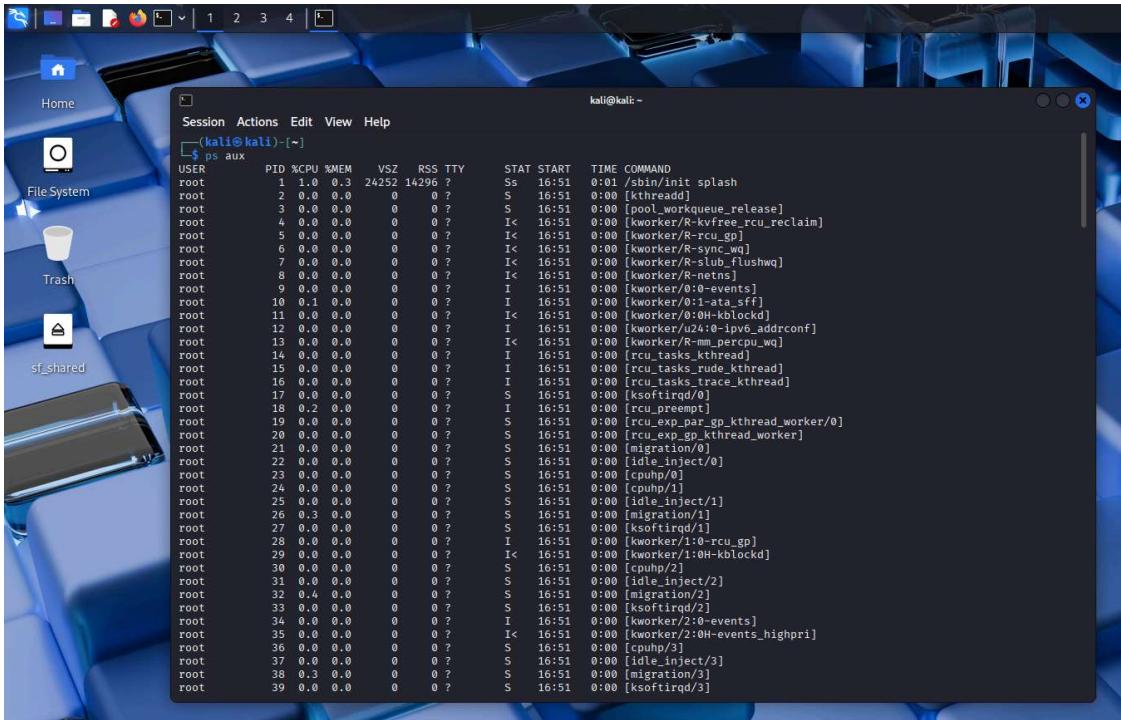
**chown** : to change owner from user to root

Ownership of the file changed to root, improving security control.



**Ufw** : Stands for Uncomplicated Firewall

Firewall helps control incoming and outgoing traffic.



```
(kali㉿kali)-[~]
Session Actions Edit View Help
[kali㉿kali)-[~]
$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root      1  0.0  0.3 24252 14296 ?        Ss   16:51  0:01 /sbin/init splash
root      2  0.0  0.0     0  0 ?        S    16:51  0:00 [kthreadd]
root      3  0.0  0.0     0  0 ?        S    16:51  0:00 [pool_workqueue_release]
root      4  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/R-kvfree_rcu_reclaim]
root      5  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/R-rcu_gp]
root      6  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/R-sync_wq]
root      7  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/R-sub_flushwq]
root      8  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/R-netns]
root      9  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/R-threads]
root     10  0.1  0.0     0  0 ?        I<   16:51  0:00 [kworker/01-ata_sff]
root     11  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/01H-kblockd]
root     12  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/u24:0-ipv6_addrconf]
root     13  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/R-mm_percpu_wq]
root     14  0.0  0.0     0  0 ?        I<   16:51  0:00 [rcu_tasks_kthread]
root     15  0.0  0.0     0  0 ?        I<   16:51  0:00 [rcu_tasks_rude_kthread]
root     16  0.0  0.0     0  0 ?        I<   16:51  0:00 [rcu_tasks_trace_kthread]
root     17  0.0  0.0     0  0 ?        I<   16:51  0:00 [ksoftirqd/0]
root     18  0.2  0.0     0  0 ?        I<   16:51  0:00 [rcu_preempt]
root     19  0.0  0.0     0  0 ?        S    16:51  0:00 [rcu_exp_par_gp_kthread_worker/0]
root     20  0.0  0.0     0  0 ?        S    16:51  0:00 [rcu_exp_gp_kthread_worker]
root     21  0.0  0.0     0  0 ?        S    16:51  0:00 [migration/0]
root     22  0.0  0.0     0  0 ?        S    16:51  0:00 [idle_inject/0]
root     23  0.0  0.0     0  0 ?        S    16:51  0:00 [cpump/0]
root     24  0.0  0.0     0  0 ?        S    16:51  0:00 [cpump/1]
root     25  0.0  0.0     0  0 ?        S    16:51  0:00 [idle_inject/1]
root     26  0.3  0.0     0  0 ?        S    16:51  0:00 [migration/1]
root     27  0.0  0.0     0  0 ?        S    16:51  0:00 [ksoftirqd/1]
root     28  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/10H-rcu_gp]
root     29  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/10H-kblockd]
root     30  0.0  0.0     0  0 ?        S    16:51  0:00 [cpump/2]
root     31  0.0  0.0     0  0 ?        S    16:51  0:00 [idle_inject/2]
root     32  0.4  0.0     0  0 ?        S    16:51  0:00 [migration/2]
root     33  0.0  0.0     0  0 ?        S    16:51  0:00 [ksoftirqd/2]
root     34  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/2:0-events]
root     35  0.0  0.0     0  0 ?        I<   16:51  0:00 [kworker/2:0-events_highpri]
root     36  0.0  0.0     0  0 ?        S    16:51  0:00 [cpump/3]
root     37  0.0  0.0     0  0 ?        S    16:51  0:00 [idle_inject/3]
root     38  0.3  0.0     0  0 ?        S    16:51  0:00 [migration/3]
root     39  0.0  0.0     0  0 ?        S    16:51  0:00 [ksoftirqd/3]
```

Multiple processes run in the background; unnecessary services increase attack surface.

### Summary & observations:

- Checked the current user and understood the difference between a normal user and root by using `sudo`, highlighting the concept of administrative privileges.
- Created a test file and analyzed file permissions using `ls -l`, learning how read, write, and execute permissions are assigned to user, group, and others.
- Modified file permissions using `chmod` and changed file ownership using `chown` to understand access control and least privilege.
- Installed and enabled UFW firewall, observing how firewalls help control network traffic and reduce the attack surface.
- Viewed running processes to understand how active services contribute to system exposure.
- Learned key Linux OS hardening practices such as limiting root access, securing file permissions, and enabling firewall protection.

## **OS Security Checklist (Windows & Linux)**

### **1. User Accounts & Privileges**

- Ensure separation between administrator/root and standard users
- Follow the least privilege principle
- Use sudo only when administrative access is required
- Avoid logging in as root by default (Linux)

### **2. Authentication & Access Control**

- Enable User Account Control (UAC) on Windows
- Use strong passwords for all user accounts
- Restrict access to sensitive files and system settings
- Monitor user permissions regularly

### **3. File Permissions & Ownership (Linux)**

- Check file permissions using ls -l
- Use chmod to restrict read, write, and execute access
- Use chown to assign correct file ownership
- Avoid giving write permissions to others unnecessarily

### **4. Antivirus & Malware Protection**

- Ensure Windows Defender real-time protection is enabled
- Keep malware definitions up to date
- Avoid downloading untrusted files or applications

### **5. Firewall Configuration**

- Enable Windows Firewall for all network profiles
- Install and enable UFW firewall in Linux
- Verify firewall status regularly
- Allow only necessary network traffic

### **6. Running Processes & Services**

- Monitor active processes using Task Manager (Windows) or ps aux (Linux)
- Identify and disable unnecessary services
- Reduce background applications to minimize attack surface

### **7. Startup Applications & Services**

- Review startup programs in Windows Task Manager
- Disable unnecessary startup services
- Prevent unauthorized applications from running at boot

## 8. System Updates & Patch Management

- Enable automatic updates in Windows
- Regularly update Linux packages
- Apply security patches promptly to fix vulnerabilities

## 9. OS Hardening Best Practices

- Use strong passwords and screen locks
- Enable firewall and antivirus at all times
- Limit administrative access
- Keep the OS and applications updated
- Remove unused software and services

## 10. Monitoring & Awareness

- Regularly review system security settings
- Stay aware of common OS-level threats
- Follow secure usage practices to prevent misuse or attacks

### **Interview Questions:**

What is OS hardening?

What are file permissions in Linux?

Why should unnecessary services be disabled?

Difference between root and normal user?

What is the least privilege principle?