

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.

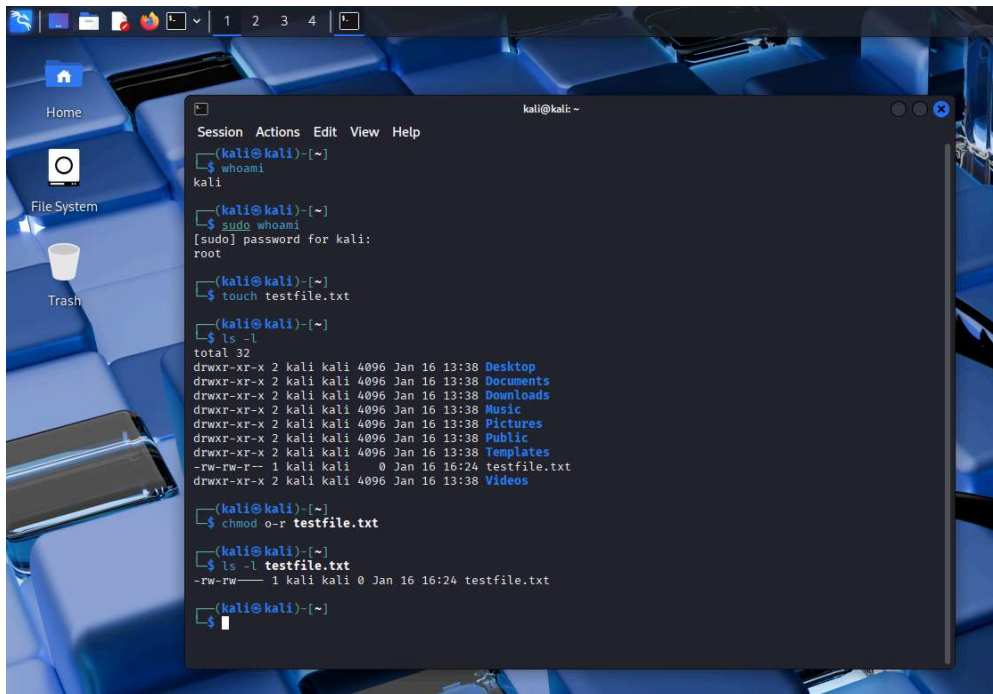


Summary and Observation:

- 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



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`

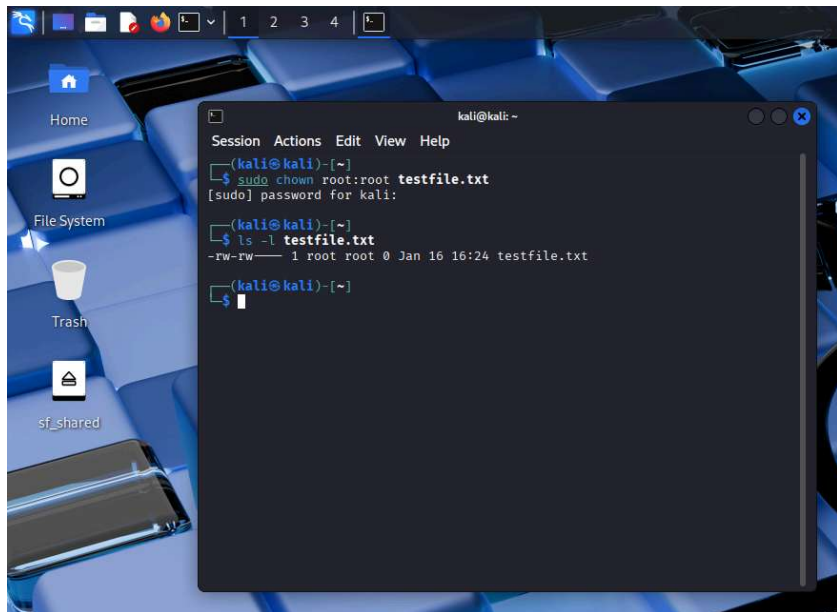
chmod = change mode; Used to change file permissions in Linux

o : others

- : remove permissions

r : read permission

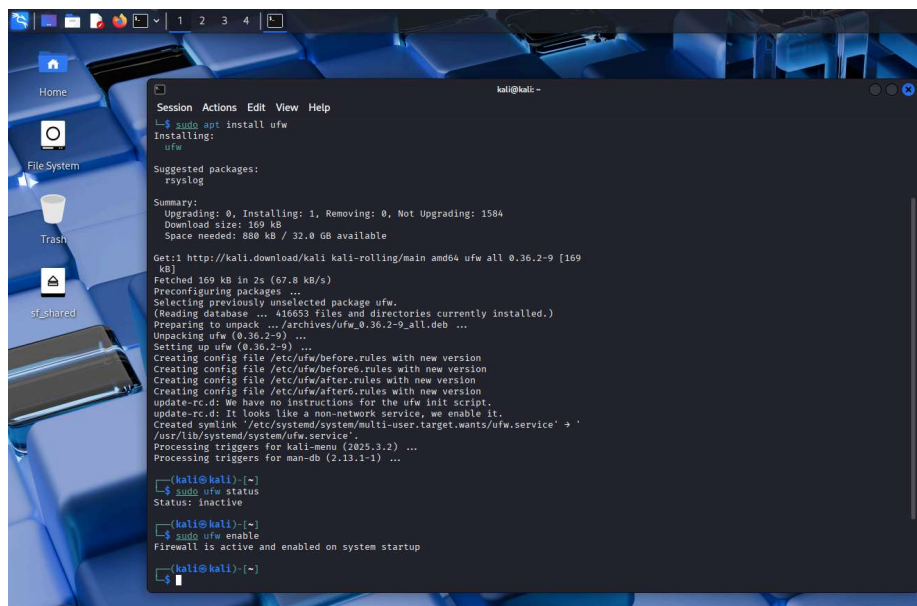
This command **removes read permission from others** for the file `testfile.txt`.

A terminal window on a Kali Linux desktop. The user runs `sudo chown root:root testfile.txt` and then `ls -l testfile.txt`. The output shows the file is now owned by root:root with permissions `-rw-rw----`.

```
kali@kali: ~  
Session Actions Edit View Help  
kali@kali:~$ sudo chown root:root testfile.txt  
[sudo] password for kali:  
kali@kali:~$ ls -l testfile.txt  
-rw-rw---- 1 root root 0 Jan 16 16:24 testfile.txt  
kali@kali:~$
```

chown : to change owner from user to root

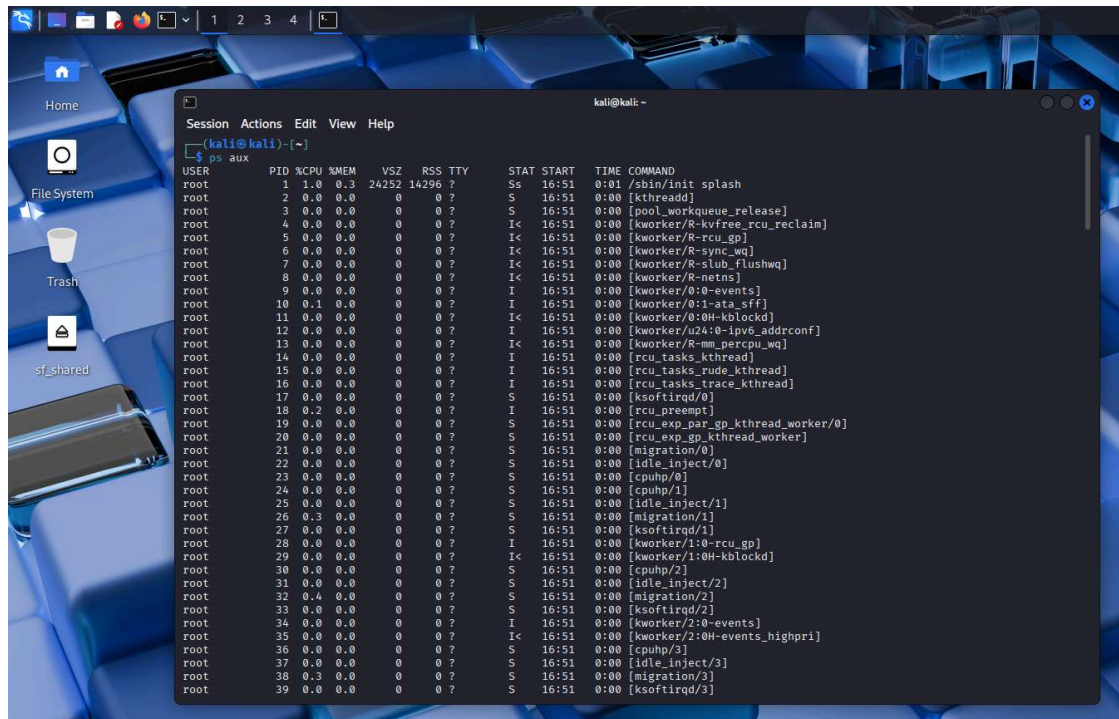
Ownership of the file changed to root, improving security control.

A terminal window on a Kali Linux desktop showing the installation of UFW. The user runs `sudo apt install ufw`, followed by `sudo ufw status` (showing inactive) and `sudo ufw enable` (showing active).

```
kali@kali: ~  
Session Actions Edit View Help  
kali@kali:~$ sudo apt install ufw  
Installing:  
  ufw  
Suggested packages:  
  rsyslog  
Summary:  
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1584  
  Download size: 169 kB  
  Space needed: 808 kB / 32.0 GB available  
Get:1 http://kali.download/kali kali-rolling/main amd64 ufw all 0.36.2-9 [169 kB]  
Fetched 169 kB in 2s (67.8 kB/s)  
Preconfiguring packages ...  
Selecting previously unselected package ufw.  
(Reading database ... 416653 files and directories currently installed.)  
Preparing to unpack .../archives/ufw_0.36.2-9_all.deb ...  
Unpacking ufw (0.36.2-9) ...  
Setting up ufw (0.36.2-9) ...  
Creating config file /etc/ufw/before.rules with new version  
Creating config file /etc/ufw/before.rules with new version  
Creating config file /etc/ufw/after.rules with new version  
Creating config file /etc/ufw/after.rules with new version  
update-rc.d: We have no instructions for the ufw init script.  
update-rc.d: It looks like a non-network service, we enable it.  
Created symlink '/etc/systemd/system/multi-user.target.wants/ufw.service' -> '/usr/lib/systemd/system/ufw.service'.  
Processing triggers for kali-menu (2025.3.2) ...  
Processing triggers for man-db (2.13.1-1) ...  
kali@kali:~$ sudo ufw status  
Status: inactive  
kali@kali:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
kali@kali:~$
```

Ufw : Stands for Uncomplicated Firewall

Firewall helps control incoming and outgoing traffic.



```
kali@kali: ~  
$ ps aux  
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
root           1   0.0  0.0  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-slub_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/0:0-events]  
root          10   0.1  0.0      0     0 ?        I    16:51   0:00 [kworker/0:1-ata_sff]  
root          11   0.0  0.0      0     0 ?        I<   16:51   0:00 [kworker/0:0H-kblockd]  
root          12   0.0  0.0      0     0 ?        I    16:51   0:00 [kworker/u24:0-lpv6_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 ?        S    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 [cpuhp/0]  
root          24   0.0  0.0      0     0 ?        S    16:51   0:00 [cpuhp/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/1:0-rcu_gp]  
root          29   0.0  0.0      0     0 ?        I<   16:51   0:00 [kworker/1:0H-kblockd]  
root          30   0.0  0.0      0     0 ?        S    16:51   0:00 [cpuhp/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/2:0H-events_highpri]  
root          36   0.0  0.0      0     0 ?        S    16:51   0:00 [cpuhp/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?