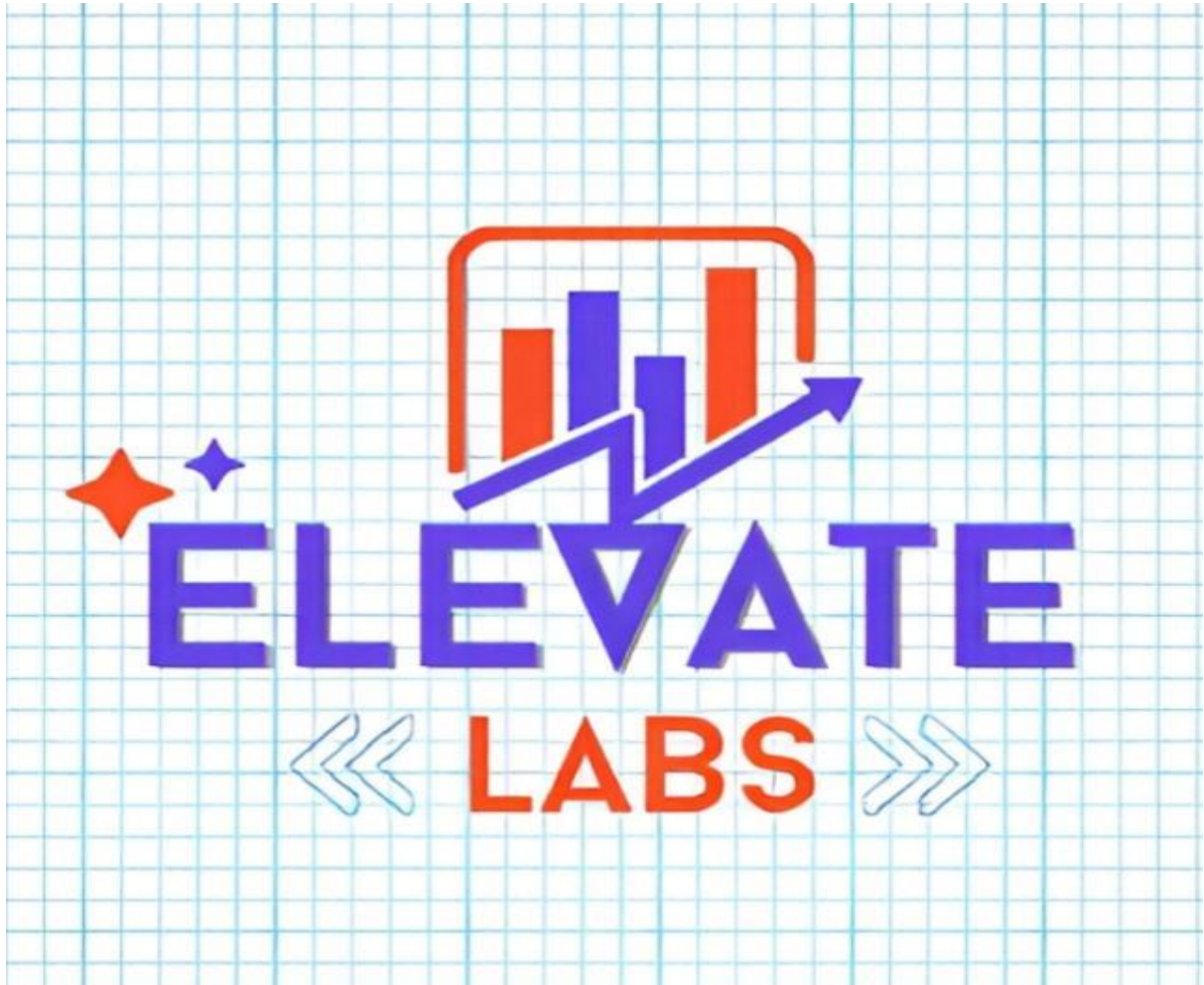


## TASK 14 Linux Server Hardening & Secure Configuration



*Name: - Ch Rohit Kumar*

*University: - Aditya University*

*Gmail: - rohitkumarrrr20@gmail.com*

## **1. Review default Linux system settings to understand users, services, and open ports.**

*Here is the theory explanation suitable for exams (written format, no commands focus, more conceptual):*

---

### **Review of Default Linux System Settings (Theory)**

*Reviewing default Linux system settings is an important task in system administration and cybersecurity. It helps administrators understand how the operating system is configured after installation and ensures that the system is secure and properly managed.*

*The main areas to review include:*

- *Users and user accounts*
- *System services*
- *Open ports and network configuration*
- *System security settings*

---

### **1. Users in Linux**

*Linux is a multi-user operating system, meaning multiple users can access and use the system simultaneously.*

#### **Types of Users:**

- 1. Root User**
  - *The superuser of the system.*
  - *Has complete control over the system.*
  - *Can install software, modify system files, and manage other users.*
- 2. System Users**
  - *Created automatically during installation.*
  - *Used by services and background processes.*
  - *Usually do not have login access.*
- 3. Regular Users**
  - *Created by the administrator.*
  - *Used for daily tasks.*
  - *Have limited permissions for security reasons.*

*Each user has:*

- *A unique User ID (UID)*
- *A home directory*
- *A default shell*
- *Group membership*

*Reviewing users helps detect:*

- *Unauthorized accounts*
- *Unnecessary privileged users*
- *Misconfigured permissions*

---

### **2. Services in Linux**

*Services are background programs that start during boot and run continuously to perform specific tasks.*

*Examples:*

- *SSH service (remote login)*
- *Web server (Apache/Nginx)*
- *Database server (MySQL)*
- *Cron (scheduled tasks)*

*Modern Linux systems use systemd to manage services.*

*Services can be:*

- *Running*
- *Stopped*
- *Enabled (start at boot)*
- *Disabled*

*Reviewing services helps to:*

- *Identify unnecessary services*
- *Reduce system resource usage*
- *Minimize attack surface*
- *Improve security*

*If too many services are running, the system becomes more vulnerable to attacks.*

---

### **3. Open Ports and Network Settings**

*Ports are communication endpoints used by services to send and receive data over a network.*

*For example:*

- *Port 22 → SSH*
- *Port 80 → HTTP*
- *Port 443 → HTTPS*

*When a service runs, it may open a port to accept network connections.*

*Reviewing open ports helps:*

- *Identify which services are accessible from outside*
- *Detect suspicious or unauthorized services*
- *Prevent remote exploitation*

*Unnecessary open ports increase security risks.*

---

### **4. Default Security Settings**

*After installation, Linux has default security configurations, such as:*

- *File permissions*
- *Firewall settings*
- *Password policies*
- *SELinux or AppArmor (security modules)*

*Reviewing these settings ensures:*

- *Proper access control*
- *Strong password enforcement*
- *Firewall protection*

- **Restricted root access**

## 5. Importance of Reviewing Default Settings

**Reviewing default system settings is important because:**

- **It helps in system hardening.**
- **It reduces security vulnerabilities.**
- **It ensures compliance with security policies.**
- **It prevents unauthorized access.**
- **It improves system performance and stability.**

**In cybersecurity, reviewing system settings is often the first step in:**

- **Security audits**
- **Penetration testing**
- **Incident response**
- **Server deployment**

## Check Default Users

**List all users:**

**cat /etc/passwd**

```
(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:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
system-network:x:998:998:systemd Network Management:/:usr/sbin/nologin
dhcpcd:x:100:65534:dhcpcd:/usr/lib/dhcpcd:/bin/false
sentrypeer:x:101:103::/var/lib/sentrypeer:/usr/sbin/nologin
tss:x:102:104:TPM software stack:/var/lib/tpm:/bin/false
strongswan:x:103:65534::/var/lib/strongswan:/usr/sbin/nologin
systemd-timesync:x:991:991:systemd Time Synchronization:/:usr/sbin/nologin
Debian-exim:x:104:105::/var/spool/exim4:/usr/sbin/nologin
uidd:x:105:106::/run/uidd:/usr/sbin/nologin
messagebus:x:990:990:system message bus:/nonexistent:/usr/sbin/nologin
clamav:x:106:107::/var/lib/clamav:/bin/false
tcpdump:x:107:108::/nonexistent:/usr/sbin/nologin
_ftp:x:108:65534::/run/ftplib:/usr/sbin/nologin
redis:x:109:111::/var/lib/redis:/usr/sbin/nologin
sshd:x:989:65534:sshd user:/run/sshd:/usr/sbin/nologin
dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin
avahi:x:110:114:Avahi mDNS daemon:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:111:29:Speech Dispatcher:/run/speech-dispatcher:/bin/false
postgres:x:112:115:PostgreSQL administrator:/var/lib/postgresql:/bin/bash
usbmux:x:113:46:usbmux daemon:/var/lib/usbmux:/usr/sbin/nologin
nm-openvpn:x:114:116:NetworkManager OpenVPN:/var/lib/openvpn/chroot:/usr/sbin/nologin
inetsim:x:115:117::/var/lib/inetsim:/usr/sbin/nologin
pipewire:x:988:988:system user for pipewire:/nonexistent:/usr/sbin/nologin
defectdojo:x:116:119::/var/log/defectdojo:/usr/sbin/nologin
nm-openconnect:x:987:987:NetworkManager OpenConnect plugin:/:usr/sbin/nologin
lightdm:x:117:120:Light Display Manager:/var/lib/lightdm:/bin/false
statd:x:118:65534::/var/lib/nfs:/usr/sbin/nologin
saned:x:119:121::/var/lib/saned:/usr/sbin/nologin
polkitd:x:986:986:User for polkitd:/:usr/sbin/nologin
rtkit:x:120:122:RealtimeKit:/proc:/usr/sbin/nologin
colord:x:985:985:colord colour management daemon:/var/lib/colord:/usr/sbin/nologin
```

Show only normal users (UID ≥ 1000):

**awk -F: '\$3 >= 1000 {print \$1}' /etc/passwd**

```
(kali@kali)~$ awk -F: '$3 >= 1000 {print $1}' /etc/passwd
nobody
kali
```

Currently logged-in users:

Who

```
(kali@kali)-[~]
$ who
kali      seat0      2026-02-09 14:17 (:0)
```

## 2. Review Groups

cat /etc/group

```
(kali@kali)-[~]
$ \cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:kali
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:kali
fax:x:21:
voice:x:22:
cdrom:x:24:kali
floppy:x:25:kali
tape:x:26:
sudo:x:27:kali
audio:x:29:kali
dip:x:30:kali
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
shadow:x:42:
utmp:x:43:
video:x:44:kali
```

## 3. Check Running Services Using systemctl (modern Linux):

systemctl list-units --type=service --state=running

```
(kali@kali)-[~]
$ systemctl list-units --type=service --state=running
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
accounts-daemon.service	loaded	active	running	Accounts Service
apache2.service	loaded	active	running	The Apache HTTP Server
colord.service	loaded	active	running	Manage, Install and Generate Color Profiles
containerd.service	loaded	active	running	containerd container runtime
cron.service	loaded	active	running	Regular background program processing daemon
dbus.service	loaded	active	running	D-Bus System Message Bus
docker.service	loaded	active	running	Docker Application Container Engine
getty@tty1.service	loaded	active	running	Getty on tty1
haveged.service	loaded	active	running	Entropy Daemon based on the HAVEGE algorithm
lightdm.service	loaded	active	running	Light Display Manager
lynis.service	loaded	active	running	Security audit and vulnerability scanner
mariadb.service	loaded	active	running	MariaDB 11.8.5 database server
ModemManager.service	loaded	active	running	Modem Manager
NetworkManager.service	loaded	active	running	Network Manager
polkit.service	loaded	active	running	Authorization Manager
rtkit-daemon.service	loaded	active	running	RealtimeKit Scheduling Policy Service
systemd-journald.service	loaded	active	running	Journal Service
systemd-logind.service	loaded	active	running	User Login Management
systemd-udev.service	loaded	active	running	Rule-based Manager for Device Events and Files
udisks2.service	loaded	active	running	Disk Manager
upower.service	loaded	active	running	Daemon for power management
user@1000.service	loaded	active	running	User Manager for UID 1000
virtualbox-guest-utils.service	loaded	active	running	Virtualbox guest utils

Legend: LOAD → Reflects whether the unit definition was properly loaded.  
ACTIVE → The high-level unit activation state, i.e. generalization of SUB.  
SUB → The low-level unit activation state, values depend on unit type.

23 loaded units listed.

View all enabled services:

systemctl list-unit-files --type=service



```
(kali@kali)~$ systemctl list-unit-files --type=service
UNIT FILE                                STATE                                PRESET
accounts-daemon.service                 enabled                             enabled
apache-httcacheclean.service            disabled                           disabled
apache2.service                         enabled                             disabled
apache2@.service                        disabled                           disabled
apparmor.service                        disabled                           disabled
apt-daily-upgrade.service               static                             -
apt-daily.service                       static                             -
auth-rpccss-module.service              alias                              -
avahi-daemon.service                    disabled                           disabled
bluetooth.service                       enabled                             enabled
blk-availability.service                 disabled                           disabled
bluetooth-mechanism.service             disabled                           disabled
breakpoint-pre-basic.service            static                             -
breakpoint-pre-mount.service            static                             -
breakpoint-pre-switch-root.service      static                             -
breakpoint-pre-udev.service             static                             -
capsule@.service                        static                             -
chkrootkit.service                      disabled                           disabled
clamav-freshclam-once.service            disabled                           disabled
clamav-freshclam.service                disabled                           disabled
colord.service                          disabled                           disabled
console-getty.service                   enabled                             disabled
console-setup.service                   static                             enabled
container-getty@.service                disabled                           disabled
containerd.service                      enabled                             disabled
cron.service                            masked                             disabled
cryptdisks-early.service                masked                             disabled
cryptdisks.service                      enabled                             disabled
dbus-org.freedesktop.hostname1.service alias                              -
dbus-org.freedesktop.locale1.service   alias                              -
dbus-org.freedesktop.login1.service     alias                              -
dbus-org.freedesktop.ModemManager1.service alias                              -
dbus-org.freedesktop.nm-dispatcher.service alias                              -
dbus-org.freedesktop.timedate1.service  alias                              -
dbus-org.freedesktop.timesync1.service  alias                              -
```

#### 4. Check Open Ports Using ss (recommended): ss -tln

```
(kali@kali)~$ ss -tln
Netid      State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port
tcp        LISTEN     0            4096        127.0.0.1:40231         0.0.0.0:*
tcp        LISTEN     0            80         127.0.0.1:3306          0.0.0.0:*
tcp        LISTEN     0            511        *:80                   *:80
tcp        LISTEN     0            511        *:511                  *:511
```

#### 5. See Which Service Uses Which Port sudo ss -tlnp

```
(kali@kali)~$ sudo ss -tlnp
[sudo] password for kali:
Netid      State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port      Process
tcp        LISTEN     0            4096        127.0.0.1:40231         0.0.0.0:*              users:((("containerd",pid=604,fd=9))
tcp        LISTEN     0            80         127.0.0.1:3306          0.0.0.0:*              users:((("mariadb",pid=689,fd=28))
tcp        LISTEN     0            511        *:80                   *:80                   users:((("apache2",pid=718,fd=4),("apache2",pid=717,fd=4),("apache2",pid=716,fd=4),("apache2",pid=715,fd=4),("apache2",pid=590,fd=4))
tcp        LISTEN     0            511        *:511                  *:511                  users:((("apache2",pid=719,fd=6),("apache2",pid=718,fd=6),("apache2",pid=717,fd=6),("apache2",pid=716,fd=6),("apache2",pid=715,fd=6),("apache2",pid=590,fd=6))
```

### Identify Startup Services

systemctl list-unit-files | grep enabled

```
(kali@kali)~$ systemctl list-unit-files | grep enabled
accounts-daemon.service                 enabled
apache2.service                         enabled
bluetooth.service                       enabled
bluetooth-mechanism.service             enabled
breakpoint-pre-basic.service            enabled
breakpoint-pre-mount.service            enabled
breakpoint-pre-switch-root.service      enabled
breakpoint-pre-udev.service             enabled
capsule@.service                        enabled
chkrootkit.service                      enabled
clamav-freshclam-once.service            enabled
clamav-freshclam.service                enabled
colord.service                          enabled
console-getty.service                   enabled
console-setup.service                   enabled
container-getty@.service                enabled
containerd.service                      enabled
cron.service                            enabled
cryptdisks-early.service                enabled
cryptdisks.service                      enabled
dbus-org.freedesktop.hostname1.service enabled
dbus-org.freedesktop.locale1.service   enabled
dbus-org.freedesktop.login1.service     enabled
dbus-org.freedesktop.ModemManager1.service enabled
dbus-org.freedesktop.nm-dispatcher.service enabled
dbus-org.freedesktop.timedate1.service  enabled
dbus-org.freedesktop.timesync1.service  enabled
dovecot.service                         enabled
docker.service                         enabled
dovecot-socket.service                  enabled
haveged.service                        enabled
keyboard-setup.service                  enabled
lightdm.service                         enabled
mariadb.service                        disabled
ModemManager.service                   enabled
networkd.service                       enabled
NetworkManager-dispatcher.service       enabled
NetworkManager-wait-online.service      enabled
NetworkManager.service                  enabled
nfs-common.service                      masked
regenerate-ssh-host-keys.service         enabled
rsync.service                           disabled
rtkit-daemon.service                    disabled
smartmontools.service                  enabled
sudo.service                            masked
systemd.confext.service                 disabled
systemd-fsck-root.service               enabled-runtime
systemd-network-generator.service        disabled
systemd-networkd-wait-online.service     disabled
systemd-networkd.service                disabled
systemd-pstore.service                  enabled
systemd-remount-fs.service               enabled-runtime
systemd-sysext.service                  disabled
systemd-timesyncd.service               disabled
systemd-tpm2-clear.service              enabled
virtualbox-guest-utils.service           enabled
dm-event.socket                         enabled
```

## 2. Remove unused user accounts and restrict sudo access based on least privilege

Removing unused user accounts and restricting sudo access are important security practices in Linux system administration.

Unused accounts can become security risks because attackers may exploit old or inactive accounts to gain unauthorized access. Regularly reviewing and deleting unnecessary user accounts helps reduce the attack surface and improves system security.

Sudo allows users to perform administrative tasks with root privileges. However, giving sudo access to all users is dangerous. According to the **Principle of Least Privilege**, users should only have the minimum permissions required to perform their tasks.

Restricting sudo access:

- Prevents misuse of administrative privileges
- Reduces risk of system damage
- Protects sensitive data
- Minimizes privilege escalation attacks

## 1. Identify Existing Users

cat /etc/passwd

```
(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:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon:/usr/lib/dhcpcd:/bin/false
sentrypeer:x:101:103::/var/lib/sentrypeer:/usr/sbin/nologin
tss:x:102:104:TPM software stack:/var/lib/tpm:/bin/false
strongswan:x:103:65534::/var/lib/strongswan:/usr/sbin/nologin
systemd-timesync:x:991:991:systemd Time Synchronization:/:/usr/sbin/nologin
Debian-exim:x:104:105::/var/spool/exim4:/usr/sbin/nologin
uuidd:x:105:106::/run/uuidd:/usr/sbin/nologin
messagebus:x:990:990:System Message Bus:/nonexistent:/usr/sbin/nologin
clamav:x:106:107::/var/lib/clamav:/bin/false
tcpdump:x:107:108::/nonexistent:/usr/sbin/nologin
_rpc:x:108:65534::/run/rpcbind:/usr/sbin/nologin
redis:x:109:111::/var/lib/redis:/usr/sbin/nologin
sshd:x:989:65534:sshd user:/run/sshd:/usr/sbin/nologin
dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin
avahi:x:110:114:Avahi mDNS daemon:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:111:29:Speech Dispatcher:/run/speech-dispatcher:/bin/false
```

Show normal users only:

awk -F: '\$3 >= 1000 {print \$1}' /etc/passwd

```
(kali@kali)-[~]
$ awk -F: '$3 >= 1000 {print $1}' /etc/passwd
nobody
kali
```

## 2. Check Currently Logged-in Users

Who

```
(kali㉿kali)-[~]  
$ who  
kali      seat0      2026-02-11 18:00 (:0)
```

3. Remove Unused User Account Delete user (keep home directory):  
sudo userdel username(give name)
  4. Lock an Account (instead of deleting)  
sudo passwd -l username( create the user again sample)  
Unlock:  
sudo passwd -u username
  5. View Users with Sudo Access  
getent group sudo
  6. Remove User from Sudo Group  
sudo deluser username sudo
  7. Grant Sudo Access (Only When Required)  
sudo usermod -aG sudo username
  8. Edit Sudo Permissions (Advanced – Least Privilege)  
Open sudoers file safely:  
sudo visudo
  9. Verify Sudo Rights  
Login as user: su  
username Test: sudo -l
- 
4. Disable root login and configure SSH using key-based authentication  
Disabling root login and configuring SSH with key-based authentication are important security measures in Linux systems.  
By default, the **root user** has full administrative privileges. Allowing direct root login increases the risk of brute-force attacks and unauthorized access. Disabling root login ensures that attackers cannot directly access the system as the superuser. Instead, administrators must log in as a regular user and use sudo for administrative tasks, which improves accountability and security.  
**SSH (Secure Shell)** is used for remote access to Linux systems. Password-based authentication can be vulnerable to brute-force and dictionary attacks. Key-based authentication is more secure because it uses a pair of cryptographic keys:
    - **Public key** (stored on the server)
    - **Private key** (kept securely by the user)Only users with the correct private key can access the system, making it much harder for attackers to break in.  
**Benefits:**
    - Prevents direct root attacks
    - Protects against password-based hacking



- Enhances remote access security
- Improves overall system hardening

#### STEP 1: Check SSH Status

`sudo systemctl status ssh`



```
(kali@kali)~]$ sudo systemctl status ssh
o ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:sshd(8)
           man:sshd_config(5)
```

#### STEP 2: Create SSH Key (Client Side)

Run on your local machine:

`ssh-keygen`

#### STEP 3: Copy Public Key to Server

#### STEP 4: Edit SSH Configuration Open config file:

`sudo nano /etc/ssh/sshd_config`

Find and modify these lines:

`PermitRootLogin no`

`PasswordAuthentication no`

`PubkeyAuthentication yes`

#### STEP 5: Restart SSH

`sudo systemctl restart ssh`

#### STEP 6: Test Login From terminal:

`ssh testuser@localhost`

#### STEP 7: Confirm Root Login Disabled

Try: `ssh root@localhost`

it should fail

## 5. Update system packages and enable automatic security updates.

Updating system packages and enabling automatic security updates are essential practices for maintaining a secure and stable Linux system.

Software packages may contain bugs or security vulnerabilities. Developers regularly release updates and patches to fix these issues. If the system is not updated, attackers can exploit known vulnerabilities to gain unauthorized access.

Regularly updating the system ensures:

- Security vulnerabilities are patched
- System performance and stability improve
- New features and improvements are applied
- Compatibility issues are reduced

Enabling **automatic security updates** ensures that critical patches are installed automatically without manual intervention. This reduces the risk of forgetting updates and keeps the system protected against newly discovered threats.

**Benefits:**

- Protects against known exploits
- Reduces risk of cyberattacks
- Maintains system reliability
- Saves administrative time

### STEP 1: Update Package List

`sudo apt update`

```
(kali@kali)-[~]
└─$ sudo apt update
Hit:1 http://http.kali.org/kali kali-rolling InRelease
Get:2 https://artifacts.elastic.co/packages/8.x/apt stable InRelease [3,248 B]
Err:2 https://artifacts.elastic.co/packages/8.x/apt stable InRelease
  Sub-process /usr/bin/sq returned an error code (1), error message is: Signing key on 46095ACC8548582C1A2699A9D27D666CD88E42B4 is not bound:
  evocation signature (PositiveCertification) requiring second pre-image resistance because: SHA1 is not considered secure since 2026-02-01T00:00:00Z
Fetched 3,248 B in 2s (1,390 B/s)
```

### STEP 2: Upgrade Installed Packages

`sudo apt upgrade -y`

```
(kali@kali)-[~]
└─$ sudo apt upgrade -y

The following packages were automatically installed and are no longer required:
libaudio2 libclang1-19 libmjpegutils-2.1-0t64 libpocketsphinx3 libsox3 libvdpau-va-gl1 python3-aiocache python3-humanize python3-wapiti-arsenic
libavfilter10 libfuse2t64 libmpg2encpp-2.1-0t64 libpostproc58 libspinxbase3t64 mesa-vdpau-drivers python3-aiomcache python3-loguru python3-yaswp
libavformat61 libgav1-1 libmpeg2-2.1-0t64 libsimdutf27 libswscale8 pocketsphinx-en-us python3-fs python3-marshmallow vdpau-driver-all

Use 'sudo apt autoremove' to remove them.

Upgrading:
7zip libavahi-common-data libheif1 libqt5qml5 libxfixes3 python3-l1
accountsservice libavahi-common3 libhidapi-hidraw0 libqt5qmlmodels5 libxfont2 python3-l1
acl libavahi-core7 libhivex-bin libqt5quick5 libxft2 python3-l1
adduser libavahi-glib1 libhivex0 libqt5sql5-sqlite libx6 python3-l1
afflib-tools libavc1394-0 libhtml-parser-perl libqt5sql5t64 libxinerama1 python3-l1
alsa-ucm-conf libavif16 libhttp-parser2.9 libqt5svg5 libxkbcommon-x11-0 python3-l1
amd64-microcode libavtp0 libhunspell-1.7-0 libqt5test5t64 libxkbcommon python3-l1
android-libbacktrace libayatana-ido3-0.4-0 libqt5waylandclient5 libxkbfile1 python3-l1
android-libbase libb2-1 libhwloc15 libqt5waylandcompositor5 libxkbregistry0 python3-l1
android-libcutils libbabeltrace1 libhwloc16 libqt5widgets5t64 libxklavier16 python3-l1
android-liblog libbcg729-0 libhyphen0 libqt5xml5t64 libxml++2.6-2v5 python3-l1
android-libutils libbde1t64 libice-dev libqt5xmlpatterns5 libxml2-16 python3-l1
```

## STEP 4: Install unattended-upgrades

```
sudo apt install unattended-upgrades -y
```

```

kali ( BEFORE ANALYZE SNAPSHOT) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

[+] 1 2 3 4

kali@kali:~$ sudo apt install unattended-upgrades -y

[sudo] password for kali:
The following packages were automatically installed and are no longer required:
  amass-connex libgeoip3.14.0 libmongoc-1.0-0 libnet4 libtheoraenc1 python3-bluepy python3-kismetcapturetrtl433 python3-lutils samba-ad-provision
  libburl2 libinstpatch-1.0-2 libmongocrypt0 libportaudio0 libusb4 libusb4-164 python3-click-plugins python3-kismetcapturetrtladsb python3-rlwt samba-dsdb-modules
  libjs-jquery-ui libnet1 libnet16 libnet16-164 python3-kismetcapturetraigear python3-kismetcapturefrealksbsigbee python3-protobuf samba-dsdb-imp
  libnet2 libjs-underscore libobjc3-14-dev libtheoradec1 libxml2 python3-kismetcapturefrealksbsigbee python3-protobuf samba-ad-dc

Use 'sudo apt autoremove' to remove them.

Installing:
  unattended-upgrades

Installing dependencies:
  python3-distutils-info

Suggested packages:
  Dsdb-mailx default-mta | mail-transport-agent needrestart

Summary:
  Upgrading: 0, Installing: 2, Removing: 0, Not Upgrading: 120
  Download size: 74.8 kB
  Space needed: 368 kB / 37.1 GB available

Get:1 http://mirrors.ustc.edu.cn/kali kali-rolling/main amd64 python3-distutils-info all 1.14 [7,848 B]
Get:2 http://mirrors.ustc.edu.cn/kali kali-rolling/main amd64 unattended-upgrades all 2.12 [66.9 kB]

```

## STEP 5: Enable Automatic Updates

```
sudo dpkg-reconfigure --priority=low unattended-upgrades
```

```
(kali@kali)-[~]$ sudo dpkg-reconfigure --priority=low unattended-upgrades
```

## STEP 6: Verify Status

```
systemctl status unattended-upgrades
```

```
(kali㉿kali)-[~]
$ systemctl status unattended-upgrades

● unattended-upgrades.service - Unattended Upgrades Shutdown
   Loaded: loaded (/usr/lib/systemd/system/unattended-upgrades.service; enabled; preset: disabled)
   Active: active (running) since Wed 2026-02-11 18:44:32 IST; 3min 14s ago
 Invocation: d5f1b3680c1744c99cf3410fa59202bb
    Docs: man:unattended-upgrade(8)
   Main PID: 2227 (unattended-upgr)
      Tasks: 2 (limit: 2116)
```

6. Configure a firewall to allow only required network traffic  
Configure a firewall to allow only required network traffic by following the **principle of least privilege** — deny everything by default, then explicitly allow only what is necessary.

## Define Your Required Traffic

Before configuring anything, identify:

- Which **services** must be accessible (e.g., web, SSH, database)
- Which **ports** they use
- Which **IP addresses or networks** should have access
- Whether access is **inbound, outbound, or both**

Example:

- Web server → Allow TCP 80 (HTTP) and 443 (HTTPS)
- Admin access → Allow TCP 22 (SSH) from admin IP only
- Database → Allow TCP 3306 only from application server

## Linux Firewall (UFW – Ubuntu/Debian)

### 1 Enable firewall

```
sudo ufw enable
```

### 2 Set default deny policy

```
sudo ufw default deny incoming
```

```
sudo ufw default allow outgoing
```

### 3 Allow required services

```
sudo ufw allow 80/tcp
```

```
sudo ufw allow 443/tcp
```

```
sudo ufw allow from 192.168.1.10 to any port 22
```

### 4 Check status

```
sudo ufw status verbose
```

## STEP 2: Check Firewall Status

```
sudo ufw status
```

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

## STEP 3: Set Default Policies (Deny Everything)

```
sudo ufw default deny incoming
```

```
sudo ufw default allow outgoing
```

```
(kali㉿kali)-[~]  
$ sudo ufw default deny incoming  
sudo ufw default allow outgoing  
Default incoming policy changed to 'deny'  
(be sure to update your rules accordingly)  
Default outgoing policy changed to 'allow'  
(be sure to update your rules accordingly)
```

#### STEP 4: Allow Required Services

Only Allow SSH:

```
sudo ufw allow ssh
```

#### STEP 5: Enable Firewall

```
sudo ufw enable
```

#### STEP 6: Verify Rules `sudo ufw status verbose`

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

#### STEP 6: Verify Rules

```
sudo ufw status verbose
```

7 Stop and disable unnecessary services running on the server.  
Attack Surface Reduction

Each service:

- Opens network ports
- Accepts input
- Has executable code

More services = more potential entry points.

Fewer services = fewer opportunities for exploitation.

#### **Principle of Least Functionality**

A core security best practice:

A system should provide only essential capabilities.

This minimizes:

- Exploitable software
- Misconfiguration risks
- Privilege escalation paths

#### **Vulnerability Containment**

Even trusted services can have:

- Zero-day vulnerabilities



- Misconfigurations
- Weak authentication

If the service isn't running, it cannot be exploited.

## Performance & Stability

Unnecessary services:

- Consume CPU and memory
- Increase boot time
- Create log noise
- Increase maintenance overhead

Minimal systems are:

- Easier to monitor
- Easier to audit
- Easier to secure

STEP 1: List Running Services

`systemctl list-units --type=service --state=running`

```
(kali㉿kali)-[~]
└─$ systemctl list-units --type=service --state=running
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
accounts-daemon.service	loaded	active	running	Accounts Service
colord.service	loaded	active	running	Manage, Install and Generate Color Profiles
cron.service	loaded	active	running	Regular background program processing daemon
dbus.service	loaded	active	running	D-Bus System Message Bus
getty@tty1.service	loaded	active	running	Getty on tty1
haveged.service	loaded	active	running	Entropy Daemon based on the HAVEGE algorithm
lightdm.service	loaded	active	running	Light Display Manager
ModemManager.service	loaded	active	running	Modem Manager
NetworkManager.service	loaded	active	running	Network Manager
polkit.service	loaded	active	running	Authorization Manager
rtkit-daemon.service	loaded	active	running	RealtimeKit Scheduling Policy Service
systemd-journald.service	loaded	active	running	Journal Service
systemd-logind.service	loaded	active	running	User Login Management
systemd-udevd.service	loaded	active	running	Rule-based Manager for Device Events and Files
tor@default.service	loaded	active	running	Anonymizing overlay network for TCP
udisks2.service	loaded	active	running	Disk Manager
unattended-upgrades.service	loaded	active	running	Unattended Upgrades Shutdown
upower.service	loaded	active	running	Daemon for power management
user@1000.service	loaded	active	running	User Manager for UID 1000
virtualbox-guest-utils.service	loaded	active	running	Virtualbox guest utils

STEP 2: List Enabled Services (Start at Boot)

`systemctl list-unit-files --type=service | grep enabled`

STEP 3: Identify Unnecessary Services (Examples) Common services you may disable (only if not needed):

```
(kali@kali)-[~]
$ systemctl list-unit-files --type=service | grep enabled
accounts-daemon.service          enabled enabled
console-setup.service           enabled enabled
cron.service                    enabled enabled
getty@tty1.service              enabled enabled
haveged.service                 enabled enabled
keyboard-setup.service          enabled enabled
lightdm.service                 enabled disabled
ModemManager.service            enabled enabled
networking.service              enabled enabled
NetworkManager-dispatcher.service enabled disabled
NetworkManager-wait-online.service enabled disabled
NetworkManager.service         enabled enabled
nfs-common.service              masked enabled
regenerate-ssh-host-keys.service enabled enabled
rsync.service                   disabled enabled
rtkit-daemon.service            enabled enabled
smartmontools.service           enabled enabled
sudo.service                    masked enabled
systemd-confext.service         disabled enabled
systemd-fsck-root.service       enabled-runtime disabled
systemd-network-generator.service disabled enabled
systemd-networkd-wait-online.service disabled enabled
systemd-networkd.service       disabled enabled
systemd-pstore.service          enabled enabled
systemd-remount-fs.service      enabled-runtime disabled
systemd-sysext.service          disabled enabled
systemd-timesyncd.service       enabled enabled
systemd-tpm2-clear.service      disabled enabled
tor.service                     enabled disabled
tor@default.service             enabled-runtime disabled
ufw.service                     enabled enabled
```

- Bluetooth
- cups (printing)
- apache2
- avahi-daemon

STEP 4: Stop a Service Example: stop Bluetooth

sudo systemctl stop Bluetooth

STEP 5: Disable Service from Startup

sudo systemctl disable Bluetooth

STEP 6: Verify Service is Disabled

systemctl status Bluetooth

STEP 8: Confirm Reduced Services

systemctl list-units --type=service --state=running

```
(kali@kali)-[~]
$ systemctl list-units --type=service --state=running
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
accounts-daemon.service            loaded active running Accounts Service
colord.service                     loaded active running Manage, Install and Generate Color Profiles
cron.service                       loaded active running Regular background program processing daemon
dbus.service                       loaded active running D-Bus System Message Bus
getty@tty1.service                 loaded active running Getty on tty1
haveged.service                    loaded active running Entropy Daemon based on the HAVEGE algorithm
lightdm.service                    loaded active running Light Display Manager
ModemManager.service               loaded active running Modem Manager
NetworkManager.service             loaded active running Network Manager
polkit.service                     loaded active running Authorization Manager
rtkit-daemon.service               loaded active running RealtimeKit Scheduling Policy Service
systemd-journald.service            loaded active running Journal Service
systemd-logind.service              loaded active running User Login Management
systemd-udevd.service              loaded active running Rule-based Manager for Device Events and Files
tor@default.service                loaded active running Anonymizing overlay network for TCP
udisks2.service                    loaded active running Disk Manager
unattended-upgrades.service         loaded active running Unattended Upgrades Shutdown
upower.service                     loaded active running Daemon for power management
user@1000.service                  loaded active running User Manager for UID 1000
virtualbox-guest-utils.service      loaded active running Virtualbox guest utils
```

## 7. Secure file permissions for sensitive system and configuration files

### The Core Principle

Every file has:

- **Owner**
- **Group**
- **Permissions (read, write, execute)**

Security goal:

Only the minimum necessary users/processes can access sensitive files.

### **Why This Matters**

Sensitive files often contain:

- Password hashes
- API keys
- Private keys
- Database credentials
- System configurations
- Security policies

If improperly permissioned:

- Attackers can escalate privileges
- Malware can modify system behavior
- Credentials can be stolen

### **Apply the Principle of Least Privilege**

- Users access only what they need
- Services run under dedicated low-privilege accounts
- No world-readable sensitive files

### **Protect Critical System Files**

Examples (Linux):

- /etc/passwd
- /etc/shadow
- /etc/sudoers
- SSH private keys
- Web server configs
- Application .env files

STEP 1: Check Current Permissions ls

-l /etc/passwd /etc/shadow /etc/sudoers

```
(kali㉿kali)-[~]
$ ls -l /etc/passwd /etc/shadow /etc/sudoers

-rw-r--r-- 1 root root 3328 Nov 18 19:08 /etc/passwd
-rw-r----- 1 root shadow 1470 Nov 18 19:08 /etc/shadow
-r--r----- 1 root root 1714 Aug 17 16:11 /etc/sudoers
```

STEP 2: Secure /etc/shadow (Most Sensitive)

sudo chown root:shadow /etc/shadow

sudo chmod 640 /etc/shadow

```
(kali㉿kali)-[~]
$ sudo chown root:shadow /etc/shadow
sudo chmod 640 /etc/shadow
```

Verify: ls -l /etc/shadow

```
(kali㉿kali)-[~]
$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1470 Nov 18 19:08 /etc/shadow
```

STEP 3: Secure /etc/passwd

sudo chmod 644 /etc/passwd

```
(kali㉿kali)-[~]
$ sudo chmod 644 /etc/passwd
```

STEP 4: Secure sudoers File

sudo chmod 440 /etc/sudoers

```
(kali㉿kali)-[~]
$ sudo chmod 440 /etc/sudoers
```

Verify: ls -l /etc/sudoers

```
(kali㉿kali)-[~]
$ ls -l /etc/sudoers
-r--r----- 1 root root 1714 Aug 17 16:11 /etc/sudoers
```

STEP 5: Secure SSH Configuration

sudo chmod 600 /etc/ssh/sshd\_config

sudo chown root:root /etc/ssh/sshd\_config

```
(kali㉿kali)-[~]
$ sudo chmod 600 /etc/ssh/sshd_config
sudo chown root:root /etc/ssh/sshd_config
```

STEP 6: Secure User SSH Keys For testuser:

```
chmod 700 /home/testuser/.ssh 22
```

```
chmod 600 /home/testuser/.ssh/authorized_keys
```

```
chown -R testuser:testuser /home/testuser/.ssh
```

```
(kali㉿kali)-[~]  
$ chmod 700 /home/testuser/.ssh  
$ chmod 600 /home/testuser/.ssh/authorized_keys  
$ chown -R testuser:testuser /home/testuser/.ssh
```

STEP 7: Find World-Writable Files (Audit)

```
sudo find / -type f -perm -0002 2>/dev/null
```

```
(kali㉿kali)-[~]  
$ sudo find / -type f -perm -0002 2>/dev/null  
  
/sys/kernel/security/apparmor/.remove  
/sys/kernel/security/apparmor/.replace  
/sys/kernel/security/apparmor/.load  
/sys/kernel/security/apparmor/.access  
/sys/kernel/security/tomoyo/self_domain  
/proc/sys/kernel/ns_last_pid  
/proc/pressure/io  
/proc/pressure/cpu  
/proc/pressure/memory  
/proc/1/task/1/attr/current  
/proc/1/task/1/attr/exec  
/proc/1/task/1/attr/fscreate  
/proc/1/task/1/attr/keycreate  
/proc/1/task/1/attr/sockcreate  
/proc/1/task/1/attr/apparmor/current  
/proc/1/task/1/attr/apparmor/exec  
/proc/1/attr/current  
/proc/1/attr/exec  
/proc/1/attr/fscreate  
/proc/1/attr/keycreate  
/proc/1/attr/sockcreate  
/proc/1/attr/apparmor/current  
/proc/1/attr/apparmor/exec  
/proc/1/timerslack_ns  
/proc/2/task/2/attr/current  
/proc/2/task/2/attr/exec  
/proc/2/task/2/attr/fscreate
```

## 8. Review system logs to monitor authentication and system activity

Log review supports three security goals:

1. **Detection** – Identify unauthorized access or abnormal behavior
2. **Accountability** – Trace actions to users or processes
3. **Forensics** – Investigate incidents after they occur

Without log monitoring, breaches can remain undetected for months.

### What Logs Contain

System logs record:

- Successful and failed logins
- Privilege escalations (e.g., sudo use)
- Service start/stop events
- Configuration changes
- File access events
- Network connections



- System errors and crashes  
Logs are the system's **security diary**.  
**Authentication Monitoring (High Priority)**

Key events to monitor:

**Failed Login Attempts**

- Brute-force attacks
- Credential stuffing
- Enumeration attempts

**Successful Logins at Unusual Times**

- Late-night access
- Access outside business hours

**Logins from Unusual Locations or IPs**

- Foreign IP addresses
- New geographic regions

**Privilege Escalation**

- Use of administrative accounts
- sudo activity
- Role or group changes

**System Activity Monitoring**

Look for:

**Unexpected Service Changes**

- Services stopping unexpectedly
- New services starting
- Disabled security tools

**File Changes**

- Modifications to system configuration files
- Changes to authentication files
- Altered binaries

**New User Accounts**

- Unexpected account creation
- Admin privilege assignment

STEP 1 View all authentication logs

```
sudo journalctl -u ssh
```

STEP 1 View all authentication logs

```
sudo journalctl -u ssh
```

View

```
sudo activity sudo journalctl | grep
```

```
sudo View login history
```

```
last
```

View system logs

```
sudo journalctl  
Recent boot only:  
sudo journalctl -b
```

## 9. Conclusion:-

Comprehensive Linux system hardening was carried out to enhance security and minimize potential attack surfaces. Default users, running services, and open ports were carefully examined to assess overall system exposure. Unnecessary user accounts were removed, and sudo privileges were limited in accordance with the principle of least privilege. SSH was secured by disabling root login and enforcing key-based authentication.

System packages were kept up to date, and automatic security updates were enabled to address known vulnerabilities promptly. A firewall was configured to permit only essential network traffic, while unnecessary services were stopped and disabled to further reduce risk. Sensitive system and configuration files were protected with appropriate file permissions, and system logs were regularly reviewed to monitor authentication events and overall system activity.

Collectively, these actions strengthened the server's security posture, enforced controlled access, improved monitoring and visibility, and established a secure baseline configuration for Linux systems.