Task 14: Linux Server Hardening & Secure Configuration

🔹 1️⃣ What is Server Hardening?

Server hardening is the process of reducing a system’s attack surface by:

- Removing unnecessary services
- Securing configurations
- Applying patches
- Enforcing access control
- Monitoring logs

Goal: **Make the system resistant to attacks.**

🔹 2️⃣ Step-by-Step Linux Hardening (Practical Commands)

Assume Ubuntu Server (similar steps for Kali).

🔎 Step 1: Review Current System State

Check Users:

```
cat /etc/passwd
```

Check sudo users:

```
getent group sudo
```

Check open ports:

```
ss -tulnp
```

or

```
netstat -tulnp
```

Check running services:

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

👤 Step 2: Remove Unused Users & Restrict Sudo

Remove unused user:

```
sudo deluser username
```

Remove sudo access:

```
sudo deluser username sudo
```

✔ Principle: **Least Privilege**

Only necessary users should have admin rights.

🔐 Step 3: Secure SSH Configuration

Edit SSH config:

```
sudo nano /etc/ssh/sshd_config
```

Change:

```
PermitRootLogin no

PasswordAuthentication no
```

Enable key-based authentication.

Restart SSH:

```
sudo systemctl restart ssh
```

✔ Disables brute-force attacks on root.

🔄 Step 4: Update System Packages

```
sudo apt update && sudo apt upgrade -y
```

Enable automatic updates:

```
sudo apt install unattended-upgrades

sudo dpkg-reconfigure unattended-upgrades
```

✔ Prevents exploitation of known vulnerabilities.

🔥 Step 5: Configure Firewall (UFW)

Enable firewall:

```
sudo ufw enable
```

Allow only required ports:

```
sudo ufw allow 22/tcp
```

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

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

Check status:

```
sudo ufw status verbose
```

✓ Blocks unnecessary incoming traffic.

🚫 Step 6: Disable Unnecessary Services

Example:

```
sudo systemctl stop apache2
```

```
sudo systemctl disable apache2
```

Check services:

```
systemctl list-unit-files --type=service
```

✓ Reduces attack surface.

📂 Step 7: Secure File Permissions

Check sensitive files:

```
ls -l /etc/shadow
```

Correct permissions:

```
sudo chmod 600 /etc/shadow
```

Secure SSH directory:

```
chmod 700 ~/.ssh
```

```
chmod 600 ~/.ssh/authorized_keys
```

✓ Prevents privilege escalation.

📜 Step 8: Review Logs

Authentication logs:

```
sudo cat /var/log/auth.log
```

System logs:

```
sudo journalctl -xe
```

Look for:

- Failed login attempts
- Unknown IP addresses
- Repeated errors

✓ Early detection of attacks.

📋 Linux Hardening Checklist (Submission Section)

Task	Status
Removed unused users	✅
Restricted sudo access	✅
Disabled root SSH login	✅
Enabled key-based authentication	✅
Updated system packages	✅
Enabled automatic updates	✅
Configured firewall (UFW)	✅

Task	Status
Disabled unused services	✅
Secured file permissions	✅
Reviewed authentication logs	✅

Security Configuration Summary

The Linux server was hardened using the following measures:

1. Disabled root login via SSH.
2. Enforced SSH key-based authentication.
3. Removed unnecessary user accounts and limited sudo access.
4. Configured firewall rules to allow only essential ports.
5. Disabled unused services to reduce attack surface.
6. Applied system updates and enabled automatic patching.
7. Secured sensitive file permissions.
8. Reviewed logs to monitor suspicious activity.

These steps significantly reduce risks from:

- Brute-force attacks
- Privilege escalation
- Unauthorized access
- Service exploitation
- Remote attacks