Task 14: Linux Server Hardening & Secure Configuration
Tools: Primary: Ubuntu / Kali Linux
Alternatives: Lynis, CIS Benchmarks

Hints / Mini Guide:
1.Review default Linux system settings to understand users, services, and open ports.
2.Remove unused user accounts and restrict sudo access based on least privilege.
3.Disable root login and configure SSH using key-based authentication.
4.Update system packages and enable automatic security updates.
5.Configure a firewall to allow only required network traffic.
6.Stop and disable unnecessary services running on the server.
7.Secure file permissions for sensitive system and configuration files.
8.Review system logs to monitor authentication and system activity.

1.Review default Linux system settings

Reviewing default Linux settings helps identify existing users, active services, and open ports that may pose security risks. Administrators examine /etc/passwd, running services, and network ports using tools like netstat or ss. This process provides visibility into the system's exposure and potential vulnerabilities. Understanding the default configuration is the first step in hardening, as it establishes a baseline for removing unnecessary components and securing essential services.

2. Remove unused users & restrict sudo access

Unused user accounts increase security risks because attackers can exploit dormant credentials. Removing unnecessary accounts reduces the attack surface. Sudo access should follow the principle of least privilege, granting administrative rights only to users who require them. Editing the sudoers file ensures controlled privilege escalation. This prevents unauthorized system changes, reduces insider threats, and limits the impact if a user account becomes compromised.

3. Disable root login & configure SSH keys

Disabling direct root login prevents attackers from targeting the most powerful account on the system. Instead, users log in with personal accounts and escalate privileges using sudo. Configuring SSH with key-based authentication replaces passwords with cryptographic keys, significantly reducing brute-force and credential theft attacks. This strengthens remote access security and ensures only authorized devices can connect to the server securely.

4. Update system & enable automatic updates

Keeping the system updated ensures security patches and bug fixes are applied promptly. Outdated software often contains vulnerabilities that attackers exploit. Running regular updates and enabling automatic security updates ensures continuous protection without manual intervention. This reduces exposure to known threats and maintains system stability, integrity, and compliance with security best practices.

5. Configure firewall

A firewall controls incoming and outgoing network traffic based on predefined rules. By default, all unnecessary ports should be blocked, allowing only required services such as SSH or HTTP. Tools like

UFW simplify firewall management in Linux. Proper firewall configuration minimizes exposure to unauthorized access, prevents exploitation of open ports, and acts as the first line of defense against network-based attacks.

6. Stop unnecessary services

Unnecessary services increase the attack surface because each running service may contain vulnerabilities. Identifying and disabling unused services reduces potential entry points for attackers. Using systemctl, administrators can stop and disable services that are not essential for server functionality. This improves performance, strengthens security, and ensures the server runs only required and monitored applications.

7. Secure file permissions

Proper file permissions protect sensitive files such as /etc/shadow, SSH configuration files, and system logs. Limiting read, write, and execute permissions ensures only authorized users can access critical data. Using chmod and chown helps enforce strict ownership and access control. Securing file permissions prevents unauthorized modifications, data leakage, and privilege escalation attacks.

8. Review system logs

System logs provide detailed records of authentication attempts, system activity, and service events. Regularly reviewing logs helps detect suspicious behavior such as failed login attempts or unauthorized access. Files like /var/log/auth.log and /var/log/syslog are crucial for monitoring. Log analysis supports incident detection, forensic investigation, and proactive threat management to maintain server security.