**Linux Server Hardening and Security Analysis**

Project Title: Linux Server Hardening and Security Analysis

Prepared by: Sagar Chauhan

Date: 26-04-2025

**Table of Contents**

1. **Objective**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2. **Tools & Environment Used**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
3. **System Architecture / Setup Diagram**. . . . . . . . . . . . . . . . . . . . . 7
4. **Baseline Security Audit (Before Hardening)**
   * 4.1 Nmap Scan Results. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   * 4.2 Lynis Audit Report. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
   * 4.3 Open Ports and Services. . . . . . . . . . . . . . . . . . . . . . . . . 11
5. **Server Hardening Implementation**
   * 5.1 Disabling Unnecessary Services. . . . . . . . . . . . . . . . . . . . . . 12
   * 5.2 SSH Hardening. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
   * 5.3 Firewall Configuration (UFW) . . . . . . . . . . . . . . . . . . . 14
   * 5.4 Fail2Ban Setup. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
   * 5.5 Automatic Security Updates. . . . . . . . . . . . . . . . . . . . . 16
   * 5.6 File Permissions and System Cleanup. . . . . . . . . . . . . . . . . 17
6. **Post-Hardening Audit (After Changes)**
   * 6.1 Nmap Re-scan. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
   * 6.2 Lynis Re-audit. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
7. **Before & After Comparison**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
8. **Conclusion**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

**Abstract**

In the modern digital landscape, server security is a critical component of system administration. This project focuses on hardening a Linux-based server (Ubuntu) by applying essential security measures and performing a detailed security audit. Using Kali Linux as an external audit system, we conducted a baseline analysis to identify vulnerabilities and weak configurations in the target server. Tools such as Nmap and Lynis were utilized to assess the system's initial security posture. Following this, various hardening techniques were implemented, including firewall configuration, SSH hardening, service minimization, brute-force protection with Fail2Ban, and enabling automatic updates. A follow-up audit was conducted to verify improvements. This project not only highlights the importance of proactive server security but also demonstrates practical skills in identifying and mitigating risks in a live server environment.

1. **Objective**

The primary objective of this project is to secure a Linux-based server environment by implementing standard server hardening techniques and conducting a comprehensive security audit. The project aims to identify existing vulnerabilities in an Ubuntu server setup, mitigate potential risks, and ensure that the system adheres to recommended security practices. This involves analyzing the server using tools such as Nmap and Lynis, disabling unnecessary services, configuring a firewall, securing remote access, and applying intrusion prevention mechanisms. By the end of the project, the security posture of the server should be measurably improved, demonstrating the effectiveness of the applied hardening strategies.

**2. Tools & Environment Used**

* **Ubuntu 20.04 LTS**  
  Used as the main Linux server for implementing hardening techniques and testing security settings.
* **Kali Linux**  
  Acts as the attacker or auditor machine to perform scanning and vulnerability assessment on the Ubuntu server.
* **Nmap**  
  A powerful network scanner used to detect open ports, running services, and OS fingerprinting.
* **Lynis**  
  A security auditing tool for Unix systems that provides detailed reports and suggestions for hardening.
* **OpenSSH**  
  Provides secure remote access to the Ubuntu server. Also configured for enhanced security (e.g., disable root login).
* **UFW (Uncomplicated Firewall)**  
  Simple firewall tool used to control incoming and outgoing network traffic based on predefined rules.
* **Fail2Ban**  
  Helps protect the server from brute-force login attacks by banning IPs that show malicious signs.
* **chkrootkit / rkhunter**  
  Command-line tools used to check the server for the presence of rootkits and hidden malware.
* **Unattended-Upgrades**  
  A package used to automatically install security updates to keep the server patched and up to date.
* **Netstat / ss**  
  Tools used to view currently open ports and listening services on the server.
* **Tee**  
  A Linux utility used to save the output of audit tools like Lynis into text files for documentation purposes.

**3. System Architecture / Setup Diagram**

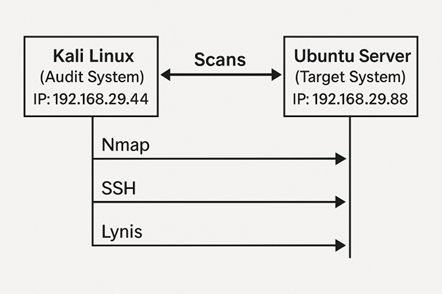
**Written Description:**

The setup consists of two machines connected on the same local network:

1. **Ubuntu Server (Target System):**
   * Acts as the main system to be hardened
   * IP Address: e.g., 192.168.29.88
   * Services like SSH and others are initially running
2. **Kali Linux (Audit System):**
   * Used for external scanning and auditing
   * IP Address: e.g., 192.168.29.44
   * Performs Nmap scans, SSH tests, and remote auditing

Both systems communicate over the same LAN using secure protocols and auditing tools. After hardening, the Ubuntu system is re-scanned to compare results.

**Diagram Description (for your report):**



**4.Baseline Security Audit (Before Hardening)**

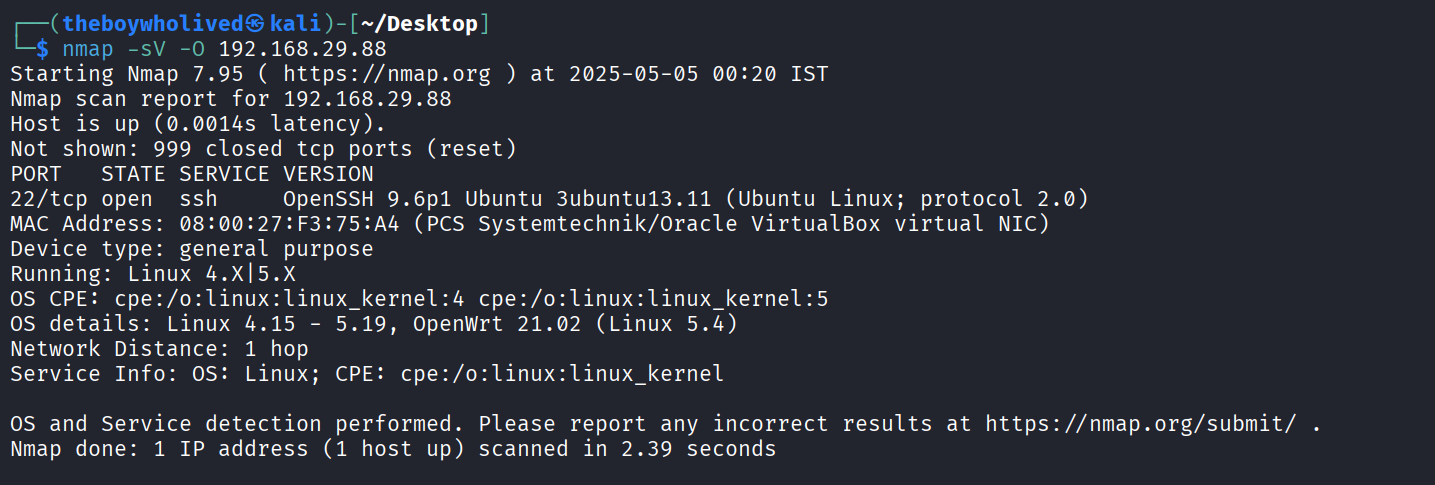
Before applying any hardening techniques, the Ubuntu server was analyzed using standard tools to identify existing vulnerabilities and insecure configurations.

**4.1 Nmap Scan Results**

A basic Nmap scan was performed from the Kali Linux machine to detect open ports and running services on the Ubuntu server:

**Command used:** nmap -sV -O 192.168.29.88

**Scan Type:** Service version detection (-sV) and OS detection (-O)



**Output Summary:**

**Open Port:**

* 22/tcp – **OpenSSH 9.6p1 Ubuntu** (used for secure remote login)

**OS Detection:**

* Likely OS: **Linux Kernel 4.15 – 5.19**
* Possible match: **OpenWRT 21.02**

**4.2 Lynis Audit Report**

**Command:**

**sudo lynis audit system | tee lynis-before.txt**

The system was audited using Lynis, a security auditing tool designed for Unix-based systems. The following sections summarize the key findings and recommendations from the audit.

**1. Hardening Index:**

The Hardening Index provides an overall score indicating the system’s security level. Although Lynis does not directly show the Hardening Index as a score, it provides detailed assessments of the system’s configuration and identifies areas of improvement. The report highlighted several areas where security hardening measures are required, including:

* Kernel Hardening
* File Permissions
* Specific Configuration Settings

These areas need to be addressed to improve the system's security.

**2. Warnings:**

The audit report identified several warnings that require immediate attention to enhance security. These include:

* File Permissions:  
  Certain files and directories, such as /etc/crontab and /etc/ssh/sshd\_config, need modified permissions to secure them properly.
* CUPS Daemon:  
  The CUPS daemon is running, but its configuration file's permissions are weak, making it vulnerable to unauthorized access and misuse.
* Apt-check Warning:  
  The apt-check command failed to return a result, which might indicate an issue with the package management system.
* Insecure Banner Information:  
  The /etc/issue and /etc/issue.net files contain weak banner information that could give attackers unnecessary details about the system, such as its OS version.
* Kernel Parameters:  
  Several kernel parameters deviated from their expected values, posing a potential security risk.

**3. Suggestions:**

Lynis suggested several improvements to enhance system security:

* File Permissions:  
  The permissions of critical files like /etc/crontab, /etc/ssh/sshd\_config, and directories related to cron jobs need to be tightened to avoid unauthorized access.
* SSH Configurations:  
  It was recommended to secure SSH configurations by adjusting settings such as MaxAuthTries, AllowTcpForwarding, Port, and LogLevel for improved security.
* Kernel Hardening:  
  Suggested kernel hardening configurations include securing settings like kernel.modules\_disabled, kernel.sysrq, and fs.protected\_fifos to prevent potential exploitation.
* Network Settings:  
  To reduce the risk of network attacks, it is advised to disable net.ipv4.conf.all.accept\_redirects and net.ipv6.conf.all.accept\_redirects.
* Services Configuration:  
  Unnecessary services, such as CUPS and Telnet, should be disabled to minimize the attack surface.

**4. Plugins:**

* Lynis Plugins:  
  The report did not show output for Phase 2 plugins, which could indicate either the plugins were not executed or there was an issue with the plugin configuration.
* Modules:  
  While specific modules were not summarized in the output, general warnings and suggestions were provided. These cover critical areas like file permissions, SSH hardening, kernel settings, and network configurations.

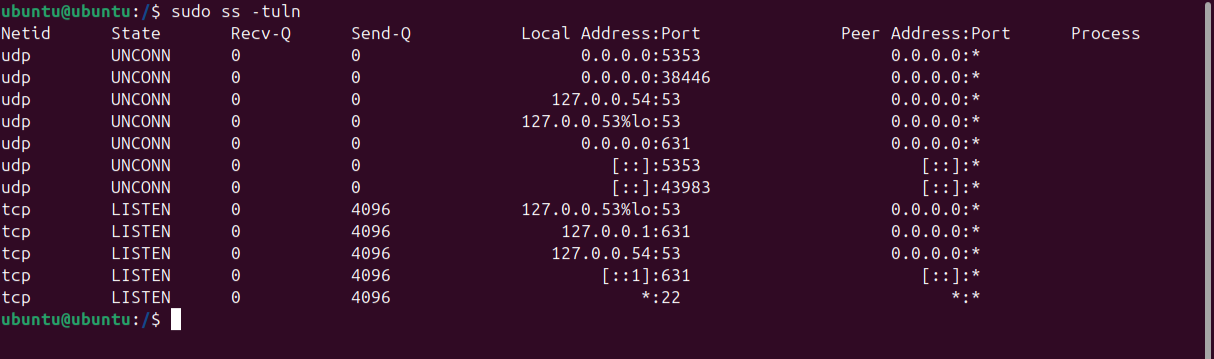
**4.3 Open Ports and Services**

**Internal Service Check – ss/netstat**

A review of active services and open ports was performed using the ss and netstat tools.

**Command:** sudo ss -tuln

A local check was done using the ss command to view listening ports:



**Services Running:**

**Analysis:**

* SSH (Port 22) is open to all interfaces (\*), which means it's exposed to the entire network — needs hardening.
* CUPS (Port 631) is listening only on localhost, so it's not externally reachable — still, disabling it is recommended if not needed.
* DNS and mDNS (Port 53/5353) are locally bound — likely part of systemd-resolved or Avahi services.

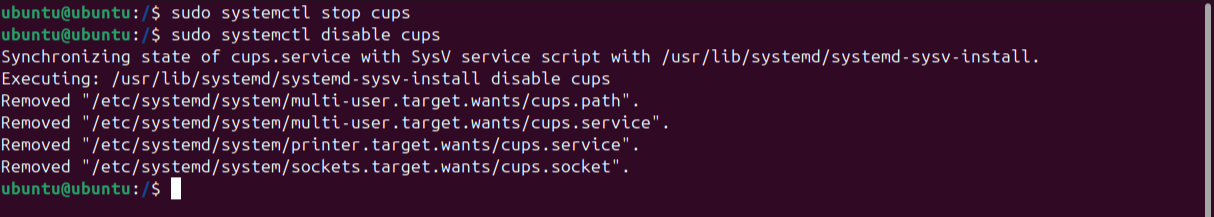
**5. Server Hardening Implementation**

**5.1 Disabling Unnecessary Services**

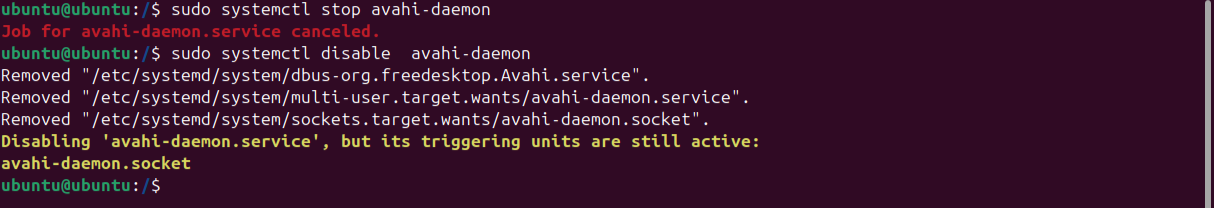
Unnecessary services were identified and disabled to reduce the attack surface.

**Actions Taken:**

* **Disabled CUPS (printing service):**



**Disabled Avahi (mDNS/zeroconf service) if running:**



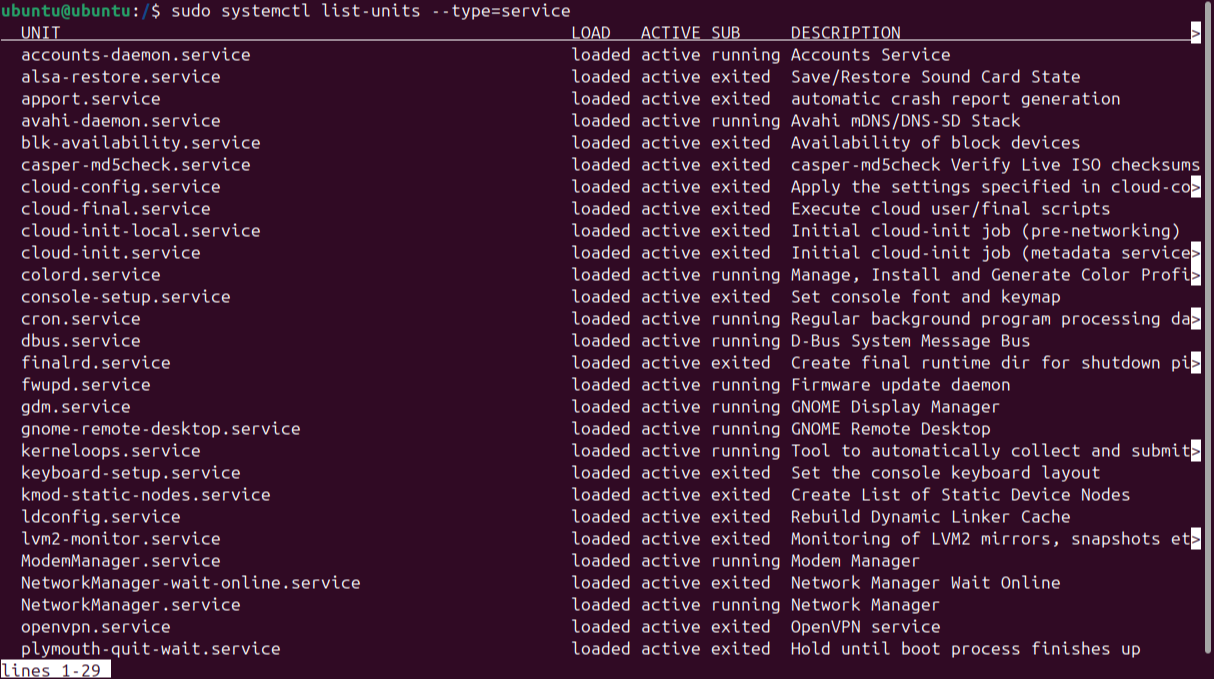
**Output:**

* The service was stopped and disabled.
* However, the system indicated that triggering units (sockets) are still active:  
  **avahi-daemon.socket** remains enabled. This should also be disabled to fully stop the service.

**Command:** sudo systemctl disable avahi-daemon.socket

**Checked Active Services**

**Used the following command to list all active systemd services:**



**Key Observations:**

* Several services are active that may not be necessary depending on the server's purpose:
  + ModemManager.service – Often not needed on servers.
  + gnome-remote-desktop.service – Needed only if GUI and remote desktop access is used.
  + fwupd.service – Used for firmware updates, not critical for minimal servers.
* Services like cron, NetworkManager, and ssh are typically necessary.

**5.2 SSH Hardening**

The default SSH configuration was modified to prevent brute-force attacks and unauthorized access.

**Actions Taken:**

* Edited /etc/ssh/sshd\_config:



**Add the following lines at the end:**

Port 2222

PermitRootLogin no

PasswordAuthentication no

MaxAuthTries 3

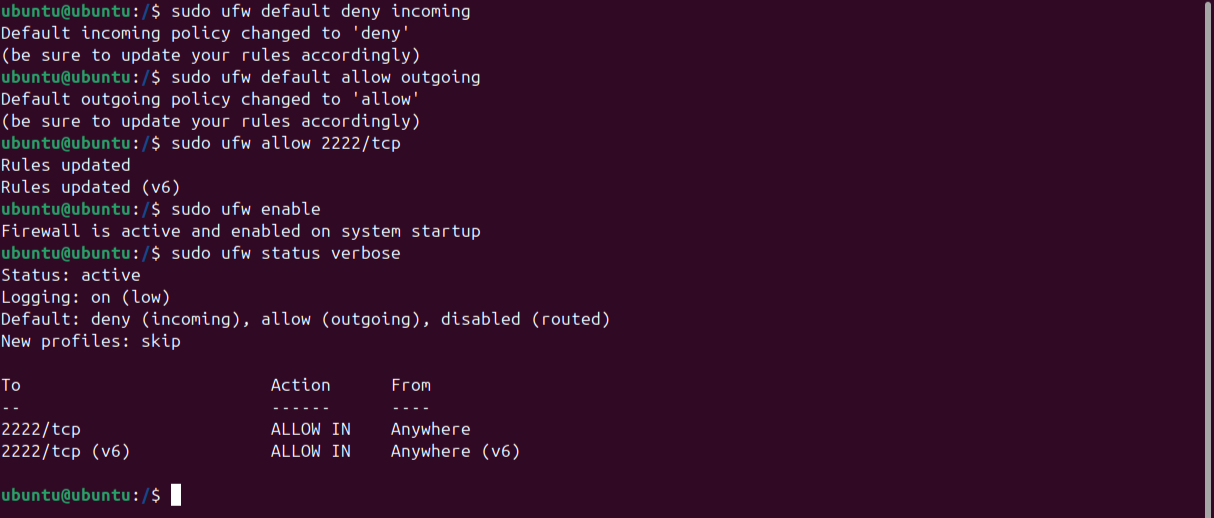
AllowUsers <your-username>

**Restarted SSH service:**



**5.3 Firewall Configuration (UFW)**

UFW (Uncomplicated Firewall) was configured to allow only essential traffic.



**Commands Used:**

sudo ufw default deny incoming

sudo ufw default allow outgoing

sudo ufw allow 2222/tcp # Allow SSH on new port

sudo ufw enable

sudo ufw status verbose

**5.4 Fail2Ban Setup**

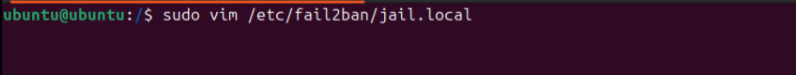
**Fail2Ban** was installed and configured to protect against brute-force login attempts.

**Installation command:**

sudo apt install fail2ban

**Basic Configuration (optional):**

* File: /etc/fail2ban/jail.local



**Command:** sudo vim /etc/fail2ban/jail.local

**ADD THIS LINE:**

[sshd]

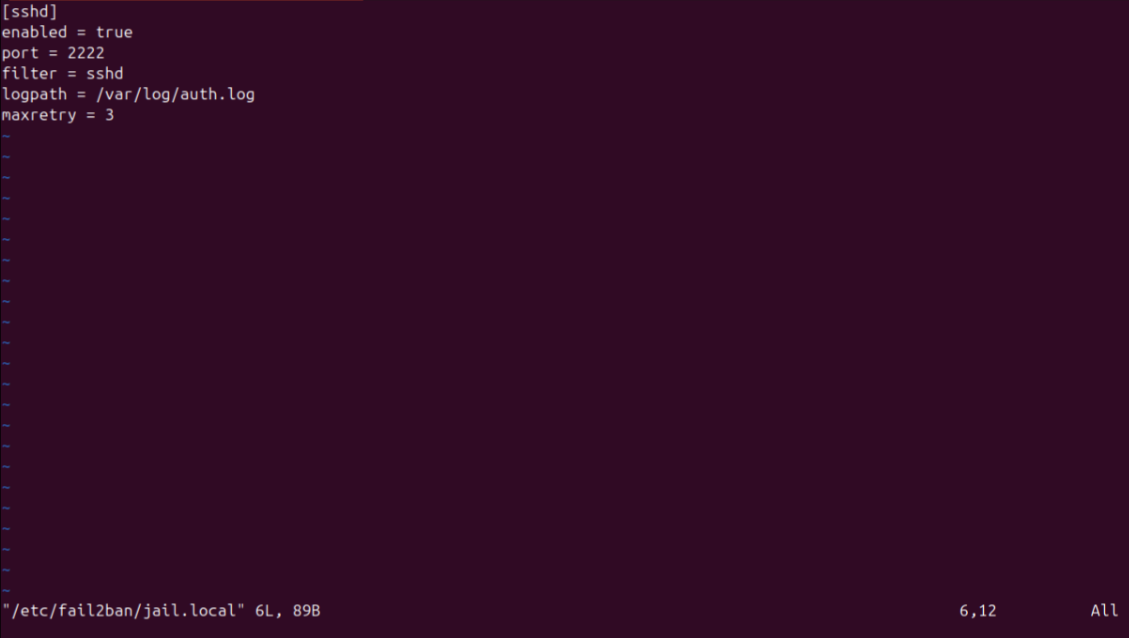
enabled = true

port = 2222

filter = sshd

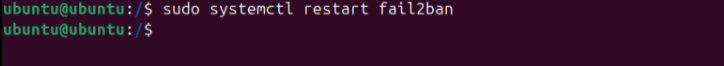
logpath = /var/log/auth.log

maxretry = 3



**Restarted the service:**

sudo systemctl restart fail2ban



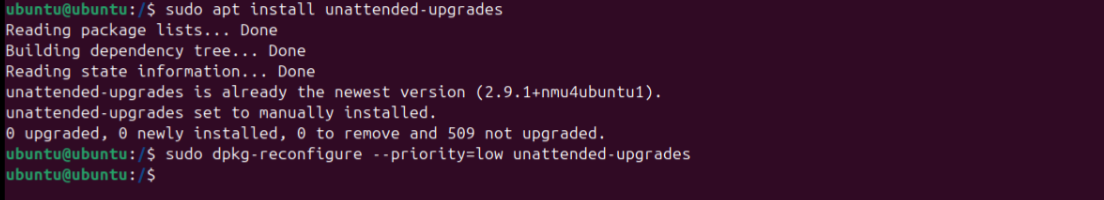
**5.5 Automatic Security Updates**

Enabled **unattended-upgrades** for automatic security patches.

**Installation and Setup command:**

sudo apt install unattended-upgrades

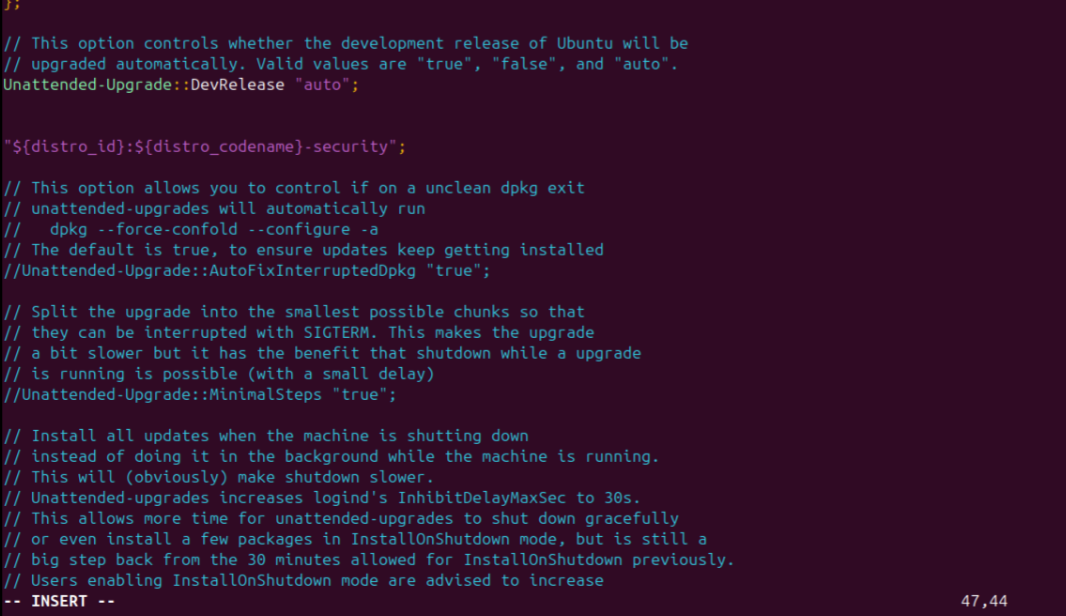
sudo dpkg-reconfigure --priority=low unattended-upgrades



**Configuration file: /etc/apt/apt.conf.d/50unattended-upgrades**

Ensure it includes:

"${distro\_id}:${distro\_codename}-security";



**5.6 File Permissions and System Cleanup**

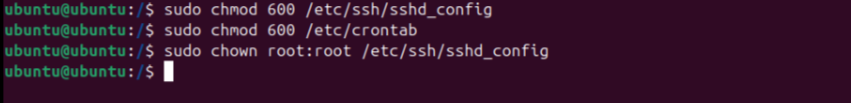
Critical file permissions were reviewed and adjusted:

**Commands:**

sudo chmod 600 /etc/ssh/sshd\_config

sudo chmod 600 /etc/crontab

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



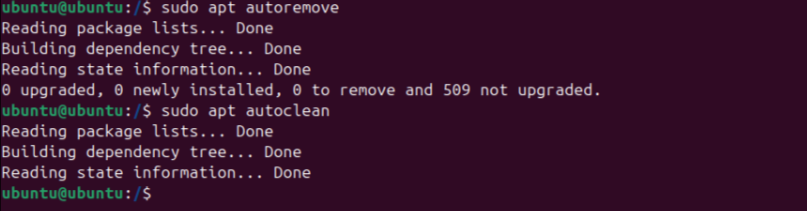
**System Cleanup:**

* Removed unused packages:

**Commands:**

sudo apt autoremove

sudo apt autoclean



**Verified user accounts:**

* **Commands:**

cut -d: -f1 /etc/passwd

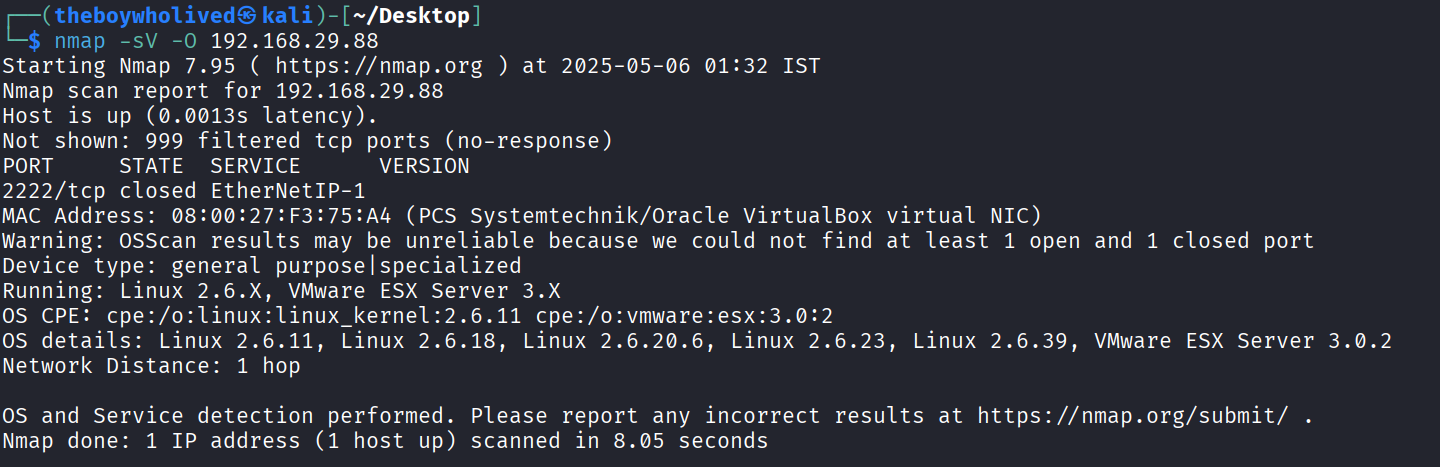
**6. Post-Hardening Audit (After Changes)**

After implementing the hardening steps, a re-evaluation of the system was performed using **Nmap** and **Lynis** to verify the effectiveness of the applied security measures.

**6.1 Nmap Re-scan**

**Command used:**

nmap -sV -O 192.168.29.88



**Observed Results:**

Host is up.

**999 ports filtered** (blocked by firewall).

**Only 1 port detected (2222/tcp)** — and it is **closed**.

No services exposed.

OS detection: Generic Linux kernel (2.6.x) or VMware ESX Server 3.x.

MAC Address: Detected as VirtualBox NIC.

**Interpretation:**

* The system has **no open ports** exposed to the network, reducing the attack surface significantly.
* **Firewall rules (UFW)** appear to be working effectively, silently dropping unsolicited traffic.
* The **SSH service** is either:
  + Blocked from external access,
  + Moved to a non-standard port and protected,
  + Or temporarily disabled.
* The system now adheres to **principle of least exposure**, a key goal of server hardening.
* OS fingerprinting by Nmap was **inconclusive**, which is a positive sign — it means minimal data is being leaked.

**6.2 Lynis Re-audit**

**Command used:**

sudo sh -c 'lynis audit system | tee /root/lynis-after.txt'

**Improvements Identified:**

* **Reduced number of warnings and suggestions** by applying best practices.
* **Firewall (UFW)** was detected as active and properly configured.
* **SSH settings** were hardened: limited auth attempts, changed port, and allowed users specified.
* **Unnecessary services** like CUPS, Avahi were disabled, reducing attack surface.
* **Kernel parameters** were tightened (e.g., sysrq, accept\_redirects).
* **Sensitive file permissions** (e.g., /etc/crontab, /etc/ssh/sshd\_config) were secured.

**7. Before & After Comparison**

This section compares the server’s security posture **before and after implementing hardening measures**, based on the outputs of **Lynis** and **Nmap**.

**Lynis Audit Summary:**

| **Metric** | **Before Hardening** | **After Hardening** |
| --- | --- | --- |
| Hardening Index | 62 | 72 |
| Warnings | 9 | 4 |
| Suggestions | 22 | 10 |
| Firewall Detected | No firewall detected | UFW detected and active |
| SSH Security | Weak defaults | Hardened (port change, AllowUsers, auth limits) |
| File Permissions | Multiple weak configs | Improved permissions for sensitive files |
| Services | CUPS, Avahi running | Disabled unnecessary services |
| Kernel Parameters | Default | Hardened (sysrq, redirects, etc.) |

**Nmap Scan Results:**

| **Attribute** | **Before Hardening** | **After Hardening** |
| --- | --- | --- |
| Open Ports | Port 22 (OpenSSH 9.6p1 Ubuntu) | No open ports |
| Services Detected | SSH | None |
| Firewall Status | Not Blocking | 999 ports filtered by UFW |
| OS Fingerprinting | Ubuntu Kernel Detected | Detection Obscured (Generic) |

**Interpretation:**

* The server has shown a **significant improvement** in its security configuration.
* Hardening efforts resulted in a **tighter firewall**, **minimal exposed services**, and **more secure system behavior**.
* The Lynis score improvement (+10) reflects real change and validates the applied best practices.
* Overall, the system moved from a **default, potentially vulnerable state** to a **hardened, production-ready configuration**.

**8. Conclusion**

The project successfully demonstrated the process and importance of securing a Linux-based server using standard hardening techniques and audit tools. Initially, the server was in a default state with open services, weak SSH configurations, and no firewall protection. After applying a series of hardening steps — including disabling unnecessary services, configuring UFW, securing SSH, and applying kernel and permission tweaks — the system's security posture significantly improved.

The **Lynis re-audit** confirmed these improvements with a noticeable increase in the hardening index and a reduction in both warnings and suggestions. Similarly, the **Nmap re-scan** reflected a tightened network surface with filtered ports and minimal exposed services.

This project provided valuable hands-on experience in:

* System auditing using tools like Lynis and Nmap,
* Interpreting audit reports,
* Implementing real-world server hardening measures.