Linux Hardening Audit – Manual Approach

By: Annu Bharti

1. Introduction

In this project, I conducted a detailed Linux system hardening audit manually using the Ubuntu environment provided by Windows Subsystem for Linux (WSL). The objective was to examine and analyze critical areas of a Linux system from a security perspective. This included identifying active services, open ports, firewall configuration, SSH login settings, and verifying file permissions of sensitive files.

Unlike automated tools, a manual audit requires close attention to detail, understanding system behaviors, interpreting configuration files, and verifying system responses to commands. This hands-on approach not only helped me become more confident using Linux but also taught me how to think like a security auditor.

This report documents the steps taken, commands used, screenshots saved, and key findings from the audit process. Each step was performed carefully, recorded, and analyzed for potential hardening opportunities.

2. Tools and Environment Used

- **Operating System:** Ubuntu (WSL on Windows)
- **Terminal Emulator:** Ubuntu Terminal (WSL)
- **Editor Used:** Nano (for configuration file reading)
- **Basic Linux Commands:** uname, lsb_release, ufw, nano, ss, ps, ls -l
- **Documentation Tools:** Screenshots, project report in PDF

3. Detailed Audit Steps

Step 1: Setting up the Environment

I installed Ubuntu through WSL (Windows Subsystem for Linux) on my Windows laptop. After successful installation, I created a Unix user account and accessed the Ubuntu terminal. This environment simulated a real Linux system and was essential for executing the audit commands.

Step 2: Gathering System Information

To understand the system version and kernel, I ran:

uname -a

lsb_release -a

These commands showed the Linux kernel version, OS version, and system architecture. This is essential for identifying if the system is running on a vulnerable or outdated version.

Step 3: Checking Firewall Configuration

I attempted to run sudo ufw status but initially found that UFW was not installed. I manually installed UFW using:

sudo apt update && sudo apt install ufw -y

After installation, I verified the firewall status. This step taught me how essential it is to not assume that security tools are enabled by default.

Step 4: Verifying SSH Root Login Policy

One of the most critical aspects of hardening is ensuring that root login over SSH is disabled. I opened the SSH configuration file using:

sudo nano /etc/ssh/sshd_config

I searched for the line starting with PermitRootLogin. It was set to prohibit-password, which is a secure setting since it disables root login via passwords.

Step 5: Identifying Open Ports and Running Services

I used the following command to check open TCP and UDP ports:

ss -tuln

This showed ports 22 (SSH) and 53 (DNS) as open, which are expected. To view all running processes, I used:

ps aux

This provided a list of services such as sshd, cron, and others. This step helped in identifying if there were any unknown or unnecessary services running.

Step 6: Validating File Permissions of Critical Files

I checked the permissions of two key system files:

ls -l /etc/passwd /etc/shadow

The output confirmed:

- /etc/passwd was readable by all users but writable only by root (-rw-r--r--)

- /etc/shadow was readable only by root and the shadow group (-rw-r-----)

This showed that password-related information is protected properly, reducing the risk of password theft.

4. Conclusion

Through this project, I explored several essential aspects of Linux system hardening. From setting up Ubuntu via WSL, learning how to use basic terminal commands, installing and verifying security tools, to analyzing services and configuration files, every step was educational and hands-on.

I now understand how to:

- Examine system version and kernel
- Install and check firewall status
- Review SSH settings securely
- Identify open ports and active processes
- Confirm proper file permissions on sensitive files

This was not just a checklist project; I manually verified each setting and interpreted results, making me more confident in basic Linux system auditing. This foundation will support my growth in cybersecurity and help me in future roles involving Linux or infrastructure security assessments.