LINUX PRIVILEGE ESCALATION AUTOMATION TOOLKIT (Detection Only)

A Project Report submitted in partial fulfillment of the requirements for the completion of

UNIFIED INTERNSHIP PROGRAM

(Cyber Security / Ethical Hacking)

Submitted By

Name : Aquib Ahmad Khan

Intern ID : UMID27112571911

Domain : Cyber Security

Submitted To

Unified Internship

Operating System Used

Parrot OS (Linux)

Project Duration

January 2026 – February 2026

1. Project Overview

Linux operating systems form the backbone of modern computing environments, including enterprise servers, cloud infrastructures, web applications, and cybersecurity laboratories. Due to their open-source nature and flexible permission model, Linux systems require careful configuration and continuous security monitoring.

Privilege escalation is a critical security issue in Linux environments where a low-privileged user gains unauthorized administrative (root) access. Such escalation often occurs due to misconfigured file permissions, insecure scheduled tasks (cron jobs), vulnerable system services, or outdated kernel versions.

This project presents a Linux Privilege Escalation Automation Toolkit designed to safely identify these misconfigurations. The toolkit automates enumeration techniques commonly used by penetration testers and security auditors, while strictly operating in detection-only mode. No exploitation or system modification is performed during the assessment.

2. Objectives

The main objectives of this project are listed below:

- To understand the concept of privilege escalation in Linux systems
- To identify misconfigurations that may lead to unauthorized privilege gain
- To automate system enumeration using secure scripting techniques
- To bridge the gap between red team enumeration and blue team defense
- To prepare a detailed and professional security assessment report

3. Tools and Technologies Used

Operating System: Parrot OS (Linux), selected for its security-focused environment
Scripting Language: Bash, used to automate enumeration tasks
Linux Utilities: whoami, id, uname, find, systemctl, crontab for system inspection
Documentation Tool: Microsoft Word, used to prepare and export the final PDF report

4. Methodology

The project follows a structured methodology to ensure complete and ethical system analysis:

Step 1: System Information Collection – Identifies current user context, groups, and kernel details.

Step 2: SUID and SGID Enumeration – Detects binaries running with elevated privileges.

Step 3: Weak File Permission Analysis – Identifies world-writable and insecure files.

Step 4: Cron Job Analysis – Reviews scheduled tasks executed with higher privileges.

Step 5: Service Configuration Review – Inspects system services for misconfigurations.

Step 6: Kernel Version Assessment – Checks for outdated or vulnerable kernels.

5. Automated Scanner Script

```
#!/bin/bash

echo "SYSTEM INFO"
whoami
id
uname -a

echo "SUID FILES"
find / -perm -4000 2>/dev/null

echo "SGID FILES"
find / -perm -2000 2>/dev/null

echo "WRITABLE FILES"
find / -writable -type f 2>/dev/null | head -20

echo "CRON JOBS"
crontab -l 2>/dev/null
ls -la /etc/cron.*

echo "SERVICES"
systemctl list-unit-files --type=service | head -20

echo "KERNEL VERSION"
uname -r
```

6. Findings

Based on the execution of the automated scanner, the following observations were recorded:

SUID Binaries (Medium Risk): Standard SUID binaries were detected which may be abused if misconfigured.

Writable Files (Medium Risk): Writable files were identified that could be modified by unauthorized users.

Cron Jobs (Low Risk): No directly writable root cron scripts were detected in the environment.

Services (Low Risk): System services appeared to be properly permissioned during the scan.

Kernel Version (Medium Risk): The kernel version should be reviewed against known CVEs.

7. Mitigation and Recommendations

- Remove unnecessary SUID and SGID permissions from binaries
- Apply strict file and directory permission policies
- Secure cron jobs by enforcing correct ownership and permissions
- Periodically audit system services and startup configurations
- Keep the Linux kernel updated with the latest security patches

8. Screenshots & Practical Evidence

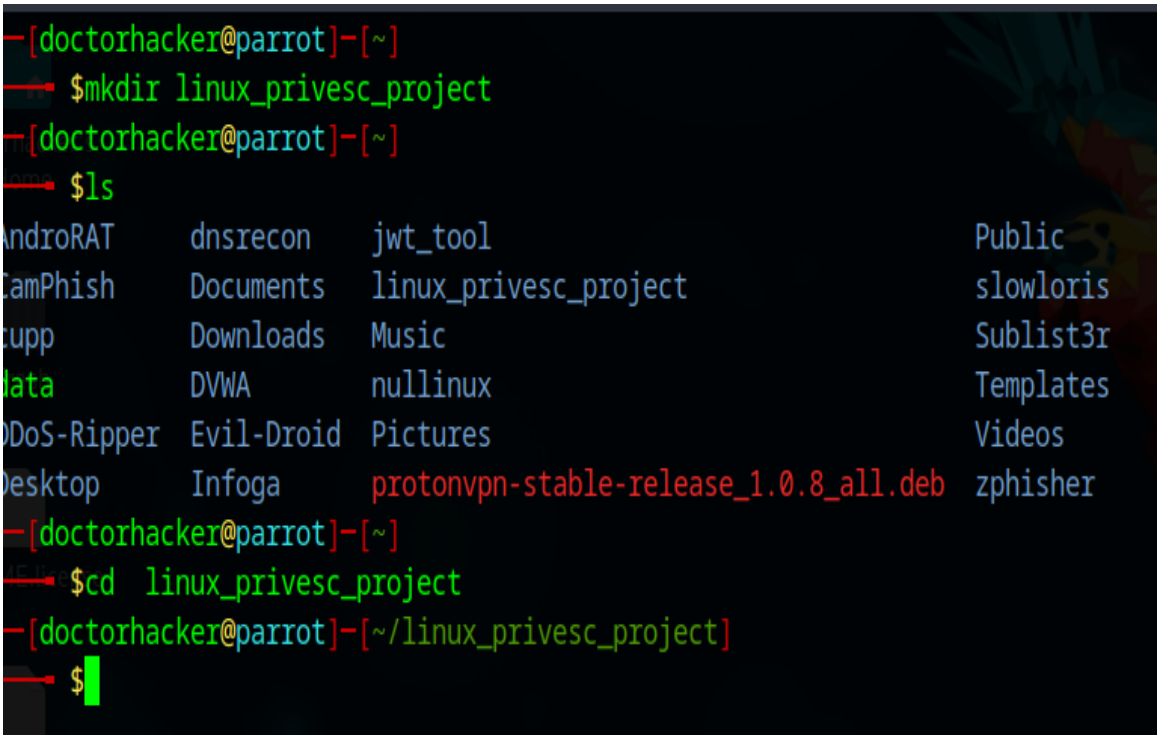

Figure 1: Practical output screenshot

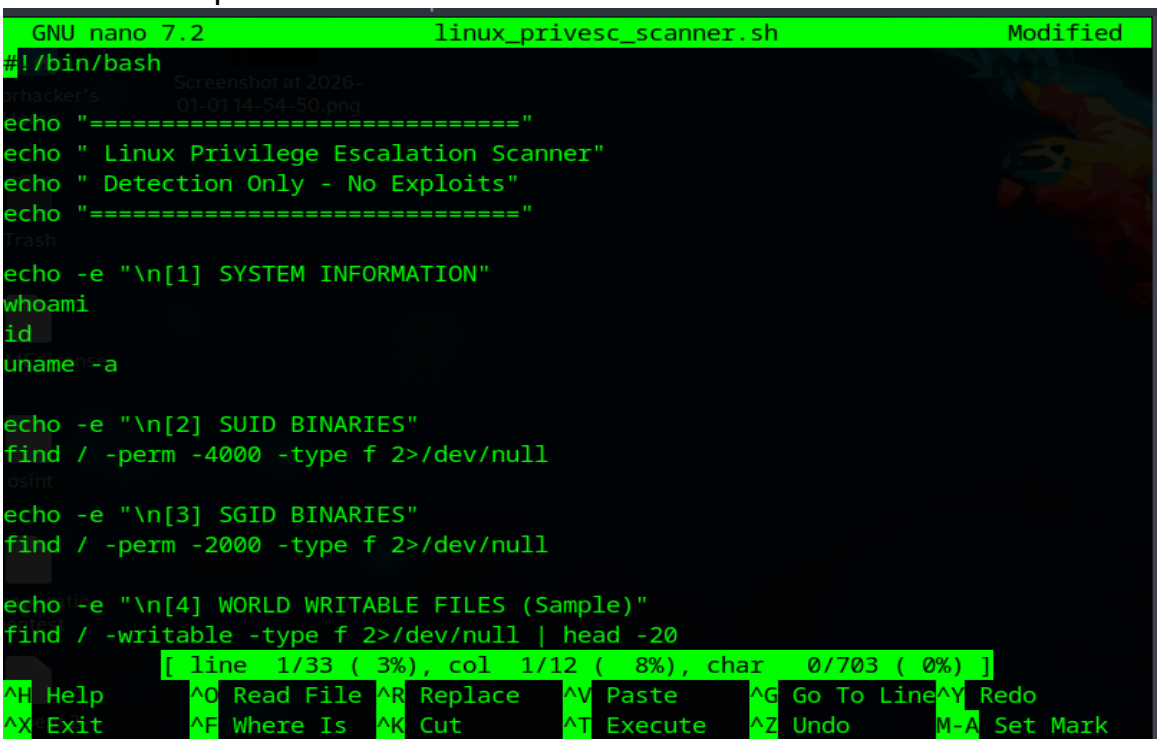

Figure 2: Practical output screenshot

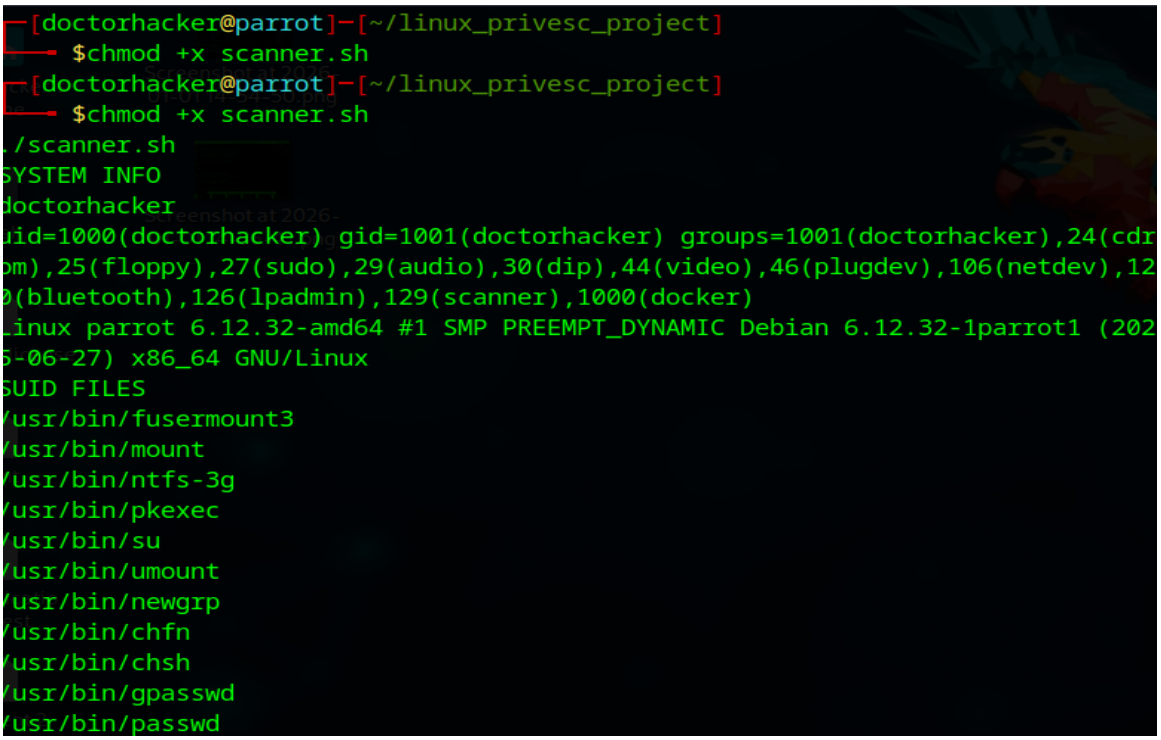

Figure 3: Practical output screenshot

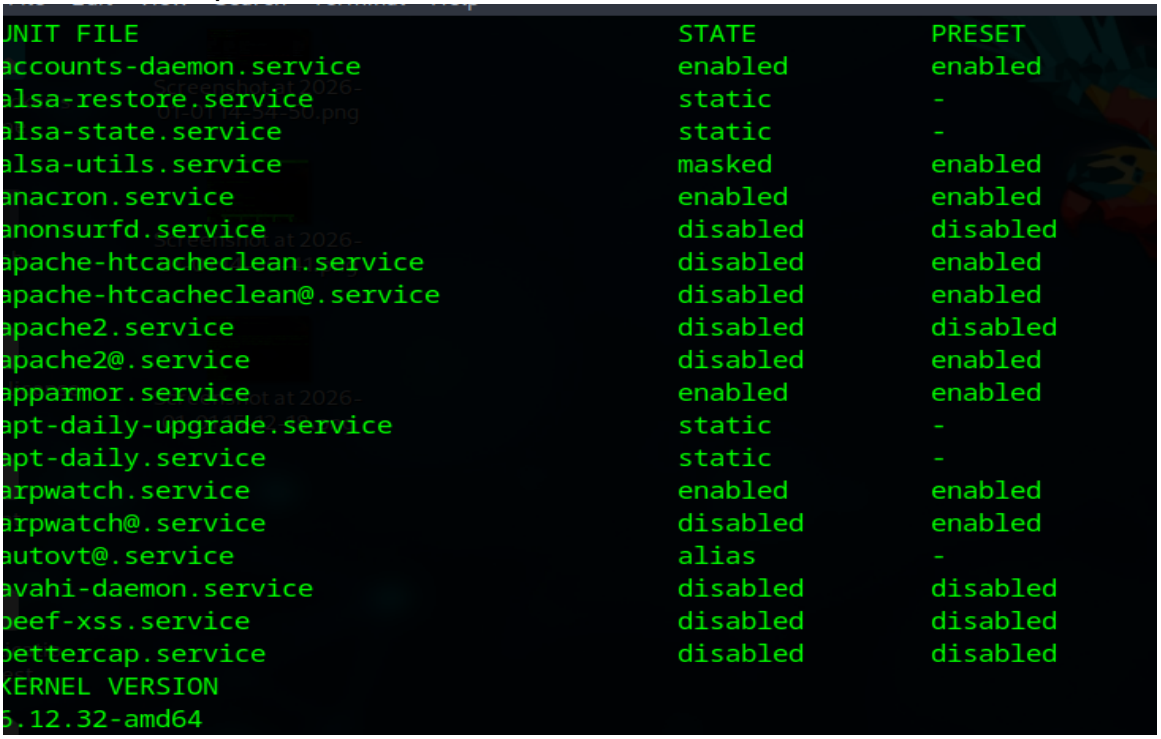

Figure 4: Practical output screenshot

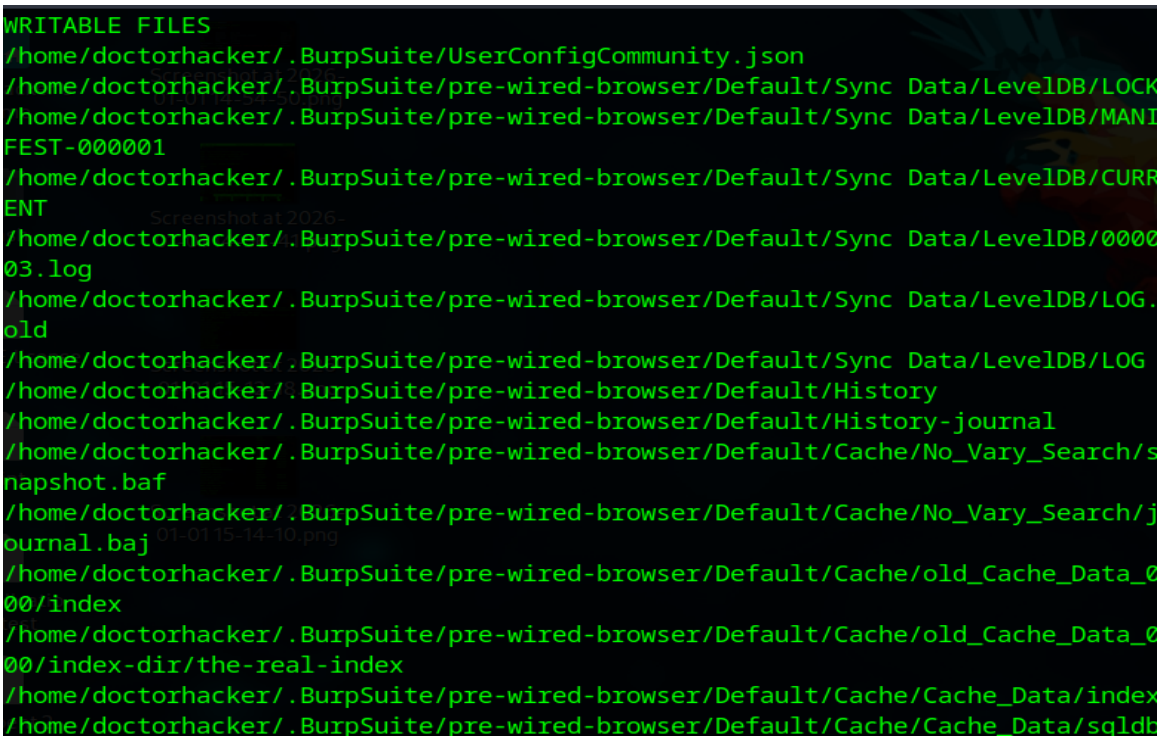

Figure 5: Practical output screenshot

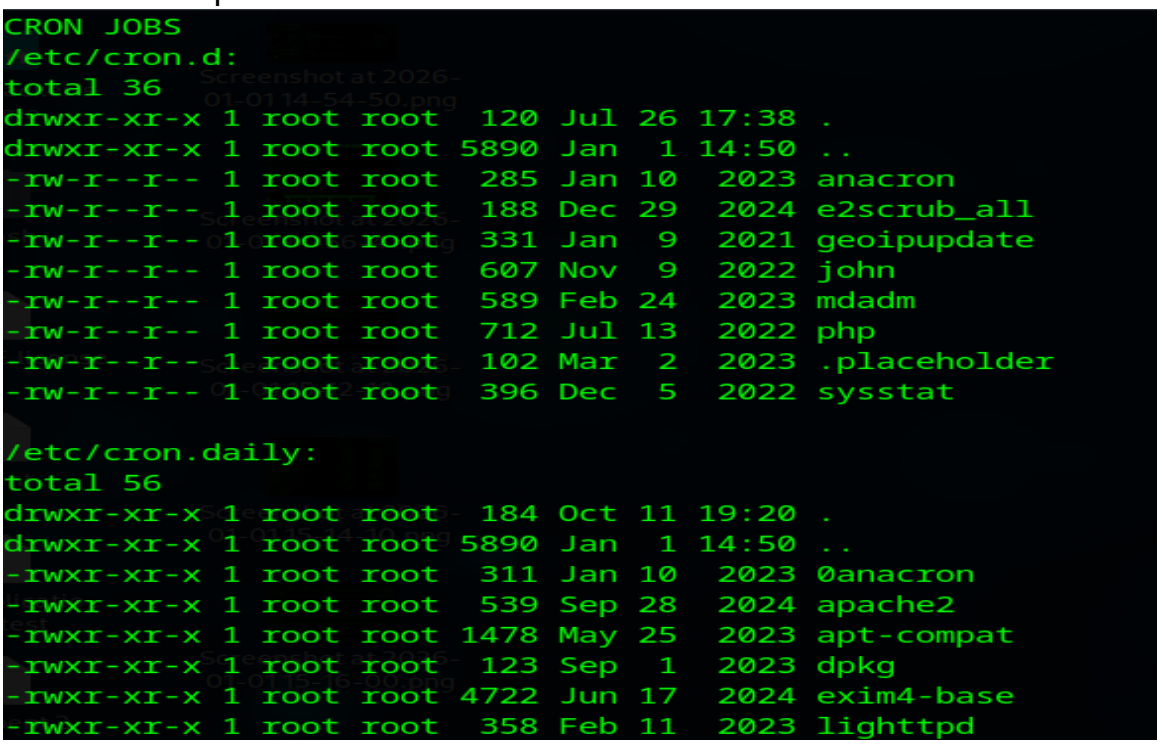

Figure 6: Practical output screenshot

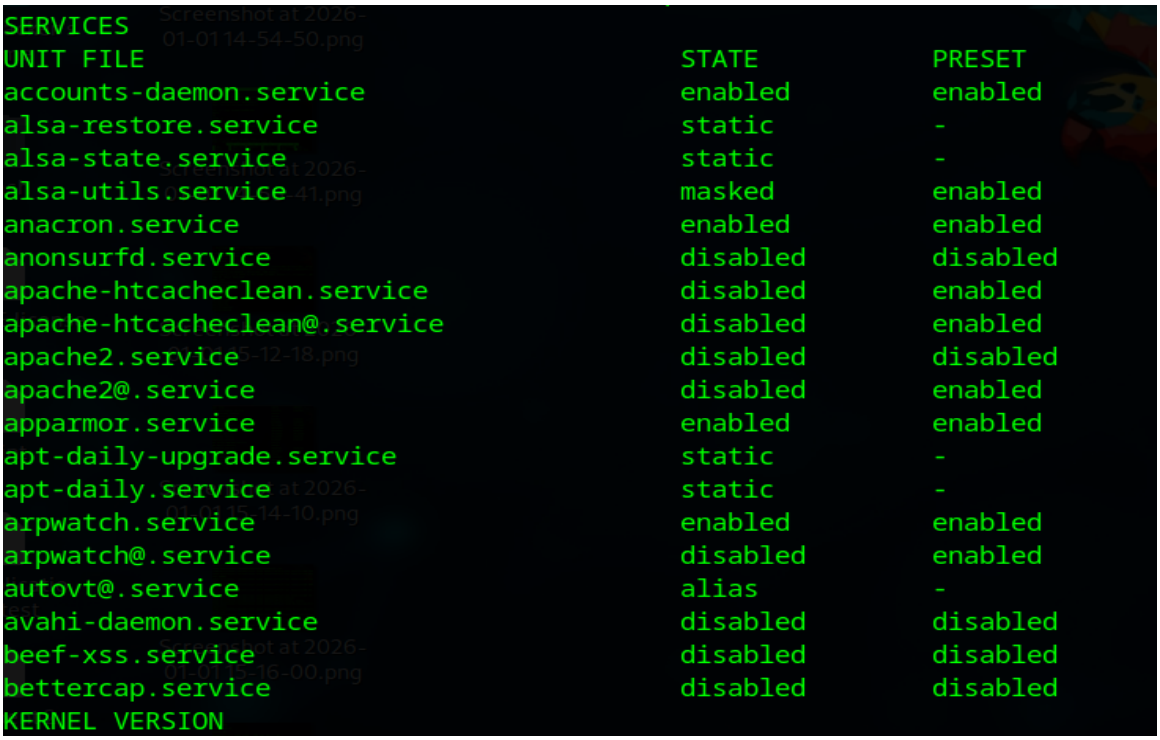

Figure 7: Practical output screenshot

Figure 8: Practical output screenshot

9. Conclusion

This project provides a comprehensive and ethical approach to identifying Linux privilege escalation risks using automated enumeration techniques. By combining scripting, analysis, and reporting, the project strengthens both offensive awareness and defensive security practices.

10. Declaration

I hereby declare that this project is my original work completed as part of the Unified Internship Program. No exploitation or harmful techniques were used during the execution of this project.

Name: Aquib Ahmad Khan