

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occurs due to misconfigured file permissions, insecure scheduled tasks (cron jobs), vulnerable system services, or outdated kernel versions.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

```
[doctorhacker@parrot]-[~]
$mkdir linux_privesc_project
[doctorhacker@parrot]-[~]
$ls
AndroRAT      dnsrecon      jwt_tool      Public
CamPhish      Documents     linux_privesc_project  slowloris
Cupp          Downloads     Music          Sublist3r
data          DVWA          nulllinux     Templates
DoS-Ripper    Evil-Droid    Pictures       Videos
Desktop       Infoga        protonvpn-stable-release_1.0.8_all.deb  zphisher
[doctorhacker@parrot]-[~]
$cd linux_privesc_project
[doctorhacker@parrot]-[~/linux_privesc_project]
$
```

Figure 1: Practical output screenshot

```
GNU nano 7.2      linux_privesc_scanner.sh      Modified
#!/bin/bash
Screenshot at 2026-01-01 14:54:50.png
echo "=====
echo " Linux Privilege Escalation Scanner"
echo " Detection Only - No Exploits"
echo "=====
echo -e "\n[1] SYSTEM INFORMATION"
whoami
id
uname -a

echo -e "\n[2] SUID BINARIES"
find / -perm -4000 -type f 2>/dev/null

echo -e "\n[3] SGID BINARIES"
find / -perm -2000 -type f 2>/dev/null

echo -e "\n[4] WORLD WRITABLE FILES (Sample)"
find / -writable -type f 2>/dev/null | head -20
[ line 1/33 ( 3%), col 1/12 ( 8%), char 0/703 ( 0%) ]
^H Help      ^O Read File ^R Replace    ^V Paste     ^G Go To Line ^Y Redo
^X Exit      ^F Where Is  ^K Cut       ^T Execute   ^Z Undo      M-A Set Mark
```

Figure 2: Practical output screenshot

```

[doctorhacker@parrot]--[~/linux_privesc_project]
$chmod +x scanner.sh
[doctorhacker@parrot]--[~/linux_privesc_project]
$chmod +x scanner.sh
./scanner.sh
SYSTEM INFO
doctorhacker
uid=1000(doctorhacker) gid=1001(doctorhacker) groups=1001(doctorhacker),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),106(netdev),129(bluetooth),126(lpadmin),129(scanner),1000(docker)
Linux parrot 6.12.32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.32-1parrot1 (2025-06-27) x86_64 GNU/Linux
SUID FILES
/usr/bin/fusermount3
/usr/bin/mount
/usr/bin/ntfs-3g
/usr/bin/pkexec
/usr/bin/su
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/passwd

```

Figure 3: Practical output screenshot

UNIT FILE	STATE	PRESET
accounts-daemon.service	enabled	enabled
alsa-restore.service	static	-
alsa-state.service	static	-
alsa-utils.service	masked	enabled
anacron.service	enabled	enabled
anonsurfd.service	disabled	disabled
apache-htcacheclean.service	disabled	enabled
apache-htcacheclean@.service	disabled	enabled
apache2.service	disabled	disabled
apache2@.service	disabled	enabled
apparmor.service	enabled	enabled
apt-daily-upgrade.service	static	-
apt-daily.service	static	-
arpwatch.service	enabled	enabled
arpwatch@.service	disabled	enabled
autovt@.service	alias	-
avahi-daemon.service	disabled	disabled
beef-xss.service	disabled	disabled
bettercap.service	disabled	disabled

KERNEL VERSION
6.12.32-amd64

Figure 4: Practical output screenshot

```

WRITABLE FILES
/home/doctorhacker/.BurpSuite/UserConfigCommunity.json
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Sync Data/LevelDB/LOCK
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Sync Data/LevelDB/MANIFEST-000001
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Sync Data/LevelDB/CURRENT
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Sync Data/LevelDB/000003.log
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Sync Data/LevelDB/LOG.old
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Sync Data/LevelDB/LOG
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/History
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/History-journal
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Cache/No_Vary_Search/snapshot.baf
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Cache/No_Vary_Search/journal.baj
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Cache/old_Cache_Data_000/index
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Cache/old_Cache_Data_000/index-dir/the-real-index
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Cache/Cache_Data/index
/home/doctorhacker/.BurpSuite/pre-wired-browser/Default/Cache/Cache_Data/sqlite

```

Figure 5: Practical output screenshot

```

CRON JOBS
/etc/cron.d:
total 36
drwxr-xr-x 1 root root 120 Jul 26 17:38 .
drwxr-xr-x 1 root root 5890 Jan 1 14:50 ..
-rw-r--r-- 1 root root 285 Jan 10 2023 anacron
-rw-r--r-- 1 root root 188 Dec 29 2024 e2scrub_all
-rw-r--r-- 1 root root 331 Jan 9 2021 geoipupdate
-rw-r--r-- 1 root root 607 Nov 9 2022 john
-rw-r--r-- 1 root root 589 Feb 24 2023 mdadm
-rw-r--r-- 1 root root 712 Jul 13 2022 php
-rw-r--r-- 1 root root 102 Mar 2 2023 .placeholder
-rw-r--r-- 1 root root 396 Dec 5 2022 sysstat

/etc/cron.daily:
total 56
drwxr-xr-x 1 root root 184 Oct 11 19:20 .
drwxr-xr-x 1 root root 5890 Jan 1 14:50 ..
-rwxr-xr-x 1 root root 311 Jan 10 2023 anacron
-rwxr-xr-x 1 root root 539 Sep 28 2024 apache2
-rwxr-xr-x 1 root root 1478 May 25 2023 apt-compat
-rwxr-xr-x 1 root root 123 Sep 1 2023 dpkg
-rwxr-xr-x 1 root root 4722 Jun 17 2024 exim4-base
-rwxr-xr-x 1 root root 358 Feb 11 2023 lighttpd

```

Figure 6: Practical output screenshot

SERVICES
 UNIT FILE
 accounts-daemon.service
 alsa-restore.service
 alsa-state.service
 alsa-utils.service
 anacron.service
 anonsurfd.service
 apache-htcacheclean.service
 apache-htcacheclean@.service
 apache2.service
 apache2@.service
 apparmor.service
 apt-daily-upgrade.service
 apt-daily.service
 arpwatch.service
 arpwatch@.service
 autovt@.service
 avahi-daemon.service
 beef-xss.service
 bettercap.service
 KERNEL VERSION

STATE	PRESET
enabled	enabled
static	-
static	-
masked	enabled
enabled	enabled
disabled	disabled
disabled	enabled
disabled	enabled
disabled	disabled
disabled	enabled
enabled	enabled
static	-
static	-
enabled	enabled
disabled	enabled
alias	-
disabled	disabled
disabled	disabled
disabled	disabled

Figure 7: Practical output screenshot

KERNEL VERSION
 6.12.32-amd64

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