Linux Privilege Escalation Automation Toolkit

Security Analysis Report

---

1. Introduction

1.1 Project Overview

Linux Privilege Escalation occurs when a low-privileged user gains unauthorized root access due to system misconfigurations, insecure permissions, or vulnerable services.

This project focuses on developing an **automated, detection-only Linux Privilege Escalation Automation Toolkit** that scans a Linux system to identify common privilege escalation vectors **without performing any exploitation**.

The toolkit simulates **real-world red-team enumeration techniques** while producing **actionable security findings** suitable for **blue-team remediation and auditing**.

1.2 Practical Motivation

Manual privilege escalation auditing is time-consuming and prone to human error. Many Linux servers remain vulnerable due to:

- Forgotten SUID binaries
- Writable root-owned files
- Misconfigured cron jobs
- Insecure system services
- Weak sudo policies
- Outdated kernels

This project automates these checks and provides structured security reports, helping students understand **how attackers escalate privileges** and **how defenders can prevent it**.

---

2. Project Objectives

The objectives of this project are:

- To develop a **fully automated Linux privilege escalation scanner**
- To identify real misconfigurations through practical testing
- To simulate **red-team enumeration techniques**
- To generate a **professional security audit report**
- To ensure the toolkit is **safe, ethical, and detection-only**

---

3. Tools & Technologies Used

3.1 Programming Languages

- Python 3
- Bash (via system commands)

3.2 Linux Utilities Used

- find
- ls -la
- systemctl
- sudo -l
- crontab
- uname -a
- grep, awk, sed

3.3 Environment

- Kali Linux
- Non-root user execution

---

4. Toolkit Architecture & Workflow

4.1 Workflow Explanation

```
START
↓
System Information Gathering
↓
SUID / SGID Binary Scan
↓
Weak Permission Scan
↓
Cron Job Scan
↓
systemd Service Scan
↓
Sudo Misconfiguration Scan
↓
Kernel Vulnerability Awareness
↓
Risk Analysis & Severity Classification
↓
Report Generation
↓
END
```

4.2 Modular Architecture

The toolkit is divided into independent modules:

- system_info.py – System enumeration
- suid_scan.py – SUID/SGID binary detection
- permission_scan.py – Weak permission detection
- cron_scan.py – Cron job analysis
- service_scan.py – systemd service misconfiguration detection
- sudo_scan.py – sudo privilege analysis
- kernel_scan.py – Kernel vulnerability awareness
- risk_engine.py – Centralized risk analysis
- report_generator.py – Report generation

---

5. Detailed Analysis of Privilege Escalation Vectors

5.1 System Information Collection

Purpose:
To identify the execution context and system attack surface.

Information Collected: - Current user and UID - Group memberships - OS version - Kernel version - Privilege level (root / non-root)

Security Importance:
Kernel version and user privileges determine which escalation techniques are feasible.

5.2 SUID / SGID Binary Discovery

Description:
The toolkit scans the filesystem for binaries with SUID or SGID permissions.

Risk Explanation:
SUID binaries execute with root privileges. If misconfigured or vulnerable, they can allow privilege escalation.

Severity: HIGH / MEDIUM

Mitigation: - Remove unnecessary SUID bits - Restrict execution permissions

---

5.3 Weak File & Directory Permissions

Description:
The toolkit identifies world-writable files, world-writable directories, and writable root-owned files.

Risk Explanation:
Writable files used by root-owned processes allow attackers to inject malicious code.

Severity: HIGH

Mitigation: - Restrict permissions using chmod - Apply proper ownership with chown

---

5.4 Cron Job Vulnerability Analysis

Description:
System-wide cron jobs are scanned to detect writable scripts or misconfigured cron files.

Risk Explanation:
Cron jobs execute automatically with elevated privileges. Writable cron scripts lead to time-based privilege escalation.

Severity: HIGH

Mitigation: - Restrict cron file permissions - Use absolute paths - Limit write access

---

5.5 systemd Service Misconfiguration Detection

Description:
The toolkit inspects running systemd services that execute as root.

Risk Explanation:
If a root-run service executes a writable file or resides in a writable directory, it creates a persistent escalation vector.

Severity: CRITICAL / HIGH

Mitigation: - Restrict permissions on service executables - Run services with least privilege

---

5.6 Sudo Misconfiguration Analysis

Description:
The toolkit analyzes sudo privileges for dangerous rules.

High-Risk Patterns: - NOPASSWD rules - Wildcards - Shell-capable commands

Severity: CRITICAL

Mitigation: - Remove unnecessary sudo permissions - Avoid wildcard rules - Require password authentication

---

5.7 Kernel Vulnerability Awareness

Description:
The toolkit evaluates kernel versions to identify outdated or end-of-life kernels.

Risk Explanation:
Outdated kernels may contain known local privilege escalation vulnerabilities.

Severity: HIGH

Mitigation: - Apply kernel updates - Enable automatic security updates

---

6. Risk Classification & Analysis Engine

All findings are centralized and classified using severity levels:

Severity	Description
CRITICAL	Direct root access
HIGH	Strong escalation path
MEDIUM	Conditional escalation
LOW	Informational

---

7. Report Generation

The toolkit generates:

- TXT report – Human-readable audit report
- JSON report – Machine-readable structured output

Each finding includes category, severity, description, and mitigation steps.

---

8. Learning Outcomes

- Practical Linux privilege escalation understanding
- Red-team enumeration techniques
- Blue-team auditing mindset
- Secure permission analysis
- Modular Python scripting
- Professional security reporting

---

9. Conclusion

The Linux Privilege Escalation Automation Toolkit successfully automates the detection of common privilege escalation vectors without exploiting vulnerabilities.

The project demonstrates real-world security auditing techniques used by penetration testers and SOC analysts and provides actionable remediation guidance for system administrators.

---

10. Disclaimer

This toolkit is developed strictly for **educational and security auditing purposes**. No exploitation is performed, and misuse of the tool is not encouraged.

---

11. References

- Linux Manual Pages
- GTFOBins
- MITRE ATT&CK
- Linux Security Best Practices