PSP0201

Week 2

Writeup

Group Name: OraOraOra

Members

ID	Name	Role
1211103141	Muhammad Haikal Afiq Bin Rafingei	Leader
1211103148	Muhamad Izzul Iqbal Bin Ismail	Member
1211103830	Hakeem Bin Aminudin	Member

Day 1: Web Exploitation – A Christmas Crisis

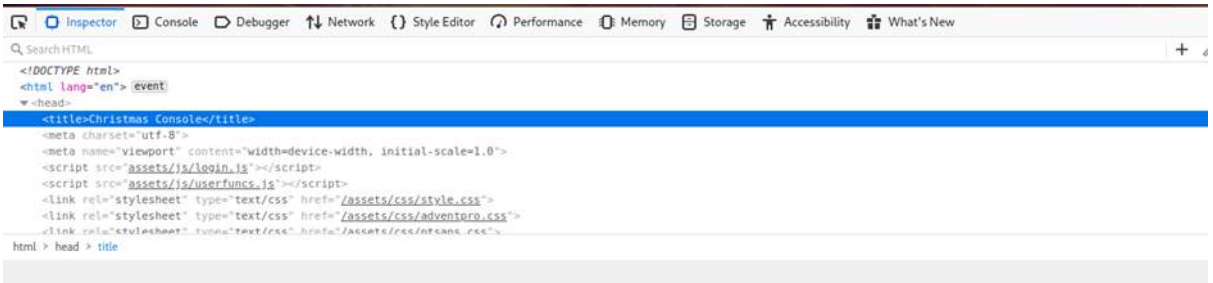
Tools used: Kali Linux

Solution/Walkthrough:

Question 1

Answer: Christmas Console

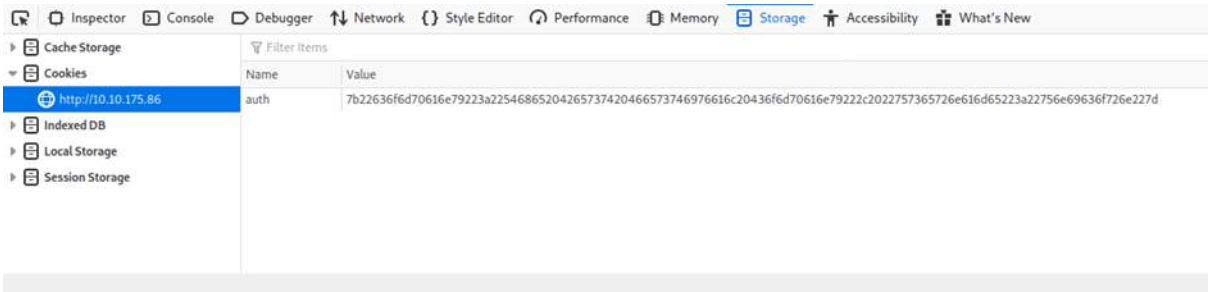
We inspect the website by clicking F12. Take a look at the title tag to get the website title.



Question 2

Answer: auth

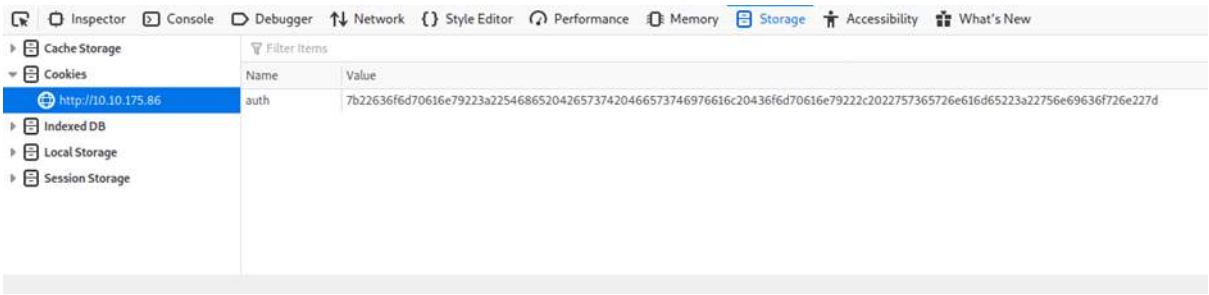
After we register and login, there will be cookies saved. By inspecting it, we can get the name of the cookie under the Storage tab.



Question 3

Answer: Hexadecimal

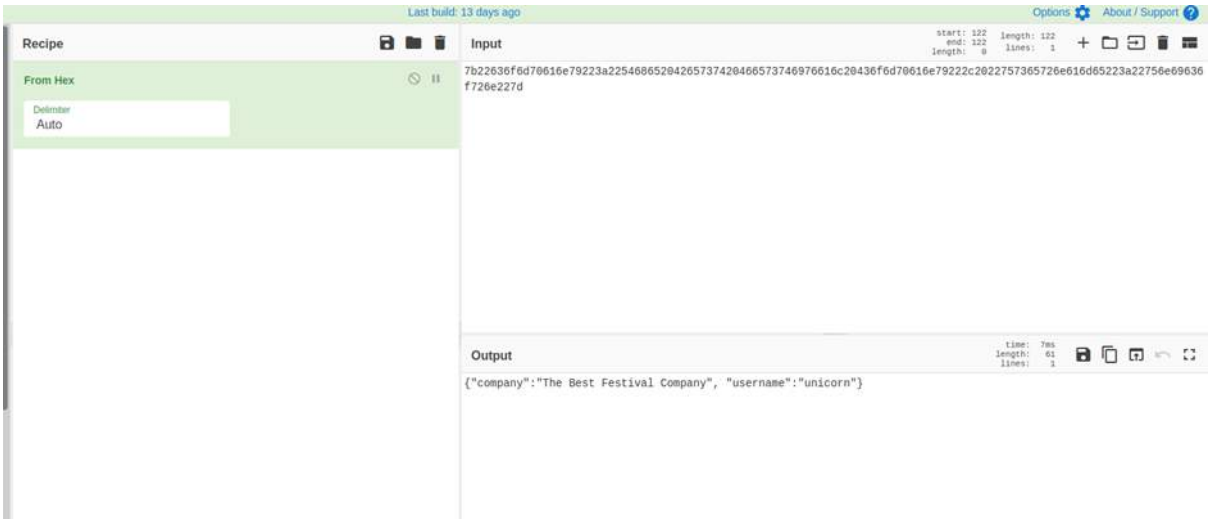
The value of the cookie starts from 0 to f. So, we know that it is saved in hexadecimal form.



Question 4

Answer: JSON

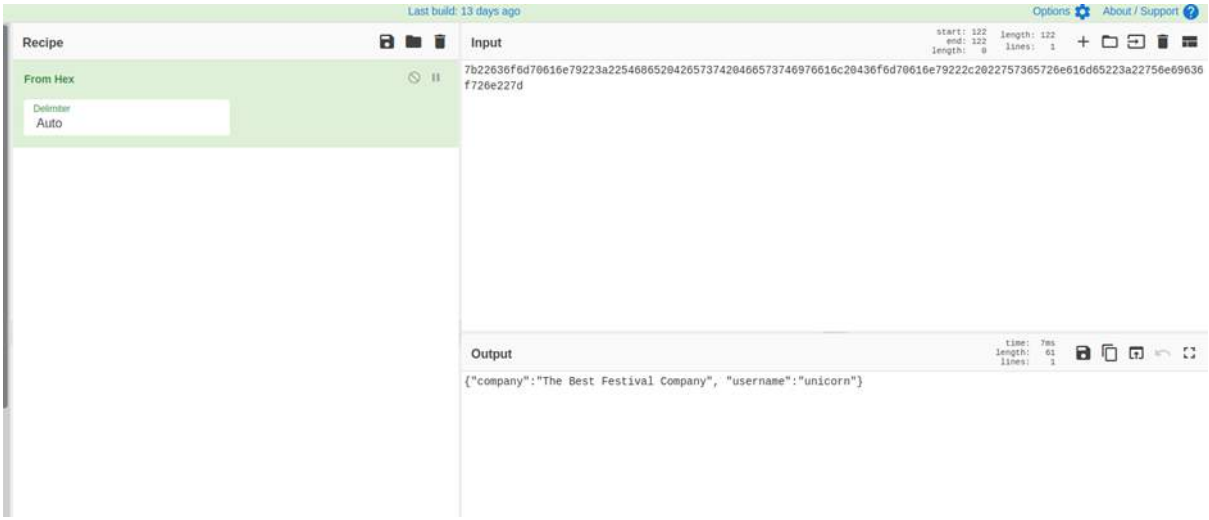
To get the value of the cookie, we use cyberchef to change the format. The output shows that it is in the form of JSON.



Question 5

Answer: The Best Festival Company

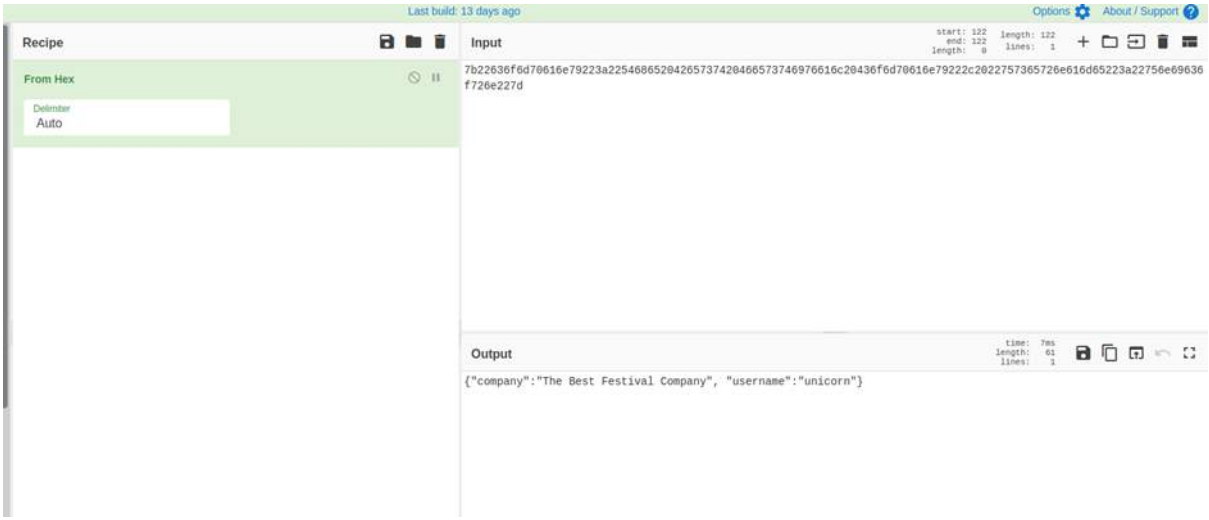
The value for the company can be seen in the output.



Question 6

Answer: username

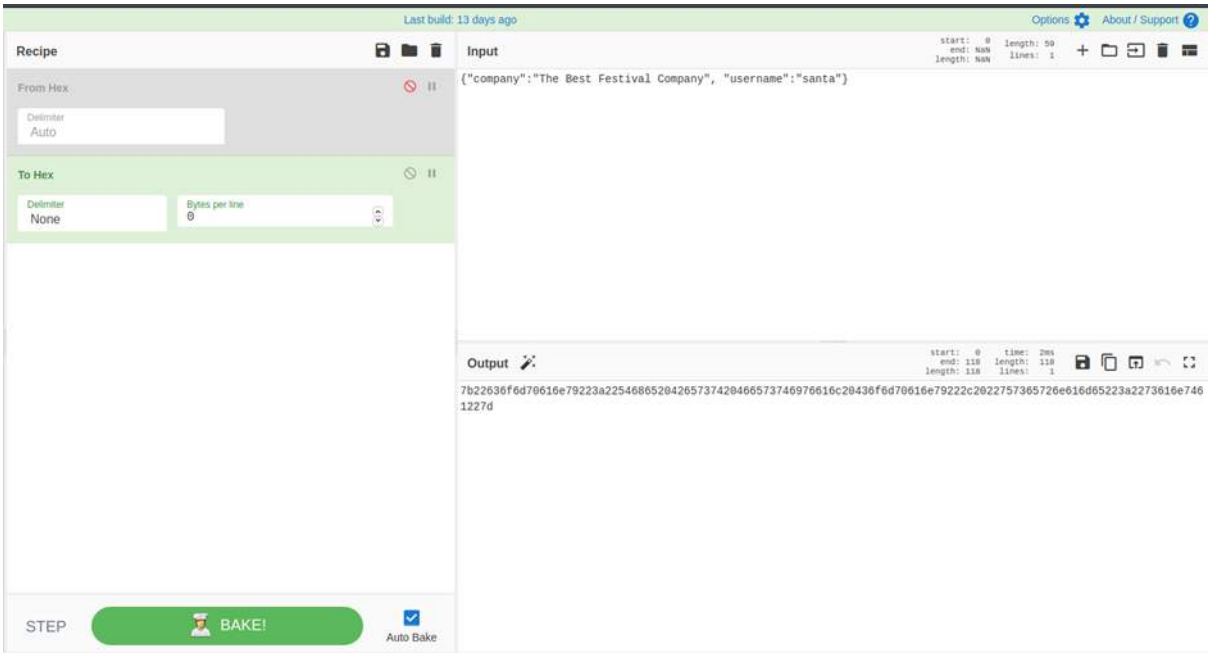
There are two pieces of information in the cookie. The other one is username.



Question 7

Answer:
7b22636f6d70616e79223a225468652042657374204665737469766 16c20436f6d70616e79222c2022757365726e616d65223a2273616e7461227d

To get the value of santa cookie, we just change the username to santa and turn it back into the hexadecimal form.



Question 8

Answer: THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWFhYmQy}

The flag can be seen after we switch on all the controls.

Methodology:

First, we register and login to get the cookie. From the cookie, we can see it is in hexadecimal format so we use cyberchef to translate it. We change the value of the username to santa to get santa's cookie and use the cookie in the website to bypass the login authentication. Now, we can control all the control panels and save Christmas hehe.

Day 2: Web Exploration – A Christmas Crisis

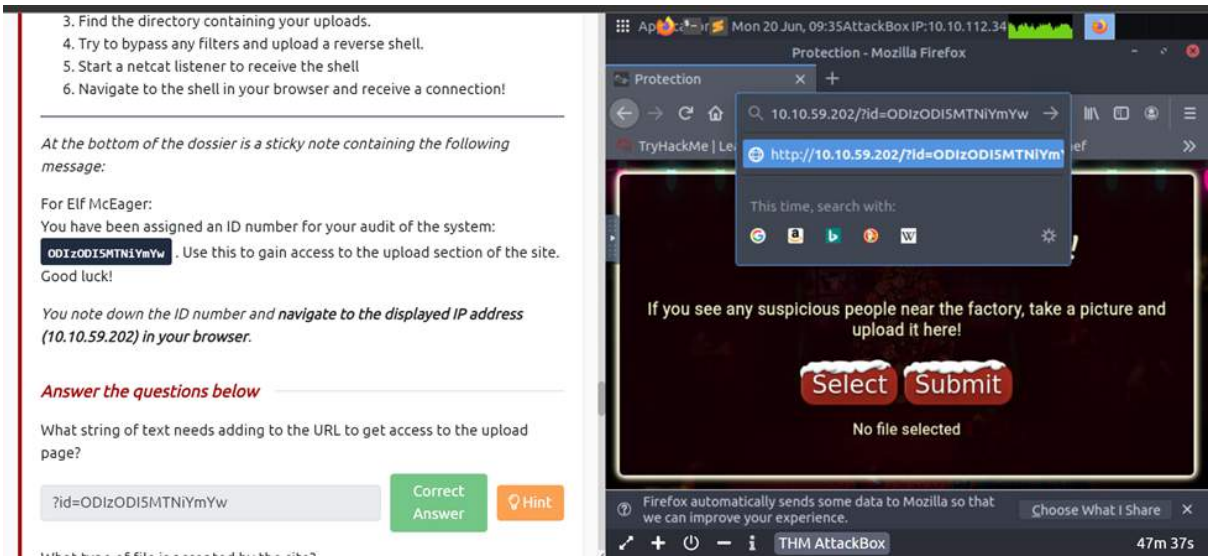
Tools used: TryHackMe AttackBox, Kali Linux (for question 4), Shell, Mozilla Firefox

Solution / Walkthrough:

Question 1

Answer: ODIzODI5MTNiYmYw

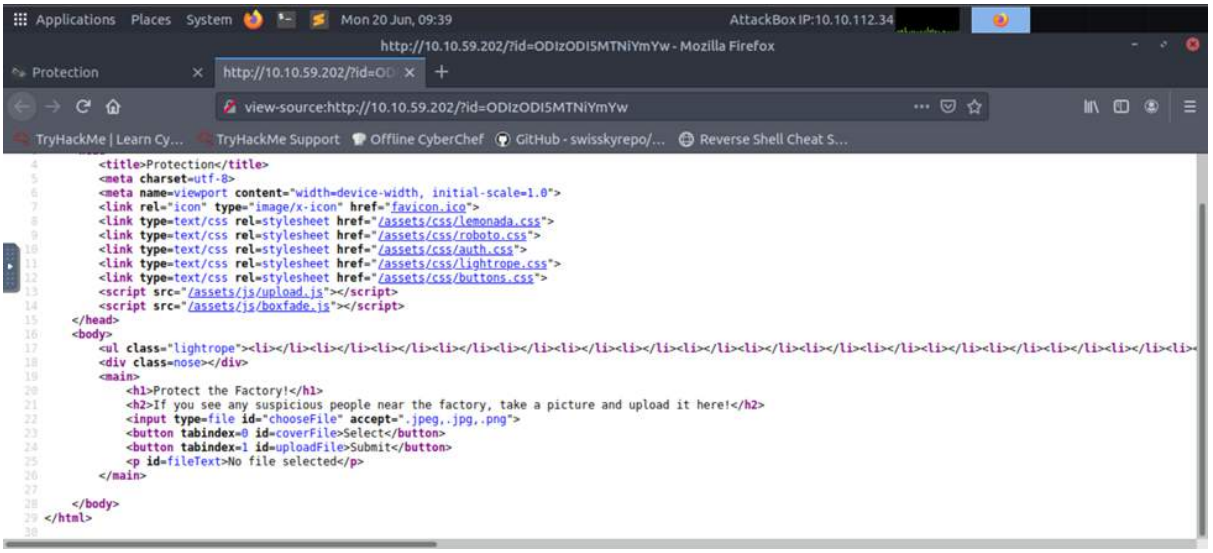
Add "?id=ODIzODI5MTNiYmYw" as mentioned in the message.



Question 2

Answer: Image

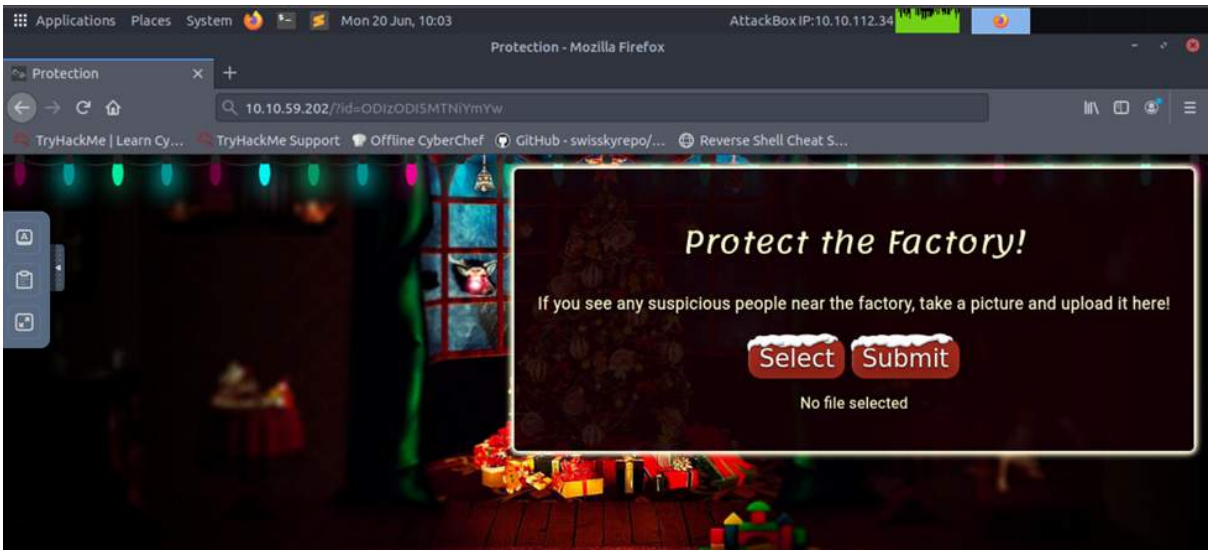
View the page source and found that it only accepts jpeg, jpg and png.



Question 3

Answer: /uploads/

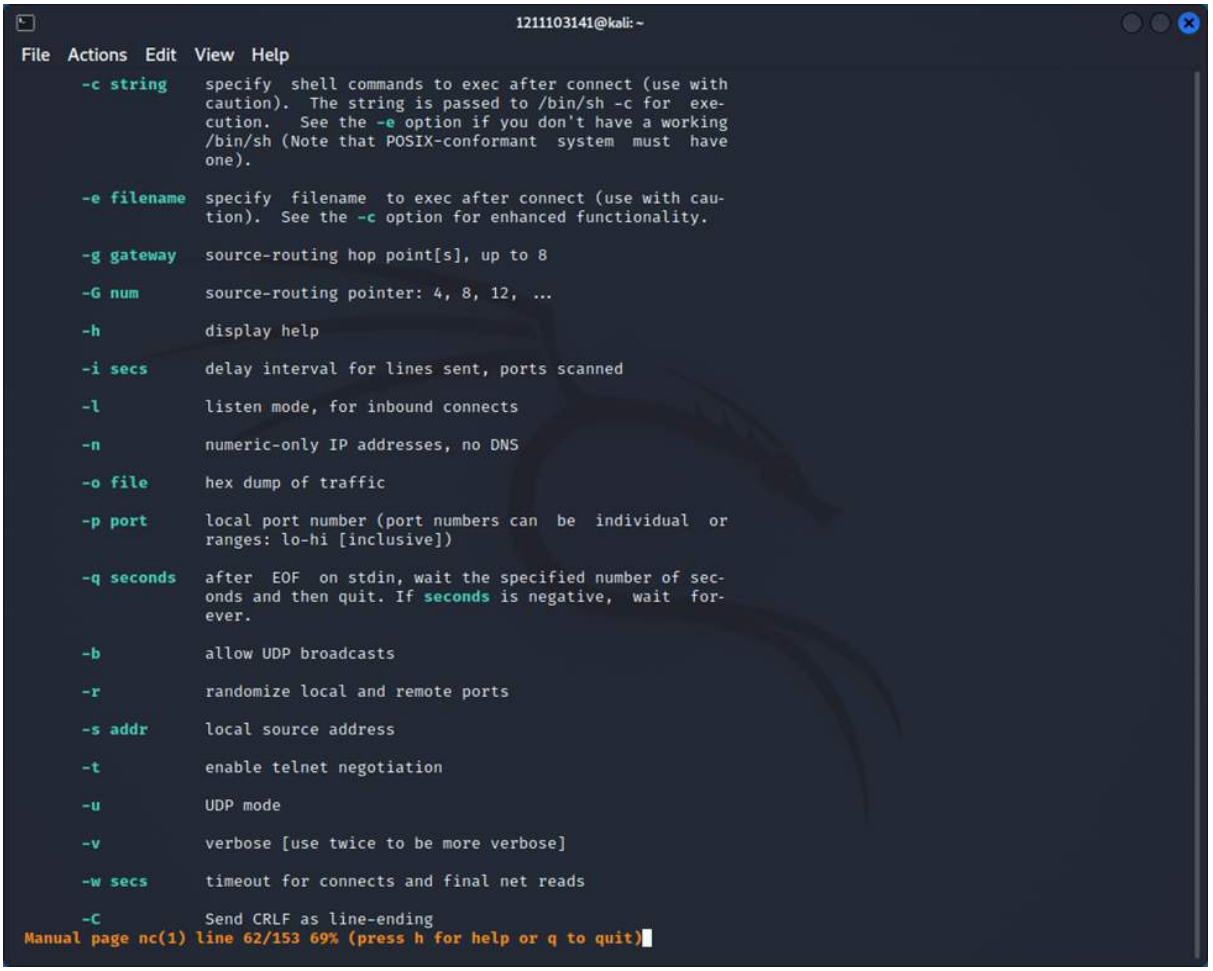
Upload the reverse shell created into the current url. Then, check at the url 10.10.59.202/uploads/. We got the directory by guess and error.



Question 4

Answer: First row 'l', Second row 'n', Third row 'p', Fourth row 'v'

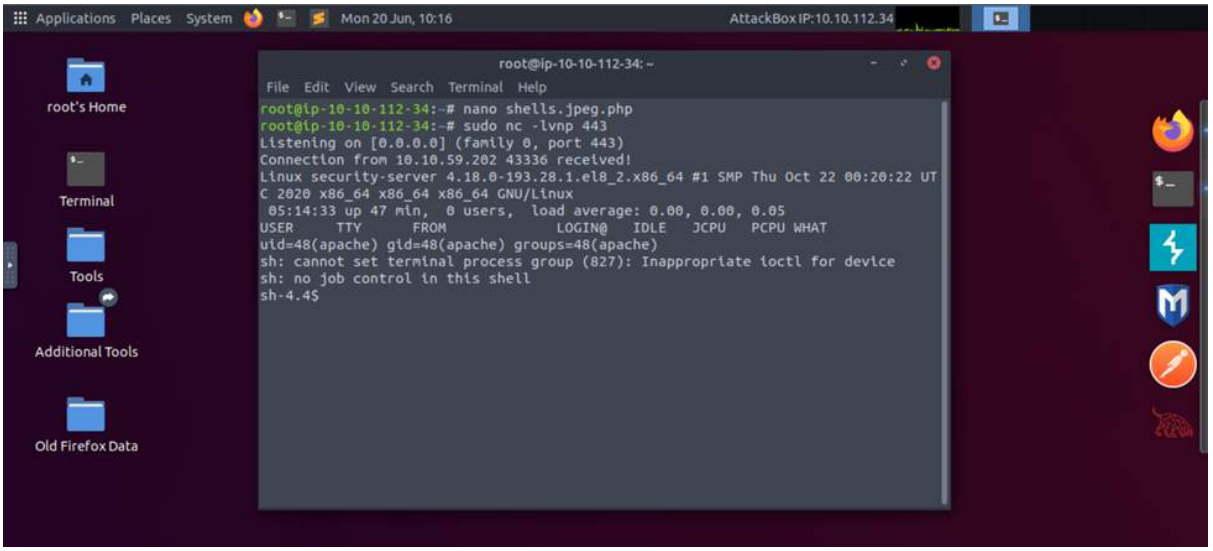
Type "man nc" on shell and read to get the answer.



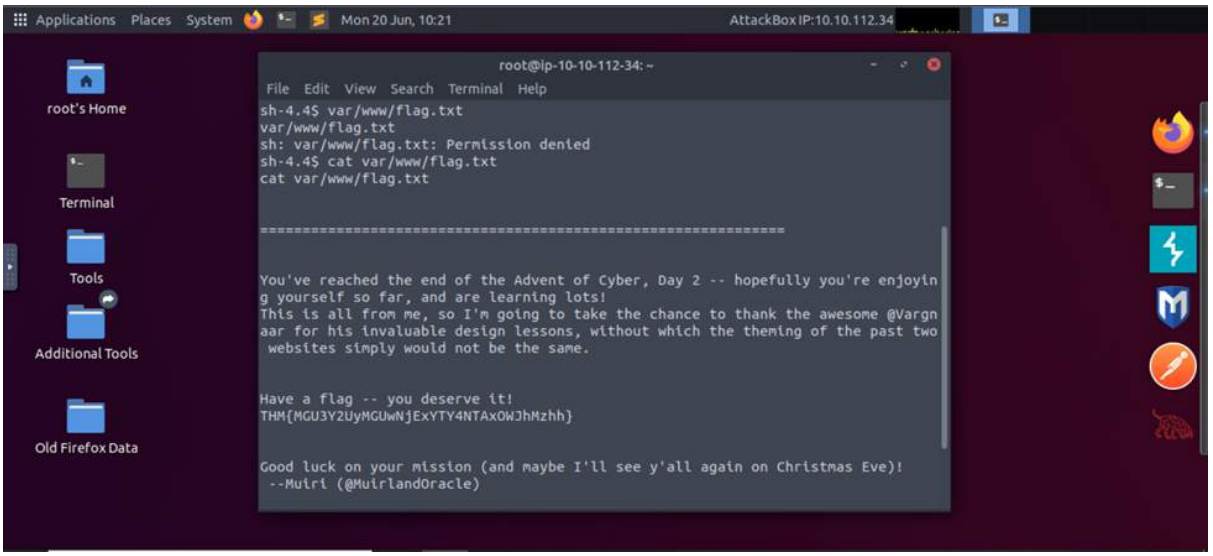
Question 5

Answer: THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

Make a listener at shell. Press the shells.jpeg.php at the /uploads/.



Type cat var/www/flag.txt at the listener and the flag is displayed.



Thought Process/Methodology:

We opened the IP address given and added the id given to access the upload page as instructed by the main page. By looking at the page source, we found that they only accept image format files which are .jpeg, .jpg, and .png. So, we made a reverse shell that can bypass the filter by adding .jpeg.php at the end of the name. We uploaded it and checked it via /uploads/ whether it's uploaded or not. Lastly, we made a listener and activated it by using the shell, thus got the flag.

Day 3: Web Exploration - Christmas Chaos

Tools used: THM Attackbox, Kali Linux

Solution/Walkthrough:

Question 1

Answer: Mirai

You can find the answer from the default 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.

Question 2

Answer: 250

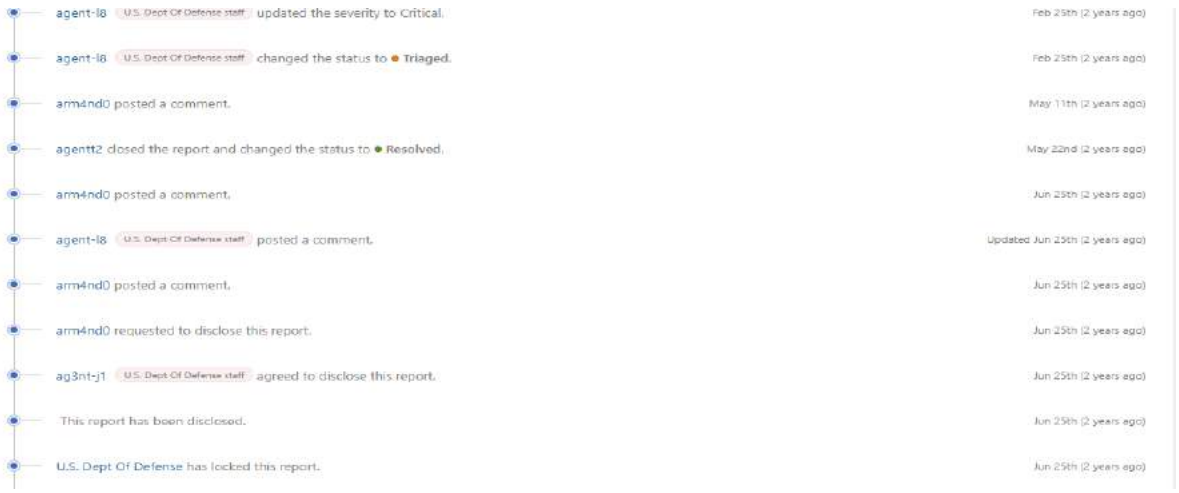
Click the link given in Tryhackme

ak1t4 posted a comment.	Updated Jan 7th (5 years ago)
verona posted a comment.	Jan 10th (5 years ago)
verona changed the status to ● **Triaged**.	Jan 10th (5 years ago)
verona closed the report and changed the status to ● **Resolved**.	Jan 10th (5 years ago)
verona **reopened** this report.	Jan 10th (5 years ago)
ak1t4 posted a comment.	Feb 19th (5 years ago)
siren closed the report and changed the status to ● **Resolved**.	Feb 21st (5 years ago)
Starbucks rewarded ak1t4 with a **$250** bounty.	Feb 21st (5 years ago)
ak1t4 posted a comment.	Updated Feb 21st (5 years ago)
ak1t4 requested to disclose this report.	Feb 21st (5 years ago)
siren changed the report title.	Mar 1st (5 years ago)
siren agreed to disclose this report.	Mar 1st (5 years ago)
This report has been disclosed.	Mar 1st (5 years ago)
overice changed the scope.	Nov 22nd (4 years ago)

Question 3

Answer: ag3nt-j1

Click the link given in Tryhackme



Question 4

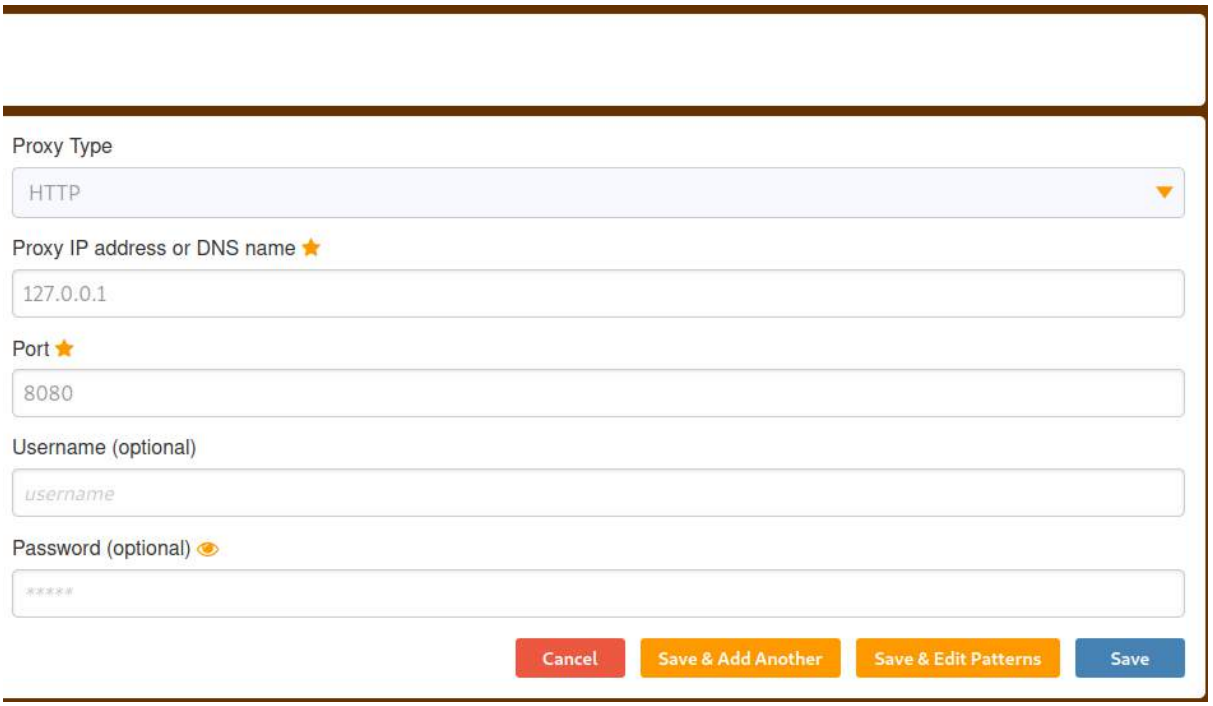
Answer: 8080

Click on foxy proxy extension and click on option and you find the answer

Question 5

Answer: HTTP

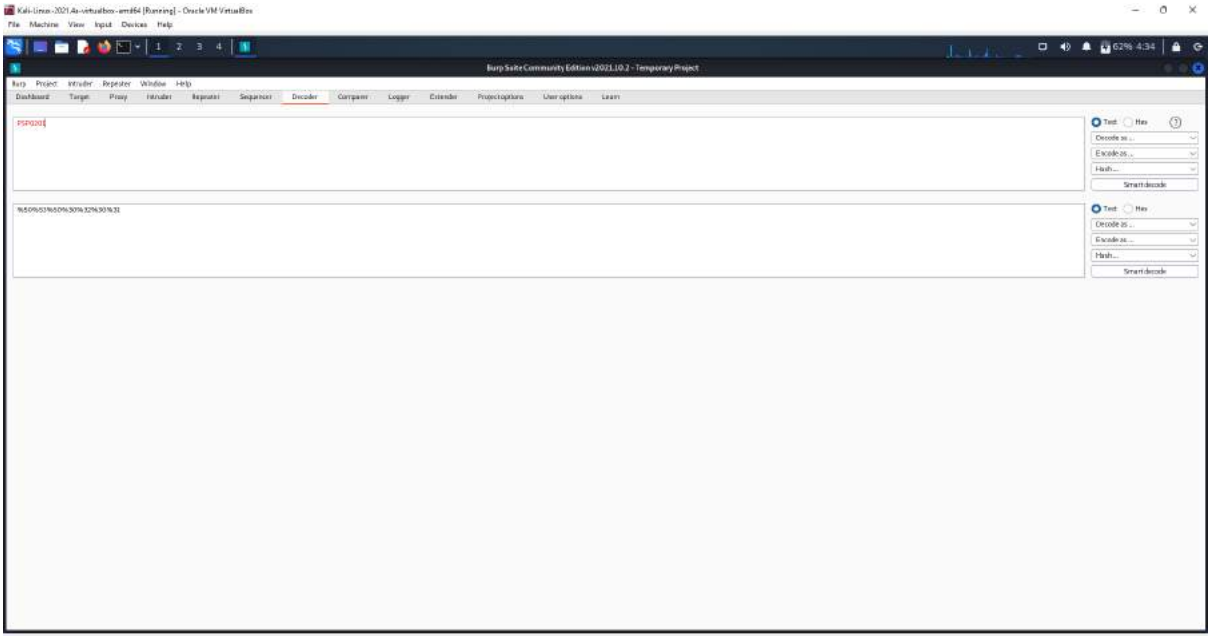
Click on foxy proxy extension and click on option and you find the answer



Question 6

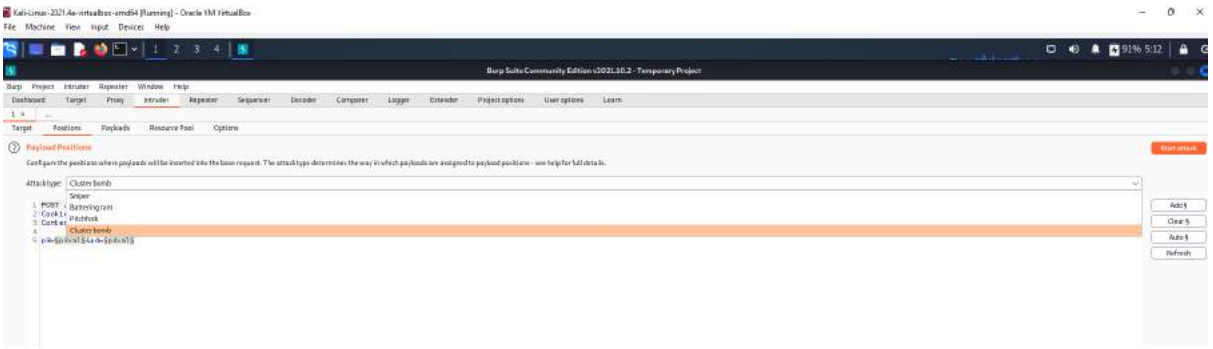
Answer: %50%53%50%30%32%30%31

encode PSP0201 in burpsuite decoder



Question 7

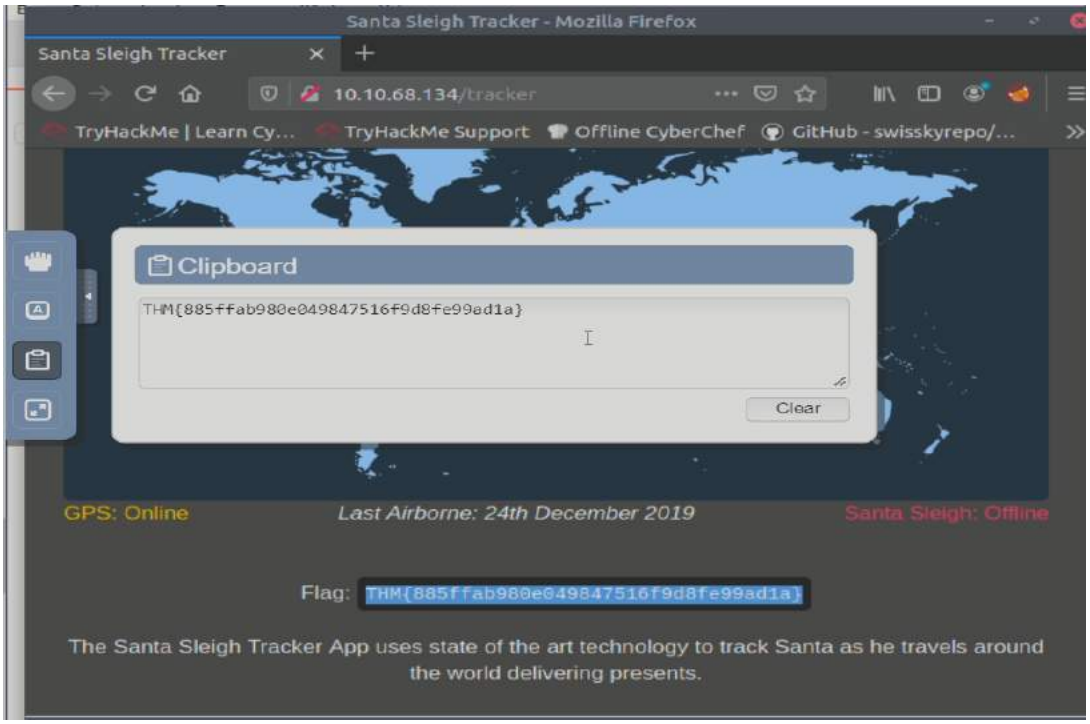
Answer: cluster bomb



Question 8

Answer: THM{885ffab980e049847516f9d8fe99ad1a}

After login into santa sleigh , you will get the THM flag



Thought Process/Methodology:

First open the IP Address given and it will direct to the website. Once this has loaded, you want to "Intercept" your traffic by proxying it through the BurpSuite, which will then forward the request to the intended destination. This will give the ability to analyse and modify your browsers traffic.After that send the generic login from proxy to intruder and select the cluster bomb to iterates through each payloads sets in turn, so every combination of each set is tested. All incorrect logins will have the same status or length, if a combination is correct it will be different.

Day 4: Web Exploration -Santa's watching

Tools used: Kali Linux

Solution/Walkthrough:

Question 1

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

Use the notes below to understand the arrangement of the address.

Let's bring this together and demonstrate some of these options. Let's say we wanted to fuzz an application on *http://shibes.thm/login.php* to find the correct credentials to the login form. After recalling our knowledge from Day 2, we know all about URL parameters! We can take a bit of a guess as to what parameters the login form may be using `username` and `password`, right? Worth a try! Our wfuzz command would look like so:

```
wfuzz -c -z file,mywordlist.txt -d "username=FUZZ&password=FUZZ" -u http://shibes.thm/login.php
```

Where wfuzz will now iterate through the wordlist we provided and replace the "FUZZ" values specified in the "**username**" and "**password**" parameters.

Question 2

Answer: site-log.php

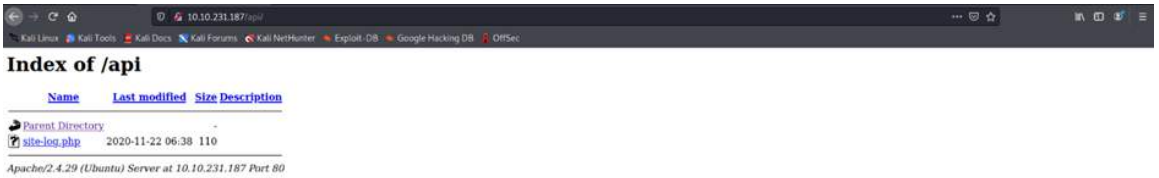
Use gobuster to bruteforce the webpage, use big,txt for the wordlist.

(https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/big.txt) – if you don't have big.txt.

```
kali@kali: ~
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://10.10.231.187/ -w /usr/share/wordlists/dirb/big.tx
t
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.231.187/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/06/22 03:47:26 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://10.10.231.187/api
/]
Progress: 2448 / 20470 (11.96%)
Progress: 2468 / 20470 (12.06%)
Progress: 2488 / 20470 (12.15%)
Progress: 2498 / 20470 (12.20%)
```

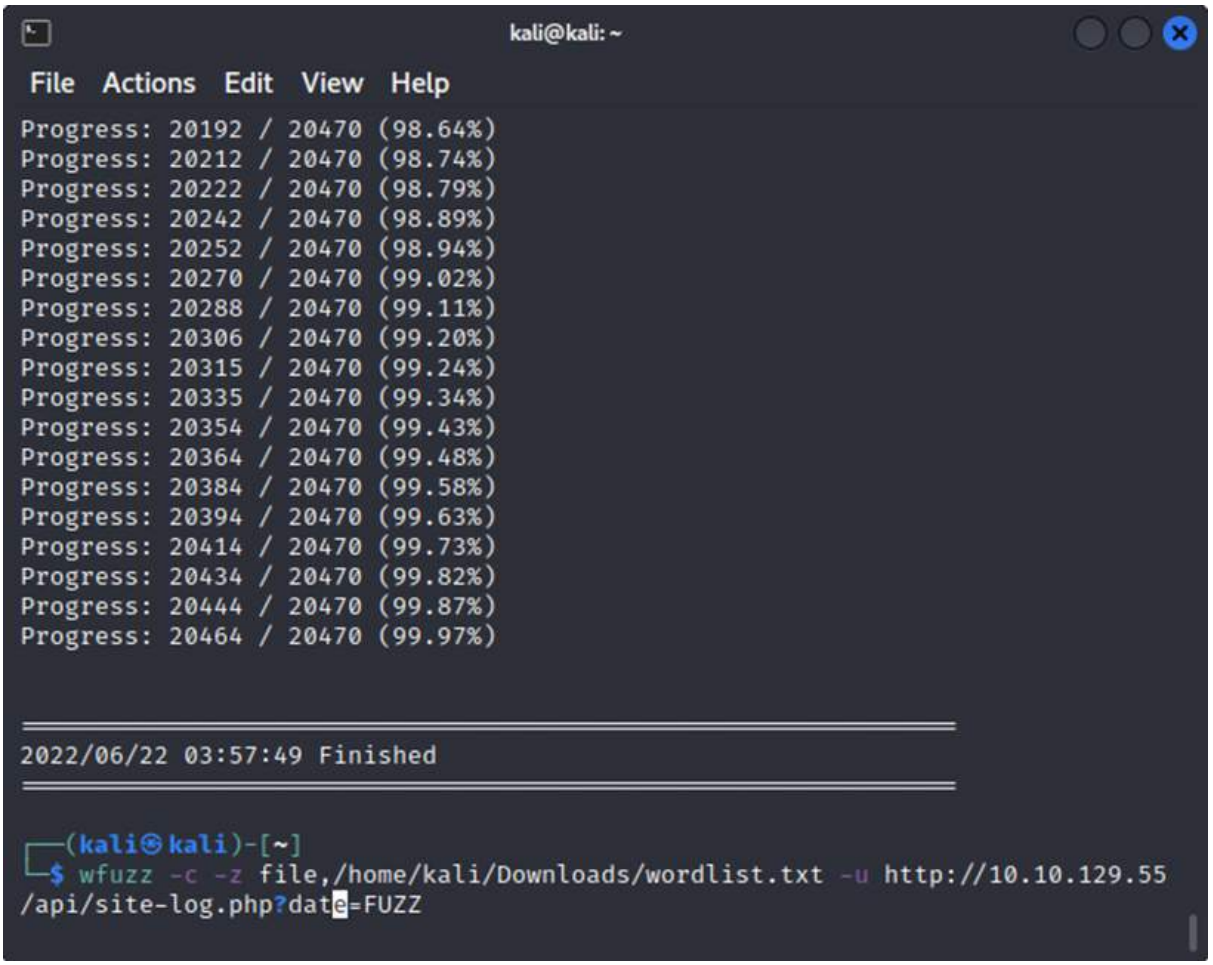
After that you will get the address to the api.



Question 3

Answer: THM{D4t3_AP1}

If you open the api, there is nothing because it has been changed. Use wfuzz to find which date still has the webpage.



The date highlighted is the correct one because it still has some content in it (look at the Ch).

```
kali@kali: ~
File  Actions  Edit  View  Help
000000018:   200        0 L      0 W        0 Ch        "20201117"
000000017:   200        0 L      0 W        0 Ch        "20201116"
000000016:   200        0 L      0 W        0 Ch        "20201115"
000000015:   200        0 L      0 W        0 Ch        "20201114"
000000012:   200        0 L      0 W        0 Ch        "20201111"
000000020:   200        0 L      0 W        0 Ch        "20201119"
000000008:   200        0 L      0 W        0 Ch        "20201107"
000000014:   200        0 L      0 W        0 Ch        "20201113"
000000011:   200        0 L      0 W        0 Ch        "20201110"
000000010:   200        0 L      0 W        0 Ch        "20201109"
000000032:   200        0 L      0 W        0 Ch        "20201201"
000000022:   200        0 L      0 W        0 Ch        "20201121"
000000026:   200        0 L      1 W        13 Ch       "20201125"
000000033:   200        0 L      0 W        0 Ch        "20201202"
000000028:   200        0 L      0 W        0 Ch        "20201127"
000000025:   200        0 L      0 W        0 Ch        "20201124"
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"
000000023:   200        0 L      0 W        0 Ch        "20201122"
000000021:   200        0 L      0 W        0 Ch        "20201120"
000000024:   200        0 L      0 W        0 Ch        "20201123"
000000034:   200        0 L      0 W        0 Ch        "20201203"
000000044:   200        0 L      0 W        0 Ch        "20201213"
000000042:   200        0 L      0 W        0 Ch        "20201211"
000000036:   200        0 L      0 W        0 Ch        "20201205"
```

Search the address, and get the flag.

<u>Question 4</u>

Answer:printer,filename

Read the link given to learn more about wfuzz options.

(https://manpages.debian.org/buster/wfuzz/wfuzz.1.en.html) – help file

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

Methodology:

First, we use gobuster to get the api address as our main webpage has broken. By bruteforcing it, we acquire the address but the content also already has been erased. So, we use wfuzz, bruteforcing it again to get the date when there is still some content in the page. Add the date to the address and we get the flag.

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

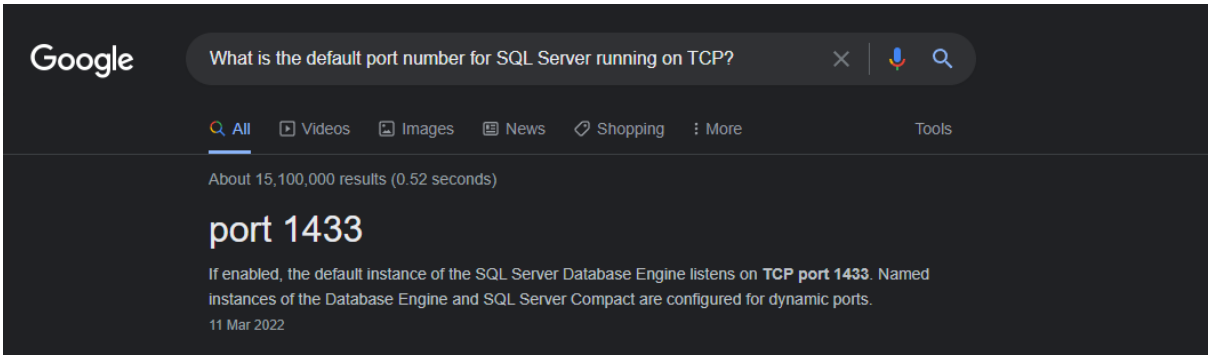
Tools used: Kali Linux

Solution/Walkthrough:

Question 1

Answer: 1433

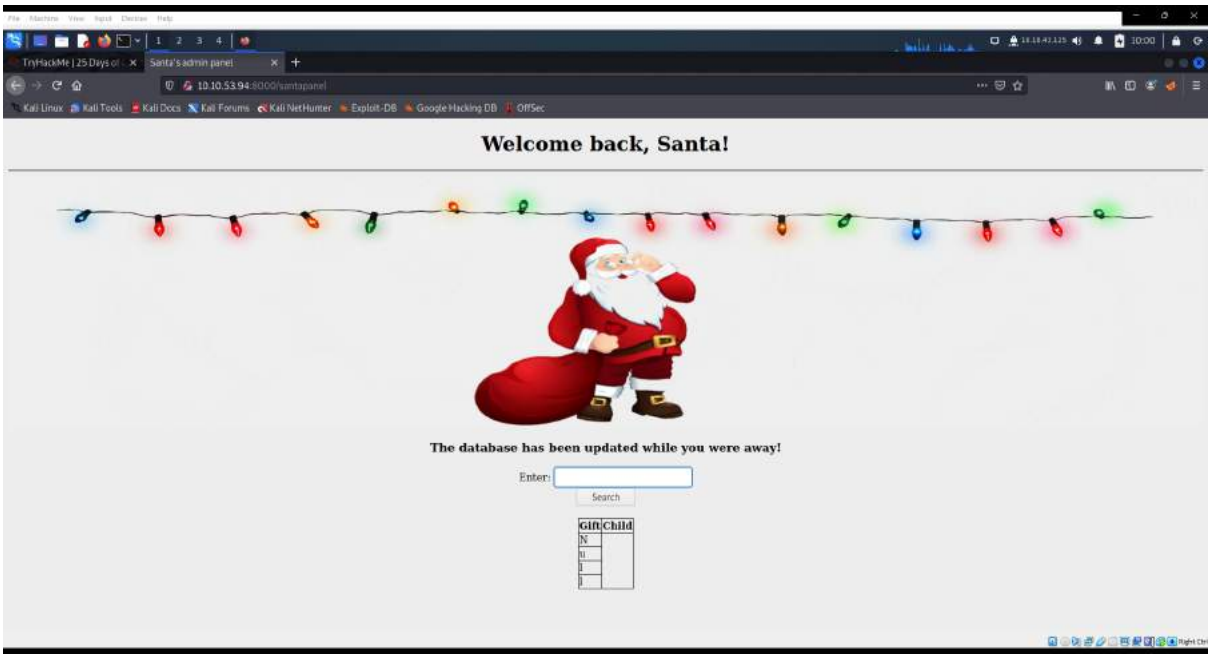
Search default port number at google and you will find the answer



Question 2

Answer: /santapanel

Then go to santa secret login panel by adding



Question 3

Answer: sqlite

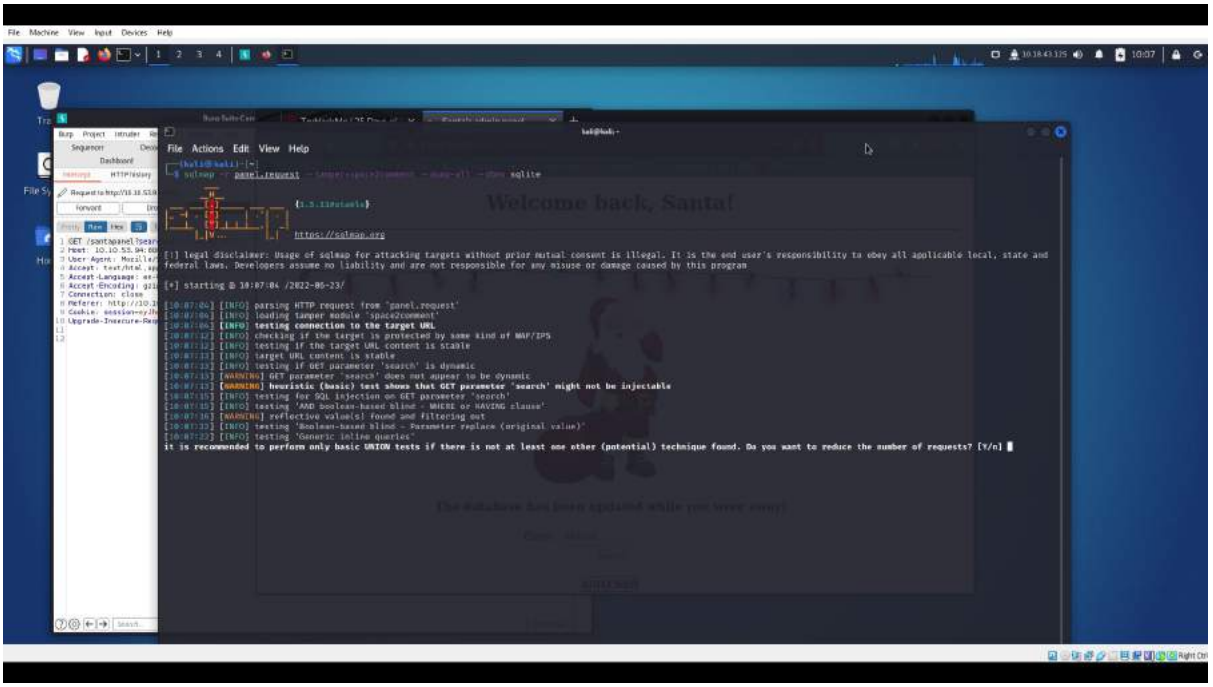
read the santas TODO in TryHackMe to find the answer

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

Question 4

Answer: 22

Open the terminal and command using SQL that we save



after that it will display the hidden table and entry of gift database

```
[07:12:00] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.95.63/dump/SQLite_masterdb/hidden_table.csv'
[07:12:00] [INFO] fetching columns for table 'sequels'
[07:12:00] [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

Answer: 8

after command using sqlmap , you can get list of table from terminal

```
[07:12:00] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.95.63/dump/SQLite_masterdb/hidden_table.csv'
[07:12:00] [INFO] fetching columns for table 'sequels'
[07:12:00] [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 6

Answer: github ownership

after command using sqlmap , you can get list of table from terminal

```
[07:12:00] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.95.63/dump/SQLite_masterdb/hidden_table.csv'
[07:12:00] [INFO] fetching columns for table 'sequels'
[07:12:00] [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 7

Answer: thmfox{All_I_Want_for_Christmas_Is_You}

scroll down a bit and find the flag from terminal

Question 8

Answer: EhCNSWzzFP6sc7gB

scroll down and you will see the password

Methodology:

We use the SQLmap to penetrate so it can automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. With BurpSuite, we can capture and save login or search information to use with SQLMap. This is done by intercepting a request. We will need to configure our browser to use BurpSuite as a proxy for this request to capture. After that SQLMap will automatically translate the request and exploit the database for us.