PSP0201

WEEK 2

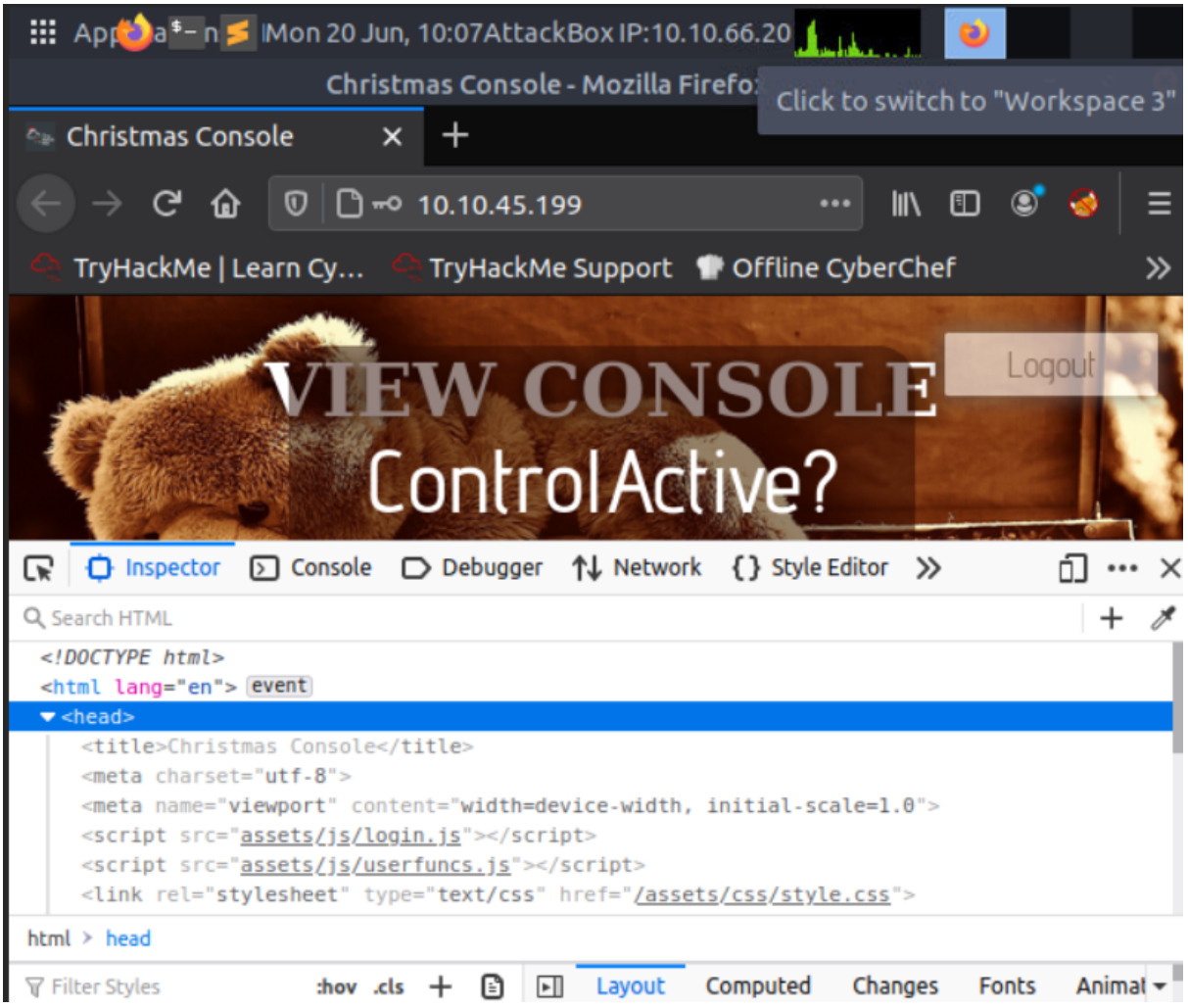
WRITE UP

ID	*NAME*	*ROLE*
1211102582	AMEER IRFAN BIN NORAZIMAN	leader
1211101873	MUHAMMAD NABEEL SHAMIME BIN KHAEROZI	member
1211102269	MUHAMMAD ANIQ SYAHMI BIN SHAHARIL	member
1211101915	NURDINA AISHAH BINTI KASUMA SATRIA	member

Day 1: A Christmas Crisis
Tools used: Attackbox, Firefox

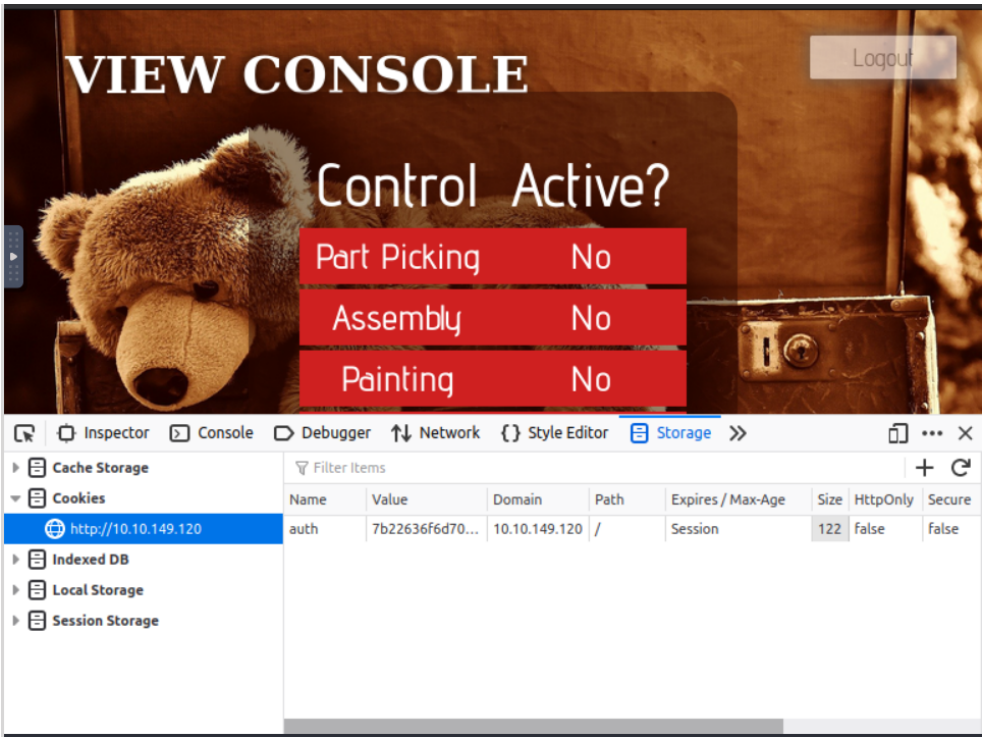
Solution:

Question 1
Inspect and search for the title of the website



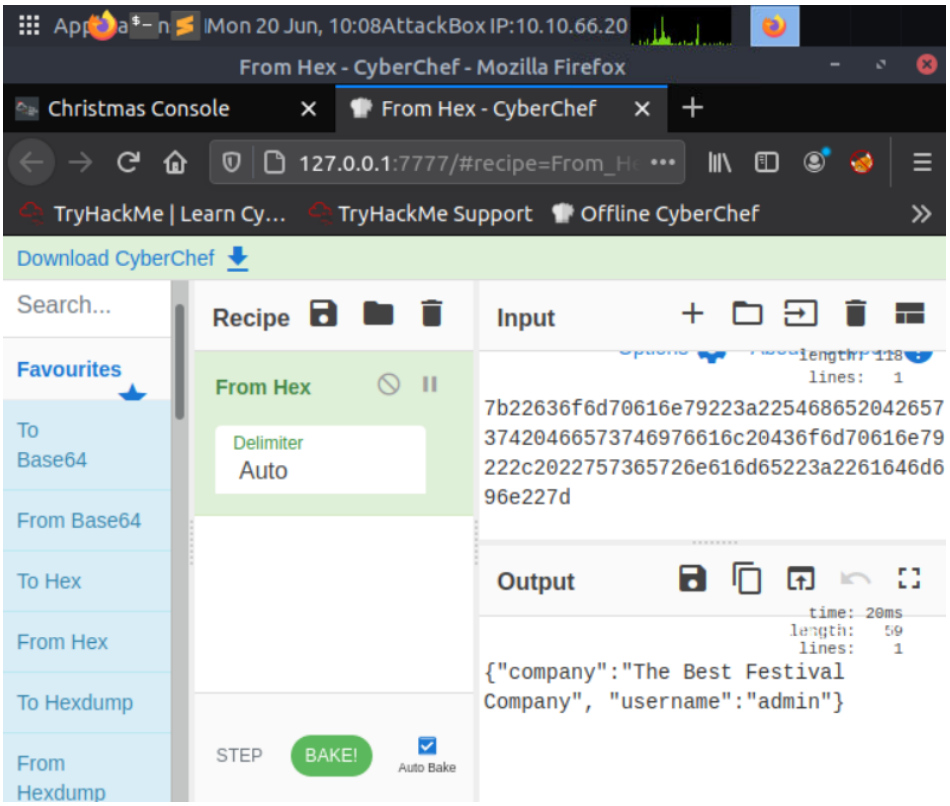
Question 2

The name of the cookie



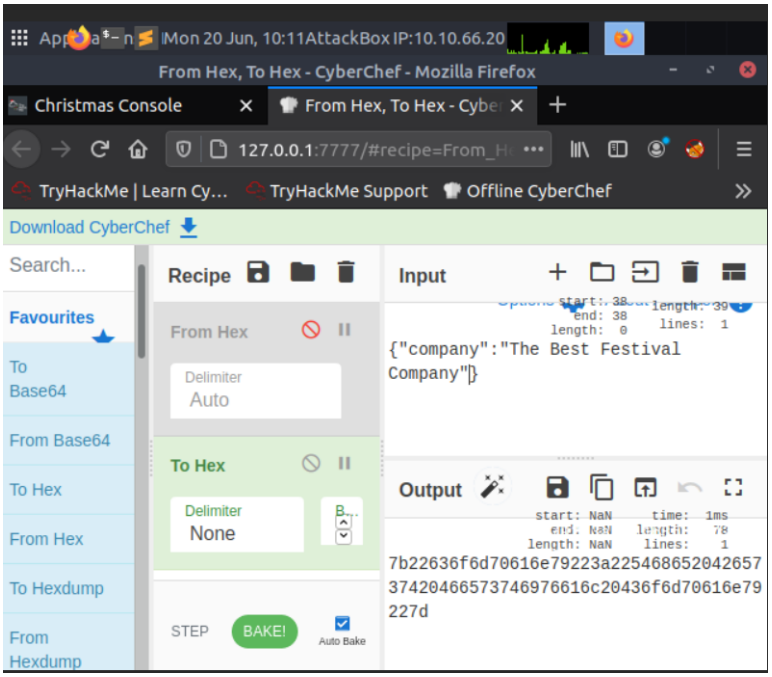
Question 3&4

The value's format and also the format that the data stored in



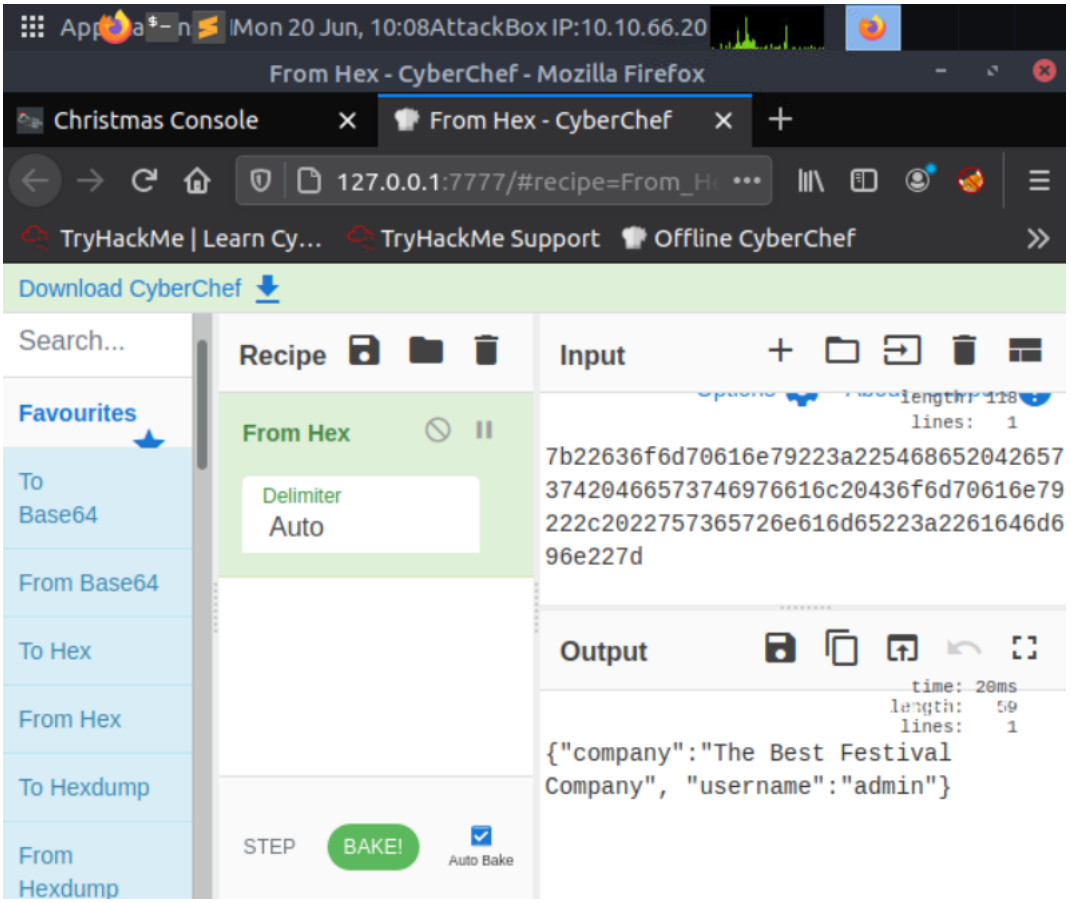
Question 5

The company field's value



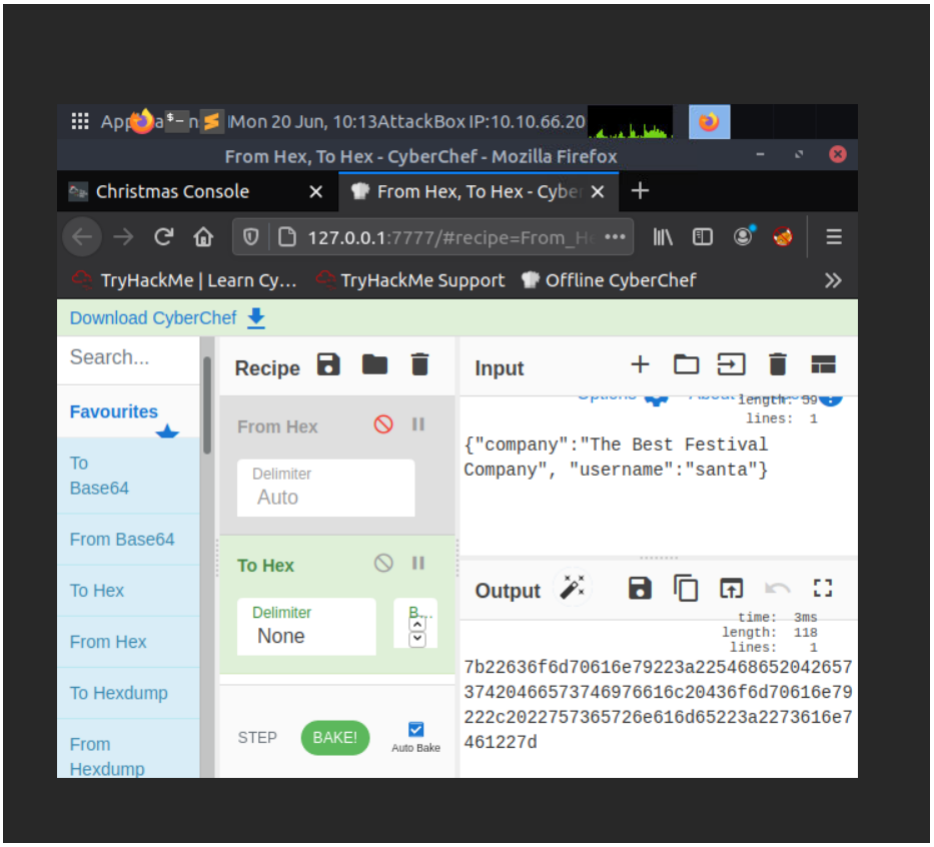
Question 6

The other field found in the cookie



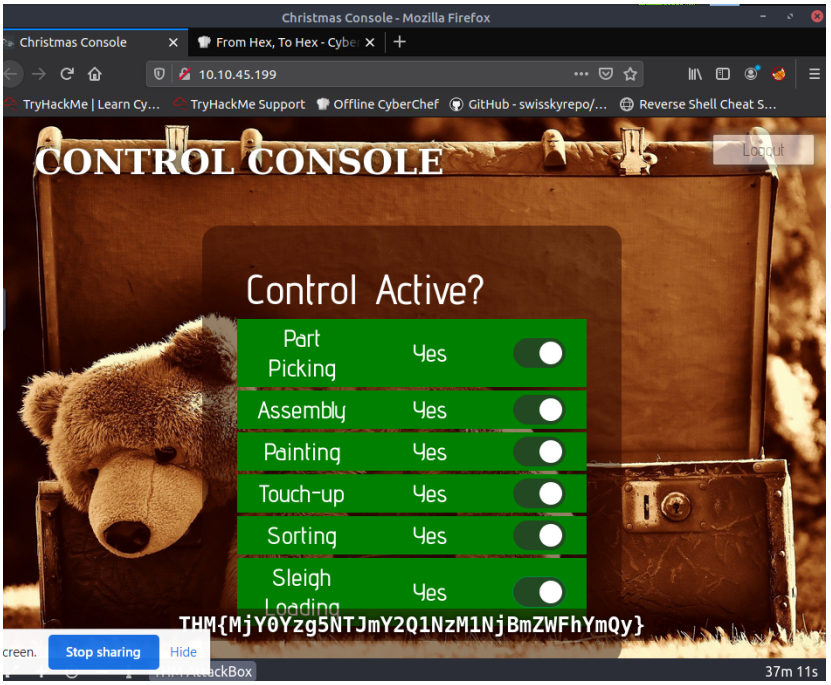
Question 7

The value of Santa's cookie



Question 8

the flag that we received when the line is fully active



The Process :

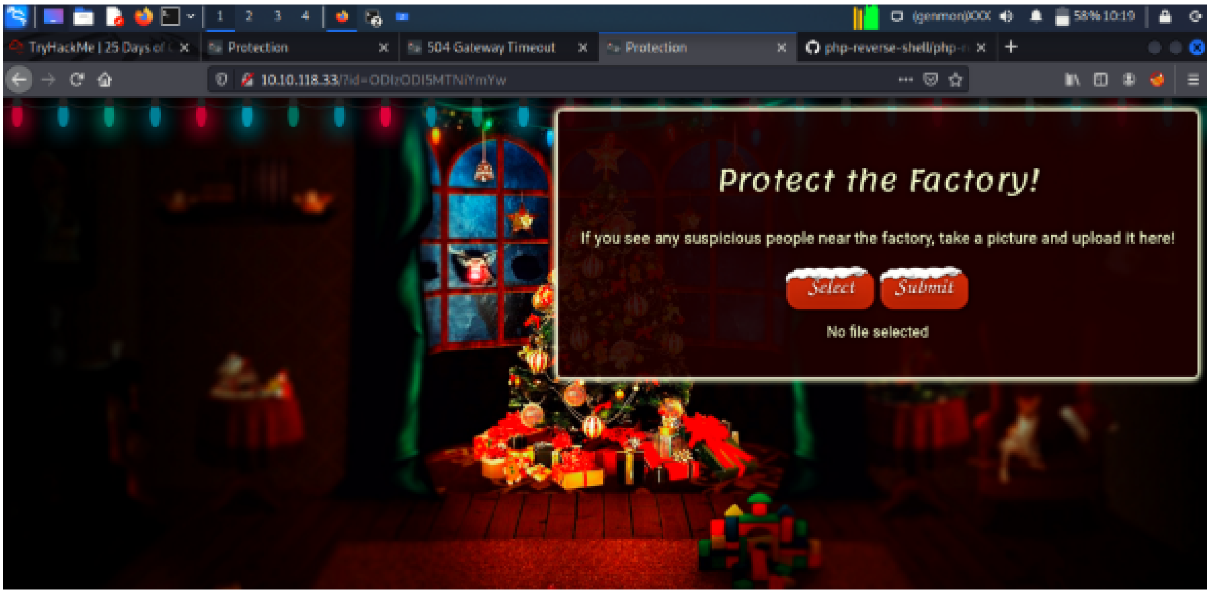
First of all, we inspect the website by pressing **ctrl+shift+i**. By inspecting the website, we can receive the website title by copying and pasting the html title. Then , by also inspecting the website, we are able to receive the cookie's name and also the value of the cookie used for authentication. Just simply go to storage and we can see the name of the cookie, which is **auth**, and also the value of the cookie. We found that the value's format is **base 16 for hexadecimal** due to the numbers ranging from 1-9. Next, we are required to decode the cookie and figure out what format the data is stored in. Therefore, we used the cyberchef to decode the cookie by using the "**from hex" operation**. We found out that the output represents **{"company":"The Best Festival Company", "username":"admin"}.** From what we have learned, the "{ }" and also the double quotation marks represent **Javascript Object Notation(JSON)**. Next, question 5 asked us for the company field's value. So we just simply remove the username from the output to get the company field's value. After that, we change the username from admin to Santa to get the Santa's cookie. From that, we take the cookie and paste it into the value of the website to log in as Santa. Therefore, we got control of the **Control Active** and activated everything to obtain the flag.

Day 2: The Elf Strike Back
Tools used: *Attackbox, Firefox,Kali Linux*

Solution:

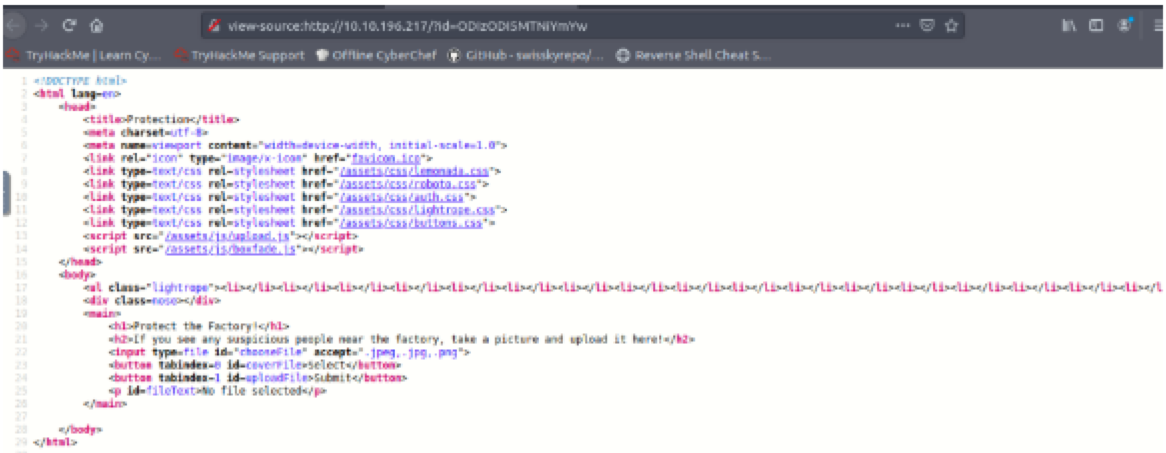
Question 1

Paste the ID number and add the GET parameter to the IP address.



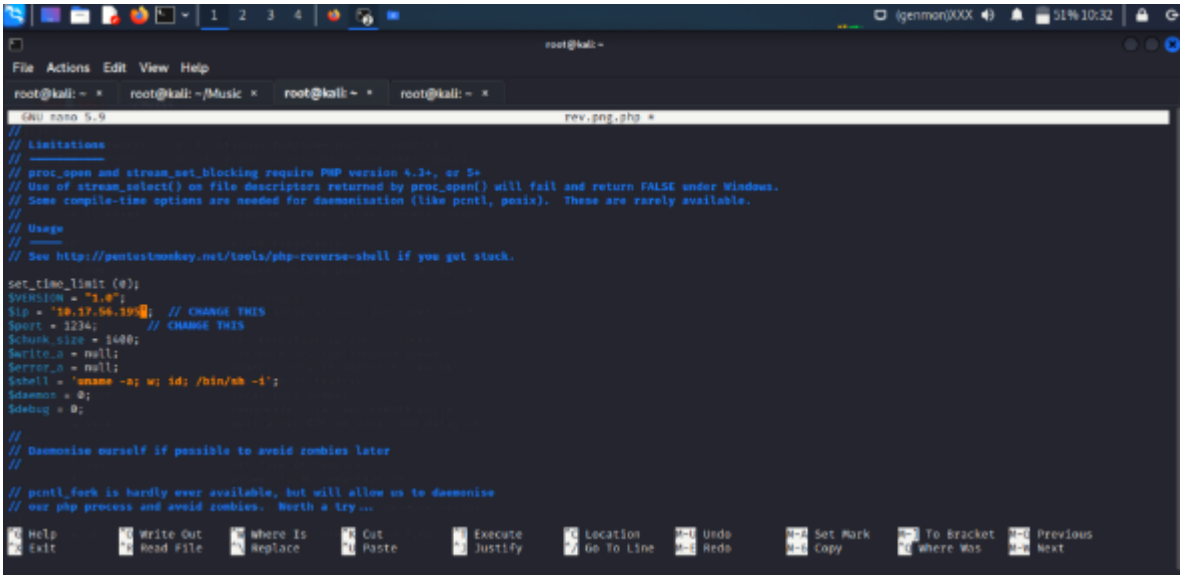
Question 2

The file types that the site accepts are.png,.jpg, and.jpeg, according to the page source.

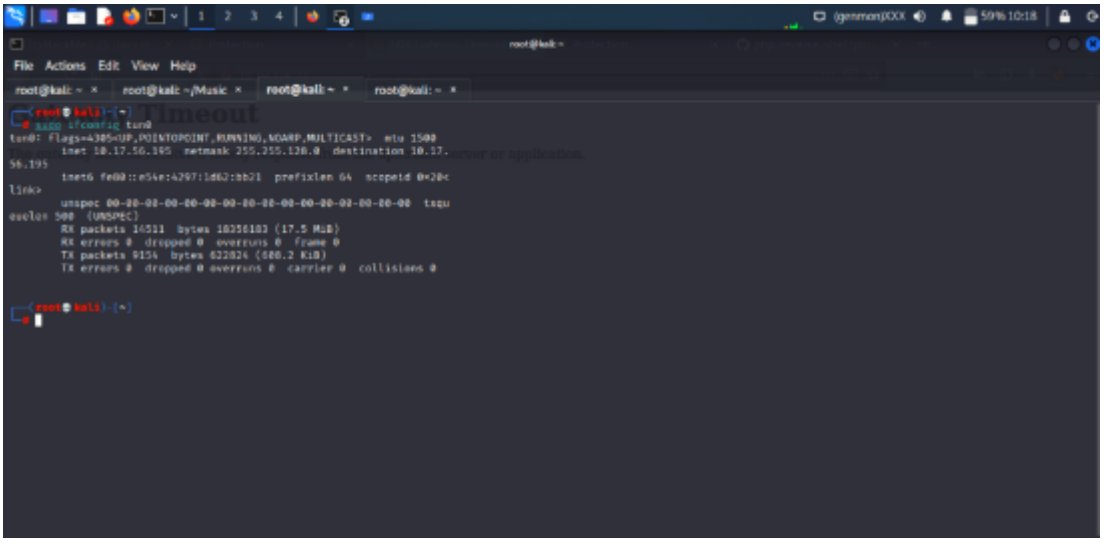


Question 3

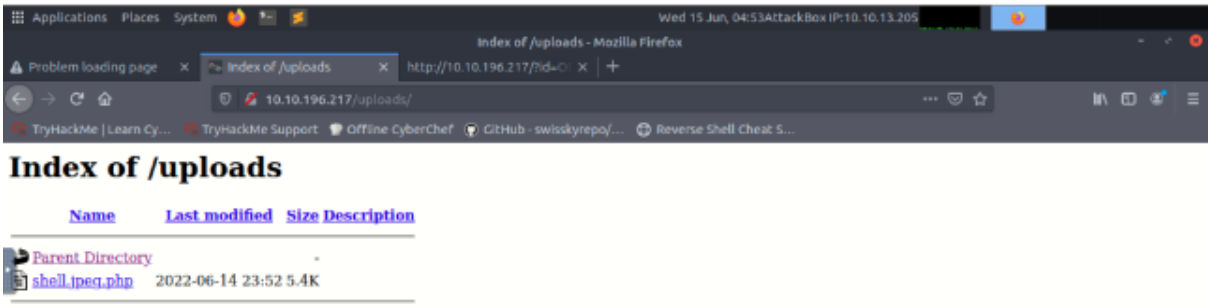
Rename the title and extension of the webshell after copying it to the directory. Use nano to open it. Change the IP address while keeping the same port number.



Run "sudo ifconfig tun0" to find the IP address.

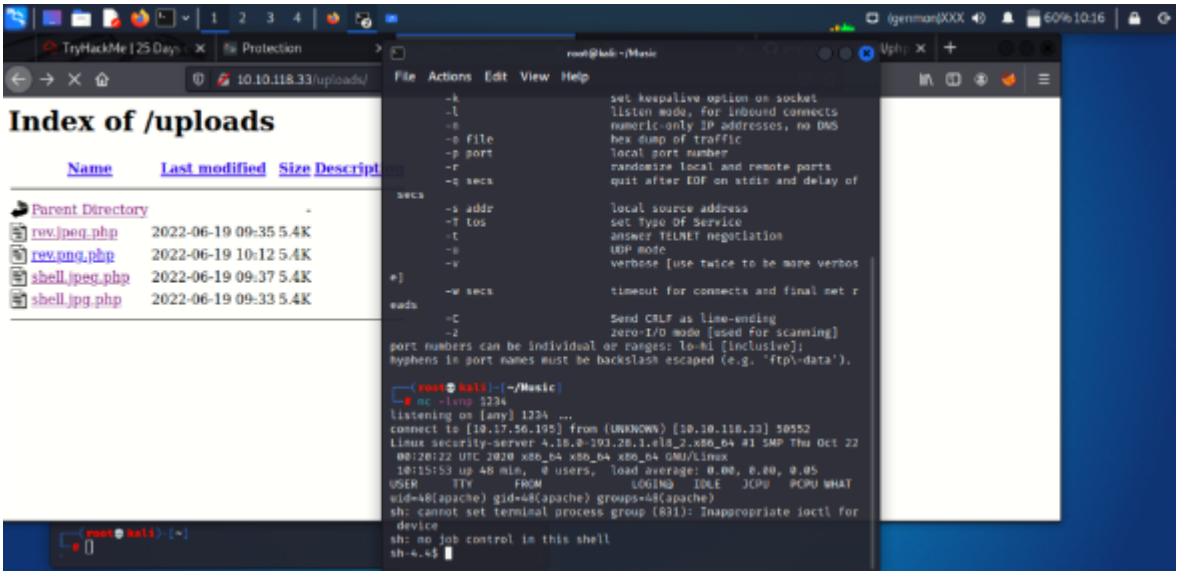


Change the argument to /uploads/ by removing it. The uploaded file is now visible.



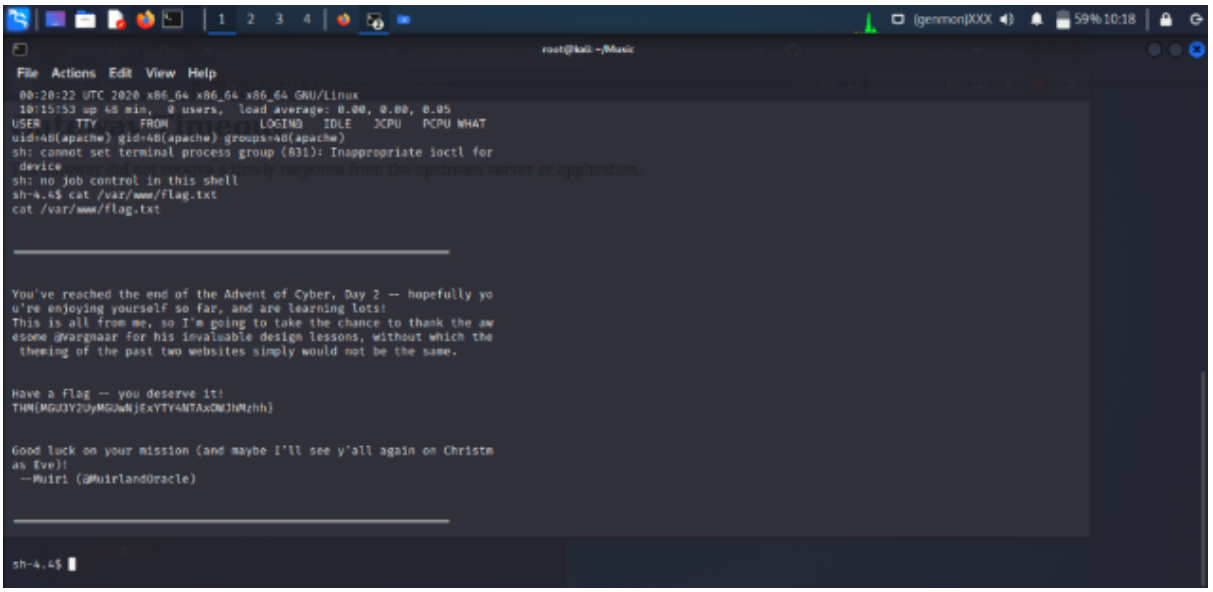
Question 4

Create the listener for the uploaded reverse shell using the 'nc lvnp 1234' command in the next tab. To obtain their shell navigation and connection, upload a reverse shell and then launch the cat listener.



Question 5

The contents of the flag are displayed when you type cat /var/www/flag.txt.



THE PROCESS

To access the upload section, copy the IP address and then add the GET parameter and the ID number. When you right-click on the page to inspect the page source, you can see that the site only accepts files in the.png,.jpg, and.jpeg file types. Then, copy the webshell to the directory and change the file name's extension and title. Scroll down to configure the ip after opening it with nano. Run "sudo ifconfig tun0" to find the new IP address; the port stays the same. Change the argument and id number to /uploads to view the uploaded file. Create a listener for the uploaded reverse shell using the 'nc lvnp 1234' command in the next tab. Start the cat listener after uploading a reverse shell to obtain their shell connection and navigation. Finally, running the command cat /var/www/flag.txt reveals the flag's contents.

Day 3: Christmas Chaos

Tool used: *Attackbox, Kali Linux, FireFox*

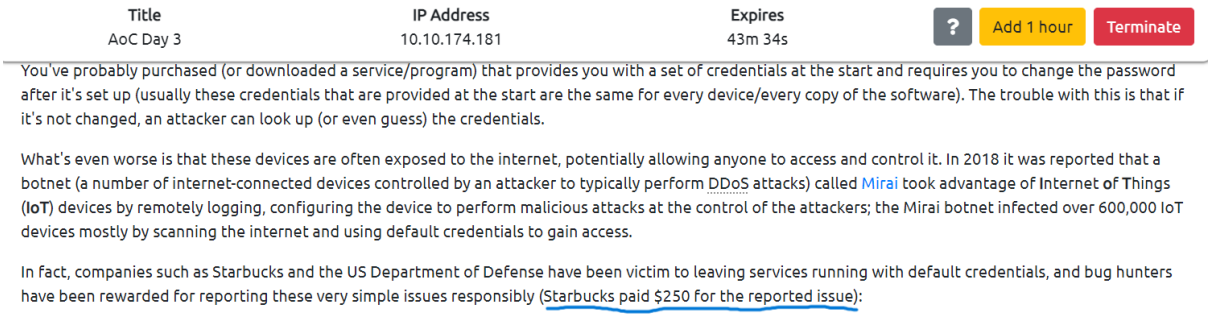
Solution:

Question 1

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.

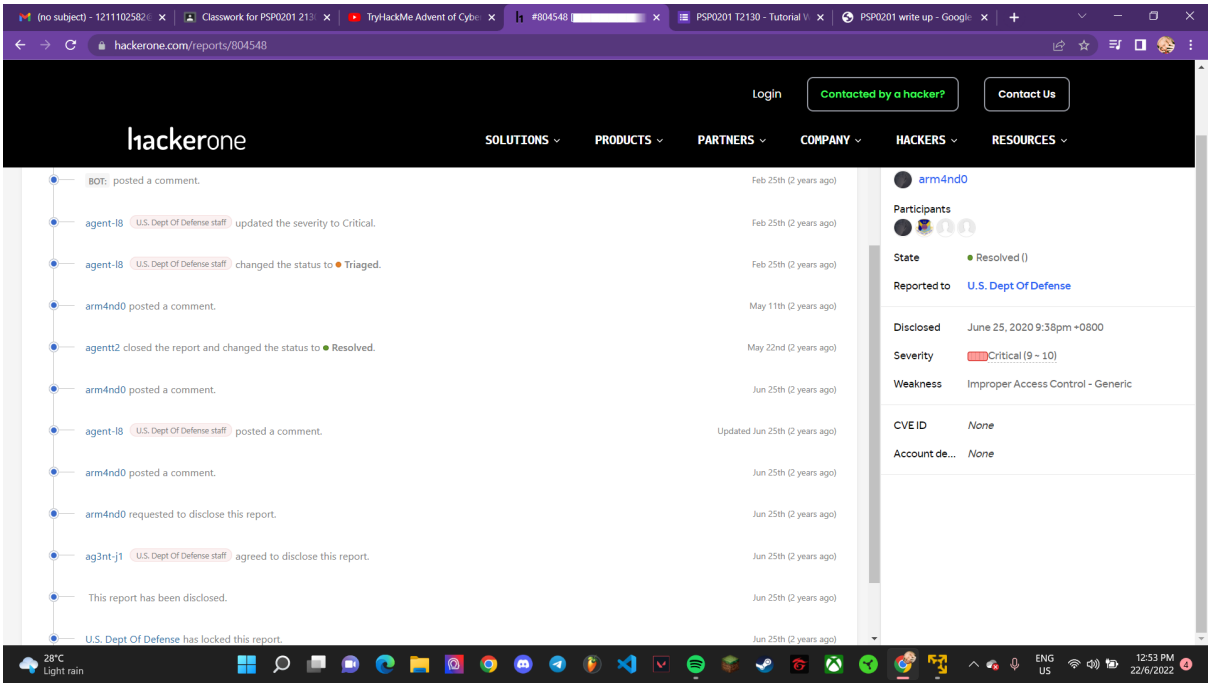
Question 2

Starbucks paid $250 USD for the reported problem, as shown by the * in the bracket.



Question 3

*According to the information provided by Hackerone ID:804548, ***ag3nt-j1*** was the agent who consented to make the report public.



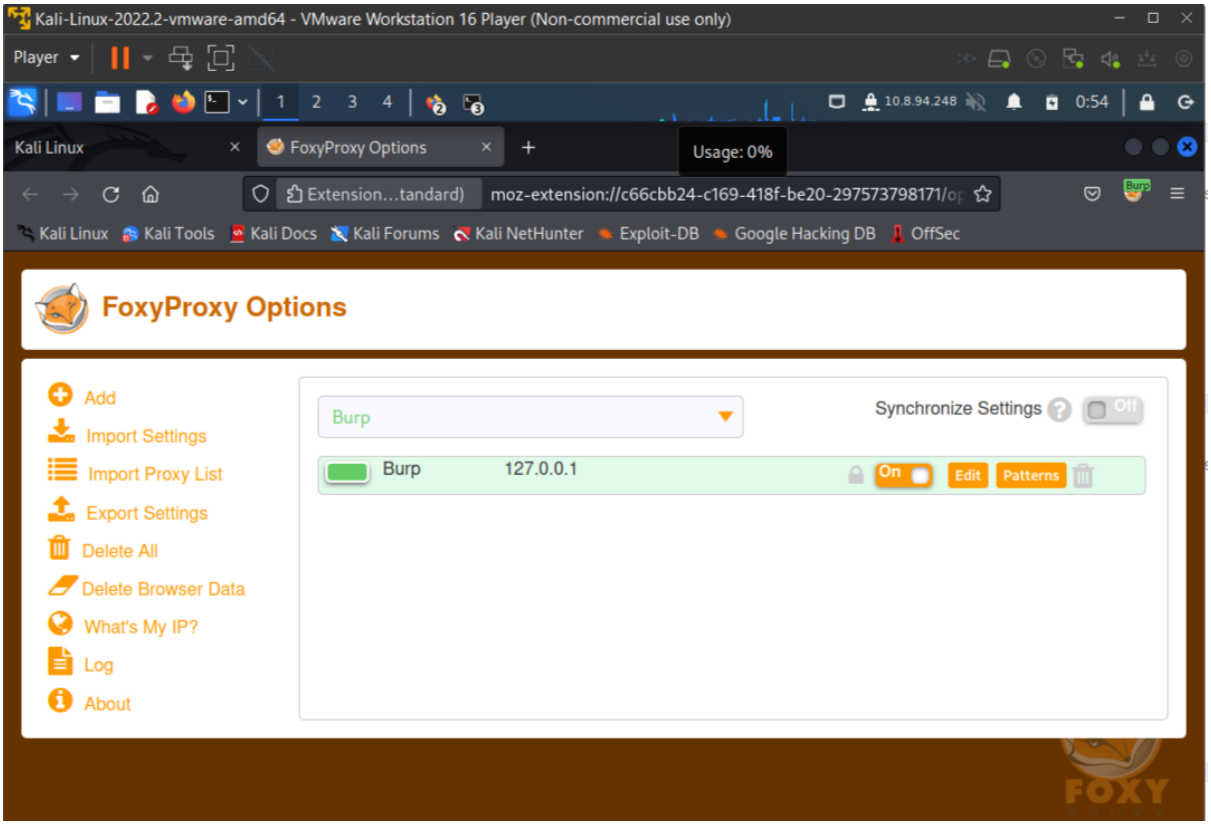
THE PROCESS

We read through the task's description and instructions before starting our Kali. There, the name of the botnet referenced and disclosed in 2018 is revealed. Additionally, we learned the USD equivalent of Starbucks' pay rate. The agent assigned by the Department of Defense was ag3nt-j1, according to the report from Hackerone ID:804548. After entering the target computer, Santa Sleigh Tracker, we were then taken to a sign-in page. To check out the options on FoxyProxy, though, we had to go to the burp room. There, we can find both Burp's port number and proxy type. Using Burp's decoder, we were able to discover the URL encoding for PSP0201. The Santa Sleigh tracker is then reloaded. To verify that the website is truly loaded, we pass the intercept in the Burp. Then, we log in to the website using the credentials admin and ****. The information, along with the username and password we set up on the website, was then obtained using a proxy. To enable the hacker to automate customised online attacks, we sent the hacker all the data that we had obtained. The pre-selected position is then cleared, and the username and password are added as position values. Next, we proceed to the intruder tab. We use the "add $" box to highlight the text and choose the "cluster bomb" assault type. We choose our payload set and add the list in the "Payload Options" section of the "Payloads" page. Then, we include a few standard default usernames like "admin," "root," and "user." We also include a few often used default passwords, like "password," "root," and "12345." Each position list in every combination will

be looped through after we press the "Start Attack" button. We entered the website using the following lists of the login and password, and we will soon receive the third day's flag.

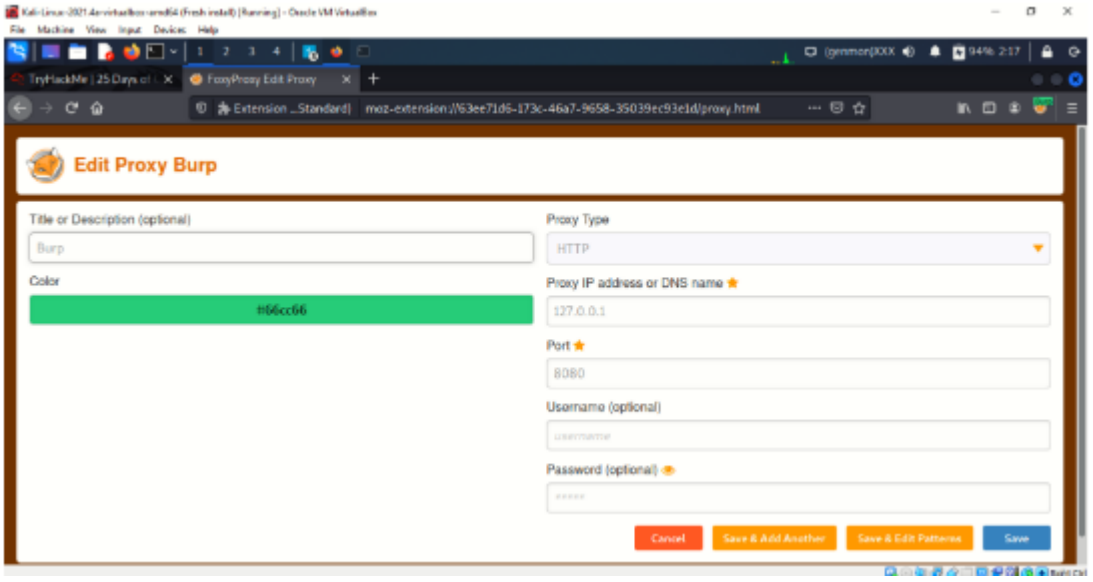
Question 4

When we click the intruder button on the burpsuite, the host and burp port are displayed.



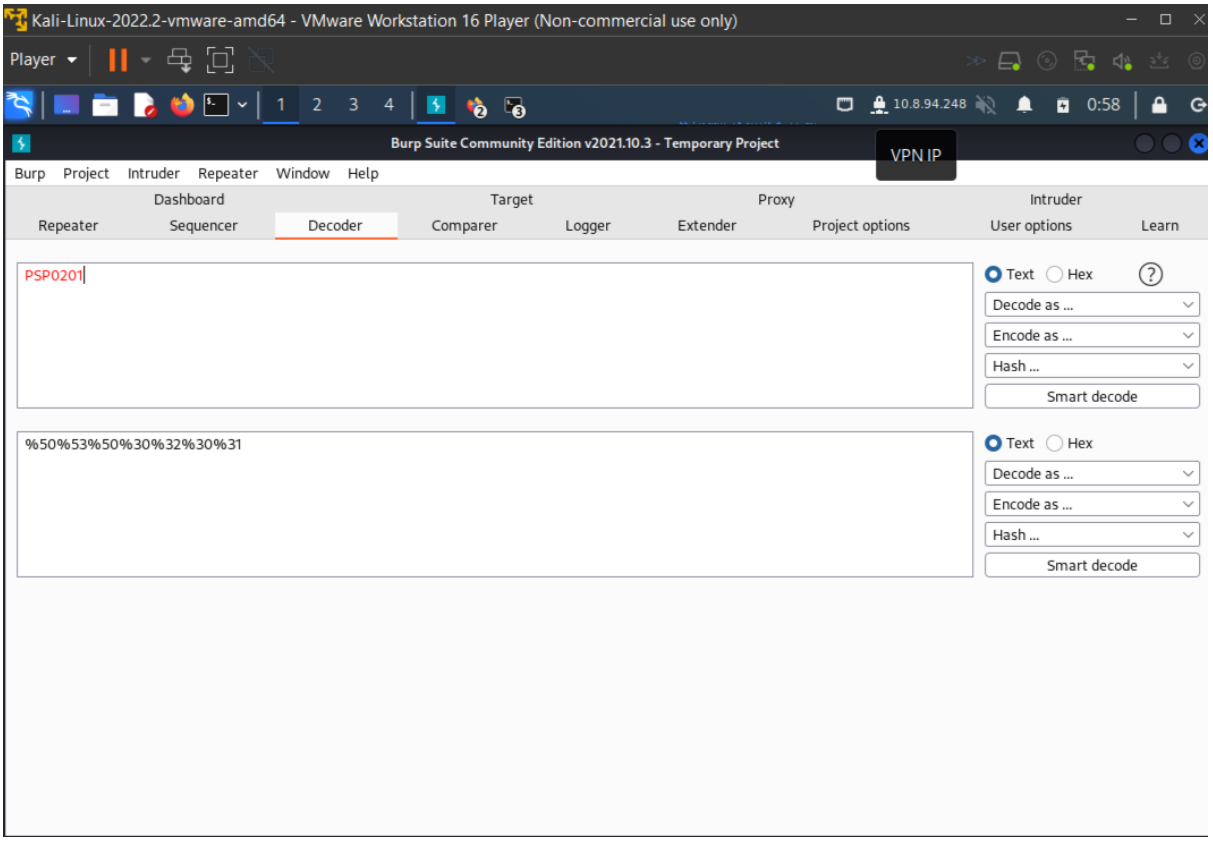
Question 5

Returning to Firefox, we click the FoxyProxy button in the sidebar, followed by the "options" button, which takes us to a different website. When we click the "add" button on the left side of the page, the proxy type is displayed.



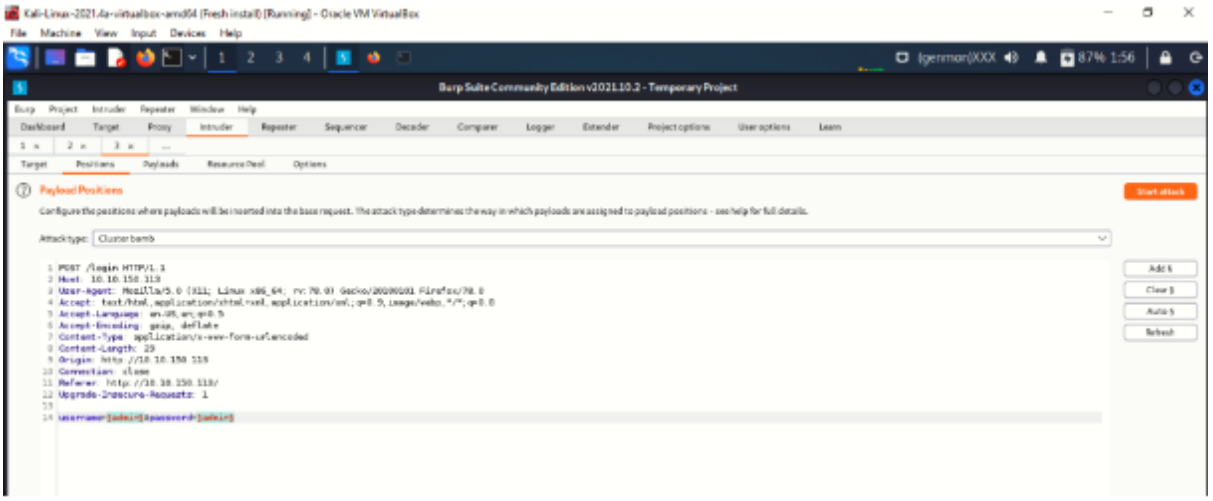
Question 6

We entered PSP0201 through the Burp's decoder to obtain the URL encoding.

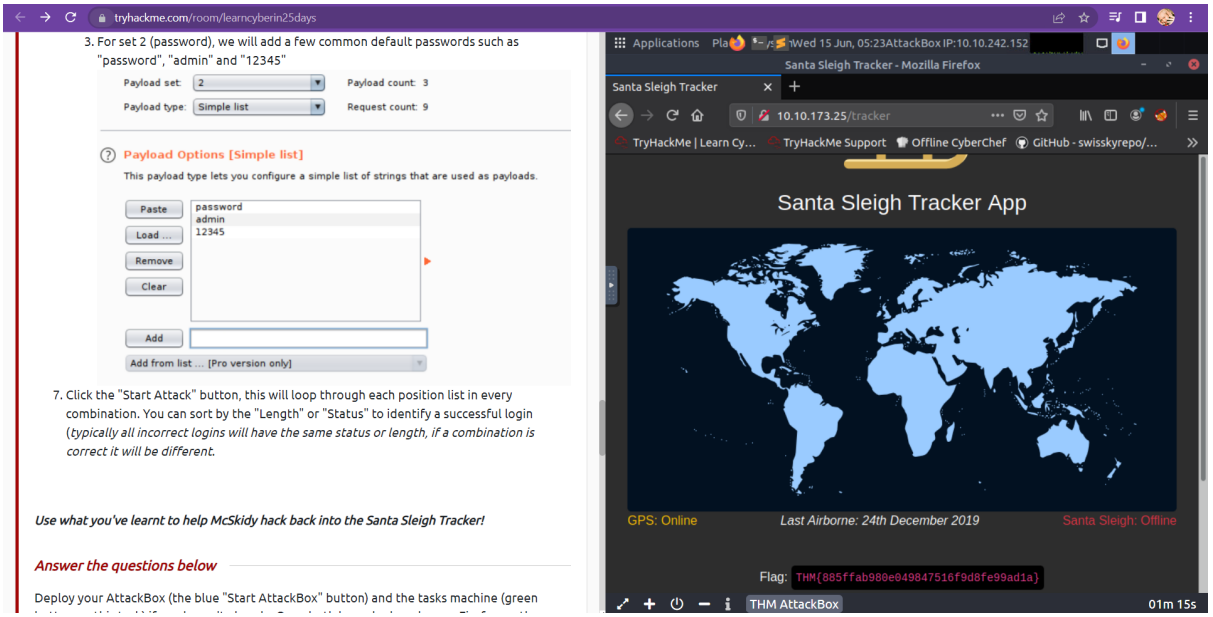


Question 7

Next, we choose the intruder and put buttons on the Burpsuite. The username and password value locations are then added. We clicked the "add $" button after highlighting the username and password. We next choose the cluster bomb attack type from the assault type menu. We choose our payload set and add the list in the "Payload Options' ' section of the "Payloads" page. Then, we include a few standard default usernames like "admin," "root," and "user." We also include a few often used default passwords, like "password," "root," and "12345." Each position list in every combination will be looped through after we press the "Start Attack" button.



Question 8



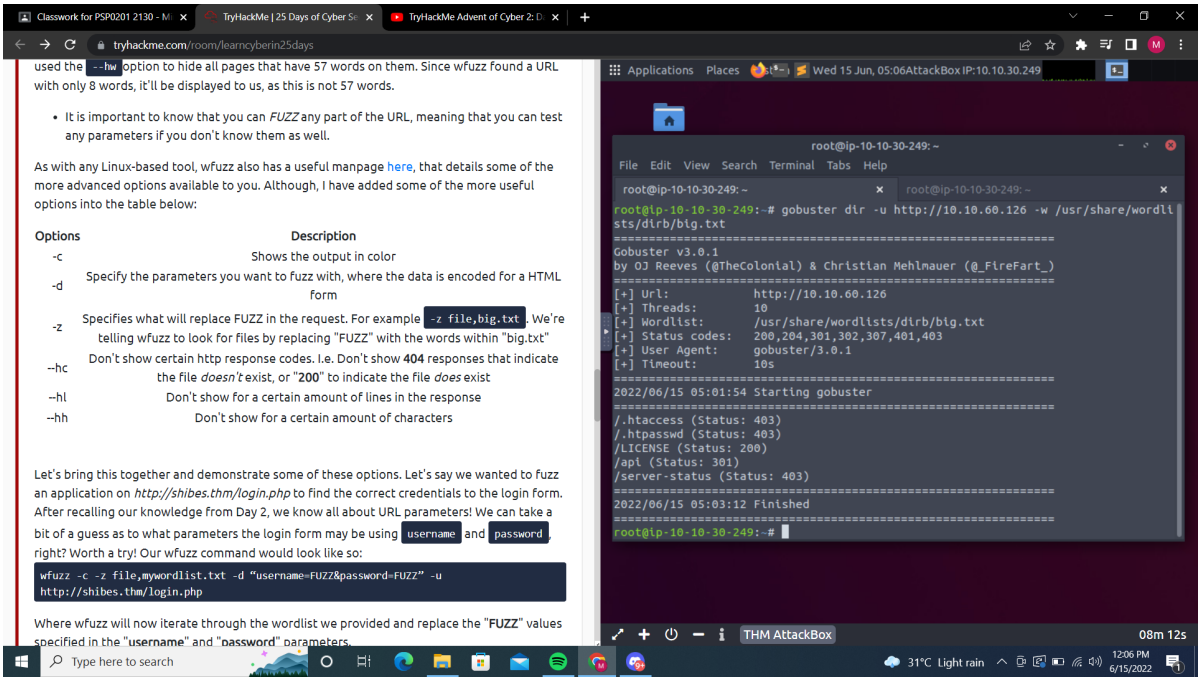
Day 4 : Santa's Watching

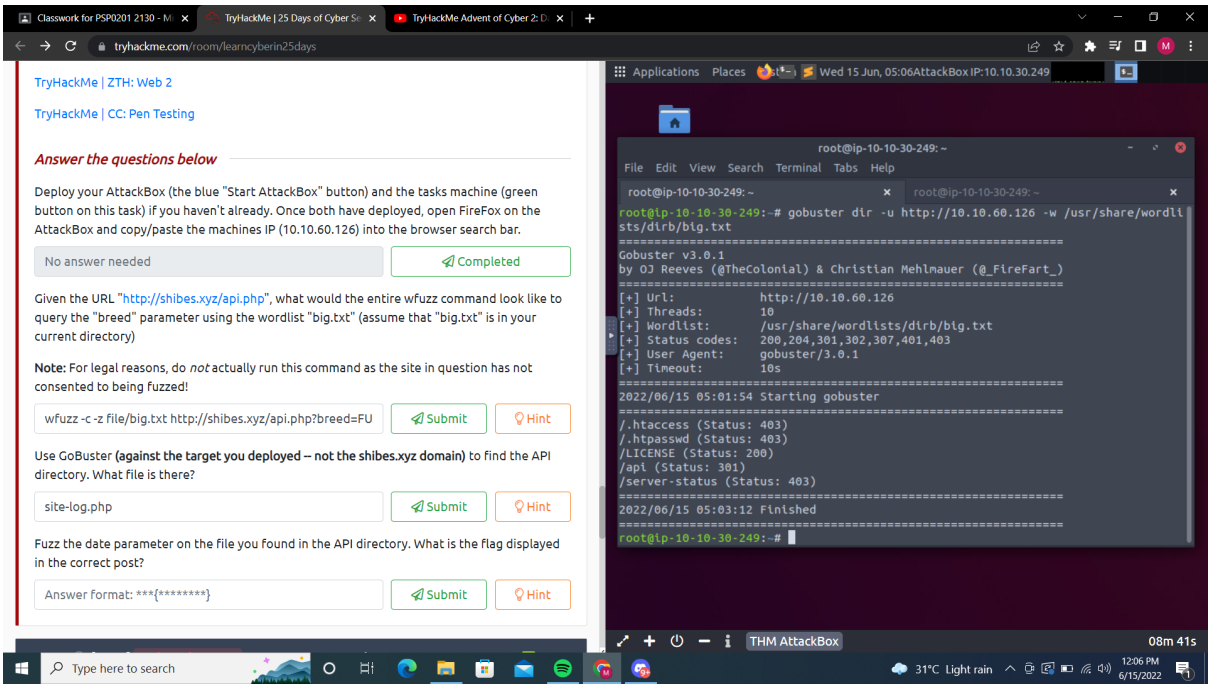
Tools used : Attackbox, Kali Linux, Firefox

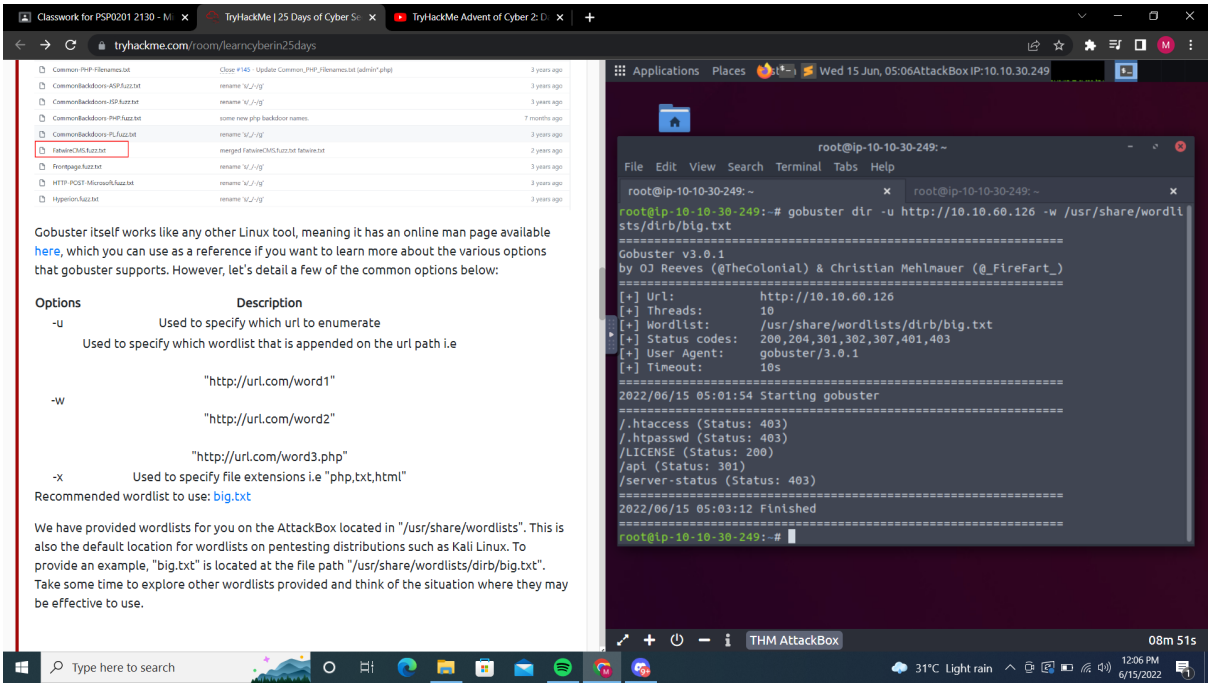
Solution :

Question 1

Sort the wfuzz command in order

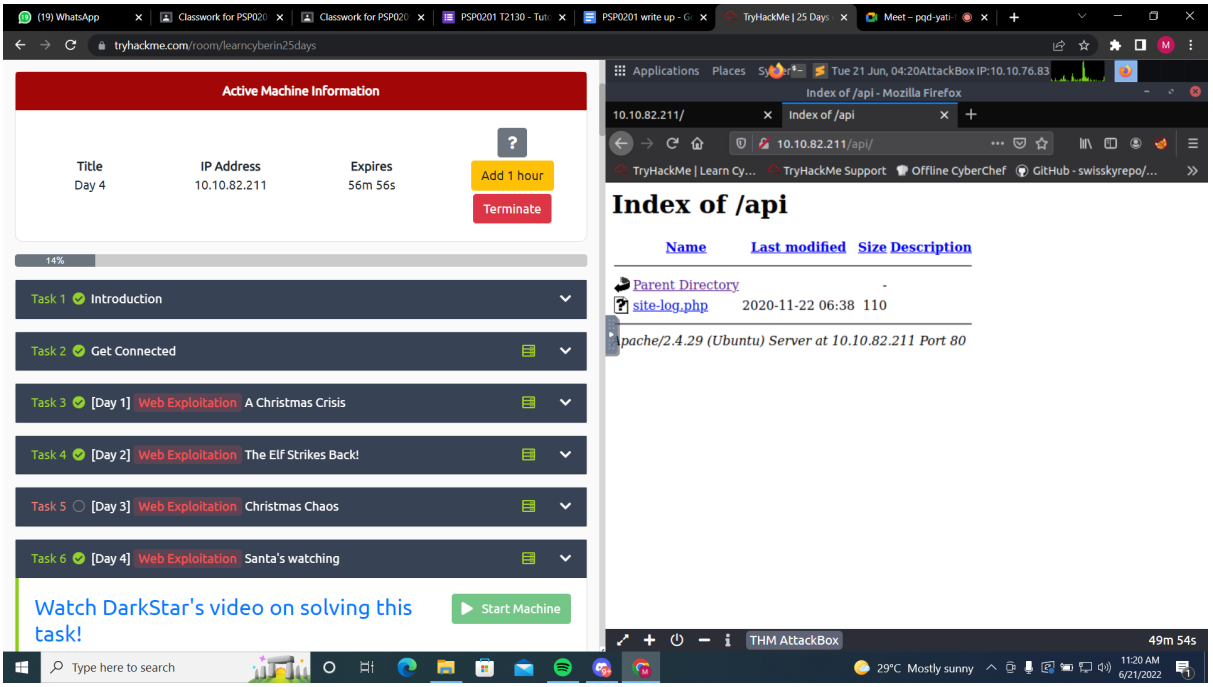

used the --hw option to hide all pages that have 57 words on them. Since wfuzz found a URL with only 8 words, it'll be displayed to us, as this is not 57 words.
It is important to know that you can FUZZ any part of the URL, meaning that you can test any parameters if you don't know them as well.
As with any Linux-based tool, wfuzz also has a useful manpage here, that details some of the more advanced options available to you. Although, I have added some of the more useful options into the table below:
Options
Description
-c
Shows the output in color
-d
Specify the parameters you want to fuzz with, where the data is encoded for a HTML form
-z
Specifies what will replace FUZZ in the request. For example -z file,big.txt. We're telling wfuzz to look for files by replacing "FUZZ" with the words within "big.txt"
--hc
Don't show certain http response codes. I.e. Don't show 404 responses that indicate the file doesn't exist, or "200" to indicate the file does exist
--hl
Don't show for a certain amount of lines in the response
--hh
Don't show for a certain amount of characters
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.
root@ip-10-10-30-249: ~
File Edit View Search Terminal Tabs Help
root@ip-10-10-30-249:~# gobuster dir -u http://10.10.60.126 -w /usr/share/wordlists/dirb/big.txt
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
[+] Url: http://10.10.60.126
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/big.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent: gobuster/3.0.1
[+] Timeout: 10s
2022/06/15 05:01:54 Starting gobuster
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/LICENSE (Status: 200)
/api (Status: 301)
/server-status (Status: 403)
2022/06/15 05:03:12 Finished
root@ip-10-10-30-249:~#
THM AttackBox
08m 12s

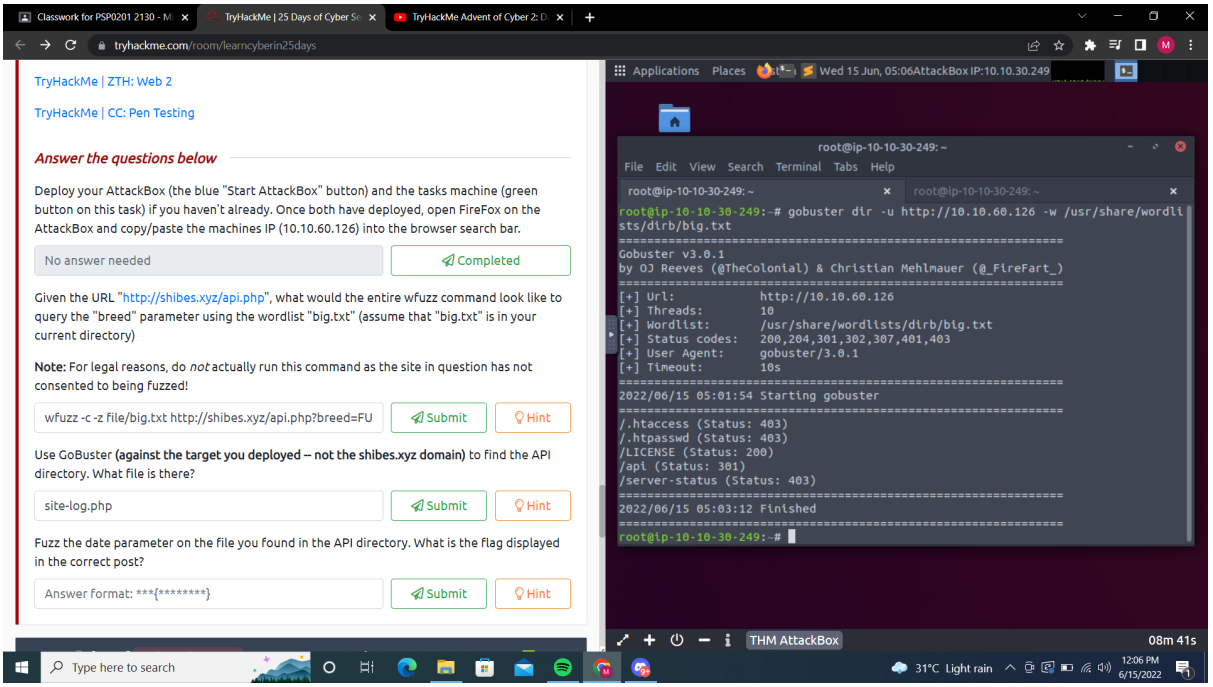

TryHackMe | ZTH: Web 2
TryHackMe | CC: Pen Testing
Answer the questions below
Deploy your AttackBox (the blue "Start AttackBox" button) and the tasks machine (green button on this task) if you haven't already. Once both have deployed, open FireFox on the AttackBox and copy/paste the machines IP (10.10.60.126) into the browser search bar.
No answer needed
Completed
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=FU
Submit
Hint
Use GoBuster (against the target you deployed -- not the shibes.xyz domain) to find the API directory. What file is there?
site-log.php
Submit
Hint
Fuzz the date parameter on the file you found in the API directory. What is the flag displayed in the correct post?
Answer format: ***{********}
Submit
Hint
root@ip-10-10-30-249: ~
File Edit View Search Terminal Tabs Help
root@ip-10-10-30-249:~# gobuster dir -u http://10.10.60.126 -w /usr/share/wordlists/dirb/big.txt
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
[+] Url: http://10.10.60.126
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/big.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent: gobuster/3.0.1
[+] Timeout: 10s
2022/06/15 05:01:54 Starting gobuster
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/LICENSE (Status: 200)
/api (Status: 301)
/server-status (Status: 403)
2022/06/15 05:03:12 Finished
root@ip-10-10-30-249:~#
THM AttackBox
08m 41s



Question 2

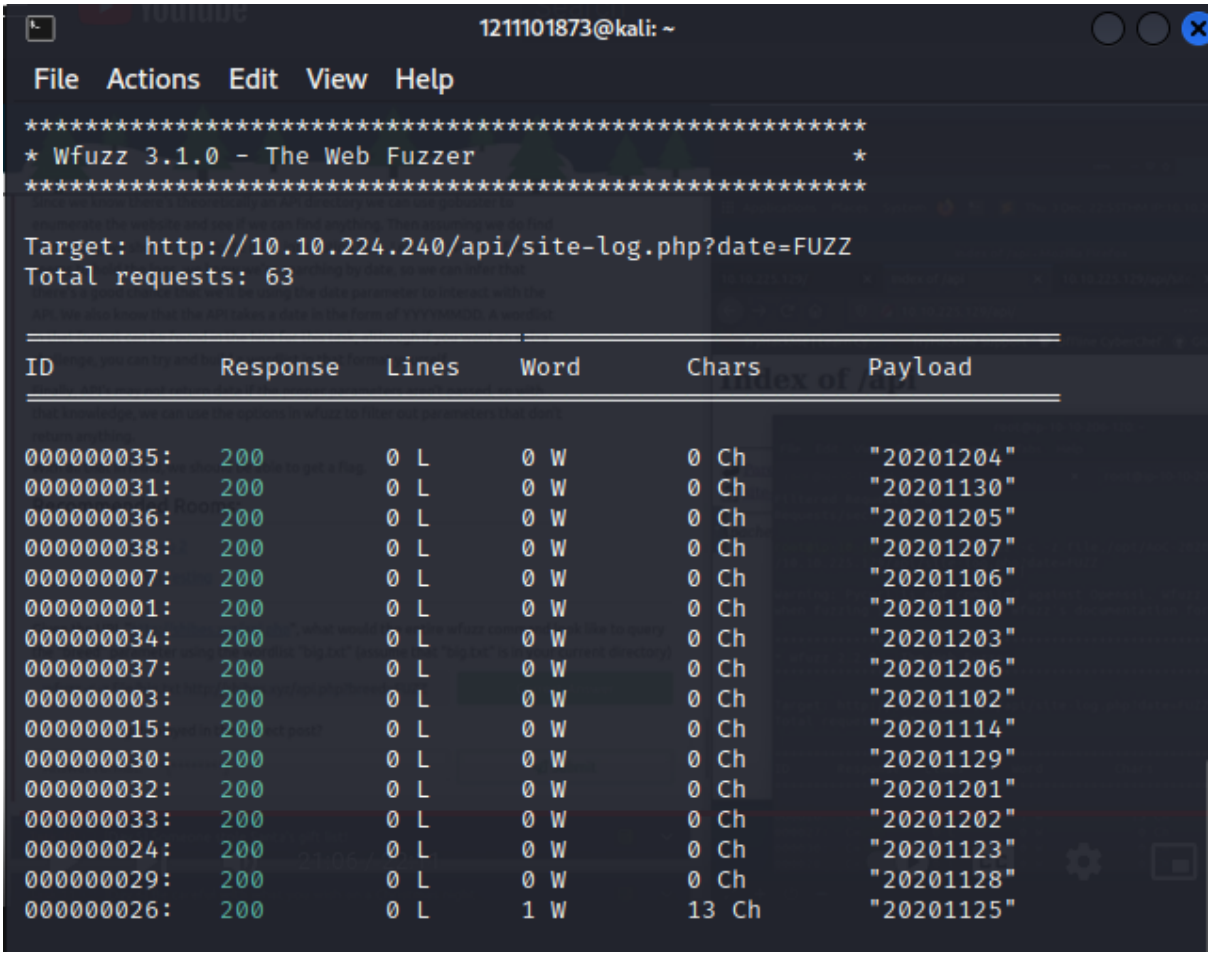
The file that available in the API directory

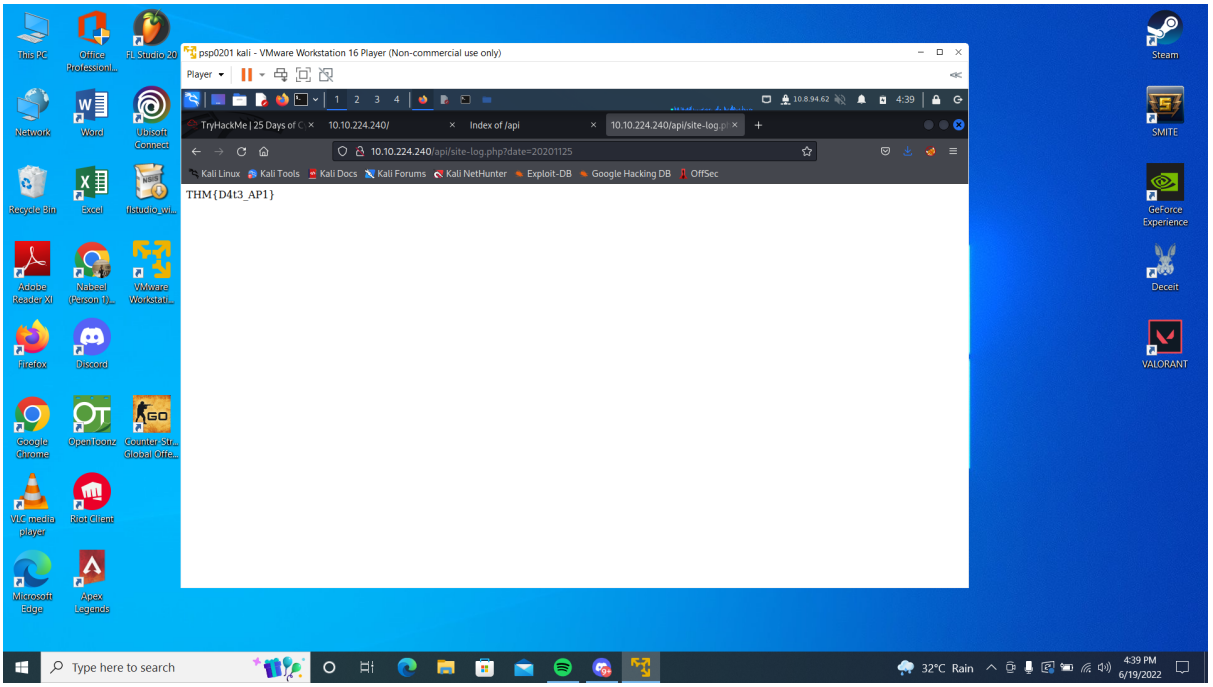




Question 3

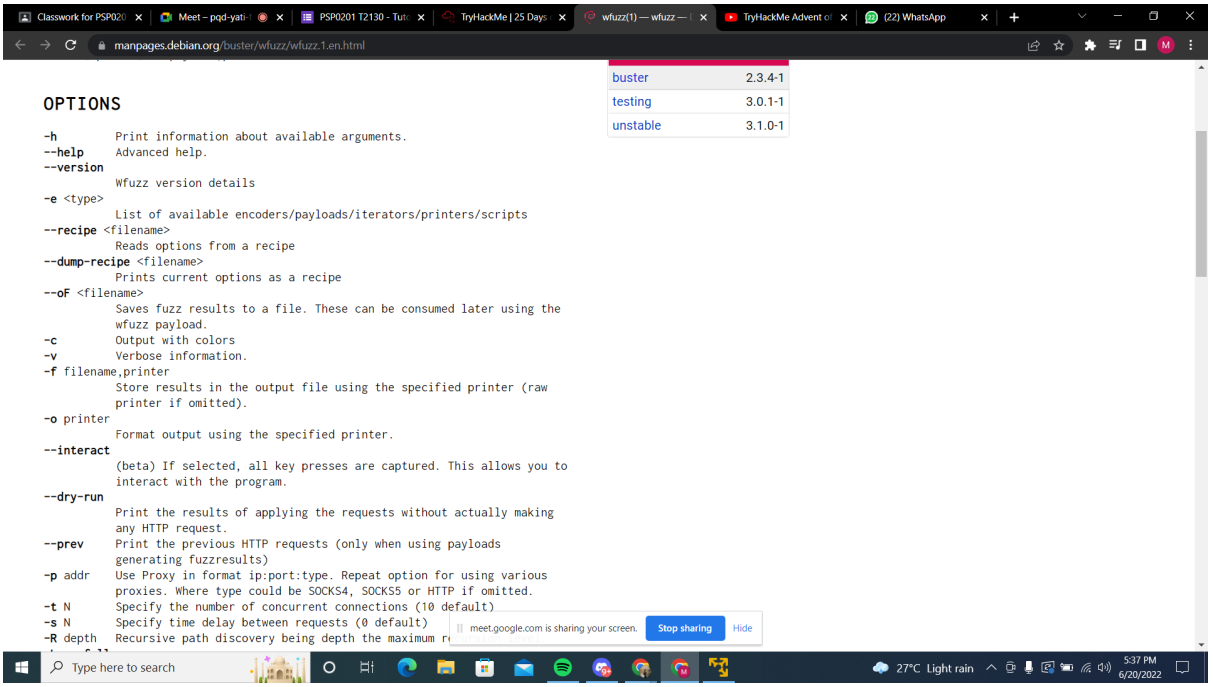
The flag that displayed





Question 4

The -f parameter specifies to printer and filename



The Throughout Process:

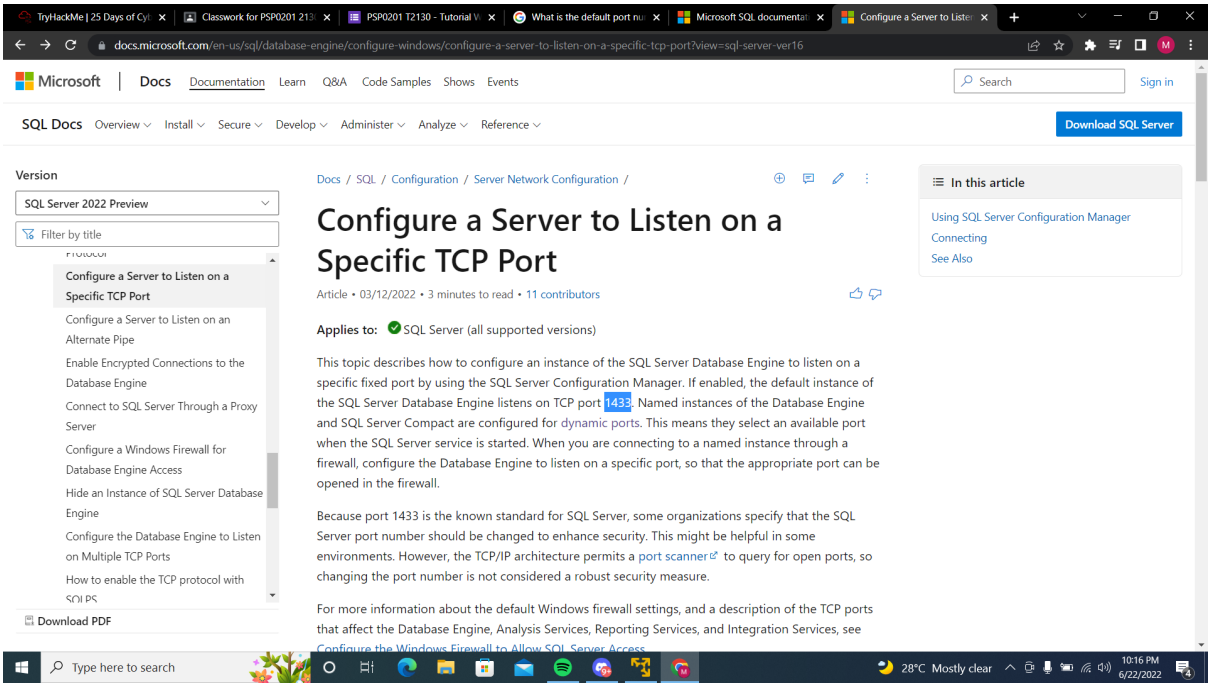
Based on the first question , we were required to sort the wfuzz command accordingly. So , to start the command, we put “wfuzz” in front of the command followed by -c and -z. The command “-c” is for output in colour and “-z” is for telling the wfuzz to search for files by replacing “FUZZ” with the word “big.txt”. Next, we’ll put the url that was given and followed by the command “?breed=FUZZ”.(explaination). Next, we were required to search for any file in the API directory. We continue by searching “**http://ipaddress/api/**”. The file that we found was “**site-log.php**”. Move to the third question, we need to fuzz the date parameter on the file that we found in the API directory. So based on the picture that we provided in **Question 3,** we used the wfuzz command to search for the date parameter. We noticed that out of all of the total requests we received, only one of the date parameters had a different amount of chars and words. We insert “**http://ipaddress/api/site-log.php?date=20201125**” and finally we received the flag. Last but not least, the “-f” parameter results to **printer** and **filename**. The answers are based on wfuzz’s help file that can be found in TryHackMe.

Day 5: Someone stole Santa's gift list!
Tools used: Kali Linux, Firefox, Burp Suite

Solution:

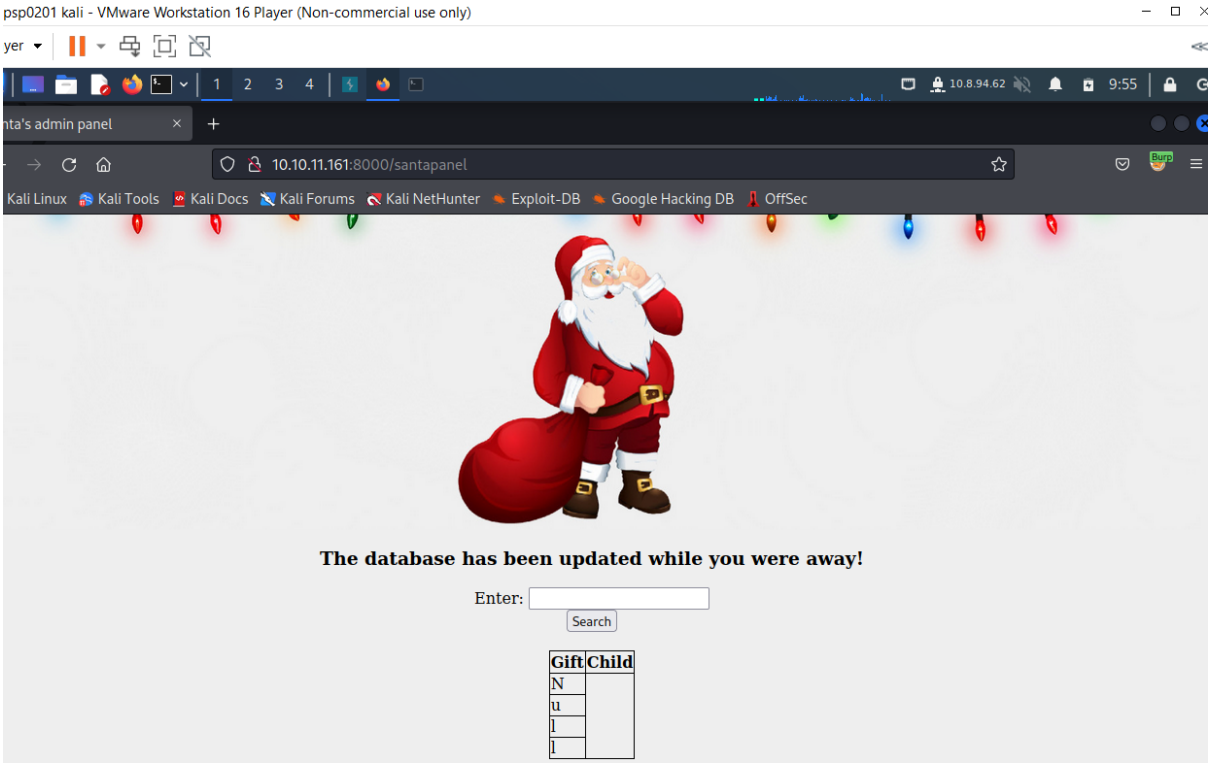
Question 1

The default port number for SQL Server running on TCP

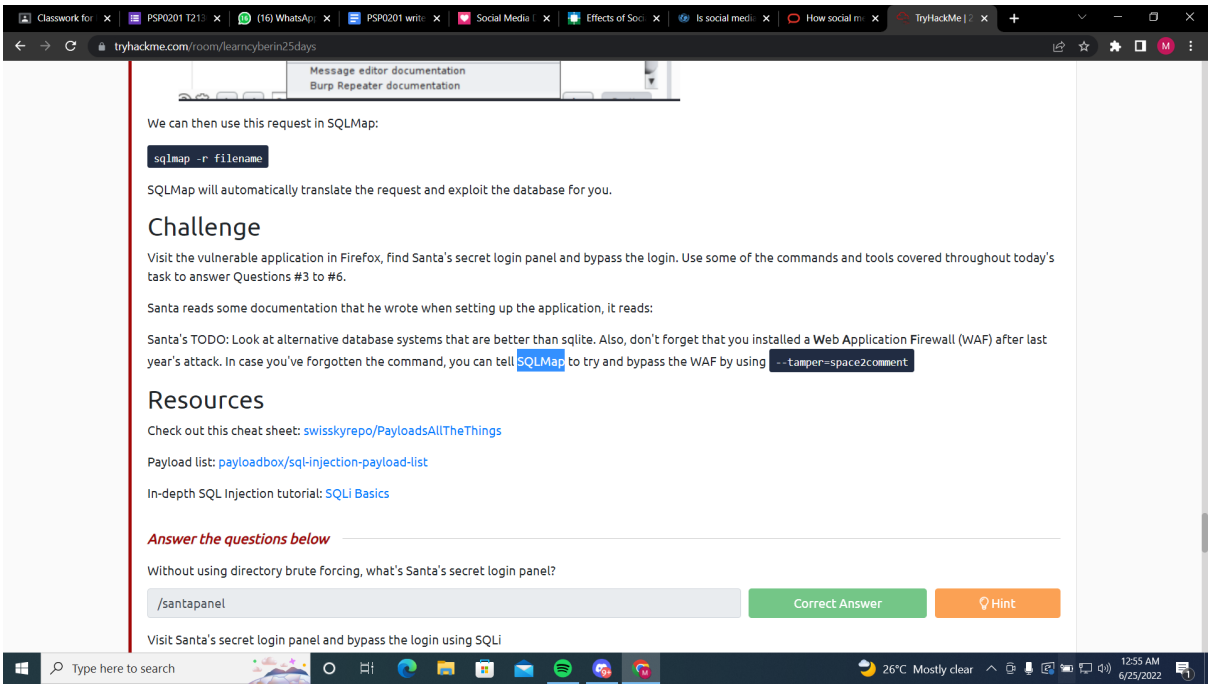


Question 2

Santa's secret login panel

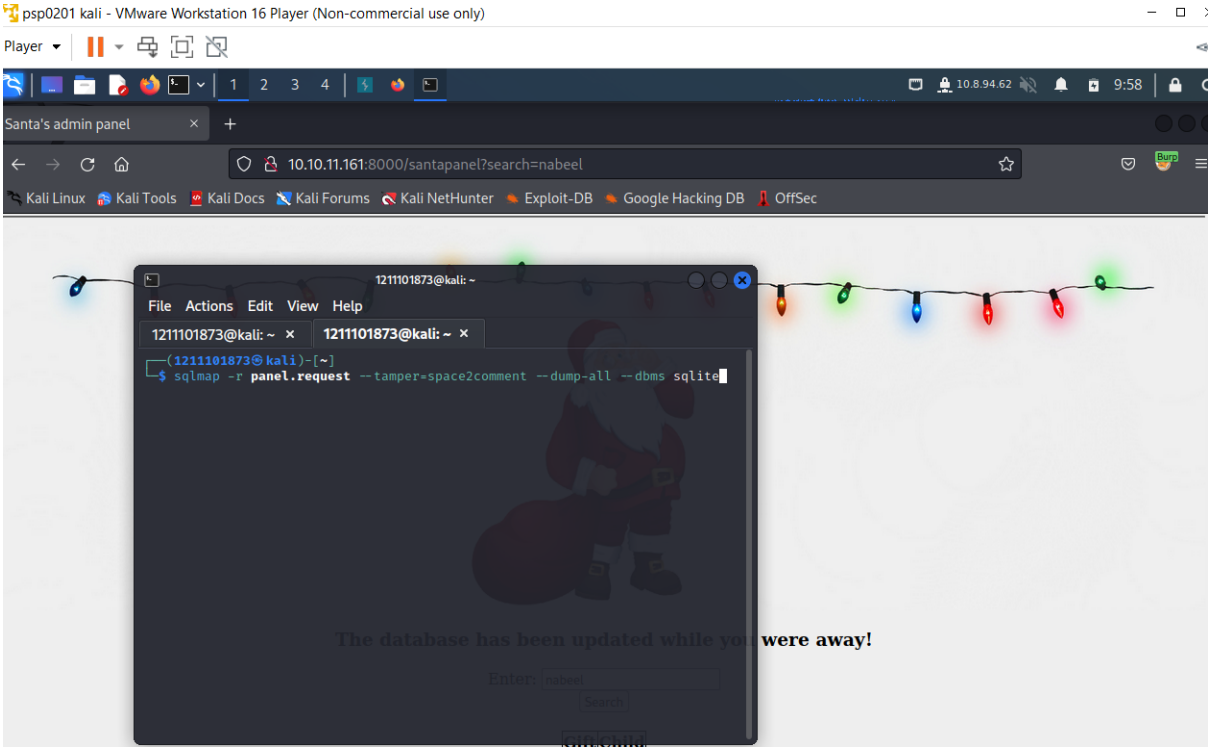


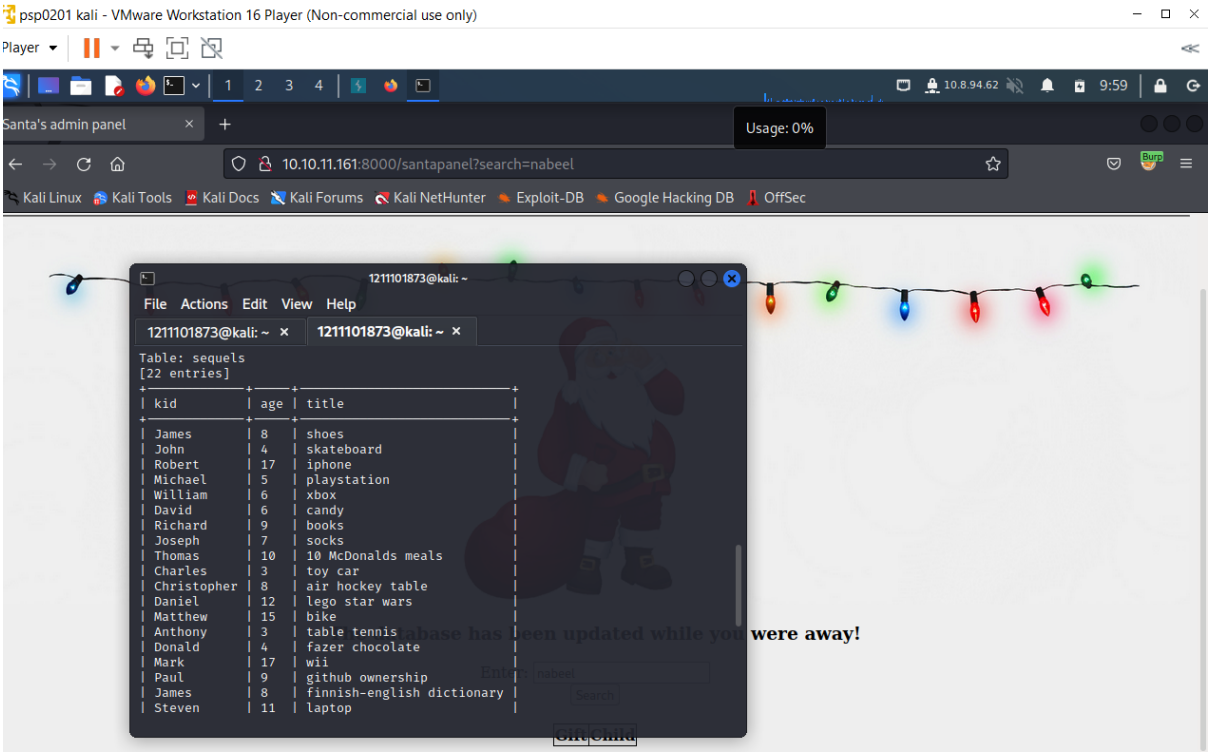
Question 3



Question 4&5&6

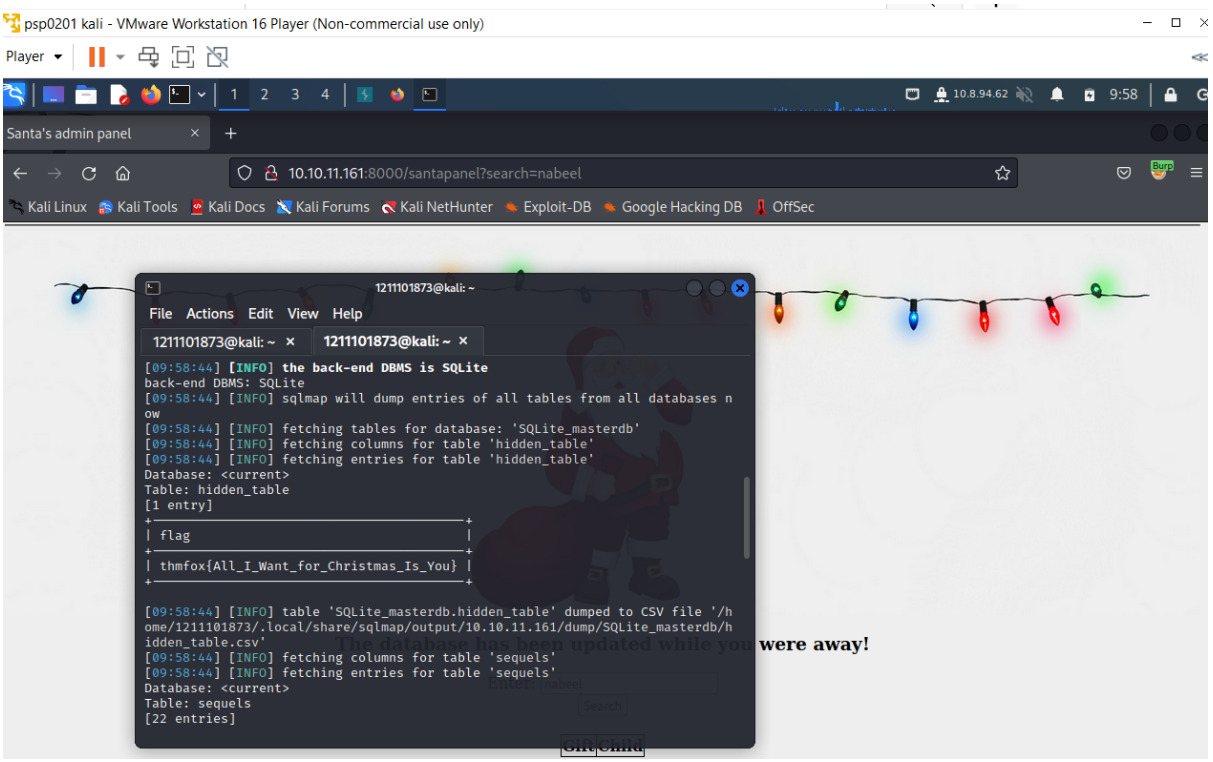
The total number of entries, kids age and wishlist in the Santa's secret panel





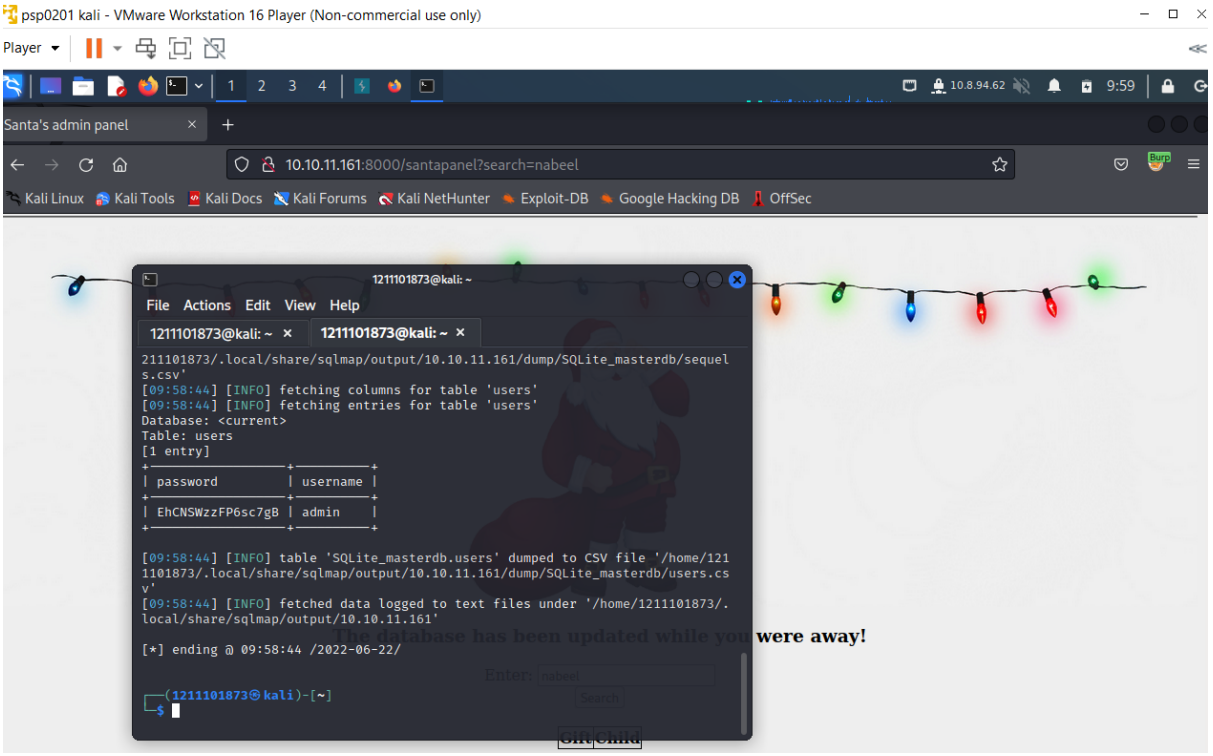
Question 7

The flag in Santa's secret panel



Question 8

The password for the admin in the Santa's secret panel



Throughout Process:

First of all , we started by searching for the default port number for SQL server running on TCP by referring to the microsoft documentation under SQL server. Then, we went to http://IP_ADDRESS/santapanel which is Santa's secret panel and we bypassed the login by inserting **admin' or 1=1 –** as the username and **admin** as the password. Next, we turned on the Burp Proxy so that we can turn on the intercept to intercept the request and save it. Then we will run the SQLmap by entering the command **sqlmap -r panel.request –tamper==space2comment –dump-all –dbms sqlite**. We are using sqlmap to translate the request and exploit the database for us. The other command we just refer to the notes that were given. Once we run the command, we can see the database and answers for the questions.