PSP0201
Week 2
Writeup

GROUP NAME: Cyborgs

MEMBERS

ID NUMBER	NAME	ROLE
1211102066	Hemma Ravindran	Leader
1211100614	Tivaasheny Ananthan	Member
1211102168	Nicholas Cheok Jia Jie	Member
1211100986	Sarvesh Munusamy	Member

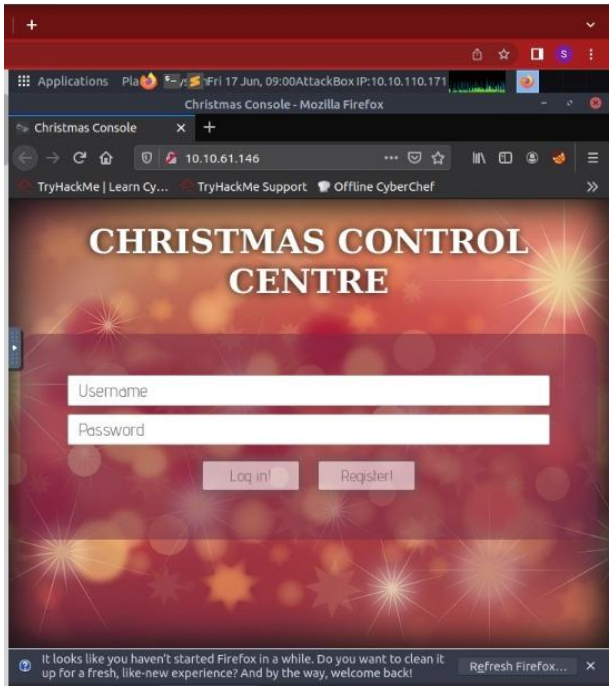
Day 1: Web Exploitation – A Christmas Crisis

Tools used: Kali Linux, Firefox

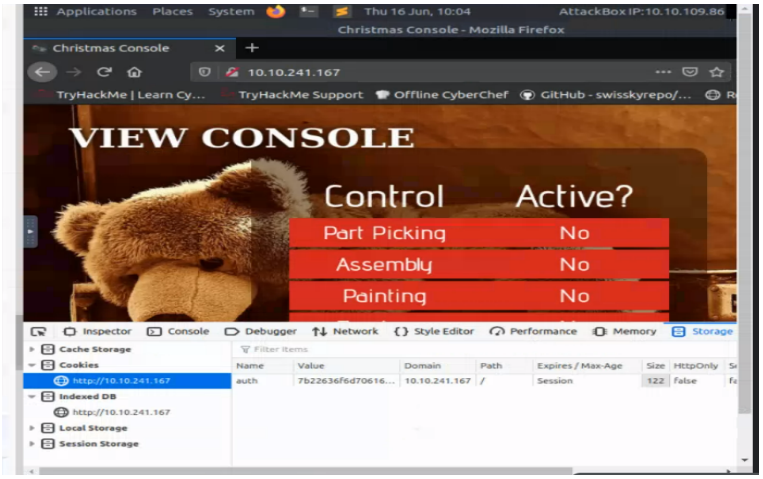
Solution/walkthrough:

Question 1:

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

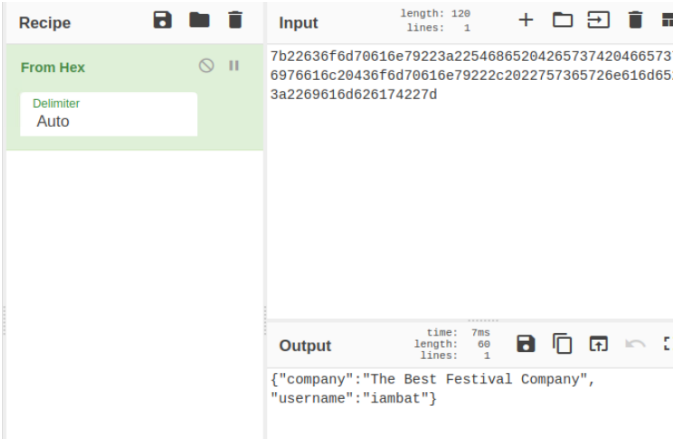


Opening up the browser developer tools to check on the cookie.



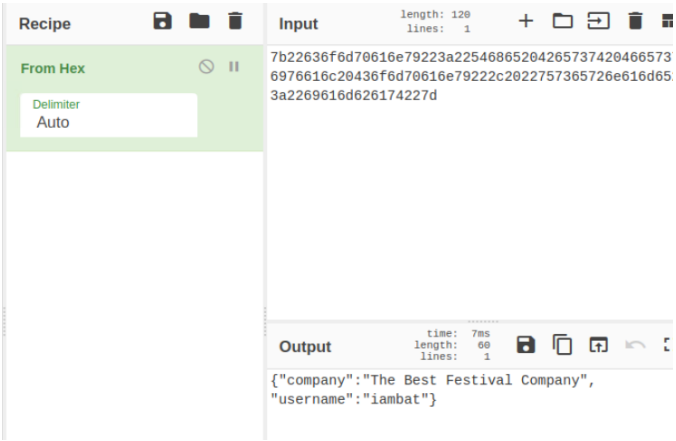
Question 2:

Obtain the value of the cookie.



Question 3:

Using Cyberchef, convert the cookie value to string.



Question 4:

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

Question 5 :

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

Thought Process/Methodology:

We are supposed to put in the IP Address in the firefox. It will lead us to the console website which is **Christmas Console**.Now in here we need to log in to the system to do that we need to register to the system first and then when we login to the web site using created account it shows the control active.Later, we should open the browser’s developer tool and then choose to view the site for the cookie value.We can see the name of the cookie which is **auth**. If we inspect it well we can see it has characters 0 to 9 and A to E so it looks like **hexadecimal** which is the answer. We can also know that the format is **JSON**.Next 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 a converted one and refreshed the page. We are now shown an administrator page (Santa’s) and proceeded to enable every control, which in turn showed the flag.

Day 2: The Elf Strikes Back

Tools used: Kali Linux, Firefox

Solution/walkthrough:

Question 1:

Copy and paste the string from the url after login in.

For Elf McEager:

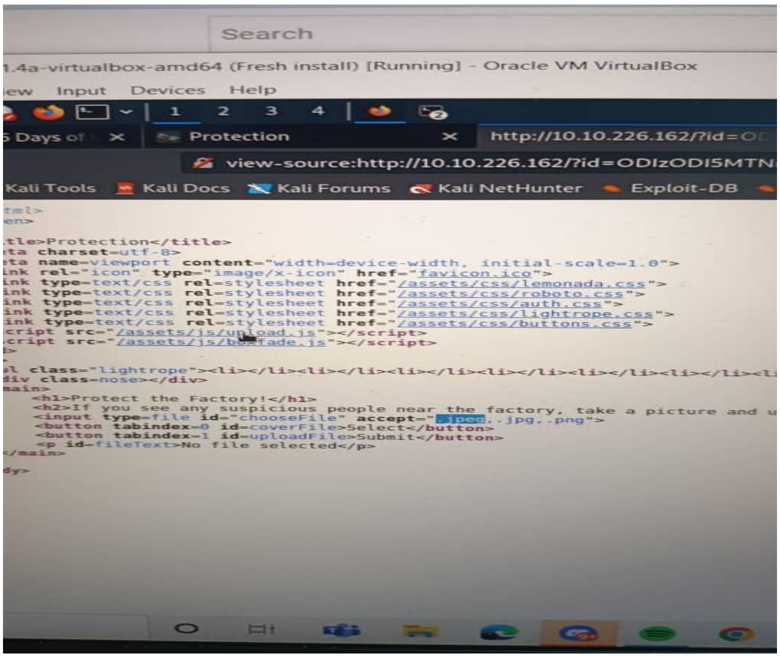
You have been assigned an ID number for your audit of the system: ODIzODI5MTNiYmYw . Use this to gain access to the upload section of the site.

Good luck!

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

Question 2:

View the page source and see what file type is accepted. (jpeg,JPG)

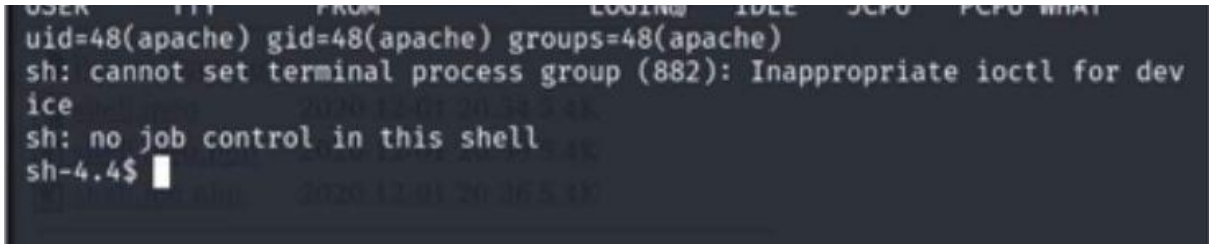


Question 3:

Manually guessed where the directory are stored by searching, (ip address)/uploads/

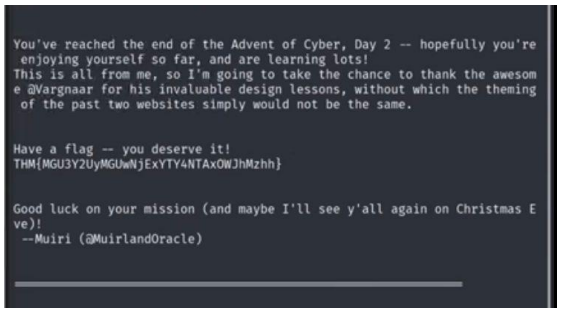
Question 4:

Activated reverse shell



Question 5:

Use terminal and get the flag after completing the task



Thought Process/Methodology:

Having accessed the target machine, copy the webshell which is given and paste it in the terminal. After gaining access, add the access code to our IP Address and refresh the page, and then we will be signed in the page. Later we should view the page source in order to know the type of file that is accepted in this site which is images. Later , we manually guess the common directory and find out which directory are our uploaded files stored, which is the /uploads directory. Then, we should activate the reverse shell. Finally, we should run command cat/var/www/flag.txt and get the flag.

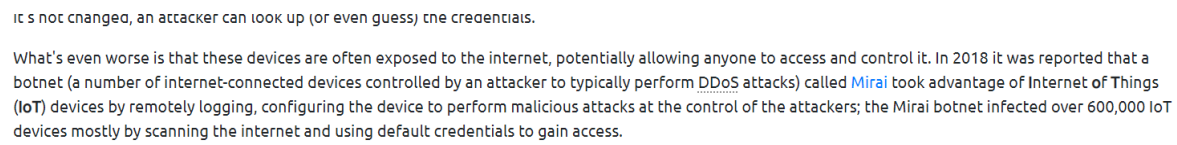
Day 3: Christmas Chaos

Tools used: Kali Linux, Firefox

Solution/walkthrough:

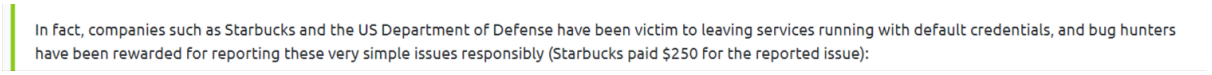
Question 1:

Got the answer from this paragraph in tryhackme.



Question 2:

Got the answer from the sentence here that says “Starbucks paid $250 for the reported issue”

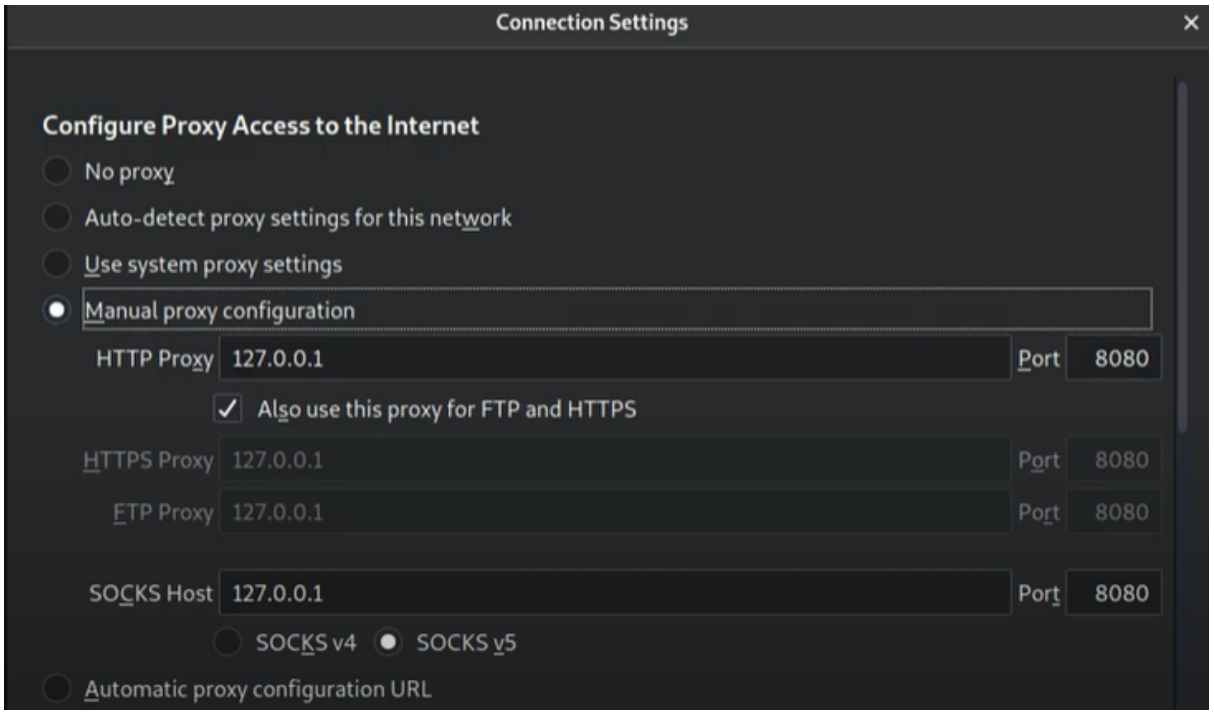


Question 3:

We clicked into the hackerone website researched about the agent assigned from the Dept of Defense that disclosed the report on June 25th which is “ag3nt-j1”.

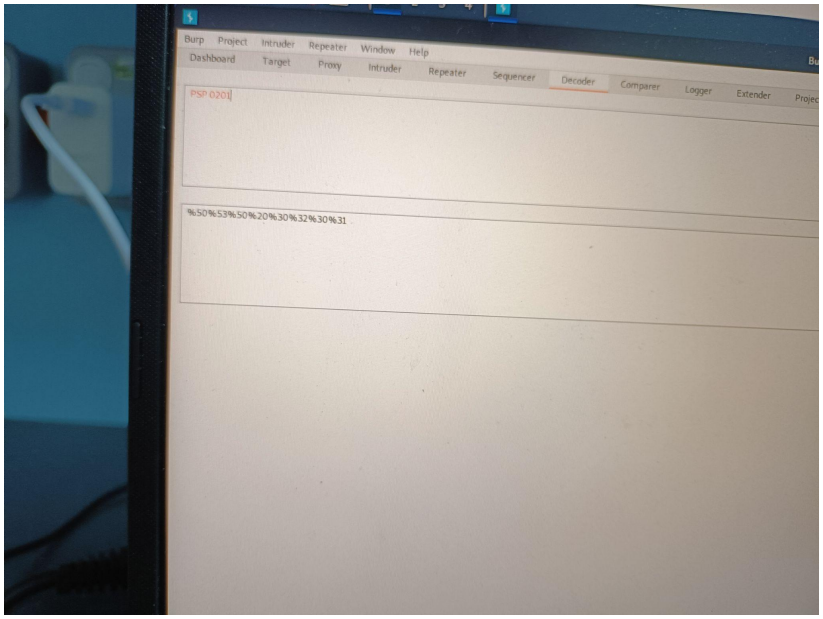
Question 4:

We went to settings and clicked on “manual proxy configuration” and typed the proxy number and entered port 8080.



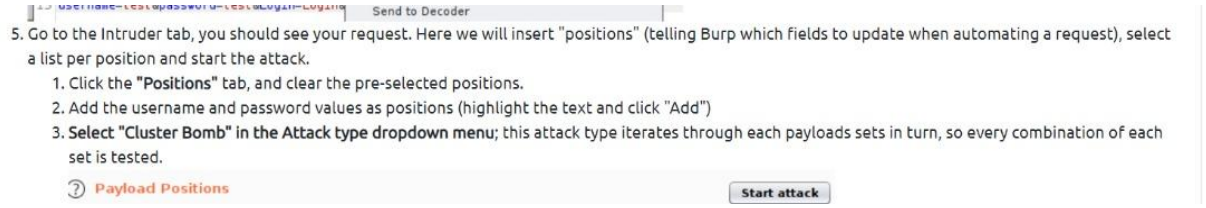
Question 6:

We clicked on the decoder tab and typed in “PSP 0201” in there and we converted it into URL and got the code.



Question 7:

We got the answer of question 7 from the third statement.



Question 8:

We login in “santa sleigh tracker” website with the username and passwords given in tryhackme and we got the flag there.

Thought Process/Methodology:

Firstly, we went to the tryhackme website and started the machine and the attack box. We also installed “BurpSuite” to do our Day 3. We read and scheme throught the text in tryhackme for Day 3 and we found out that Mirai is the name of the botnet mentioned in the text that was reported in 2018 and also found out that Starbucks paid $250 for reporting default credentials. After that, we went to the hackerone website to find the agent who was assigned from the Dept of Defense that disclosed the report on Jun 25th which is ag3nt-j1. After that, we went to the settings in the firefox in the attack box and clicked on “manual proxy configuration” and typed in the proxy number and the port number 8080. We went to BurpSuite and tab on decoder to type in PSP 0201 and we converted it into URL and we got the code then we looked into the text in tryhackme and looked for the correct attack typer which is cluster bomb. We login in “santa sleigh tracker” website with the username and passwords in the firefox and we got our flag there.

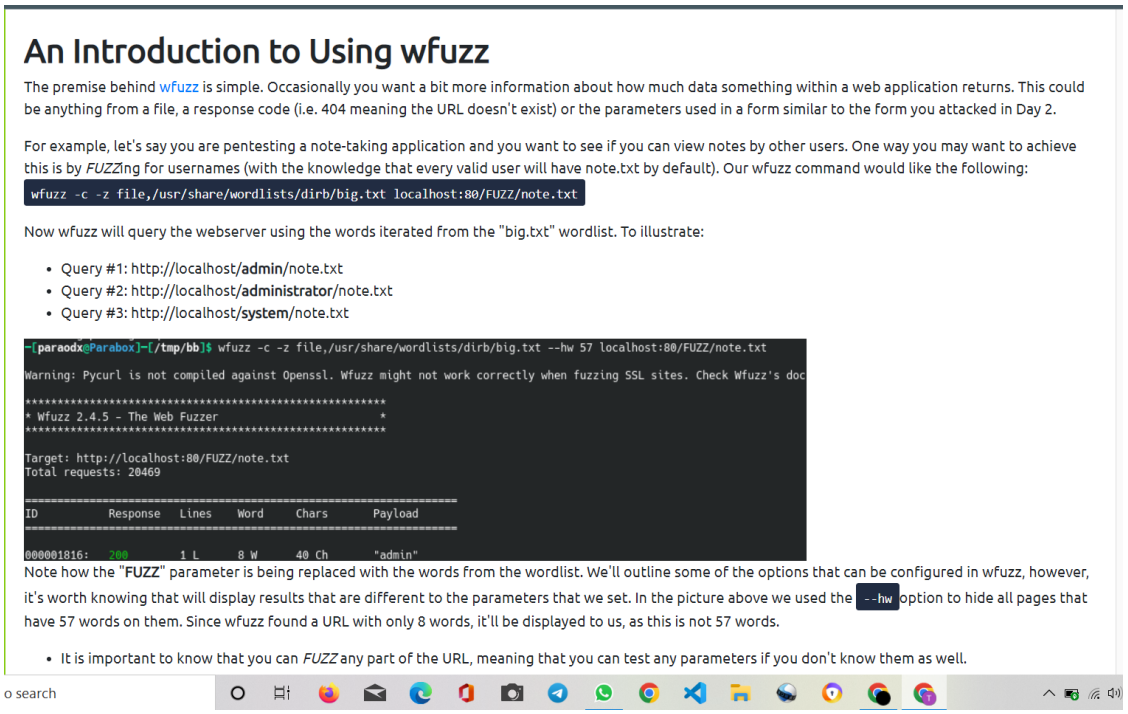
Day 4: Web Exploitation – Santa's watching

Tools used: google, attack box, Firefox

Solution/walkthrough:

Question 1:

From try hack me when we read this we knew that the highlighted one is basic line of wfuzz. So copy paste first this in answer box then we have to change one by one to get the correct answer. Then we need to change the locol host:80 to the given url in question ("http://shibes.xyz/api.php").



An Introduction to Using wfuzz

The premise behind wfuzz is simple. Occasionally you want a bit more information about how much data something within a web application returns. This could be anything from a file, a response code (i.e. 404 meaning the URL doesn't exist) or the parameters used in a form similar to the form you attacked in Day 2.

For example, let's say you are pentesting a note-taking application and you want to see if you can view notes by other users. One way you may want to achieve this is by *FUZZ*ing for usernames (with the knowledge that every valid user will have note.txt by default). Our wfuzz command would like the following:

```
wfuzz -c -z file,/usr/share/wordlists/dirb/big.txt localhost:80/FUZZ/note.txt
```

Now wfuzz will query the webserver using the words iterated from the "big.txt" wordlist. To illustrate:

- Query #1: http://localhost/**admin**/note.txt
- Query #2: http://localhost/**administrator**/note.txt
- Query #3: http://localhost/**system**/note.txt

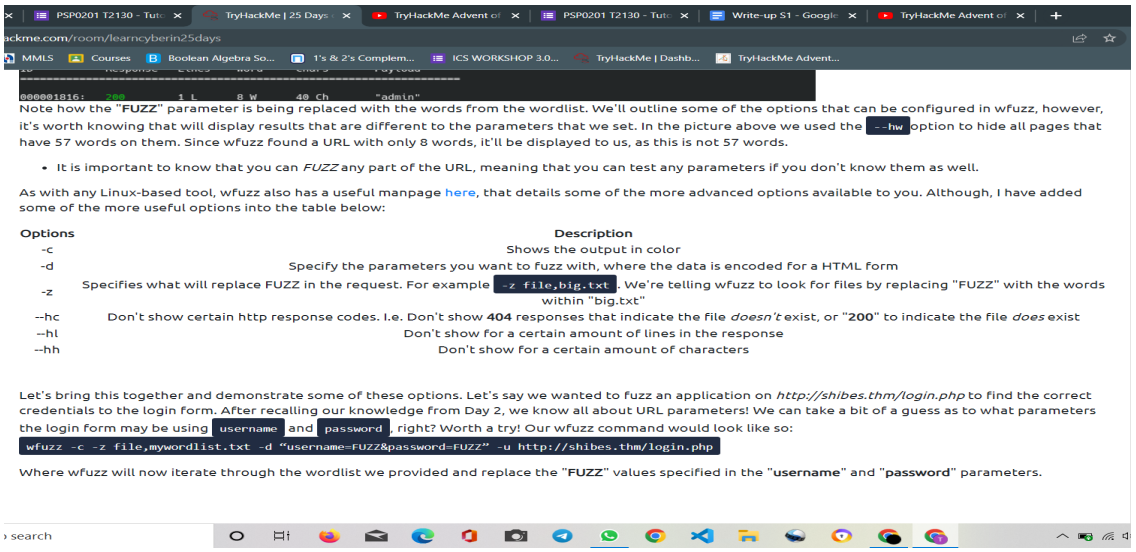
```
┌[paraodx@Parabox]-[/tmp/bb]$ wfuzz -c -z file,/usr/share/wordlists/dirb/big.txt --hw 57 localhost:80/FUZZ/note.txt

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's doc

********************************************************
* Wfuzz 2.4.5 - The Web Fuzzer                         *
********************************************************

Target: http://localhost:80/FUZZ/note.txt
Total requests: 20469

=====================================================================
ID           Response   Lines    Word     Chars       Payload
=====================================================================

000001816:   200        1 L      8 W      40 Ch       "admin"
```

Note how the "**FUZZ**" parameter is being replaced with the words from the wordlist. We'll outline some of the options that can be configured in wfuzz, however, it's worth knowing that will display results that are different to the parameters that we set. In the picture above we 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.

o search

Then follow it's instruction that after -z file, will be big.text so just leave it. Next, based on given video we followed the last part(/FUZZ/note.txt) need to change to ?breed=FUZZ. From that new line we get order.



ackme.com/room/learncyberin25days

MMLS Courses Boolean Algebra So... 1's & 2's Complem... ICS WORKSHOP 3.0... TryHackMe | Dashb... TryHackMe Advent...

```
000001816:   200        1 L      8 W      40 Ch       "admin"
```

Note how the "**FUZZ**" parameter is being replaced with the words from the wordlist. We'll outline some of the options that can be configured in wfuzz, however, it's worth knowing that will display results that are different to the parameters that we set. In the picture above we 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:

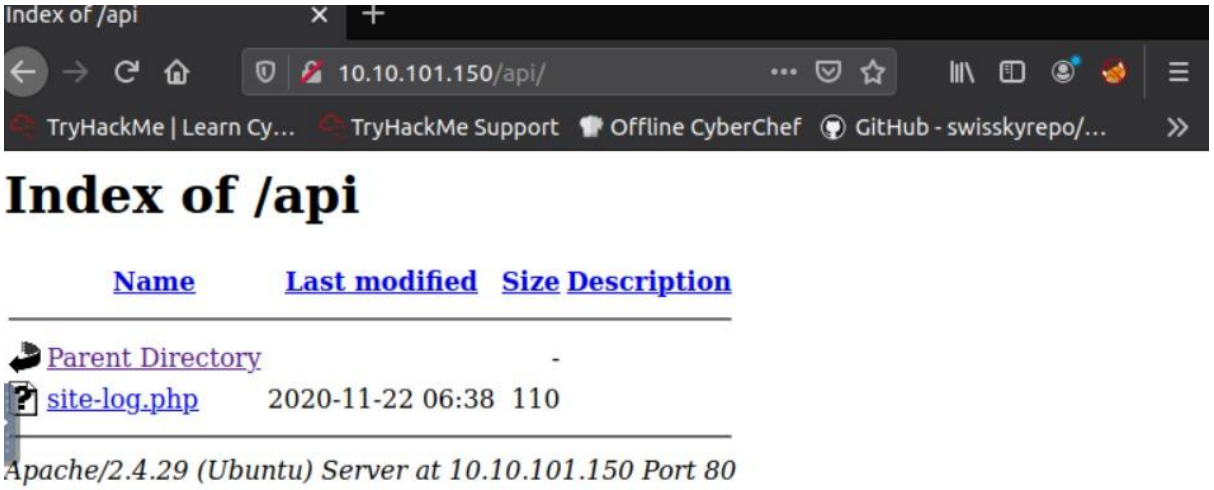
```
wfuzz -c -z file,mywordlist.txt -d "username=FUZZ&password=FUZZ" -u http://shibes.thm/login.php
```

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

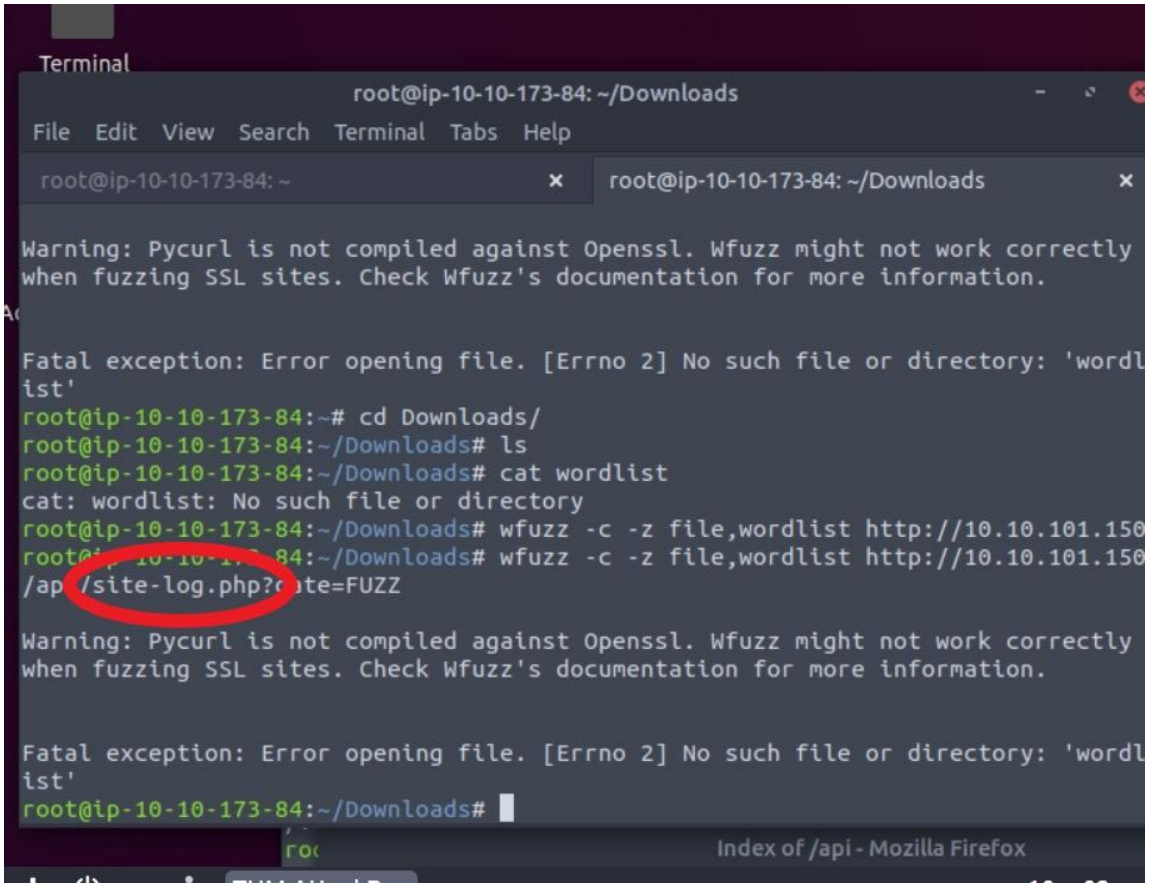
› search

Question 2:

We need to type /api/ so we will get the answer which is site-log.php

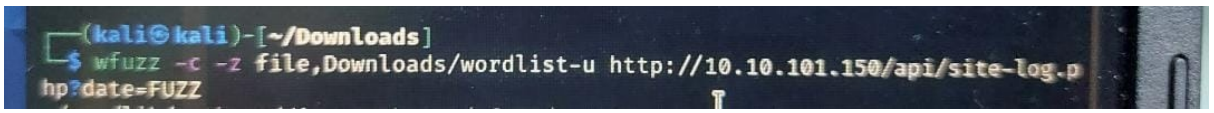


Actually we saw (site-log.php) at the terminal too.

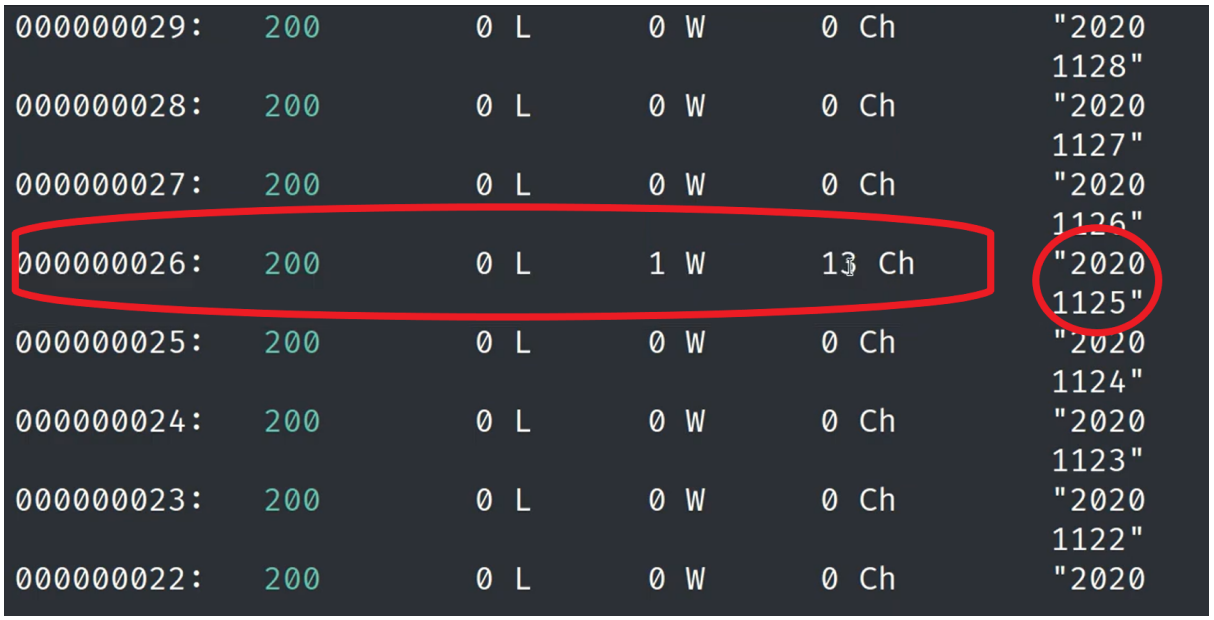


Question 3:

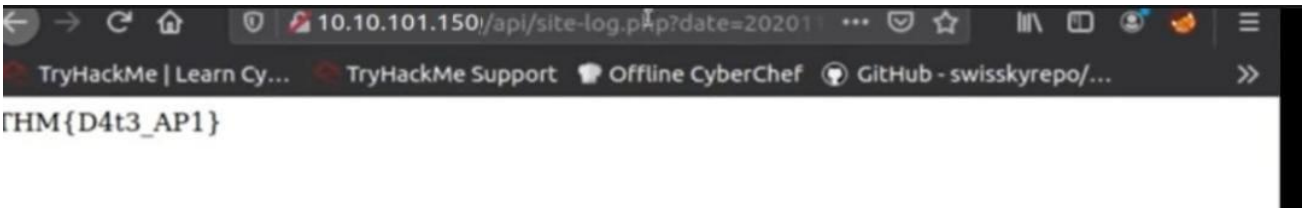
We need to type this in order to get the date in the form of YYYYMMDD.



After we get the correct one, we need to put it in.

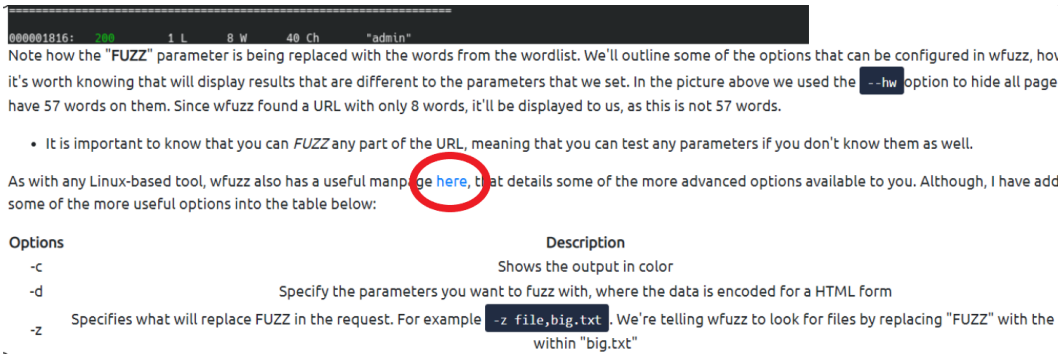


Then we need to put it beside our IP address like that to get the flag

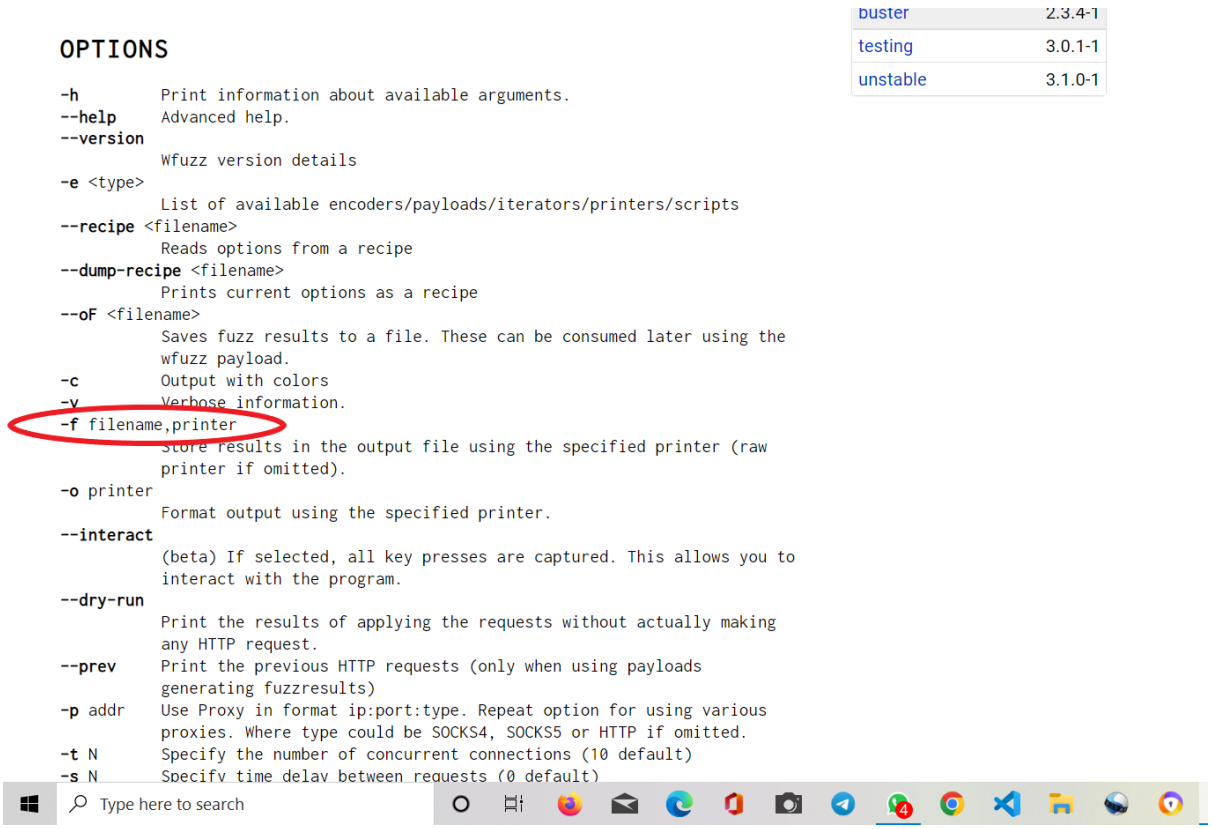


Question 4:

We need to press this to get more advanced options.



Then we can find the answer from the list.



```
OPTIONS

-h          Print information about available arguments.
--help      Advanced help.
--version
            Wfuzz version details
-e <type>
            List of available encoders/payloads/iterators/printers/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 using 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 making
            any HTTP request.
--prev      Print the previous HTTP requests (only when using payloads
            generating fuzzresults)
-p addr     Use Proxy in format ip:port:type. Repeat option for using various
            proxies. Where type could be SOCKS4, SOCKS5 or HTTP if omitted.
-t N        Specify the number of concurrent connections (10 default)
-s N        Specify time delay between requests (0 default)
```

buster	2.3.4-1
testing	3.0.1-1
unstable	3.1.0-1

Type here to search

Thought Process/Methodology:

Throughout the process we able to know some of the fundamental tool used in web application testing, Fuzzing, and also we will know how to use go buster to enumerate a web server for hidden files. Firstly, we start the machine and also the attack box. After get the IP address we type it in firefox in attackbox. An animated chritsmas tree will appear. Then we to open our terminal and before that download the wordlist that given in try hack me. We need to type gobuster dir -u our IP address(copy paste from firefox) -w /usr/share/wordlists/dirb/big.txt -x .php. Then go back to web browser and type /api/ to find the api directory to get the file. Go to wfuzz to find the log file to get information. When the wfuzz runs we get response like 13 characters to copy that date in the form of YYYYMMDD. Lastly, we need to type(/api/site-log.php?date=20201125) then we will get the flag.

Day 5: Someone stole Santa's Gift

Tools used: google, attack box, Firefox , Burpsuite

Solution/walkthrough

Question 1:

The default port number for SQL Server running on TCP

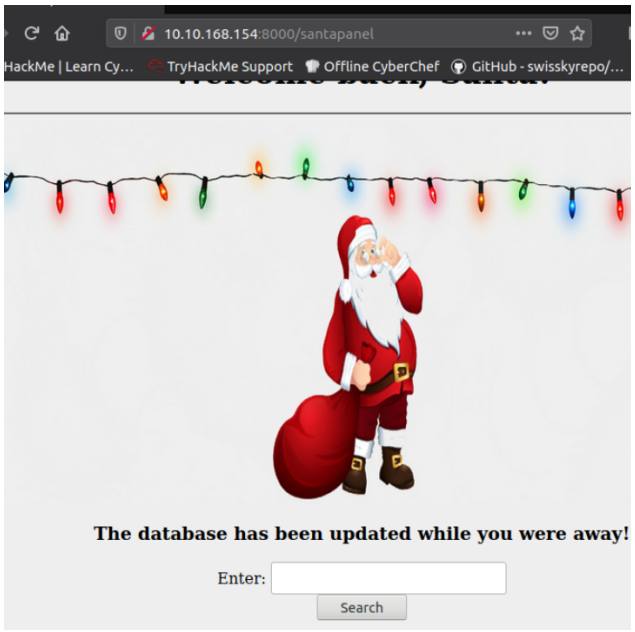
port 1433

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.

11 Mar 2022

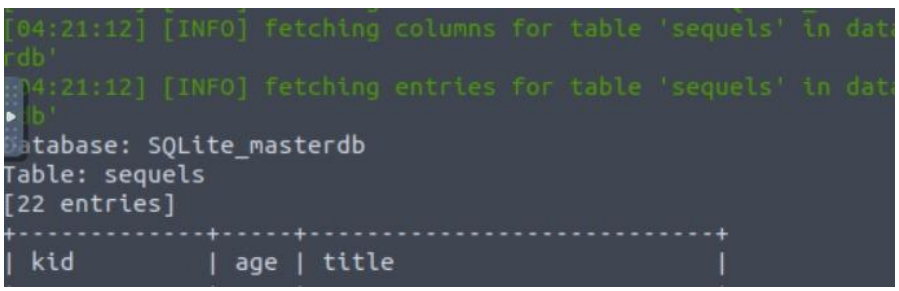
Question 2:

We could eventually guess that the secret login panel will be santapanel from the hint.



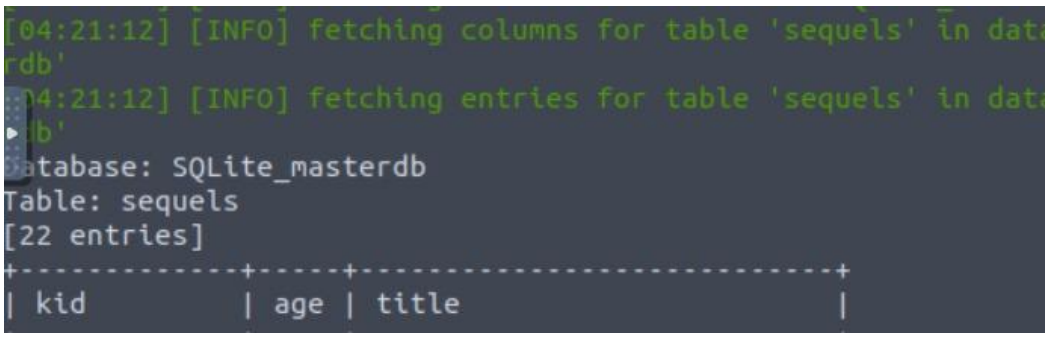
Question 3:

The database used from the hint in Santa's TODO list from the terminal



Question 4:

The entry number from the database



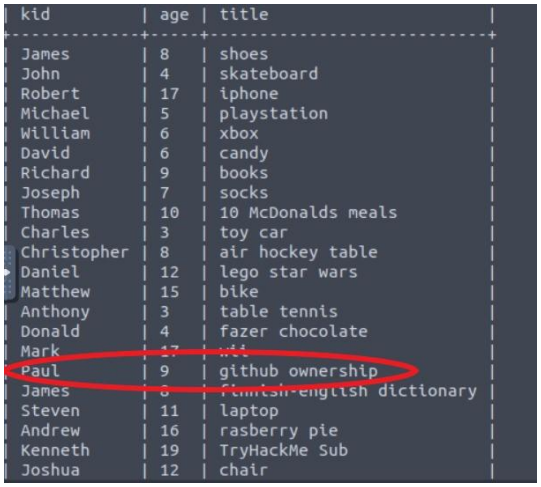
Question 5:

Jake's age from the entry table

kid	age	title
James	8	shoes
John	4	skateboard
Robert	17	iphone
Michael	5	playstation
William	6	xbox
David	6	candy
Richard	9	books
Joseph	7	socks
Thomas	10	10 McDonalds meals
Charles	3	toy car
Christopher	8	air hockey table
Daniel	12	lego star wars
Matthew	15	bike
Anthony	3	table tennis
Donald	4	fazer chocolate
Mark	17	wii
Paul	9	github ownership
James	8	finnish-english dictionary
Steven	11	laptop
Andrew	16	rasberry pie
Kenneth	19	TryHackMe Sub
Joshua	12	chair

Question 6:

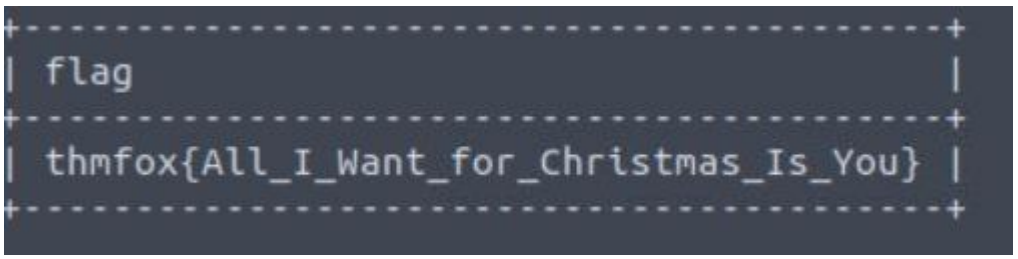
Paul's wish from the entry table



kid	age	title
James	8	shoes
John	4	skateboard
Robert	17	iphone
Michael	5	playstation
William	6	xbox
David	6	candy
Richard	9	books
Joseph	7	socks
Thomas	10	10 McDonalds meals
Charles	3	toy car
Christopher	8	air hockey table
Daniel	12	lego star wars
Matthew	15	bike
Anthony	3	table tennis
Donald	4	fazer chocolate
Mark	17	wii
Paul	9	github ownership
James	8	finnish-english dictionary
Steven	11	laptop
Andrew	16	rasberry pie
Kenneth	19	TryHackMe Sub
Joshua	12	chair

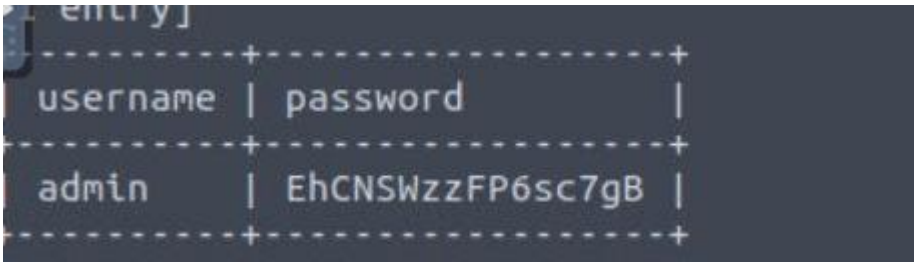
Question 7:

Flag that we got from the hidden table in the terminal.



Question 8:

Admin password that we also got from the hidden table in the terminal.



Thought Process/Methodology:

This time we are using SQL Injection on Santa's Forum. For question 2, we just guessed based on the hint..From there we can see Santa's Secret Login Panel. Later,that will take us to a new page where we will be able to traverse the database. We used Burp Suite and SqlMap to automate this . After we get Burp Suite opened, turn on Foxy Proxy to start intercepting. Make sure the intercept is on.Then head back over to the webpage and do a test request.After hit search, check back with Burp and you will see the request. Save this so SqlMap can use it. Can turn intercept and Foxy Proxy off at this point.From there, head to a command line to start SqlMap. We used the following command. From the results, we were able to answer the rest of the questions.We can see there are 22 entries in the gift database.Using that same table, we can see that James is 8 years old and Paul asked for github ownership.The admin password and flag is located in the "hidden_table".