

PSP0201

Week 3

Writeup

Group Name: F4urDeveloper

Members:

ID	NAME	ROLE
1211101242	RAJA FITRI HAZIQ BIN RAJA MOHD FUAD	LEADER
1211104237	ALIA MAISARA BINTI SHAHIRIN	MEMBER
1211102287	TERRENCE CHENG	MEMBER
1211101153	MISCHELLE THANUSHA JULIUS	MEMBER

Day 6: Web Exploitation - Be careful with what you wish on a Christmas night

Tools used: THM Machine/Kali Linux/Mozilla Firefox/Zaproxy

Solution/Walkthrough:

Question 1:

Examine the OWASP Cheat Sheet. Match the input validation level with the correct description.

The answers for Syntactic and Semantic Validation can be found in the statement shown in the image below (highlighted for viewing purposes).

The screenshot shows two side-by-side sections of the OWASP Cheat Sheet. The left section is titled 'Syntactic Validation' and discusses the format of email addresses according to RFC 5321. It lists several invalid email address examples:

- "><script>alert(1);</script>@example.org"
- user+subaddress@example.org
- user@[IPv6:2001:db8::1]
- "+@example.org"

It notes that parsing email addresses for validity with regular expressions is complicated. The right section is titled 'Semantic Validation' and discusses determining email address correctness. It lists requirements for a valid email address:

- The email address is correct.
- The application can successfully send emails to it.
- The user has access to the mailbox.

It also specifies that ownership links should contain tokens and lists requirements for those tokens.

Question 2:

Examine the OWASP Cheat Sheet. What is the regular expression used to validate a US Zip code?

The regular expression used to validate a US Zip code can be found as shown in the image below (highlighted for viewing purposes).

The screenshot shows a section of the OWASP Cheat Sheet titled 'Validating a U.S. Zip Code (5 digits plus optional -4)'. It displays the regular expression used for validation:

```
^\d{5}(-\d{4})?$/
```

Question 3:

What vulnerability type was used to exploit the application?

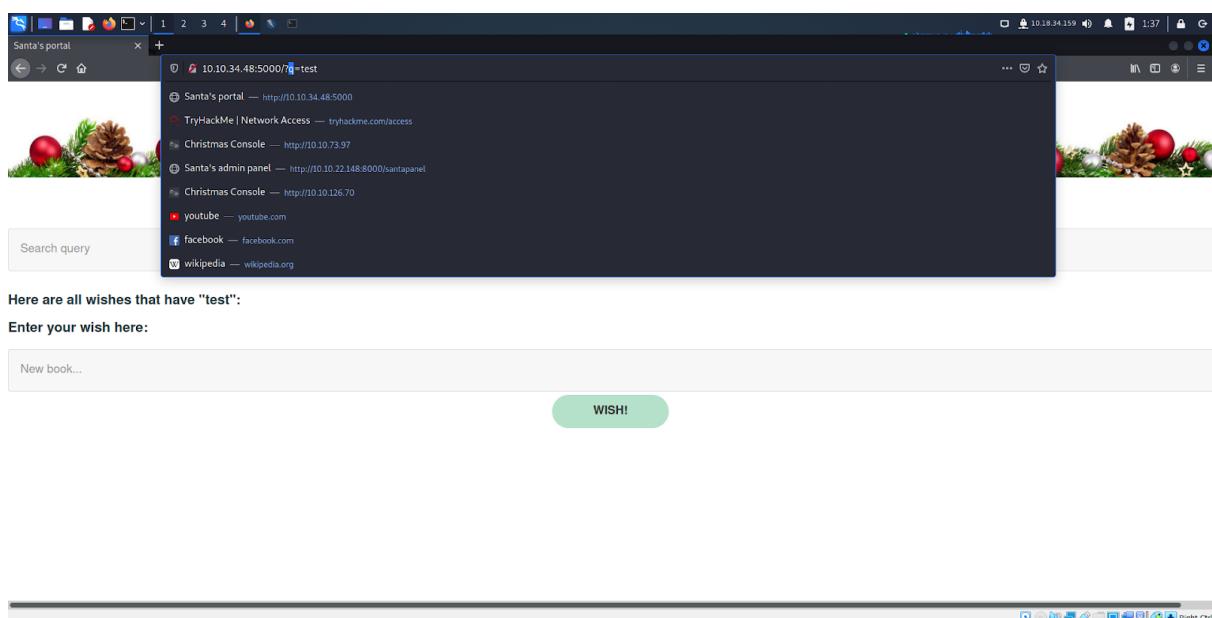
The vulnerability type that was used to exploit the application can be found as shown in the image below (highlighted for viewing purposes).

Stored XSS works when a certain malicious JavaScript is submitted and later on stored directly on the website. For example, comments on a blog post, user nicknames in a chat room, or contact details on a customer order. In other words, in any content that persistently exists on the website and can be viewed by victims.

Question 4:

What query string can be abused to craft a reflected XSS?

In order to find the query string, one must input any type of text in the 'Search query' section. Entering the typed text will pop up the query string in the search bar on top of the website as shown in the image below.



Question 5:

Run a ZAP (zaproxy) automated scan on the target. How many XSS alerts of high priority are in the scan?

Click the 'Automated Scan' option and type in the website's address in the 'URL to attack' section. After doing so, press the 'Attack' button to start the attack on the website.

<

Automated Scan



This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack:

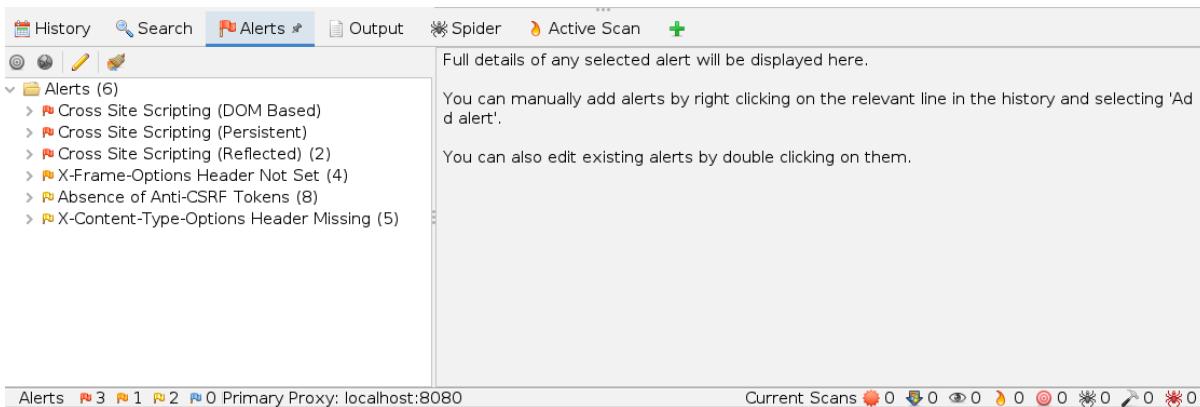
Use traditional spider:

Use ajax spider: with

 Attack

Progress: Attack complete - see the Alerts tab for details of any issues...

After waiting for a few seconds, click on the 'Alerts' tab to identify how many XSS alerts of high priority are available in the scan.



History Search Alerts * Output Spider Active Scan +

Full details of any selected alert will be displayed here.

Alerts (6)

- >  Cross Site Scripting (DOM Based)
- >  Cross Site Scripting (Persistent)
- >  Cross Site Scripting (Reflected) (2)
- >  X-Frame-Options Header Not Set (4)
- >  Absence of Anti-CSRF Tokens (8)
- >  X-Content-Type-Options Header Missing (5)

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

Alerts  3  1  2  0 Primary Proxy: localhost:8080 Current Scans  0  0  0  0  0  0  0  0

Question 6:

What Javascript code should you put in the wish text box if you want to show an alert saying "PSP0201"?

The answer for this particular question can be found either through Google or having Javascript knowledge. As shown in the image below, typing in `<script>alert('PSP0201')</script>` is the correct way of making 'PSP0201' appears as the wish.

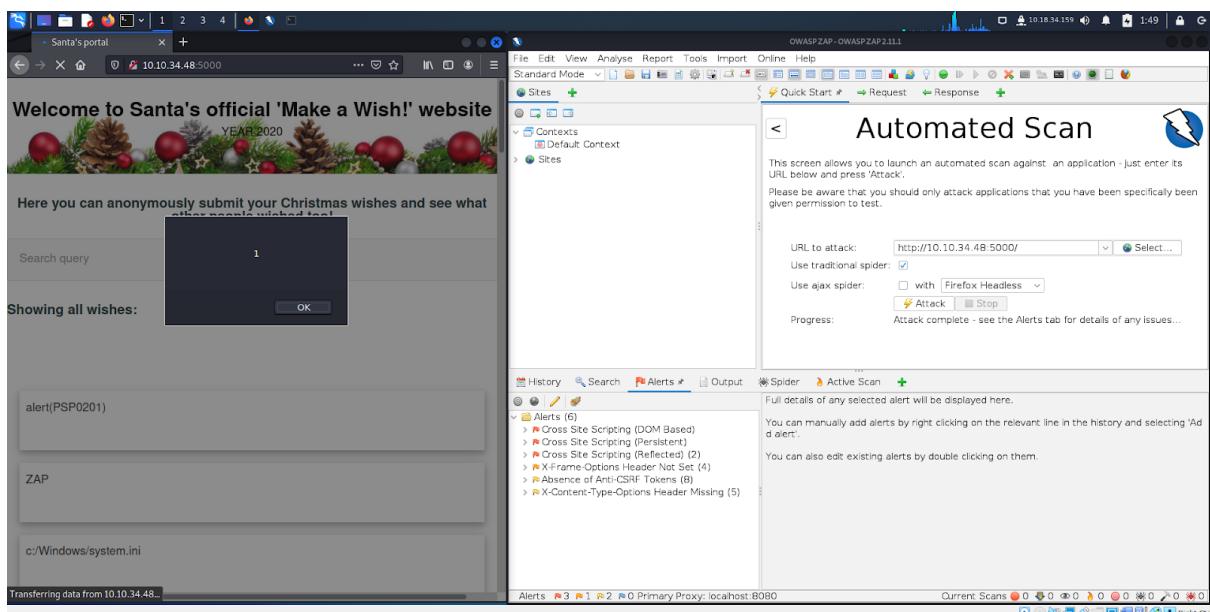
What is alert function in JavaScript?

The **alert()** method displays an alert box with a message and an OK button. The **alert()** method is used when you want information to come through to the user.

Question 7:

Close your browser and revisit the site MACHINE-IP:5000 again. Does your XSS attack persist?

Closing and revisiting the website again will make the text in the black box appear. Therefore, the XSS attack still persists.



Thought Process/Methodology:

Based on question 1 and question 2, the answer for those two particular questions can be found by looking through OWASP Cheat Sheet. As for question 3, the hint for the answer can be found in the THM website under Day 6 section. After opening up the website by typing in the IP Address:5000 in the search bar, we are exposed to the page titled 'Santa's Portal'. In order to find the query string that we can abuse, we must type in anything that we want in the 'Search query' section and press Enter. The query string should appear in the search bar (q). Finding the amount of XSS alerts of high priority require us to start an attack on the website with Zaproxy. After clicking the 'Automated Scan' option, we are able to type in the URL link to attack. After doing so, we can simply go to the 'Alerts' tab to find the amount of XSS alerts asked in the question. Question 6's answer can be found by doing a bit of researching through

Google. The correct function to make 'PSP0201' appear in the wish text box should be alert(). Last but not least, we can confirm that the XSS attack persist by closing and revisiting the website as a black box with the text '1' pops up in the main page.

Day 7: Networking - The Grinch Really Did Steal Christmas

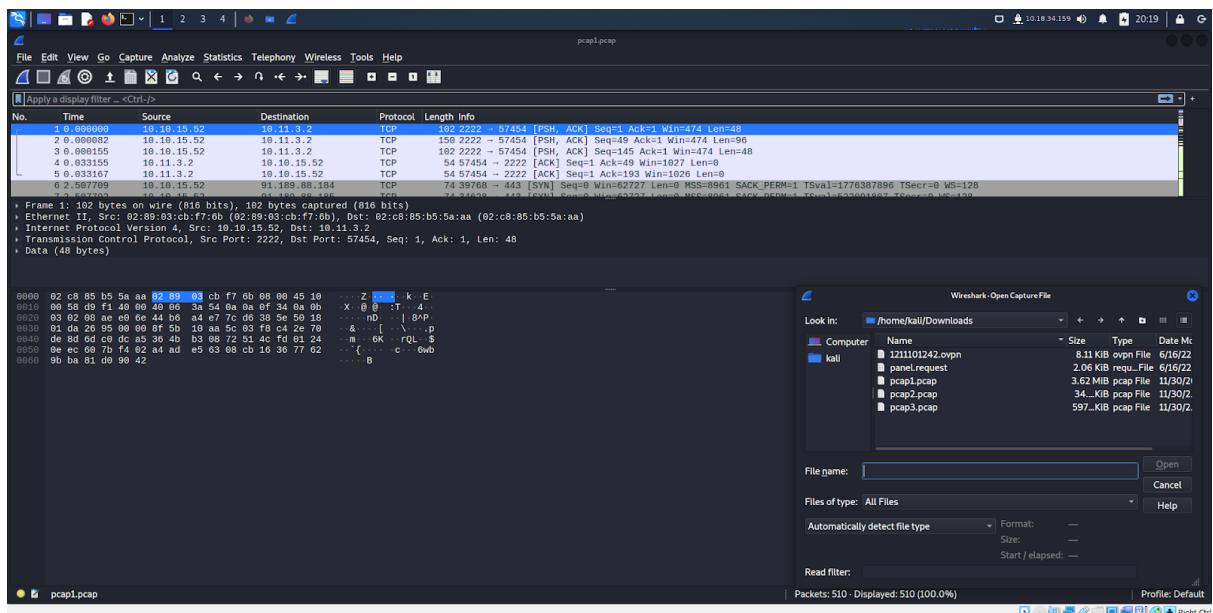
Tools used: Kali Linux/THM Task Files/Wireshark

Solution/Walkthrough:

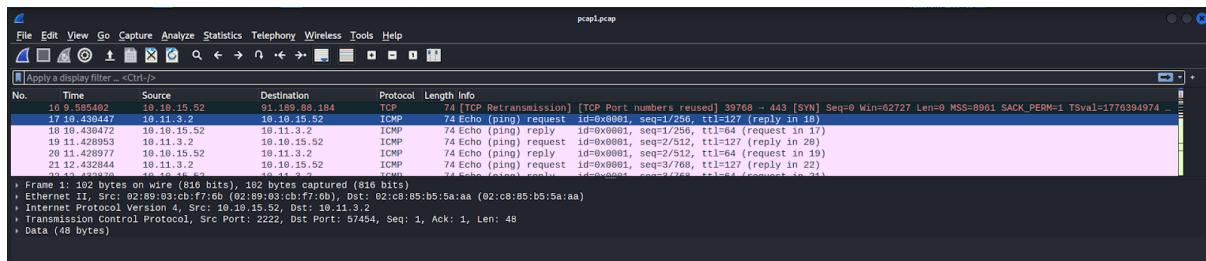
Question 1:

Open "pcap1.pcap" in Wireshark. What is the IP address that initiates an ICMP/ping?

By default, Wireshark is not installed in Kali Linux. Therefore, use the following command 'sudo apt install wireshark' to install Wireshark in order to complete this task. After doing so, open up Wireshark and use the 'drag and drop' method to open up the downloaded task file from THM which is 'pcap1.pcap' to capture it.



After capturing the file, scroll down until you see the first IP address that initiates an ICMP/ping



Question 2:

If we only wanted to see HTTP GET requests in our "pcap1.pcap" file, what filter would we use?

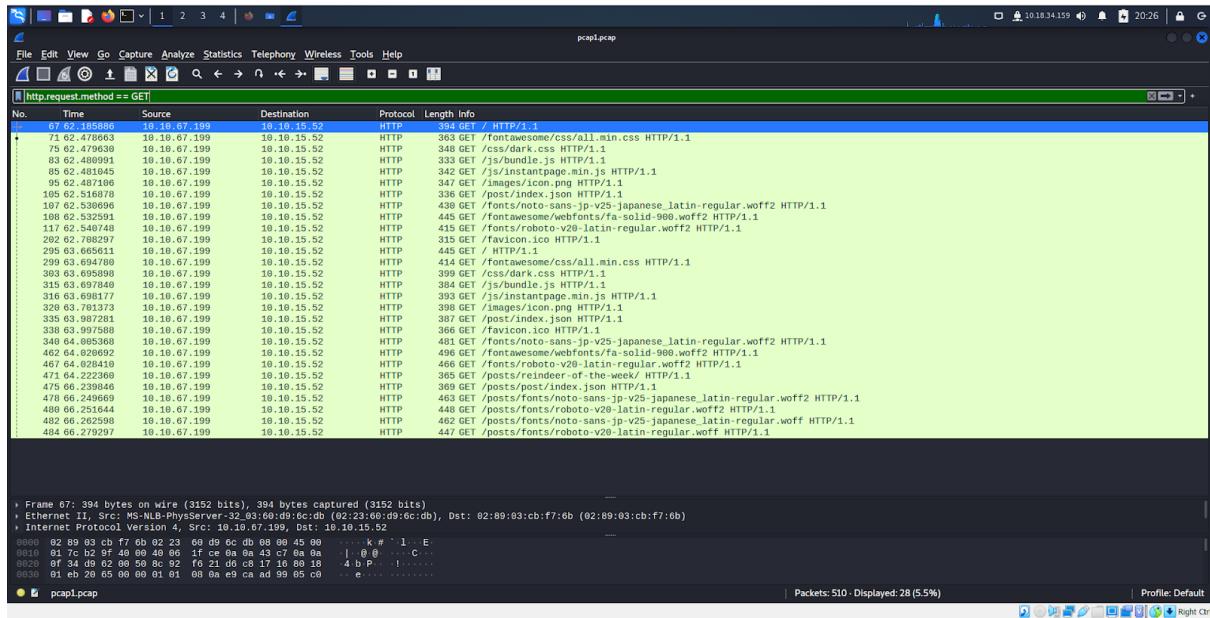
The appropriate filter that we would use to see HTTP GET requests in our "pcap1.pcap" file should be `http.request.method == GET` as shown in the image below.

Networks are, however, rather noisy...Wireshark captured 2,648 packets after a single minute on my machine. This makes analysing very hard. Thankfully, we can use filters to narrow down the results. We can filter by many things, but we'll only cover a couple of important ones in the table below. Note that all the examples below use the <code>==</code> operator to see if the filter exactly matches the value we give it.		
Filter	Description	Example
<code>ip.src</code>	Show all packets that originate from the specified IP address	<code>ip.src == 192.168.1.1</code>
<code>ip.dst</code>	Show all packets that are destined to the specified IP address	<code>ip.dst == 192.168.1.1</code>
<code>tcp/udp.port</code>	Show all packets that are sent via the protocol and port specified	<code>tcp.port == 22 / udp.port == 67</code>
<code>protocol.request.method</code>	Show all packets that use a specific method of the protocol given. For example, HTTP allows for both a <code>GET</code> and <code>POST</code> to retrieve and submit data accordingly.	<code>http.request.method == GET / POST</code>

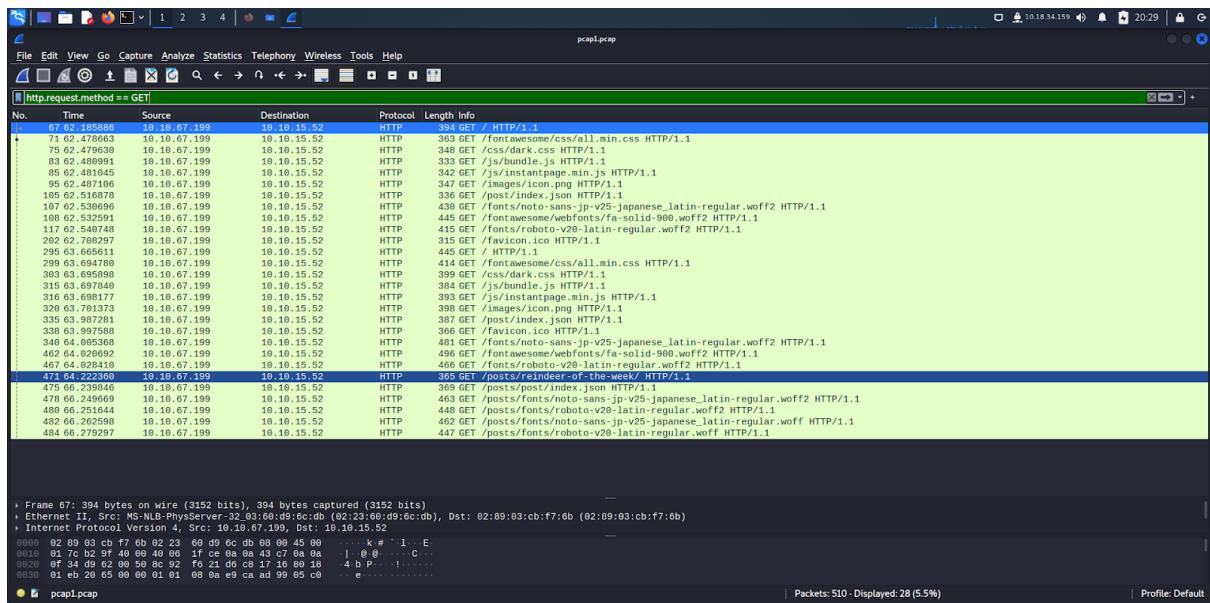
Question 3:

Now apply this filter to "pcap1.pcap" in Wireshark, what is the name of the article that the IP address "10.10.67.199" visited?

Type in the following filter in Wireshark to get all of the information as shown in the image below.



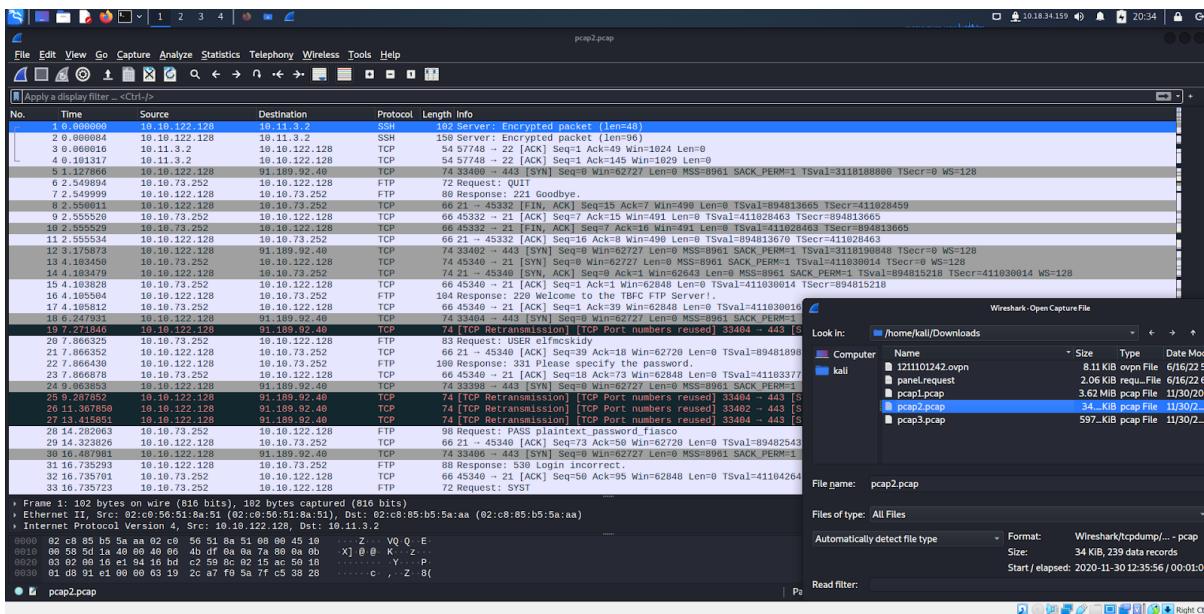
The name of the article that the IP address "10.10.67.199" visited should be as shown in the image below.



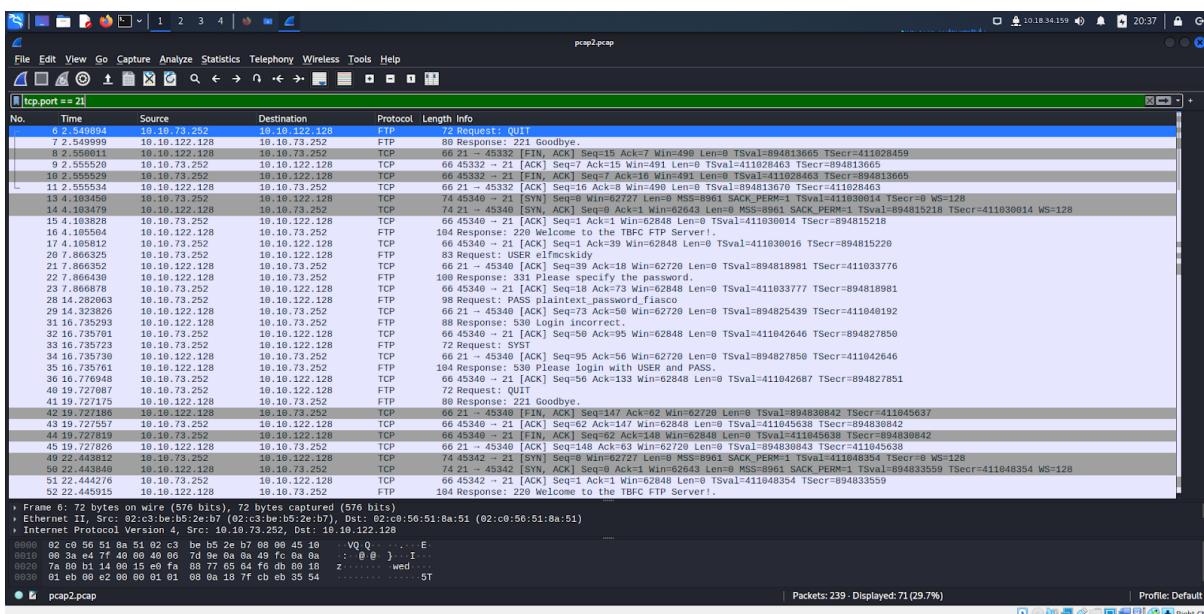
Question 4:

Let's begin analysing "pcap2.pcap". Look at the captured FTP traffic; what password was leaked during the login process?

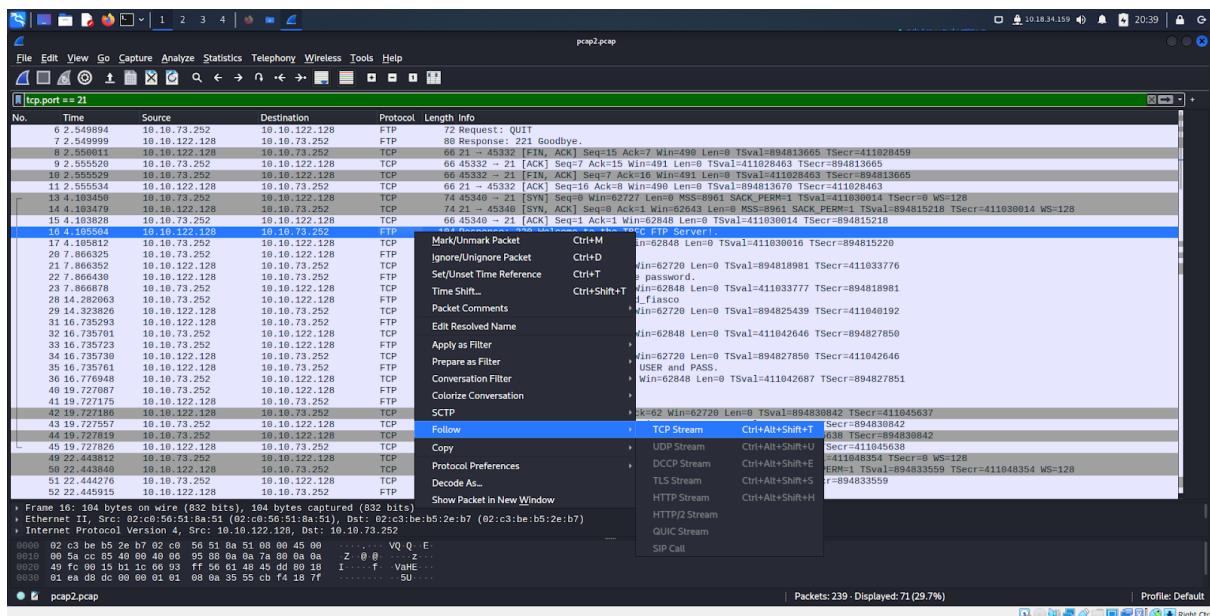
Open up "pcap2.pcap" by using the 'drag and drop' method.



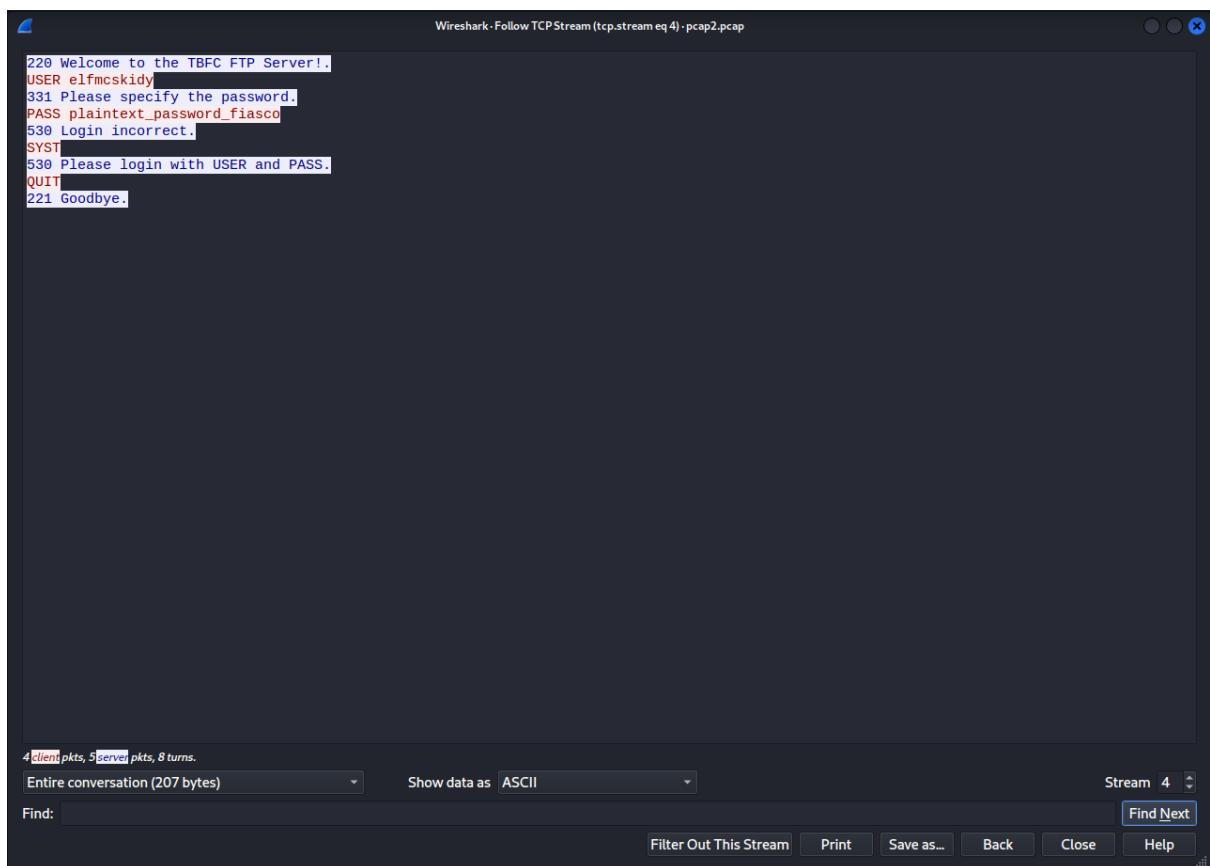
Use the following filter to get rid of the irrelevant data and make our finding much easier.



Find the FTP traffic that showed user login successful attempt. Right click the FTP traffic, click Follow and click TCP stream to find the leaked password.



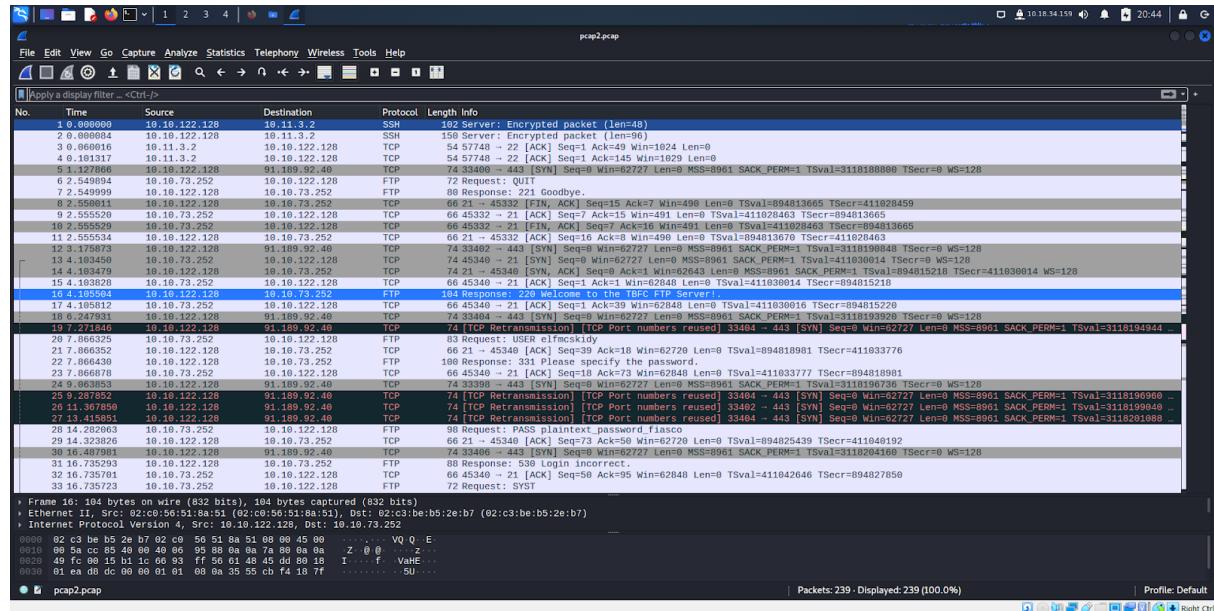
A new pop up should appear with the information needed as shown in the image below.



Question 5:

Continuing with our analysis of "pcap2.pcap", what is the name of the protocol that is encrypted?

The protocol that is encrypted can be found by removing the filter used in the previous question and by identifying the first result in Wireshark as shown in the image below.

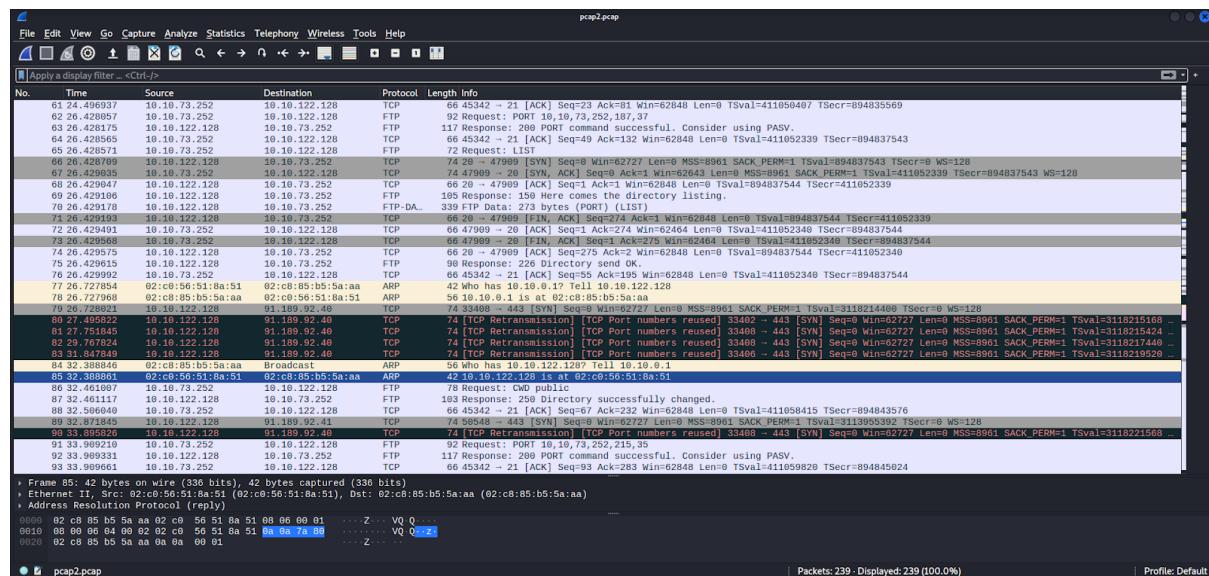


Question 6:

Examine the ARP communications. Who has 10.10.122.128? Tell 10.10.10.1.

Answer: 10.10.122.128 is at

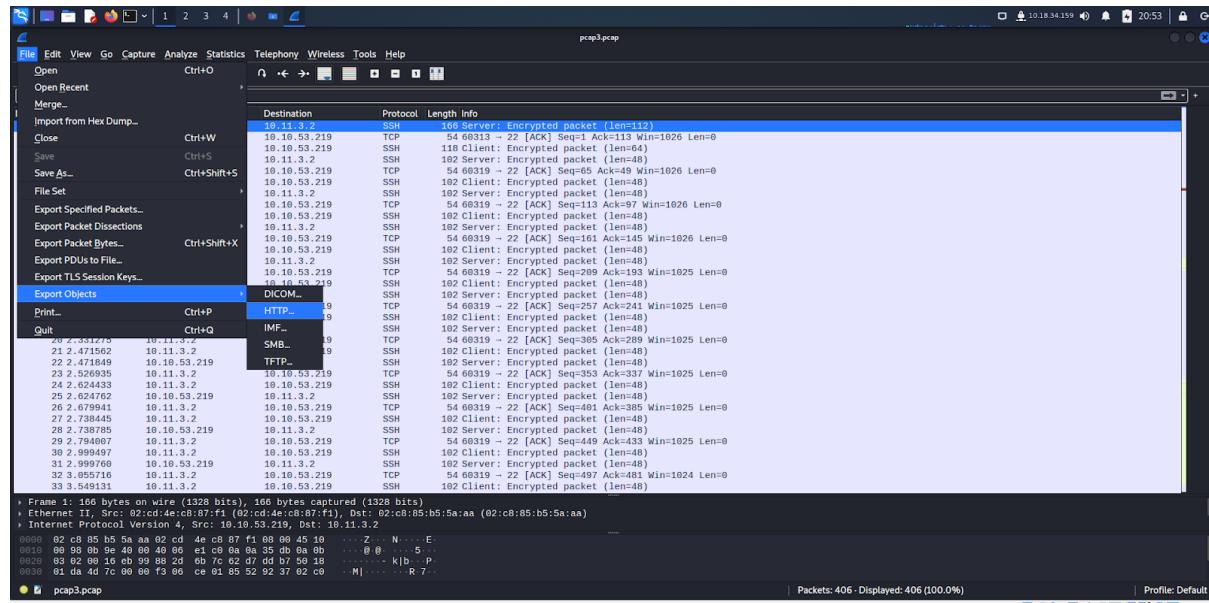
Scroll down the find the correct ARP communications with 'Who has 10.10.122.128? Tell 10.10.10.1.' as its info. The answer can be found on the second ARP communications as shown in the image below.



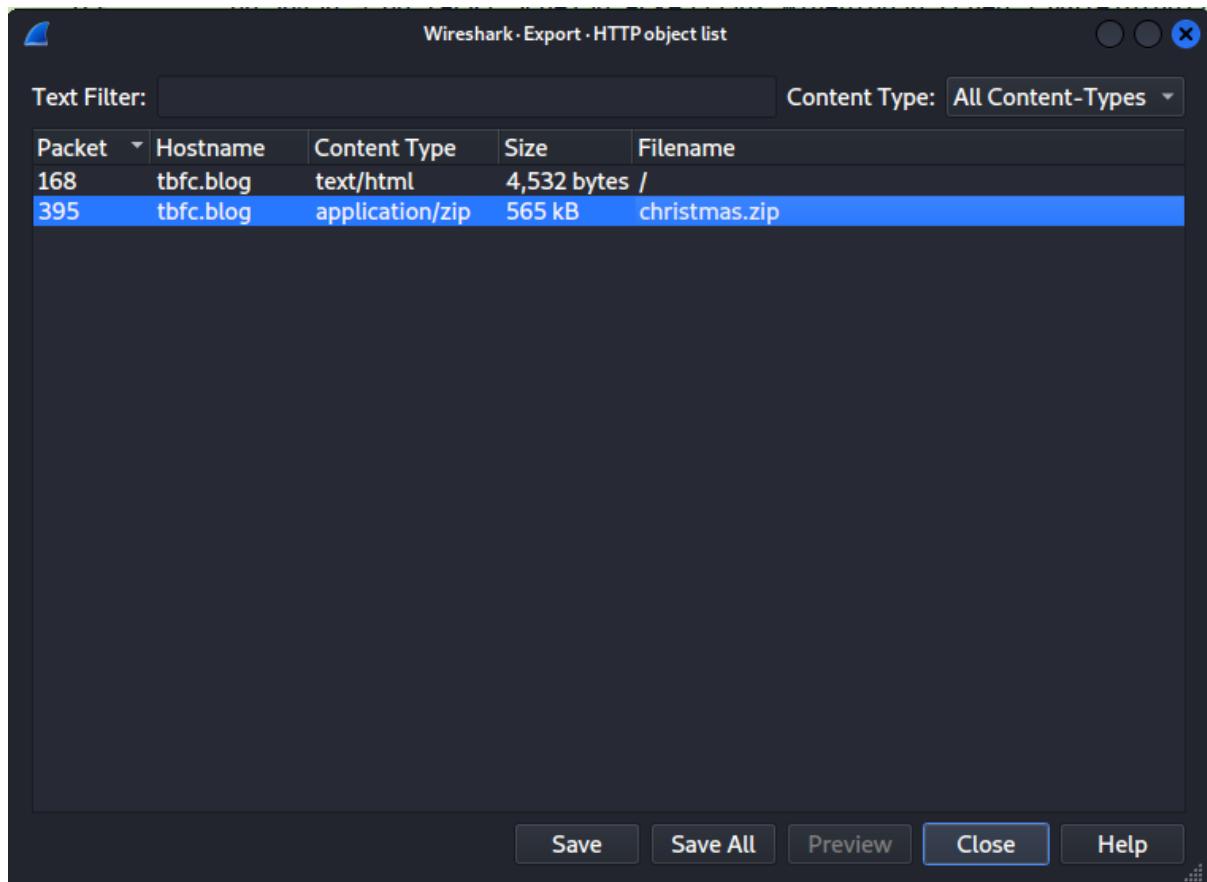
Question 7:

Analyse "pcap3.pcap" and recover Christmas! What is on Elf McSkidy's wishlist that will be used to replace Elf McEager?

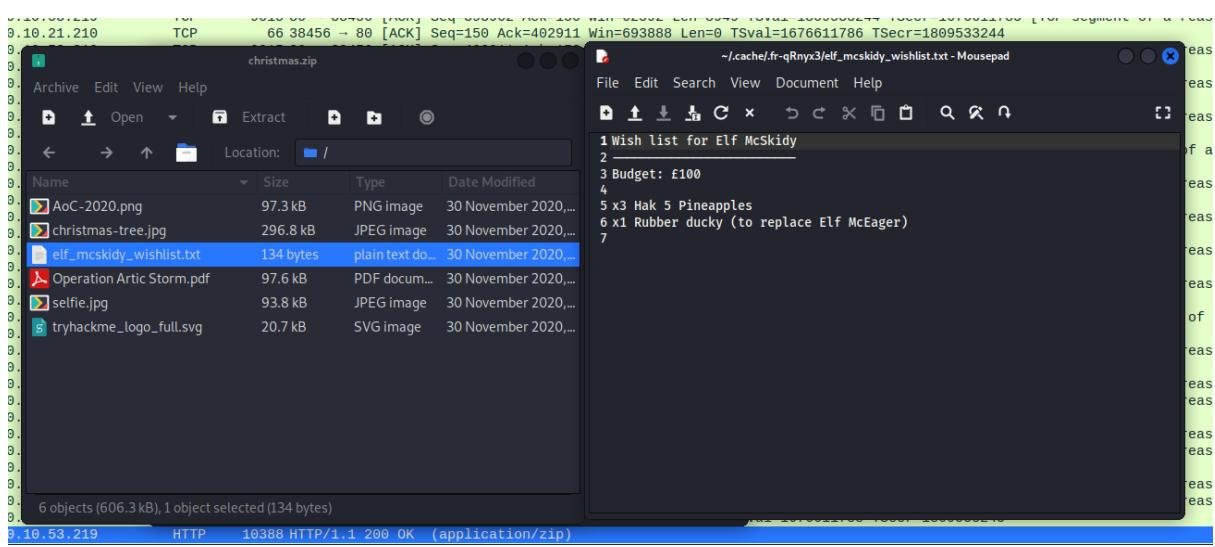
To make our job easier, go to File -> Export Objects -> HTTP.



Save the highlighted zip file in any directory one prefers.



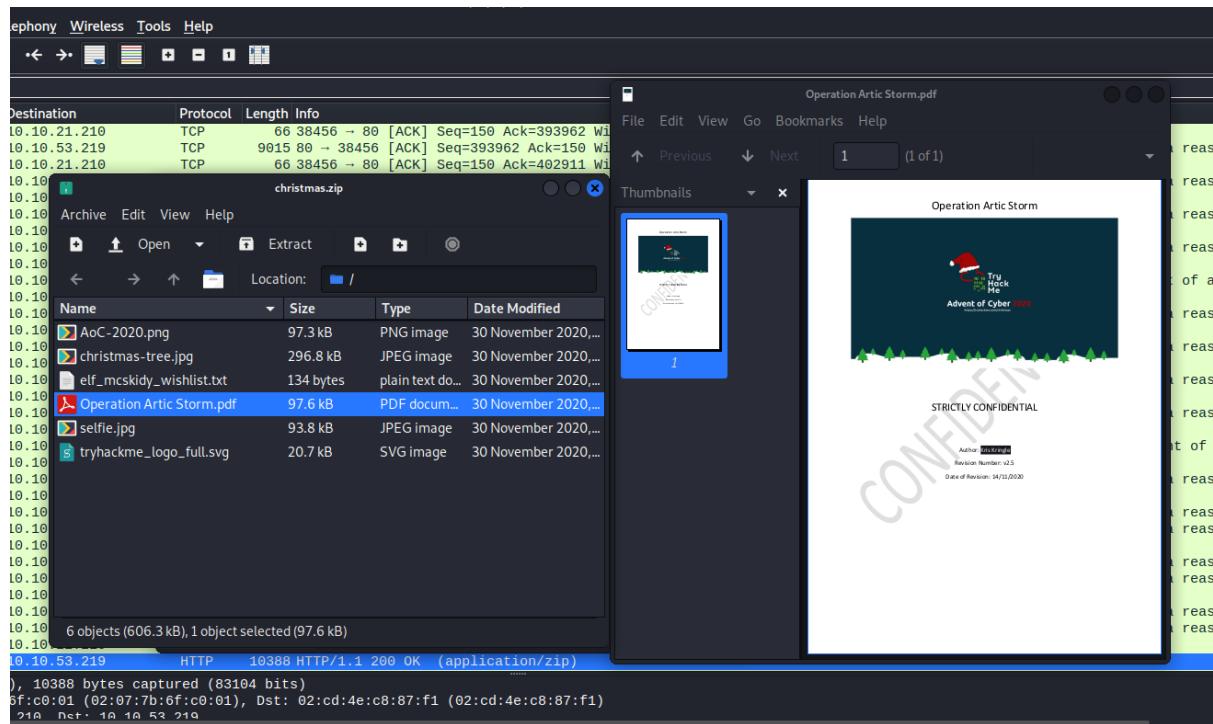
Open up the saved zip file and open the `elf_mcskidy_wishlist.txt` file to find the answer as shown in the image below.



Question 8:

Who is the author of Operation Artic Storm?

In the same directory, open up the Operation Artic Storm.pdf pdf file to find the author's name as show in the image below.



Thought Process/Methodology:

In order to get the definite answer for question 1, we need to install Wireshark in Kali Linux as it is not installed by default. This can be simply done by going to the terminal and typing in the following command 'sudo apt install wireshark'. After doing so, open up Wireshark and use the 'drag and drop' method to open up the 'pcap1.pcap' file found by downloading the Task Files given in THM to capture it. After opening up the file, scroll down until you see the first IP address that initiates an ICMP/ping (10.11.3.2 should be the answer). As for question 2, the answer can be found by reading through Day 7 as the hint is already given in one of the statements/tables (`http.request.method == GET` should be the answer). As for question 3, we need to type in the filter we obtained from previous question to find the correct article that the IP Address '10.10.67.199' visited. After doing so, scroll down until you see reindeer-of-the-week as the answer. As for question 4, first and foremost, we need to open up the second file which is 'pcap2.pcap' by using the 'drag and drop' method. Since there is a lot of data irrelevant to the question asked, we can use the following filter 'tcp.port == 21' to make our finding much easier. Find the FTP traffic that showed a successful user login attempt. Right click the FTP traffic, click Follow and click TCP stream to find the leaked password. A new pop up will appear containing the password (plaintext_password_fiasco should be the answer). As for question 5, we need to remove the filter used in the previous question and the first

result should be the protocol we are looking for. As for question 6, scroll down the find the correct ARP communications with 'Who has 10.10.122.128? Tell 10.10.10.1.' as its info. The answer can be found on the second ARP communications just below the first one we identified which is 02:c0:56:51:8a:51. As for question 7 and question 8, both of these can be done simultaneously by going to File -> Export Objects -> HTTP to narrow down the possible file we need the most. Save the christmas.zip file in any directory one prefers and open it up to see both the elf_mcskidly_wishlist.txt (open it up to obtain the answer for question 7) and Operation Artic Storm.pdf (open it up to obtain the answer for question 8) files.

Day 8: Networking - What's Under the Christmas Tree?

Tools used: THM Machine/Kali Linux/Mozilla Firefox/Nmap

Solution/Walkthrough:

Question 1

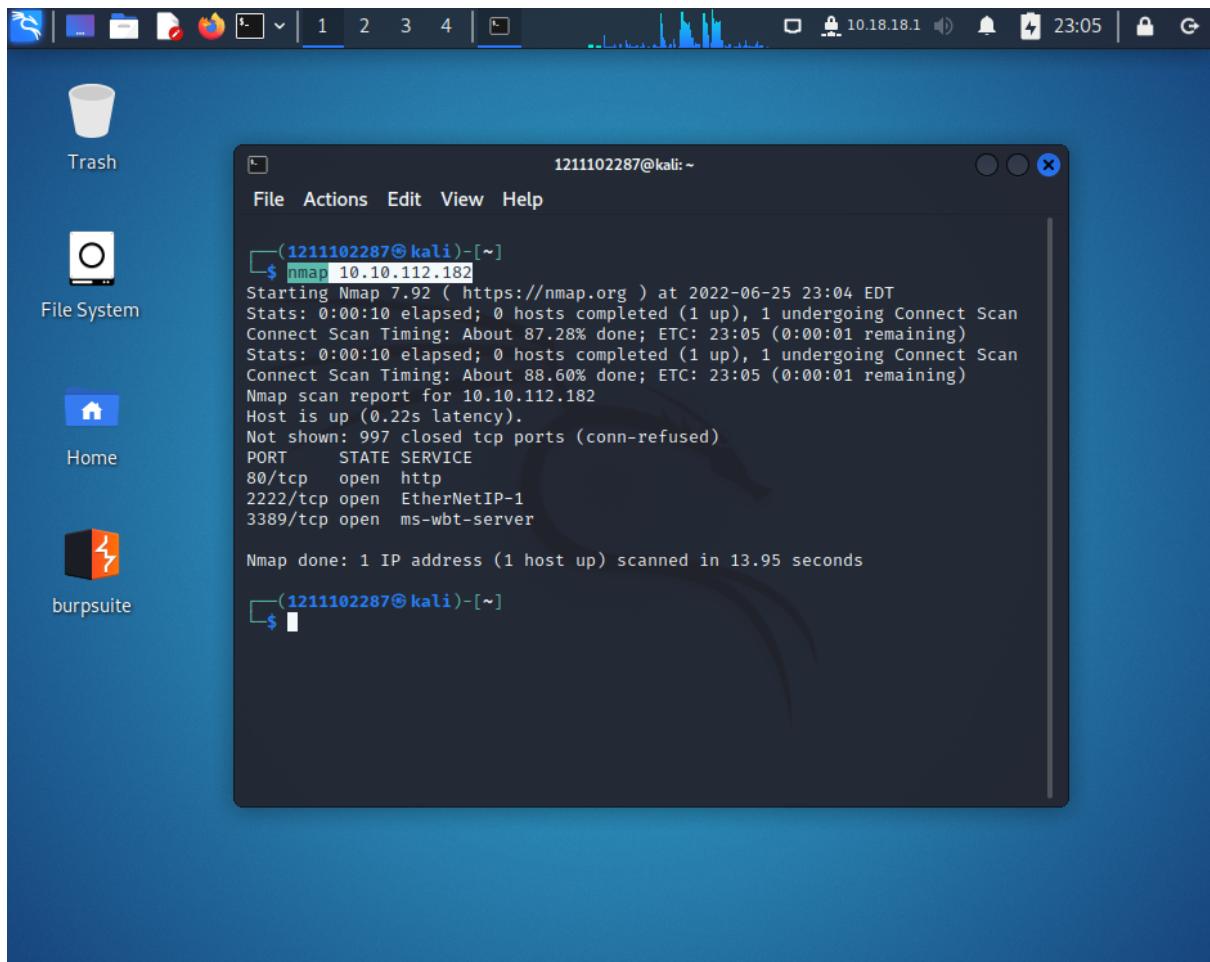
When was Snort created?

A screenshot of a Google search results page. The search query "when was snort created" is entered in the search bar. Below the search bar, there are navigation links for All, Images, News, Videos, Shopping, More, and Tools. A message indicates "About 3,140,000 results (0.38 seconds)". The top result is a large blue box containing the year "1998". To the right of the year is a small image of a cartoon dog wearing a yellow "SNORT" shirt. The rest of the search results are cut off by a horizontal line.

By searching on Google, Snort can be known to be created in the year **1998** which is highlighted in the picture.

Question 2

Using Nmap on MACHINE_IP , what are the port numbers of the three services running?



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is "1211102287@kali:~". The terminal displays the following Nmap scan output:

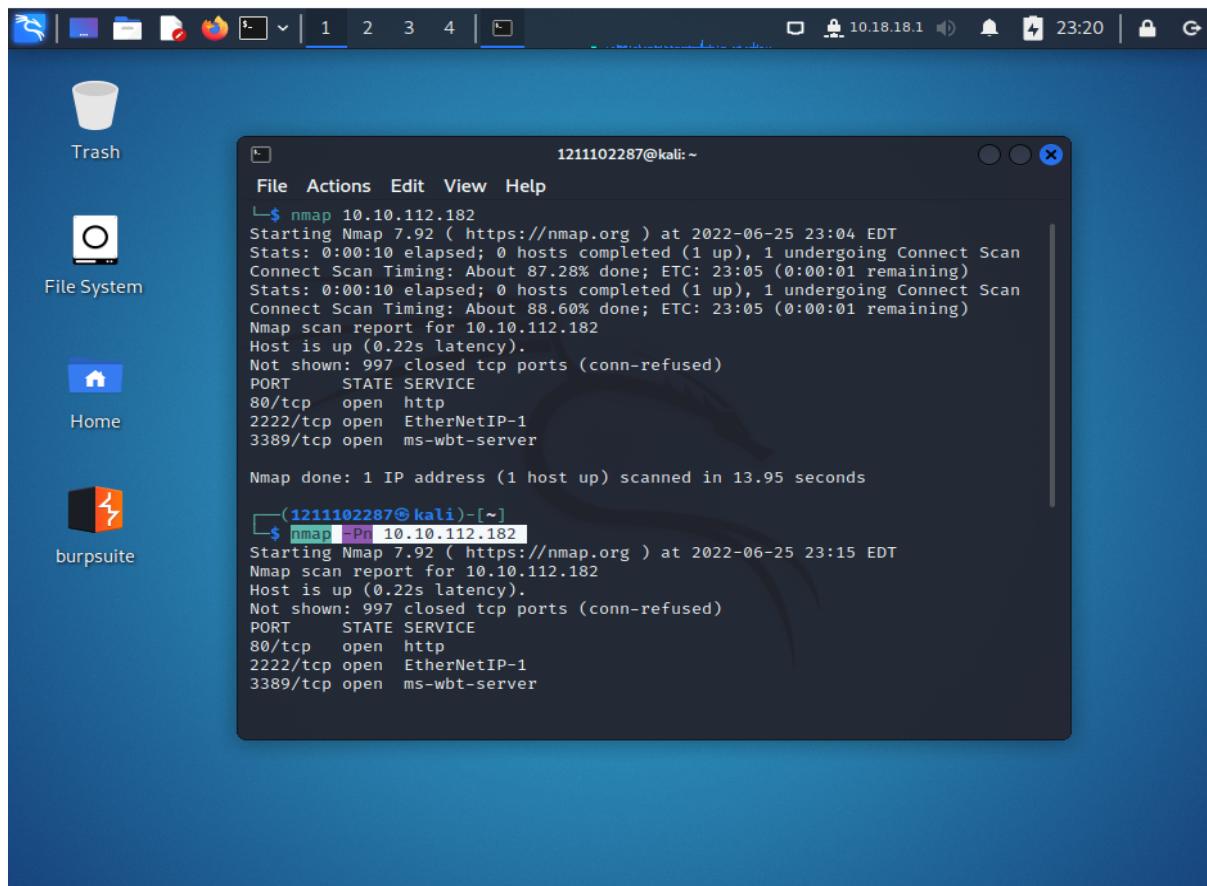
```
(1211102287@kali)-[~]
$ nmap 10.10.112.182
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 23:04 EDT
Stats: 0:00:10 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 87.28% done; ETC: 23:05 (0:00:01 remaining)
Stats: 0:00:10 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 88.60% done; ETC: 23:05 (0:00:01 remaining)
Nmap scan report for 10.10.112.182
Host is up (0.22s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 13.95 seconds
(1211102287@kali)-[~]
$
```

Insert the command nmap followed by the IP address (10.10.112.182) in the terminal window to run a scan and find the Services which are running. The Port numbers could be seen in the terminal (**80.2222.3389**).

Question 3

Use Nmap to determine the name of the Linux distribution that is running, what is reported as the most likely distribution to be running?



The -Pn flag is used to scan and determine whether the host is up.

```
1211102287@kali:~
```

```
File Actions Edit View Help
```

```
1211102287@kali: ~ × 1211102287@kali: ~ ×
```

```
Nmap scan report for 10.10.112.182
Host is up (0.22s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 25.34 seconds
```

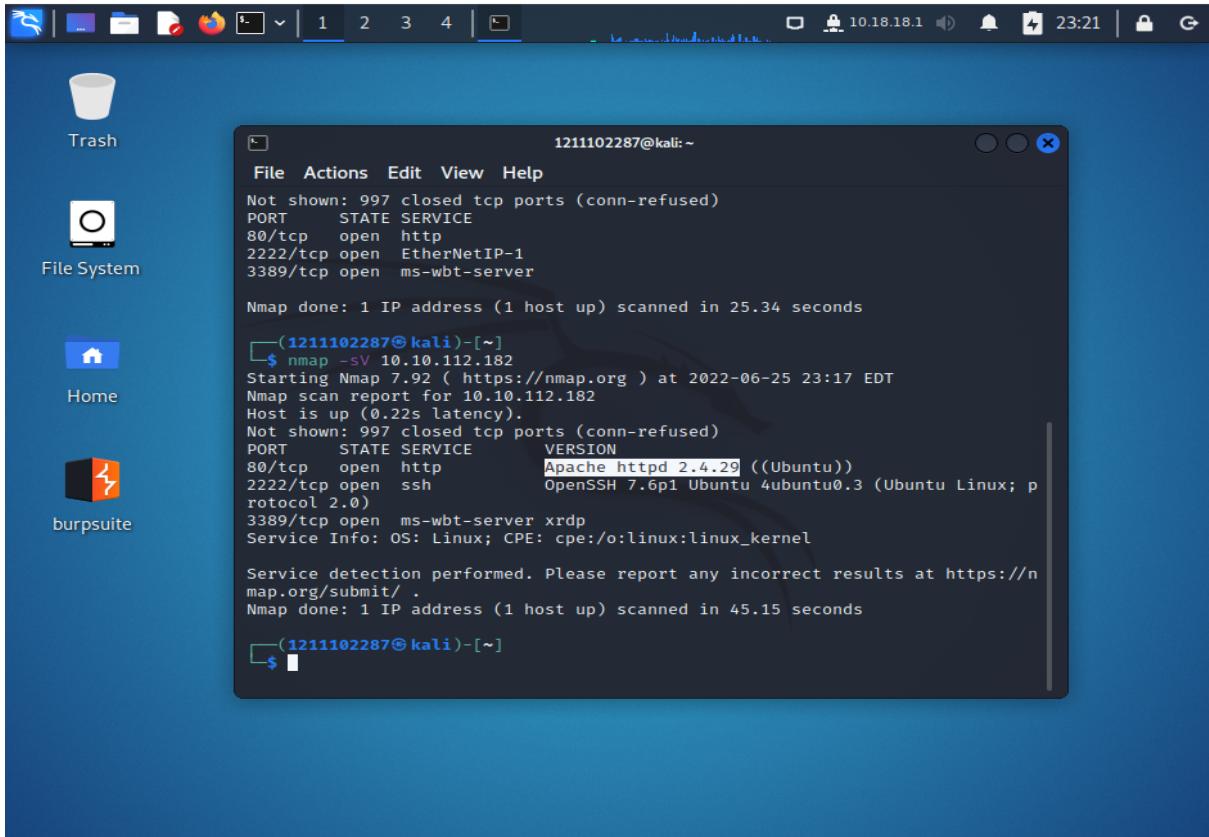
```
—(1211102287㉿kali)-[~]
$ nmap -sV 10.10.112.182
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 23:17 EDT
Nmap scan report for 10.10.112.182
Host is up (0.22s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
2222/tcp  open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; p
rotocol 2.0)
3389/tcp  open  ms-wbt-server xrdp
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
```

By using the Nmap's TCP Scan, the Flag **-sV** used after the command nmap allows us to scan the host and perform version fingerprinting. The name (highlighted) in the terminal window is the name of the Linux distribution that is running.

Question 4

What is the version of Apache?



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is "1211102287@kali:~". The terminal content displays the output of an Nmap scan:

```
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 25.34 seconds

└─(1211102287㉿kali)-[~]
$ nmap -sV 10.10.112.182
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 23:17 EDT
Nmap scan report for 10.10.112.182
Host is up (0.22s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
2222/tcp  open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; p
rotocol 2.0)
3389/tcp  open  ms-wbt-server xrdp
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

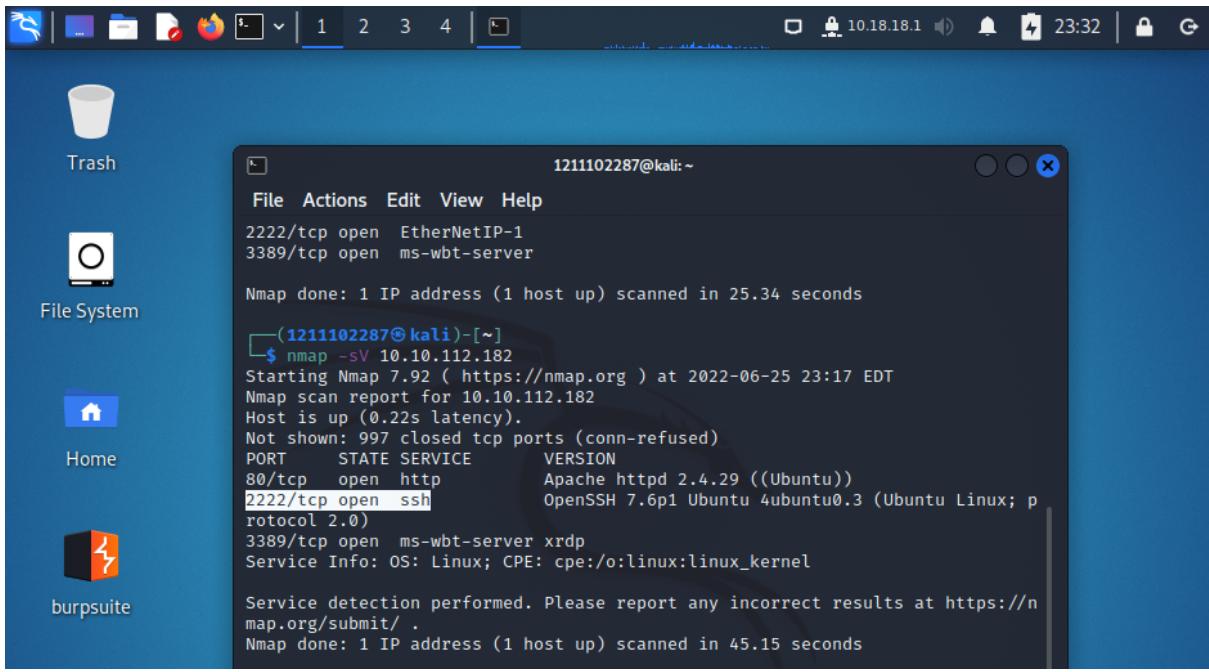
Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 45.15 seconds

└─(1211102287㉿kali)-[~]
$
```

The Apache version is shown in the terminal window (which is being highlighted)

Question 5

What is running on port 2222?



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is "1211102287@kali:~". The terminal displays the following Nmap scan output:

```
File Actions Edit View Help
2222/tcp open EtherNetIP-1
3389/tcp open ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 25.34 seconds

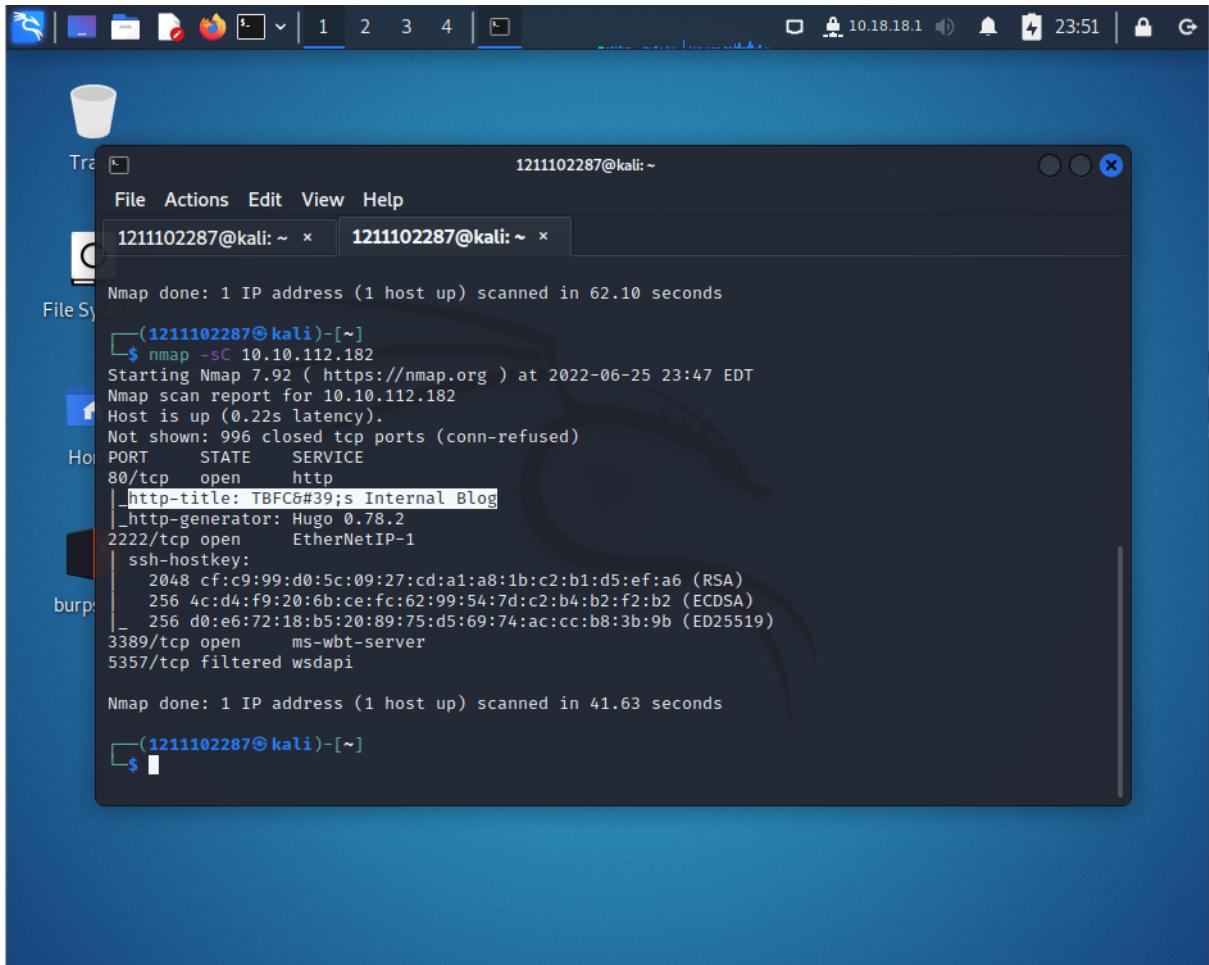
(1211102287@kali)-[~]
$ nmap -sV 10.10.112.182
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 23:17 EDT
Nmap scan report for 10.10.112.182
Host is up (0.22s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
2222/tcp  open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; p
rotocol 2.0)
3389/tcp  open  ms-wbt-server  xrdp
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 45.15 seconds
```

On Port 2222, we can see **SSH** is running on it.

Question 6

Use Nmap's Network Scripting Engine (NSE) to retrieve the "HTTP-TITLE" of the webserver. Based on the value returned, what do we think this website might be used for?



The screenshot shows a terminal window titled 'Terminal' with the command prompt '1211102287@kali:~'. The window displays the output of an Nmap scan. The output shows that port 80/tcp is open and returns the following information:

```
Nmap done: 1 IP address (1 host up) scanned in 62.10 seconds
(1211102287@kali)-[~]
$ nmap -sC 10.10.112.182
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 23:47 EDT
Nmap scan report for 10.10.112.182
Host is up (0.22s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE    SERVICE
80/tcp    open     http
|_http-title: TBFC6#39;s Internal Blog
|_http-generator: Hugo 0.78.2
2222/tcp  open     EtherNetIP-1
| ssh-hostkey:
|   2048 cf:c9:99:d0:5c:09:27:cd:a1:a8:1b:c2:b1:d5:ef:a6 (RSA)
|   256 4c:d4:f9:20:6b:ce:fc:62:99:54:7d:c2:b4:b2:f2:b2 (ECDSA)
|_  256 d0:e6:72:18:b5:20:89:75:d5:69:74:ac:cc:b8:3b:9b (ED25519)
3389/tcp  open     ms-wbt-server
5357/tcp  filtered wsdapi

Nmap done: 1 IP address (1 host up) scanned in 41.63 seconds
(1211102287@kali)-[~]
$
```

With Nmap's Network Scripting Engine (NSE), to do a brief and full scanning or checking on the host to retrieve some information, the script "-sC " is inserted after the command nmap in the terminal window. We could see the HTTP-TITLE of the web server, and based on the value returned, we know the website is used for **Blog**.

Thought Process/Methodology:

The date when SNORT was created could be found through Google searches which is 1998. To scan and find the Services which are running, the command nmap followed by the IP address is used in the terminal window. The Port numbers could be seen in the terminal under the PORT section. From the hint of the question, we could use the -Pn flag to scan and determine whether the host is up. By using the Nmap's TCP Scan, we should know that the Flag -sV is used after the command nmap and it will let us scan the host and perform version fingerprinting. The name in the terminal window will be shown for the Linux distribution that is running. The Apache version is also shown in the terminal window. Scrolling down the terminal, we could see that the Port 2222 is running with SSH. With Nmap's Network Scripting Engine (NSE), to do a brief and full scanning or checking on the host to retrieve some information, the script "-sC " is inserted after the command nmap in the terminal window. We could see the HTTP-TITLE of the web server, and based on the value returned, we know the website is used for Blog.

Day 9 - [Networking] Anyone can be Santa!

Solution/walkthrough

Question 1

What are the directories you found on the FTP site?

The screenshot shows a browser window with several tabs open. The active tab is 'tryhackme.com/room/learncyberin25days'. The content of the page includes instructions for using the 'ftp' command and a terminal session demonstrating the login process.

Instructions:

use `ftp` and provide the IP address of the instance. In my case, I would use `ftp 10.10.185.239`, but you would need to use `ftp 10.10.79.2` for your vulnerable instance.

When prompted for our "Name", we enter "anonymous". If successful, we have confirmed that the `FTP` Server has "anonymous" mode enabled - successful login looking like so:

```
root@ip-10-10-141-42:~# ftp 10.10.185.239
Connected to 10.10.185.239.
220 Welcome to the TBFC FTP Server!.
Name (10.10.185.239:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

You can use the `help` command to list some of the commands you can run whilst connected to the `FTP` Server. Here's a quick rundown of the fundamentals:

Command	Description
<code>ls</code>	List files and directories in the working directory on the <code>FTP</code> server
<code>cd</code>	Change our working directory on the <code>FTP</code> server
<code>get</code>	Download a file from the <code>FTP</code> server to our device
<code>put</code>	Upload a file from our device to the <code>FTP</code> server

Let's look at the directories available to us using `ls`. There is only one folder with data that our user has permission to access:

```
ftp> ls
```

The terminal session on the right shows the user connecting via `ftp 10.10.79.2` and listing the contents of the directory. The output shows three directories: `backups`, `elf_workshops`, and `human_resources`.

```
root@ip-10-10-47-133:~#
File Edit View Search Terminal Help
root@ip-10-10-47-133:~# ftp 10.10.79.2
Connected to 10.10.79.2.
220 Welcome to the TBFC FTP Server!.
Name (10.10.79.2:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x 2 0 0 4096 Nov 16 2020 backups
drwxr-xr-x 2 0 0 4096 Nov 16 2020 elf_workshops
drwxr-xr-x 2 0 0 4096 Nov 16 2020 human_resources
226 Directory send OK.
ftp> 
```

The directories we can find on the ftp site right after we type the “anonymous” are `backups`, `elf_workshops` and `human_resources`.

Question 2

Name the directory on the FTP server that has data accessible by the "anonymous" user

The screenshot shows a browser window with several tabs open, including WhatsApp, Courses, GitHub, PSP0201 T2130, TryHackMe, TryHackMe Adv, and Week 3 Write-Up. Below the tabs, there is a snippet of text from a document:

9.2. Today's Learning Objectives:
Understand the fundamentals of an FTP file server and some common misconfigurations to ultimately exploit these ourselves to gain entry to *tbfc-ftp-01*.

9.3. What is FTP & Where is it Used?
The File Transfer Protocol (FTP) offers a no-thrills means of file sharing in comparison to alternative protocols available. Whilst this protocol is unencrypted, it can be accessed through a variety of means; from dedicated software like FileZilla, the command line, or web browsers, FTP Servers have been long used to share files between devices across the Internet due to its compatibility.

Accessing an FTP server using the Mozilla Firefox Web Browser.

FTP uses two connections when transferring data, as illustrated below:

The right side of the screenshot shows a terminal window titled "root@ip-10-10-27-238:~". The terminal displays the following output:

```
root@ip-10-10-27-238:~$ ls
bell      glob      mode     quote    suniq
ue        hash      modtime  recv     tenex
binary   help      mput    reget    tick
bye      idle      newer   rstatus  trace
case     image     nmap    rhelp   type
cd       ipany    nlist   rename  user
cdup    ipv4     ntrans  reset   umask
chmod   ipv6     open    restart  verbose
close
cr      lcd      prompt  rmdir   ?
delete  ls      passive runque
debug   macdef  proxy
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x 2 0      0          4096 Nov 16 2020 backups
drwxr-xr-x 2 0      0          4096 Nov 16 2020 elf_workshops
drwxr-xr-x 2 0      0          4096 Nov 16 2020 human_resources
drwxrwxrwx 2 65534 65534      4096 Nov 16 2020 public
226 Directory send OK.
ftp> 
```

The terminal window also shows the status "42m 39s" at the bottom right.

The name of the directory on the ftp server is public which has data accessible by the “anonymous” user.

Question 3

What script gets executed within this directory?

The screenshot shows a browser window with several tabs open. On the left, a terminal window displays a listener on port 4444, receiving a connection from 10.10.185.239. It also shows an attempt to set a terminal process group and a failure due to inappropriate ioctl for device. The user is root at tbfc-ftp-01. On the right, a file manager window titled 'root@ip-10-10-27-238: ~' shows a directory listing. The 'backup.sh' file is highlighted. The terminal window below shows an FTP session where the user retrieves 'backup.sh' and 'shoppinglist.txt'. The file manager window shows the contents of 'backup.sh' and 'shoppinglist.txt'. The status bar at the bottom indicates '22m 30s'.

Proceed to use commands similar to what we have used before to find the contents of root.txt located in the root directory! Let's break down exactly what happened here and explain the reasons as why this exploit happened:

9.6.5.1. The FTP Server has anonymous mode enabled allowing us to authenticate. This isn't inherently insecure and has many legitimate uses.

9.6.5.2. We've discovered that we have permission to upload and download files. Whilst is also normal behaviour for FTP servers, anonymous users should not be able to upload files.

9.6.5.3. We've interpreted the information from a legitimate backup script to create a reverse shell onto our host.

9.6.5.4. The script executes as the "root" user - the most powerful on a Linux system. This is also a vulnerability, as now we have full access to the system. The use of this user should be restricted wherever possible. If the script were to execute as "elfmeager", we'd only have access to the system as that user (a much less powerful one in comparison)

9.7. Conclusion, where to go from here and additional Material:
We've covered the fundamentals of FTP servers and why they're still used today. Not only this, but we've also learned how simple misconfigurations can lead to a full-blown

The backup.sh script gets executed within this directory as per highlighted in the terminal.

Question 4

What movie did Santa have on his Christmas shopping list?

The screenshot shows a web browser window with several tabs open. The active tab is titled "Active Machine Information" and displays details about a machine named "aoc20cmnftp" with IP address 10.10.248.111, which expires in 1h 29m 13s. It includes buttons for "Add 1 hour" and "Terminate". Below this is a progress bar at 63%. A list of tasks follows:

- Task 1 ✓ Introduction
- Task 2 ✓ Get Connected
- Task 3 ✓ [Day 1] Web Exploitation A Christmas Crisis
- Task 4 ✓ [Day 2] Web Exploitation The Elf Strikes Back!
- Task 5 ✓ [Day 3] Web Exploitation Christmas Chaos
- Task 6 ✓ [Day 4] Web Exploitation Santa's watching
- Task 7 ✓ [Day 5] Web Exploitation Someone stole Santa's gift list!

To the right of the browser is a terminal window titled "root@ip-10-10-45-192: ~". The terminal shows a successful login as root, the system type as UNIX, and the user as anonymous. It then lists files in the public directory: backup.sh and shoppinglist.txt. The user then runs the command "cat shoppinglist.txt" which outputs "The Polar Express Movie".

The Polar Express was the movie Santa had on his Christmas shopping list.

Question 5

Re-upload this script to contain malicious data (just like we did in section 9.6. Output the contents of /root/flag.txt!

The screenshot shows a web browser interface with several tabs open. The main content area displays 'Active Machine Information' with a title of 'aoc20cmnftp', IP address '10.10.248.111', and an expiration time of '1h 28m 33s'. There are buttons for 'Add 1 hour' and 'Terminate'. Below this, a progress bar is at 63%. A sidebar lists seven tasks: Task 1 (Introduction), Task 2 (Get Connected), Task 3 ([Day 1] Web Exploitation - A Christmas Crisis), Task 4 ([Day 2] Web Exploitation - The Elf Strikes Back!), Task 5 ([Day 3] Web Exploitation - Christmas Chaos), Task 6 ([Day 4] Web Exploitation - Santa's watching), and Task 7 ([Day 5] Web Exploitation - Someone stole Santa's gift list!). To the right, a terminal window titled 'THM AttackBox' shows a root shell on the host 'ip-10-10-45-192'. The user runs 'nc -lvp 4444' to listen on port 4444. A connection from '10.10.248.111' is received. The user then runs 'cat /root/flag.txt' which outputs the flag: 'THM{even_you_can_be_santa}'.

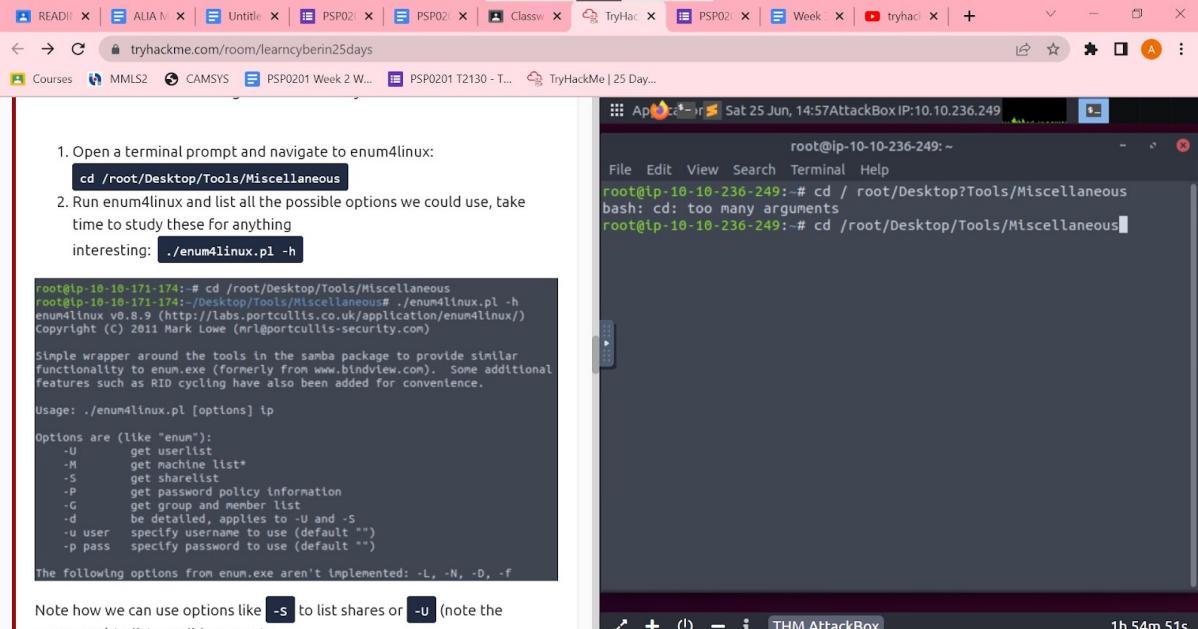
Thought Process/Methodology:

According to question 1 we have to go to the ftp ip address in the terminal and press anonymous and choose ls in order to ding the directories in the ftp site. As for question 2, type in ls in the ftp terminal so that we can get the directory on the FTP server that has data accessible by the "anonymous" user .Next , for question 3, to know what script gets executed within this directory is the get backup.sh directory. According to question 4, to find the movie that was in Santa's shopping list, type in cat shopping list.txt so that we will be able to find the movie which is The Polar Express Movie. For the last question which is about to find the flag, we have to go to the new terminal and type in /root/flag.txt to find out the flag for this question.

Day 10 : Networking - Don't be sElfish!

Tools used : THM Attackbox, Chrome

Question 1 : Examine the help options for enum4linux. Match the following flags with the descriptions.



1. Open a terminal prompt and navigate to enum4linux:
`cd /root/Desktop/Tools/Miscellaneous`

2. Run enum4linux and list all the possible options we could use, take time to study these for anything interesting: `./enum4linux.pl -h`

```
root@ip-10-10-171-174: # cd /root/Desktop/Tools/Miscellaneous
root@ip-10-10-171-174:~/Desktop/Tools/Miscellaneous# ./enum4linux.pl -h
enum4linux v0.8.9 (http://Labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com). Some additional
features such as RID cycling have also been added for convenience.

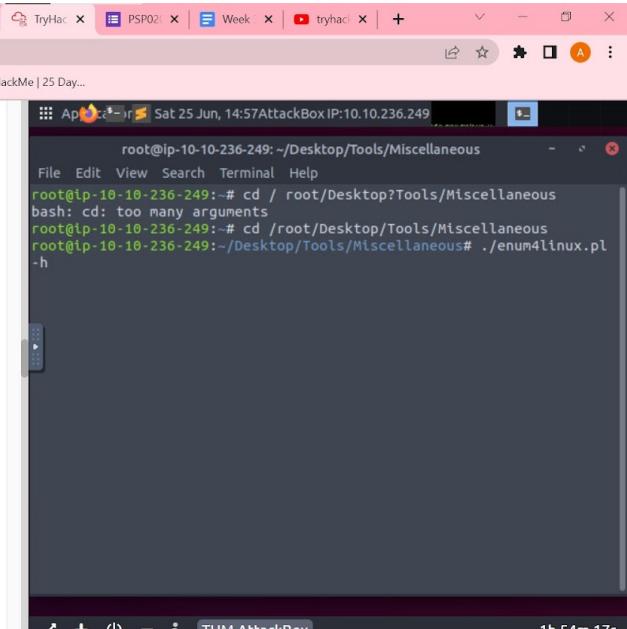
Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
-U      get userlist
-M      get machine list*
-S      get sharelist
-P      get password policy information
-G      get group and member list
-d      be detailed, applies to -U and -S
-u      specify username to use (default "")
-p      specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f
```

Note how we can use options like `-s` to list shares or `-u` (note the uppercase) to list possible users. In

Open /root/Desktop/Tools/Miscellaneous directory



1. Open a terminal prompt and navigate to enum4linux:
`cd /root/Desktop/Tools/Miscellaneous`

2. Run enum4linux and list all the possible options we could use, take time to study these for anything interesting: `./enum4linux.pl -h`

```
root@ip-10-10-171-174: # cd /root/Desktop/Tools/Miscellaneous
root@ip-10-10-171-174:~/Desktop/Tools/Miscellaneous# ./enum4linux.pl -h
enum4linux v0.8.9 (http://Labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com). Some additional
features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
-U      get userlist
-M      get machine list*
-S      get sharelist
-P      get password policy information
-G      get group and member list
-d      be detailed, applies to -U and -S
-u      specify username to use (default "")
-p      specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f
```

Note how we can use options like `-s` to list shares or `-u` (note the uppercase) to list possible users. In

Use enum4linux.pl -h to get help and get the list of command available

1. Open a terminal prompt and navigate to enum4linux:

```
cd /root/Desktop/Tools/Miscellaneous
```

2. Run enum4linux and list all the possible options we could use, take time to study these for anything interesting:

```
./enum4linux.pl -h
```

```
root@ip-10-10-171-174:~# cd /root/Desktop/Tools/Miscellaneous
root@ip-10-10-171-174:~/Desktop/Tools/Miscellaneous# ./enum4linux.pl -h
enum4linux v0.8.9 (http://Labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar functionality to enum.exe (formerly from www.bindview.com). Some additional features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
  -U      get userlist
  -M      get machine list*
  -S      get sharelist
  -P      get password policy information
  -G      get group and member list
  -d      be detailed, applies to -U and -S
  -u user  specify username to use (default "")
  -p pass   specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f
```

Note how we can use options like **-s** to list shares or **-U** (note the uppercase) to list possible users. In

```
root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous
File Edit View Search Terminal Help
enum4linux v0.8.9 (http://Labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar functionality to enum.exe (formerly from www.bindview.com). Some additional features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
  -U      get userlist
  -M      get machine list*
  -S      get sharelist
  -P      get password policy information
  -G      get group and member list
  -d      be detailed, applies to -U and -S
  -u user  specify username to use (default "")
  -p pass   specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f
```

1. Open a terminal prompt and navigate to enum4linux:

```
cd /root/Desktop/Tools/Miscellaneous
```

2. Run enum4linux and list all the possible options we could use, take time to study these for anything interesting:

```
./enum4linux.pl -h
```

```
root@ip-10-10-171-174:~# cd /root/Desktop/Tools/Miscellaneous
root@ip-10-10-171-174:~/Desktop/Tools/Miscellaneous# ./enum4linux.pl -h
enum4linux v0.8.9 (http://Labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar functionality to enum.exe (formerly from www.bindview.com). Some additional features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
  -U      get userlist
  -M      get machine list*
  -S      get sharelist
  -P      get password policy information
  -G      get group and member list
  -d      be detailed, applies to -U and -S
  -u user  specify username to use (default "")
  -p pass   specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f
```

Note how we can use options like **-s** to list shares or **-U** (note the uppercase) to list possible users. In

```
root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous
File Edit View Search Terminal Help
-a      get sharelist
-P      get password policy information
-G      get group and member list
-d      be detailed, applies to -U and -S
-u user  specify username to use (default "")
-p pass   specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f

Additional options:
  -a      Do all simple enumeration (-U -S -G -P -r -o -n -i).
          This option is enabled if you don't provide any other options.
  -h      Display this help message and exit
  -r      enumerate users via RID cycling
  -R range RID ranges to enumerate (default: 500-550,1000-1050, implies -r)
  -K n    Keep searching RIDs until n consecutive RIDs don't correspond to a username. Impies RID range ends at 999999. Useful against DCs.
  -l      Get some (limited) info via LDAP 389/TCP (for DCs only)
  -s file brute force guessing for share names
  -u user User(s) that exists on remote system (default: administrator)
```

1. Open a terminal prompt and navigate to enum4linux:

```
cd /root/Desktop/Tools/Miscellaneous
```

2. Run enum4linux and list all the possible options we could use, take time to study these for anything interesting: ./enum4linux.pl -h

```
root@ip-10-10-171-174:~# cd /root/Desktop/Tools/Miscellaneous
root@ip-10-10-171-174:~/Desktop/Tools/Miscellaneous# ./enum4linux.pl -h
enum4linux v0.8.9 (http://Labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com). Some additional
features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] lp

Options are (like "enum"):
  -U      get userlist
  -M      get machine list*
  -S      get sharelist
  -P      get password policy information
  -G      get group and member list
  -d      be detailed, applies to -U and -S
  -u user  specify username to use (default "")
  -p pass  specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f
```

Note how we can use options like **-s** to list shares or **-u** (note the uppercase) to list possible users. In

1. Open a terminal prompt and navigate to enum4linux:

```
cd /root/Desktop/Tools/Miscellaneous
```

2. Run enum4linux and list all the possible options we could use, take time to study these for anything interesting: ./enum4linux.pl -h

```
root@ip-10-10-171-174:~# cd /root/Desktop/Tools/Miscellaneous
root@ip-10-10-171-174:~/Desktop/Tools/Miscellaneous# ./enum4linux.pl -h
enum4linux v0.8.9 (http://Labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com). Some additional
features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] lp

Options are (like "enum"):
  -U      get userlist
  -M      get machine list*
  -S      get sharelist
  -P      get password policy information
  -G      get group and member list
  -d      be detailed, applies to -U and -S
  -u user  specify username to use (default "")
  -p pass  specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f
```

Note how we can use options like **-s** to list shares or **-u** (note the uppercase) to list possible users. In

Question 2 : Using enum4linux, how many users are there on the Samba server?

The terminal window shows the usage information for the enum4linux.pl script. It includes options for specifying a file (-s), user (-k), and various output and verbosity levels (-o, -i, -w, -n, -v). It also discusses RID cycling and dependency information for samba. The command run was ./enum4linux.pl -U 10.10.81.153.

```
root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous
File Edit View Search Terminal Help
-s file      Set Samba (Windows) share via LDAP, Samba, or WINS (for DCS only)
-k user      User(s) that exists on remote system (default: administrator,guest,krbtgt,domain admins,root,b
in,none)
-o           Used to get sid with "lookupsid known_username"
-i           Use commas to try several users: "-k admin,user1,user2"
-w wrkg      Get OS information
-n           Get printer information
-w wrkg      Specify workgroup manually (usually found automatically)
-n           Do an nmblookup (similar to nbtstat)
-v           Verbose. Shows full commands being run (net, rpcclient, etc.)

RID cycling should extract a list of users from Windows (or Samba) hosts
which have RestrictAnonymous set to 1 (Windows NT and 2000), or "Network
access: Allow anonymous SID/Name translation" enabled (XP, 2003).

NB: Samba servers often seem to have RIDs in the range 3000-3050.

Dependency info: You will need to have the samba package installed as this
script is basically just a wrapper around rpcclient, net, nmblookup and
smbclient. Polenum from http://labs.portcullis.co.uk/application/polenum/
is required to get Password Policy info.

root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous# ./enum4linux.pl -U 10.10.81.153
WARNING: polenum.py is not in your path. Check that polenum is installed and your PATH is correct
```

Use enum4linux.pl -U to get a list of username

The terminal window shows the output of the enum4linux.pl -U command, listing three users: elfmcskidy, elfmceager, and elfmcelferson. The output includes their respective RIDs, ACBs, accounts, and names.

```
root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous
File Edit View Search Terminal Help
=====
| Getting domain SID for 10.10.81.153 |
=====
Domain Name: TBFC-SMB-01
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup
=====
| Users on 10.10.81.153 |
=====
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskidy      Name:
      Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager      Name:
      Desc:
index: 0x3 RID: 0x3e9 acb: 0x00000010 Account: elfmcelferson   Name:
      Desc:

user:[elfmcskidy] rid:[0x3e8]
user:[elfmceager] rid:[0x3ea]
user:[elfmcelferson] rid:[0x3e9]
enum4linux complete on Sat Jun 25 17:02:26 2022
root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous#
```

The number of users in this server is 3

Question 3 : Now how many "shares" are there on the Samba server?

Use enum4linux.pl -s to get a list of the shares that this server has

root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous

```
[+] Attempting to map shares on 10.10.81.153
//10.10.81.153/tbfc-hr Mapping: DENIED, Listing: N/A
//10.10.81.153/tbfc-it Mapping: DENIED, Listing: N/A
//10.10.81.153/tbfc-santa Mapping: OK, Listing: OK
//10.10.81.153/IPCS [E] Can't understand response:
WARNING: The "syslog" option is deprecated
NT_STATUS_OBJECT_NAME_NOT_FOUND listing /*
enum4linux complete on Sat Jun 25 15:17:49 2022

root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous# smbclient //10.10.81.153
```

The list of share this server has is 4

Question 4: Use smbclient to try to login to the shares on the Samba server. What share doesn't require a password?

The screenshot shows a terminal window titled "root@ip-10-10-236-249: ~/Desktop/Tools/Miscellaneous". It displays the usage information for the smbclient command and several failed login attempts:

```
\\" Not enough '\\' characters in service
Usage: smbclient [-?EgBVNkPeC] [-?|---help] [--usage] [-R|--name-resolve=NAME-RESOLVE-ORDER]
[-M|---message=HOST] [-I|---ip-address=IP] [-E|---stderr] [-L|---list=HOST]
[-m|---max-protocol=LEVEL] [-T|---tar=<|>IXFqgbNan] [-D|---directory=DIR] [-c|--command=STRING]
[-b|---send-buffer=BYTES] [-t|---timeout=SECONDS] [-p|---port=PORT] [-g|---grepable] [-B|---browse]
[-d|---debuglevel=DEBUGLEVEL] [-s|---configfile=CONFIGFILE] [-l|---log basename=LOGFILEBASE]
[-V|---version] [--option=name=value] [-O|---socket-options=SOCKETOPTIONS]
[-n|---netbiosname=NETBIOSNAME] [-W|---workgroup=WORKGROUP] [-l|---scope=SCOPE]
[-U|---user=USERNAME] [-N|---no-pass] [-k|---kerberos] [-A|---authentication-file=FILE]
[-S|---signing=on|off|required] [-P|---machine-pass] [-e|---encrypt] [-C|---use-ccache]
[--pw-nthash] service <password>
root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous# smbclient //10.10.81.153/tbfc-hr
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous# smbclient //10.10.81.153/tbfc-it
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous# smbclient //10.10.81.153/tbfc-santa
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> |
```

The share that doesn't require a password is tbfc-santa

Question 5: Log in to this share, what directory did ElfMcSkidy leave for Santa?

The screenshot shows a terminal window titled "root@ip-10-10-236-249: ~/Desktop/Tools/Miscellaneous". It shows a successful login to the tbfc-santa share and a directory listing:

```
root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous# Try "help" to get a list of possible commands.
smb: \> pwd
Current directory is \\10.10.81.153\tbfc-santa\
smb: \> ^c
root@ip-10-10-236-249:~/Desktop/Tools/Miscellaneous# smbclient //10.10.81.153/tbfc-santa
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> get tbfc-santa
NT_STATUS_OBJECT_NAME_NOT_FOUND opening remote file \tbfc-santa
smb: \> ls
.
D 0 Thu Nov 12 02:12:07
2020
D 0 Thu Nov 12 01:32:21
2020
jingle-tunes
D 0 Thu Nov 12 02:10:41
2020
note_from_mcskidy.txt
N 143 Thu Nov 12 02:12:07
2020
10252564 blocks of size 1024. 5369396 blocks available
smb: \> |
```

After logging in to the share, we used ls command to get a list of directories that exist in this share. The directory that ElfMcSkidy leave for Santa is jingle-tunes

Thought Process/Methodology:

Enum4linux is a very important command that can help you a lot with your work. To know further about enum4linux, we used the help command to get a list of commands that can be used for our mission. To get a list of users on the server, we use the -u command from

enum4linux. After getting the list of users that are in the server, we also find the shares from the server by using the -s command. Using smbclient, we tried to log in to all the shares in the server to find one that doesn't require a password which is tbfc-santa. After we successfully log in to the share, we get a list of directories that was given to santa by using the ls command.