

# PSP0201

## Week 3

## Write-up

Group Name: **SuiBian**

Members:

ID	Name	Role
1211101851	Ang Zhe Jie	Leader
1211103039	Ooi Yi Siang	Member
1211103790	Kok Yew Yan	Member
121110	Wong Chun Rong	Member

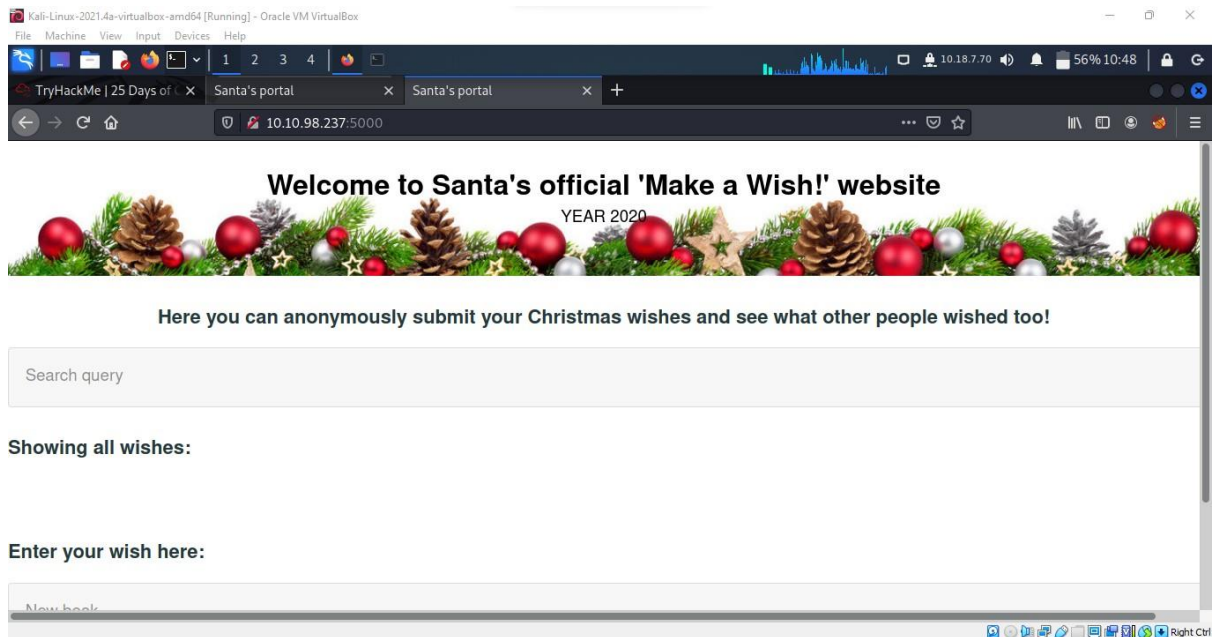
## Day 6: [Exploitation] Be careful with what you wish on a Christmas Night

**Tools used:** Kali Linux, Firefox, OWASP ZAP

### **Walkthrough:**

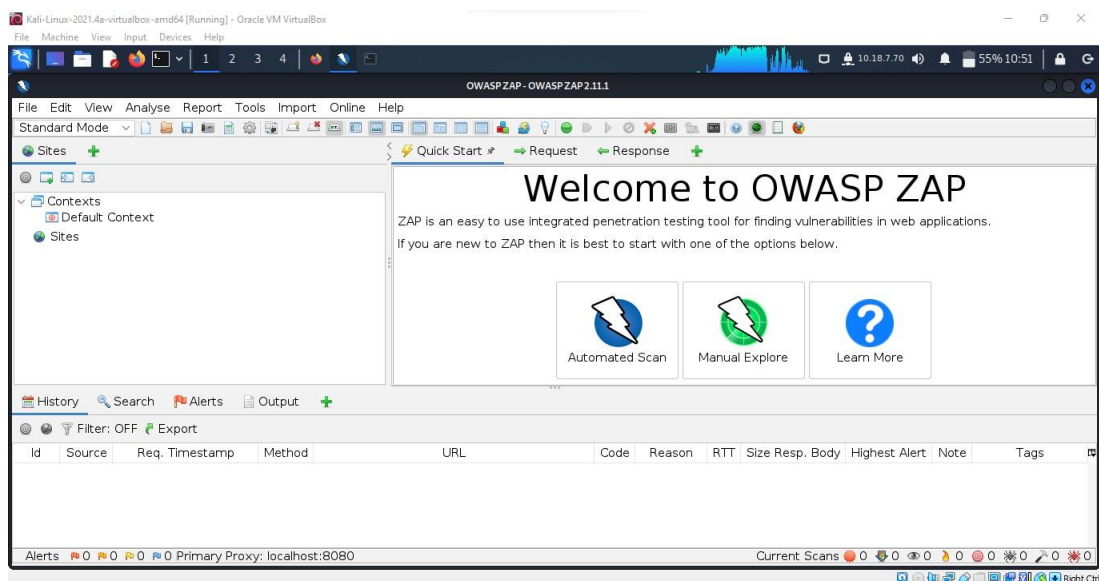
Question 1:

We paste the ip address and the page below is shown.



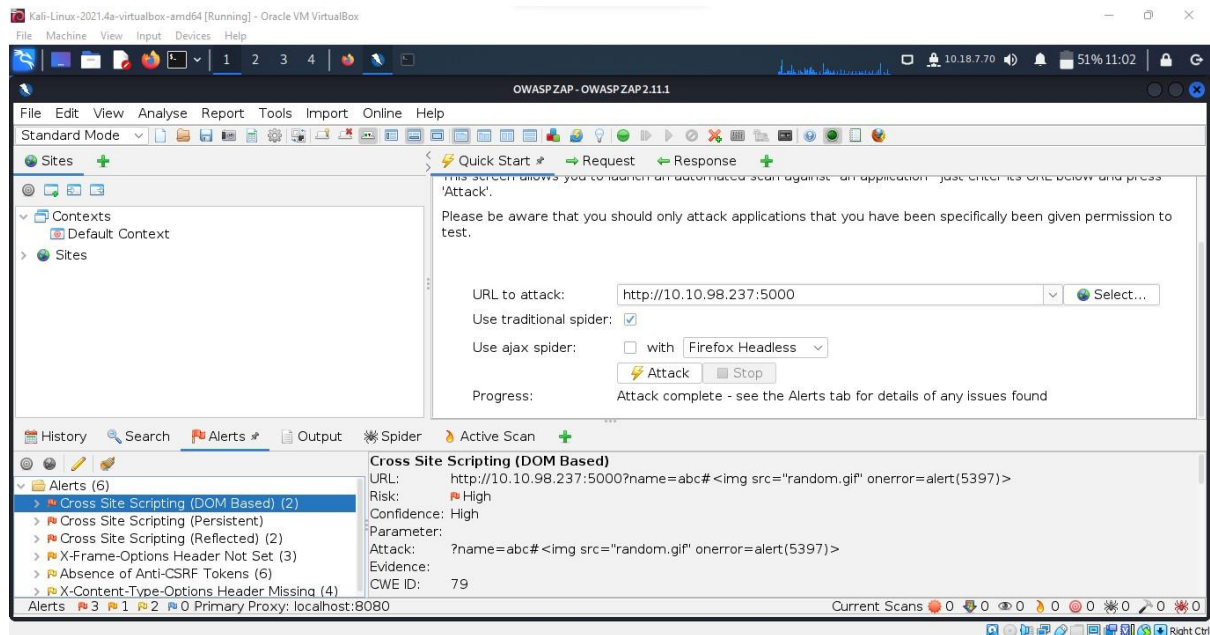
Question 2:

We open OWASP ZAP in Kali Linux machine to detect vulnerabilities, we use automated scan.



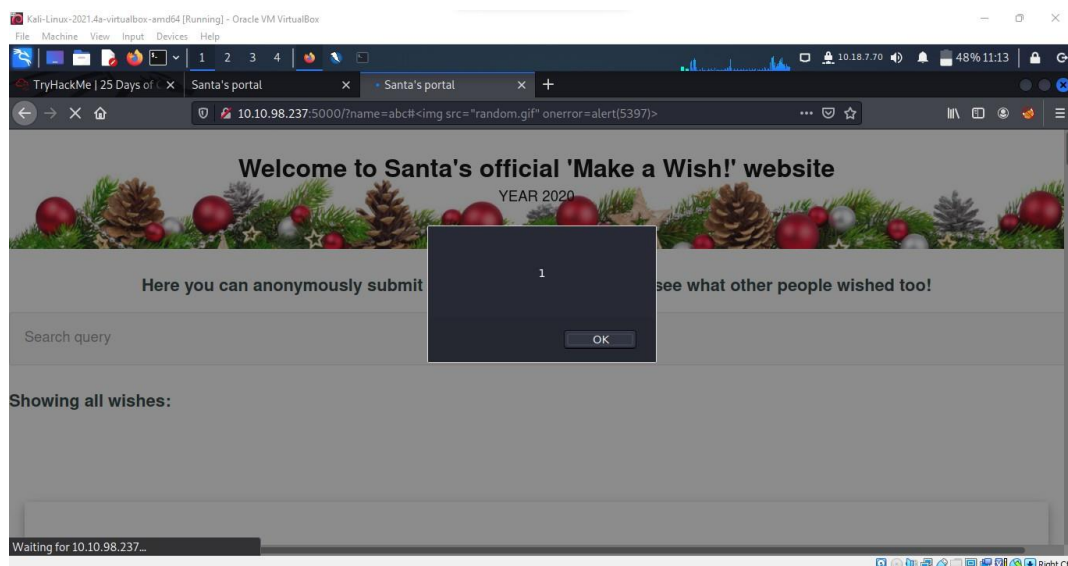
### Question 3:

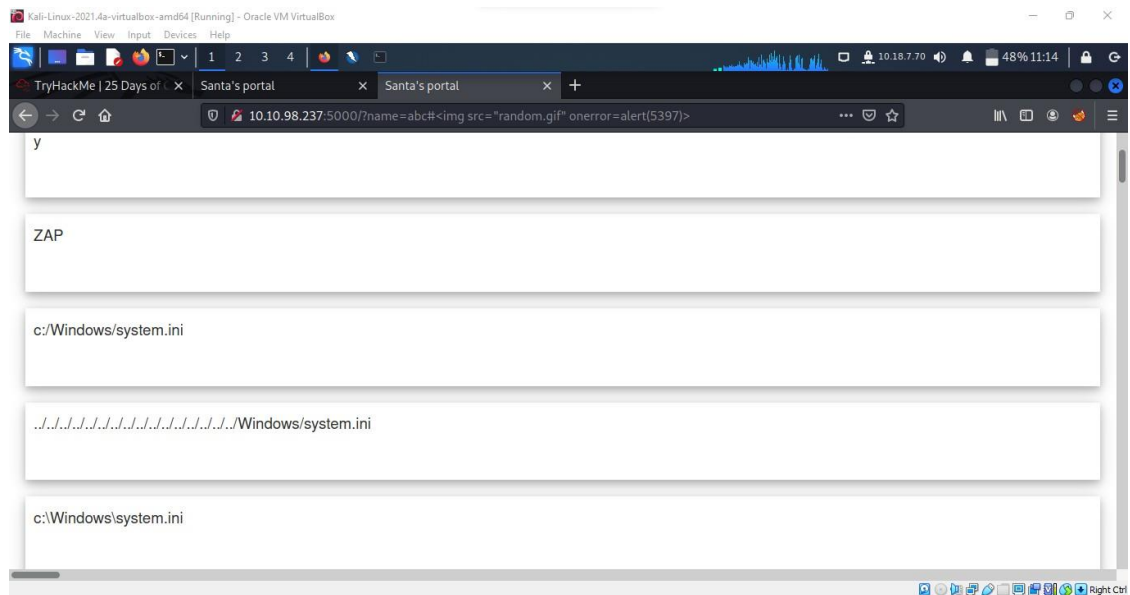
We start the attack after pasting the given URL of our webpage. After scanning, we notice a XSS (DOM Based) vulnerability with a malicious URL and copy the URL.



### Question 4:

Then we paste the URL in the Firefox browser. Multiple pop-ups will appear. The list of the wishes will be shown some abnormal entries. It means we have successfully done the attack.





### **Thought Process/Methodology:**

We go through the given IP address in the browser. We scan for vulnerabilities of the website using OWASP ZAP. We use automated scan. Later, we pasted the website's IP address in the column given and clicked on the attack button. Thus, a XSS (DOM Based) vulnerability can be seen after some times. A URL was there, and we copied the URL as we planned to perform a stored XSS attack. We pasted the URL in the browser to attack the website. During the running of URL, several pop-ups will appear. The list of the wishes will show some abnormal entries.

## Day 7 - [Networking] The Grinch Really Did Steal Christmas

**Tools used:** Kali linux, Wireshark, Terminal, Mousepad


### Walkthrough and Question:

Question 1:

Download the pcap files

Task 9 [Day 7] Networking The Grinch Really Did Steal Christmas

Watch DarkStar's Video On Solving This Task [Download Task Files](#)



(Javatpoint, 2018)

pcap1.pcap									
icmp									
No.	Time	Source	Destination	Protocol	Length	Info			
17	10.430447	10.11.3.2	10.19.15.52	ICMP	74	Echo (ping) request	id=0x0001, seq=1/256, ttl=127 (reply in 18)		
18	10.430472	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply	id=0x0001, seq=1/256, ttl=64 (request in 17)		
19	11.428953	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request	id=0x0001, seq=2/512, ttl=127 (reply in 20)		
20	11.428977	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply	id=0x0001, seq=2/512, ttl=64 (request in 19)		
21	12.432844	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request	id=0x0001, seq=3/768, ttl=127 (reply in 22)		
22	12.432879	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply	id=0x0001, seq=3/768, ttl=64 (request in 21)		
23	13.433469	10.11.3.2	10.10.15.52	ICMP	74	Echo (ping) request	id=0x0001, seq=4/1024, ttl=127 (reply in 24)		
24	13.433495	10.10.15.52	10.11.3.2	ICMP	74	Echo (ping) reply	id=0x0001, seq=4/1024, ttl=64 (request in 23)		

Question 2:

We are able to find the source of the request after opening it.

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
```

pcap1.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request.method == GET

No.	Time	Source	Destination	Protocol	Length	Info
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
480	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:

We find the source of ip address entering the password.

pcap2.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ftp

No.	Time	Source	Destination	Protocol	Length	Info
6	2.549894	10.10.73.252	10.10.122.128	FTP	72	Request: QUIT
7	2.549999	10.10.122.128	10.10.73.252	FTP	80	Response: 221 Goodbye.
16	4.105504	10.10.122.128	10.10.73.252	FTP	104	Response: 220 Welcome to the TBFC FTP Server!.
20	7.866325	10.10.73.252	10.10.122.128	FTP	83	Request: USER elfmcskidy
22	7.866430	10.10.122.128	10.10.73.252	FTP	100	Response: 331 Please specify the password.
28	14.282063	10.10.73.252	10.10.122.128	FTP	98	Request: PASS plaintext_password_fiasco
31	16.735293	10.10.122.128	10.10.73.252	FTP	88	Response: 530 Login incorrect.
33	16.735723	10.10.73.252	10.10.122.128	FTP	72	Request: SYST
35	16.735761	10.10.122.128	10.10.73.252	FTP	104	Response: 530 Please login with USER and PASS.
40	19.727087	10.10.73.252	10.10.122.128	FTP	72	Request: QUIT
41	19.727175	10.10.122.128	10.10.73.252	FTP	80	Response: 221 Goodbye.
52	22.445915	10.10.122.128	10.10.73.252	FTP	104	Response: 220 Welcome to the TBFC FTP Server!.
55	24.441994	10.10.73.252	10.10.122.128	FTP	82	Request: USER anonymous

Question 4:

We find the protocol that has encrypted packets.



Wireshark packet capture showing a Telnet session. The interface includes a menu bar, toolbar, packet list, packet details, and packet bytes panes. The packet list shows 14 packets, with the first two being SSH and the rest being Telnet. The packet details pane shows the structure of the selected packet (Telnet).

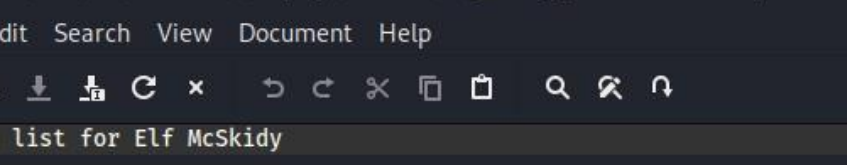
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.122.128	10.11.3.2	SSH	102	Server: Encrypted packet (len=48)
2	0.000004	10.10.122.128	10.11.3.2	SSH	150	Server: Encrypted packet (len=90)
3	0.000016	10.11.3.2	10.10.122.128	TCP	54	57748 → 22 [ACK] Seq=1 Ack=49 Win=1024 Len=0
4	0.101317	10.11.3.2	10.10.122.128	TCP	54	57748 → 22 [ACK] Seq=1 Ack=145 Win=1024 Len=0
5	1.127866	10.10.122.128	91.189.92.40	TCP	74	33402 → 443 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=3118188800 TSecr=0 WS=128
6	2.543994	10.10.73.252	10.10.122.128	FTP	72	Request: QUIT
7	2.543999	10.10.122.128	10.10.73.252	FTP	88	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	2.555529	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
12	3.175873	10.10.122.128	91.189.92.40	TCP	74	33402 → 443 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=3118190848 TSecr=0 WS=128
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=4110300014 TSecr=0 WS=128

Question 5:  
Examine the ARP communications.

The image shows a Wireshark packet capture window titled 'pcap2.pcap'. The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help), a toolbar with icons for file operations and packet analysis, and a packet list pane on the left showing the first 138 packets. The main packet details pane is currently displaying the selected packet (No. 46) as an ARP request. The packet list shows a series of ARP requests from source 02:c0:56:51:8a:51 to various destinations, including 10.10.122.128 and 10.10.10.17. The packet details pane shows the structure of an ARP request, including the Ethernet II header and the ARP payload.

No.	Time	Source	Destination	Protocol	Length	Info
46	19.785010	02:c0:56:51:8a:51	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:c0:56:51:8a:51	ARP	42	10.10.122.128 is at 02:c0:56:51:8a:51
77	26.727854	02:c0:56:51:8a:51	02:c0:56:51:8a:51	ARP	42	Who has 10.10.0.1? 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 10.10.0.1? 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:  
Open “pcap3.pcap” file.

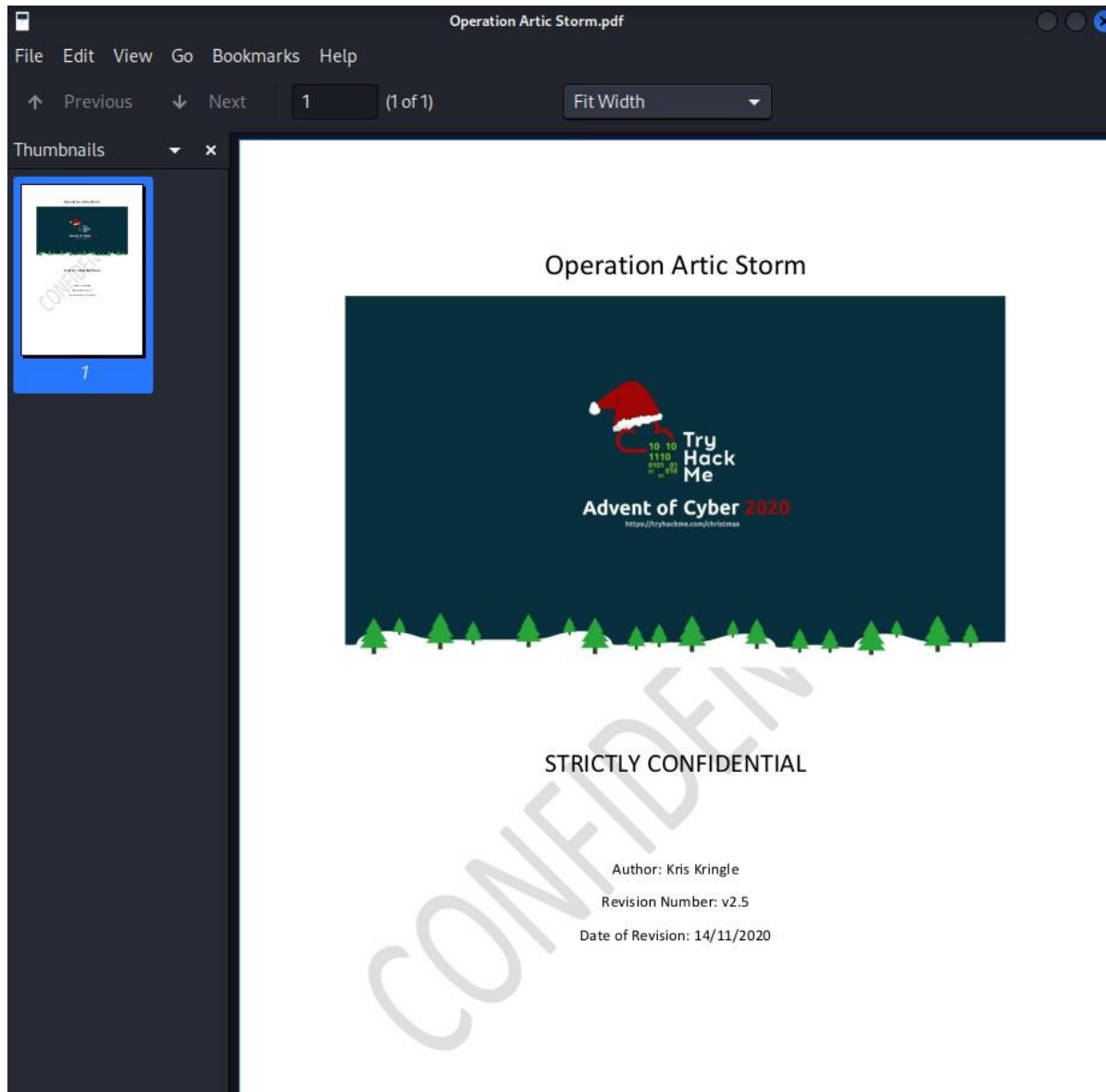


The screenshot shows a text editor window titled `/home/kali/Downloads/aoc-pcaps/christmas/elf_mcskidy_wishlist.txt - Mousepad`. The window has a menu bar with `File`, `Edit`, `Search`, `View`, `Document`, and `Help`. Below the menu is a toolbar with icons for file operations (new, open, save, print, etc.) and editing (undo, redo, cut, copy, paste, etc.). The text content of the file is as follows:

```
1Wish list for Elf McSkidy
2_____
3Budget: £100
4
5x3 Hak 5 Pineapples
6x1 Rubber ducky (to replace Elf McEager)
7
```

Question 7:

Find the text file from the zip downloaded and unzip the pdf file in the zip and look for the author.





**Thought Process/Methodology:**

We open Wireshark. Firstly, open pcap1.pcap and filter ICMP to look for the IP address that initiates an ICMP/ping. Filter with http.request.method == GET to find the article visited by the given IP address. We also open pcap2.pcap and apply ftp filter to find the password used to login. We also looking for the protocol that has encrypted packets. We open pcap3.pcap and download the christmas.zip file to look for what is used to replace Elf Mc Eager and who is the author of Operation Artic Storm after we examine the ARP.

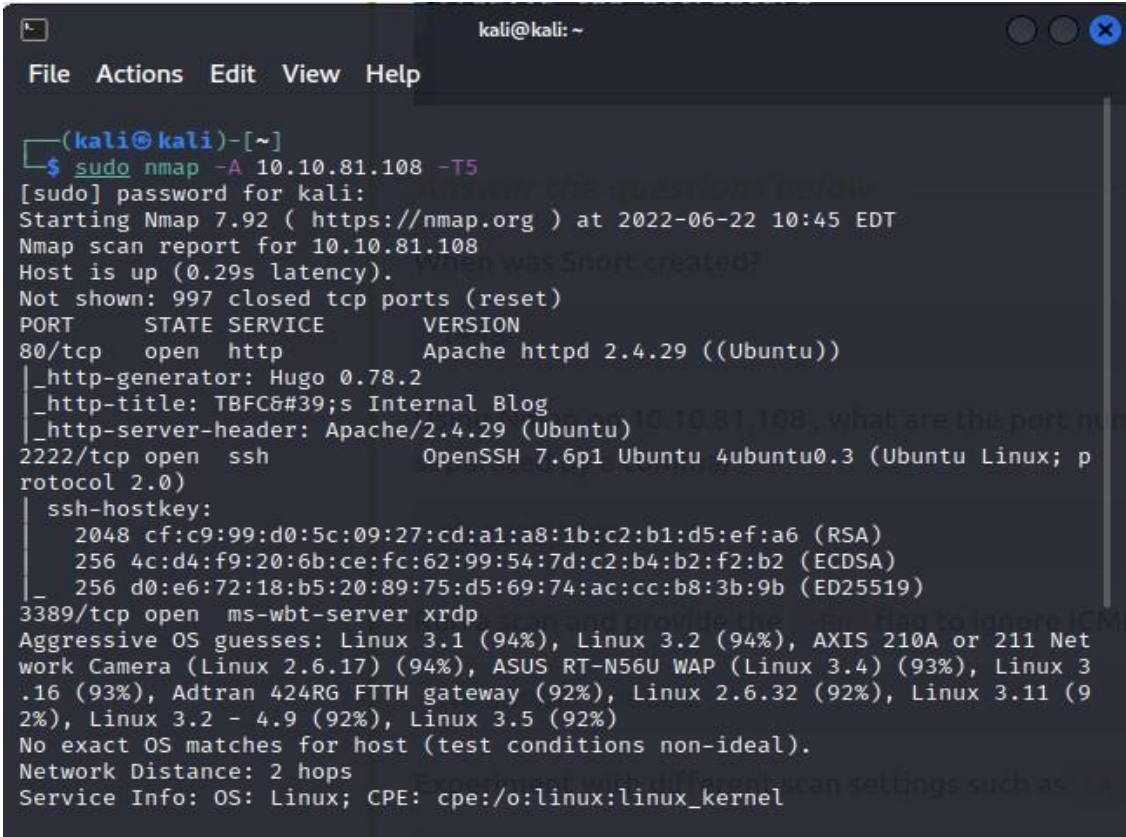
## Day 8 - [Networking] What's Under the Christmas Tree?

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

### **Walkthrough:**

Question 1:

Open the terminal and type Nmap with IP address command. Then the information as below will be seen.



```
(kali@kali)-[~]
$ sudo nmap -A 10.10.81.108 -T5
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-22 10:45 EDT
Nmap scan report for 10.10.81.108
Host is up (0.29s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Apache httpd 2.4.29 ((Ubuntu))
|_ http-generator: Hugo 0.78.2
|_ http-title: TBFC&#39;s Internal Blog
|_ http-server-header: Apache/2.4.29 (Ubuntu)
2222/tcp  open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; p
rotocol 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:ce:fc:62:99:54:7d:c2:b4:b2:f2:b2 (ECDSA)
|_  256 d0:e6:72:18:b5:20:89:75:d5:69:74:ac:cc:b8:3b:9b (ED25519)
3389/tcp  open  ms-wbt-server xrdp
Aggressive OS guesses: Linux 3.1 (94%), Linux 3.2 (94%), AXIS 210A or 211 Net
work Camera (Linux 2.6.17) (94%), ASUS RT-N56U WAP (Linux 3.4) (93%), Linux 3
.16 (93%), Adtran 424RG FTTH gateway (92%), Linux 2.6.32 (92%), Linux 3.11 (9
2%), Linux 3.2 - 4.9 (92%), Linux 3.5 (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
```

Question 2:

Type Nmap script command.

```
(kali@kali)-[~]  
$ nmap --script http-title 10.10.81.108 -T5 1 x  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-22 11:02 EDT  
Warning: 10.10.81.108 giving up on port because retransmission cap hit (2).  
Nmap scan report for 10.10.81.108: Was Short created?  
Host is up (0.29s latency).  
Not shown: 687 closed tcp ports (conn-refused), 311 filtered tcp ports (no-re  
sponse)  
PORT      STATE SERVICE  
80/tcp    open  http  
|_http-title: TBFC&#39;s Internal Blog  
3389/tcp  open  ms-wbt-server  
Nmap done: 1 IP address (1 host up) scanned in 38.13 seconds
```

### Thought Process/Methodology:

Firstly, we typed the Nmap and open all command together with our IP address in terminal. We able to see some information and figure out the port numbers, name of the Linux distribution which is running and the certain version of Apache. We use Nmap script command to check what is the website for after that.

## Day 9 - [Networking] Anyone can be Santa!

**Tools used:** Kali Linux, FTP, Terminal

Question 1:

Enter the File Transfer Protocol (FTP) server of the given IP Address as anonymous and list the directories.

```
ftp> ls
229 Entering Extended Passive Mode (|||50284|)
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:

**Only** the directory - 'public' has data in it.

Question 3:

backup.sh is found in the directory.

```
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||46783|)
150 Here comes the directory listing.
-rwxr-xr-x  1 111    113        341 Nov 16  2020 backup.sh
```

#### Question 4:

Find the shoppinglist.txt in the public directory and The Polar Express is in it. Find theflag.

```
GNU nano 6.2 shoppinglist.txt *
The Polar Express Movie
9.7. Conclusion, where to go from here and additional
```

```
root@tbfc-ftp-01:~# cat flag.txt
cat flag.txt
THM{even you can be santa}
```

#### Thought Process/Methodology:

Fill in the File Transfer Protocol (FTP) server of the given IP address with the terminal and login as an anonymous, we list out all the directories, check for the directory that has data accessible by the “anonymous” user. We can see the scripting language commands file that will be run to back up the server then download the file and replace the command in the script to our own reverse shell script. We upload the modified scripting language commands file after setting up a listener to catch the connection. We now take over the server with the help of reverse shell and now able to find the flag in the server’s directory.

## Day 10 -[Networking] Don't be sElfish!

**Tools used:** Kali Linux, enum4linux, smbclient, Terminal

### Walkthrough and Question:

Question 1:

Examine the help options for enum4linux.

```
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, -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. 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,kr
          btgt,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.)
  -A      Aggressive. Do write checks on shares etc
```

Question 2:

Use command `./enum4linux.pl -U MACHINE_IP` to check for the number of users on the server.

```
===== ( Users on 10.10.58.231 ) =====  
=====  
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskidy Name: Desc:  
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager Name: elfmceager D  
esc:  
index: 0x3 RID: 0x3e9 acb: 0x00000010 Account: elfmcelferson Name: Desc:  
  
user:[elfmcskidy] rid:[0x3e8]  
user:[elfmceager] rid:[0x3ea]  
user:[elfmcelferson] rid:[0x3e9]  
enum4linux complete on Mon Jun 20 20:32:35 2022
```

Question 3:

Use command `./enum4linux.pl -S MACHINE_IP` to check for the number of shares on the server.

```
===== ( Share Enumeration on 10.10.58.231 ) =====  
=====  


| 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 |


```

Question 4:

Login each share found on the server and looking for the share can be login successfully without the password.



Question 5:

Two directories in the tbfc-santa share were found.

```
(1211102575@kali)-[/usr/share/enum4linux]
$ smbclient //10.10.58.231/tbfc-santa
Password for [WORKGROUP\1211102575]:
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 |


```

### Thought Process/Methodology:

After navigating to enum4linux, we use enum4linux to check for the users and shares on the server. We find the share that we can access without a password. Therefore, we list out the directory in the share, download and read through the note\_from\_mcskidy.txt. Then, we are able to find the directory Elf McSkidy leaves for Santa.