PSP0201

Week 5

Writeup

Group Name: DASH

Members

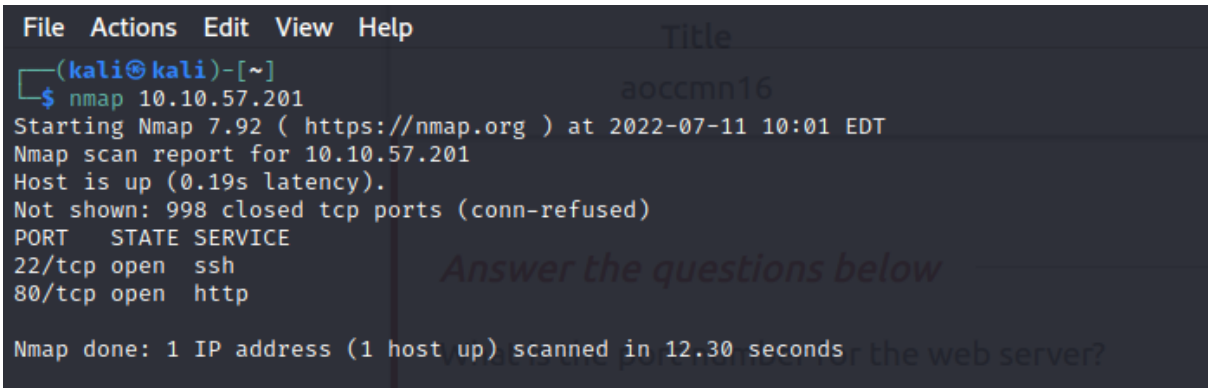
ID	Name	Role
1211101775	Lam Yuet Xin	Leader
1211101749	Teoh Xin Pei	Member
1211101398	Poh Ern Qi	Member
1211101800	Tan Jia Jin	Member

Day 16 (Scripting)- Help! Where is Santa?

Tools Used: Kali Linux

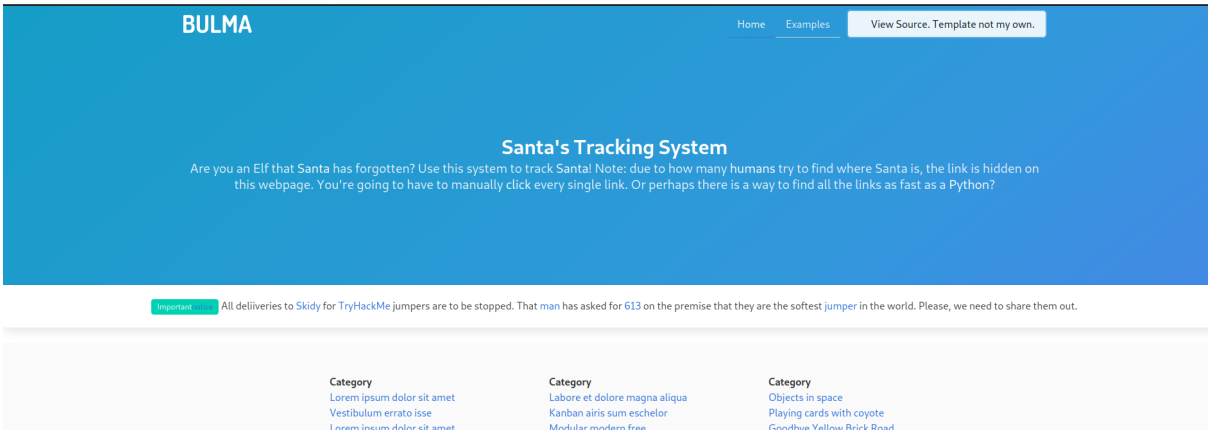
Question 1

Use nmap to scan our IP 10.10.57.201. We can see two ports, 22 and 80 are open. Since port 80 is used for hosting http web server, the port number for the web server is 80.



Question 2

To see the template, the url 10.10.57.201/static/index.html is entered. The name of the template is located at the top left of the website, BULMA.



Question 3

To find the directory of the API, we need to use python as what the website suggested. To extract all links in a webpage,we need to use the libraries, requests and beautifulsoup from day 15. Copy the code from day 15 and paste in a text editor. We replace the testurl.com with our url that we want to extract, http://10.10.57.201.

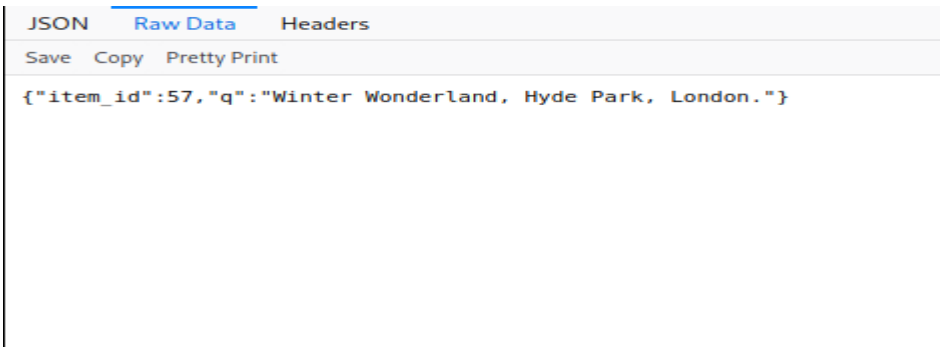
```
GNU nano 5.9
# Import the libraries we downloaded earlier
# if you try importing without installing them, this step will fail
from bs4 import BeautifulSoup
import requests

# replace testurl.com with the url you want to use.
# requests.get downloads the webpage and stores it as a variable
html = requests.get('http://10.10.57.201')

# this parses the webpage into something that beautifulsoup can read over
soup = BeautifulSoup(html.text, "lxml")
# lxml is just the parser for reading the html

# this is the line that grabs all the links # stores all the links in the links variable
links = soup.find_all('a')
for link in links:
    # prints each link
    print(link)
```

By setting soup.find_all('a'), it will print all the links that start with the 'a' tag. Now, we run the command python3 test.py, which is the text file we just created. We can see the link of the API, http://machine_ip/api/api_key is printed. The directory is /api/.

```
┌──(kali㉿kali)-[~]
└─$ python3 test.py
<a class="navbar-item" href="../">
<img alt="Logo" src="http://bulma.io/images/bulma-type-white.png" />
</a>
<a>Home</a>
<a href="">Examples</a>
<a class="button is-white is-outlined" href="https://github.com/BulmaTemplates/bulma-templates/blob/master/templates/hero.html">
<span class="icon">
<i class="fa fa-github"></i>
</span>
<span title="Hello from the other side">View Source. Template not my own.</span>
</a>
<a href="https://tryhackme.com">Santa</a>
<a href="https://tryhackme.com">Santa</a>
<a href="https://tryhackme.com">humans</a>
<a href="https://tryhackme.com">click</a>
<a href="https://tryhackme.com">Python</a>
<a href="https://tryhackme.com">notice</a>
<a href="https://tryhackme.com">Skidy</a>
<a href="https://tryhackme.com">TryHackMe</a>
<a href="https://tryhackme.com">man</a>
<a href="https://tryhackme.com">613</a>
<a href="https://tryhackme.com">jumper</a>
<a href="#">Lorem ipsum dolor sit amet</a>
<a href="#">Vestibulum errato isse</a>
<a href="#">Lorem ipsum dolor sit amet</a>
<a href="#">Aisia caisia</a>
<a href="#">Murphy's law</a>
<a href="#">Flimsy Lavenrock</a>
<a href="#">Maven Mousie Lavender</a>
<a href="#">Labore et dolore magna aliqua</a>
<a href="#">Kanban airis sum eschelor</a>
<a href="http://machine_ip/api/api_key">Modular modern free</a>
<a href="#">The king of clubs</a>
<a href="#">The Discovery Dissipation</a>
```

Question 4

By entering http://10.10.57.201/api/57, we can see the raw tab it shows {"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}.



Question 5

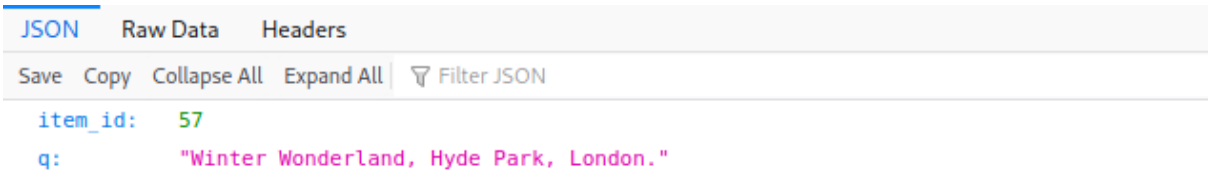
To see Santa's location, we need to find the api key. Insert the code using the for loop in a text editor, in this way, we can test by looping through every api_key starting from 1 to 100 in odd numbers.

We can see for item_id: 57, we found Santa's location, Winter Wonderland, Hyde Park, London.

```
┌──(kali㉿kali)-[~]
└─$ python3 api.py
{"item_id":1,"q":"Error. Key not valid!"}
{"item_id":3,"q":"Error. Key not valid!"}
{"item_id":5,"q":"Error. Key not valid!"}
{"item_id":7,"q":"Error. Key not valid!"}
{"item_id":9,"q":"Error. Key not valid!"}
{"item_id":11,"q":"Error. Key not valid!"}
{"item_id":13,"q":"Error. Key not valid!"}
{"item_id":15,"q":"Error. Key not valid!"}
{"item_id":17,"q":"Error. Key not valid!"}
{"item_id":19,"q":"Error. Key not valid!"}
{"item_id":21,"q":"Error. Key not valid!"}
{"item_id":23,"q":"Error. Key not valid!"}
{"item_id":25,"q":"Error. Key not valid!"}
{"item_id":27,"q":"Error. Key not valid!"}
{"item_id":29,"q":"Error. Key not valid!"}
{"item_id":31,"q":"Error. Key not valid!"}
{"item_id":33,"q":"Error. Key not valid!"}
{"item_id":35,"q":"Error. Key not valid!"}
{"item_id":37,"q":"Error. Key not valid!"}
{"item_id":39,"q":"Error. Key not valid!"}
{"item_id":41,"q":"Error. Key not valid!"}
{"item_id":43,"q":"Error. Key not valid!"}
{"item_id":45,"q":"Error. Key not valid!"}
{"item_id":47,"q":"Error. Key not valid!"}
{"item_id":49,"q":"Error. Key not valid!"}
{"item_id":51,"q":"Error. Key not valid!"}
{"item_id":53,"q":"Error. Key not valid!"}
{"item_id":55,"q":"Error. Key not valid!"}
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
{"item_id":59,"q":"Error. Key not valid!"}
{"item_id":61,"q":"Error. Key not valid!"}
{"item_id":63,"q":"Error. Key not valid!"}
```

Question 6

We just found our api key which is the same as our item_id, 57. By entering the url, http://10.10.57.201/api/57, we can see we are landed on the right page.



Thought process/ methodology

To find the port number for the web server, we think that we should use nmap to see the available ports. We can see two ports are open, 22 and 80, each with service ssh and http. Since we are looking for web servers (http), we deduced that the port number used is 80, which is the default port for http. To find the API directory, we entered the url 10.10.57.201/static/index.html to find hidden hints. Based on the website, we noted that we can find all hidden links using python. We recalled that we can extract the links using

two libraries, requests and beautifulsoup that we learnt on day 15. Hence, by entering the code and replacing the url in a text editor, it successfully printed the API directory. To find the API key, we get the idea from the video that we can use the for loop to iterate the odd numbers from 1 to 100, hence, by entering the code in another text editor, it successfully printed the api key, 57 as well as Santa's location, which is the result we wanted.

Day 17 (Reverse Engineering) ReverseELFneering

Tool Used: THM attack box

Question 1:

The thm articles provide a table which shows the initial data type, Suffix and Size in bytes.

Initial Data Type	Suffix	Size (bytes)
Byte	b	1
Word	w	2
Double Word	l	4
Quad	q	8
Single Precision	s	4
Double Precision	l	8

Question 2:

The command for analyse the program in radare22 is aa.

```
elfmceager@tbfc-day-17: ~
File  Edit  View  Search  Terminal  Help
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Jul 13 12:39:49 UTC 2022

  System load:  0.96               Processes:           103
  Usage of /:   40.8% of 11.75GB   Users logged in:     0
  Memory usage: 8%                 IP address for ens5: 10.10.69.72
  Swap usage:   0%


packages can be updated.
updates are security updates.


Last login: Wed Dec 16 18:25:51 2020 from 192.168.190.1
elfmceager@tbfc-day-17:~$ ls
challenge1  file1
elfmceager@tbfc-day-17:~$ cd file1
-bash: cd: file1: Not a directory
elfmceager@tbfc-day-17:~$ r2 =d ./file1
Warning: Cannot initialize dynamic strings
[0x00400a30]> aa
[x] Analyze all flags starting with sym. and entry0 (aa)
[0x00400a30]>
```

Question 3&4:

Read through the articles, we will find a command to set a breakpoint is db and the command to execute the program is dc.

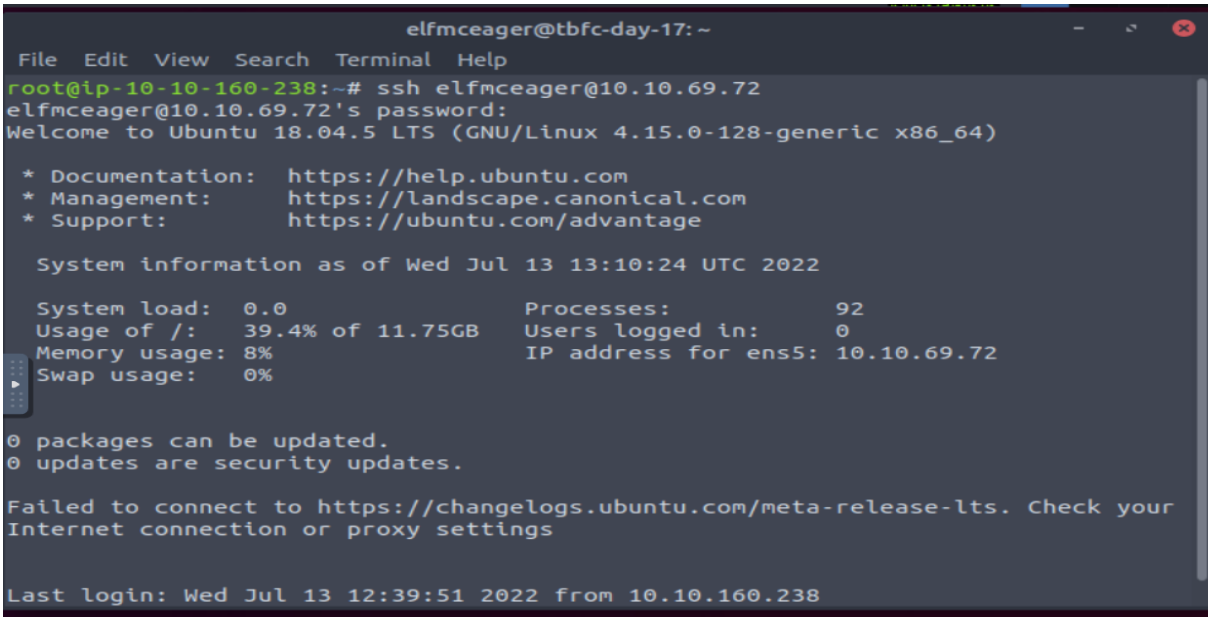
A **breakpoint** specifies where the program should stop executing. This is useful as it allows us to look at the state of the program at that particular point. So let's set a breakpoint using the command `db` in this case, it would be `db 0x00400b55` To ensure the breakpoint is set, we run the `pdf @main` command again and see a little **b** next to the instruction we want to stop at.

```
0x00400a30]> pdf @main
            ;-- main:
 (fcn) sym.main 68
   sym.main (int argc, char **argv, char **envp);
            ; var int local_ch @ rbp-0xc
            ; var int local_8h @ rbp-0x8
            ; var int local_4h @ rbp-0x4
            ; DATA XREF from entry0 (0x400a4d)
            0x00400b4d      55              pushq %rbp
            0x00400b4e      4889e5          movq %rsp, %rbp
            0x00400b51      4883ec10        subq $0x10, %rsp
            0x00400b55 b    c745f4040000.   movl $4, local_ch
```

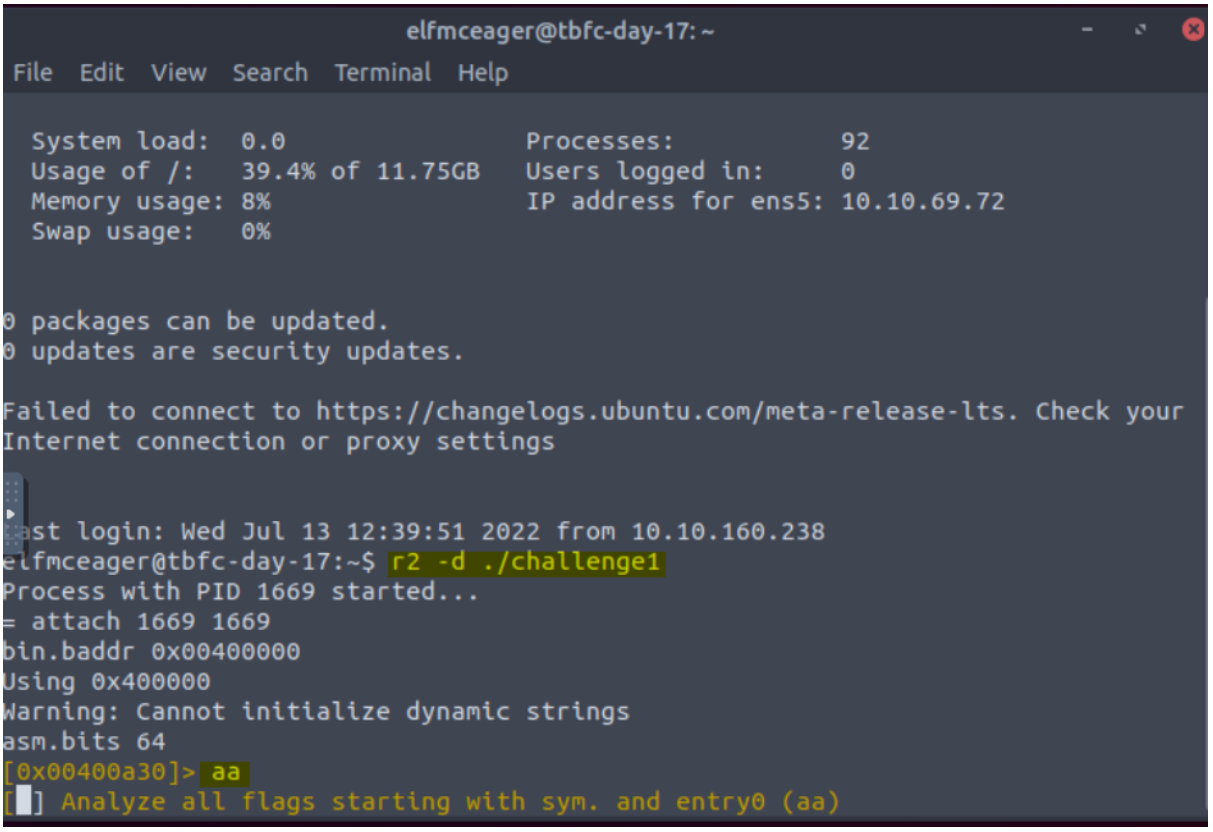
Now that we've set a breakpoint, let's run the program using `dc`

Question 5:

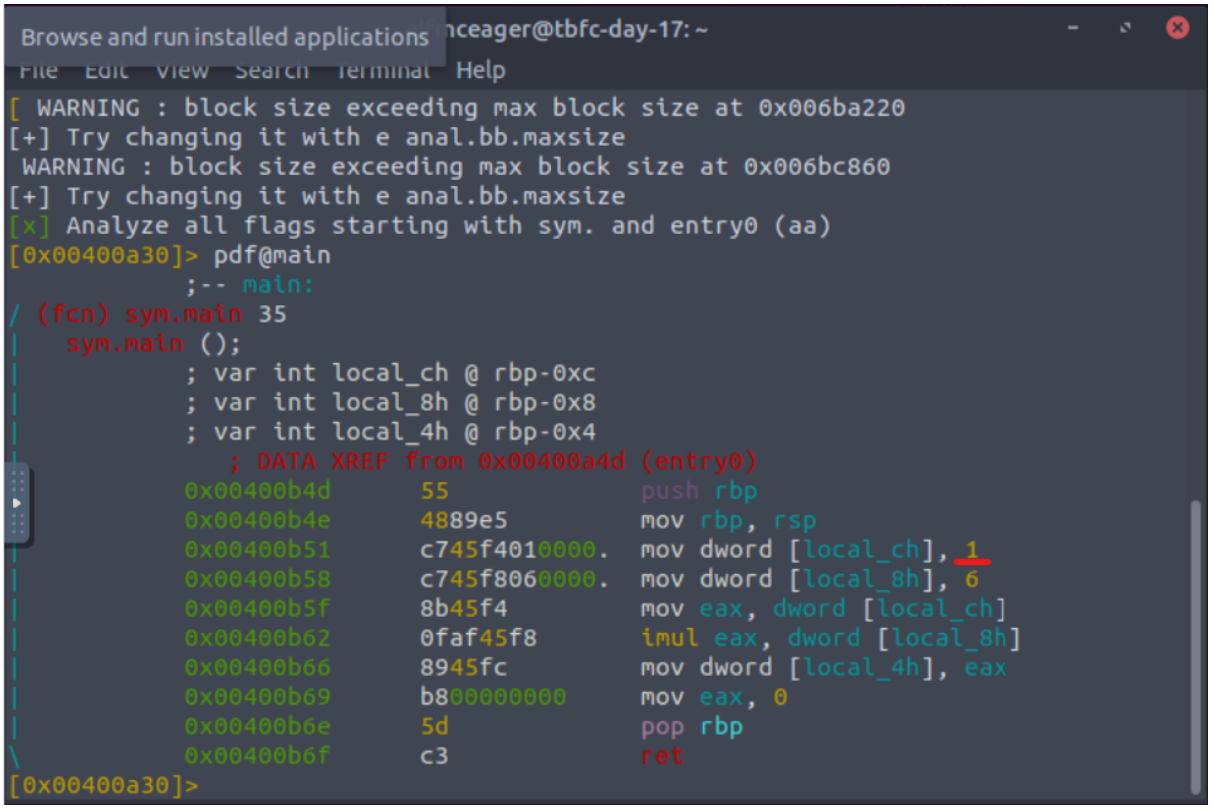
First, type the command ssh Machine_IP to login into the Machine_IP with username – elfmceager, password – adventofcyber



Then, run the radare2 command to analyse the challenge1.

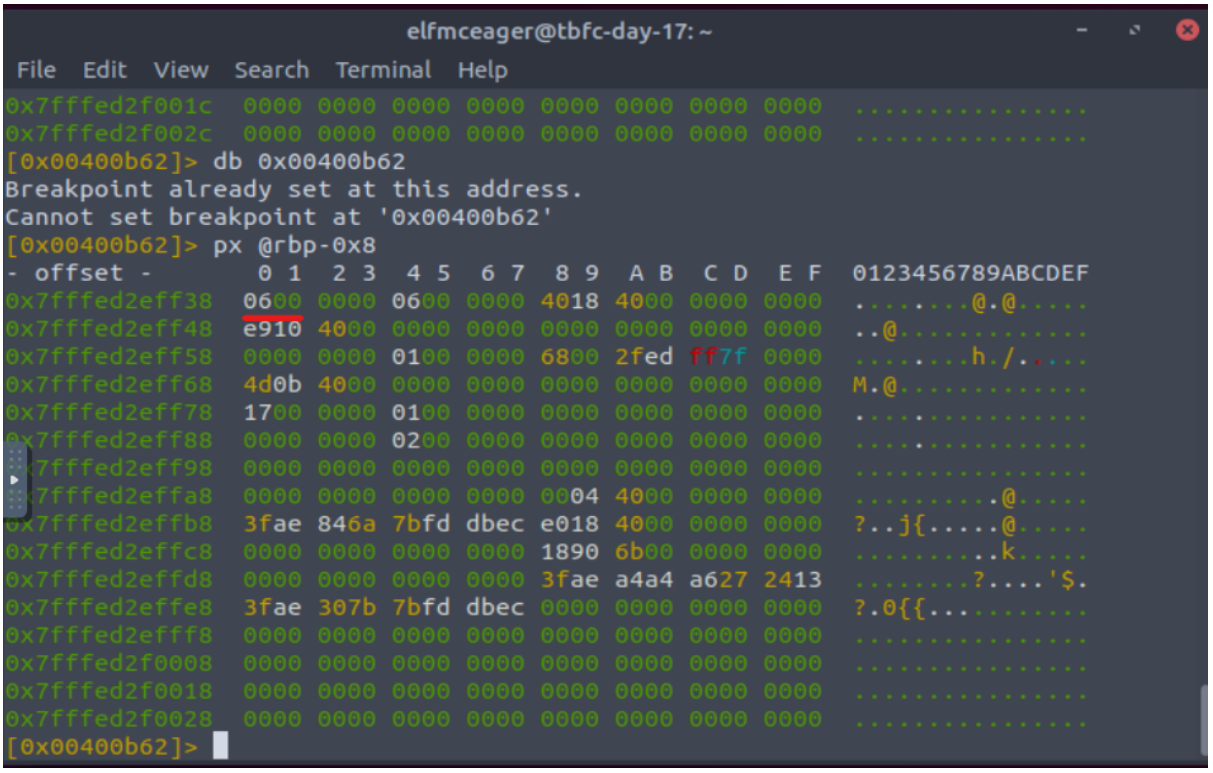


Last we enter the command pdf@main to view the information. We could see that the local_ch value is 1.



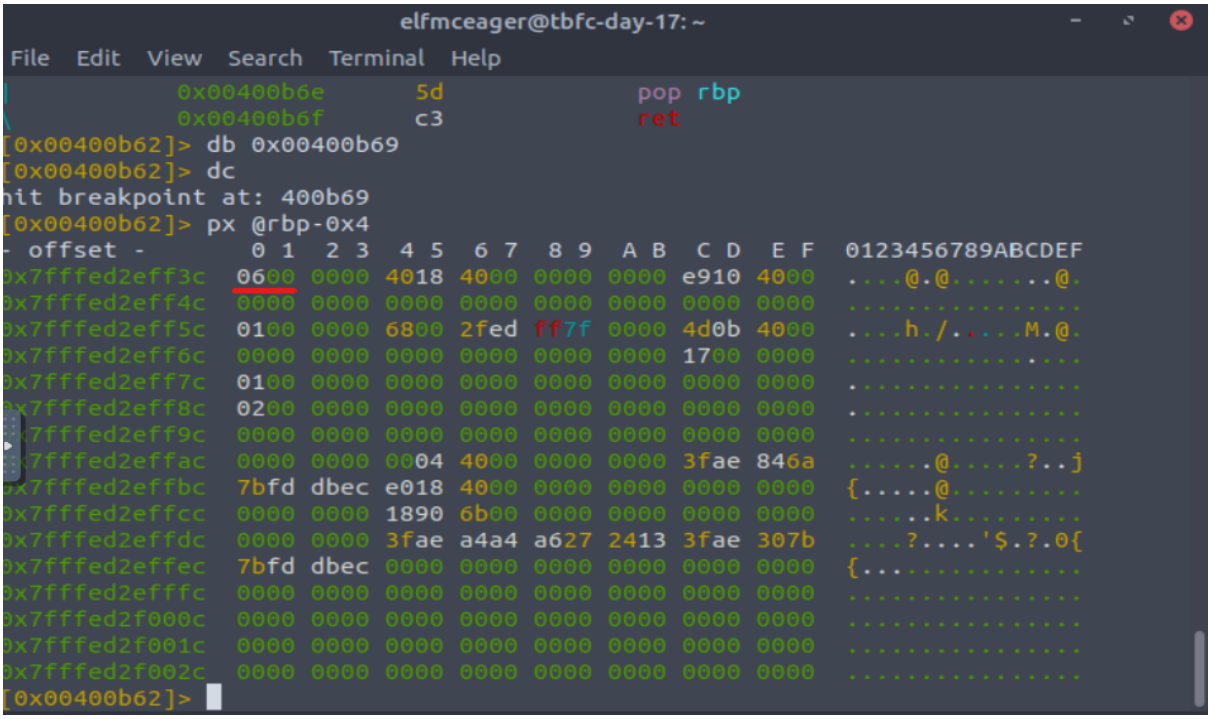
Question 6:

By viewing the information in pdf@main again, we find that the instruction for imul eax is 0x00400b62, so we set a breakpoint to checkout the value of local_8h and we get 6.



Question 7:

Looking through the pdf@main again, we find that the instruction for eax,0 is 0x00400b69, so we set a breakpoint to checkout the value of local_4h and we get 6.



Thought/ methodology:

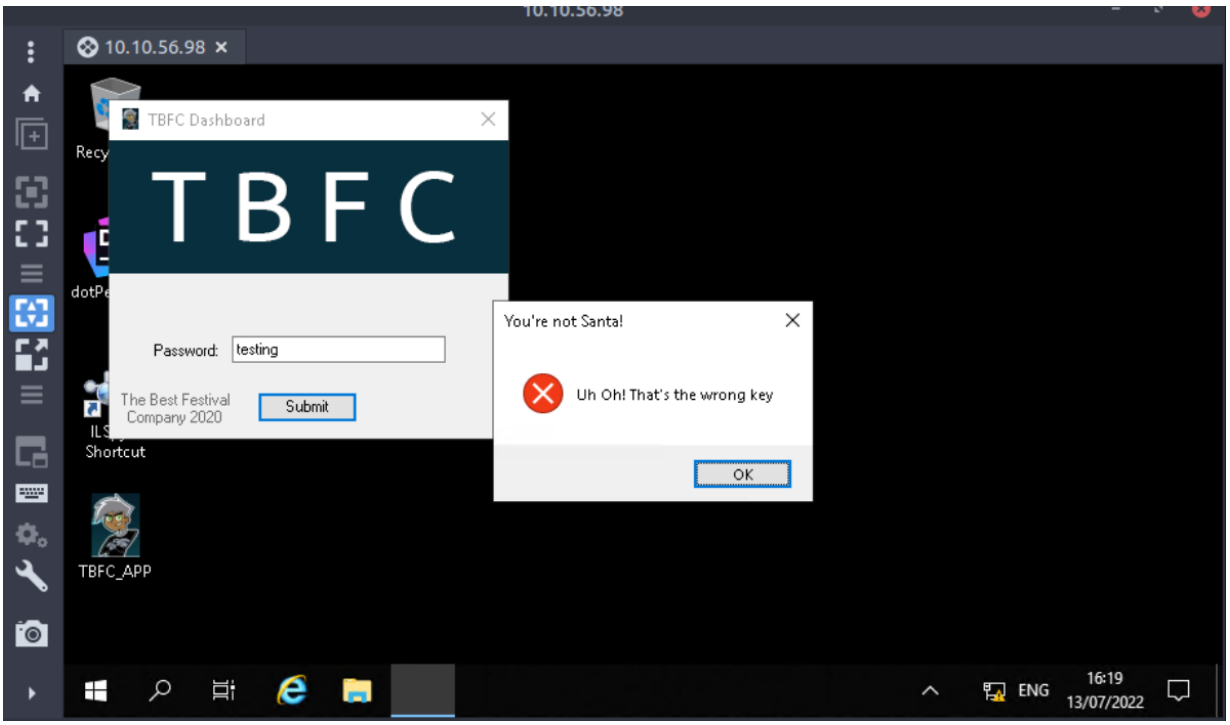
First, launch the thm attack box and open the terminal. Login to the Machine_IP by using the username and password provided. For day17, we are focusing on the Radare2 function. Analyse the challenge1 file we found by type in the command pdf@main or aa to view the information. We are looking for local_ch,eax and local_4h before eax and it can be found in the information. To verify, set a breakpoint using the command – db, dc to run.

Day 18 (Reverse Engineering) The Bits of Christmas

Tool used: Tryhackme attack box, Firefox, ILSpy

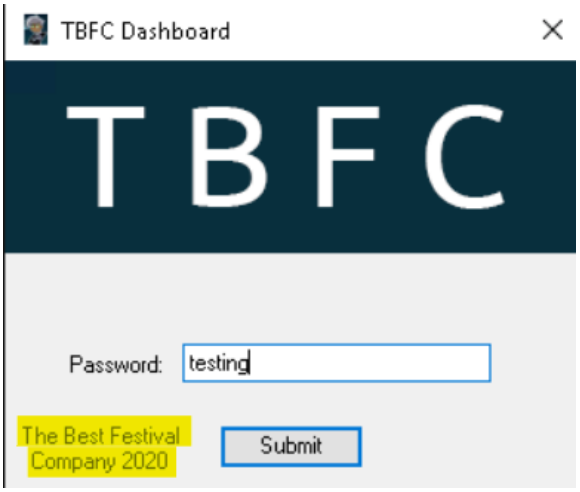
Question 1:

Open to the Remote desktop tool – Remima, then connect and login using the username and password. On the windows desktop, open the TBFC_APP and key in a random password to test it. If the password is wrong, "Uh Oh! That's the wrong key" will pop out.



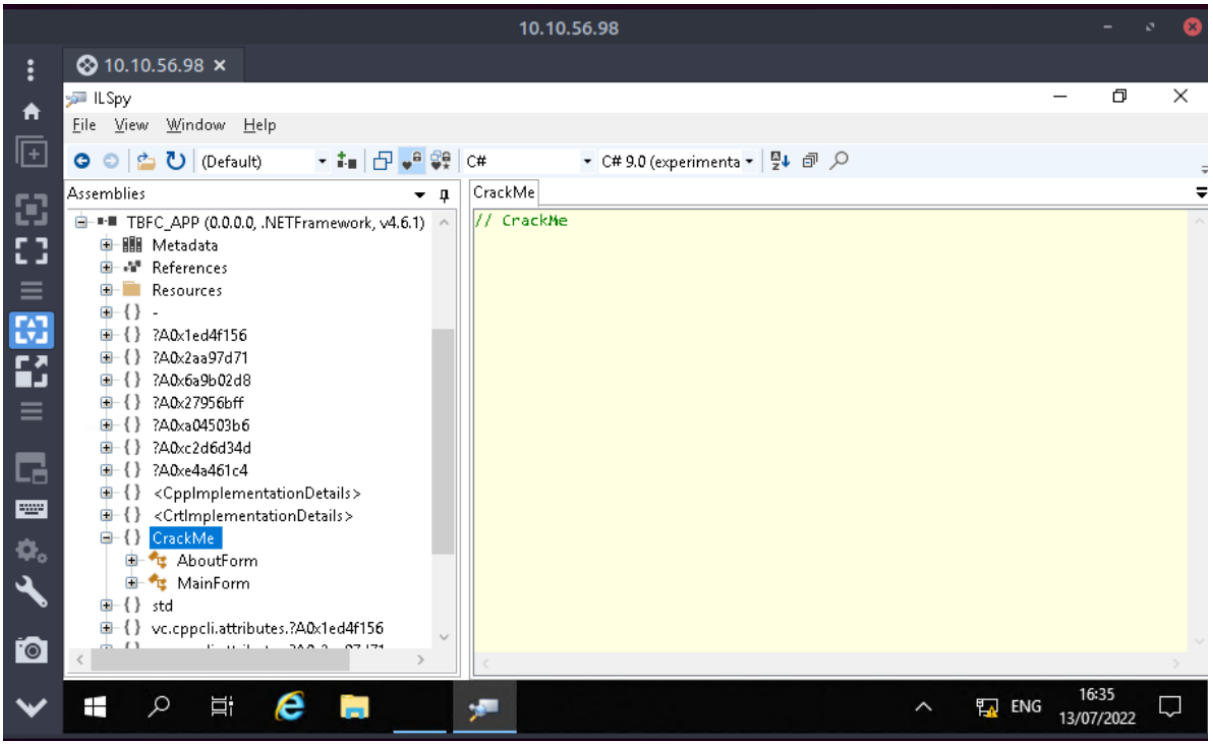
Question 2:

Run the TBFC_APP, the left bottom of the tab has the full name of TBFC which is The Best Festival Company.



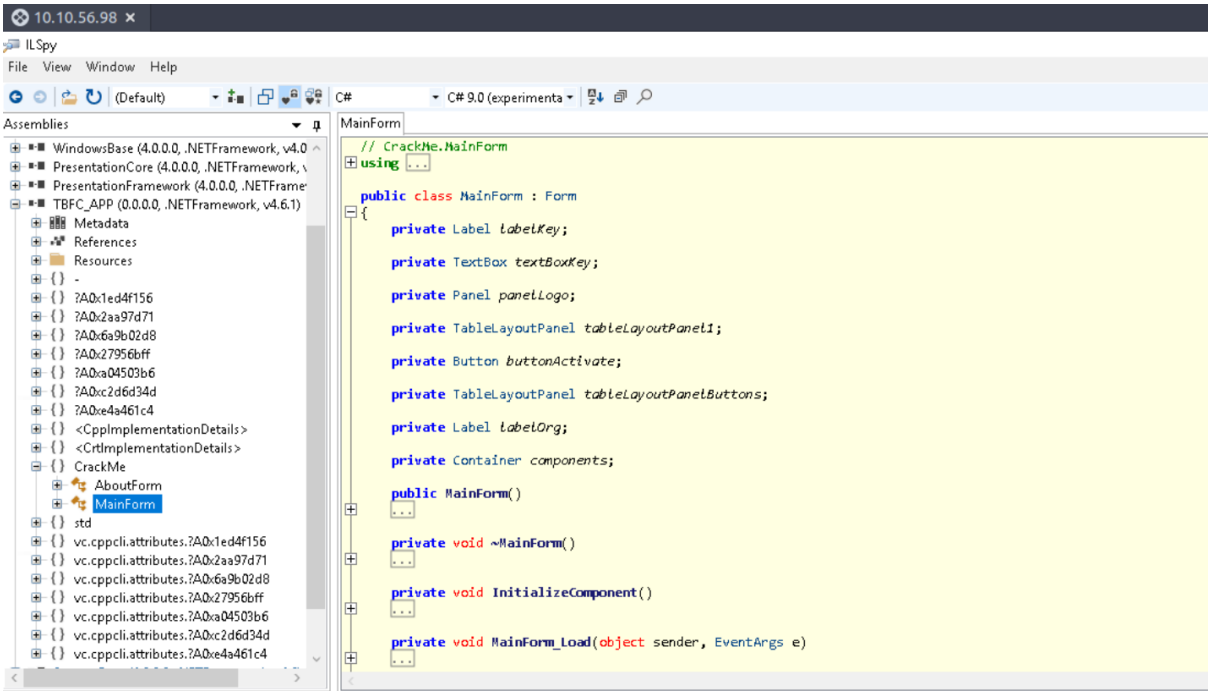
Question 3:

The CrackMe module catches my attention because of its title and it has AboutForm and MainForm. These modules explain the function of the Apps.



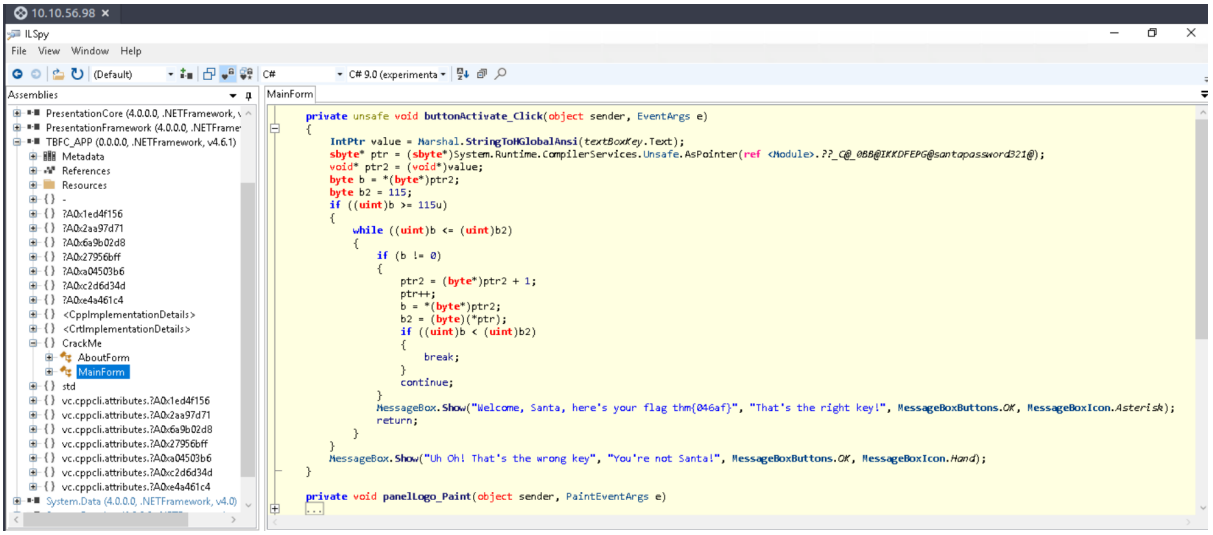
Question 4:

The Mainform contains the information we are looking for. This module contains the button function and display function.



Question 5

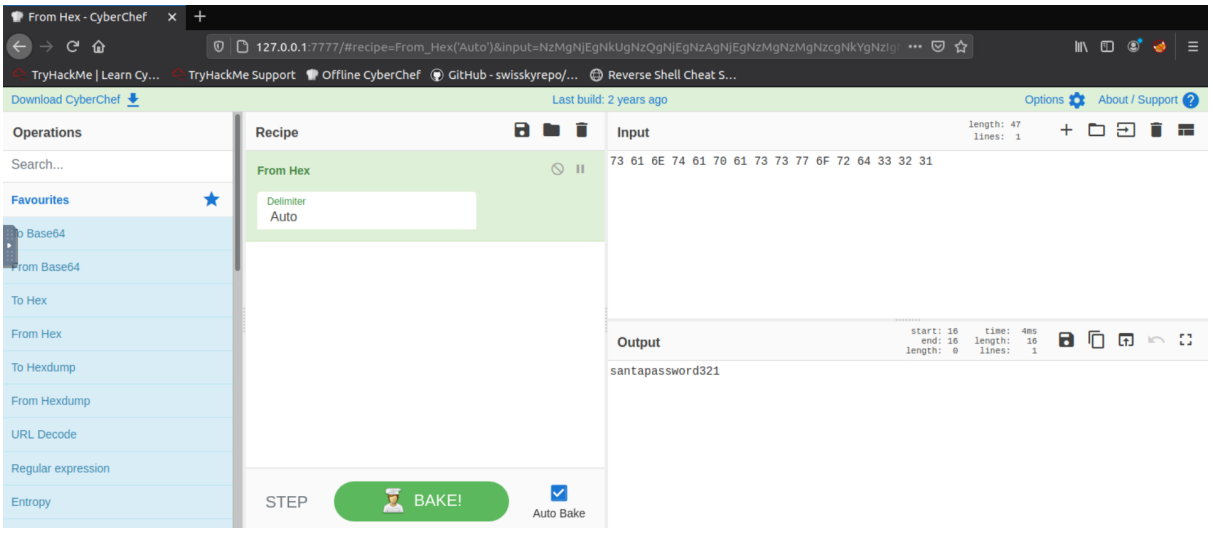
The buttonActivate_click contains the information we are looking for. This function is for the button for login. Looking through the module, we found a sensitive value and we could guess the password is santapassword321.



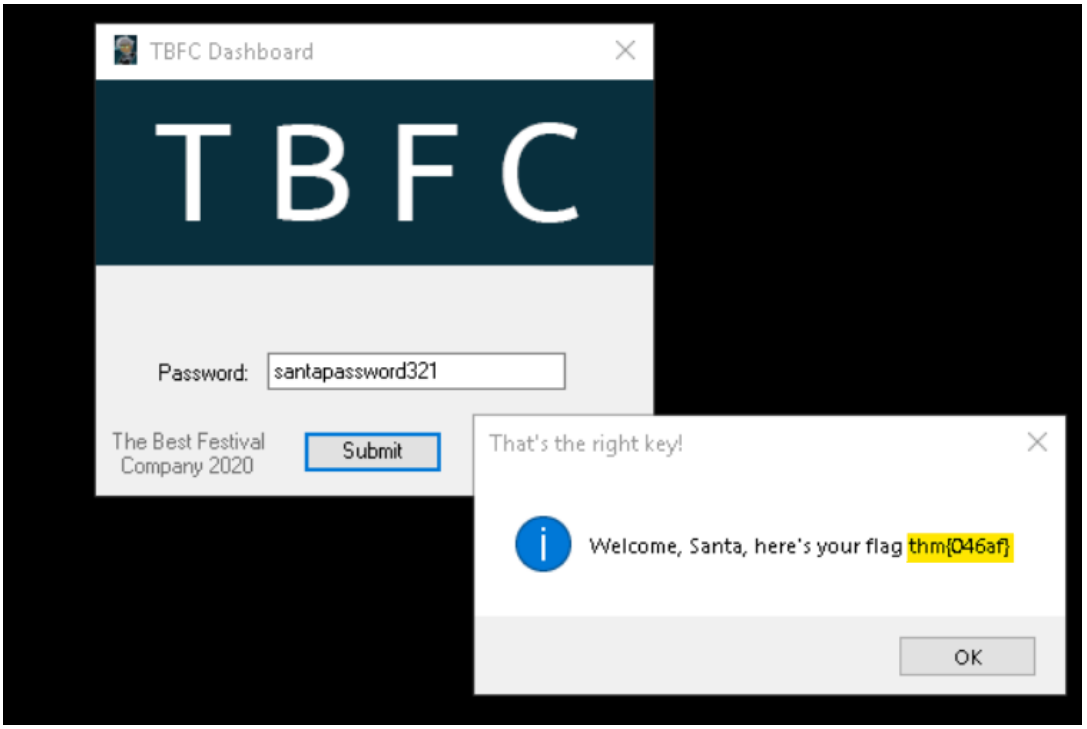
Question 7&8:

To confirm the santa password, click into the ptr line module. We will get direct to another module that contains a line of hexadecimal.

Use cyberchef to convert it to alphabetical form. The password is santapassword321.



To verify the password, go to the TBFC_APP and login. Then we get the flag – thm{046af}.



Thought/ methodology:

First, we launch the thm attack box and open an application — Remima to connect to remote machines. Then, fill in the username cmnatic and password Adventofcyber! given by thm. We would see a Windows desktop once we are connected to the Machine_IP. Next, open the TBFC_APP in ILSpy which we can find in the desktop folder. Once we decompile the TBFC_APP, look through the modules and we found an interesting title – Crackme. Going into Crackme, there is a MainForm module which would contain the information we are seeking for. There is a buttonActivate_Click function which has the password – santapassword321. To further verify the password, we click into the other module that contains a line of hexadecimal. Then, we convert it using cyberchef and we get the password as the same as the password shown in buttonActivate_Click function. Now, login to TBFC_APP using the password and we get a message "Welcome, Santa, here's your flag thm{046af}"

Day 19 - [Web Exploitation] The Naughty or Nice List

Tools used: THM attack box

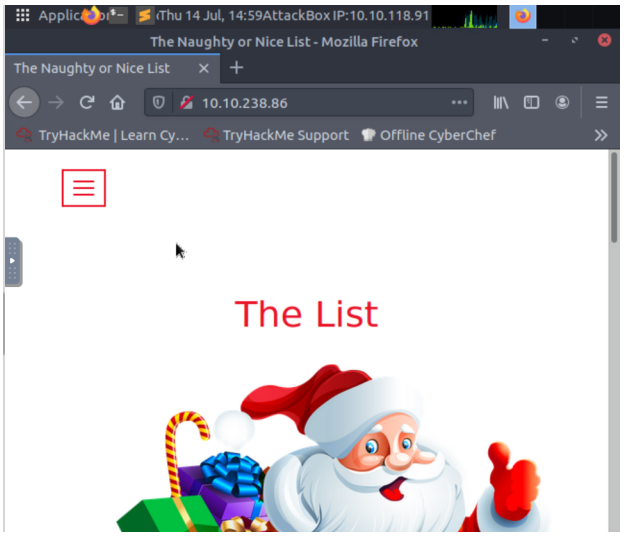
Solutions:

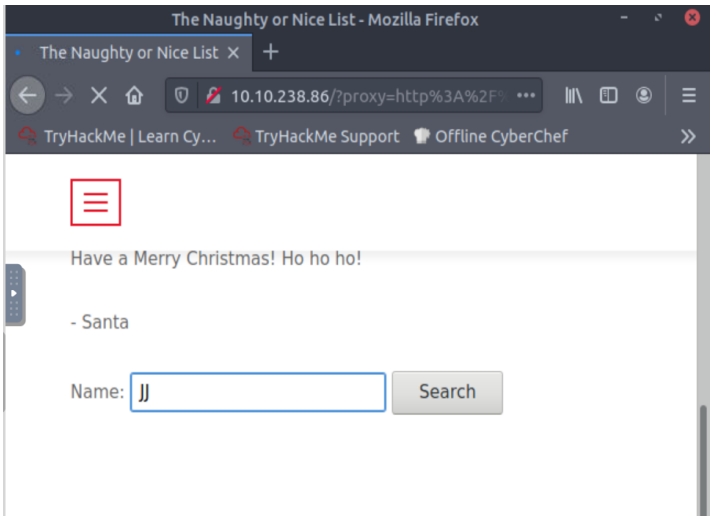
Question 1

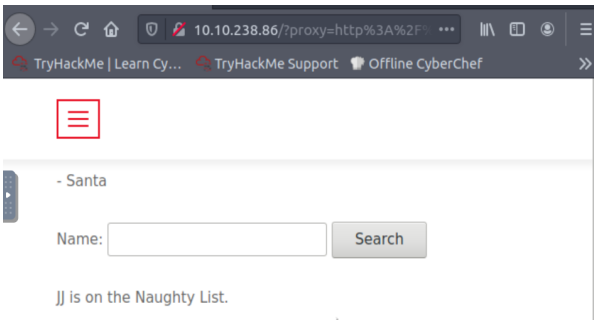
connect to the web app with "http://10.10.117.217", then enter the names to determine which is "naughty" or "nice".

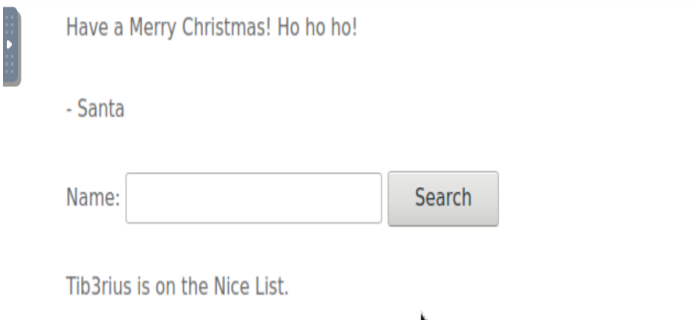
Walkthrough

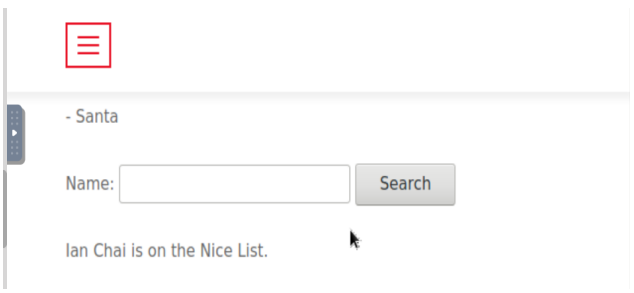
1. Once the VM is deployed, connect to the web app: http://10.10.238.86

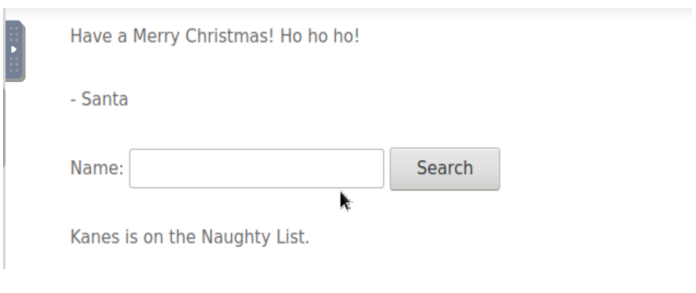


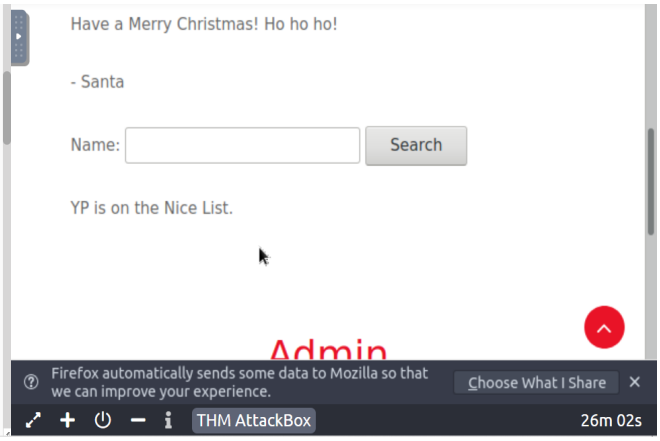


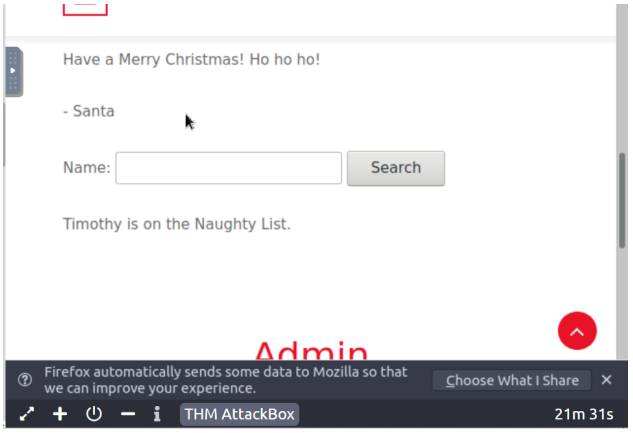


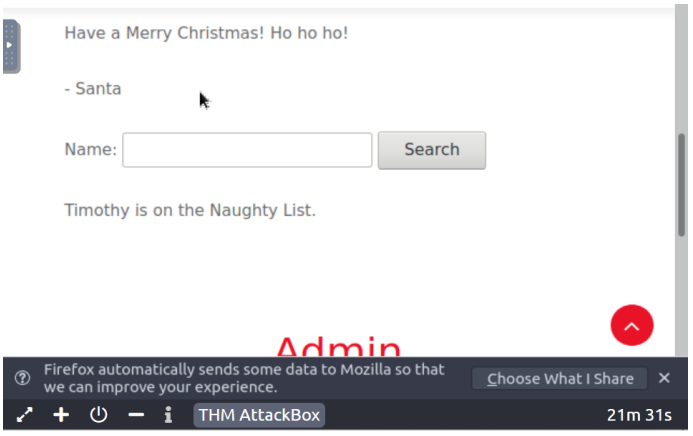






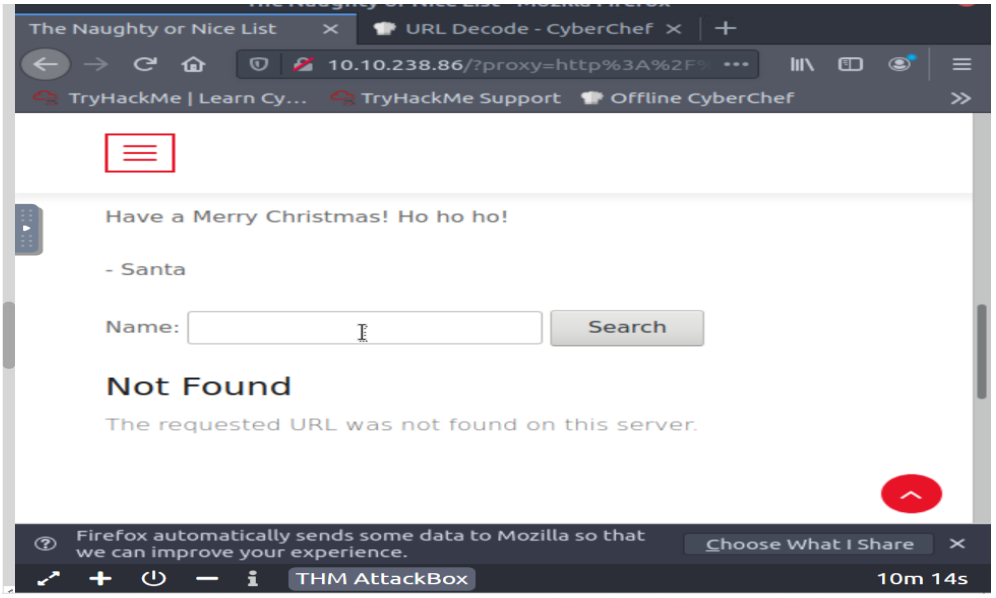






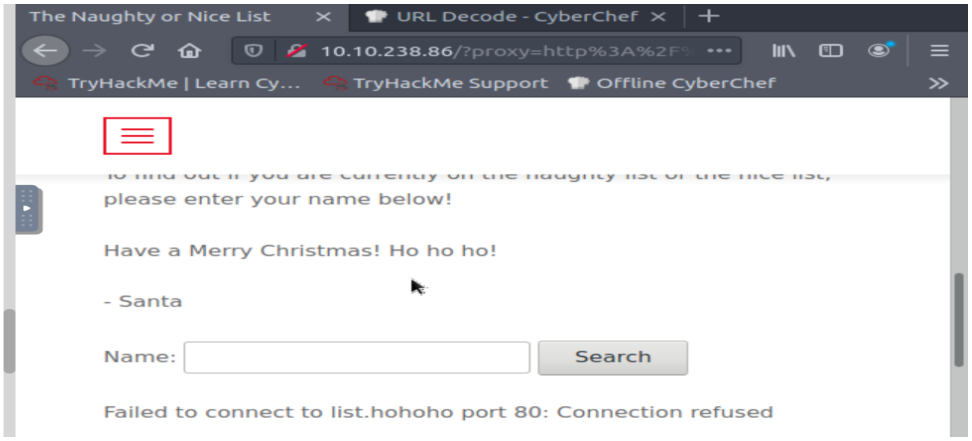
Question 2

when you use "/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F", "Not Found. The requested URL was not found on this server."appeared on the page.



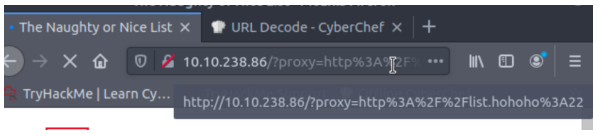
Question 3

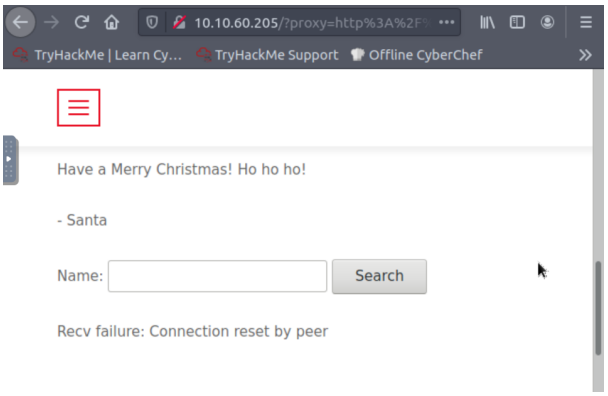
when you use "/?proxy=http%3A%2F%2Flist.hohoho%3A80", "Failed to connect to list.hohoho port 80: Connection refused" appeared on the page.



Question 4

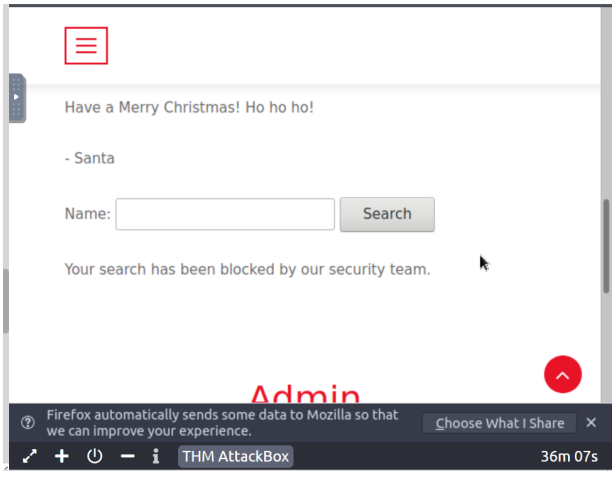
when you use "/?proxy=http%3A%2F%2Flist.hohoho%3A22", "Recv failure: Connection reset by peer" appeared on the page.





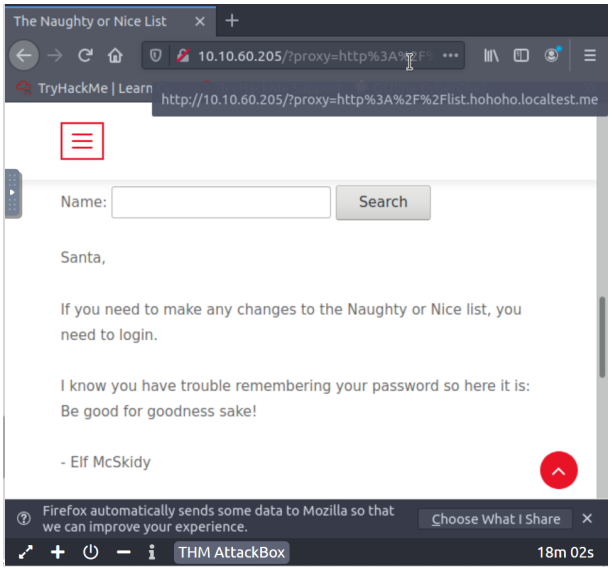
Question 5

when you use "/?proxy=http%3A%2F%2Flocalhost", "Your search has been blocked by our security team." appeared on the page.



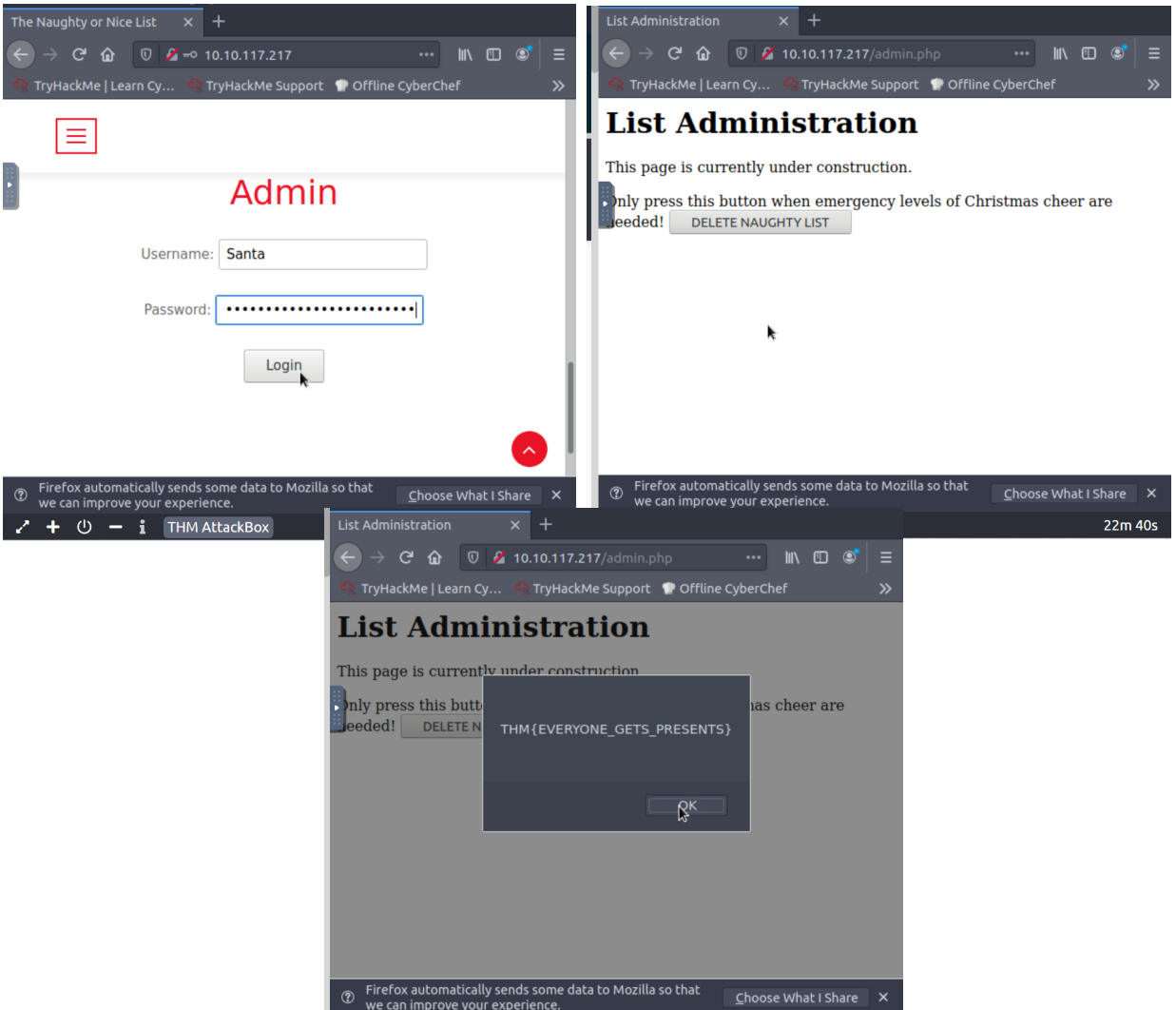
Question 6

Santa's password is "Be good for goodness sake!" by key in the website, "http://MACHINE_IP/?proxy=http%3A%2F%2Flist.hohoho.localtest.me"



Question 7

Kindly scroll down the page. To login to the admin page, key in username with "Santa" and with the password " Be good for goodness sake!". We are then directed to a page with the title "List Administration". Tap the "Delete Naughty List" button. "THM{EVERYONE_GETS_PRESENTS}" appeared on the screen.



Thought Process:

Open firefox, type in 10.10.177.223 to direct to The Naughty or nice list. The page appears with a search machine. Type the names to determine who is on the naughty list or the nice list. The result will be shown below the search machine, for example, "Tib3rius is on the Nice List.". When we type the name in click search, after the page is loaded, we have the proxy parameter at the firefox search engine, such as "/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%3DTib3rius". This is the proxy parameter and a URL. while "http%3A%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%3DTib3rius" is URL encoded. To fetch the root of the same site. Browse to "http://MACHINE_IP/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F", we will see "Not Found. The requested URL was not found on this server." on the page. It seems like a generic 404 message, indicating that we were able to make the server request the modified URL and return the response. Next, change the port number from 8080 to just 80 (the default HTTP port) "http://MACHINE_IP/?proxy=http%3A%2F%2Flist.hohoho%3A80" it appeared with the text "Failed to connect to list.hohoho port 80: Connection refused" which suggests that port 80 is not open on list.hohoho. Then, changing the port number to 22 " http://MACHINE_IP/?proxy=http%3A%2F%2Flist.hohoho%3A22" The message now changes to "Recv failure: Connection reset by peer" which suggests that port 22 is open but did not understand what was sent. Lastly, try this "http://MACHINE_IP/?proxy=http%3A%2F%2Flocalhost", the message returned says "Your search has been blocked by our security team.". To access local services, go to "http://MACHINE_IP/?proxy=http%3A%2F%2Flist.hohoho.localtest.me", the web server running locally, and it has a message that contains a password to login to the admin page. Key in username with "Santa" and with the password given, we will be directed to the List administration, the admin.php page. Lets delete the naughty list and we get "THM{EVERYONE_GETS_PRESENTS}".

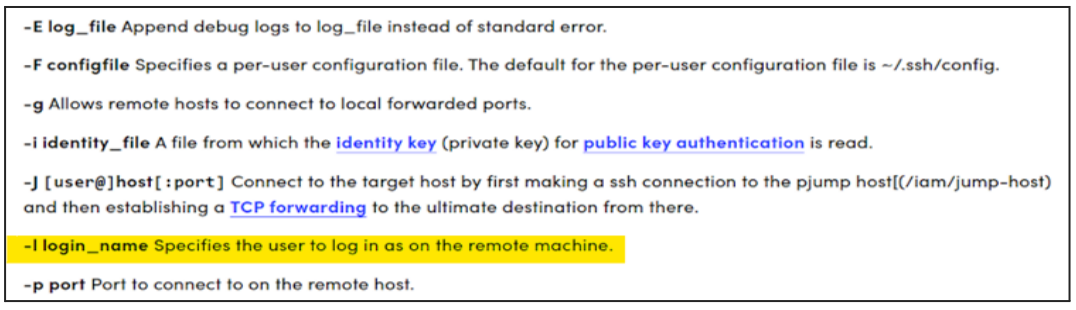
Day 20: [Blue Teaming] PowershELlF to the rescue

Tools used: THM attack box

Solutions:

Question1

By conducting a google search, we can discover from the ssh manual that parameter-l functions as a login name on the remote machine.



-E log_file Append debug logs to log_file instead of standard error.

-F configfile Specifies a per-user configuration file. The default for the per-user configuration file is ~/.ssh/config.

-g Allows remote hosts to connect to local forwarded ports.

-i identity_file A file from which the identity key (private key) for public key authentication is read.

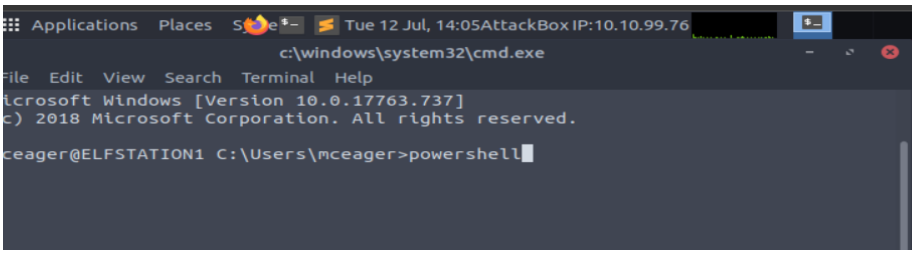
-J [user@]host[:port] Connect to the target host by first making a ssh connection to the pjump host[(/iam/jump-host) and then establishing a TCP forwarding to the ultimate destination from there.

-l login_name Specifies the user to log in as on the remote machine.

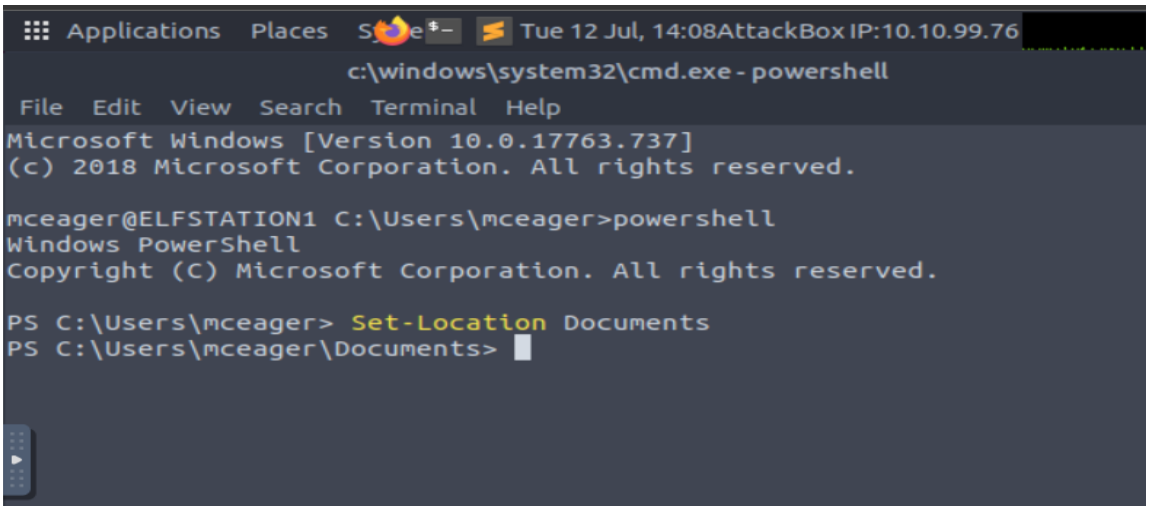
-p port Port to connect to on the remote host.

Question2

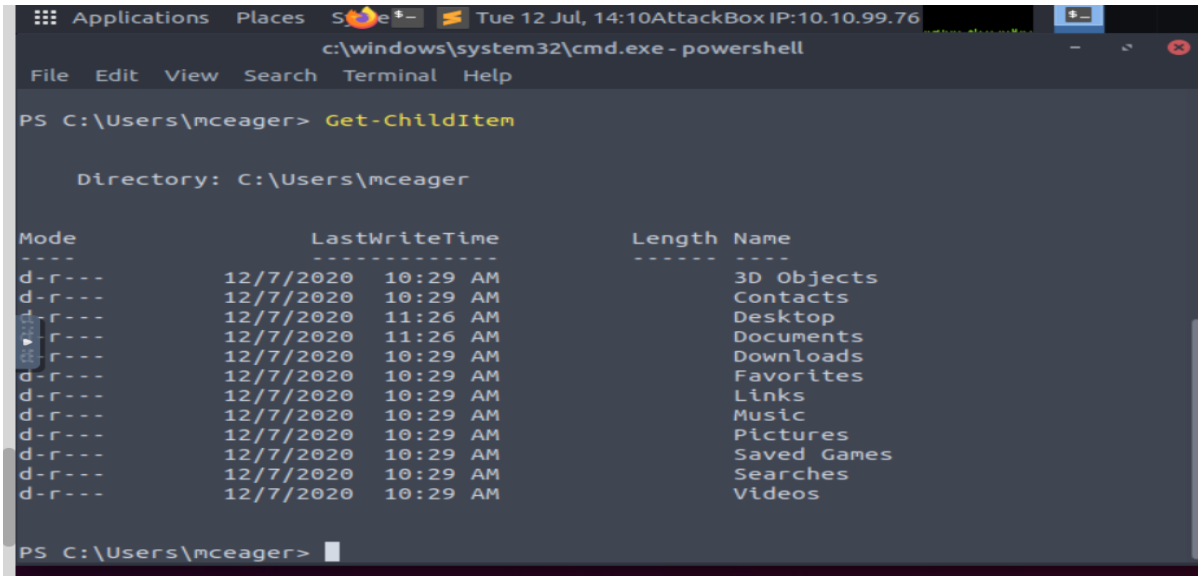
First, we will need to launch the powershell.



Then, we change the location for the documents.

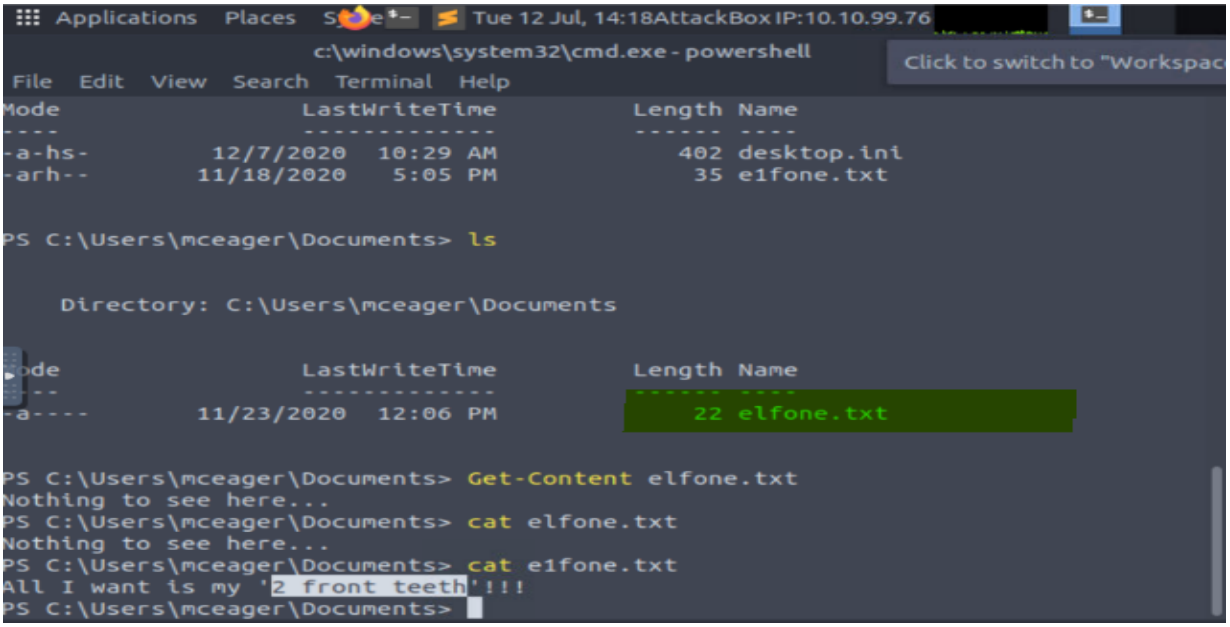


To get the directory of child-item, we insert Get-ChildItem -File -Hidden.



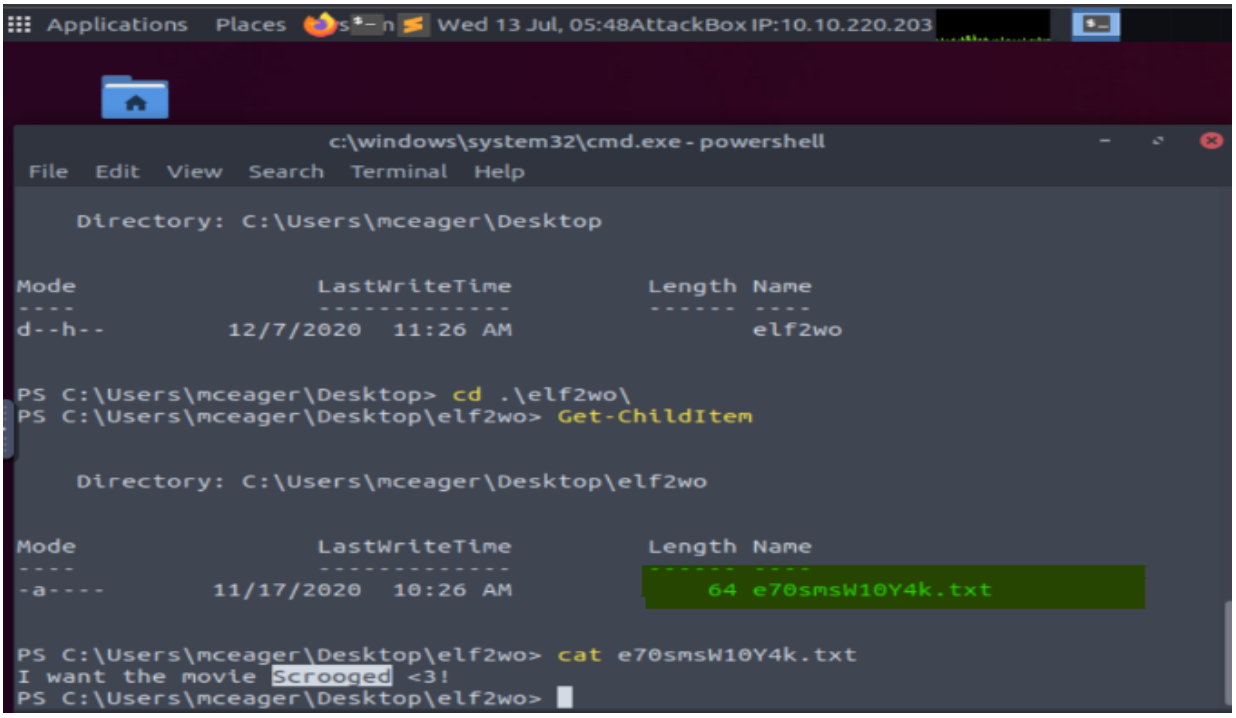
```
PS C:\Users\mceager\Documents> Get-ChildItem -File -Hidden


    Directory: C:\Users\mceager\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-hs-         12/7/2020  10:29 AM            402 desktop.ini
-arh--        11/18/2020   5:05 PM             35 e1fone.txt


PS C:\Users\mceager\Documents>
```

Here, we will see a hidden file called elfone.txt, then we insert cat e1fone.txt, it will display '2 front teeth' which the Elf1 wants



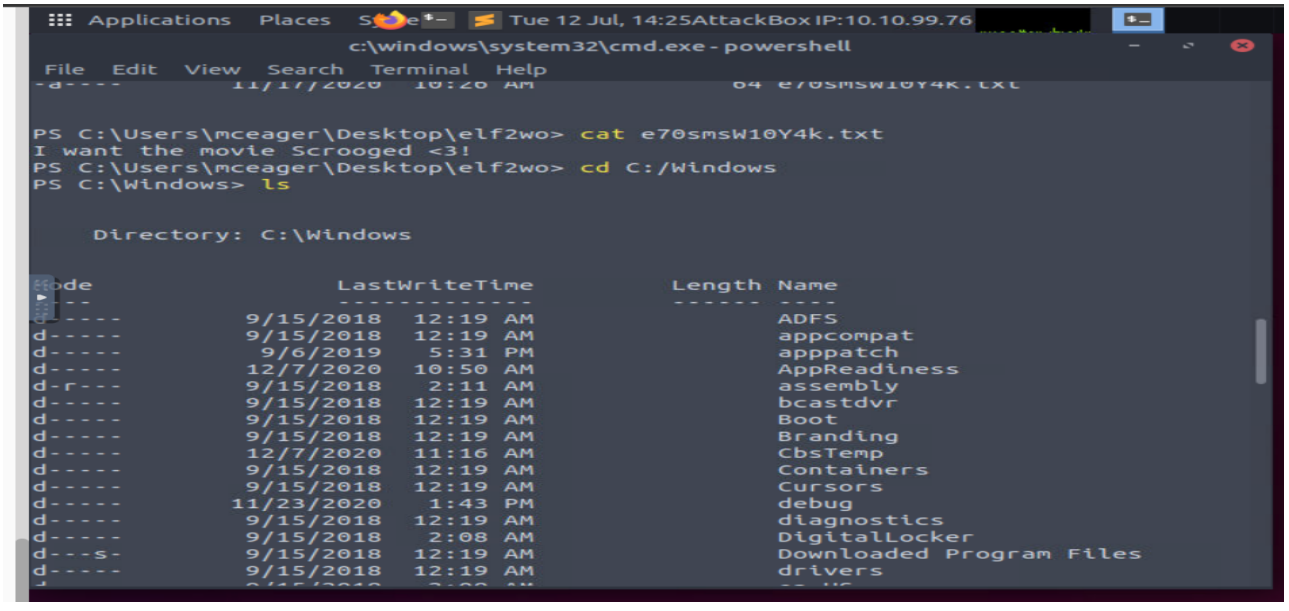
Question3

We searched the content of the directory and we saw a file called e70smsW10Y4k.txt, then we pasted in the file elf2wo and we got the movie name Scrooged.

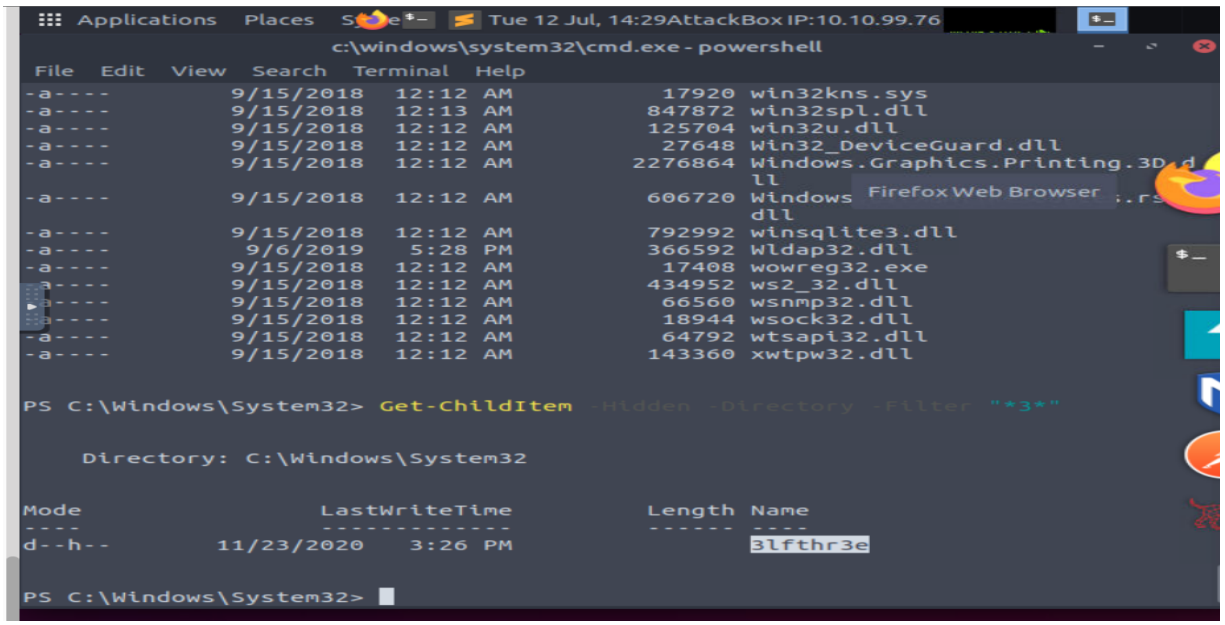


Question4

After that we searched the hidden directory in windows, it showed many unnecessary items ,it makes it hard to find the things we want.

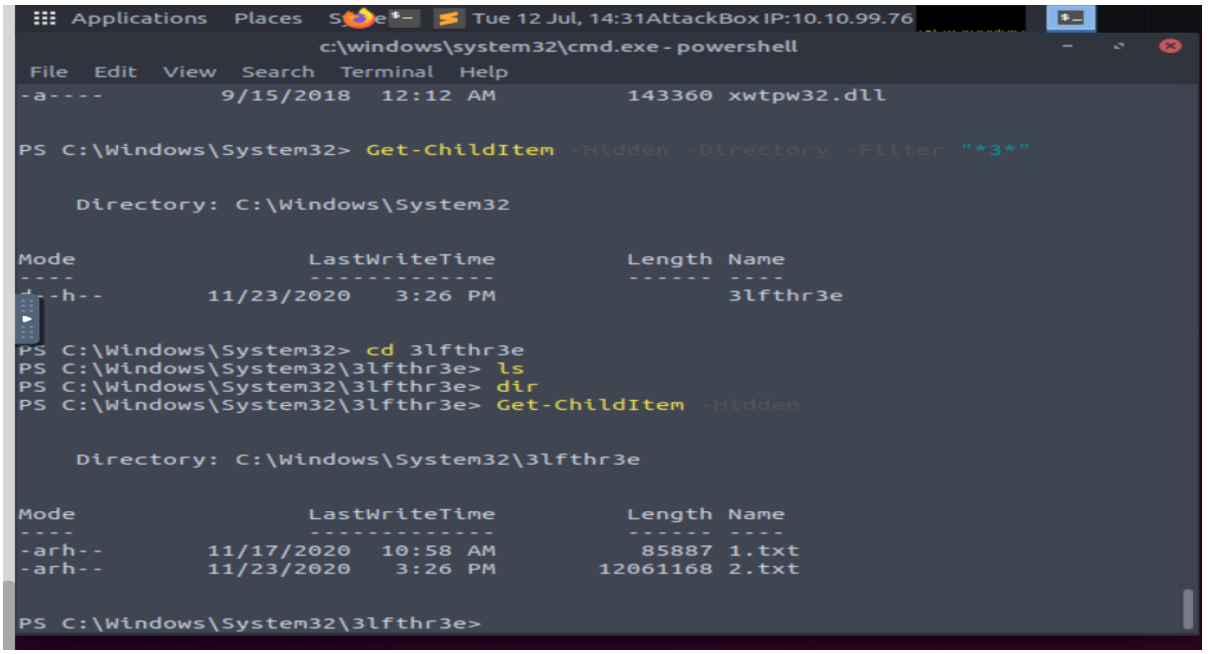


So, we insert command Get-ChildItem -Hidden -Directory -Filter “*3*” in the directory \windows\system32 and we got the name of the hidden folder which is 3lfthr3e.

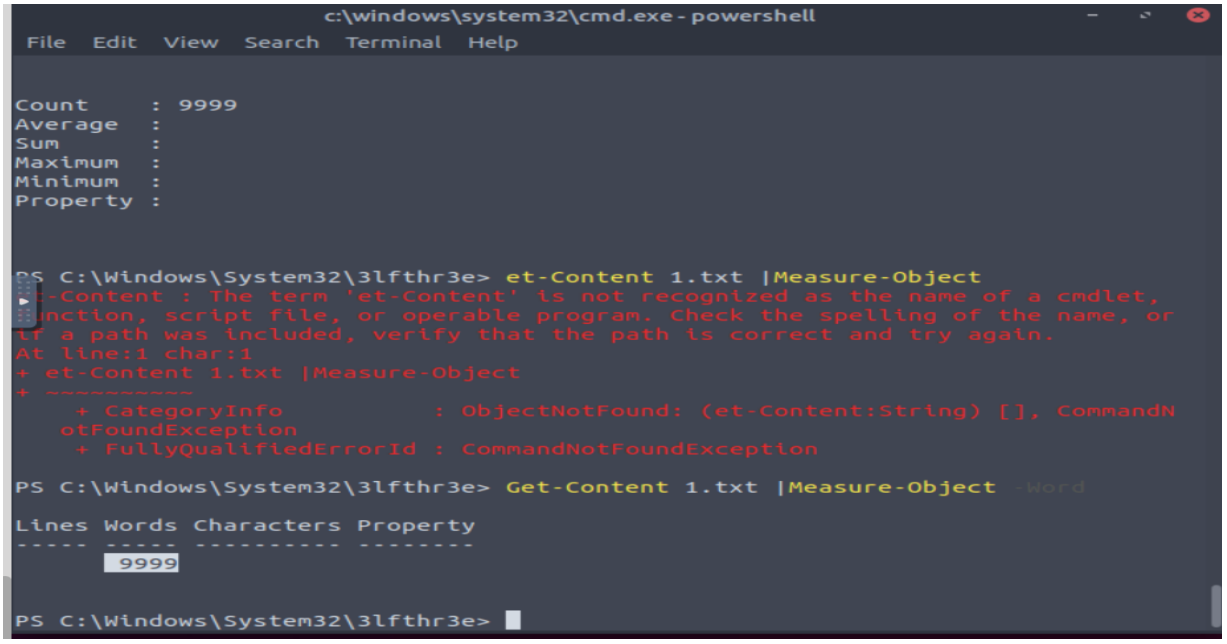


Question5

We searched through the hidden folder we got and there are 2 txt files inside.

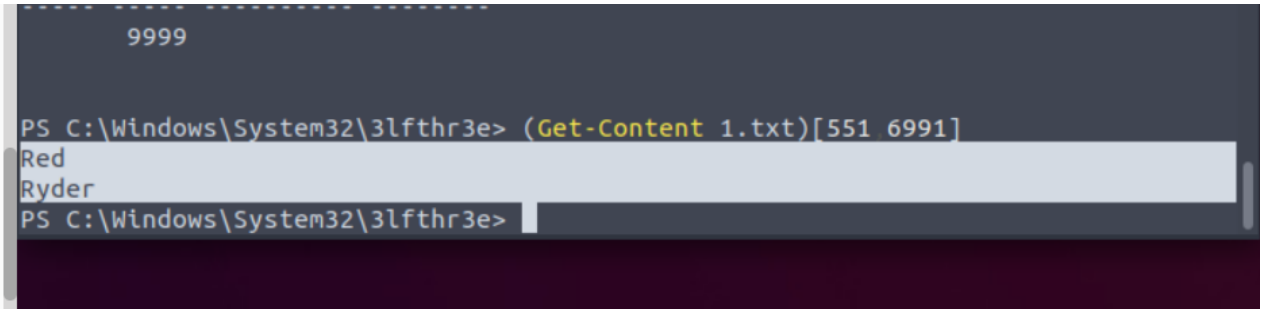


To find how many words does the first file contains, we need insert command Get-Content 1.txt | Measure-Object -Word and we can see that the file has 9999 words.



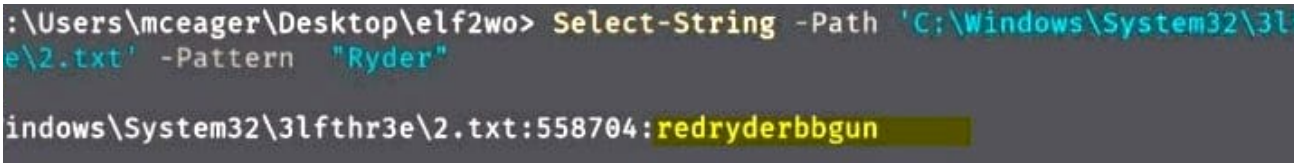
Question6

By inserting the command which is (Get-Content 1.txt)[551,6991], we can see red, ryder these two words at index 551 and 6991 in the first file.



Question7

Lastly, we need to look in the second file for the wording from the first query. We insert command Select-String -Path -Pattern "Ryder" and we know elf3 want redryderbbgun.



Thought/ methodology:

First, we launched powershell and set the location to the documents folder after connecting to the machine via ssh -l mceager MACHINE IP: Documents Set-Location Get-ChildItem -File -Hidden should be used to view a hidden file first. A hidden file with the name "e1fone.txt" is present. When we use Get-Content .e1fone.txt to read the contents of the file, we discover that it contains a message about two front teeth. Then we pasted the file elf2wo into the directory's e70smsW10Y4k.txt file after searching for it, the

name of the movie Scrooged appeared. After that we searched the hidden directory in windows, it showed many unnecessary items ,it makes it hard to find the things we want. So, we insert command Get-ChildItem -Hidden -Directory -Filter “*3*” in the directory \windows\system32 and we got the name of the hidden folder which is 3lfthr3e. We will proceed to see Two text files are located in this directory. We need to insert the command Get-Content 1.txt | Measure-Object -Word to determine how many words are in the first file, and we can see that there are 9999 words in the file. In the first file, at indexes 551 and 6991, we can see the words red and ryder by inserting the command (Get-Content 1.txt)[551,6991]. In order to find the wording of the first query, we must eventually search the second file. When we enter the command Select-String -Path -Pattern "Ryder," we can be certain that elf3 wants redryderbbgun.