PSP0201

Week 5

Writeup

Group Name: Hack Me No

Members

ID	Name	Role
1211102630	Chan Kar Kin	Leader
1211100925	Ang Jin Nan	Member
1211103311	Ng Yun Shi	Member
1211102777	Tai Qi Tong	Member

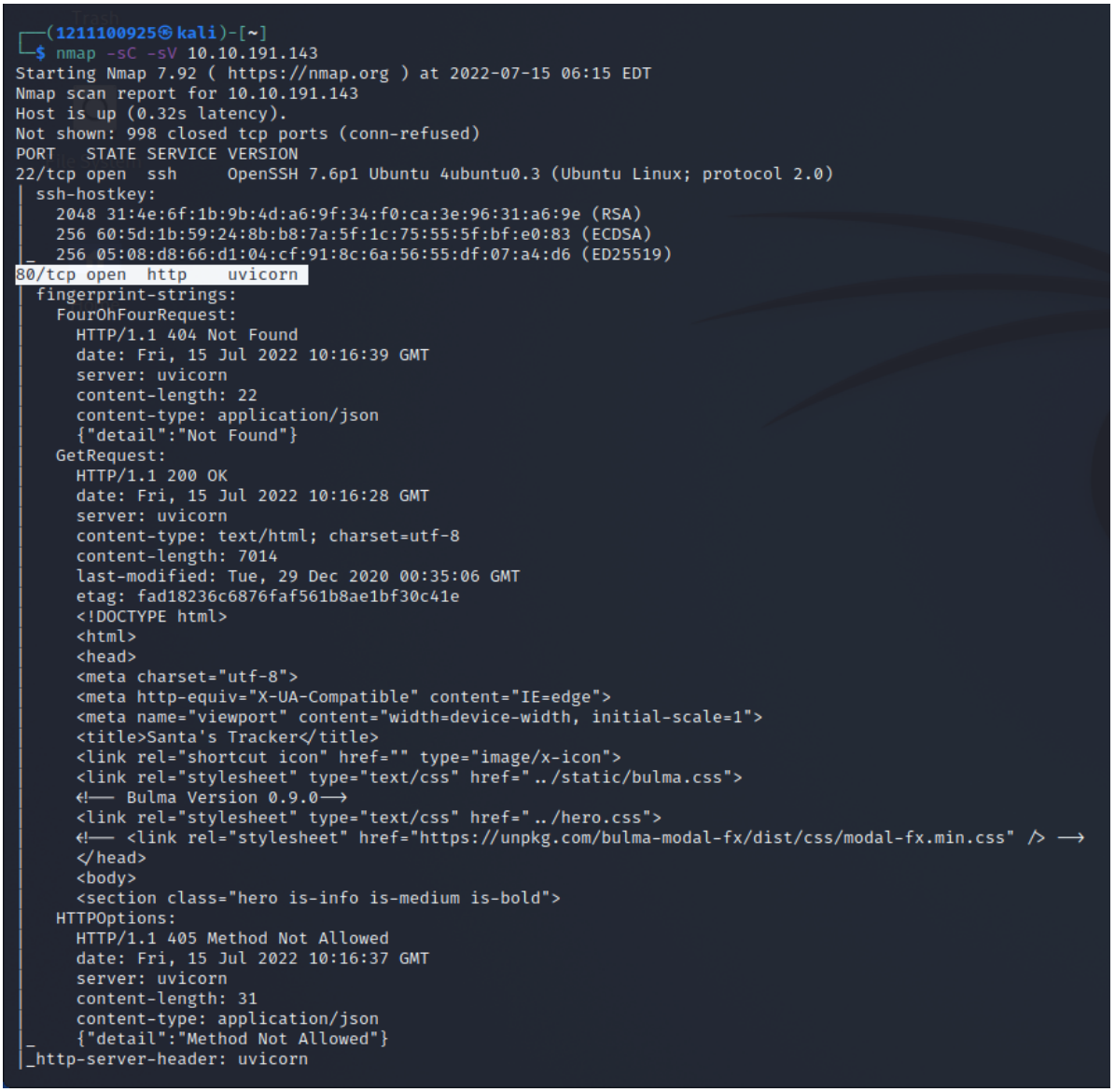
Day 16: Scripting – Help! Where is Santa?

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

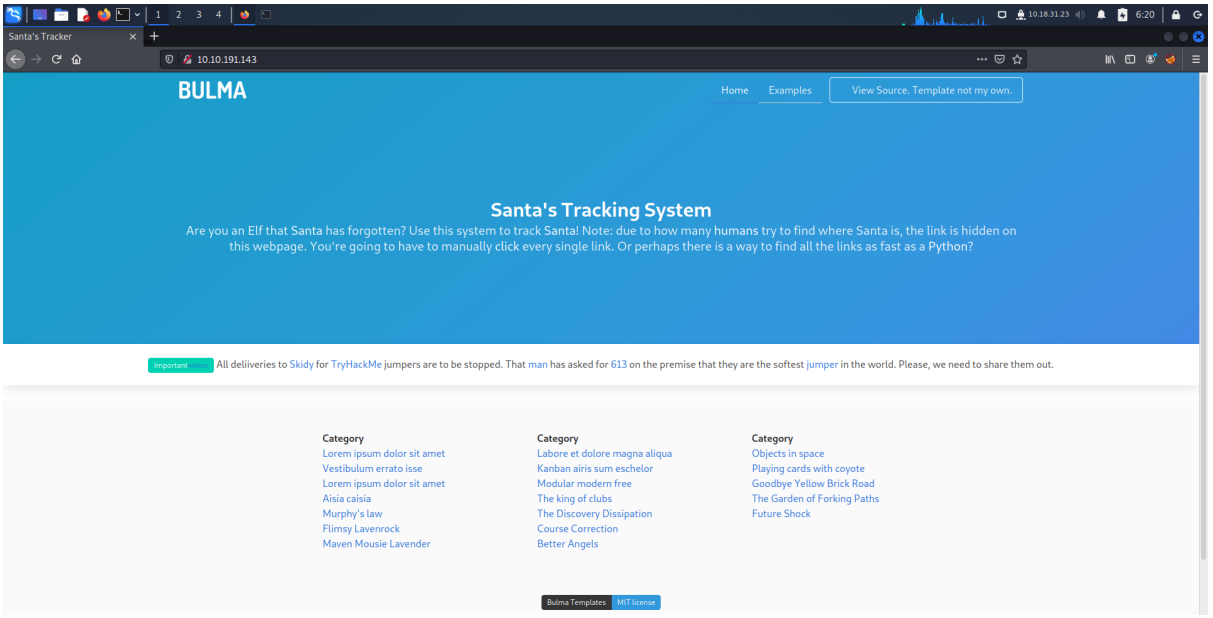
Question 1

We first start to find the port number using nmap. We found a http port is open, and the port number is **80.**



Question 2

After finding the port number, we open our browser and search for our target machine's IP address. A webpage appeared, we found that the templates used by this webpage is called **BULMA** which is located at the top left of the website.



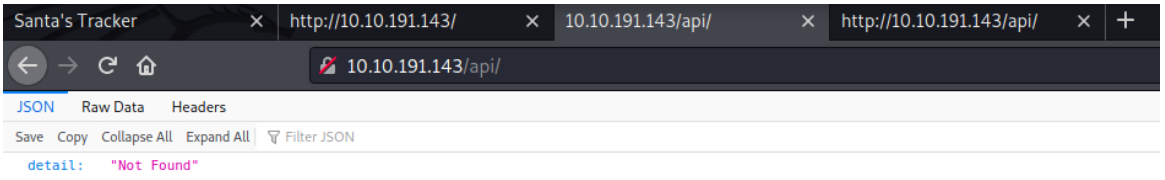
Question 3

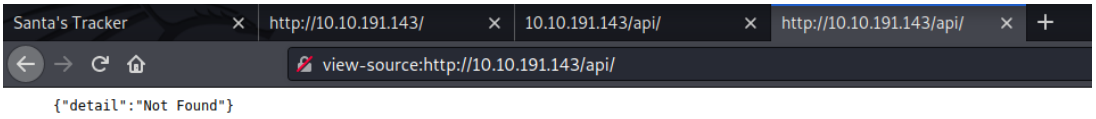
We then proceeded to look at the source code of the main page, and found a URL in the source code. By looking at the URL, we can know that the directory for the API is **/api/**.

```
<div class="column is-3">
    <h2><strong>Category</strong></h2>
    <ul>
        <li><a href="#">Labore et dolore magna aliqua</a></li>
        <li><a href="#">Kanban airis sum eschelor</a></li>
        <li><a href="http://machine_ip/api/api_key">Modular modern free</a></li>
        <li><a href="#">The king of clubs</a></li>
        <li><a href="#">The Discovery Dissipation</a></li>
        <li><a href="#">Course Correction</a></li>
        <li><a href="#">Better Angels</a></li>
    </ul>
</div>
```

Question 4

We navigated to the page and tested the page without a API key. A webpage appeared, we then check on the source code and found the raw data which is **{"detail":"Not Found"}**.





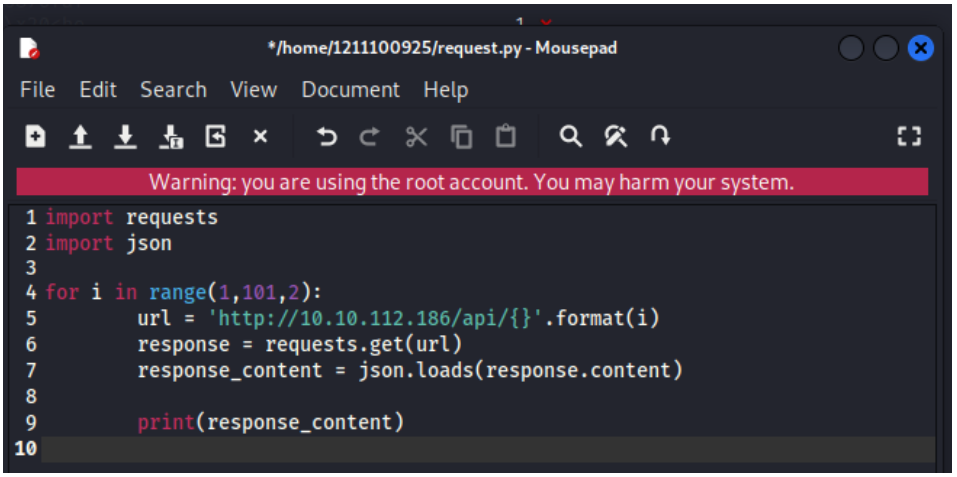
Question 5

By looking at the hint provided from the question, we know that the correct API key is an odd number between 0-100 and Santa's Sled will block our IP address if we tried too many times.

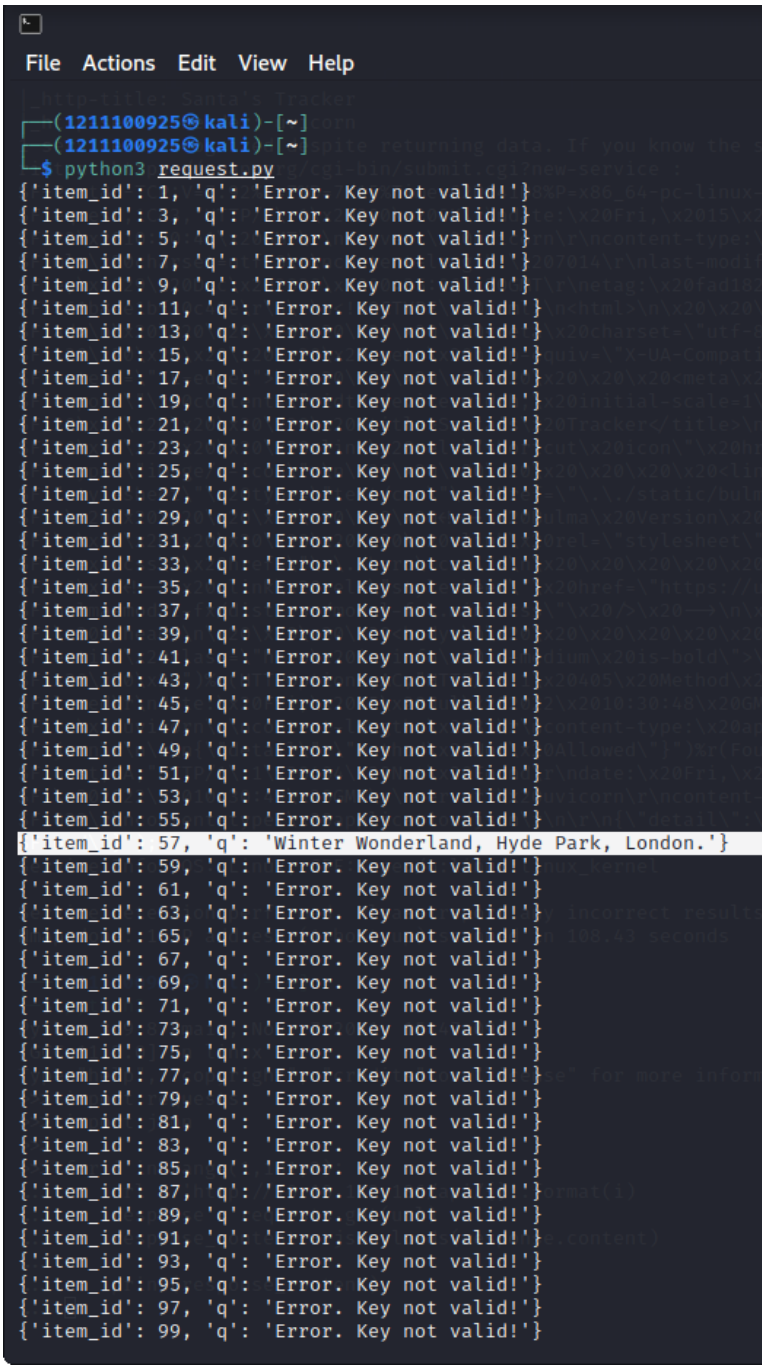
Find out the correct API key. Remember, this is an odd number between 0-100. After too many attempts, Santa's Sled will block you.

To unblock yourself, simply terminate and re-deploy the target instance (MACHINE_IP)

We opened our terminal and started a new python file called request.py to run through odd numbers appended to the URL and print out if we get the correct API key.

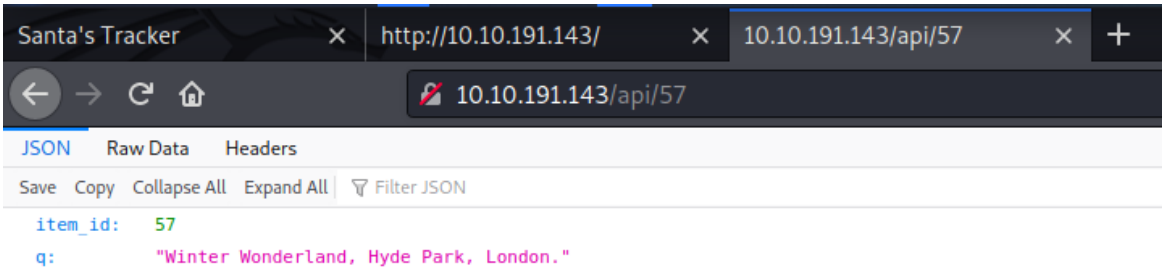


After writing the python file, we start to run the file in our terminal. We then found the correct API key of the webpage and Santa's location which is **Winter Wonderland, Hyde Park, London**.



Question 6

We navigate to that page with the API key **57**, and different wepage had appeared.



Thought Process/Methodology:

To search for the port number of the IP address, we used the nmap scan. A open port has shown, which is port number **80**. After that, we search our attack machine's IP address on our browser and a webpage appeared. We found that the webpage used **BULMA** as their templates. We proceed to look for the source code and found a URL in the source code. By looking it, we know that the directory for API is **/api/**. Next, we tried the URL without typing the API key, and found the raw data of that page which is **{"detail":"Not Found"}**. By looking at the question given from TryHackMe, we know that when we try a different API key for many times, our IP address will be block. The hint given by the question is the correct API key is an odd number between 0-100. After knowing the hint, we open a new python file and write a code into it that could let us know which is the correct API key. After writing it, we start the program and run it in our terminal. The correct API key was shown and we also get to know Santa's location which is **Winter Wonderland, Hyde Park, London**. We then tried the correct API key using our browser and the result was different with other wrong correct API key. From this, we know that **57** is the correct API key.

Day 17: Reverse Engineering – ReverseELFneering

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

Question 1

Match the data type with size in bytes based on the notes given in TryHackMe.

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

Read the notes in TryHackMe, it is given that command **aa** is used to analyse program in radare2

> This will open the binary in debugging mode. Once the binary is open, one of the first things to do is ask r2 to analyze the program, and this can be done by typing in: `aa`

Question 3

TryHackMe notes stated that command to set a breakpoint in radare2 is **db**

> 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.

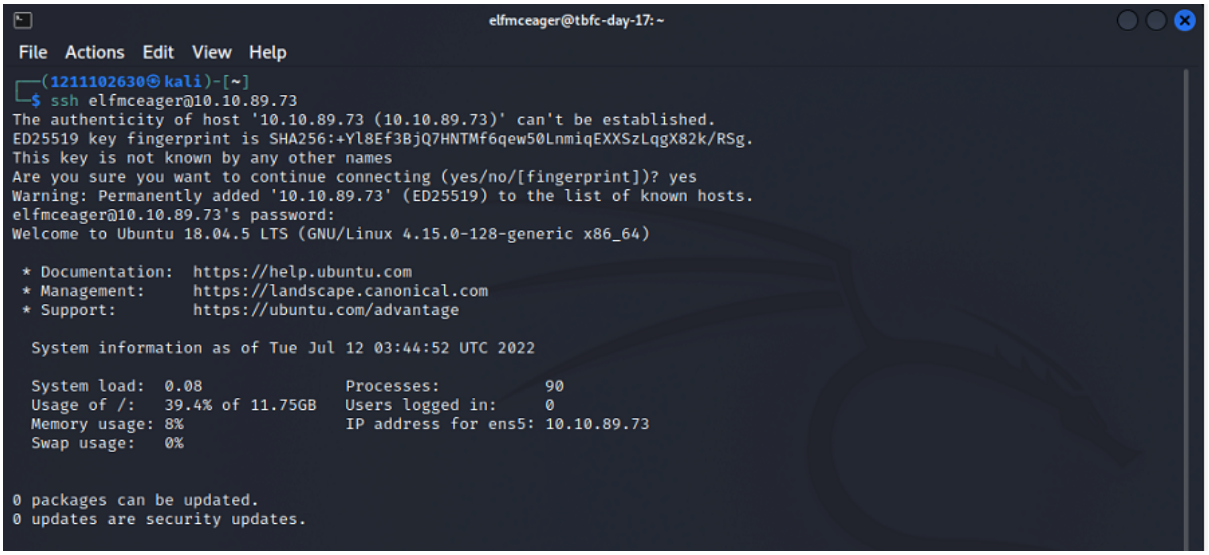
Question 4

Read the TryHackMe notes, command to execute the program until we hit a breakpoint is **dc**

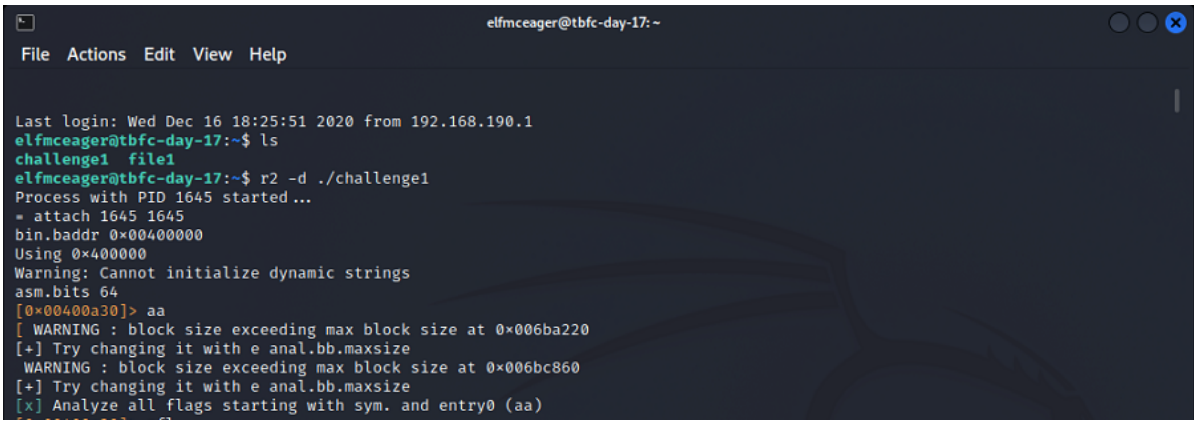
> Running `dc` will execute the program until we hit the breakpoint. Once we hit the breakpoint and print out the main function, the rip which is the current instruction shows where execution has stopped. From the notes above, we know that the **mov** instruction is used to transfer values. This statement is transferring the value 4 into the `local_ch` variable. To view the contents of the `local_ch` variable, we use the following instruction `px @memory-address` In this case, the corresponding memory address for `local_ch` will be `rbp-0xc` (from the first few lines of **@pdf main**) This instruction prints the values of memory in hex:

Question 5

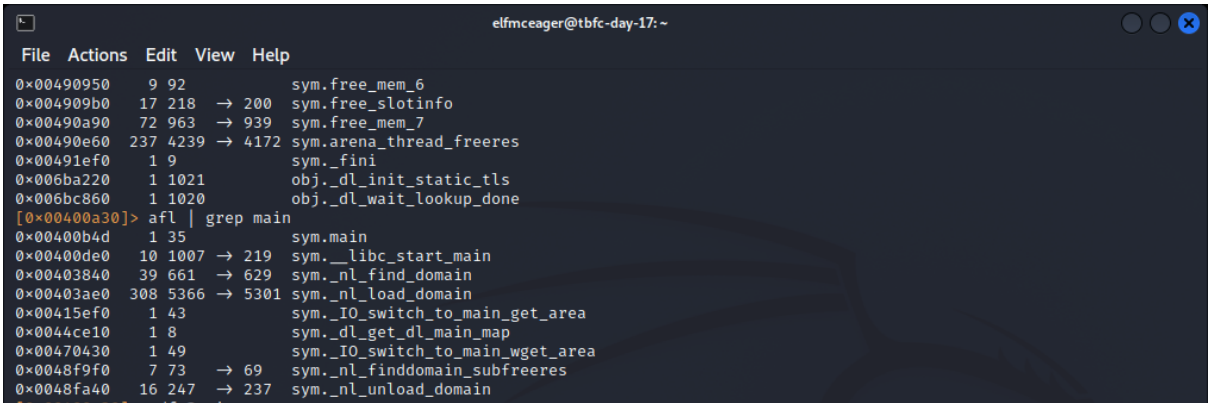
Access to target machine using ssh



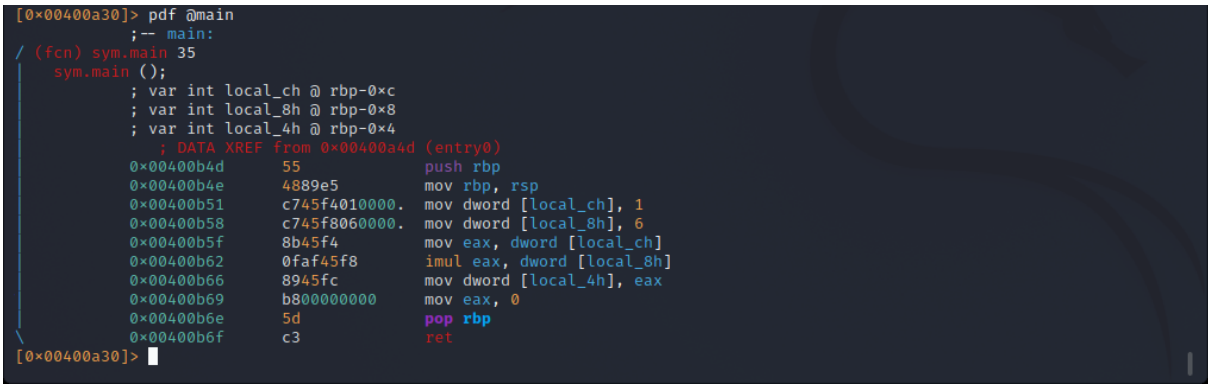
Open binary in debugging mode and ask r2 to analyse the program challenge1



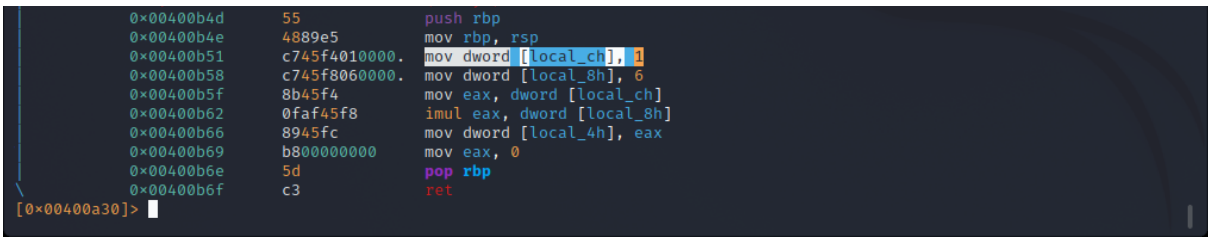
Find list of functions with the entry point called ‘main’



Examine the assembly code by running **pdf@main** command

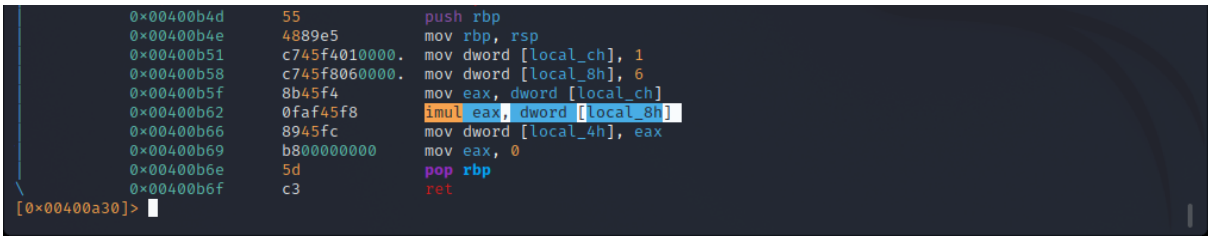


Based on the assembly code, the value 1 is moved to local_ch and local_ch gets the value **1**.



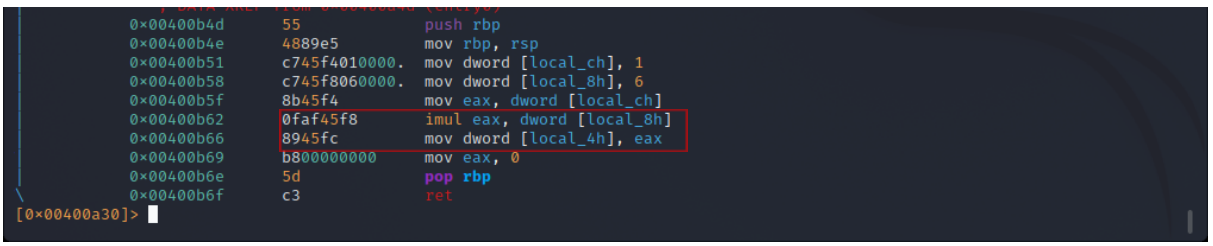
Question 6

Based on the assembly code, value 6 is copied into local_8h, value of local_ch which is 1 is copied to eax, imull instruction multiplies the values of local_8h and eax (6*1=6), and the value is return back to eax, so value of eax when imull instruction is called is **6**



Question 7

The value of eax previously is copied to local_4h, so the value of local_4h before it is set to 0 is 6



Thought Process/Methodology:

Data type is matched with size in bytes based on the notes given in TryHackMe. Read the notes given in TryHackMe and we found that the command to analyse the program in radare2 is **aa**. Based on the notes in TryHackMe, we also found that the command to set a breakpoint in radare2 is **db** and the command to execute the program in until we hit a breakpoint is **dc**. We access the target machine using ssh and open binary in debugging mode to ask r2 to analyse the program challenge1. We use 'afl | grep main' to find the list of functions with the entry point that is defined as main. Then, pdf@main command is run to examine the assembly code. According to the assembly code, the value 1 is copied to local_ch. Next, value 6 is copied into local_8h, and value of local_ch which is 1 is copied to eax. The imul instruction multiplies the values of local_8h and eax (6*1=6), and the value is returned back to eax, sowhen imul instruction is called we get an eax value of **6**. Next, the previous value of eax, which is 6, is copied to local_4h, so the value of local_4h before it is set to 0 is 6.

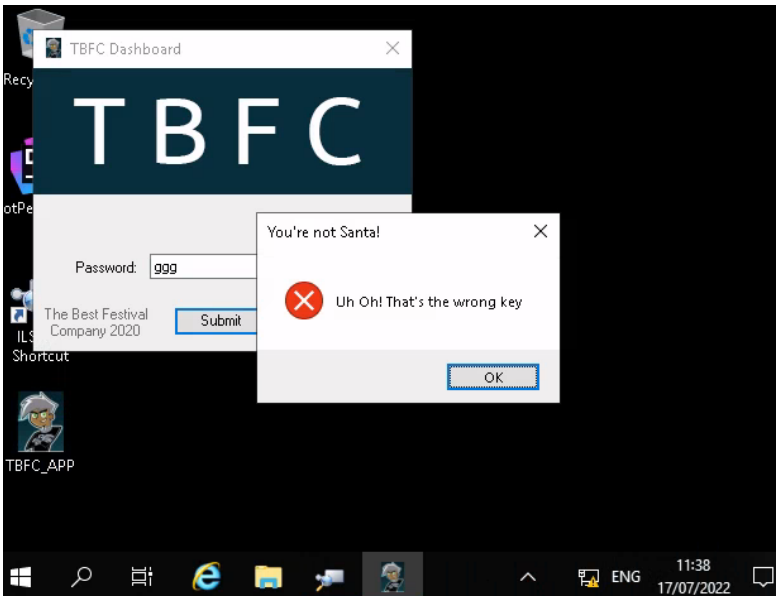
Day 18: Reverse Engineering – The Bits of Christmas

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

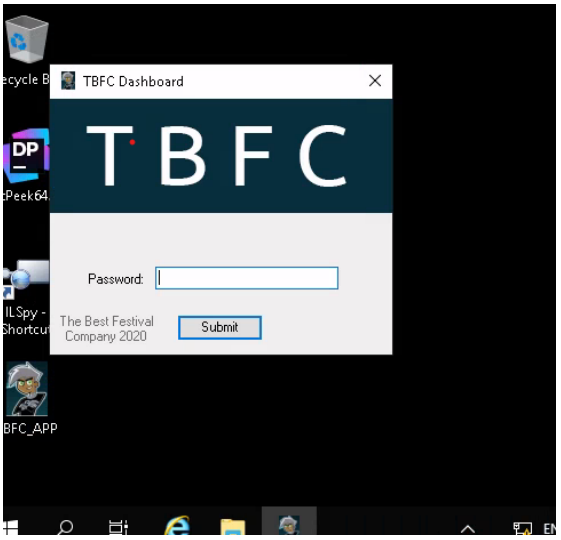
Question 1

The message will be "**Uh Oh! That's the wrong key**"



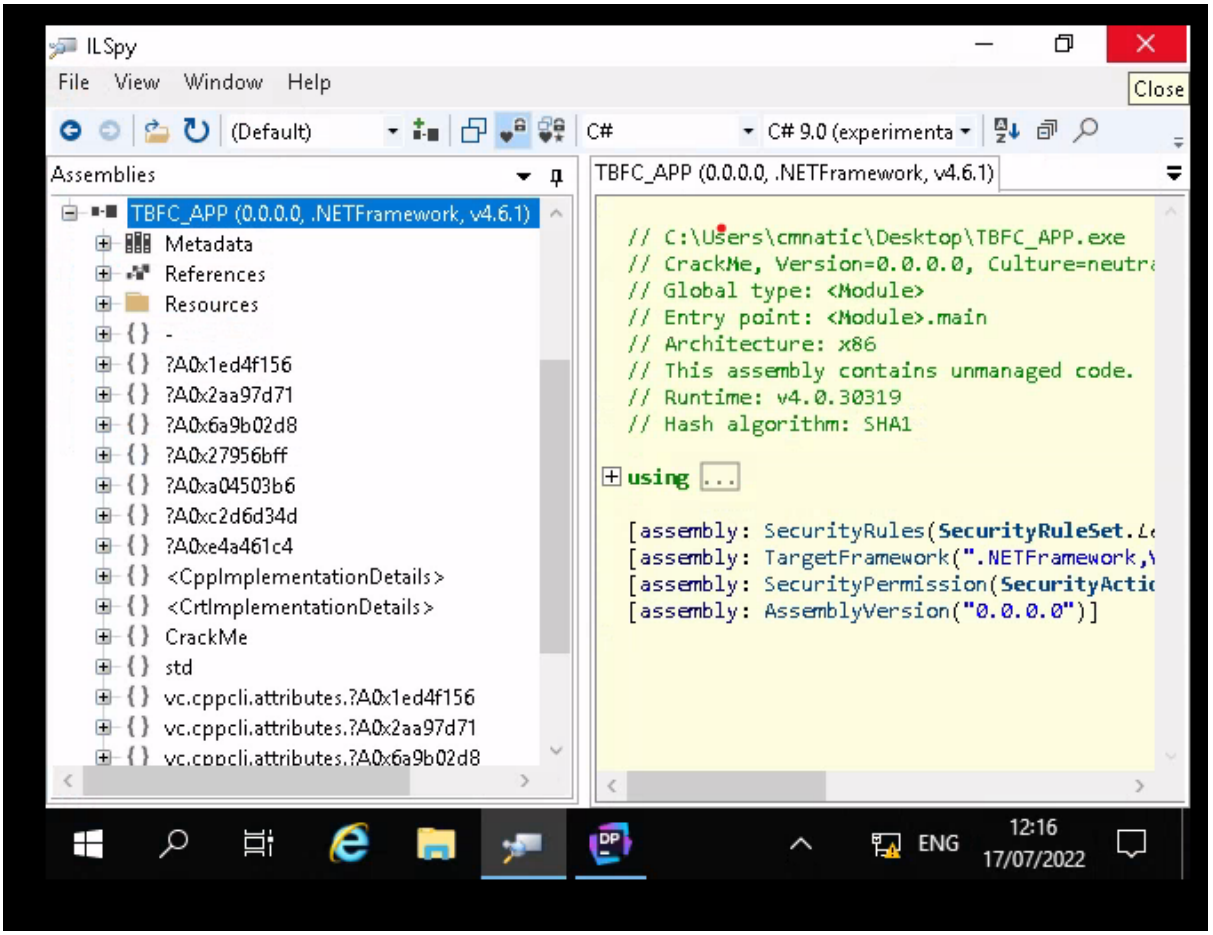
Question 2

TBFC stand for **The Best Festival Company 2020**



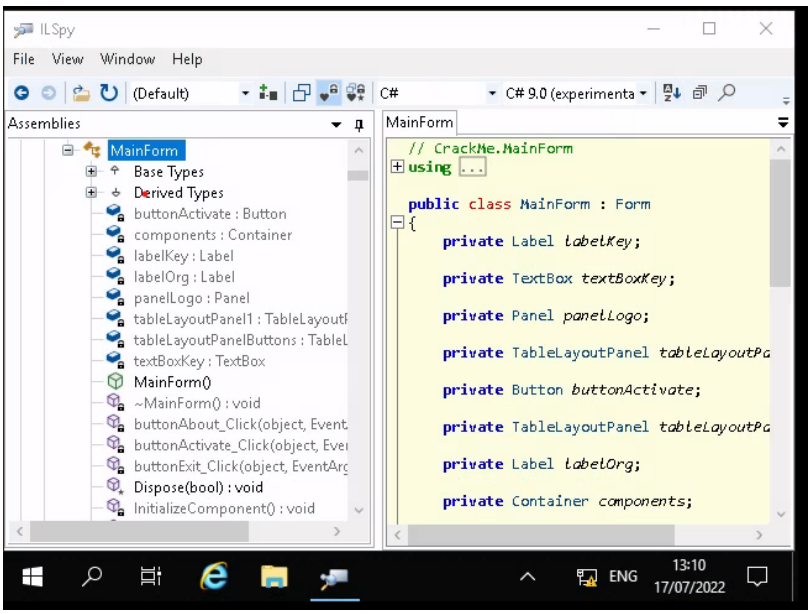
Question 3

The module that catches attention is **CrackMe**



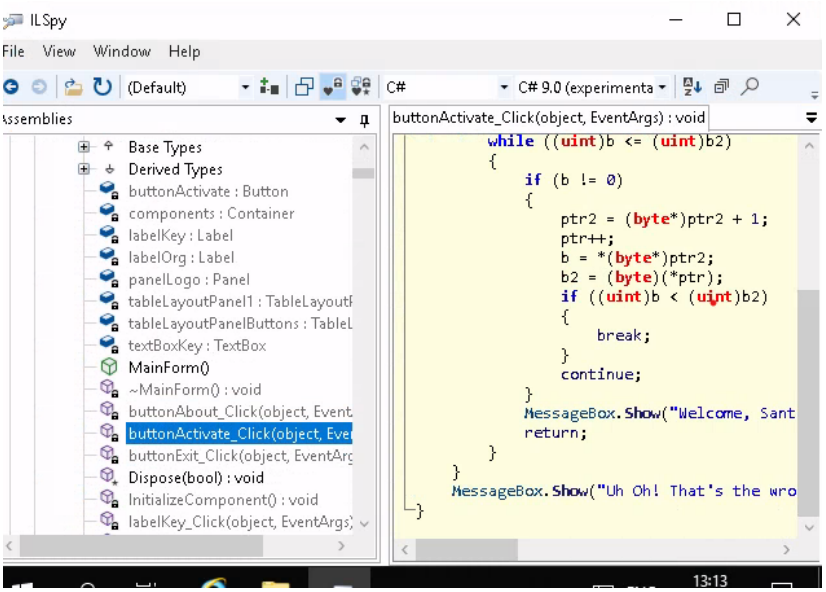
Question 4

MainForm contains the information we are looking for.



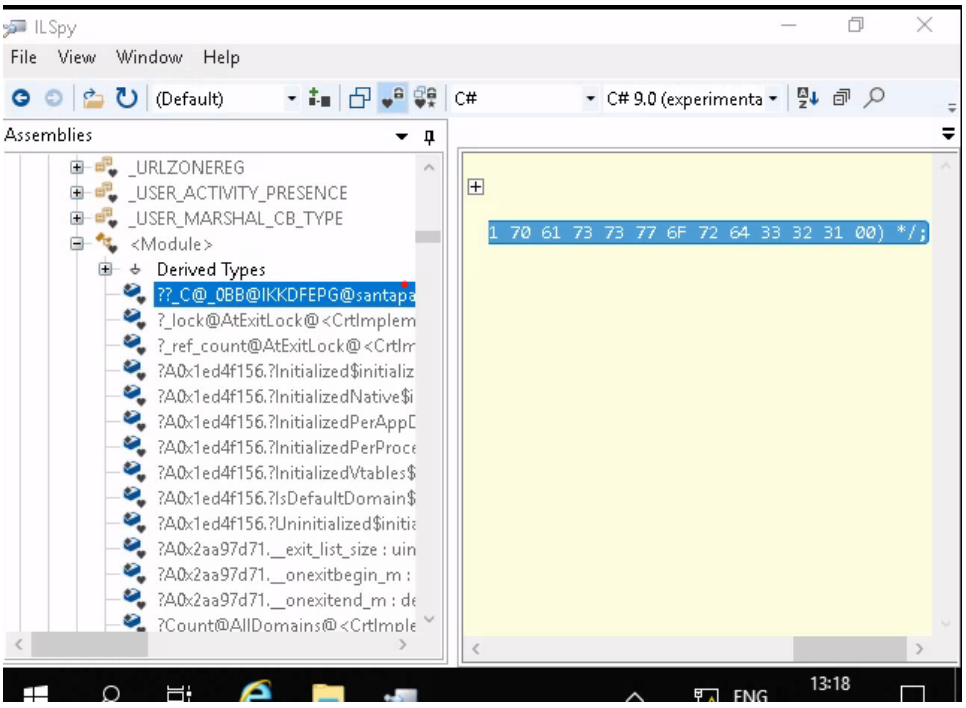
Question 5

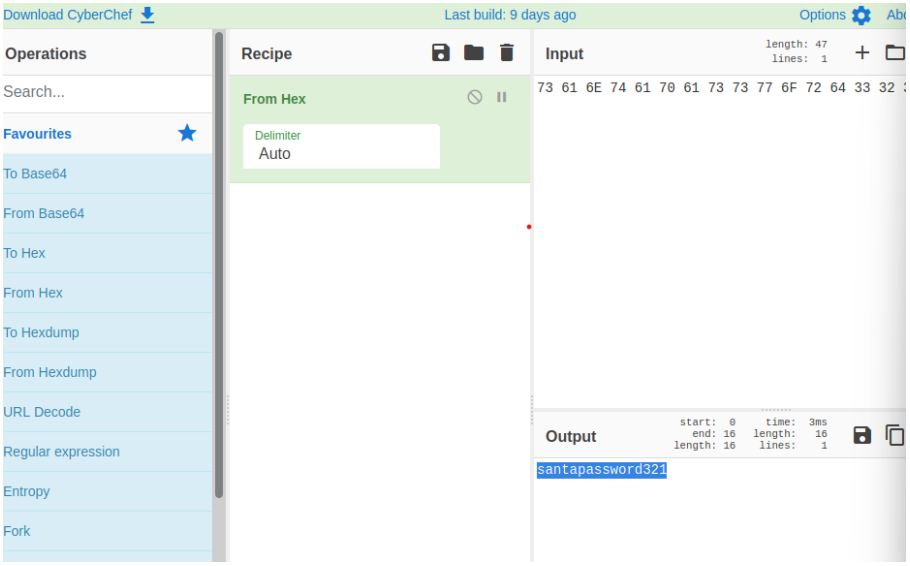
buttonActivate_Click contains the information we are seeking.



Question 6

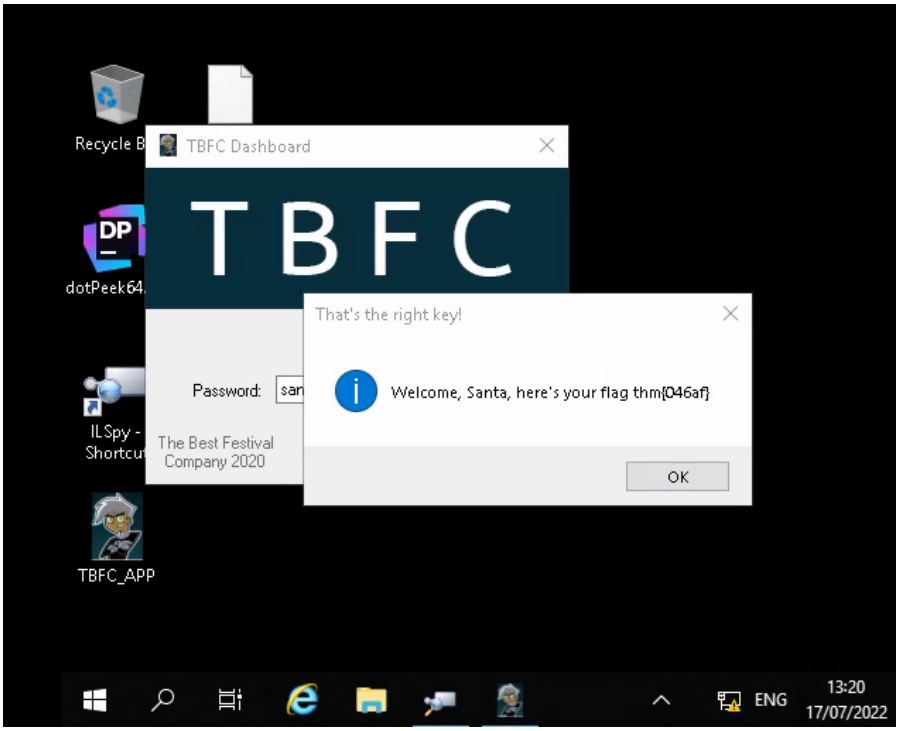
When we open buttonActivate_Click , we can see that the coding have some sensitive values and the password we can saw is santapassword123, but we are not sure whether it's the correct password or no, so we double click on it and it will direct us to another file which contain a hexadecimal number. We then use cyberchef to convert it into words and we can get the password that we are expecting, which is **santapassword321**.





Question 7

The flag will be **thm{046af}**



Thought Process/Methodology:

First of all, we should download a tool named Remmina by using the 'sudo apt-get install remmina' command. Then, we can start to connect to the machine using the username and password that have already been given by the website which is cmnatic for username and Adventofcyber! For password.Once we successfully log in, we will see a windows desktop.Then, we will open TBFC_APP and we will be asked to log in with password. After that, we can then open ILSpy and this will allow us to get a look at the source code of the app. We can then press CrackMe and click MainForm and finally look for

buttonActivate_Click. We can see inside the buttonActivate_Click source code, we can see that the coding have some sensitive values and the password we can saw is santapassword123, but we are not sure whether it's the correct password or no, so we double click on it and it will direct us to another file which contain a hexadecimal number. We then use cyberchef to convert it into words and we can get the password that we are expecting, which is santapassword123. We can then use the expected password to log in to the TBFC_APP and we will then get our flag.

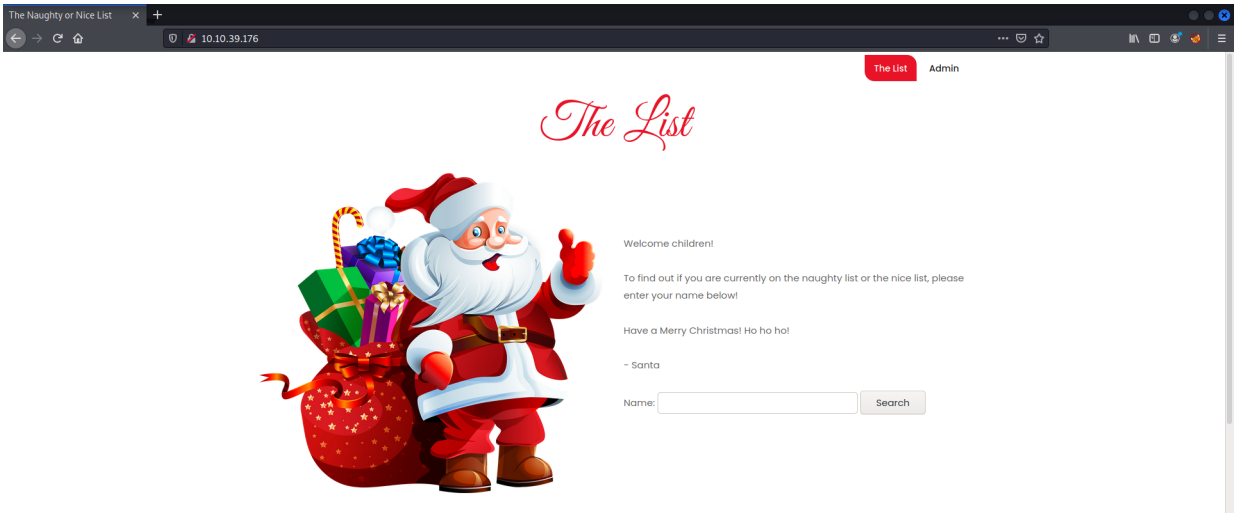
Day 19: Web Exploitation – The Naughty or Nice List

Tools used: Kali Linux, Firefox

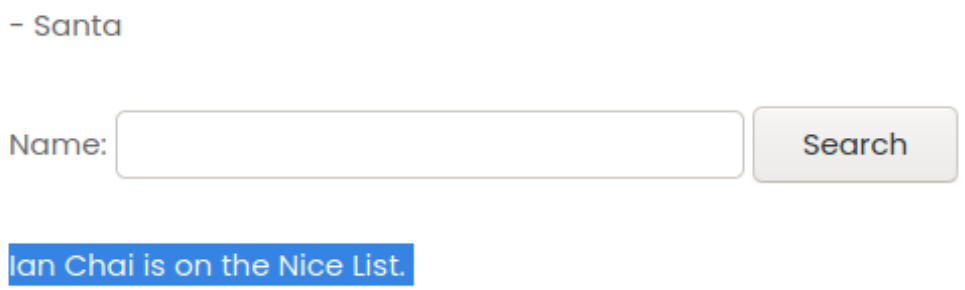
Solution/Walkthrough:

Question 1

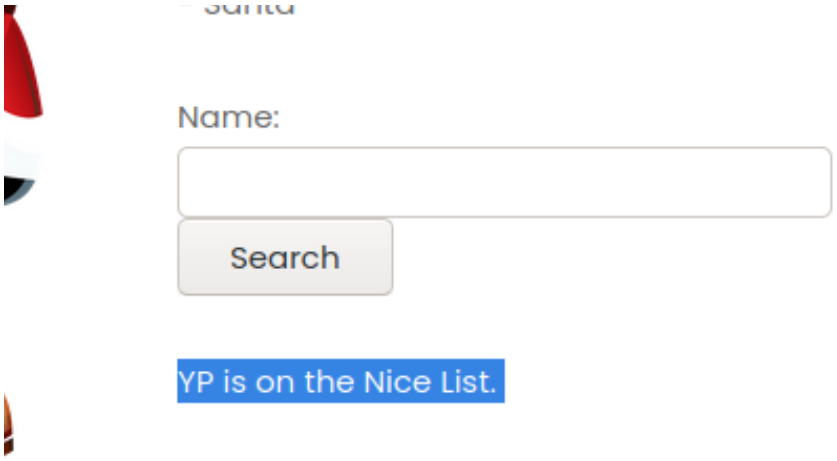
Once the attack machine is deployed, we proceed to connect to the web app. The webpage "The Naughty or Nice List" had appeared.



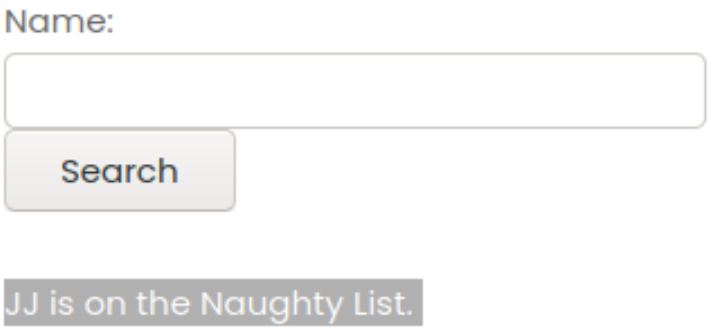
After opening the webpage, we enter a name in the form and click the "Search" button to see whether the name is on the Naughty List or the Nice List. We entered the first name, "**Ian Chai**" and found it is on the **Nice List**.



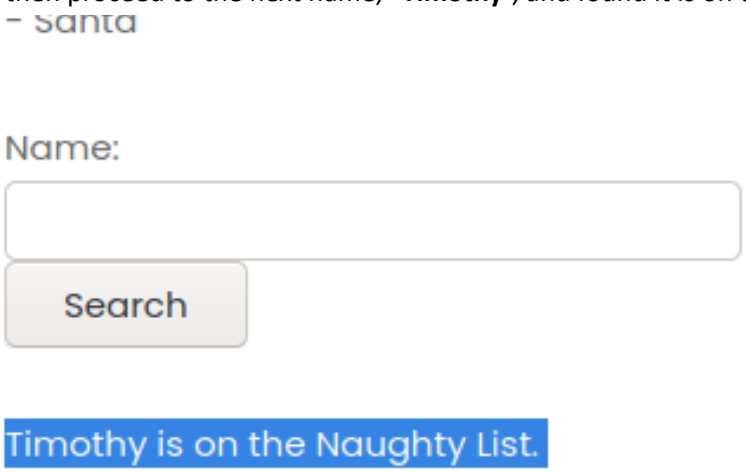
We then proceed to the next name which is "**YP**" and found it is on the **Nice List** too.



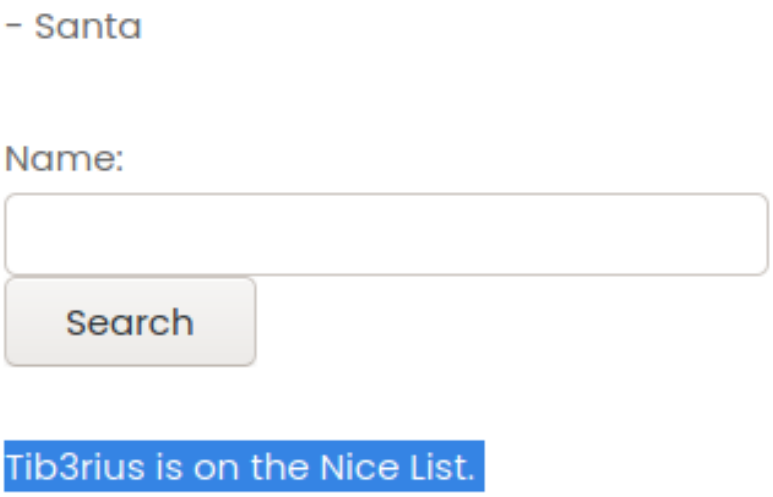
Next, we entered the name "**JJ**" and found it is on the **Naughty List**.



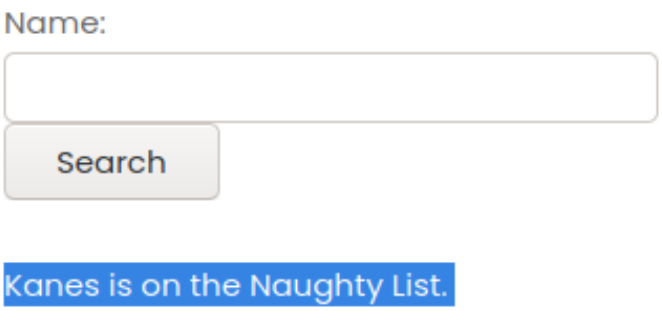
We then proceed to the next name, "**Timothy**", and found it is on the **Naughty List**.



Next, the name "**Tib3rius**" was entered, and we found that it is on the **Nice List**.

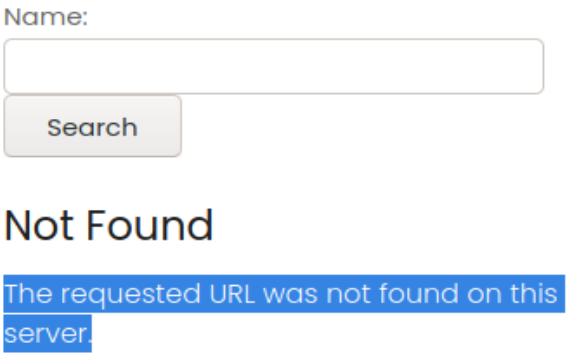


Last, the name "**Kanes**" was entered, and we found that it is on the **Naughty List**.



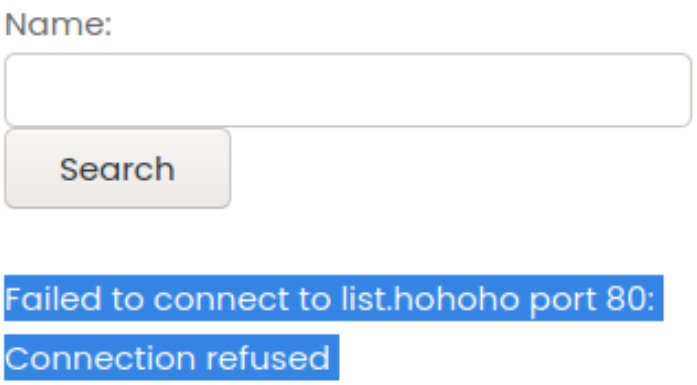
Question 2

To fetch the root of the same site, we apply "/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F" at the end of the web app URL, the page displays the text "**Not Found. The requested URL was not found on this server.** "



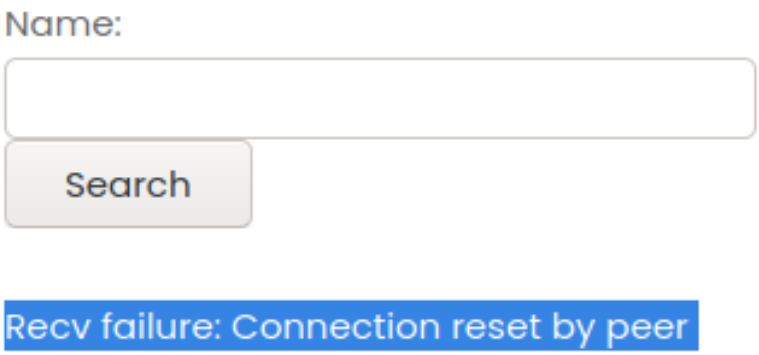
Question 3

Next, we tried changing the port number from 8080 to just 80. By entering "/?proxy=http%3A%2F%2Flist.hohoho%3A80" at the end of the web app URL, the page displayed the text "**Failed to connect to list.hohoho port 80: Connection refused** " which suggests that port 80 is not open on list.hohoho.



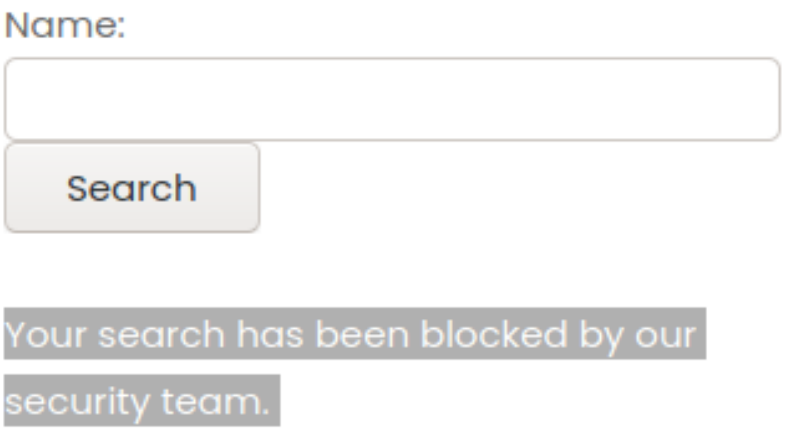
Question 4

We then tried changing the port number to 22 which is the default SSH port. We used "/?proxy=http%3A%2F%2Flist.hohoho%3A22" at the end of the URL, and the message now changed into "**Recv failure: Connection reset by peer**" which suggests that port 22 is open but did not understand what was sent.



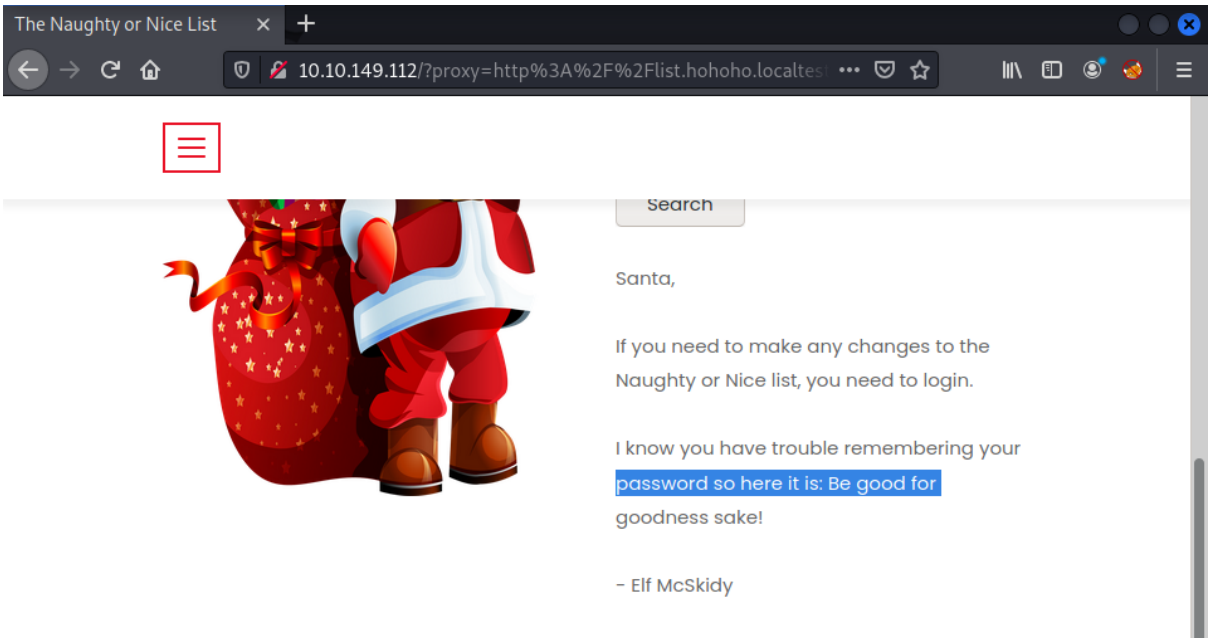
Question 5

We tried another way by replacing the “list.hohoho” hostname with “localhost” by entering "/?proxy=http%3A%2F%2Flocalhost” at the end of the URL. We found that the message returned says “ **Your search has been blocked by our security team. “** By looking at the message, we know that the developer has implemented a check to ensure that the hostname provided starts with “list.hohoho”, and will block any hostnames that don’t.

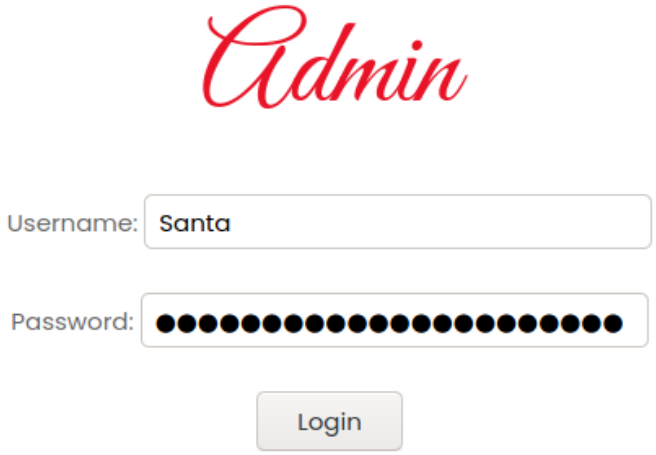


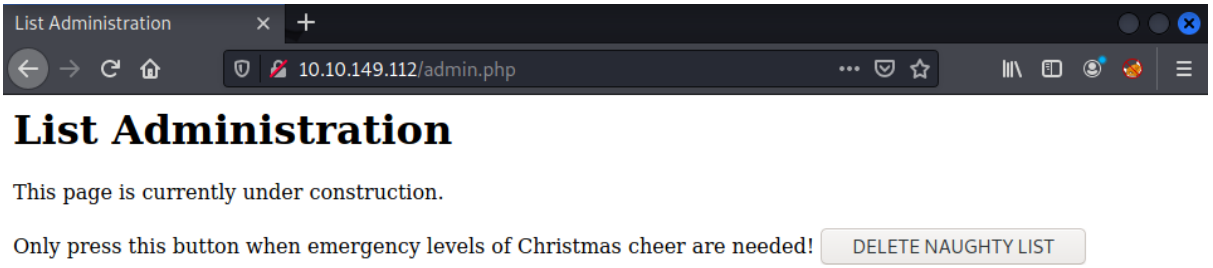
Question 6

We know that the hostname needs to starts with “list.hohoho”, we than set the hostname in the URL to “list.hohoho.localtest,.me”. We used "/?proxy=http%3A%2F%2Flocalhost" at the end of the URL. By using it, we successfully access the web server and found a message from Elf McSkidy that contains the Santa’s password.

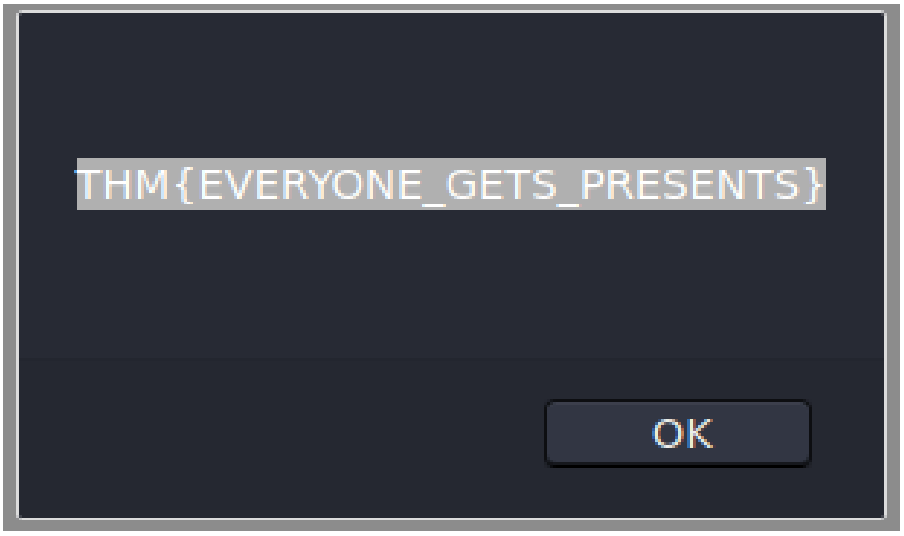


We know that Santa's password is "**Be good for goodness sake!**". We then use the password provided by Elf McSkidy and successfully entered the admin page.





After we entered the admin page, we clicked the "DELETE NAUGHTY LIST" and the challenge flag appeared which is **THM{EVERYONE_GETS_PRESENTS}**.



Thought Process/Methodology:

After we navigated to the web app, we entered the names in the form and click the “Search” button to check whether the name is on the Naughty List or Nice List. Next, we tried to fetch the root of the same site and browse to **http://machine_ip/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F.** The message, “**Not Found. The requested URL was not found on this server.**” has displayed on the web app. We know that we were able to make the server request the modified URL and return the response. We then tried changing the port number from 8080 to just 80. We entered the URL, **http://machine_ip/?proxy=http%3A%2F%2Flist.hohoho%3A80** and the message now change to “**Failed to connect to list.hohoho port 80: Connection refused**” which suggests that port 80 is not open on list.hohoho. We then tried by changing the port number to 22 and entered the URL **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. We tried another way by replacing the list.hohoho hostname with “localhost”. We entered the URL **http://machine_ip/?proxy=http%3A%2F%2Flocalhost**, the message returned says “**Your search has been blocked by our security team.**" By looking at the message, we know that the developer has implemented a check to ensure that the hostname provided starts with “list.hohoho”, and will block any hostnames that don't. By knowing this, we tried and set the hostname in the URL to “list.hohoho.localtest.me”, and we had successfully access the local service by entering the URL **http://machine_ip/?proxy=http%3A%2F%2Flist.hohoho.localtest.me.** The admin password is given in the message written by Elf McSkidy. After knowing the password, we clicked at the “Admin” link at the top and proceeded to login to it. We put the username as “Santa” and the password we found to login as santa. A list administration page appeared after we have login. Lastly, we proceeded to delete the naughty list and the challenge flag appeared.

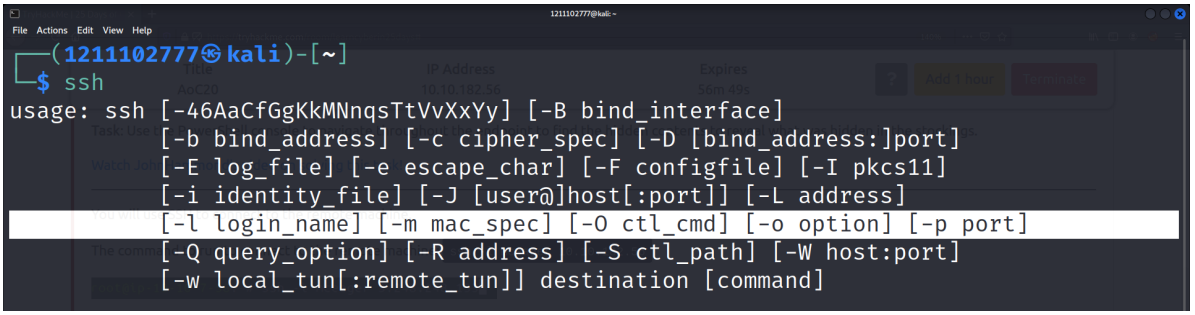
Day 20: Blue Teaming – PowershELIF to the rescue

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

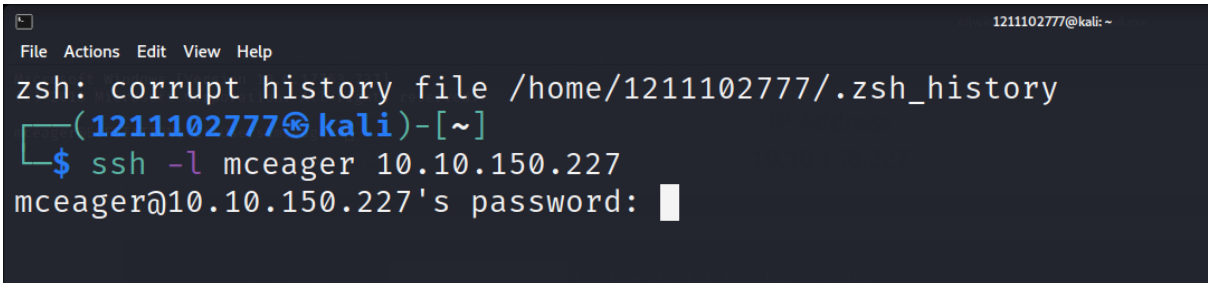
Question 1

We accessed the terminal in Kali Linux to check ssh manual, we know that parameter -l do **login name**

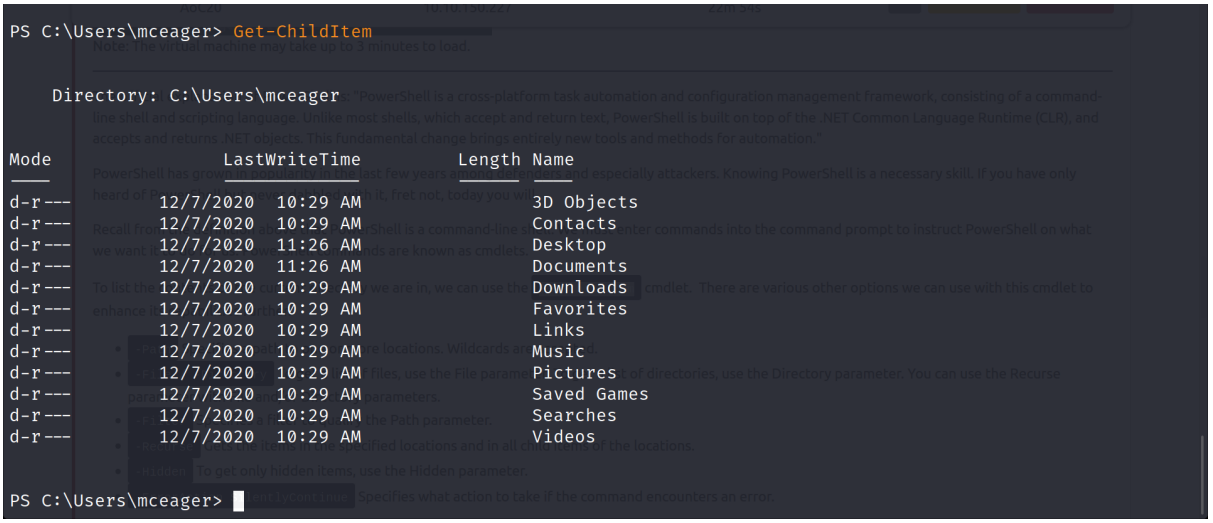


Question 2

We connected to the remote machine by entering ssh -l mceager MACHINE_IP and password r0ckStar!



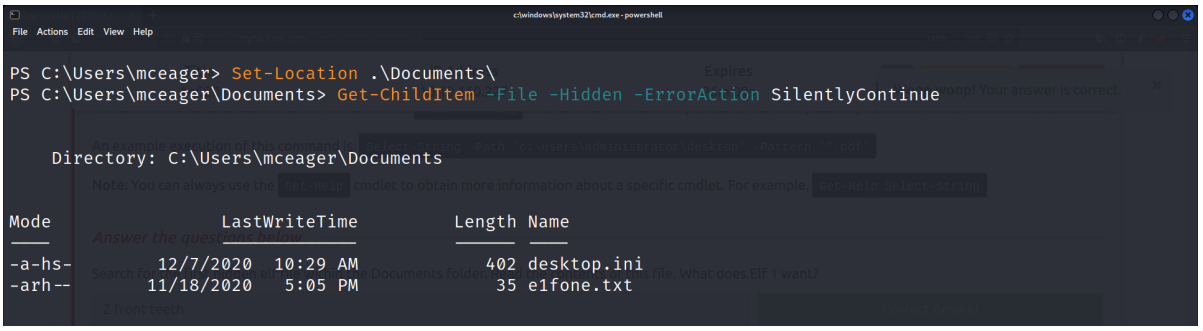
We checked for the current directory we are in by using Get-ChildItem



Then, we managed to get into Documents folder

```
PS C:\Users\mceager> Set-Location .\Documents\
PS C:\Users\mceager\Documents>
```

We proceeded with searching for the hidden content



By reading the content of hidden elf file, we know that Elf 1 want **2 front teeth**

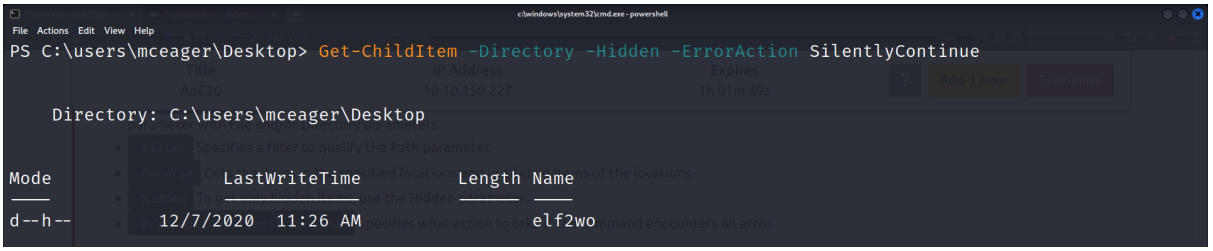
```
PS C:\Users\mceager\Documents> Get-Content -Path e1fone.txt
All I want is my '2 front teeth'!!!
PS C:\Users\mceager\Documents>
```

Question 3

We changed the directory to desktop.

```
PS C:\users\mceager\Documents> Set-Location -Path c:\users\mceager\Desktop
```

We proceeded to search for the hidden directory.



Afterwards, we read the hidden text file and realised **Scrooged** is the name of that movie that Elf 2 wants.

```
PS C:\users\mceager\Desktop> Set-Location .\elf2wo\
PS C:\users\mceager\Desktop\elf2wo> Get-ChildItem


    Directory: C:\users\mceager\Desktop\elf2wo


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/17/2020  10:26 AM             64 e70smsW10Y4k.txt


PS C:\users\mceager\Desktop\elf2wo> Get-Content -Path e70smsW10Y4k.txt
I want the movie Scrooged <3!
PS C:\users\mceager\Desktop\elf2wo>
```

Question 4

Then, we changed the directory to Windows by entering Set-Location -Path C:\Windows

Hidden folder that contains files for Elf 3 has been found – **3lfthr3e**

```
PS C:\users\mceager\desktop\elf2wo> Set-Location -Path C:\Windows
PS C:\Windows> Get-ChildItem -Directory -Hidden -ErrorAction SilentlyContinue -Path C:\Windows\ -Filter '*3*' -
Recurse


    Directory: C:\Windows\System32


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--h--        11/23/2020   3:26 PM                3lfthr3e
```

Question 5

Access the hidden folder by entering Set-Location .\System32\3lfthr3e

Later, we searched for the hidden content by entering Get-ChildItem -Hidden

In order to find the word count of the first file contain, we entered Get-Content -Path 1.txt | Measure-Object -Word

We realised the file contained **9999** words.

```
PS C:\Windows> Set-Location .\System32\3lfthr3e
PS C:\Windows\System32\3lfthr3e> Get-ChildItem -Hidden


    Directory: C:\Windows\System32\3lfthr3e


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-arh--        11/17/2020  10:58 AM          85887 1.txt
-arh--        11/23/2020   3:26 PM       12061168 2.txt


PS C:\Windows\System32\3lfthr3e> Get-Content -Path 1.txt | Measure-Object -Word

Lines Words Characters Property
----- ----- ---------- --------
       9999


PS C:\Windows\System32\3lfthr3e>
```

Question 6

We used (Get-Content -Path file.txt)[index] to get the exact position of a string within a file.

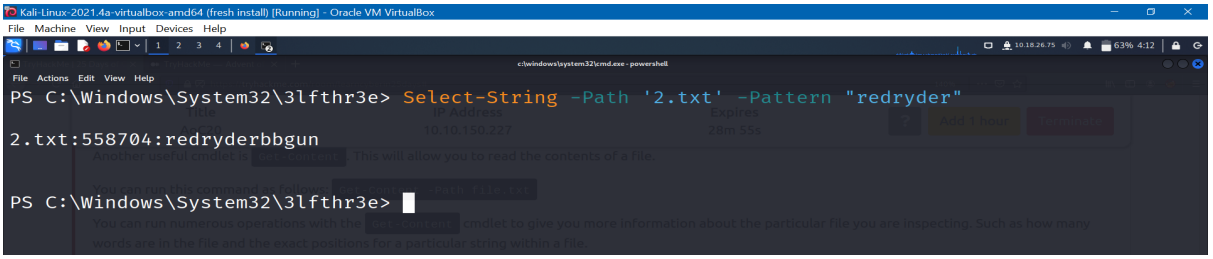
We discovered **Red** & **Ryder** are at index 551 and 6691 respectively

```
PS C:\Windows\System32\3lfthr3e> (Get-Content -Path 1.txt)[551]
Red
PS C:\Windows\System32\3lfthr3e> (Get-Content -Path 1.txt)[6991]
Ryder
PS C:\Windows\System32\3lfthr3e>
```

Question 7

Finally, we searched in the 2nd file which is 2.txt for the phrase from the previous question by entering Select-String -Path '2.txt' -Pattern "redryder"

We eventually found **redryderbbgun**.



Thought Process/Methodology:

We begin with accessing the terminal and search for the ssh manual followed by connecting to the remote machine. Once we entered successfully, we directed to the Documents folder and explored its hidden content to find the items Elf 1 wanted, which is **2 front teeth**. Next, we searched on the Desktop folder and explored its hidden content to get the movie that Elf 2 wants, which is **Scrooged**.

Then, we accessed to Windows directory in order to search for the name of the hidden folder in it, which is **3lfthr3e.** After gaining access to the hidden folder, we managed to get the word count of the first file, which is **9999** words. We then proceeded to get the specific words, which is **Red** & **Ryder** at index 551 and 6991 respectively. Last but not least, we searched in the 2.txt file for the phrase from redryder to find what Elf 3 wants, which is **redryderbbgun.**