

PSP0201

WEEK 5

WRITE-UP

Group: 1K HONDA

Members

ID	Name	Role
1211100415	Muhammad Ummar Hisham bin Ahmad Madzlan	Leader
1211103066	Balqis Afiqah binti Ahmad Fahmi	Member
1211101925	Nur Alya Nabilah binti Md. Naser	Member

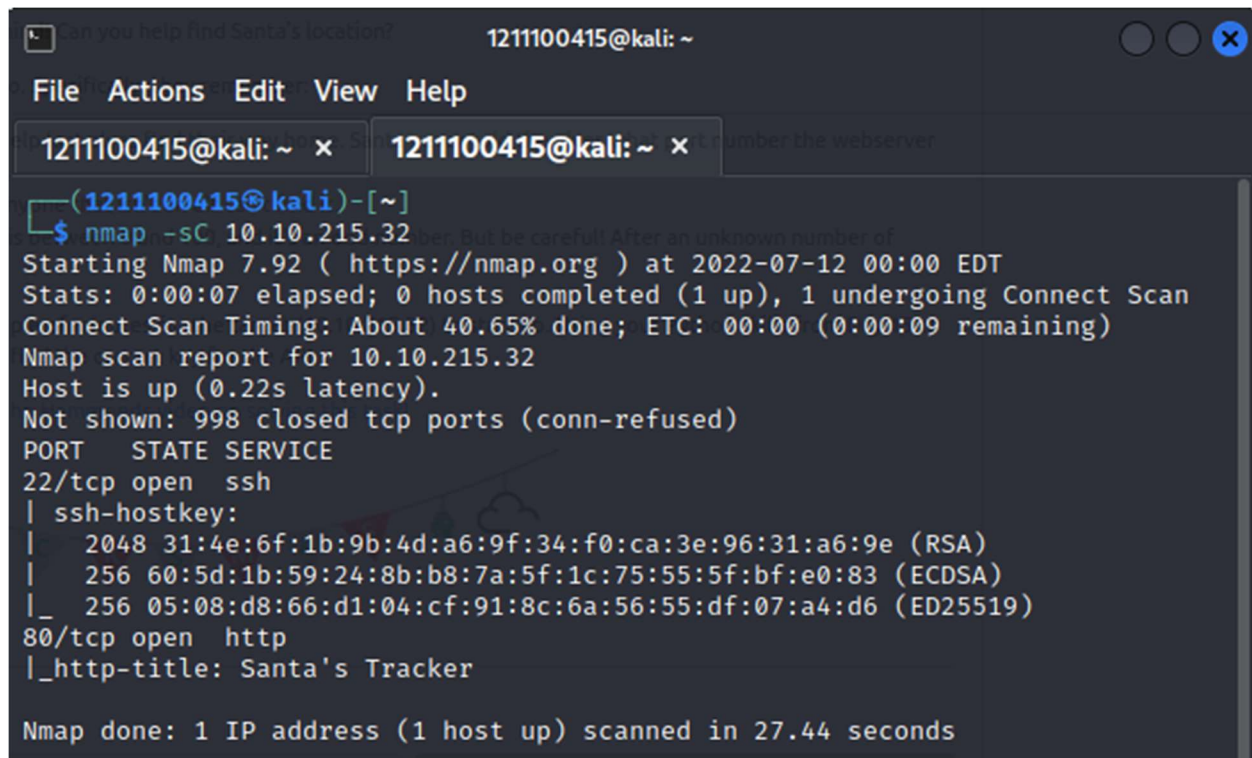
Day 16: Scripting – Help! Where Is Santa?

Tools: Kali Linux, Terminal, Nmap, Python, Sublime Text, Mozilla FireFox

Solution:

Question 1:

Scan the IP address using Nmap.



```
Can you help find Santa's location? 1211100415@kali: ~
File Actions Edit View Help
1211100415@kali: ~ x 1211100415@kali: ~ x number the webserver
(1211100415@kali)-[~]
$ nmap -sC 10.10.215.32
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-12 00:00 EDT
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 40.65% done; ETC: 00:00 (0:00:09 remaining)
Nmap scan report for 10.10.215.32
Host is up (0.22s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
| ssh-hostkey:
|   2048 31:4e:6f:1b:9b:4d:a6:9f:34:f0:ca:3e:96:31:a6:9e (RSA)
|   256 60:5d:1b:59:24:8b:b8:7a:5f:1c:75:55:5f:bf:e0:83 (ECDSA)
|_  256 05:08:d8:66:d1:04:cf:91:8c:6a:56:55:df:07:a4:d6 (ED25519)
80/tcp    open  http
|_ http-title: Santa's Tracker

Nmap done: 1 IP address (1 host up) scanned in 27.44 seconds
```

Question 2:

Head to the IP address. The template is on the upper left corner of the webpage.



Question 3:

Create a Python file and import the “BeautifulSoup” library.

```
linkgrabber.py
1  # Import the libraries we downloaded earlier
2  # if you try importing without installing them, this step will fail
3  from bs4 import BeautifulSoup
4  import requests
5
6  # replace testurl.com with the url you want to use.
7  # requests.get downloads the webpage and stores it as a variable
8  html = requests.get('http://10.10.215.32/')
9
10 # this parses the webpage into something that beautifulsoup can read over
11 soup = BeautifulSoup(html.text, "lxml")
12 # lxml is just the parser for reading the html
13
14 # this is the line that grabs all the links # stores all the links in the links v
15 links = soup.find_all('a')
16 for link in links:
17     # prints each link
18     if "href" in link.attrs:
19         print(link["href"])
```

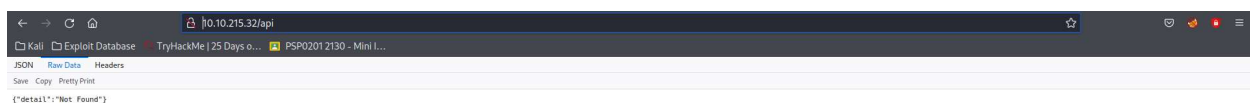
Use the Python file to look for the API directory.

```
1211100415@kali: ~
File Actions Edit View Help
1211100415@kali: ~ x 1211100415@kali: ~ x 1211100415@kali: ~ x

(1211100415@kali)-[~]
$ python3 linkgrabber.py | uniq
../
https://github.com/BulmaTemplates/bulma-templates/blob/master/templates/hero.
https://tryhackme.com
#
http://machine_ip/api/api_key
#
https://github.com/BulmaTemplates/bulma-templates
```

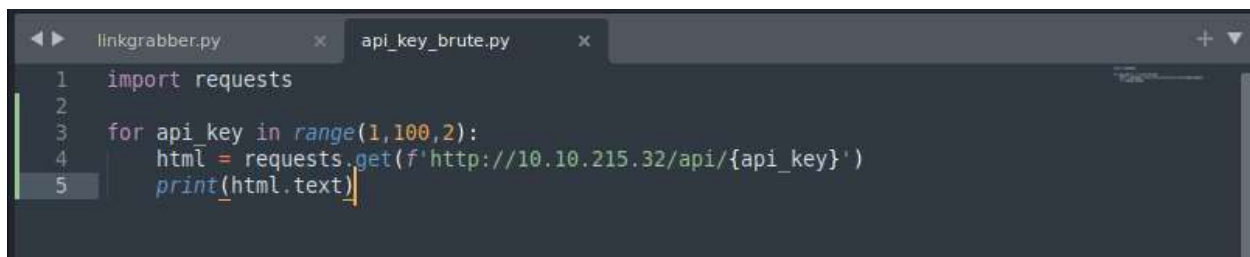
Question 4:

Go to the API endpoint without setting any parameters.



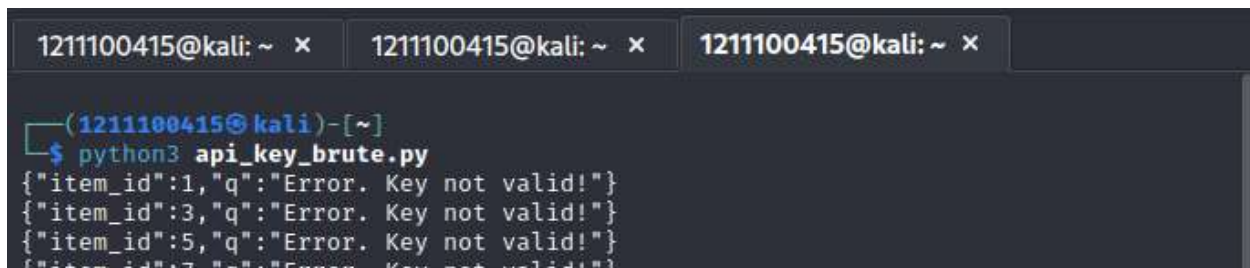
Question 5 & 6:

Create a Python file and import the “requests” library.



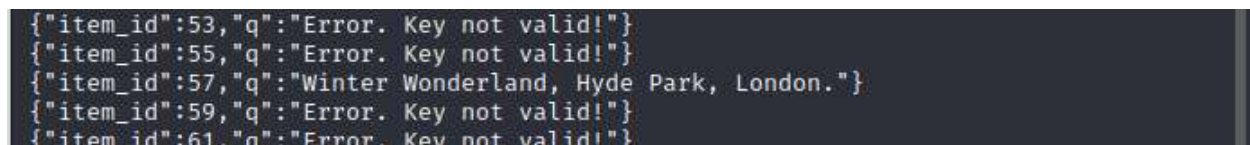
```
1 import requests
2
3 for api_key in range(1,100,2):
4     html = requests.get(f'http://10.10.215.32/api/{api_key}')
5     print(html.text)
```

Use the Python file on command prompt.



```
(1211100415@kali)-[~]
$ python3 api_key_brute.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":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!"}
```

Search for the valid API key.



```
{"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!"}
```

Thought Process/Methodology:

Once we had gained the IP address of the targeted machine, we opened the terminal and scan the IP address using Nmap. We were given 2 ports for the SSH and HTTP service. We learned that the web server was running on port number 80 and navigated to the webpage where we were brought to webpage named BULMA. Then, we created a Python file using Sublime Text and import the “BeautifulSoup” library and named the file as “linkgrabber.py”. Getting back on terminal, we used the file to look for the API directory. Once we had found the API directory, we navigated to the API endpoint without setting any parameters. Lastly, we created another Python file, importing the “requests” library and named it api_key_brute.py. We used the file on the terminal and searched for the valid API key.

Day 17: Reverse Engineering – ReverseELFneering

Tools: Kali Linux, Nmap, Terminal, Radare2

Solutions:

Question 1:

Read 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 Radare2 cheatsheet.

 Analyse all `aa[a[a]]`

Question 3:

Read the Radare2 cheatsheet.

 Set breakpoint `db [addr]`

Question 4:

Read the Radare2 cheatsheet.

 Continue execution `dc`

Question 5, 6 & 7:

Scan the IP address using Nmap.

```
(1211100415@kali)-[~]
$ nmap -sC 10.10.70.122
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-13 00:30 EDT
Stats: 0:00:02 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 7.55% done; ETC: 00:30 (0:00:24 remaining)
Nmap scan report for 10.10.70.122
Host is up (0.21s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
| ssh-hostkey:
|   2048 c2:85:b0:50:68:f8:5a:9c:24:45:4e:5a:df:1d:56:79 (RSA)
|   256 02:22:b6:12:86:7a:9b:23:a8:75:8e:6a:d4:46:7b:bc (ECDSA)
|_  256 70:d1:03:f3:78:c9:89:b3:9b:be:17:7c:91:ad:c7:d4 (ED25519)

Nmap done: 1 IP address (1 host up) scanned in 30.57 seconds
```

Get connected to the webserver using SSH.

```
(1211100415@kali)-[~]
$ ssh elfmceager@10.10.70.122
elfmceager@10.10.70.122's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-128-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Jul 13 04:33:15 UTC 2022

System load:  0.0               Processes:    92
Usage of /:   39.4% of 11.75GB   Users logged in: 0
Memory usage: 9%               IP address for ens5: 10.10.70.122
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

Last login: Wed Jul 13 04:32:51 2022 from 10.8.92.127
elfmceager@tbfc-day-17:~$
```

Open the “challenge1” using Radare2 to enter binary debugging mode.

```
elfmceager@tbfc-day-17:~$ ls
challenge1  file1
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1850 started...
= attach 1850 1850
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
asm.bits 64
```


Analyse the program and examine the assembly code. Set a breakpoint and run the program until it hit the breakpoint. View the content of the memory address repeat the same steps.

```
[0x00400a30]> pdf @main
;-- main:
/ (fcn) sym.main 35
sym.main ();
; var int local_ch @ rbp-0xc
; var int local_8h @ rbp-0x8
; var int local_4h @ rbp-0x4
; DATA XREF from 0x00400a4d (entry0)
0x00400b4d 55 push rbp
0x00400b4e 4889e5 mov rbp, rsp
0x00400b51 c745f4010000. mov dword [local_ch], 1
0x00400b58 c745f8060000. mov dword [local_8h], 6
0x00400b5f 8b45f4 mov eax, dword [local_ch]
0x00400b62 0faf45f8 imul eax, dword [local_8h]
0x00400b66 8945fc mov dword [local_4h], eax
0x00400b69 b800000000 mov eax, 0
0x00400b6e 5d pop rbp
0x00400b6f c3 ret
```

Thought Process/Methodology:

First, we scanned the target machine IP address to learn what service the webserver was running on. Then, we accessed the webserver using SSH and entered the credential given in TryHackMe. Afterwards, used Radare2 to open the binary debugging mode. Inside the bugging mode, use the command “aa” to analyse the r2 program. Once the analysis had completed, we searched for an entry point using the command “afl”. Once we had found the main function, we examined the assembly code by running the command “pdf@main”. Then, we could set a breakpoint using the command “db”. To run the program until it hit a breakpoint, we used the command “dc”. Then, we viewed the contents of the variable using the command “px @rbx-0xc”. Finally, we repeated the same process by using the command “ds” to execute and move on to the next binary values.

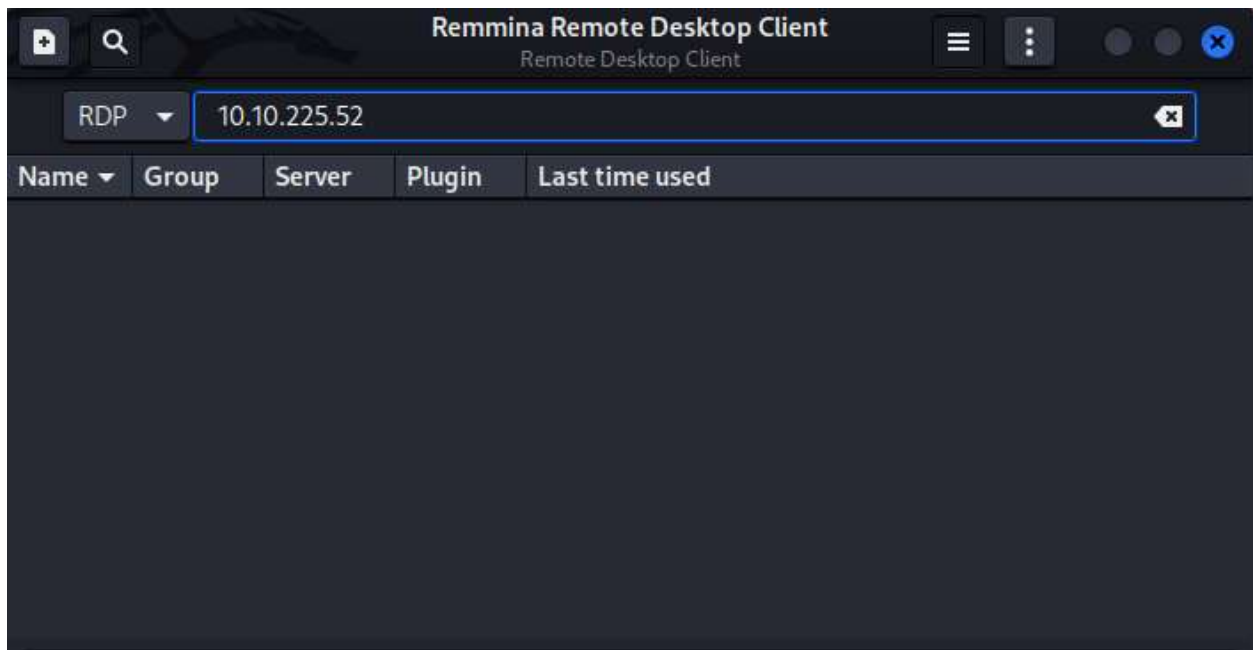
Day 18: Reverse Engineering – The Bits of Christmas

Tools: Kali Linux, Remmina RDP, IL Spy, Mozilla FireFox, CyberChef

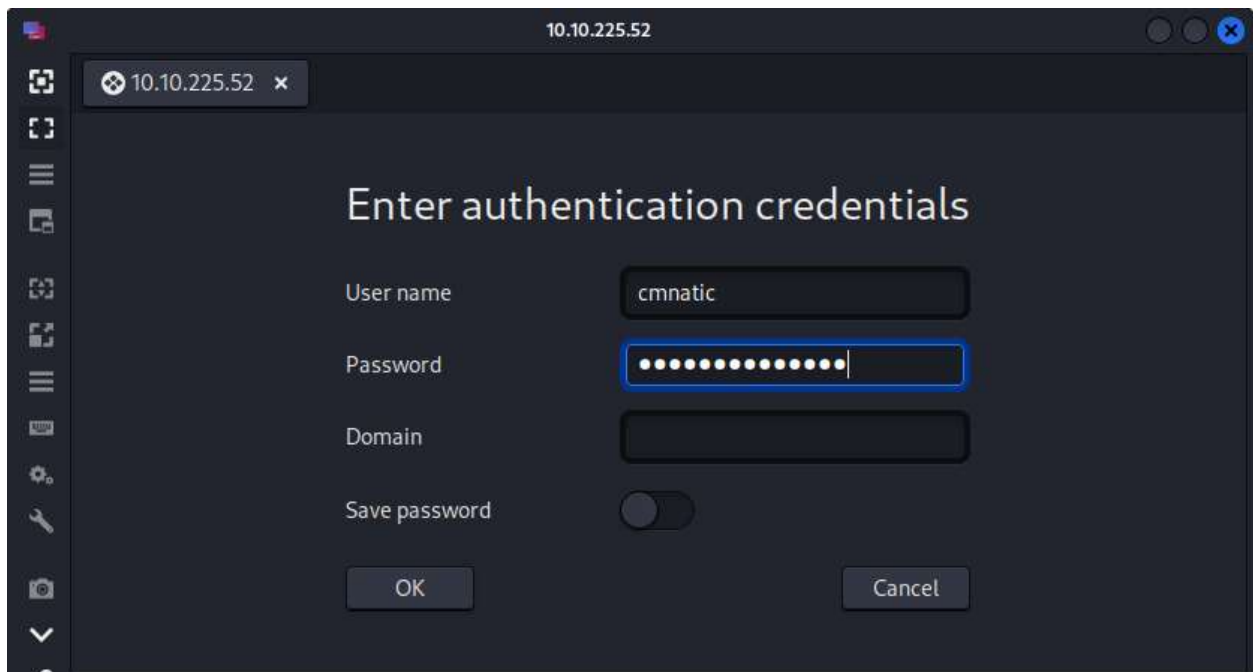
Solutions:

Question 1:

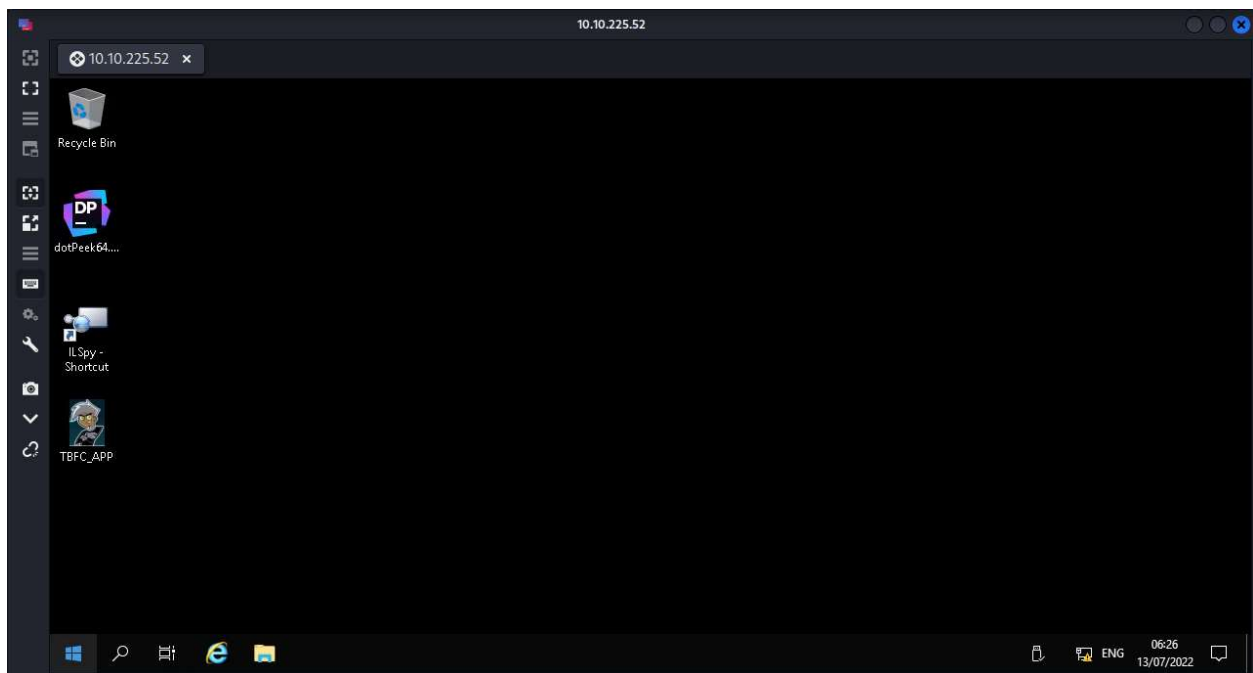
Enter the target machine's IP address into Remmina RDP and run it.



Using the credentials in TryHackMe, enter the username and password.



Open the TBF_APP.



Enter the password.



If the password is incorrect, we will receive this message.



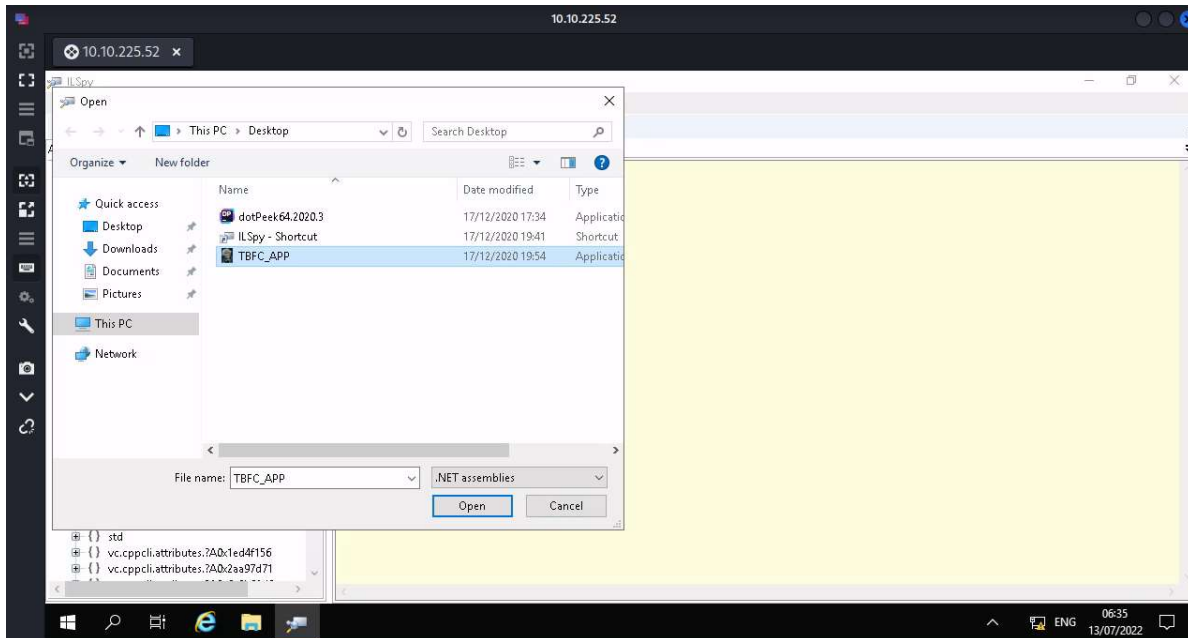
Question 2:

Look at the bottom left corner of the TBFC_APP.

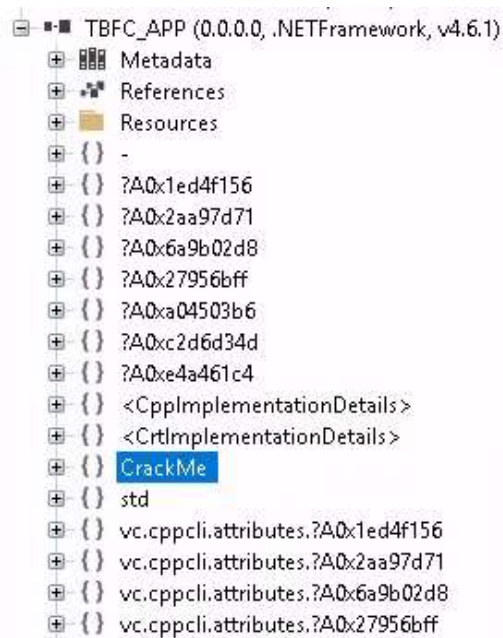
The Best Festival
Company 2020

Question 3:

Start the IL Spy and open the TBFC_APP.



Decompile the TBFC_APP and notice the CrackMe module.



Question 4:

Search for the password in the MainForm.

```
// CrackMe.MainForm
using ...

public class MainForm : Form
{
    private Label labelKey;
    private TextBox textBoxKey;
    private Panel panelLogo;
    private TableLayoutPanel tableLayoutPanel1;
    private Button buttonActivate;
    private TableLayoutPanel tableLayoutPanelButtons;
    private Label labelOrg;
    private Container components;

    public MainForm()
    {
        try
        {
            InitializeComponent();
        }
        catch
        {
            //try-fault
            base.Dispose(disposing: true);
            throw;
        }
    }

    private void ~MainForm()
    {
    }
}
```

Question 5:

As we understand that once we entered the password, we need to click the “Submit” button and there will be an event. So, look for the `buttonActivate_Click`.

```
private unsafe void buttonActive_Click(object sender, EventArgs e)
{
    IntPtr value = Marshal.StringToHGlobalAnsi(textBoxKey.Text);
    sbyte* ptr = (sbyte*)System.Runtime.CompilerServices.Unsafe.AsPointer(ref &Module.??C_00B8E1KKDFEP@santapassword321@);
    void* ptr2 = (void*)value;
    byte b = *(byte*)ptr2;
    byte b2 = 115;
    if ((uint)b >= 115u)
    {
        while ((uint)b <= (uint)b2)
        {
            if (b != 0)
            {
                ptr2 = (byte*)ptr2 + 1;
                ptr++;
                b = *(byte*)ptr2;
                b2 = (byte)(*ptr);
                if ((uint)b < (uint)b2)
                {
                    break;
                }
                continue;
            }
            MessageBox.Show("Welcome, Santa, here's your flag thm{046af}", "That's the right key!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
            return;
        }
        MessageBox.Show("Uh Oh! That's the wrong key", "You're not Santa!", MessageBoxButtons.OK, MessageBoxIcon.Hand);
    }
}
```

Question 6:

Double click on the array under the buttonActivate_Click method.

```
private unsafe void buttonActivate_Click(object sender, EventArgs e)
{
    IntPtr value = Marshal.StringToGlobalAnsi(textBoxKey.Text);
    sbyte* ptr = (sbyte*)System.Runtime.CompilerServices.Unsafe.AsPointer(ref <Module>._??_C@_0BA@IKXDFEPG@satapassword321@);
    void* ptr2 = (void*)value;
    byte b = *(byte*)ptr2;
    byte b2 = 115;
    if ((int)b >= 115u)
    {
        internal static <CpImplementationDetails>.$ArrayType$40V0B0E$5CBD global::<Module>._??_C@_0BA@IKXDFEPG@satapassword321@
    }
}
```

Copy the value of the data.

```
using ...

internal static $ArrayType$$B0Y0B0$$CBD ??_@_0BB@IKKDFEPG@santapassword321@/* Not supported: data(73 61 6E 74 61 70 61 73 77 6F 72 64 33 32 31 00) */;
```

Paste the copied data into CyberChef.

The image shows the CyberChef web application. On the left, the 'Recipe' panel is set to 'From Hex' with a 'Delimiter' of 'Auto'. The 'Input' panel contains a single line of hex data: 73 61 6E 74 61 70 61 73 73 77 6F 72 64 33 32 31. The 'Output' panel shows the result of the conversion: 'santapassword321'. Metadata for the input shows a length of 47 and 1 line. Metadata for the output shows a start/end of 10, a length of 16, and 1 line.

Question 7:

Enter the password into the TBFC_APP.

The image shows a web browser window titled 'TBFC Dashboard'. The main heading is 'TBFC' in large white letters on a dark blue background. Below this, there is a 'Password:' label and a text input field containing 'santapassword321'. At the bottom left, it says 'The Best Festival Company 2020'. To the right of this text is a blue 'Submit' button.

Receive the flag.

The image shows a small dialog box with the title 'That's the right key!'. It contains an information icon (a blue circle with a white 'i') followed by the text 'Welcome, Santa, here's your flag thm{046af}'. At the bottom right of the dialog is an 'OK' button.

Thought Process/Methodology:

Firstly, we entered the targeted IP address into Remmina RDP and ran it. We were brought to the authentication screen, and we entered the credentials that were given in TryHackMe into the username and password box. Once we had gained access, we opened the TBFC_APP and entered the wrong password. We were given a message declaring that we had the wrong password. Then, we started IL Spy and decompiled the TBFC_APP. Then, we searched for the password inside the MainForm under the CrackMe module. As we understand that once we entered the password, we needed to click the "Submit" button and there would be an event. So, we searched for the buttonActivate_Click. Afterwards, we double clicked on the array under the buttonActivate_Click method. We knew from the value of the data that it was in hexadecimal. We copied the data and pasted it in CyberChef. Lastly, we entered the password in the TBFC_APP to receive the flag.

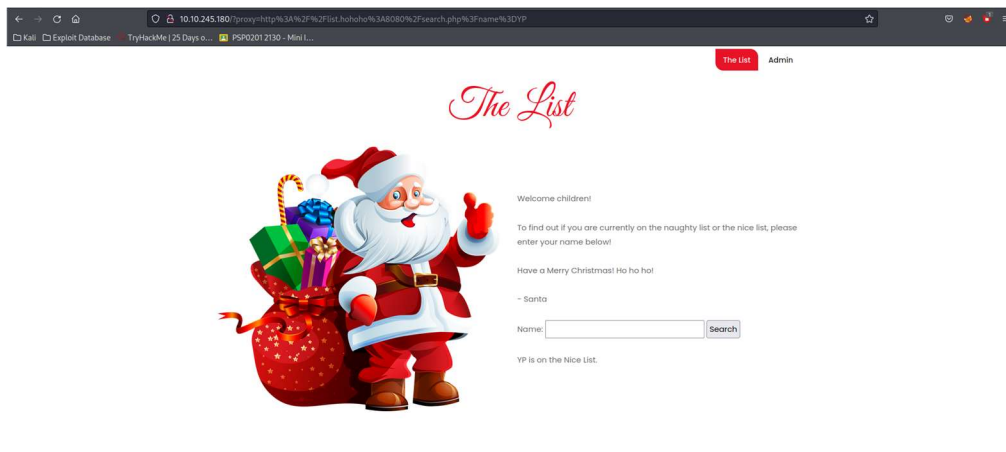
Day 19: Web Exploitation – The Naughty or Nice List

Tools: Kali Linux, Firefox

Solutions:

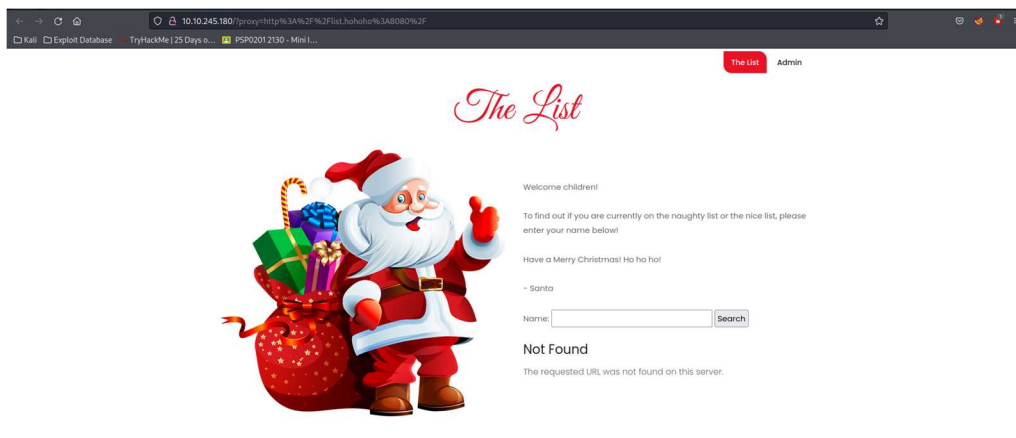
Question 1:

Go to targeted machine IP address and enter the name from the list into the search box.



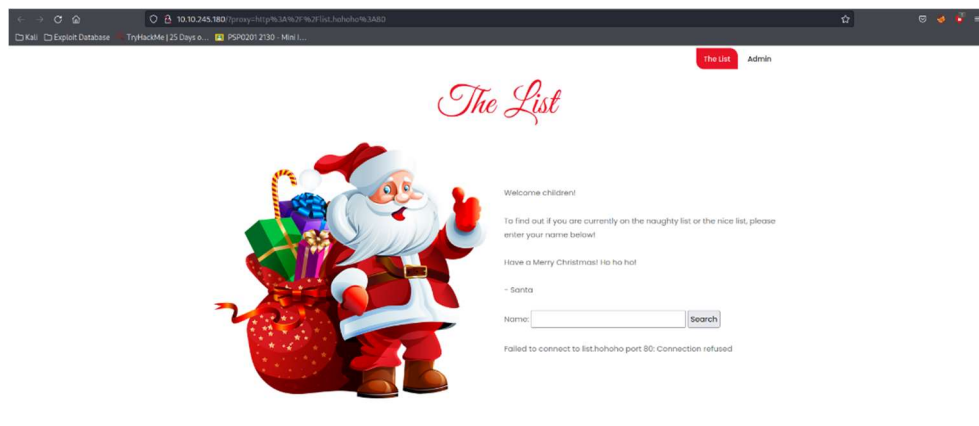
Question 2:

Browse to: `http://10.10.245.180/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F` .



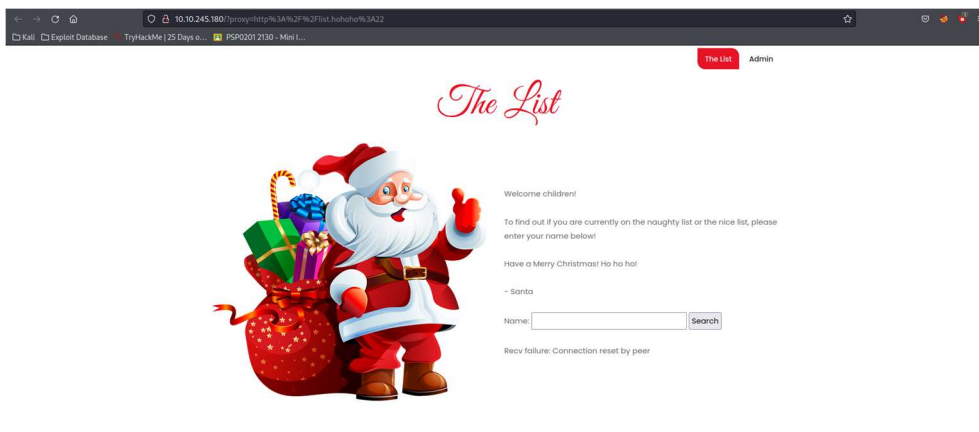
Question 3:

Change the port number from 8080 to 80.



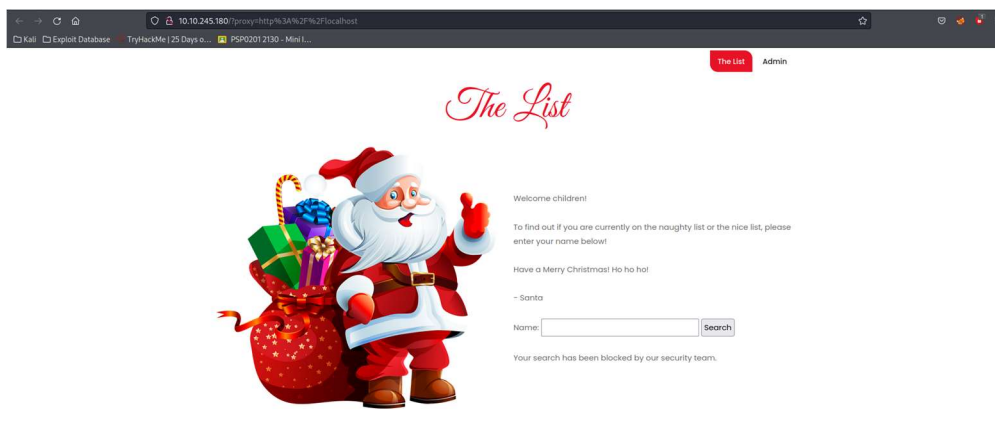
Question 4:

Change the port number from 8080 to 22.



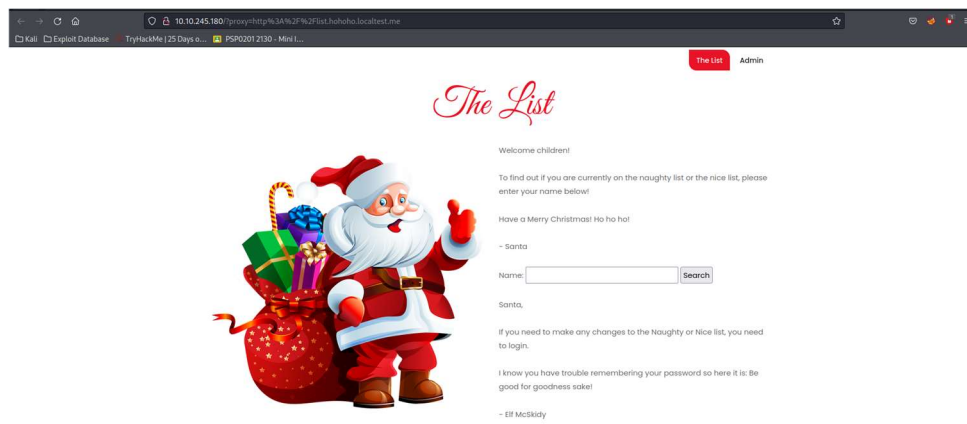
Question 5:

Replace the "list.hohoho" hostname with "localhost".



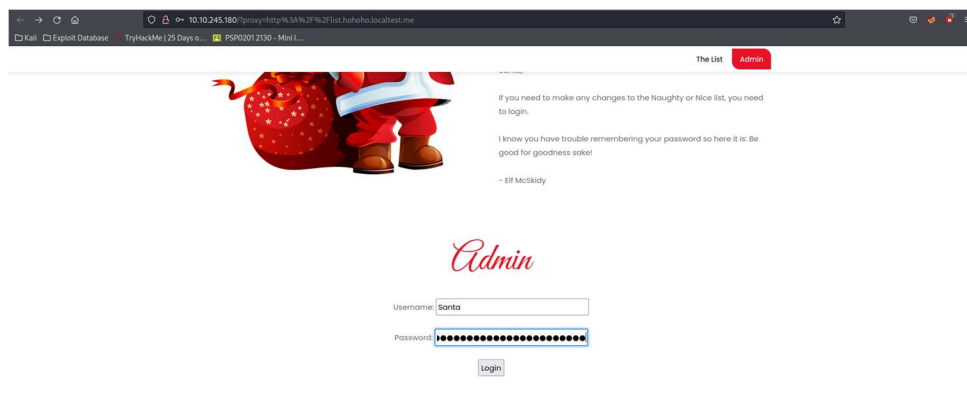
Question 6:

Set the hostname in the URL to "list.hohoho.localtest.me".

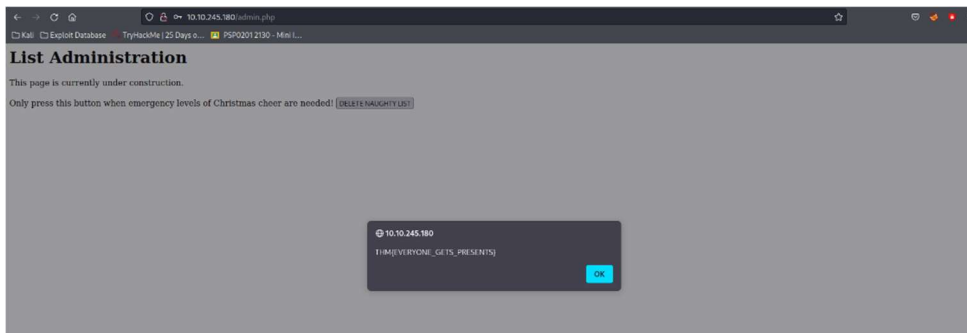


Question 7:

Enter the username and password.



Delete the naughty list to receive the flag.



Thought Process/Methodology:

First, navigate to the Machine IP Address (10.10.222.155). We encounter the list form along with an admin login below it. I put Tib3rius name in the search button to check whether that name is on the Naughty or Nice list. Then we use URL decoder on the value of parameter until we get : <http://list.hohoho:8080/search.php?name=Tib3rius>. Since "list.hohoho" is not a valid hostname on the internet, this likely refers to some back-end machine. Then we try to fetch the root of the same site browse to [<http://10.10.222.155/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F>]. The message we see "Not Found. The requested URL was not found on this server." This seems like we are looking for in a failure site (the root side). Then we move on and use another way, by changing the port number from 8080 to just 80 (the default HTTP port). The messages changes saying that the connection is refused, which suggest that port 80 is not open on list.hohoho. Then we try to the port number 22, the messages now change which the connection reset by peer. This suggests that port 22 is open but did not understand what was sent. Another thing we try to do with SSRF is access running locally in the server by replacing the list.hohoho with "127.0.0.1". We see that we get blocked (by their security team). Since the hostname simply needs to start with "list.hohoho", we can take advantage of DNS subdomains and set the hostname in the URL to "list.hohoho.localtest.me". There it is we succeed; it appears there is web server running locally. We get the password leaves by Elf McSkidy. Now we get to enter the username and password given to get to Santa list administration. Once we delete the naughty list , we received the flag for it.

Day 20: Blue Teaming - Powershell to the rescue

Tools: Kali Linux, PowerShell, SSH

Solutions:

Question 1:

Read the SSH manual.

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

Question 2:

Use SSH to connect to the machine and use the credentials given in TryHackMe.

```
(1211100415@kali)-[/home/1211100415]
PS> ssh -l mceager 10.10.196.60
mceager@10.10.196.60's password: [Day 20] Blue Teaming - Powershell to the rescue
```

Launch PowerShell and navigate to Documents folder.

```
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

mceager@ELFSTATION1 C:\Users\mceager>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

List the hidden contents inside the directory and access the hidden file.

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

Directory: C:\Users\mceager\Documents
hidden folder that contains the file for Elf 2 * 200ms

Mode is the name of LastWriteTime Elf 2 Length Name
----
d--hsl 12/7/2020 10:28 AM My Music
d--hsl 12/7/2020 10:28 AM My Pictures
d--hsl 12/7/2020 10:28 AM My Videos
-a-hs- 12/7/2020 10:29 AM 402 desktop.ini
-arh-- 11/18/2020 5:05 PM 35 elfone.txt

PS C:\Users\mceager\Documents> cat elfone.txt
All I want is my '2 front teeth'!!!
```

Question 3:

Navigate to Desktop and look for the hidden folder.

```
PS C:\Users\mceager\Documents> cd..
PS C:\Users\mceager> cd Desktop
PS C:\Users\mceager\Desktop> Get-ChildItem
PS C:\Users\mceager\Desktop> Get-ChildItem -Hidden
```

Directory: C:\Users\mceager\Desktop

Mode	LastWriteTime	Length	Name
d--h--	12/7/2020 11:26 AM		elf2wo
-a-hs-	12/7/2020 10:29 AM	282	desktop.ini

Access the hidden folder and list the files in the folder. Access the file.

```
PS C:\Users\mceager\Desktop> cd 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> cat e70smsW10Y4k.txt
I want the movie Scrooged <3!
```

Question 4:

Navigate to C:\Windows\System32 and list the hidden files.

```
PS C:\Windows\System32> Get-ChildItem -Hidden
```

Directory: C:\Windows\System32

Mode	LastWriteTime	Length	Name
d--h--	11/23/2020 3:26 PM		3lfthr3e
d--h--	11/23/2020 2:26 PM		GroupPolicy

Question 5:

Access the hidden folder and list the hidden files.

```
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

Count the words inside the first file.

```
PS C:\Windows\System32\3lfthr3e> Get-Content 1.txt | Measure-Object -Word
Lines Words Characters Property
-----
9999
```

Question 6:

Find the string on index 551 and 6991.

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

Question 7:

Open the second file and type in the patterned string from the first file.

```
PS C:\Windows\System32\3lfthr3e> Get-Content 2.txt | Select-String "redryder"
redryderbbgun PowerShell over SSH
```

Thought Process/Methodology:

Firstly, we opened PowerShell and used SSH to connect to the machine and use the credentials given in TryHackMe. Then, we launched the PowerShell and navigated to the Documents folder. Then we listed the hidden contents inside the directory using the command `Get-Command` and opened a hidden file in a document that has elf 1. We found out what elf 1 wanted after opening the hidden file. Afterwards, we wanted to know what elf 2 wanted by navigating to the desktop and listing out the hidden files. We noticed the hidden folder and we accessed it to list the files in the folder. We accessed the file and learned elf 2's wish. Then, we navigated to `C:\Windows\System32` and list the hidden files. We opened the `3lfthr3e` and listed the hidden files. We were provided by two files, and we opened the first file and count the words in it using the command `Get-Content 1.txt | Measure-Object -Word`. Afterwards, we searched for the string on index 551 and 6991 using the command `(Get-Content 1.txt)[551,6991]`. Lastly, we opened the second file and typed in the patterned string from the first file using the command `Get-Content 2.txt | Select-String "redryder"`.