PSP0201

Week 5

Writeup

Group Name: 404 Not Found

Members

ID	Name	Role
1211102687	Emily Phang Ru Ying	Leader
1211102975	Loi Xinyi	Member
1211102751	Teo Yu Jie	Member
1211102753	Lim Cai Qing	Member

Day 16 [Scripting] – Help! Where is Santa?

Tools used: Kali Linux, Firefox, nmap, python 3

Solution/walkthrough:

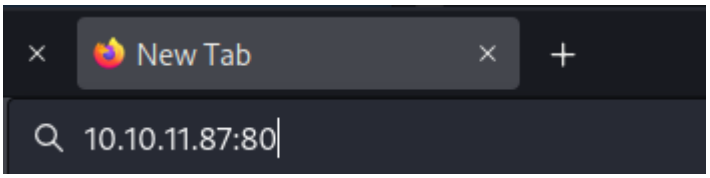
Question 1

We found two open ports after using nmap to scan the IP address. Port 80 is the answer as it is used to host an HTTP service.

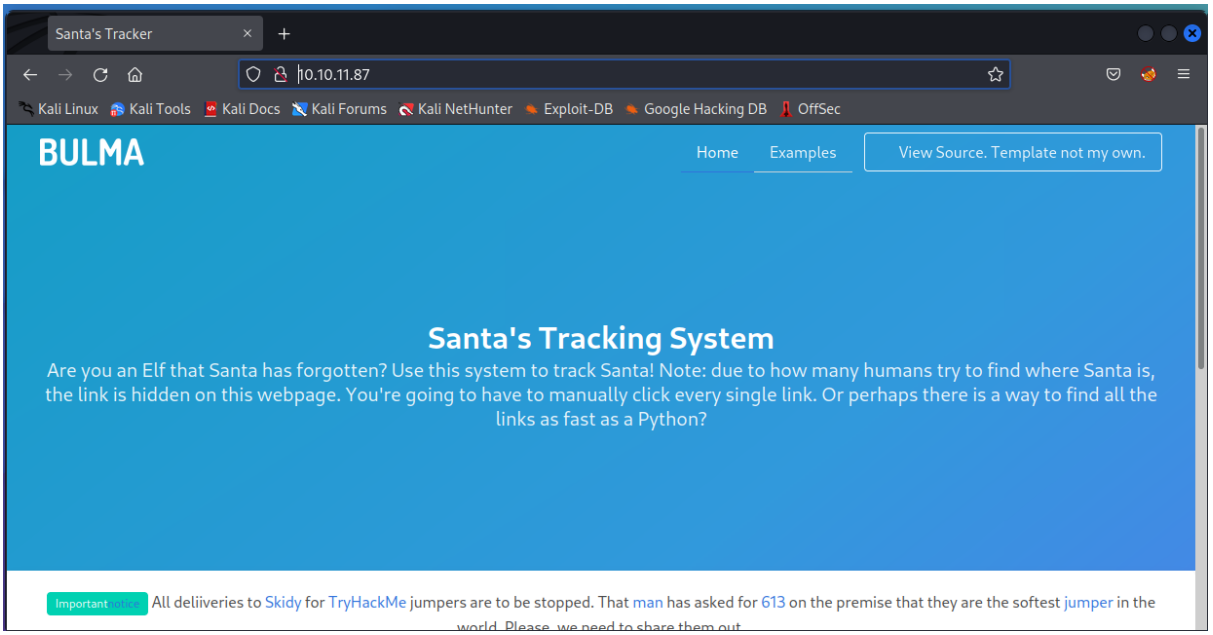
```
┌──(1211102687㉿kali)-[~]
└─$ nmap -v 10.10.11.87
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-14 08:48 EDT
Initiating Ping Scan at 08:48
Scanning 10.10.11.87 [2 ports]
Completed Ping Scan at 08:49, 0.19s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 08:49
Completed Parallel DNS resolution of 1 host. at 08:49, 0.08s elapsed
Initiating Connect Scan at 08:49
Scanning 10.10.11.87 [1000 ports]
Discovered open port 80/tcp on 10.10.11.87
Discovered open port 22/tcp on 10.10.11.87
Increasing send delay for 10.10.11.87 from 0 to 5 due to 73 out of 241 dropped probes since l
ase.
Completed Connect Scan at 08:49, 14.67s elapsed (1000 total ports)
Nmap scan report for 10.10.11.87
Host is up (0.21s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT     STATE    SERVICE
22/tcp   open     ssh
80/tcp   open     http
5357/tcp filtered wsdapi

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 14.98 seconds
```

Question 2

We entered the IP address with the port number on firefox's search bar.



We successfully entered Santa's Tracking System. The template used is BULMA which is shown on the top left of the page.



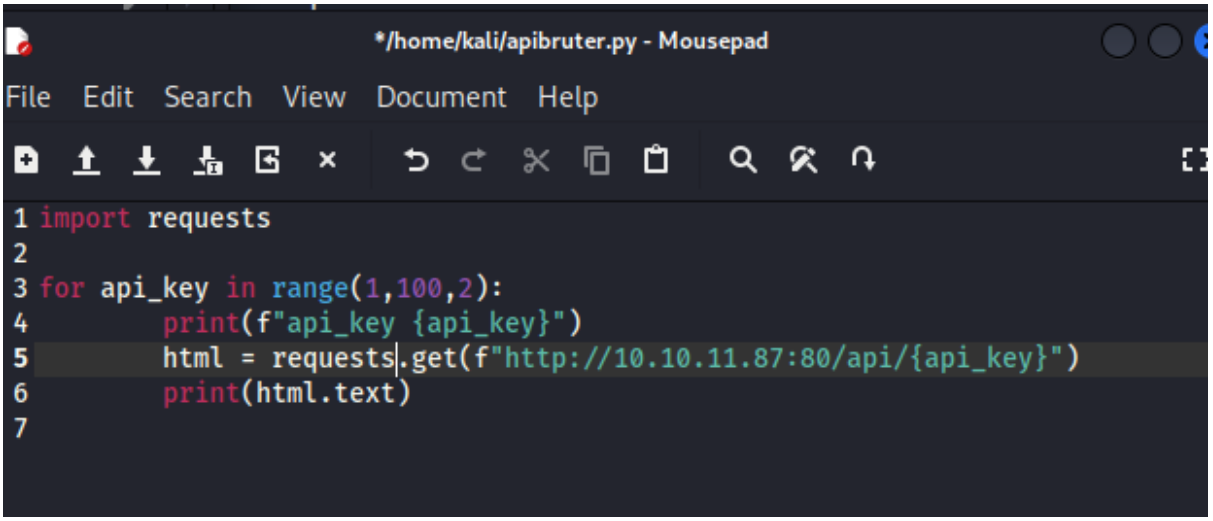
Question 3

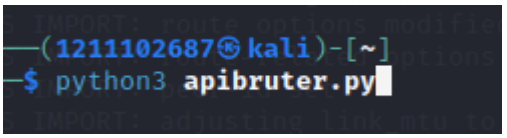
We view the page source of the page and found the directory for the API.

```
            <li><a href="#">Flimsy Lavenrock</a></li>
            <li><a href="#">Maven Mousie Lavender</a></li>
        </ul>
    </div>
    <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>
```

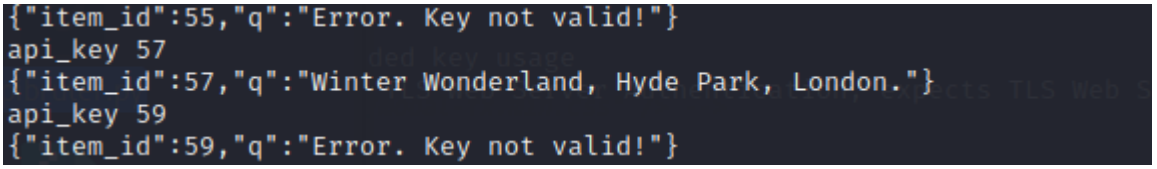
Question 4

We created a python script to loop through the possible odd number API key and run it in the terminal.



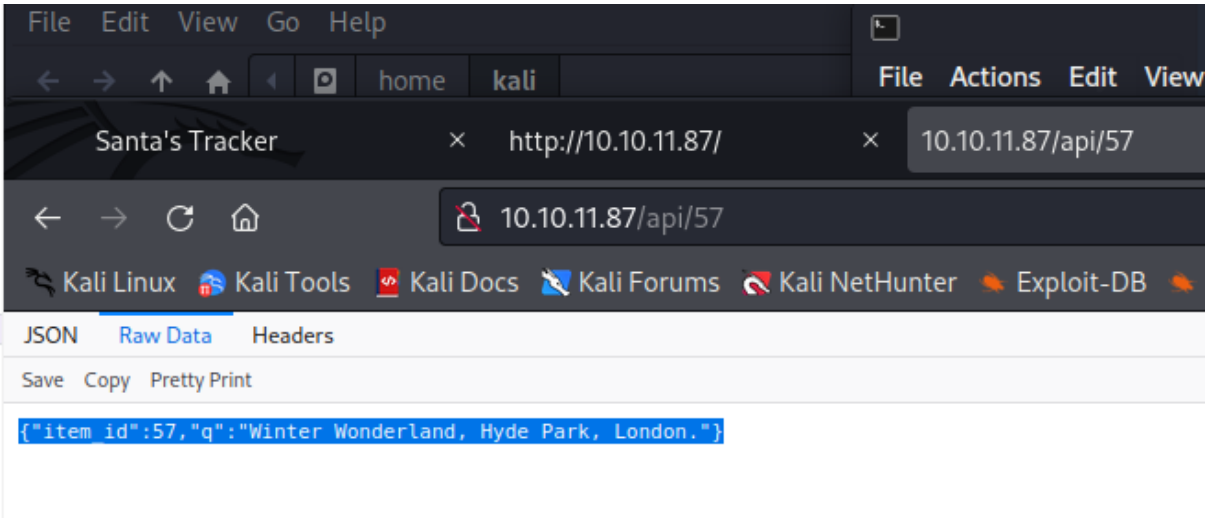


We got a response from API key 57.



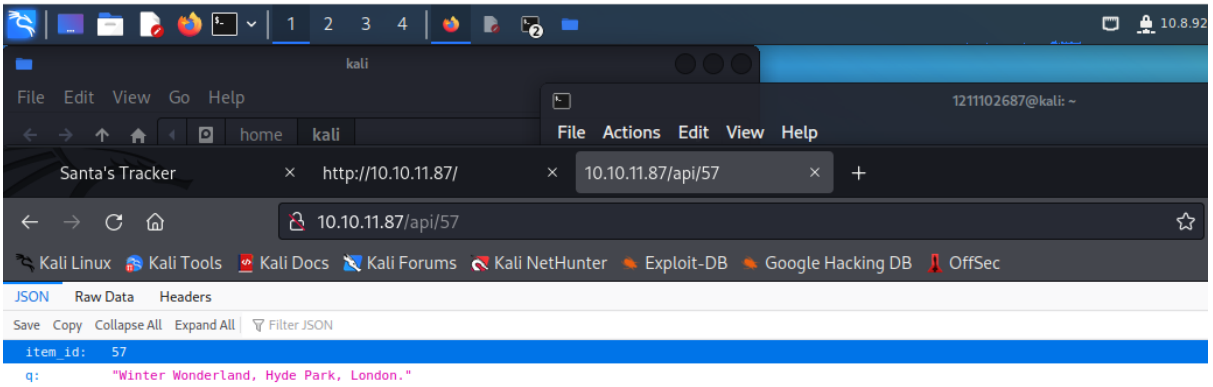
We visited the API directory and use 57 as our API key.

The Raw Data was shown after pressing the raw data tab in Firefox.



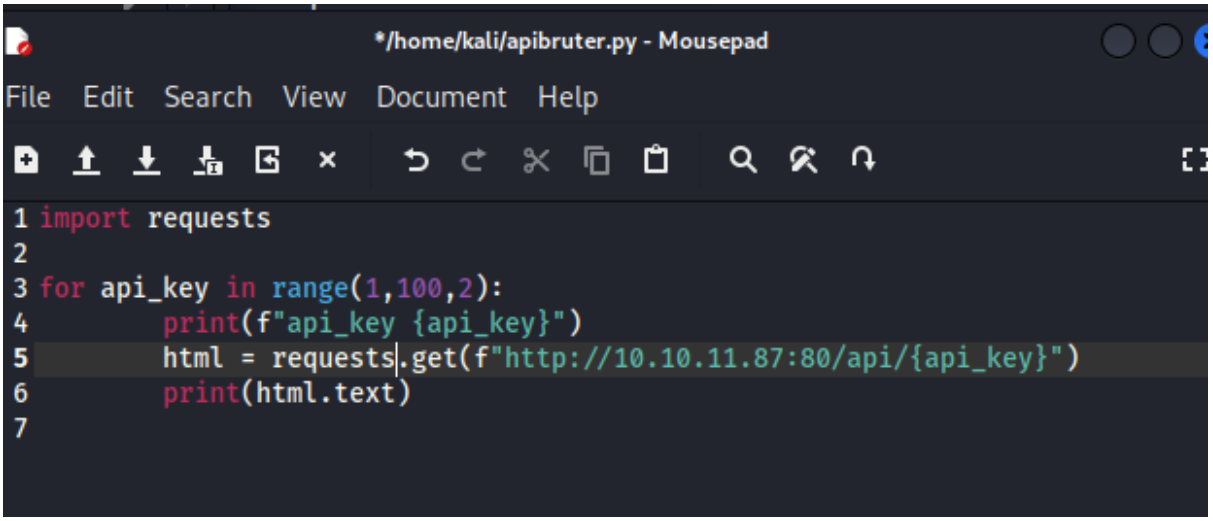
Question 5

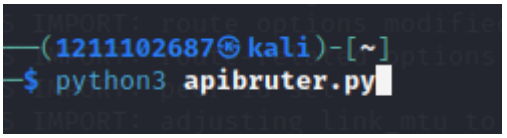
Santa's location was shown after we successfully entered the API directory with 57 as the API key.

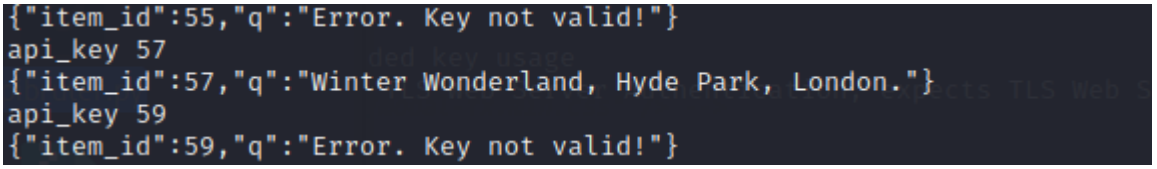


Question 6

We created a python script to loop through the possible odd number API key on the API directory from the range of 1-100 and run it in the terminal.







Thought Process/Methodology:

First, we ran a nmap scan on the IP address to find open ports, and two open ports were found. We decided to investigate port 80 as it is used to host an HTTP service. We entered the IP address with the port number on firefox's search bar and successfully entered Santa's Tracking System. The template used which is BULMA was shown on the top left of the page. We right-clicked and view the page source of the page. We found the directory for the API in the page source. We created a python script to loop through the possible odd number API key on the API directory from the range of 1-100 and run it in the terminal. We got a response from API key 57. We visited the API directory and use 57 as our API key. Santa's location was shown after we successfully entered the API directory with 57 as the API key. The Raw Data was also shown after pressing the raw data tab in Firefox.

Day 17[Reverse Engineering]- ReverseELFneering

Tools used: Kali Linux, Firefox, ssh. radare2

Solution/walkthrough:

Question 1

The size in bytes for the data type can be obtained from the table in the task description in THM.

3. Register me this, register me that...

The core of assembly language involves using registers to do the following:

- Transfer data between memory and register, and vice versa
- Perform arithmetic operations on registers and data
- Transfer control to other parts of the program Since the architecture is x86-64, the registers are 64 bit and Intel has a list of 16 registers:

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 to analyze the program in radare2 can be seen in the task description in THM.

> 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

The command to set a breakpoint in radare2 can be seen in the task description in THM.

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

The command to execute the program until we hit a breakpoint can be seen in the task description in THM.

> 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

Question 5

We SSH into our target machine with the username elfmceager and the password adventofcyber

```
File  Actions  Edit  View  Help
┌──(1211102753㉿kali)-[~]
└─$ ssh elfmceager@10.10.253.253
The authenticity of host '10.10.253.253 (10.10.253.253)' can't be establishe
d.
ED25519 key fingerprint is SHA256:+Yl8Ef3BjQ7HNTMf6qew50LnmiqEXXSzLqgX82k/RS
g.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.253.253' (ED25519) to the list of known ho
sts.
elfmceager@10.10.253.253'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 Thu Jul 14 15:44:36 UTC 2022

  System load:  0.93               Processes:           103
  Usage of /:   40.8% of 11.75GB   Users logged in:     0
  Memory usage: 8%                 IP address for ens5: 10.10.253.253
  Swap usage:   0%


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


Last login: Wed Dec 16 18:25:51 2020 from 192.168.190.1
```

We use radare2 to enter debug mode so that we can learn more about the challenge1 file. We do a full analysis of the file using the aa command. We run the afl command to find a list of the functions and then filter our results with grep.

```
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1497 started...
= attach 1497 1497
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
asm.bits 64
[0x00400a30]> aa
[ WARNING : block size exceeding max block size at 0x006ba220
[+] Try changing it with e anal.bb.maxsize
 WARNING : block size exceeding max block size at 0x006bc860
[+] Try changing it with e anal.bb.maxsize
[x] Analyze all flags starting with sym. and entry0 (aa)
[0x00400a30]> afl | grep main
0x00400b4d    1 35           sym.main
0x00400de0   10 1007 → 219   sym.__libc_start_main
0x00403840   39 661  → 629   sym._nl_find_domain
0x00403ae0  308 5366 → 5301  sym._nl_load_domain
0x00415ef0    1 43           sym._IO_switch_to_main_get_area
0x0044ce10    1 8            sym._dl_get_dl_main_map
0x00470430    1 49           sym._IO_switch_to_main_wget_area
0x0048f9f0    7 73   → 69    sym._nl_finddomain_subfreeres
0x0048fa40   16 247  → 237   sym._nl_unload_domain
```

There is a function at main. We examine the assembly code at main by running the command pdf @main.

```
[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
```

We use db to put a breakpoint at the address of local_ch. We use dc to run the program and we received a message that we have stopped at the breakpoint we set. We use the command px @rbp-0xc to view the contents of the local_ch variable.

```
[0×00400a30]> db 0×00400b51
[0×00400a30]> dc
hit breakpoint at: 400b51
[0×00400b51]> px @ rbp-0×c
- offset -       0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0×7ffea2ce51b4  0000 0000 1890 6b00 0000 0000 4018 4000  ......k.....@.@.
0×7ffea2ce51c4  0000 0000 e910 4000 0000 0000 0000 0000  ......@.........
0×7ffea2ce51d4  0000 0000 0000 0000 0100 0000 e852 cea2  .............R..
0×7ffea2ce51e4  fe7f 0000 4d0b 4000 0000 0000 0000 0000  ....M.@.........
0×7ffea2ce51f4  0000 0000 1700 0000 0100 0000 0000 0000  ................
0×7ffea2ce5204  0000 0000 0000 0000 0200 0000 0000 0000  ................
0×7ffea2ce5214  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffea2ce5224  0000 0000 0000 0000 0000 0000 0004 4000  ..............@.
0×7ffea2ce5234  0000 0000 74a9 3af8 964d eaf2 e018 4000  ....t.:..M....@.
0×7ffea2ce5244  0000 0000 0000 0000 0000 0000 1890 6b00  ..............k.
0×7ffea2ce5254  0000 0000 0000 0000 0000 0000 74a9 1a6b  ............t..k
0×7ffea2ce5264  8a08 170d 74a9 8ee9 964d eaf2 0000 0000  ....t....M......
0×7ffea2ce5274  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffea2ce5284  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffea2ce5294  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffea2ce52a4  0000 0000 0000 0000 0000 0000 0000 0000  ................
```

We use ds to go to the next one and we obtained the value of 1.

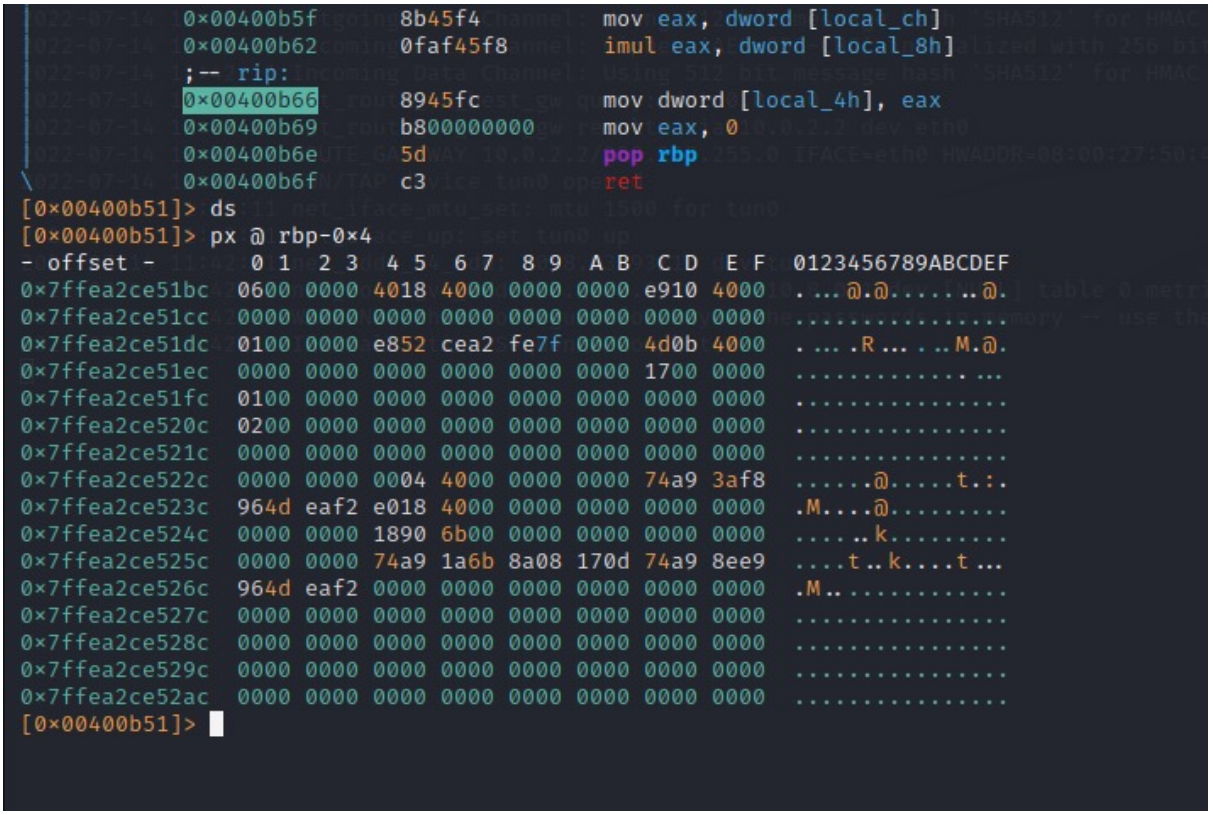
```
[0×00400b51]> ds
[0×00400b51]> px @rbp-0×c
- offset -       0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0×7ffea2ce51b4  0100 0000 1890 6b00 0000 0000 4018 4000  ......k.....@.@.
0×7ffea2ce51c4  0000 0000 e910 4000 0000 0000 0000 0000  ......@.........
0×7ffea2ce51d4  0000 0000 0000 0000 0100 0000 e852 cea2  .............R..
0×7ffea2ce51e4  fe7f 0000 4d0b 4000 0000 0000 0000 0000  ....M.@.........
0×7ffea2ce51f4  0000 0000 1700 0000 0100 0000 0000 0000  ................
0×7ffea2ce5204  0000 0000 0000 0000 0200 0000 0000 0000  ................
0×7ffea2ce5214  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffea2ce5224  0000 0000 0000 0000 0000 0000 0004 4000  ..............@.
0×7ffea2ce5234  0000 0000 74a9 3af8 964d eaf2 e018 4000  ....t.:..M....@.
0×7ffea2ce5244  0000 0000 0000 0000 0000 0000 1890 6b00  ..............k.
0×7ffea2ce5254  0000 0000 0000 0000 0000 0000 74a9 1a6b  ............t..k
0×7ffea2ce5264  8a08 170d 74a9 8ee9 964d eaf2 0000 0000  ....t....M......
0×7ffea2ce5274  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffea2ce5284  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffea2ce5294  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffea2ce52a4  0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Question 6

We use ds to move to the next line and then use dr to see the value of eax.

Question 7

We use the ds command to get to 0x00400b66 and then check the value of local_4h with px @ rbp-0x4. The value is 6.



Thought Process/Methodology:

After reading the task description in THM, the size in bytes for the data type can be obtained from the table in the task description. The command aa is used to analyze the program in radare2. The command db is used to set a breakpoint in radare2. The command dc is used to execute the program until we hit a breakpoint. We SSH into our target machine with the username elfmceager and the password adventofcyber. We use radare2 to enter debug mode so that we can learn more about the challenge1 file. We do a full analysis of the file using the aa command. As most programs have an entry point defined as main, we run the afl command to find a list of the functions and then filter our results with grep. It turns out that there is a function at main. We examine the assembly code at main by running the command pdf @main. We use db to put a breakpoint at the address of local_ch. Then, we use dc to run the program and we received a message that we have stopped at the breakpoint we set. We use the command px @rbp-0xc to view the contents of the local_ch variable and then used ds to go to the next one. We obtained the value of 1. We use ds to move to the next line and then use dr to see the value of eax. We use the ds command to get to 0x00400b66 and then check the value of local_4h with px @ rbp-0x4. The value is 6.

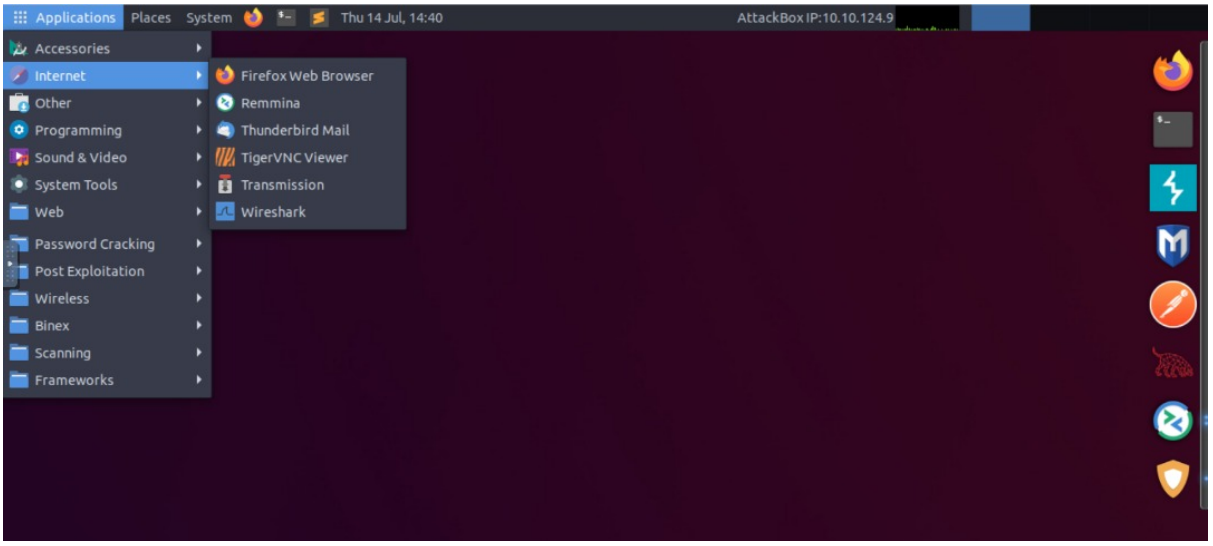
Day 18 [Reverse Engineering] - The Bits of Christmas

Tools used: Kali Linux, Remmina, CyberChef

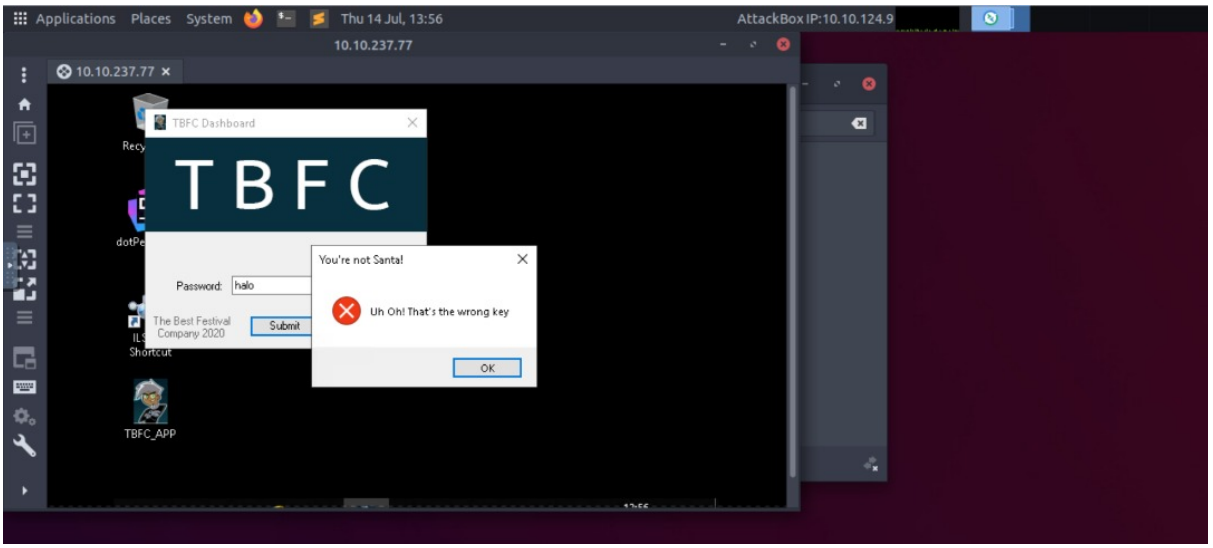
Solution/walkthrough:

Question 1

First we open Remmina. We key in our IP address, username, and password which were provided in the task description in THM.

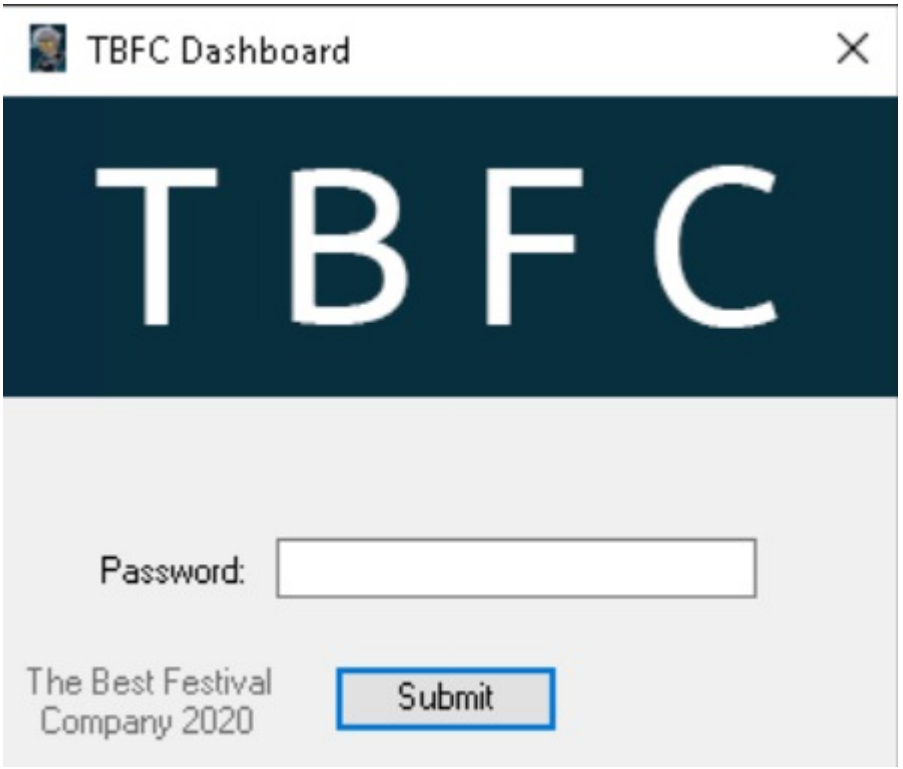


We tried entering the wrong password to see what will be shown.



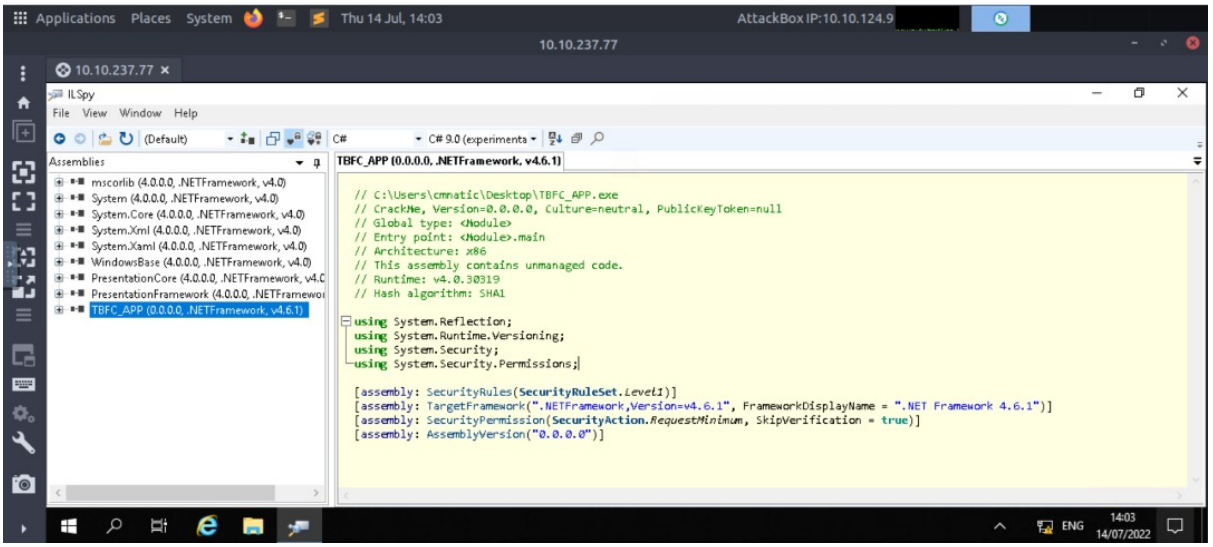
Question 2

The full name of TBFC can be seen on the bottom left of the TBFC Dashboard.

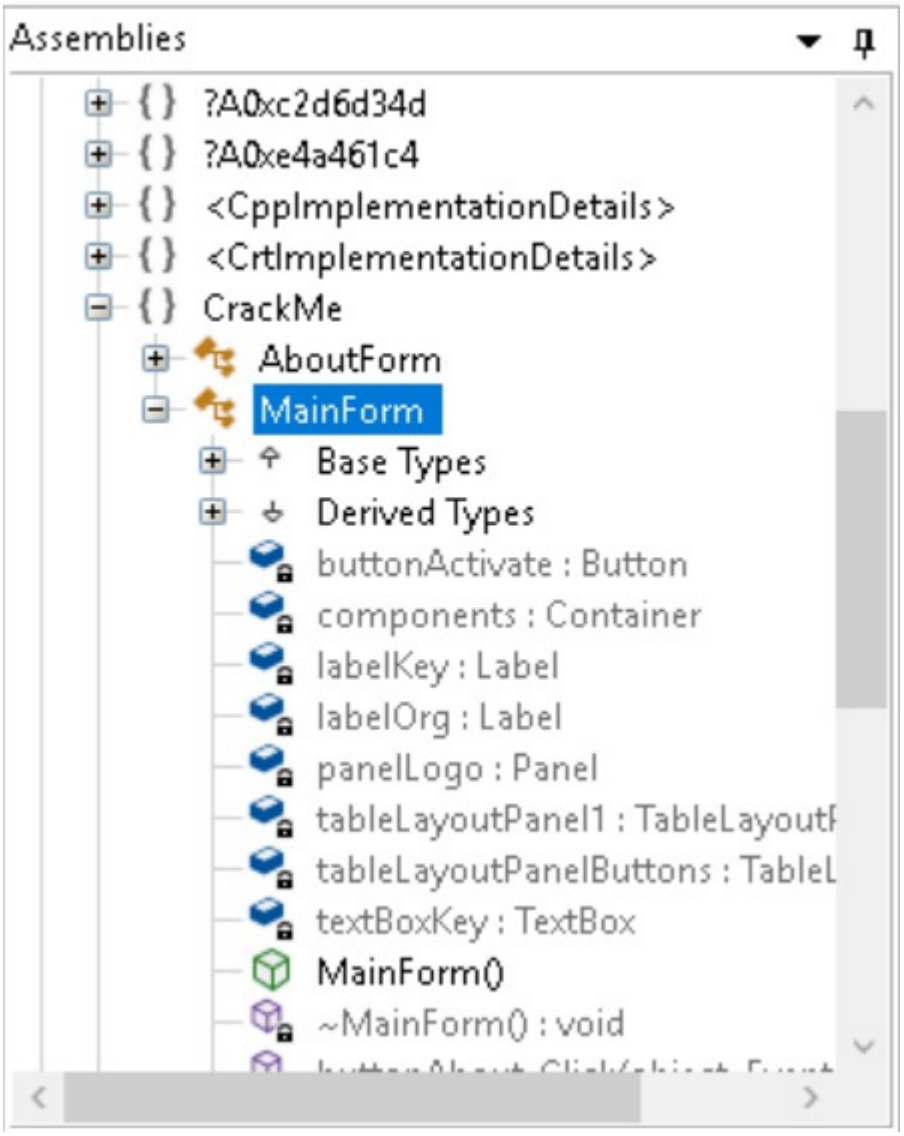


Question 3

We opened TBFC_APP in ILSpy and decompile it. We are asked to find Santa's password.

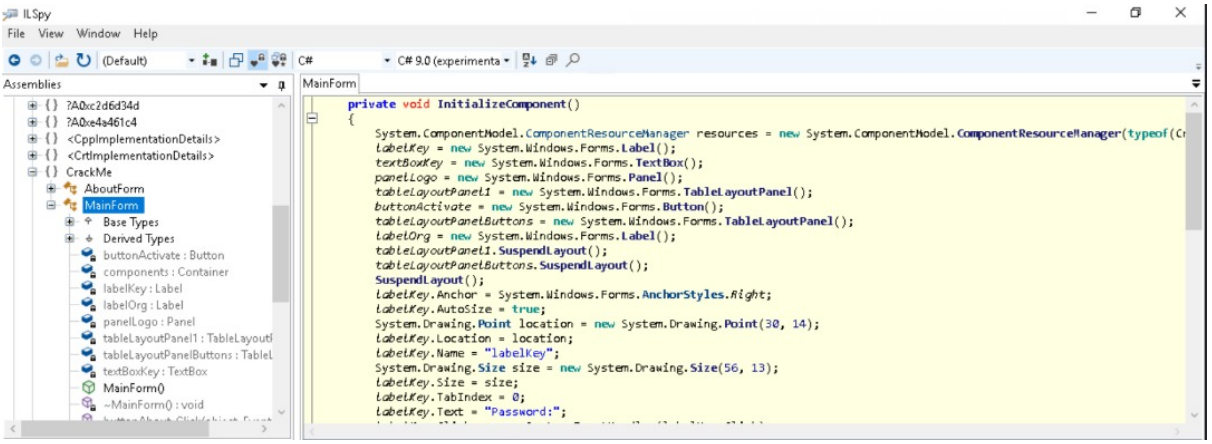


The module that catches our attention is CrackMe as it contains a lot of information.



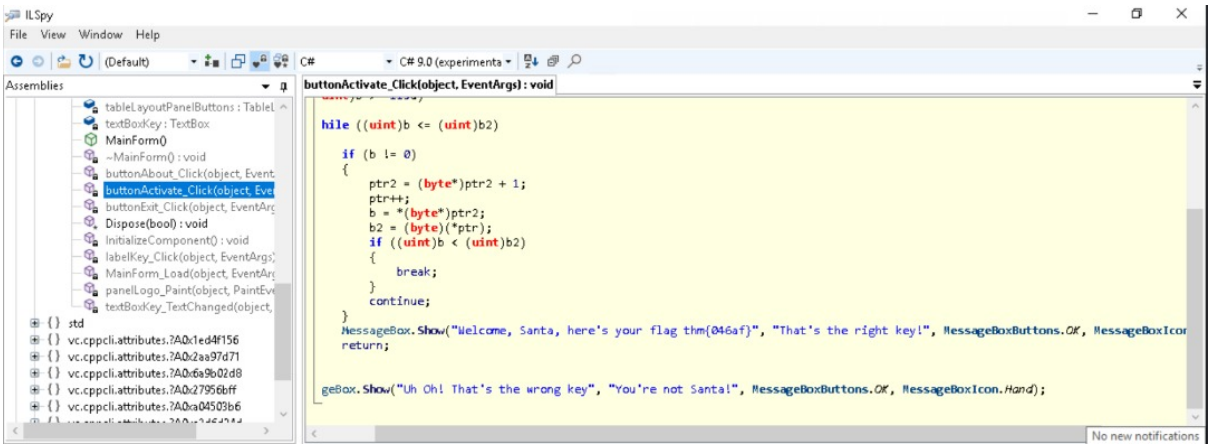
Question 4

It was shown that MainForm contains the information we are looking for.



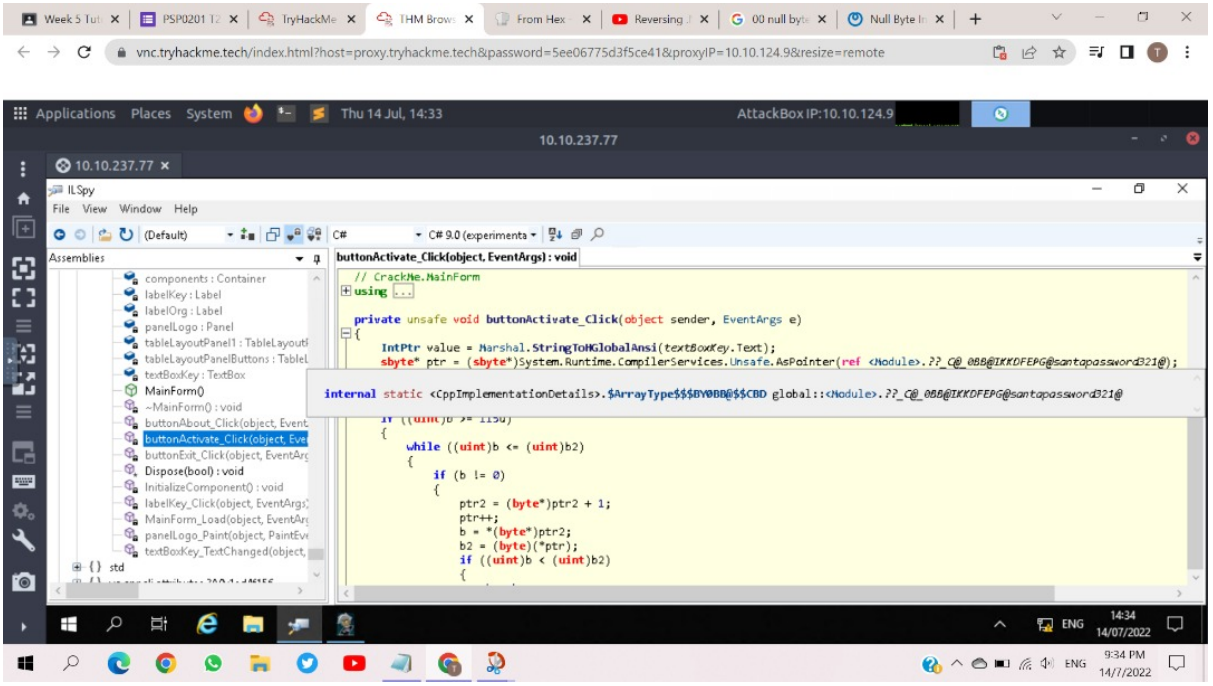
Question 5

After a bit of searching, we ended up finding something interesting in buttonActivate_Click, it contains the word password.

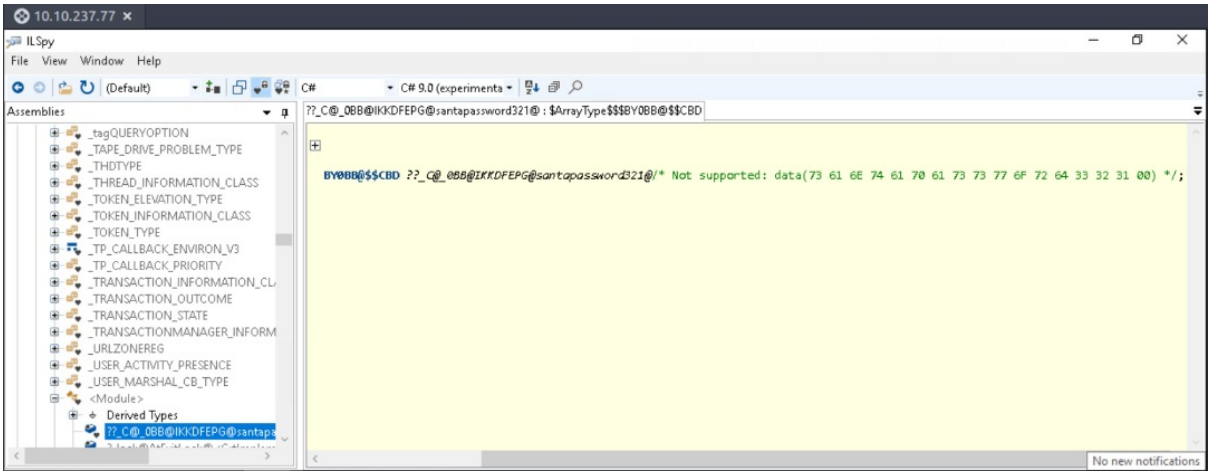


Question 6

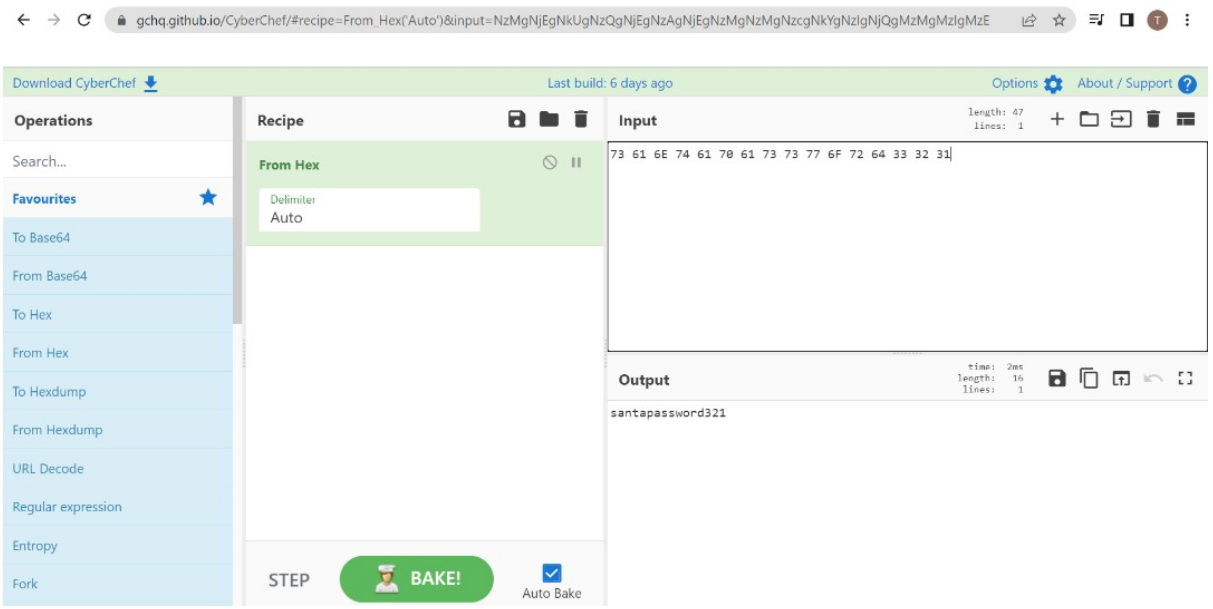
We double-clicked the password string to check.



We copied the hexadecimal and convert it to string using CyberChef. We left out the 00 as 00 is a null byte that terminates the string.

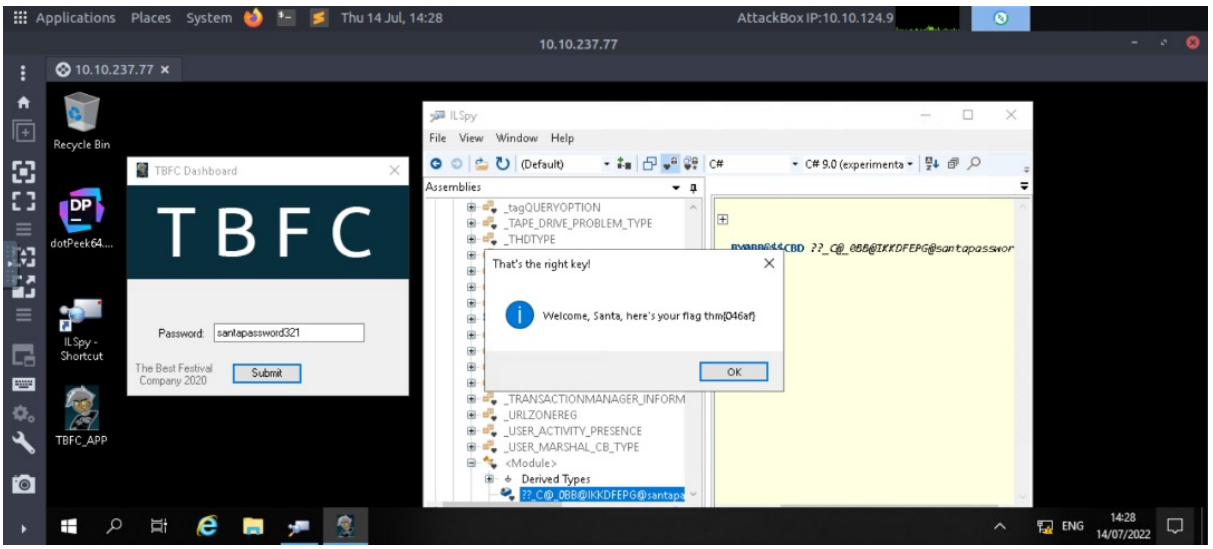


After converting, Santa's password was shown.



Question 7

After keying in the correct password, the flag was shown.



Thought Process/Methodology:

First we open Remmina. We key in our IP address, username, and password which were provided in the task description in THM. Before entering the correct password, we tried entering the wrong password to see what will be shown. The full name of TBFC was shown on the bottom left of the TBFC Dashboard. We opened TBFC_APP in ILSpy and decompile it. We are asked to find Santa's password. The module CrackMe caught our attention as it contains a lot of information. It was shown that MainForm contains the information we are looking for. After a bit of searching, we ended up finding something interesting in buttonActivate_Click. It contains the word password. We double-clicked the password string to check it. We copied the hexadecimal and convert it to string using CyberChef. We left out the 00 as 00 is a null byte that terminates the string. After converting, Santa's password was shown. We keyed in the password and the flag was shown.

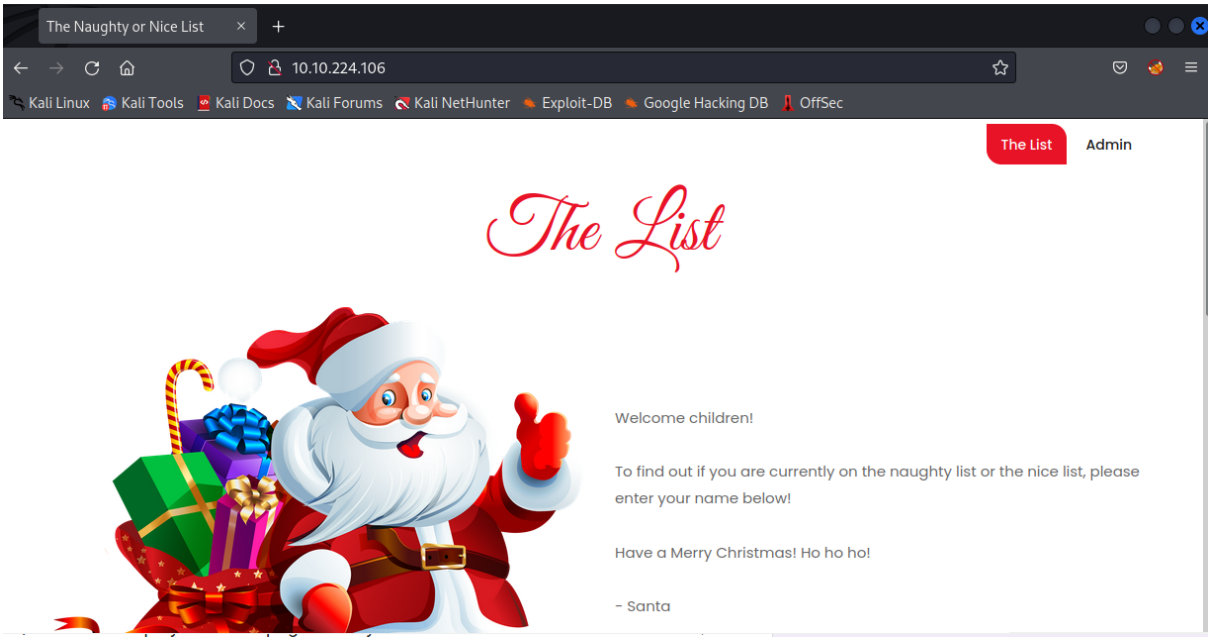
Day 19 [Web Exploitation] - The Naughty or Nice List

Tools used: Kali Linux, Firefox, CyberChef

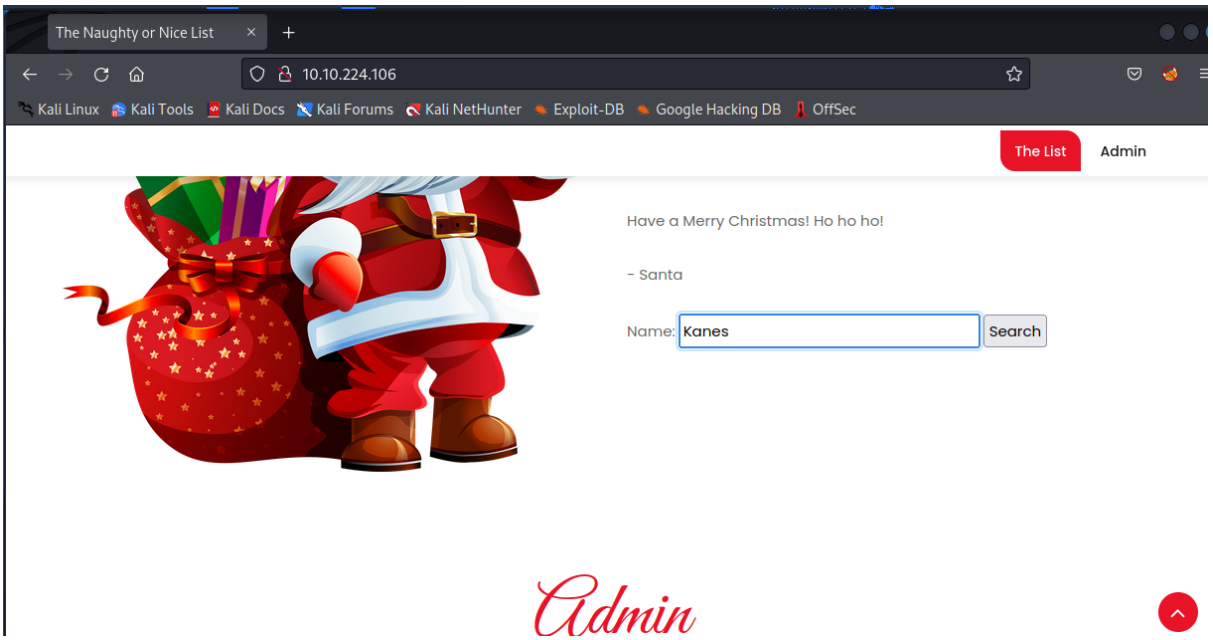
Solution/walkthrough:

Question 1

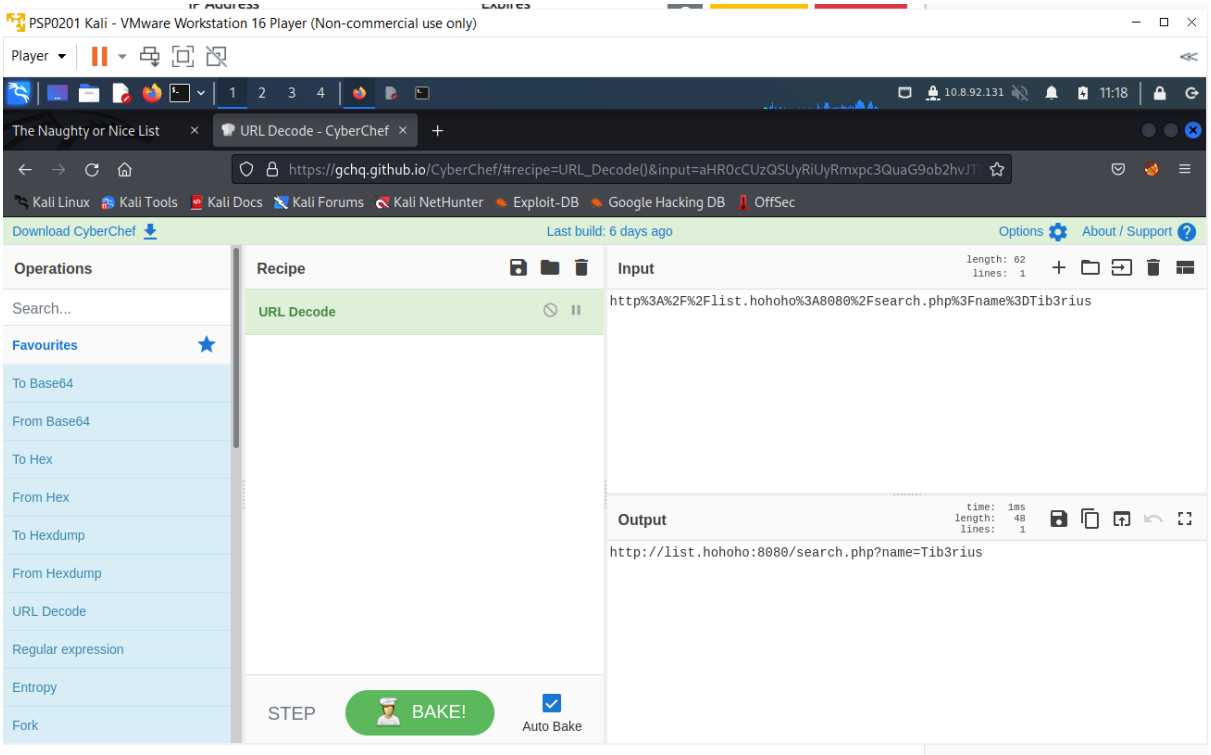
We entered the IP address and were shown a list.



We entered the given names in the form to determine if they are on the naughty or nice list.

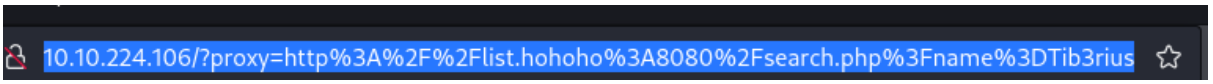


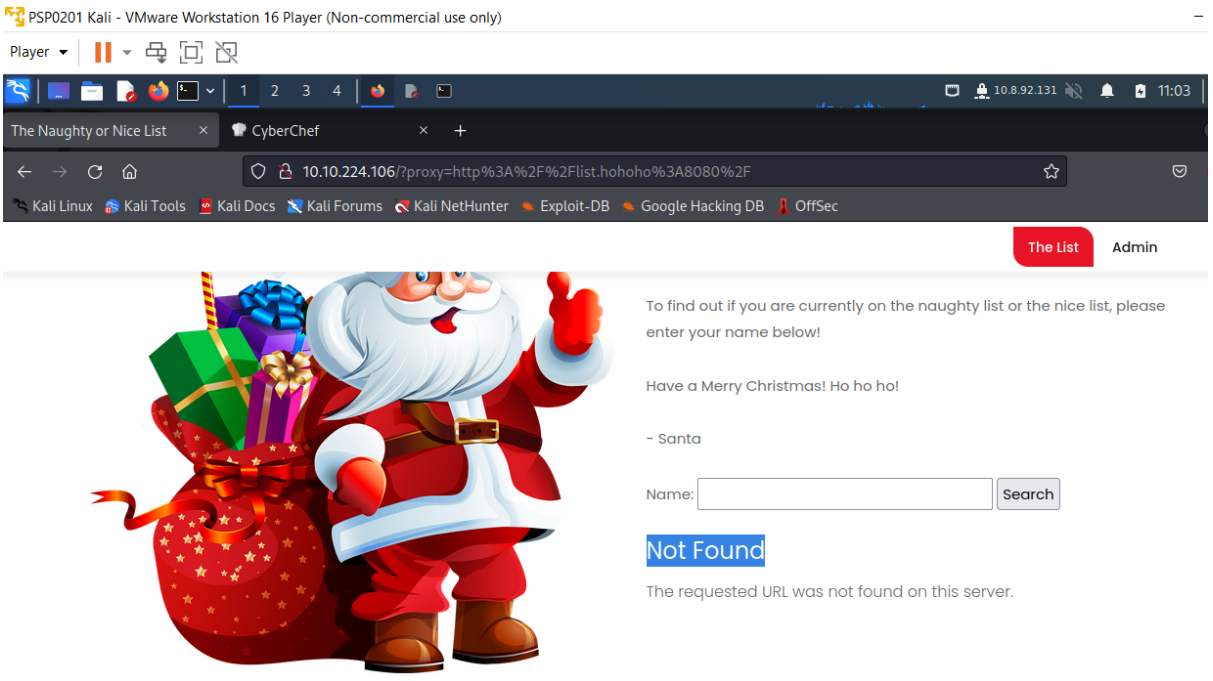
We use URL Decoder on the value of the "proxy" parameter, and we get http://list.hohoho:8080/search.php?name=Tib3rius



Question 2

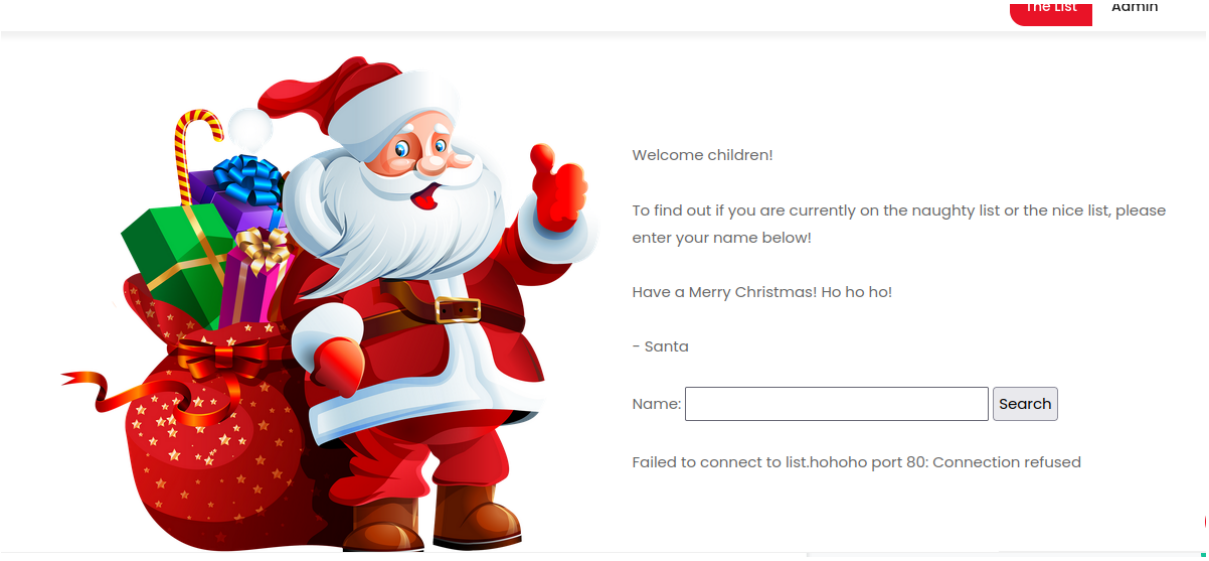
We fetch the root of the same site and browse it. Not Found was displayed.





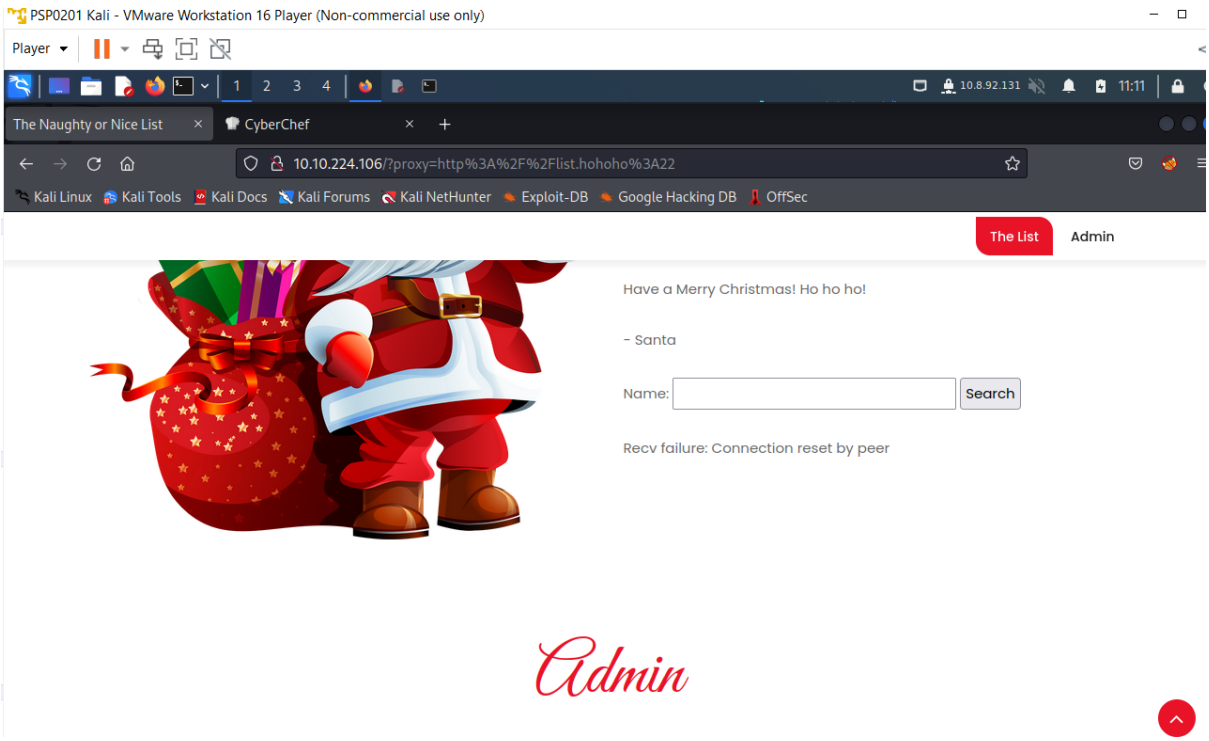
Question 3

We try changing the port number from 8080 to just 80.



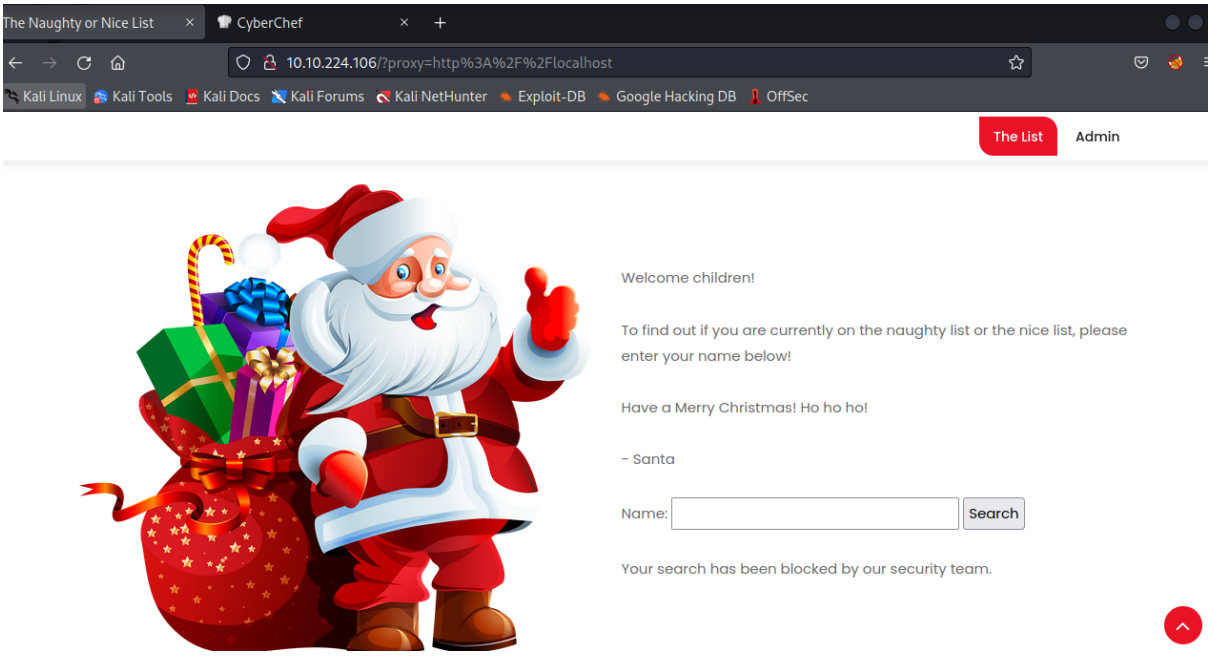
Question 4

We try changing the port number to 22.



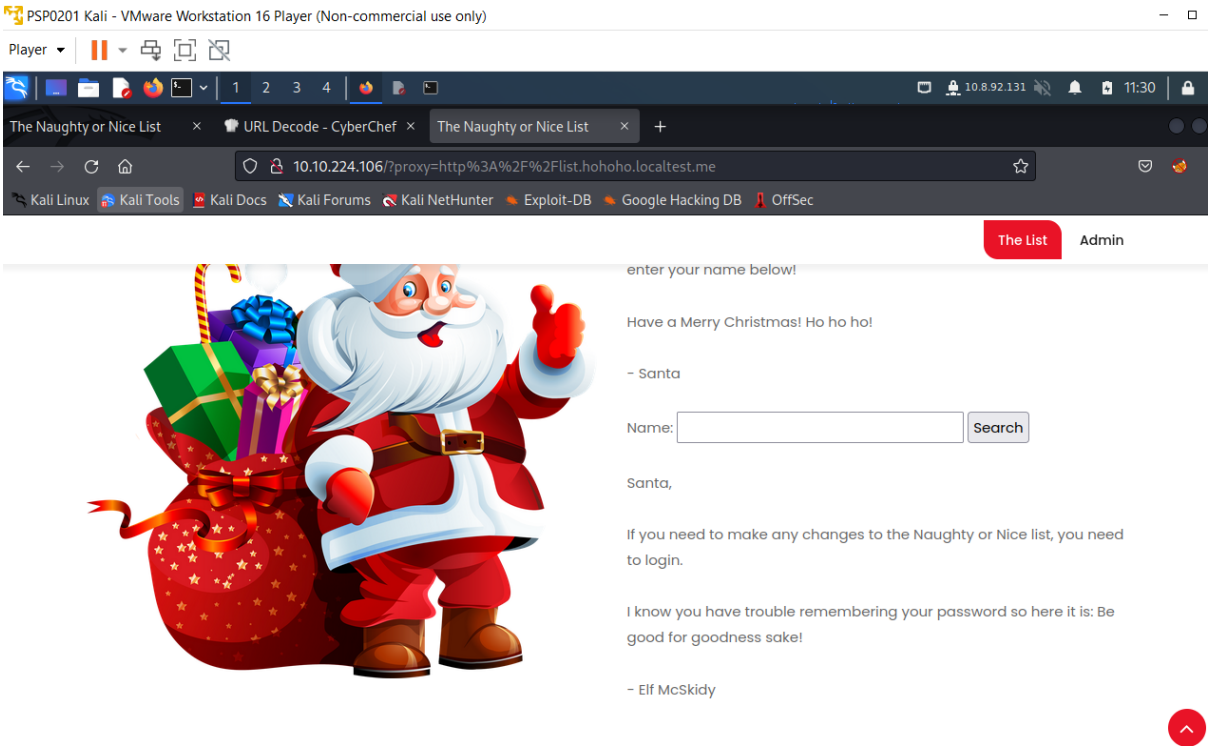
Question 5

We access services running locally on the server by replacing the list.hohoho hostname with "localhost".



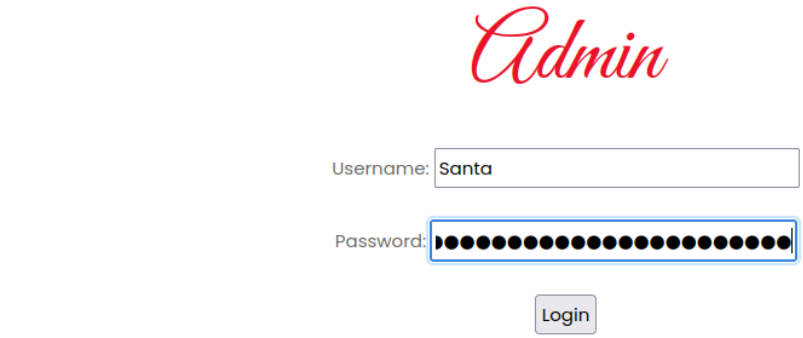
Question 6

We set the hostname in the URL to "list.hohoho.localtest.me", bypass the check, and access local services. Santa's password was shown.

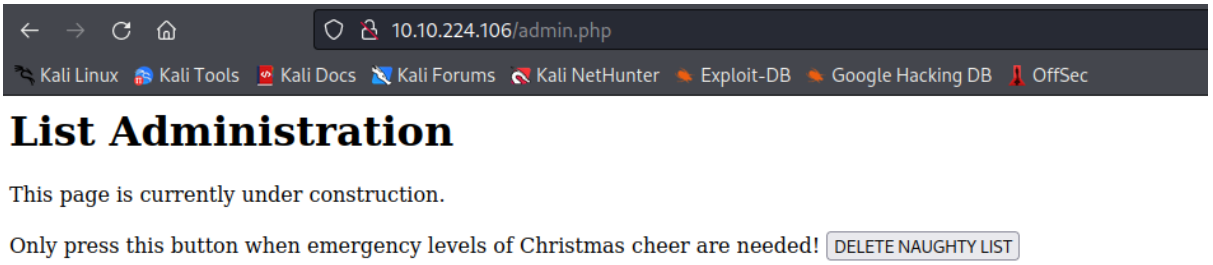


Question 7

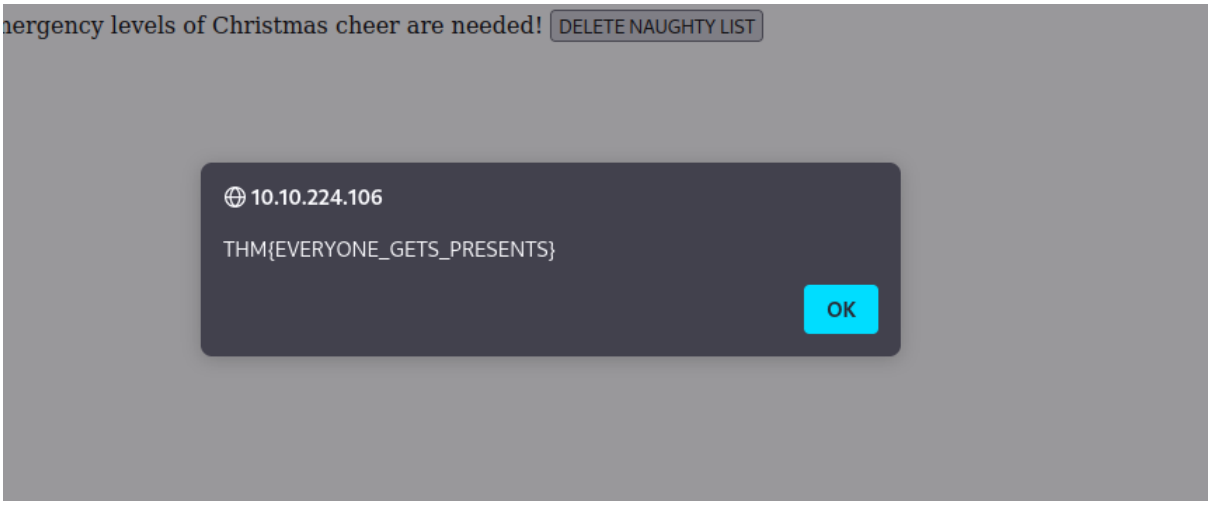
We guessed Santa's username as Santa and keyed in the password.



We were then shown the administration list.



After deleting the naughty list, the flag was shown.



Thought Process/Methodology:

We entered the IP address and were shown a naughty or nice list. We entered the given names in the form to determine if they are on the naughty or nice list. We use URL Decoder on the value of the "proxy" parameter, and we get http://list.hohoho:8080/search.php?name=Tib3rius. We fetch the root of the same site and browse it. Not Found was then displayed. We try changing the port number from 8080 to just 80 but it was not open. We try changing the port number to 22, port 22 is open but did not understand what was sent. We access services running locally on the server by replacing the list.hohoho hostname with "localhost" but our search was blocked. We set the hostname in the URL to "list.hohoho.localtest.me", bypass the check, and access local services. Santa's password was then shown. We guessed Santa's username as Santa and keyed in the password. We were then shown the administration list and the option to delete the naughty list. After deleting the naughty list, the flag was shown.

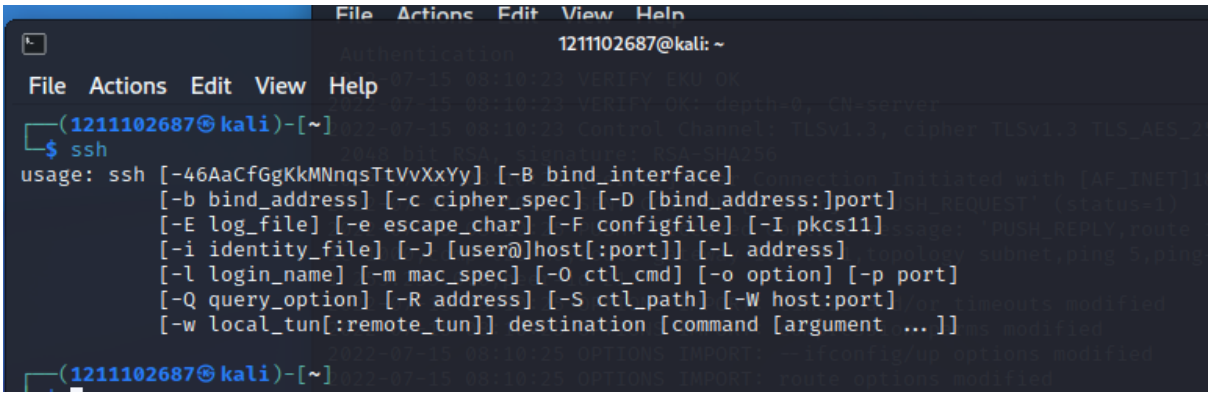
Day 20 [Blue Teaming] - PowershELlF to the rescue

Tools used: Kali Linux, Firefox, Powershell, SSH

Solution/walkthrough:

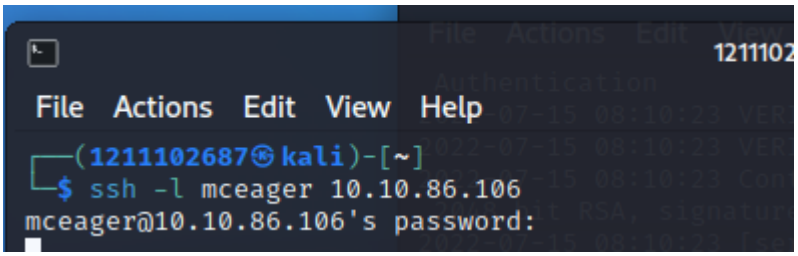
Question 1

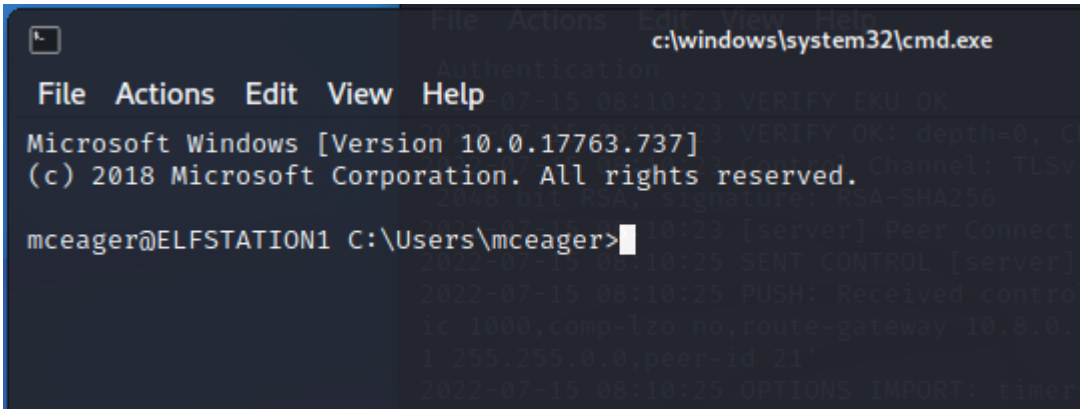
We entered the ssh manual using SSH command and got the answer.



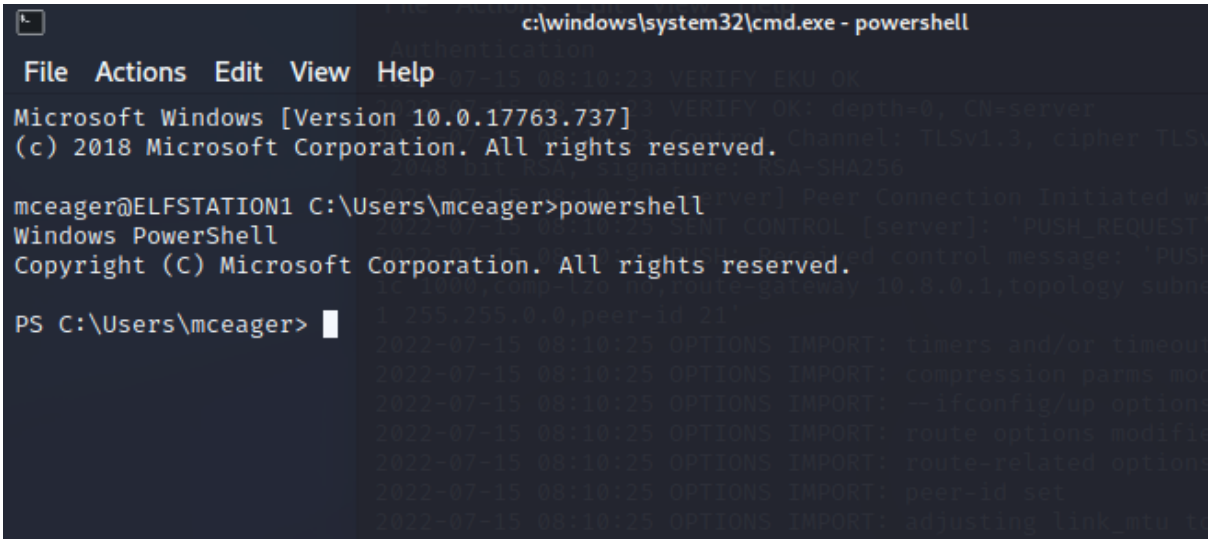
Question 2

We use SSH to connect to the remote machine and key in the password.

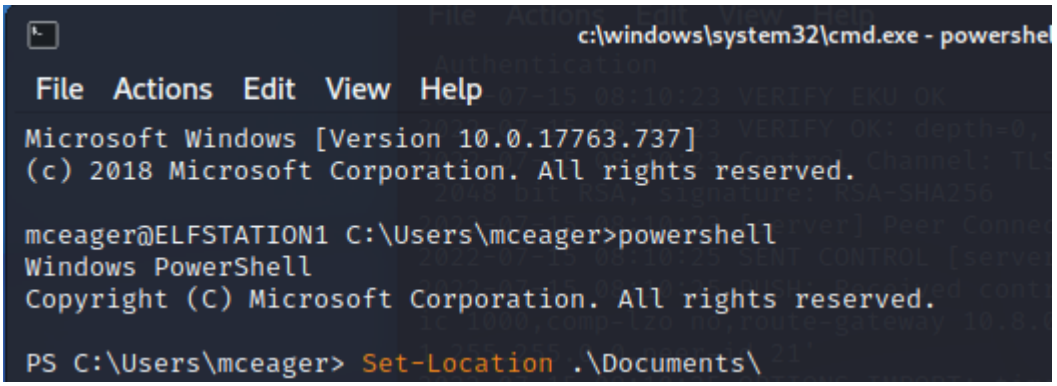




We launched Powershell.



We navigate to the Documents folder.



We use GET to obtain the hidden elf file within the Documents folder and read the content.

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

Question 3

We change the directory to Desktop.

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

We searched for the hidden folder in Desktop. We change our directory to the folder and proceeded to read the contents inside the text file. The movie elf 2 wants was shown.

```
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


PS C:\Users\mceager\Desktop> cd elf2wo
PS C:\Users\mceager\Desktop\elf2wo> ls


    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!
PS C:\Users\mceager\Desktop\elf2wo>
```

Question 4

We change the directory to Windows and proceeded to System32 as it contains most of the important files of the Operating System.

```
PS C:\Users\mceager\Desktop\elf2wo> cd C:/Windows
PS C:\Windows> ls


    Directory: C:\Windows


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/15/2018  12:19 AM                ADFS
d-----         9/15/2018  12:19 AM                appcompat
d-----          9/6/2019   5:31 PM                apppatch
d-----         12/7/2020  10:50 AM                AppReadiness
d-r---         9/15/2018   2:11 AM                assembly
d-----         9/15/2018  12:19 AM                bcastdvr
d-----         9/15/2018  12:19 AM                Boot
```

```
PS C:\Windows> cd System32
PS C:\Windows\System32>
```

We filtered the contents of System 32 and set it so that we find directories that contain the number 3.

```
PS C:\Windows\System32> Get-ChildItem -Hidden -Directory -Filter "*3*"


    Directory: C:\Windows\System32


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


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

Question 5

There is no list of contents in the folder which means the files are hidden, so we tried to get the hidden files.

```
PS C:\Windows\System32> Get-ChildItem -Hidden -Directory -Filter "*3*"


    Directory: C:\Windows\System32


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


PS C:\Windows\System32> cd 3lfthr3e
PS C:\Windows\System32\3lfthr3e> ls
PS C:\Windows\System32\3lfthr3e> dir
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>
```

There are too many words in the file so we used the measure-object command to count the total number of words in the file.

```
arbor
mediawiki
configurations
poison
PS C:\Windows\System32\3lfthr3e> cat 1.txt | Measure-Object -Word

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


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

Question 6

We try to find what words are at the indexes of 551 and 6991 in the 1 text file.

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

Question 7

We try to find the string that has the pattern of “redryder” in the 2 text file.

```
yhunkadqujzeo
cqbrbonoednnd
PS C:\Windows\System32\3lfthr3e> cat 2.txt | Select-String -Pattern "redryder"

redryderbbgun


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

Thought Process/Methodology:

First, we entered the ssh manual using SSH command. We use SSH to connect to the remote machine and key in the password. Then we launched Powershell and navigated to the Documents folder. We use GET to obtain the hidden elf file within the Documents folder and read the content. Elf 1 wants 2 front teeth. We change the directory to Desktop. We searched for the hidden folder in Desktop. Then, we change our directory to the folder and proceeded to read the contents inside the text file. We change the directory to Windows and proceeded to System32 as it contains most of the important files of the Operating System. We filtered the contents of System 32 and set it so that we find directories that contain the number 3. There is no list of contents in the folder which means the files are hidden, so we tried to get the hidden files. There are too many words in the file so we used the measure-object command to count the total number of words in the file. We try to find what words are at the indexes of 551 and 6991 in text file 1. We try to find the string that has the pattern of “redryder” in text file 2 and found redryderbbgun.