PSP0201

Week 5

Writeup

Group Name: Stellar

Members

ID	Name	Role
1211101145	Nurul Humairah binti Mohamad Kamaruddin	Leader
1211101216	Fatin Qistina binti Kamarul Irman	Member
1211102030	Ilyana Sofiya binti Muhammad Najeli	Member
1211103480	Nurul Afiqah binti Ismail	Member

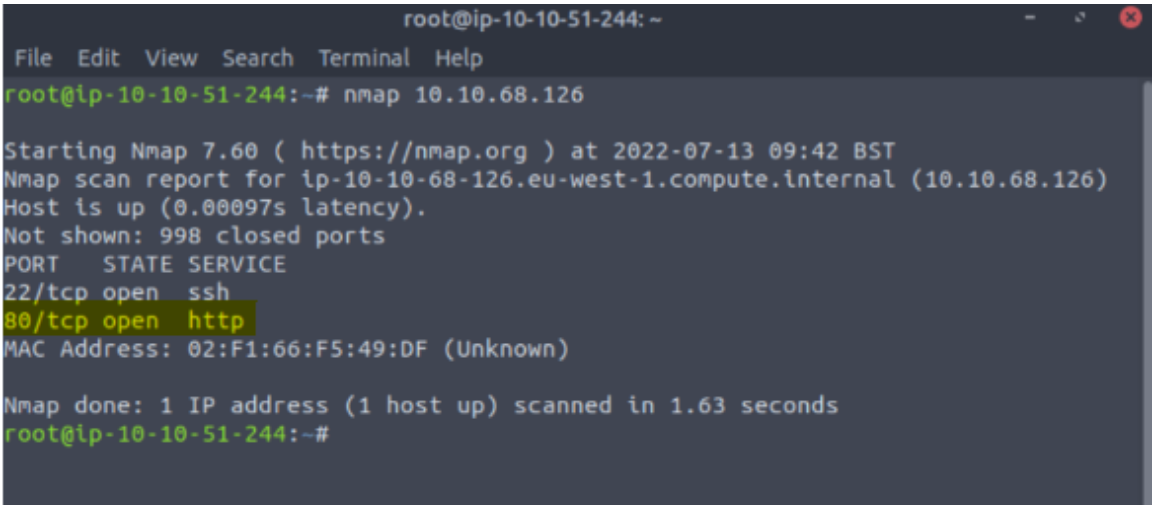
Day 16: Scripting – Help! Where is Santa?

Tools used: Kali Linux, Firefox, Python

Solution/walkthrough:

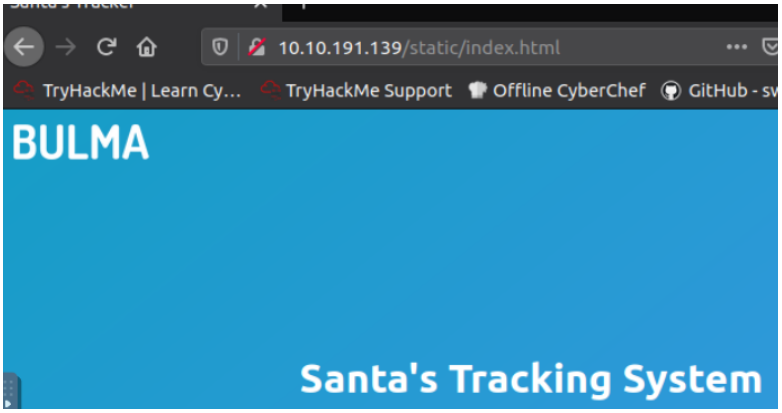
Question 1

The port number for the web server is **80**.



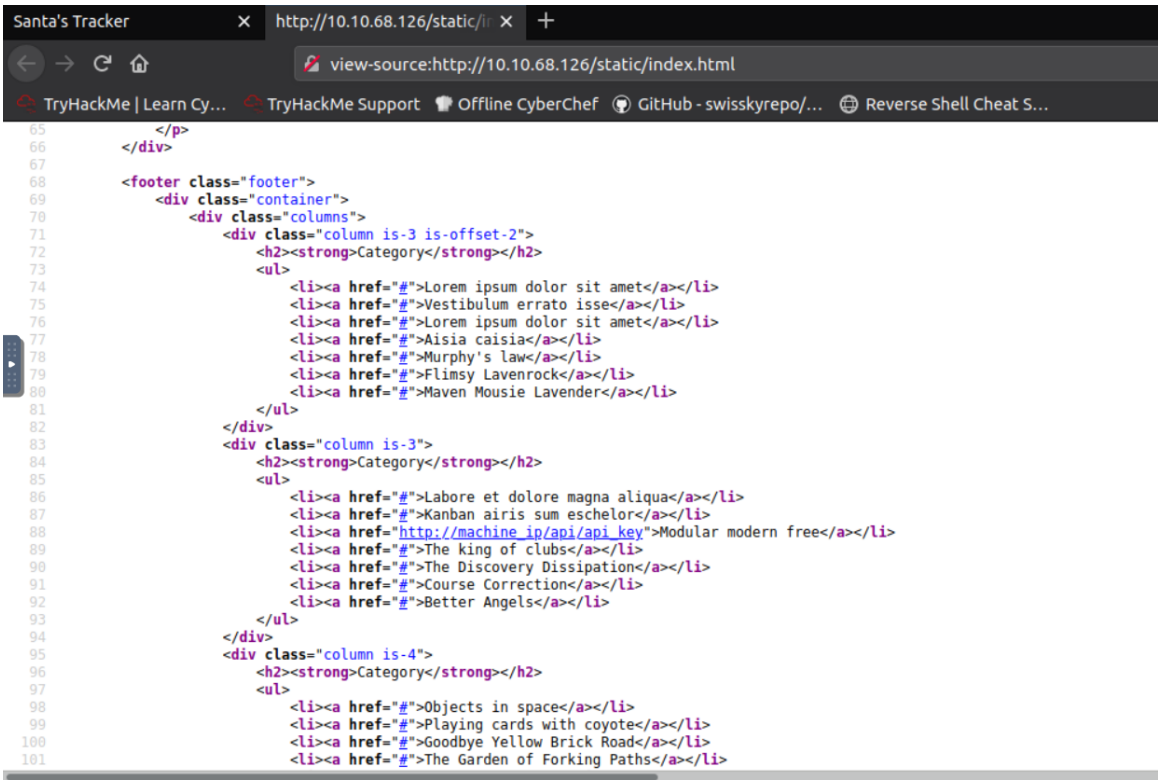
Question 2

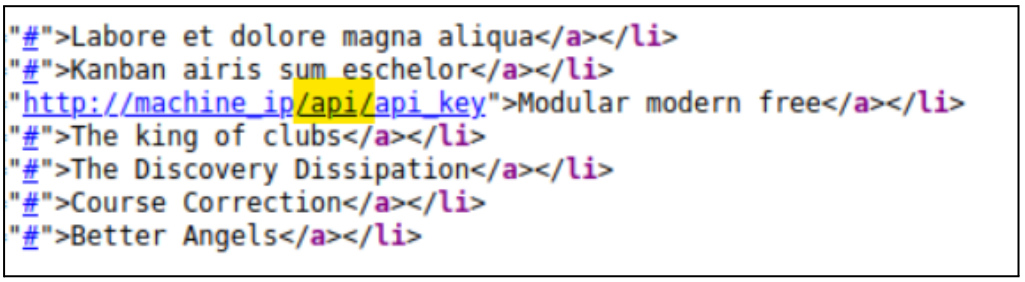
The templates being used are **BULMA**.



Question 3

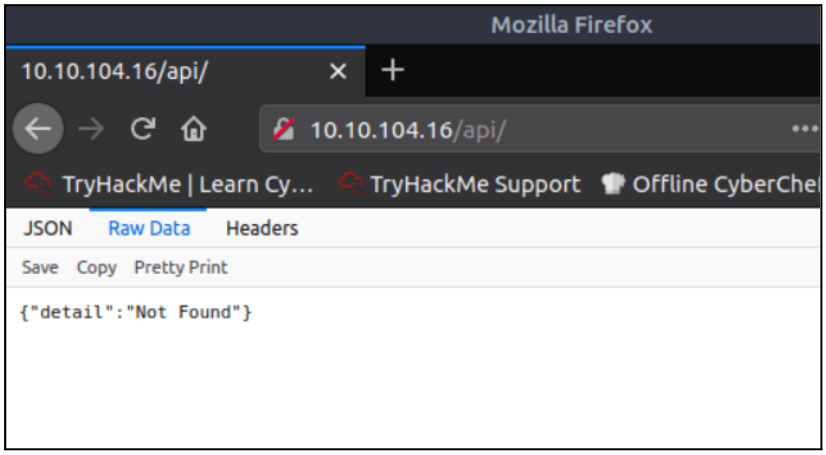
Without using enumerations tools such as Dirbuster, the directory for the API is **/api/** .





Question 4

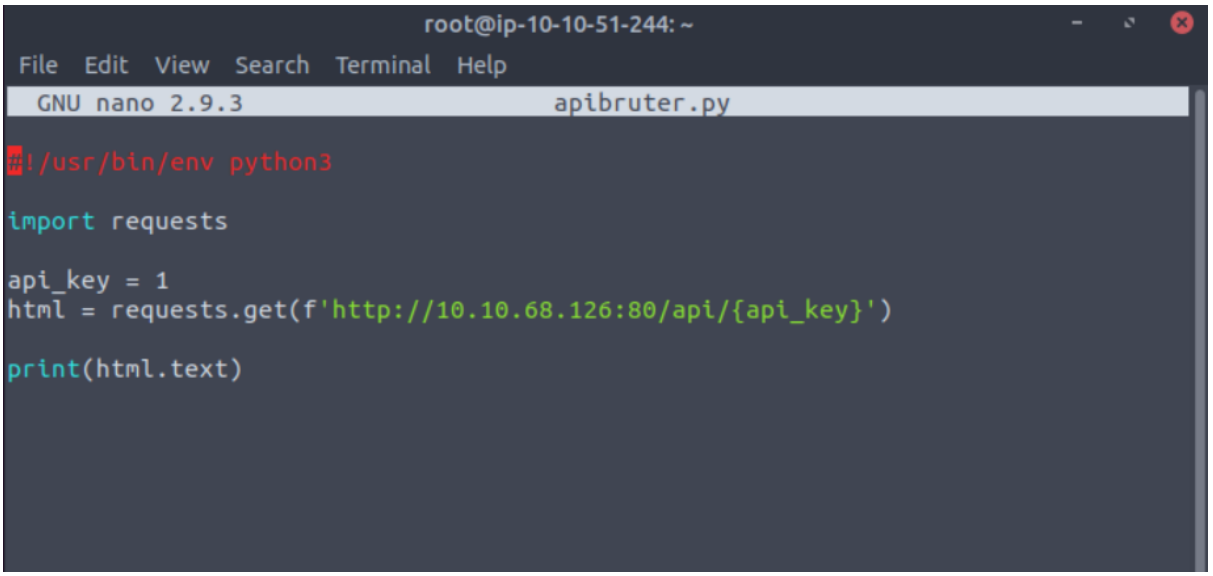
The Raw Data that was returned if no parameters are entered are {"detail":"Not Found"} .

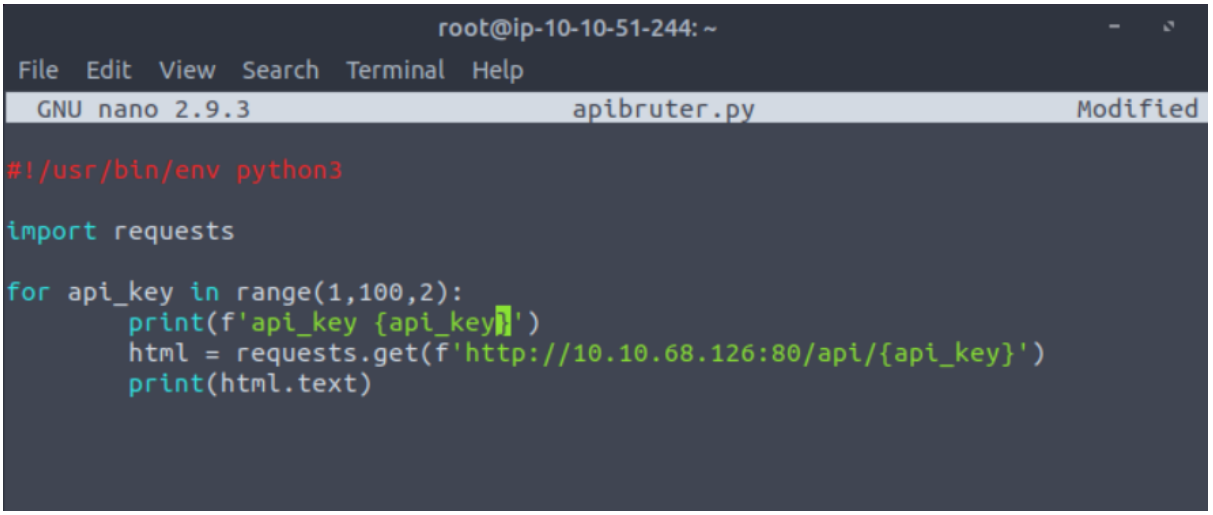


Question 5

Right now, Santa is at **Winter Wonderland, Hyde Park and London.**

```
root@ip-10-10-51-244:~# nano apibruter.py
root@ip-10-10-51-244:~# python3 apibruter.py
{"item_id":1,"q":"Error. Key not valid!"}
root@ip-10-10-51-244:~#
```





```
root@ip-10-10-51-244:~# python3 apibruter.py
api_key 1
{"item_id":1,"q":"Error. Key not valid!"}
api_key 3
{"item_id":3,"q":"Error. Key not valid!"}
api_key 5
{"item_id":5,"q":"Error. Key not valid!"}
api_key 7
{"item_id":7,"q":"Error. Key not valid!"}
api_key 9
{"item_id":9,"q":"Error. Key not valid!"}
api_key 11
{"item_id":11,"q":"Error. Key not valid!"}
api_key 13
{"item_id":13,"q":"Error. Key not valid!"}
api_key 15
{"item_id":15,"q":"Error. Key not valid!"}
```

```
{"item_id":51,"q":"Error. Key not valid!"}
api_key 53
{"item_id":53,"q":"Error. Key not valid!"}
api_key 55
{"item_id":55,"q":"Error. Key not valid!"}
api_key 57
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
api_key 59
{"item_id":59,"q":"Error. Key not valid!"}
api_key 61
{"item_id":61,"q":"Error. Key not valid!"}
api_key 63
{"item_id":63,"q":"Error. Key not valid!"}
```

Question 6

The correct API key is **57**.

```
{"item_id":51,"q":"Error. Key not valid!"}
api_key 53
{"item_id":53,"q":"Error. Key not valid!"}
api_key 55
{"item_id":55,"q":"Error. Key not valid!"}
api_key 57
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
api_key 59
{"item_id":59,"q":"Error. Key not valid!"}
api_key 61
{"item_id":61,"q":"Error. Key not valid!"}
api_key 63
{"item_id":63,"q":"Error. Key not valid!"}
```

Thought Process/Methodology:

First, question 1 asks for the port number we should use. As we learn before an easy way to scan the IP address to obtain the port number is by using Nmap. We'll need to write nmap Machine_ip. It will show which port numbers are available and which one we can use. According to the scan results, we need to use port 80 as it is open and not secured. Next is question 2, which wants to know what templates are being used on the website. We can see the answers on the left side of the website which is BULMA. Question 3, asks for the API directory. So, just go to the website, right-click, and chose the view page source. As it says on the website, there are a lot of hidden links on the website and we need to find a different link from the others. Later we'll be able to see the links which are http://machine_ip/api/api_key and as we can see here the directory for the API should be /api/. Moving on, question 4 wants to know the Raw Data that was returned if no parameters are entered. We can simply use the links that we've found earlier by inserting our machine IP address and deleting the api_key part. Search in firefox using the links that we've modified earlier, click on the Raw Data tab and we've found the answer which is {"detail":"Not Found"}. Question 5 then ask for the Santa locations right now while question 6 wants to know the correct API key. To solve both of these questions, we'll be applying the python knowledge we've learned before. Now, open a text editor such as nano in the terminal, and create a new file with python extension. Now, type import requests which is the library we need to download from day 15. We'll need to use loops to brute force into the website by trying all numbers from 1 to 100 that is odd numbers. Using for loops, set the range (1,100,2), and for every API key, we'll make it print the API key so that we know which one is successful. Also, do a get request for the link that was given in the task and print the results. Lastly, we need to type python3 file_name.py in the terminal and we'll see a list of attempts starting from number 1 and most of them will say that the key is not valid which means that was not the API key we wanted. Scroll until we found an API key that results in a successful attempt which is number 57. It will say where is Santa right now and in this case, Santa is at Winter Wonderland, Hyde Park, and London.

Day 17: Reverse Engineering – ReverseELFneering

Tools used: Terminal Emulator

Solution/walkthrough:

Question 1

Match the data type with the size in bytes:

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

Question 2

The command to analyse the program in radare2 is **aa**.

```
[0×00400a30]> aa
[ WARNING : block size exceeding max block size at 0×006ba220
[+] Try changing it with e anal.bb.maxsize
 WARNING : block size exceeding max block size at 0×006bc860
[+] Try changing it with e anal.bb.maxsize
[x] Analyze all flags starting with sym. and entry0 (aa)
```

Question 3

The command to set a breakpoint in radare2 is **db**.

```
[0×00400b5f]> ds
[0×00400b5f]> db 0×00400b69
[0×00400b5f]> pdf @ main
            ;-- main:
/ (fcn) sym.main 35
|   sym.main ();
|           ; var int local_ch @ rbp-0×c
|           ; var int local_8h @ rbp-0×8
|           ; var int local_4h @ rbp-0×4
|              ; DATA XREF from 0×00400a4d (entry0)
|           0×00400b4d      55              push rbp
|           0×00400b4e      4889e5          mov rbp, rsp
|           0×00400b51      c745f4010000.   mov dword [local_ch], 1
|           0×00400b58      c745f8060000.   mov dword [local_8h], 6
|           0×00400b5f b    8b45f4          mov eax, dword [local_ch]
|           0×00400b62      0faf45f8        imul eax, dword [local_8h]
|           0×00400b66      8945fc          mov dword [local_4h], eax
|           0×00400b69 b    b800000000      mov eax, 0
|           ;-- rip:
|           0×00400b6e      5d              pop rbp
\           0×00400b6f      c3              ret
```

Question 4

The command to execute the program until we hit a breakpoint is **dc**.

```
[0×00400a30]> dc
hit breakpoint at: 400b5f
```

Question 5

The value of local_ch when its corresponding movl instruction is called **1**.

```
|           0×00400b4e      4889e5          mov rbp, rsp
|           0×00400b51      c745f4010000.   mov dword [local_ch], 1
|           0×00400b58      c745f8060000.   mov dword [local_8h], 6
```

```
[0×00400b5f]> px @ rbp-0×c
- offset -       0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0×7ffe38b772d4  0100 0000 0600 0000 0000 0000 4018 4000  ..... ........@.@.
0×7ffe38b772e4  0000 0000 e910 4000 0000 0000 0000 0000  ....... ..@.......
0×7ffe38b772f4  0000 0000 0000 0000 0100 0000 0874 b738  ........ ..... .t.8
0×7ffe38b77304  fe7f 0000 4d0b 4000 0000 0000 0000 0000  .... ..M.@.........
0×7ffe38b77314  0000 0000 1700 0000 0100 0000 0000 0000  ..... ..... ........
0×7ffe38b77324  0000 0000 0000 0000 0200 0000 0000 0000  ................
0×7ffe38b77334  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffe38b77344  0000 0000 0000 0000 0000 0000 0004 4000  ..............@.
0×7ffe38b77354  0000 0000 f0c7 ad49 7553 2f67 e018 4000  ..... ... IuS/g..@.
0×7ffe38b77364  0000 0000 0000 0000 0000 0000 1890 6b00  ............ ..k.
0×7ffe38b77374  0000 0000 0000 0000 0000 0000 f0c7 cd9c  ................
0×7ffe38b77384  9b22 d398 f0c7 1958 7553 2f67 0000 0000  ."..... XuS/g....
0×7ffe38b77394  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffe38b773a4  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffe38b773b4  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffe38b773c4  0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Question 6

The value of eax when the imull instruction is called is **6**.

```
[0×00400b5f]> ds
[0×00400b5f]> dr
rax = 0×00000006
rbx = 0×00400400
rcx = 0×0044b9a0
rdx = 0×7ffd92660ff8
r8 = 0×01000000
r9 = 0×006bb8e0
r10 = 0×00000015
r11 = 0×00000000
r12 = 0×004018e0
r13 = 0×00000000
r14 = 0×006b9018
r15 = 0×00000000
rsi = 0×7ffd92660fe8
rdi = 0×00000001
rsp = 0×7ffd92660ec0
rbp = 0×7ffd92660ec0
rip = 0×00400b66
rflags = 0×00000246
orax = 0×ffffffffffffffff
```

Question 7

The value of local_4h before eax is set to 0 is **6**.

```
orax = 0×ffffffffffffffff
[0×00400b5f]> ds
[0×00400b5f]> px @ rbp-0×4
- offset -       0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0×7ffd92660ebc  0600 0000 4018 4000 0000 0000 e910 4000  ..... @.@.....  ..@.
0×7ffd92660ecc  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd92660edc  0100 0000 e80f 6692 fd7f 0000 4d0b 4000  ..... ..f.... ..M.@.
0×7ffd92660eec  0000 0000 0000 0000 0000 0000 1700 0000  ..............  .....
0×7ffd92660efc  0100 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd92660f0c  0200 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd92660f1c  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd92660f2c  0000 0000 0004 4000 0000 0000 992a 3e1d  ......@.......*>.
0×7ffd92660f3c  ed32 39dc e018 4000 0000 0000 0000 0000  .29 ... @.........
0×7ffd92660f4c  0000 0000 1890 6b00 0000 0000 0000 0000  ..... ..k.........
0×7ffd92660f5c  0000 0000 992a 1e30 a116 c223 992a 8a0c  .....*.0 ... #.* ..
0×7ffd92660f6c  ed32 39dc 0000 0000 0000 0000 0000 0000  .29..............
0×7ffd92660f7c  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd92660f8c  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd92660f9c  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd92660fac  0000 0000 0000 0000 0000 0000 0000 0000  ................
[0×00400b5f]> ds
```

Thought Process/Methodology:

Firstly, we did SSH into the terminal with the username elfmceager and the password adventofcyber. We used radare2 to enter debug mode to learn more about the file. We did this using r2 -d ./challenge1. When we were inside debug mode, we did a full analysis of the file using the aa command. Next we search for main that most executable programs have an entry point. We did this using the alf command and then filtered the results with grep. The final command will be alf | grep main. We took a closer look at it using the print disassembly function(pdf). The command will be pdf @ main. Then we can see there were the Assembly instructions for the main function. Next, we need to find the value of local_ch when its corresponding movl instruction. We used db to put a breakpoint at address 0x00400b5f so the command will become db 0x00400b5f. Then we ran pdf @ main again to see the breakpoint set correctly. Next, we ran the program using dc and we received a message that we had stopped at the breakpoint set. Then, we analyse the contents of local_ch with the command px @ rbp-0xc. The value we saw is 1. Next, to find the value of eax at the first imul instruction, we used ds to move the next line and then used dr to see the value of eax. We saw that the value is 6. Lastly, to find the value of local_4h before the eax is set to 0, we used ds again to move the next line and then we ran px @ rbp-0x4. The value we saw is 6.

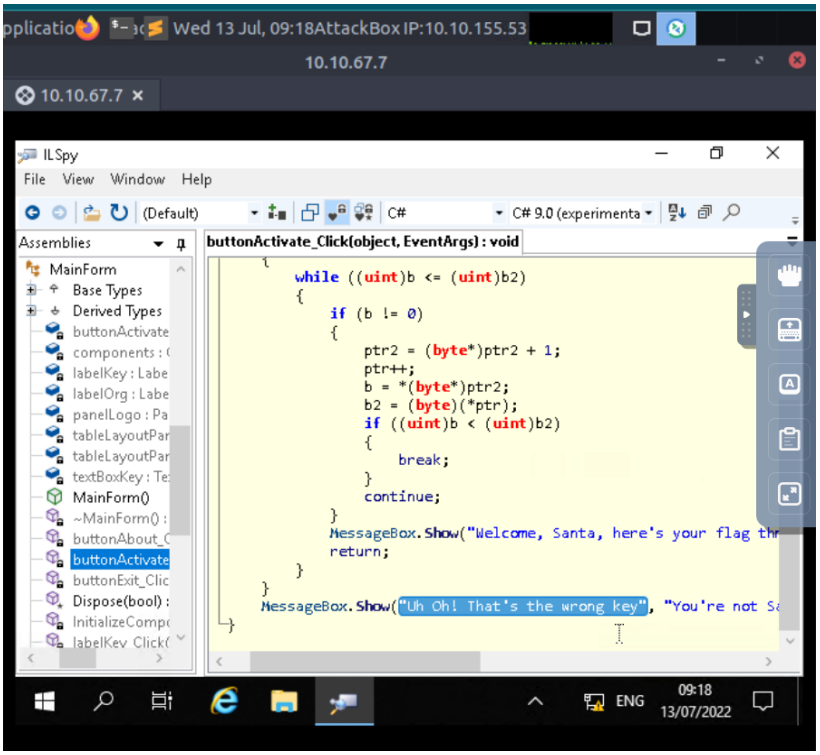
Day 18: Reverse Engineering – The Bits of Christmas

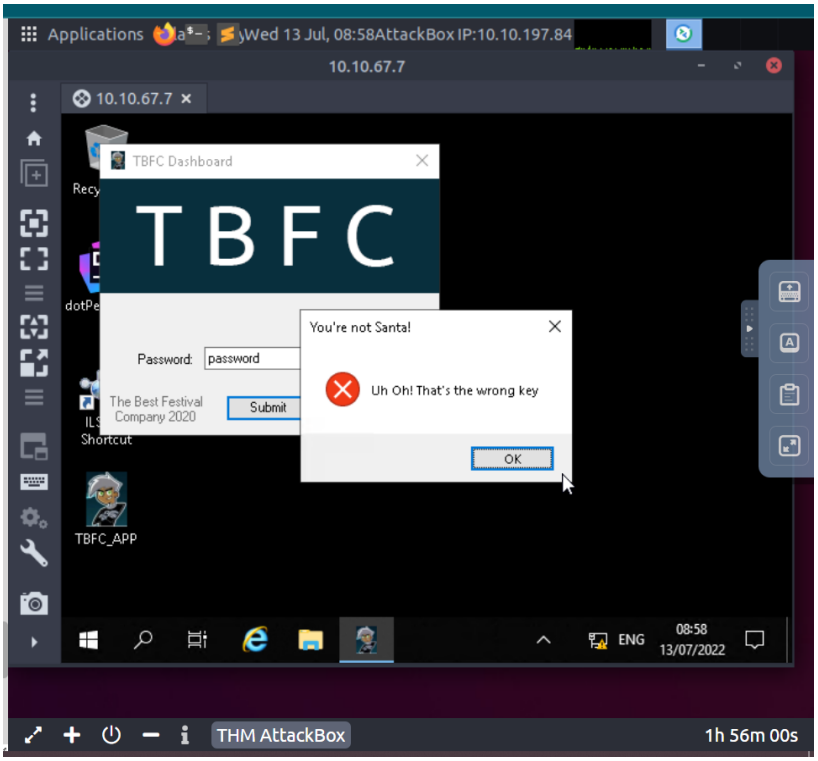
Tools used: Kali Linux, Firefox, ILSpy, Remmina

Solution/walkthrough:

Question 1

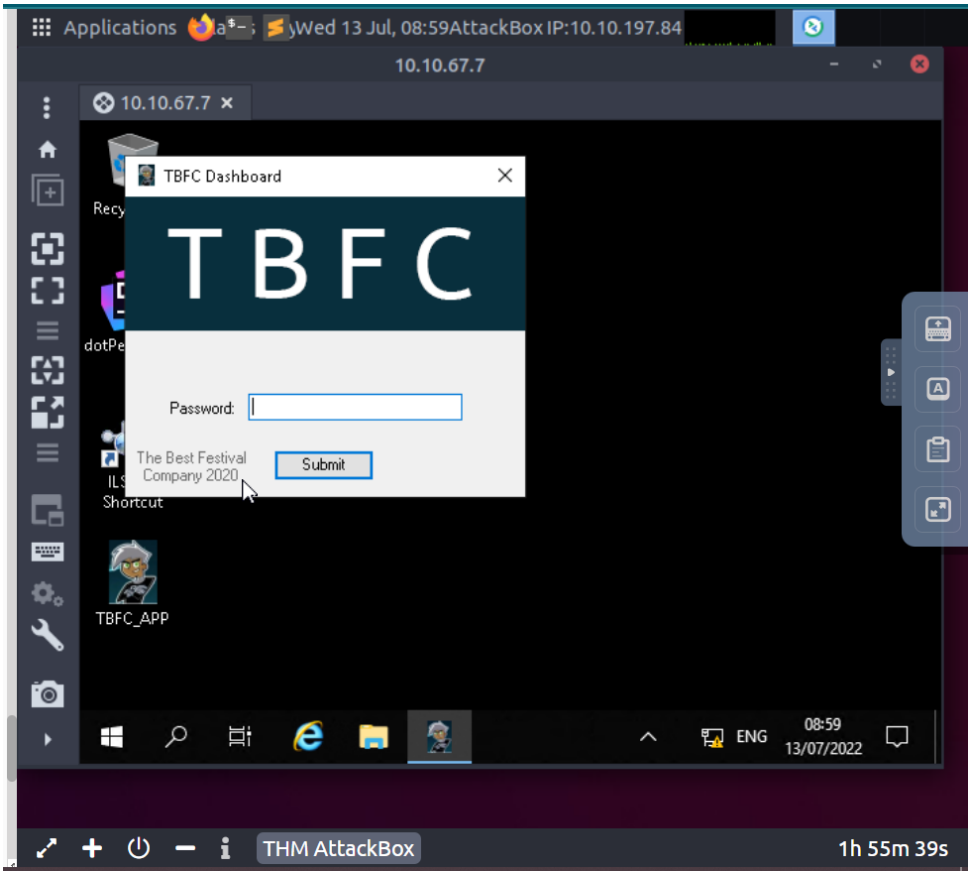
The message that shows up if we enter the wrong password for TBFC_APP is **Uh Oh! That's the wrong key .**





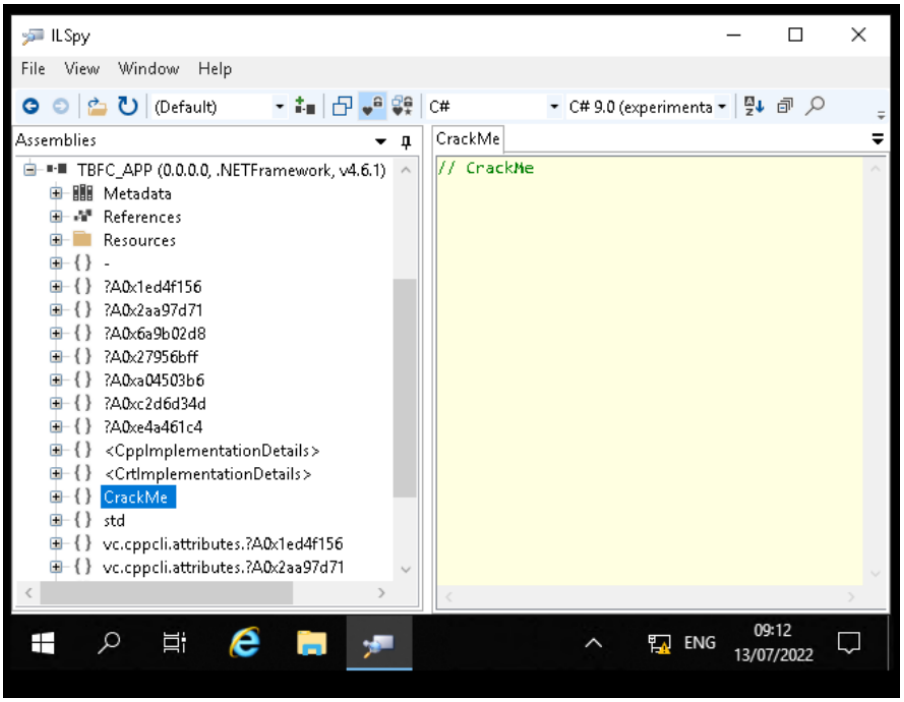
Question 2

TBFC stands for **The Best Festival Company**.



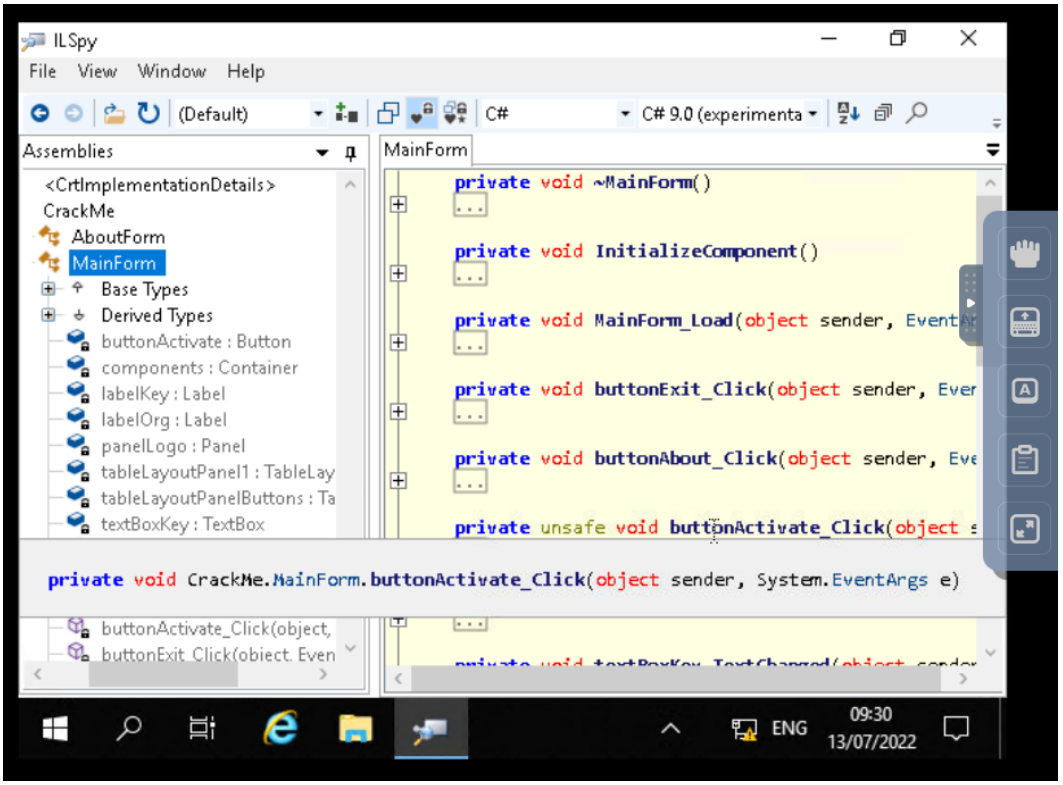
Question 3

Decompile the TBFC_APP with ILSpy. The module that catches our attention is **CrackMe**.



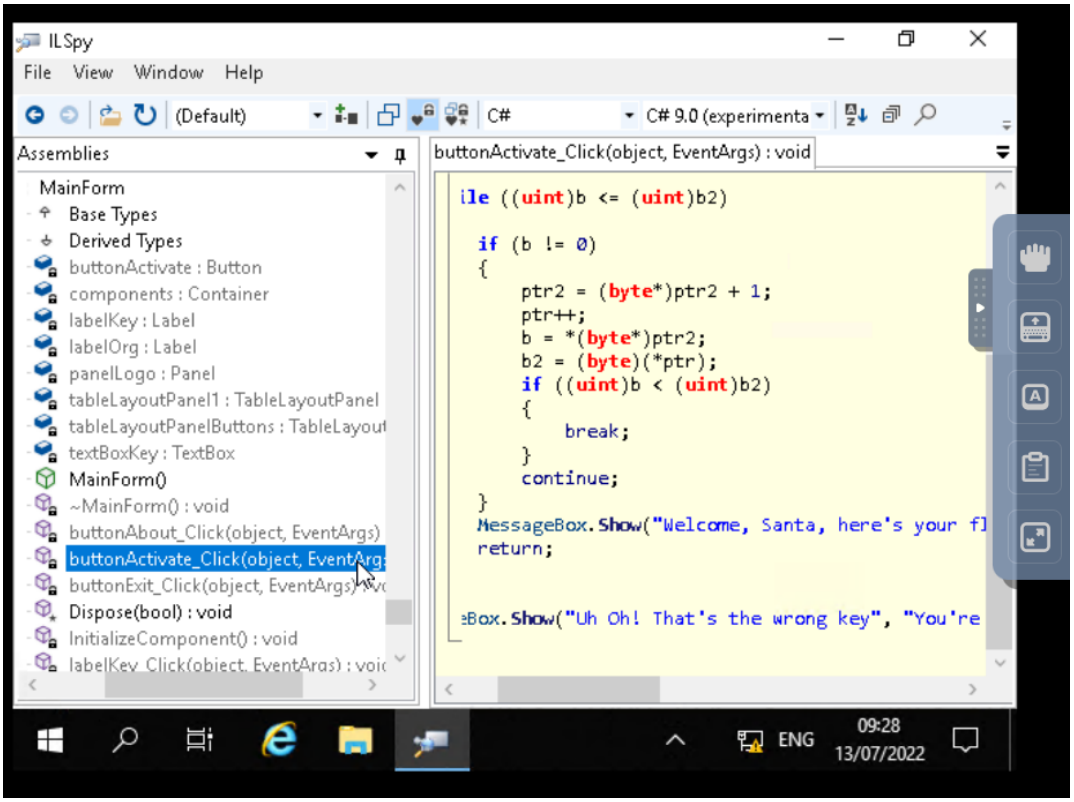
Question 4

Within the module, there are two forms. **MainForm** contains the information we are looking for.



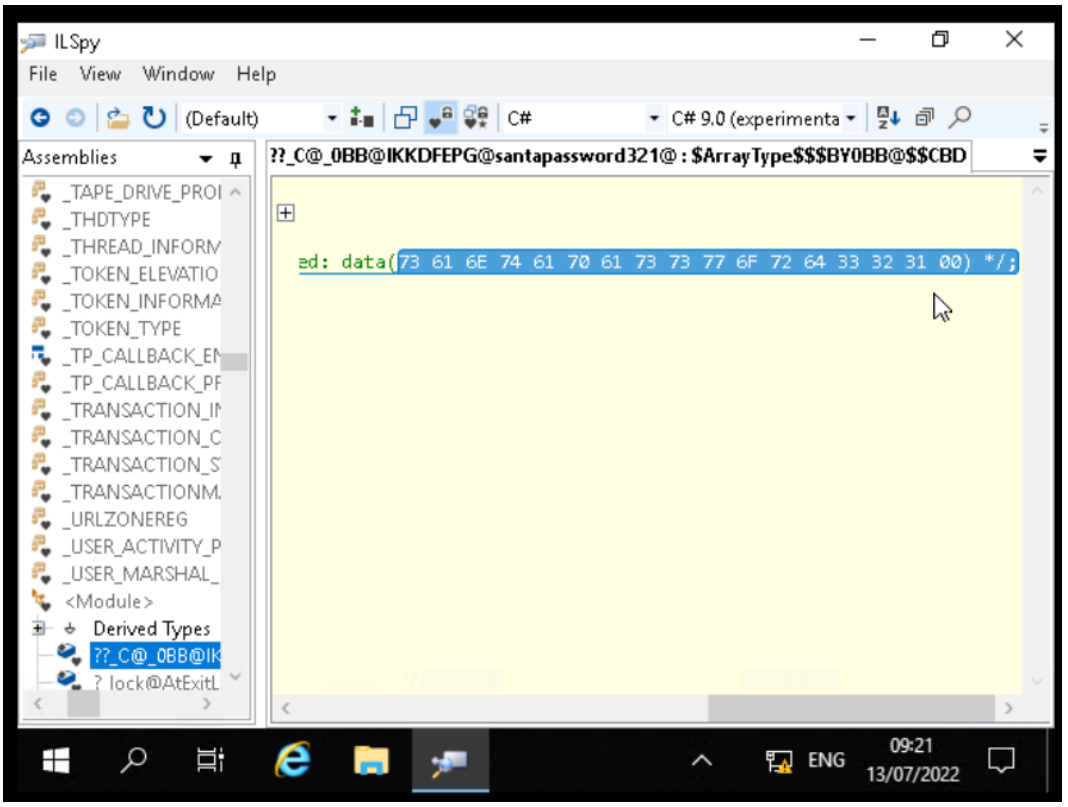
Question 5

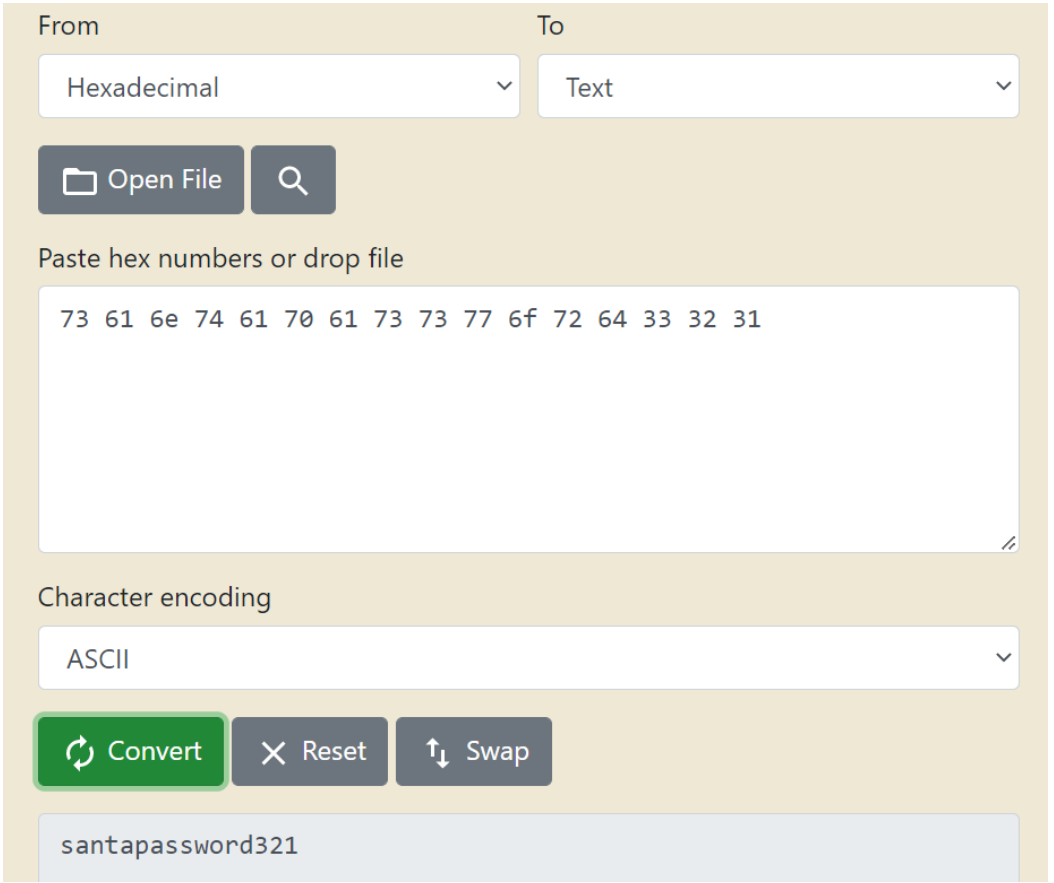
The form from Q4 will contain the information we are seeking is **buttonActivate_Click**.



Question 6

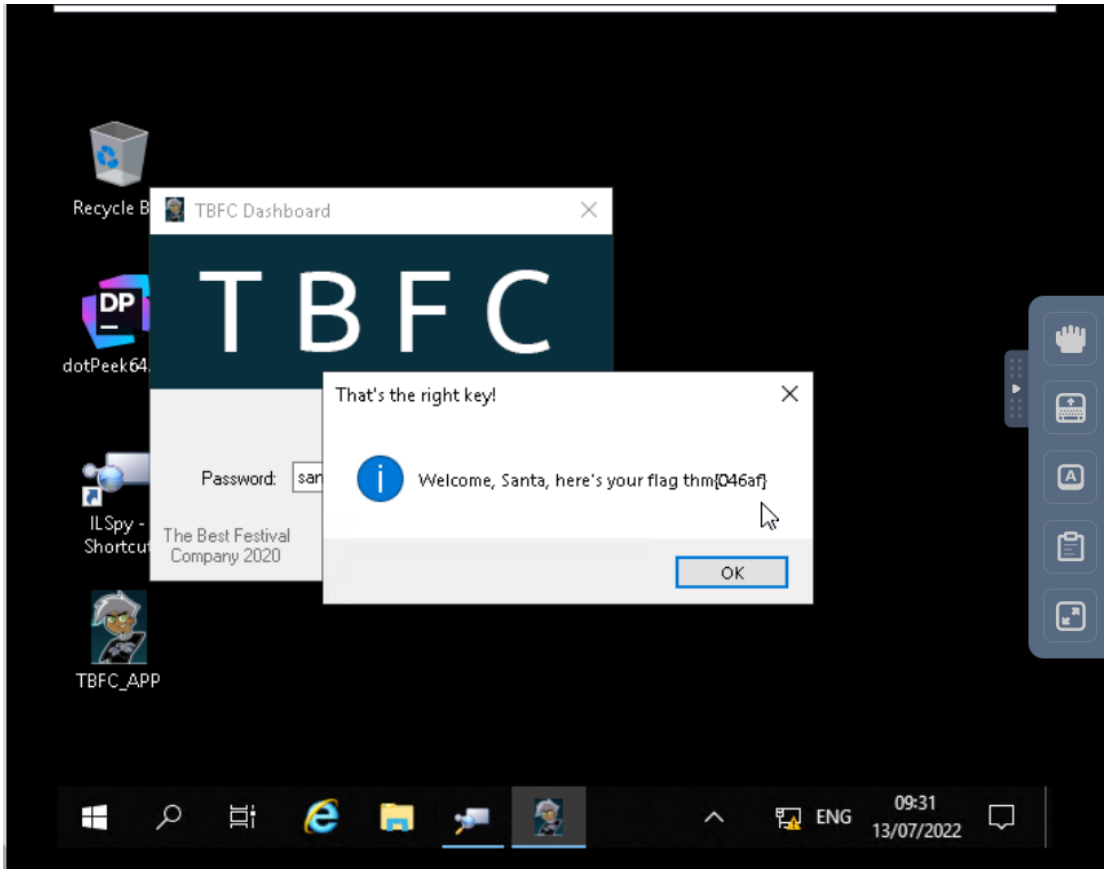
Santa's password is **santapassword321**.





Question 7

Now that we've retrieved this password, try to login. The flag is **thm{046af}** .



Thought Process/Methodology:

For Day 18, we opened up Remmina on the TryHackMe AttackBox and we connected to an IP address that has been given which is 10.10.67.7. After that, we key in the username and password for RDP authentication credentials and we are now connected. Then, we opened TBFC_APP and we saw TBFC which stands for The Best Festival Company 2020. On the side note, we opened ILSpy and had TBFC_APP loaded. Once we open it, we are asked to find Santa's password. After a bit of searching, there are components named 'CrackMe' which is a very weird name. We expand the thing, and inside it there is the MainForm() where the main function is. On the TBFC dashboard picture, the login mechanism is using a button to trigger the action and is also declared in the main function. After going through all the 'button' things, we arrived at 'buttonActivate_Click' and luckily the flag is hardcoded on the source code. In the same file, there is also Santa's password written on the code. After we click on that code, it displays the password that is formatted in hexadecimal. We use a converter to decode that. The password that we are expecting is santapassword123. Still on the same file, we can see the message that shows up if we enter the wrong password for TBFC_APP is Uh Oh! That's the wrong key. Now that we have retrieved the password, we try to login and the flag is revealed which is thm{046af}.

Day 19: Web Exploitation – The Naughty or Nice List

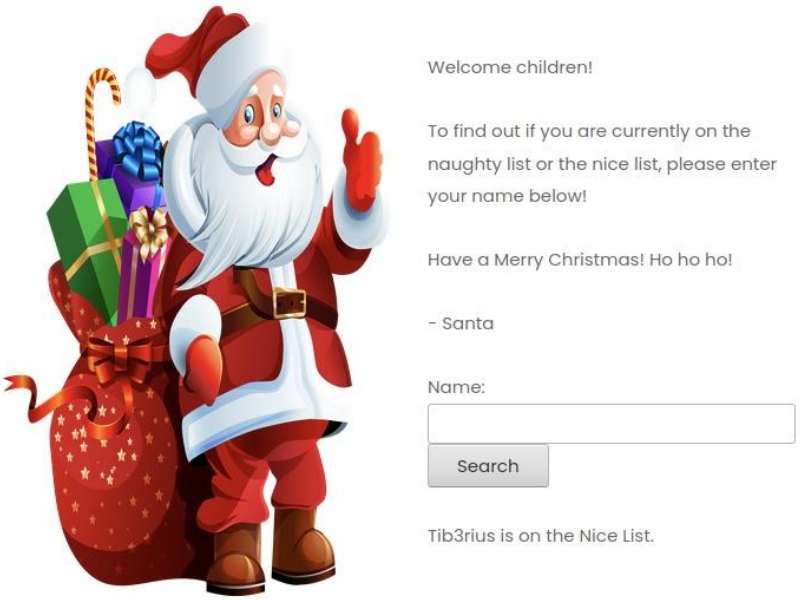
Tools used: Firefox, Sublime Text, Offline CyberChef

Solution/walkthrough:

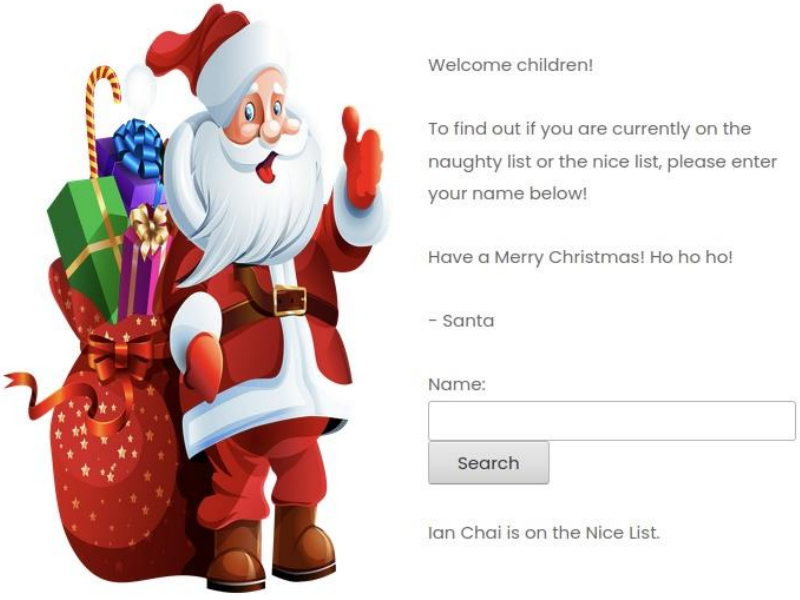
Question 1

Which list is this person on?

Tib3rius - Nice



Ian Chai - Nice



YP - Nice

Welcome children!

To find out if you are currently on the naughty list or the nice list, please enter your name below!

Have a Merry Christmas! Ho ho ho!

- Santa

Name:

Search

YP is on the Nice List.

Timothy - Naughty

Welcome children!

To find out if you are currently on the naughty list or the nice list, please enter your name below!

Have a Merry Christmas! Ho ho ho!

- Santa

Name:

Search

Timothy is on the Naughty List.

Kanes - Naughty

Welcome children!

To find out if you are currently on the naughty list or the nice list, please enter your name below!

Have a Merry Christmas! Ho ho ho!

- Santa

Name:

Search

Kanes is on the Naughty List.

JJ - Naughty

Welcome children!

To find out if you are currently on the naughty list or the nice list, please enter your name below!

Have a Merry Christmas! Ho ho ho!

- Santa

Name:

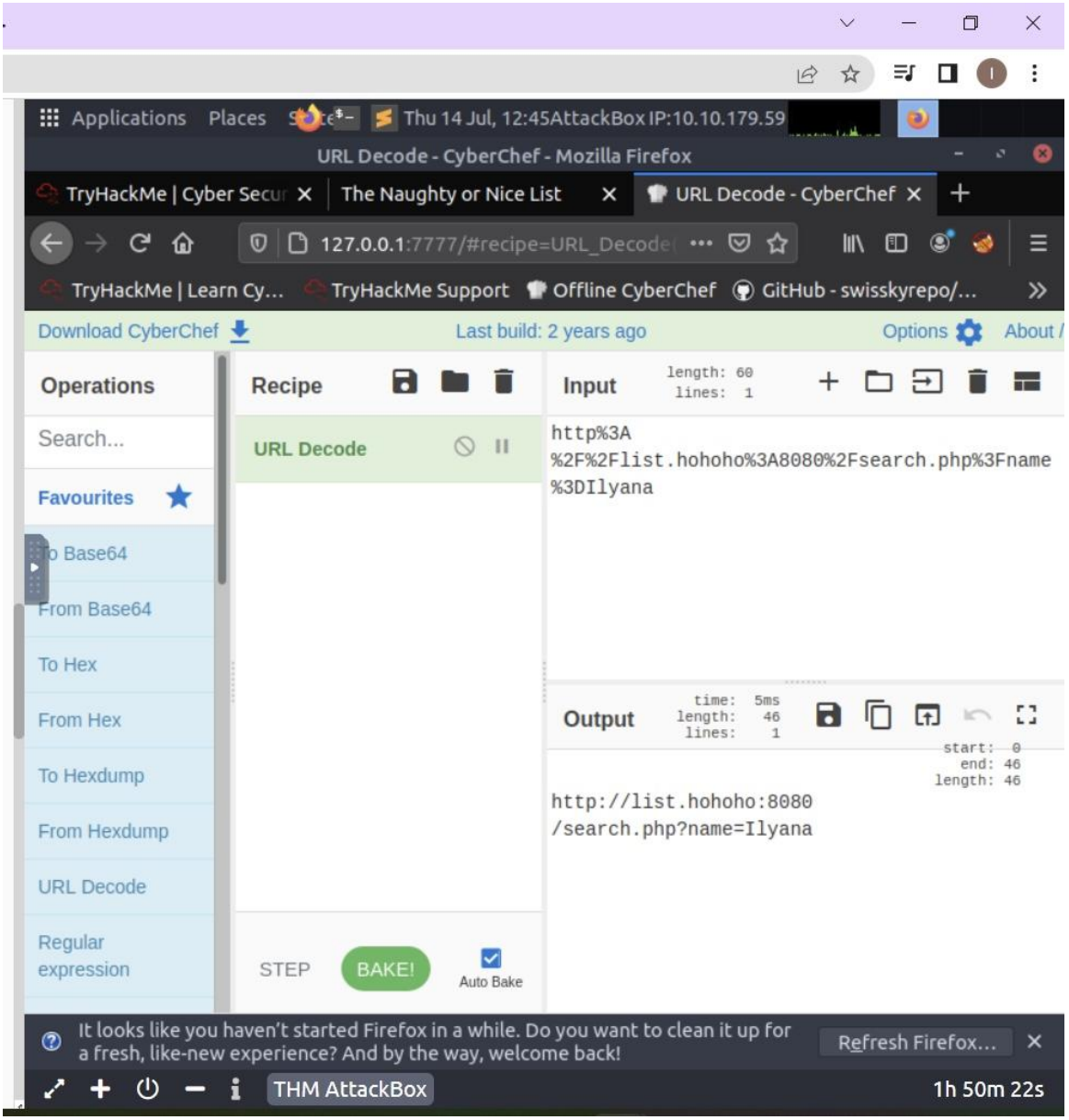
Search

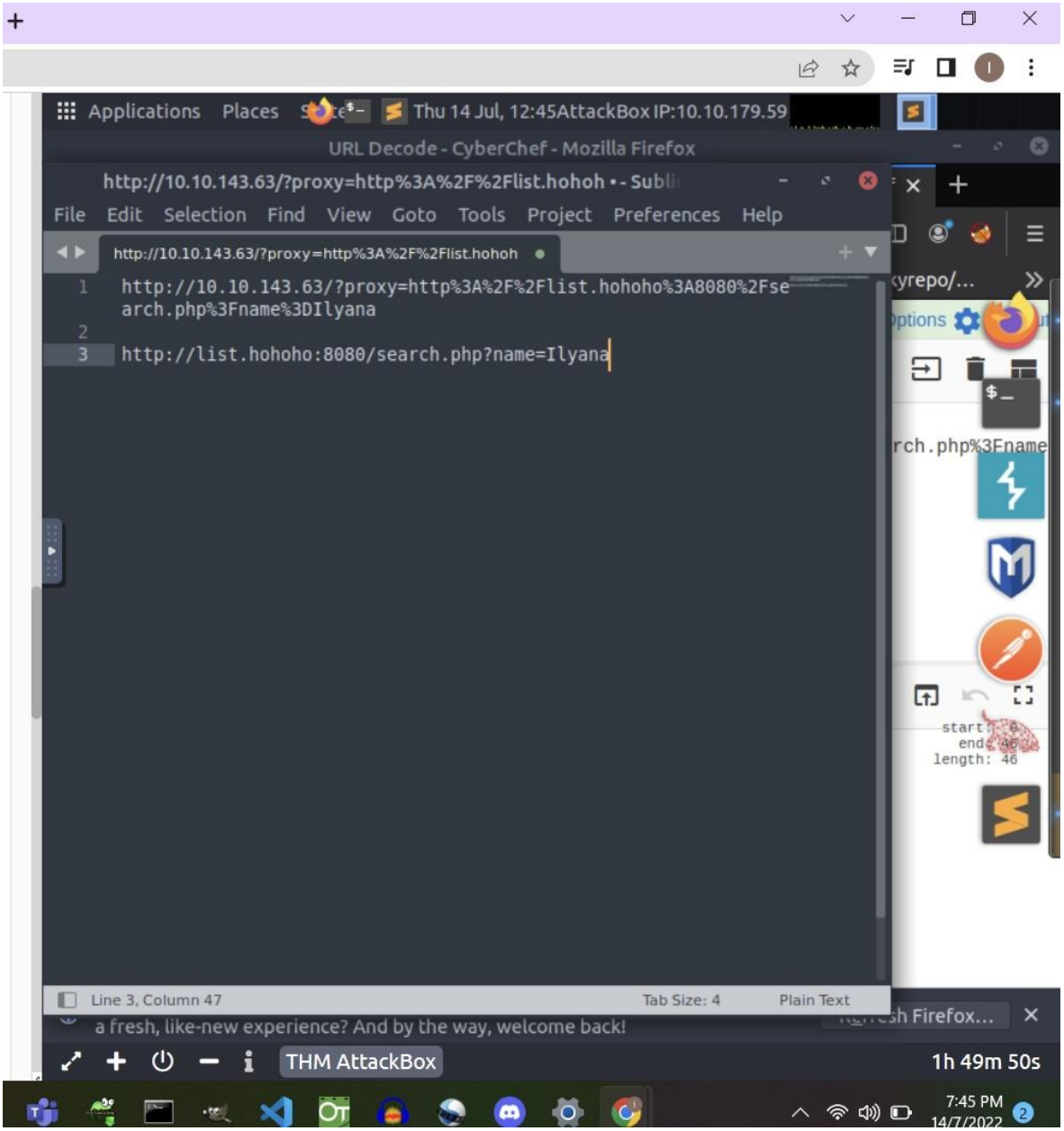
JJ is on the Naughty List.

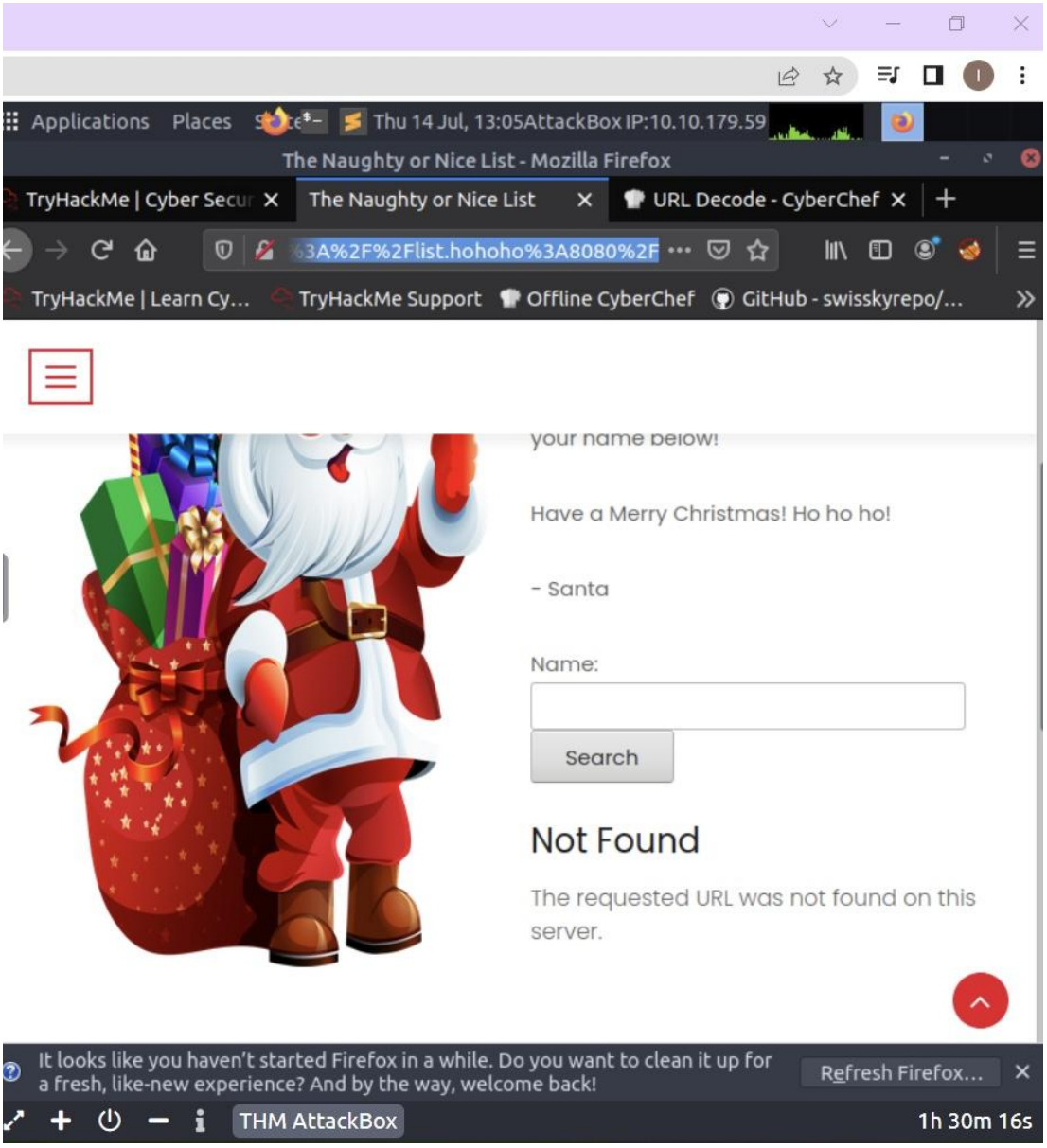
<u>Question 2</u>

What is displayed on the page when you use "/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F"?

Not Found. The requested URL was not found on this server.


Applications
Places
Thu 14 Jul, 12:45
AttackBox IP:10.10.179.59
URL Decode - CyberChef - Mozilla Firefox
TryHackMe | Cyber Secur
The Naughty or Nice List
URL Decode - CyberChef
127.0.0.1:7777/#recipe=URL_Decode(
TryHackMe | Learn Cy...
TryHackMe Support
Offline CyberChef
GitHub - swisskyrepo/...
Download CyberChef
Last build: 2 years ago
Options
About /
Operations
Search...
Favourites
To Base64
From Base64
To Hex
From Hex
To Hexdump
From Hexdump
URL Decode
Regular expression
Recipe
URL Decode
STEP
BAKE!
Auto Bake
Input
length: 60
lines: 1
http%3A%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%3DIlyana
Output
time: 5ms
length: 46
lines: 1
start: 0
end: 46
length: 46
http://list.hohoho:8080/search.php?name=Ilyana
It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!
Refresh Firefox...
THM AttackBox
1h 50m 22s

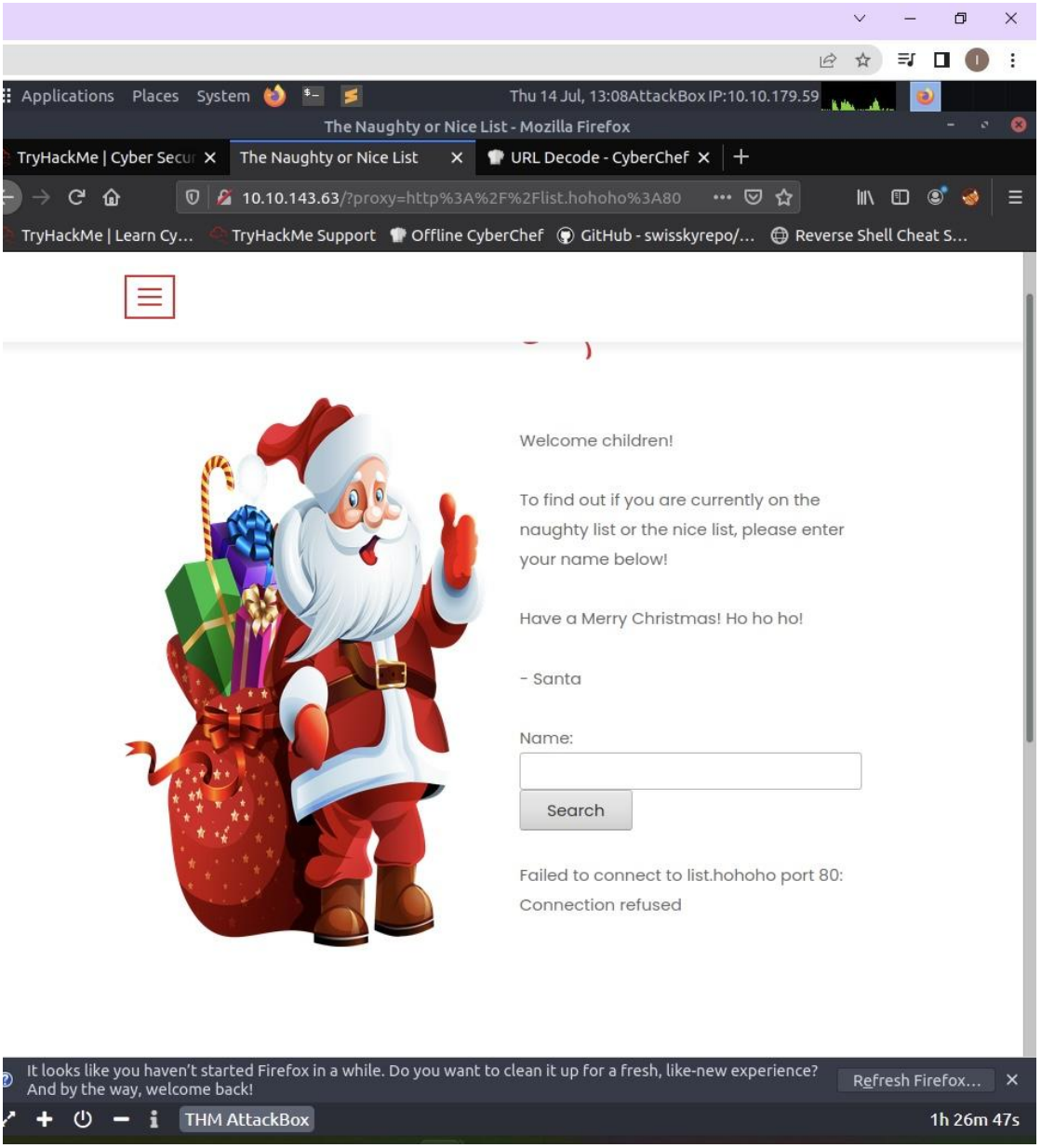

Applications Places
Thu 14 Jul, 12:45
AttackBox IP:10.10.179.59
URL Decode - CyberChef - Mozilla Firefox
http://10.10.143.63/?proxy=http%3A%2F%2Flist.hohoh • - Subli
File Edit Selection Find View Goto Tools Project Preferences Help
http://10.10.143.63/?proxy=http%3A%2F%2Flist.hohoh
1 http://10.10.143.63/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2Fse
arch.php%3Fname%3DIlyana
2
3 http://list.hohoho:8080/search.php?name=Ilyana
Line 3, Column 47
Tab Size: 4
Plain Text
a fresh, like-new experience? And by the way, welcome back!
THM AttackBox
1h 49m 50s
7:45 PM
14/7/2022



Question 3

What is displayed on the page when you use "/?proxy=http%3A%2F%2Flist.hohoho%3A80"?

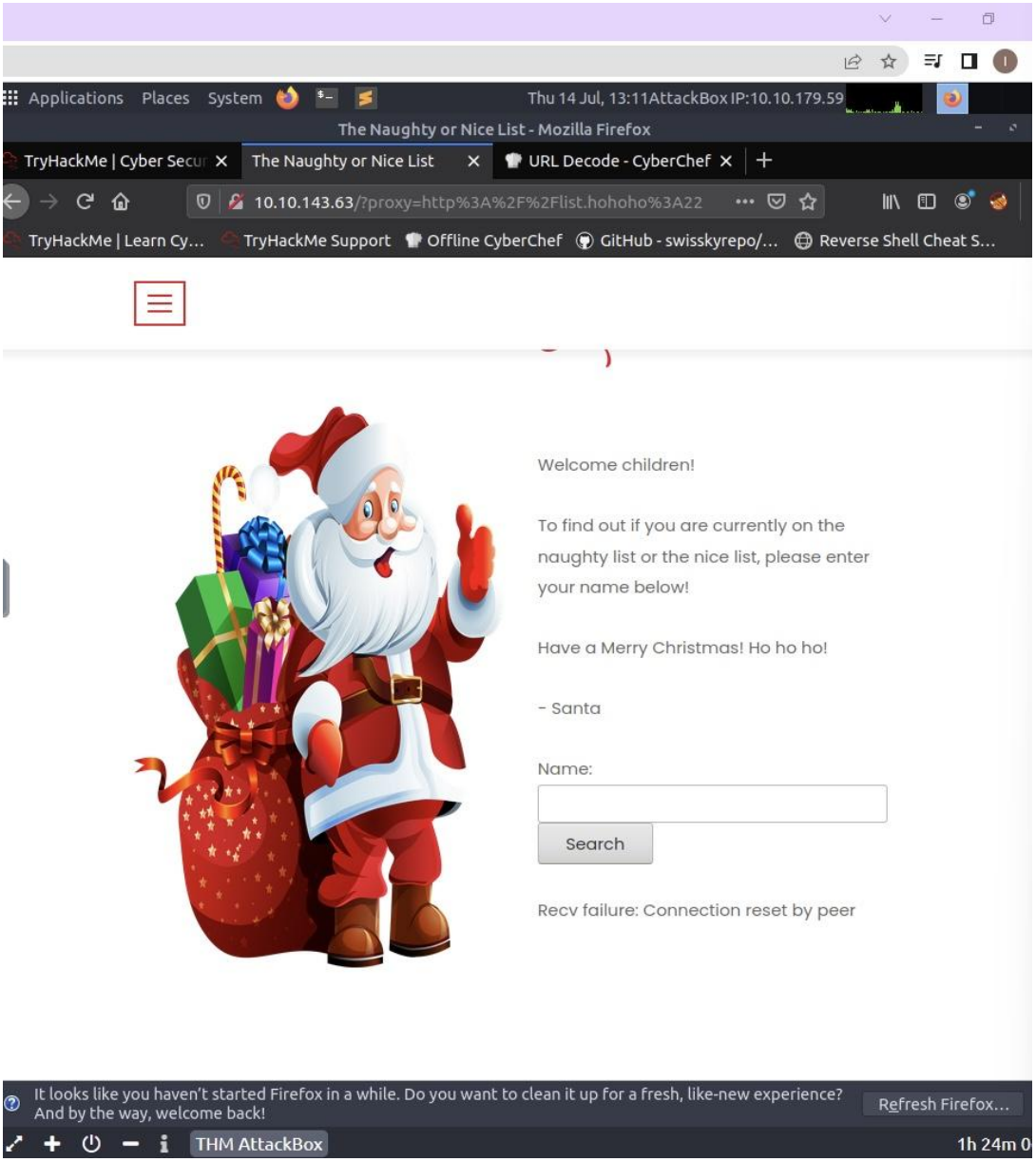
Failed to connect to list.hohoho port 80: Connection refused



Question 4

What is displayed on the page when you use "/?proxy=http%3A%2F%2Flist.hohoho%3A22"?

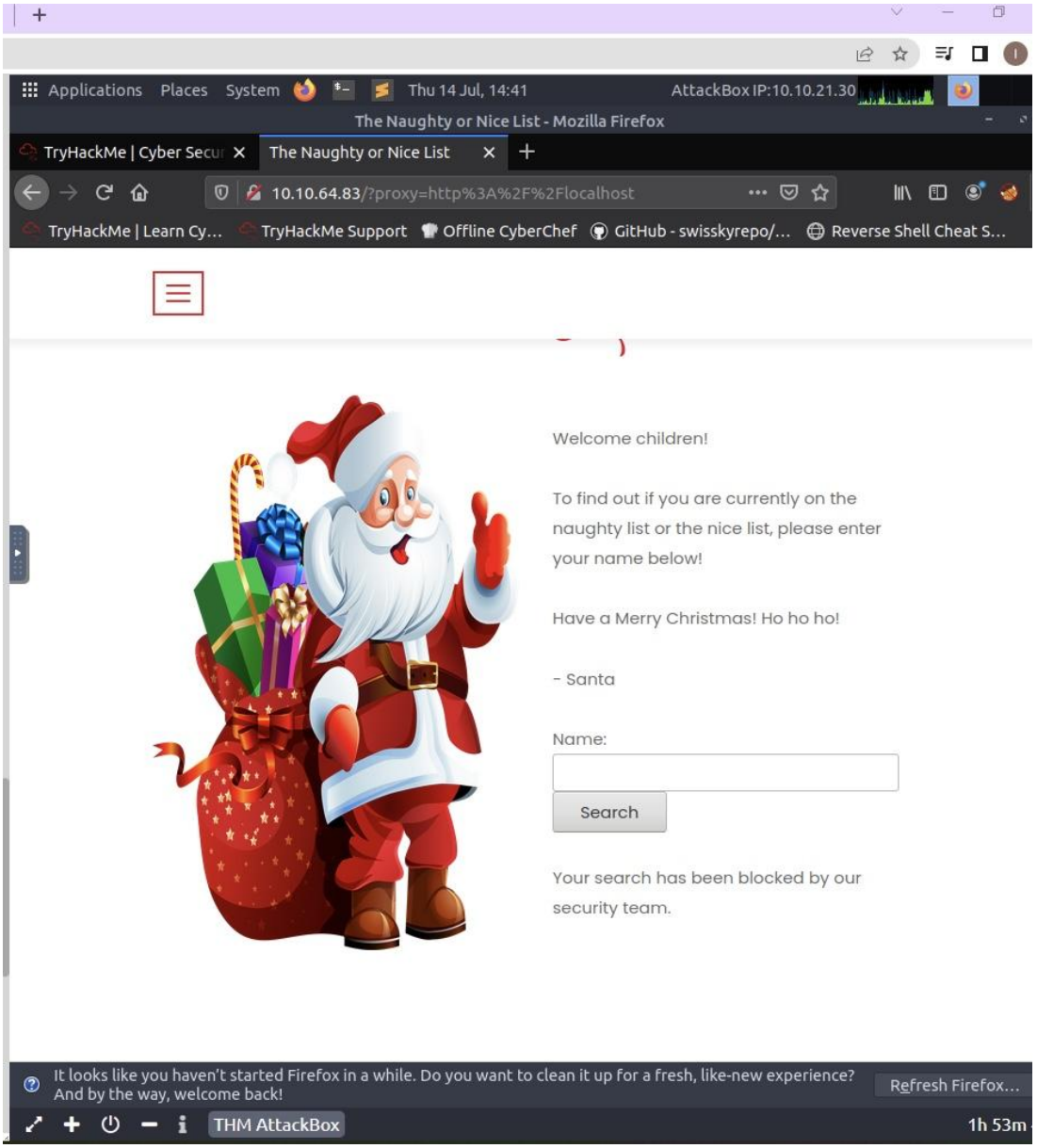
Recv failure: Connection reset by peer



Question 5

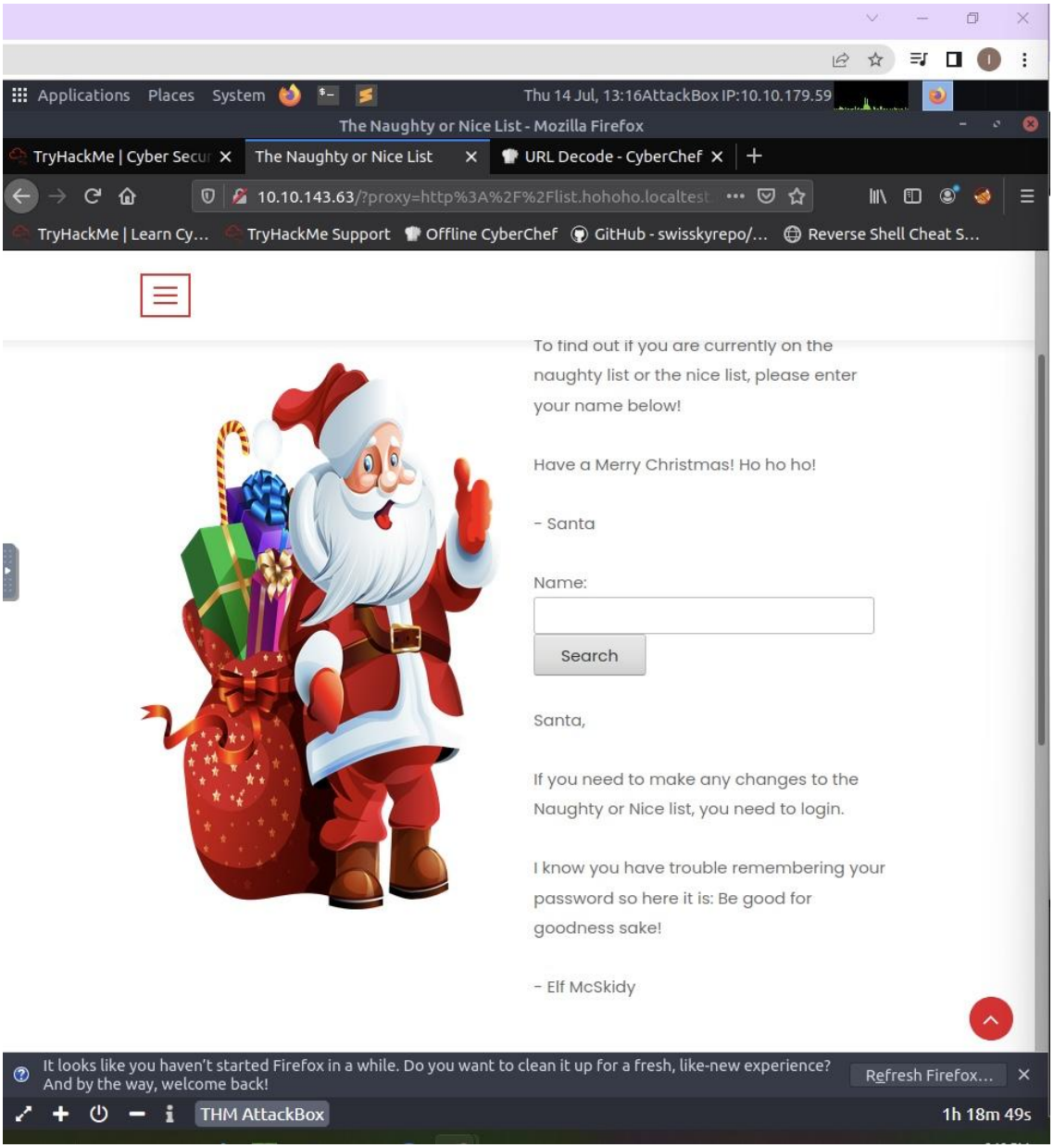
What is displayed on the page when you use "/?proxy=http%3A%2F%2Flocalhost"?

Your search has been blocked by our security team.



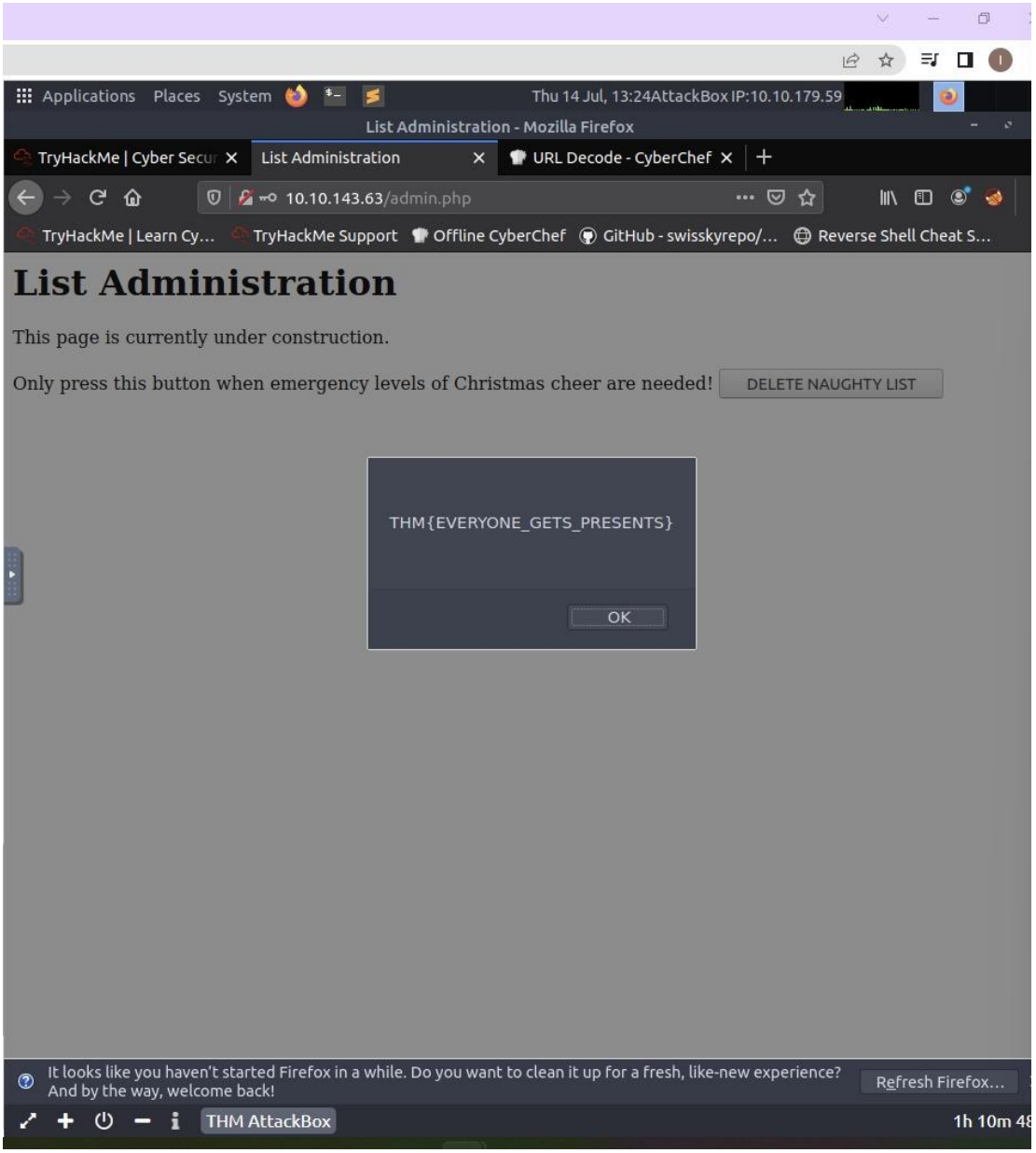
Question 6

Santa's password is **Be good for goodness sake!**



Question 7

The challenge flag is **THM{EVERYONE_GETS_PRESENTS}**



Thought Process/Methodology:

Firstly, once the VM is deployed, we connect to the web application by entering the IP address in firefox. Upon entry into the website, we can enter a name and check if that person is on the Naughty or Nice List. There is also an admin panel on the homepage. So we tried searching for the names **Tib3rius, Ian Chai, YP, Timothy, Kanes and JJ**. Therefore, we get to find out whether their names are on the Naughty List or the Nice List. Then, we enter the name Ilyana, and we notice that when we search for the name, there is a proxy parameter in the URL. Next, we open Sublime Text and paste the URL. Next, we copy http%3A%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%3DIlyana and paste it into Offline Cyberchef. After we decode the URL, we get the decoded output which is http://list.hohoho:8080/search.php?name=Ilyana and we paste it back on Sublime Text. Since "list.hohoho" is not a valid hostname, it seems to be communicating with a backend service that takes the user's name and returns whether they are Naughty or Nice. We can take advantage of this by performing a SSRF (Server-Side Request Forgery) attack. Next, we try to fetch the root of the same

site. We enter the url http://10.10.143.63/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F and we can see **“Not Found. The requested URL was not found on this server.”** This looks like a generic 404 message, which we were able to make the server request the modified URL and it returned the response. Next, we try changing the port number from 8080 to 80 and we can see there is “**Failed to connect to list.hohoho port 80: Connection refused”** messages are displayed which implies that port 80 is not open on list.hohoho. After that, we try to change the port number to 22 and now there is **"Recv failure: Connection reset by peer"** which implies that port 22 is open but was unable to understand what was sent. Next, we access services running locally on the server by replacing the list.hohoho hostname with "localhost" and we can see there is **“Your search has been blocked by our security team”** messages are displayed because the developer has added a check to ensure that the hostname provided begins with "list.hohoho" and any hostnames that do not will be blocked. We discovered that only the domain list.hohoho works, but perhaps we can try some different subdomains, which is localtest.me. The full URL is http://10.10.143.63/?proxy=http%3A%2F%2Flist.hohoho.localtest.me. This URL appears to respond with a message from Elf McSkidy containing a password to log into the admin which is **“Be good for goodness sake!”** We guessed the username and found out that the username is Santa and we logged in. Then, we delete the naughty list and the flag **THM{EVERYONE_GETS_PRESENTS}** appears.

Day 20: Blue Teaming – PowershELlF to the rescue

Tools used: Terminal Emulator

Solution/walkthrough:

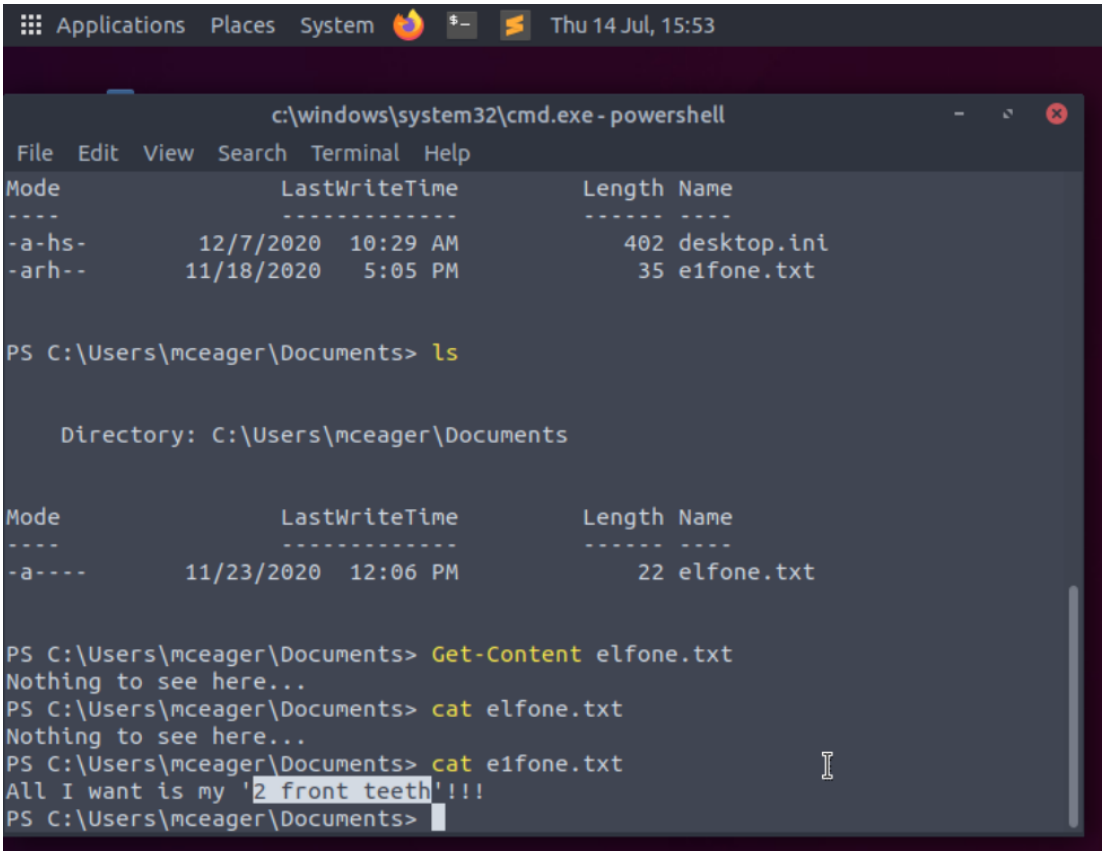
Question 1

The parameter -l is used for the login name.

```
-l login_name
        Specifies the user to log in as on the remote machine.  This also
        may be specified on a per-host basis in the configuration file.
```

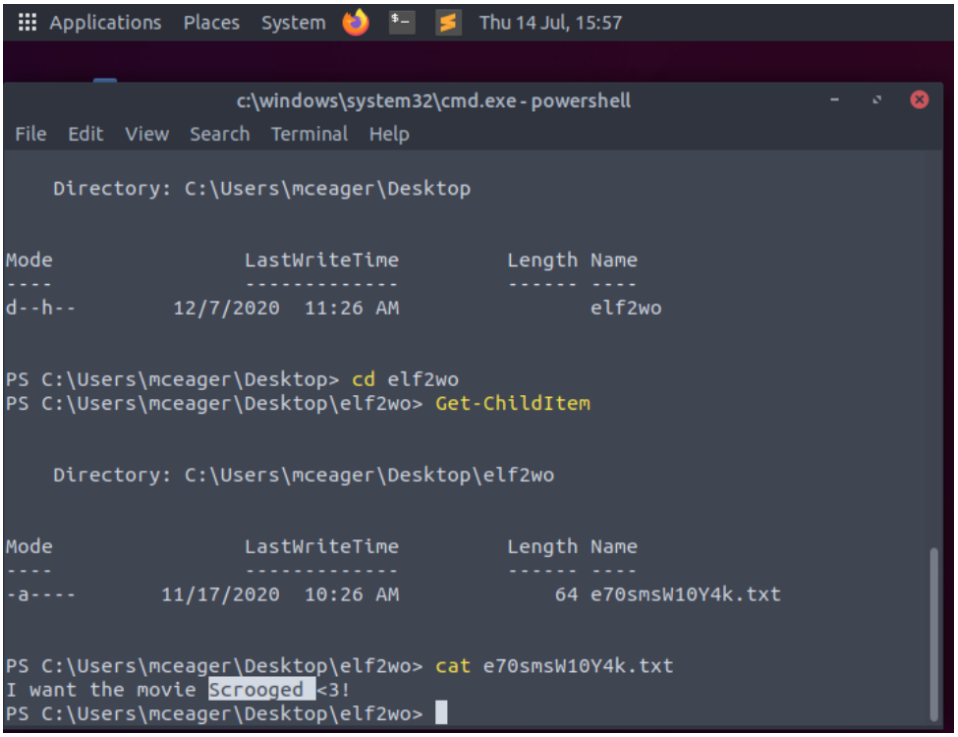
Question 2

Search for the first hidden elf file within the Documents folder. Read the contents of this file. What does Elf 1 want **2 front teeth**.



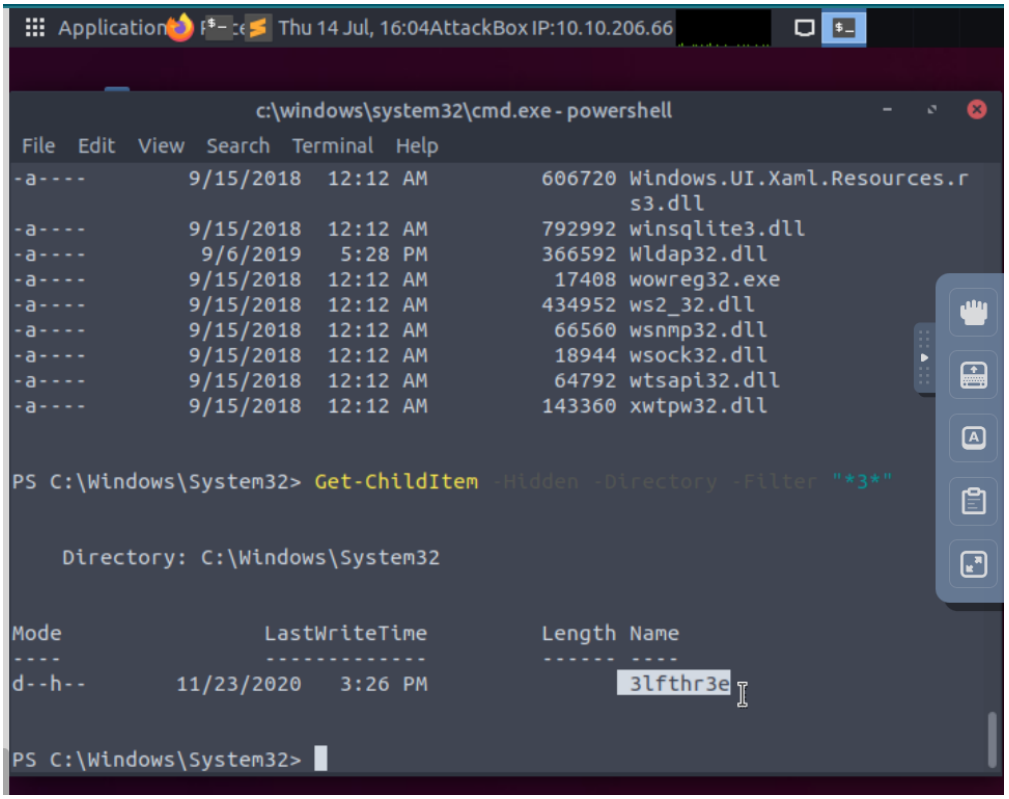
Question 3

The name of that movie that Elf 2 wants is **Scrooged.**



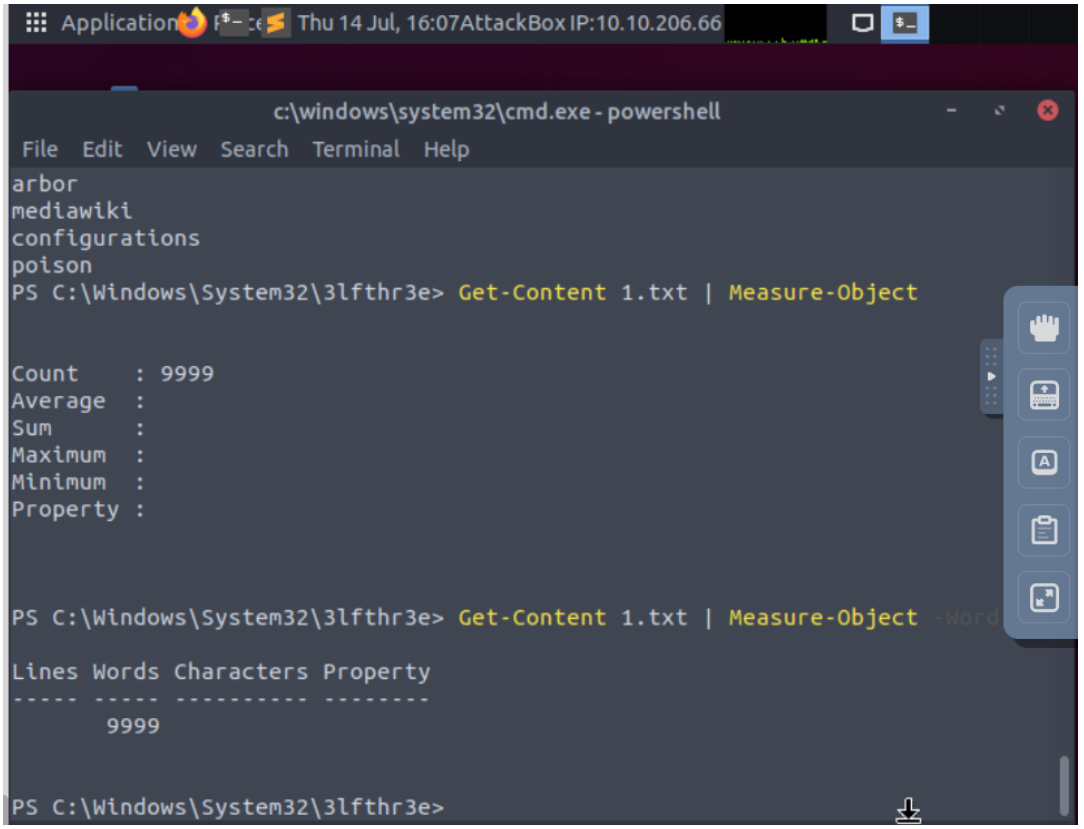
Question 4

Search the Windows directory for a hidden folder that contains files for Elf 3. The name of the hidden folder is **3lfthr3e.**



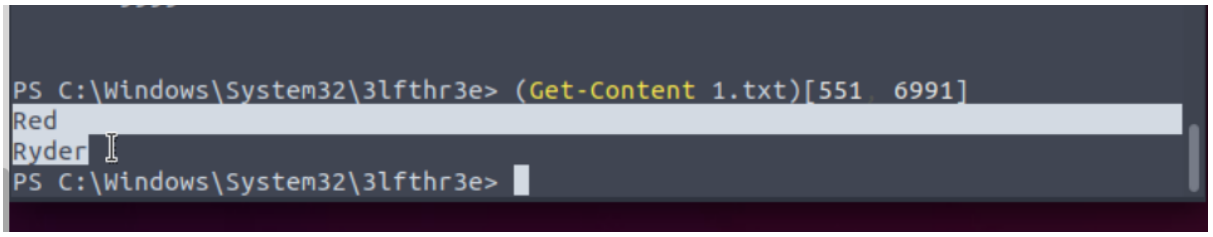
Question 5

The first file contains **9999** words.



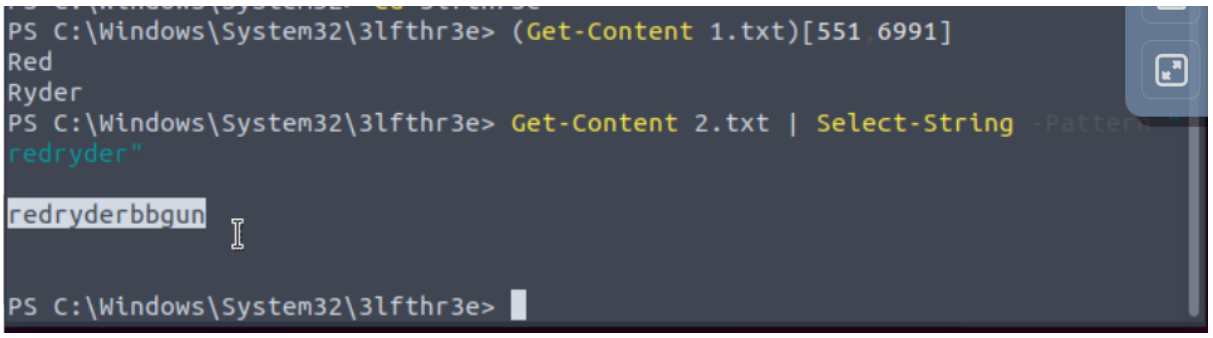
Question 6

The 2 words at index 551 and 6991 in the first file are **Red Ryder.**



Question 7

This is only half the answer. Search in the 2nd file for the phrase from the previous question to get the full answer. Elf 3 want **redryderbbgun.**



Thought Process/Methodology:

For Day 20, we used PowerShell over SSH to connect to the remote machine. The command that we use to run to connect to the remote machine is ssh -l mceager 10.10.206.66. For Question1, parameter -l used for the login name. When prompted, enter the password r0ckStar! and once we have logged in successfully, we will begin to launch PowerShell and navigate to the Documents folder. To access the 'Documents' directory we can use powershell command - Set-Location Documents. After that, we can list the directory contents with command Get-ChildItem. Based on Question 2, we need to find the hidden elf file so in order to display only the hidden files, we can use the additional flag Get-ChildItem -Hidden -File. As a result, it shows us "e1fone.txt". To display the content of a file, in powershell we use command Get-Content e1fone.txt and it shows that he wants his 2 front teeth. Next, same as before, we navigated into the Desktop directory and listed all the contents inside it. But since the thing that we are looking for is a directory, we used another flag on our powershell command Get-ChildItem -Hidden -Directory. We navigated into the 'elf2wo' directory and listed the directory's content inside. There is a file 'e70smsW10Y4k.txt' in it. We see the file content using cat e70smsW10Y4k.txt. In the file, Elf 2 wants to watch a movie called "Scrooged". In John's walkthrough video, he is searching the file on C:\Windows\System32 and we are going to do the same. The command that we'll using is Get-ChildItem -Hidden -Filter '*3*'. The command will display us the hidden content of the directory where it shows 3lfthr3e. For Question 5, we navigated into the directory '3lfthr3e', and list the file inside. There are two files by the name of '1.txt' and '2.txt'. To count the all words from file 1.txt, we can use the Measure-Object with flag -Word which input piped from Get-Content. It shows that the first file contains 9999 words. To find the 2 words at index 551 and 6991 in the first file, the command that we use is (Get-Content 1.txt)[551, 6991] and the results are Red Ryder. For the last question, the command that we'll be using is Select-String -Pattern "redryder". It shows that Elf 3 want redryderbbgun.