PSP0201

Week 5

Writeup

Group Name: SendHelp

Members

ID	Name	Role
1211102757	Sri Raam	Leader
1211101615	Thanirmalai	Member
1211101662	Yap Tze Lam, Robbie	Member
1211101416	Keshaav A/L Thamil Selvam	Member

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

Solution/Walkthrough:

Tools: kali, nmap, python

Q1: What is the port number for the web server?

Answer: 80

1. Open the terminal using nmap -v(which give a more verbose output) and will find the port under http

```
root@ip-10-10-184-232:~# nmap -v 10.10.235.27

Starting Nmap 7.60 ( https://nmap.org ) at 2022-07-17 12:16 BST
Initiating ARP Ping Scan at 12:16
Scanning 10.10.235.27 [1 port]
Completed ARP Ping Scan at 12:16, 0.22s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:16
Completed Parallel DNS resolution of 1 host. at 12:16, 0.00s elapsed
Initiating SYN Stealth Scan at 12:16
Scanning ip-10-10-235-27.eu-west-1.compute.internal (10.10.235.27) [1000 ports]
Discovered open port 22/tcp on 10.10.235.27
Discovered open port 80/tcp on 10.10.235.27
Completed SYN Stealth Scan at 12:16, 1.45s elapsed (1000 total ports)
Nmap scan report for ip-10-10-235-27.eu-west-1.compute.internal (10.10.235.27)
Host is up (0.014s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

Q2: Without using enumerations tools such as Dirbuster, what is the directory for the API? (without the API key)

Answer: /api/

1. Open the web browser and input the ip of the machine following with the port number

2. This will now bring you to website, then view the page source

- Home
- Examples

View Source. Template not my own.

Santa's Tracking System

Are you an Elf that Santa has forgotten? Use this system to track Santa! Note: due to how many humans try to find where Santa is, the link is hidden on this webpage. You're going to have to manually click every single link. Or perhaps there is a way to find all the links as fast as a Python?

Important notice All deliiveries to Skidy for TryHackMe jumpers are to be stopped. That man has asked for 613 on the premise that they are the softest jumper in the world. Please, we need to share them out.

Category

- Lorem ipsum dolor sit amet
- Vestibulum errato isse
- Lorem ipsum dolor sit amet
- Aisia caisia
- Murphy's law
- Flimsy Lavenrock
- Maven Mousie Lavender

Category

- Labore et dolore magna aliqua
- Kanban airis sum eschelor
- Modular modern free
- The king of clubs
- The Discovery Dissipation

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Santa's Tracker</title>
        <link rel="shortcut icon" href="" type="image/x-icon">
        <link rel="stylesheet" href="bulma.css">
        <!-- Bulma Version 0.9.0-->
        <link rel="stylesheet" type="text/css" href="../css/hero.css">
         <!-- <link rel="stylesheet" href="https://unpkg.com/bulma-modal-fx/dist/css/modal-fx.min.css" /> -->
    </head>
    <body>
        <section class="hero is-info is-medium is-bold">
            <div class="hero-head">
                <nav class="navbar">
                    <div class="container">
                        <div class="navbar-brand">
                            <a class="navbar-item" href="../">
                                <img src="http://bulma.io/images/bulma-type-white.png" alt="Logo">
                            </a>
                            <span class="navbar-burger burger" data-target="navbarMenu">
                                <span></span>
                                <span></span>
                                <span></span>
                            </span>
                        </div>
                        <div id="navbarMenu" class="navbar-menu">
                            <div class="navbar-end">
                                <div class="tabs is-right">
                                    <ul>
                                        <li class="is-active"><a>Home</a></li>
                                        <li><a href="">Examples</a></li>
                                    </ul>
                                    <span class="navbar-item">
                                        <a class="button is-white is-outlined" href="https://github.com/BulmaTemplates/bulma-templates/blob/master/templates/hero.html">
                                            <span class="icon">
                                                <i class="fa fa-github"></i>
                                            </span>
                                            <span title="Hello from the other side">View Source. Template not my own.</span>
                                        </a>
                                    </span>
                                </div>
                            </div>
                        </div>
                    </div>
                </nav>
            </div>
            <div class="hero-body">
                <div class="container has-text-centered">
                    <h1 class="title">
                    Santa's Tracking System
                    </h1>
                    <h2 class="subtitle">
```

3. Under category you will find a list with all the href attribute showing “#” except for one, called “Modular modern free”

```
<li><a href="http://machine_ip/api/api_key">Modular modern free</a></li>
```

Q3: Where is Santa right now?

Answer: Winter Wonderland, Hyde Park, London

1. Using python write and run a code using the range function.

```
url = 'http://10.10.49.56:8000/api/'
```

2. Since we know the api number is somewhere within 0 to 100 and is an odd number we can set our range function as such.

```
for i in range(1,100,2):
        r = requests.get(url+str(i))
        if 'Error' not in r.text:
                print(i)
                print(r.text)
```

3. Then we run the program using python3 which will not only out Santa's location but also the api number for the final question.

```
57
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
```

Q4: Find out the correct API key. Remember, this is an odd number between 0-100. After too many attempts, Santa's Sled will block you.

Answer: 57

Thought Process/Methodology:

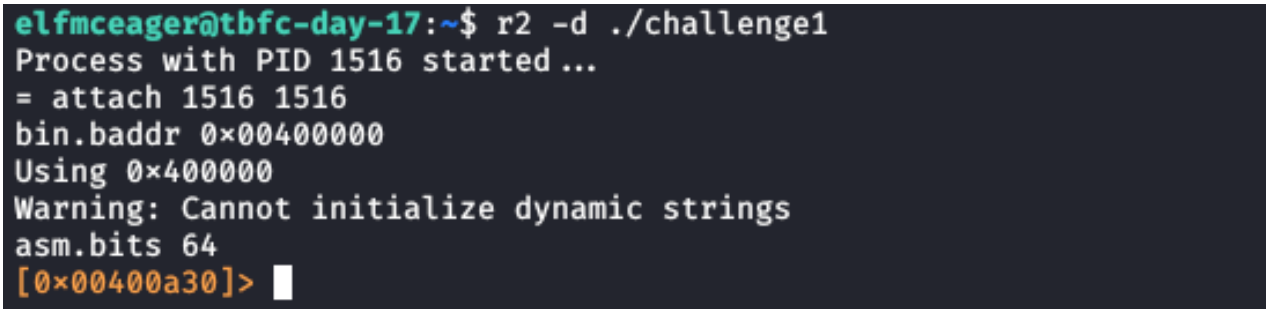
The process in which was used to solve question number 1 was by using nmap to scan our network to display the port we are using. As for the second question, we need to use the information that we have about the port number with the webpage link given to visit the website. There, by viewing the page source we can find the hidden link. From there, using python(which we learned on day 15) we can write some code which can get both the api number and Santa's location.

Day 17: [Reverse Engineering] ReverseELFneering

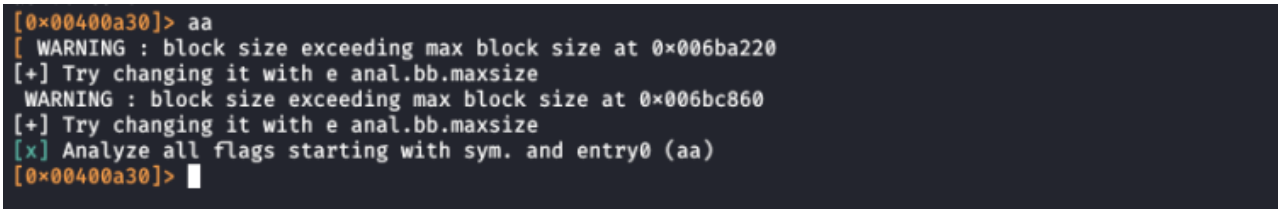
Solution/Walkthrough:

1. We first want to SSH into our target machine with the username **elfmceager** and the password **adventofcyber**.

2. Use **radare2** to enter debug mode so that we can learn more about our file. We can do this using **r2 -d ./challenge1**.



3. Once we are inside debugging mode, we can do a full analysis of the file using the **aa** command



4. Most executable programs have an entry point named **main**. Let's first search for this. We can do this using the **afl** command and then filter our results with **grep**. The final command will be **afl | grep main**

5. Let's take a closer look at it using the **print disassembly function(pdf)**. We can do this with the command **pdf @ main**.

6. What we see here are the **Assembly** instructions for the main function!.

Q: What is the value of local_ch when its corresponding movl instruction is called (first if multiple)?

Answer: 1

1. To put a breakpoint at this location we can use **db** and then pass in the address. Now when we can run **pdf @ main** again we should see the breakpoint set correctly.

```
[0×00400a30]> 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      b800000000     mov eax, 0
|           0×00400b6e      5d             pop rbp
\           0×00400b6f      c3             ret
[0×00400a30]>
```

2. Now use **dc** to run the program. We should receive a message that we have stopped at the breakpoint we set. Once we stop at this point, we can analyze the contents of **local_ch** with the command **px @ rbp-0xc**. We see the value is **1**!

```
[0×00400b5f]> px @ rbp-0×c
- offset -       0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0×7ffd6477e174  0100 0000 0600 0000 0000 0000 4018 4000  ..... ........@.@.
0×7ffd6477e184  0000 0000 e910 4000 0000 0000 0000 0000  ....  ..@.........
0×7ffd6477e194  0000 0000 0000 0000 0100 0000 a8e2 7764  ........ ... ..wd
0×7ffd6477e1a4  fd7f 0000 4d0b 4000 0000 0000 0000 0000  ....M.@.........
0×7ffd6477e1b4  0000 0000 0600 0000 5500 0000 5000 0000  ..... ... U ... P ...
0×7ffd6477e1c4  0400 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd6477e1d4  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd6477e1e4  0000 0000 0000 0000 0000 0000 0004 4000  ..............@.
0×7ffd6477e1f4  0000 0000 444f 2294 3f25 6c3b e018 4000  ....DO".?%l;..@.
0×7ffd6477e204  0000 0000 0000 0000 0000 0000 1890 6b00  ..............k.
0×7ffd6477e214  0000 0000 0000 0000 0000 0000 444f 8267  ............DO.g
0×7ffd6477e224  50ed 96c4 444f 9685 3f25 6c3b 0000 0000  P ... DO .. ?%l;....
0×7ffd6477e234  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd6477e244  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd6477e254  0000 0000 0000 0000 0000 0000 0000 0000  ................
0×7ffd6477e264  0000 0000 0000 0000 0000 0000 0000 0000  ................
[0×00400b5f]>
```

Q2: What is the value of eax when the imull instruction is called?
Answer: 6

1. We can simply use **ds** to move to the following line and then use **dr** to see the value of **eax**. We see that the value is 6.

```
[0x00400b5f]> dr
rax = 0x00000006
rbx = 0x00400400
rcx = 0x0044b9a0
rdx = 0x7ffd6477e2b8
r8 = 0x00000000
r9 = 0x00000003
r10 = 0x00000002
r11 = 0x00000000
r12 = 0x004018e0
r13 = 0x00000000
r14 = 0x006b9018
r15 = 0x00000000
rsi = 0x7ffd6477e2a8
rdi = 0x00000001
rsp = 0x7ffd6477e180
rbp = 0x7ffd6477e180
rip = 0x00400b66
rflags = 0x00000206
orax = 0xffffffffffffffff
[0x00400b5f]>
```

Q3: What is the value of local_4h before eax is set to 0?

Answer: 6

1. Use the **ds** command to get to that point and then check the value of **local_4h** with **px @ rbp-0x4**. The value here is **1**.

```
[0x00400b66]> px @ rbp-0x4
- offset -       0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0x7fffa8954a1c  0600 0000 4018 4000 0000 0000 e910 4000  ....@.@.......@.
0x7fffa8954a2c  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x7fffa8954a3c  0100 0000 484b 95a8 ff7f 0000 4d0b 4000  ....HK......M.@.
0x7fffa8954a4c  0000 0000 0000 0000 0000 0000 0600 0000  ..............
0x7fffa8954a5c  5500 0000 5000 0000 0400 0000 0000 0000  U...P.........
0x7fffa8954a6c  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x7fffa8954a7c  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x7fffa8954a8c  0000 0000 0004 4000 0000 0000 7f7d 6352  ......@.....}cR
0x7fffa8954a9c  88b9 f9ee e018 4000 0000 0000 0000 0000  ......@.........
0x7fffa8954aac  0000 0000 1890 6b00 0000 0000 0000 0000  ......k.........
0x7fffa8954abc  0000 0000 7f7d 83f6 22e8 0611 7f7d d743  .....}..".....}.C
0x7fffa8954acc  88b9 f9ee 0000 0000 0000 0000 0000 0000  ................
0x7fffa8954adc  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x7fffa8954aec  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x7fffa8954afc  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x7fffa8954b0c  0000 0000 0000 0000 0000 0000 0000 0000  ................
[0x00400b66]>
```

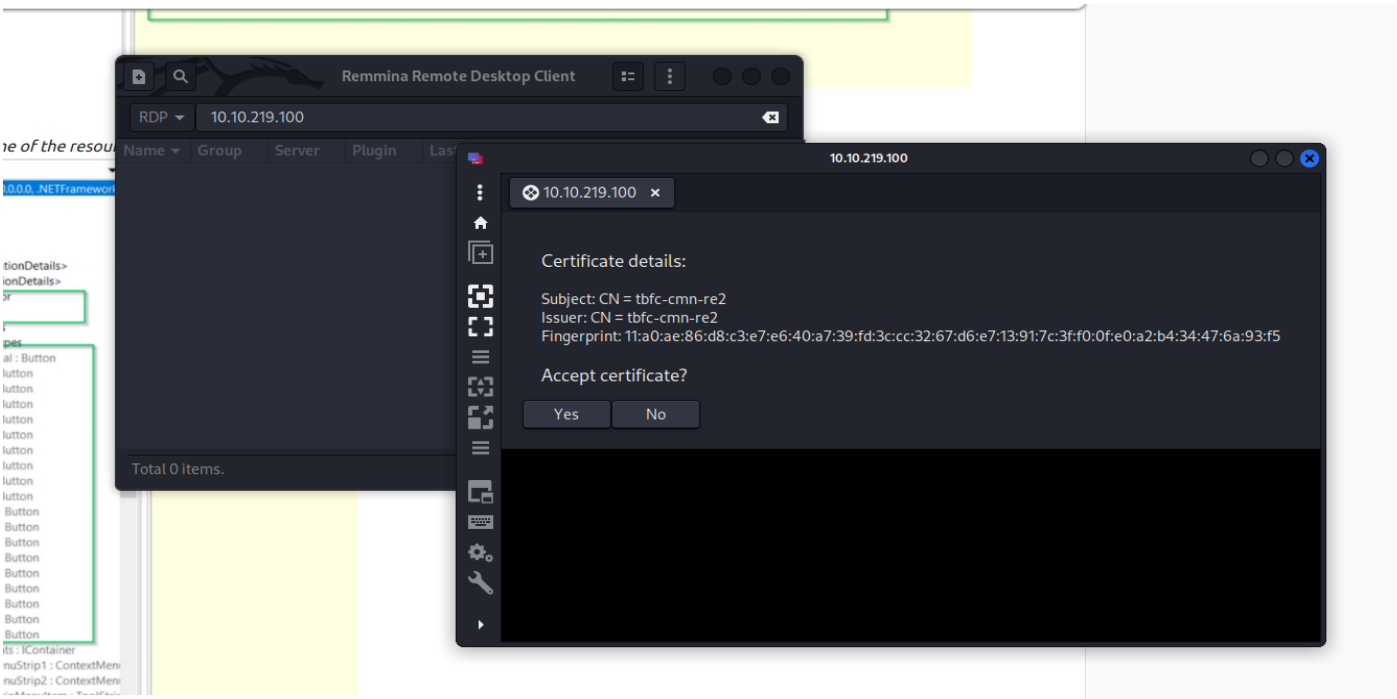
Day 18: [Reverse Engineering] The Bits of Christmas - Story
Tools: Linux, remmina, ILSpy, cyberchef

Solution/Walkthrough:

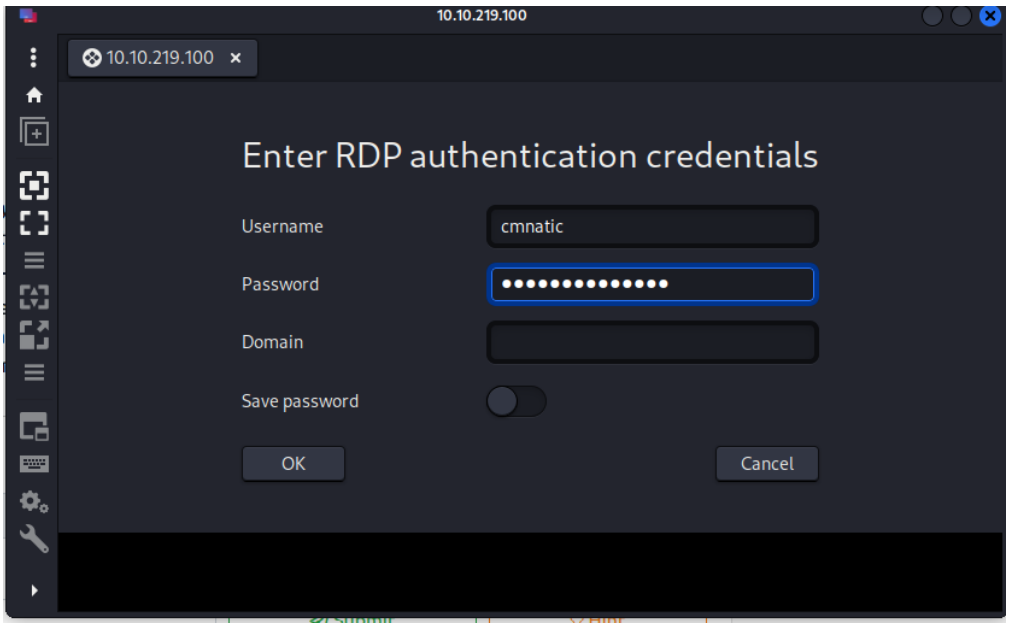
Q: Open the "TBFC_APP" application in ILspy and begin decompiling the code

Answer: No answer needed

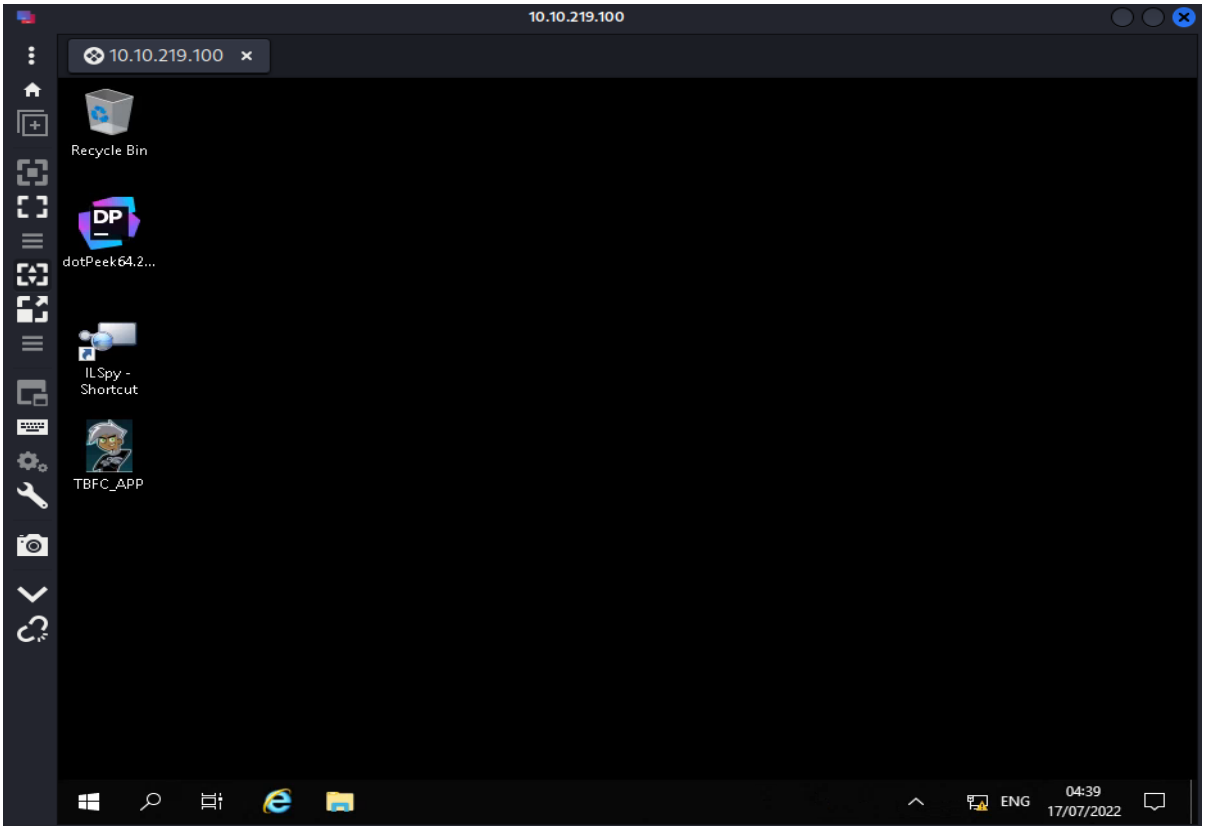
1. First we begin by installing the remmina and launching it.
2. Using the machine's ip address to log in with the Remote Desktop Protocol (RDP)

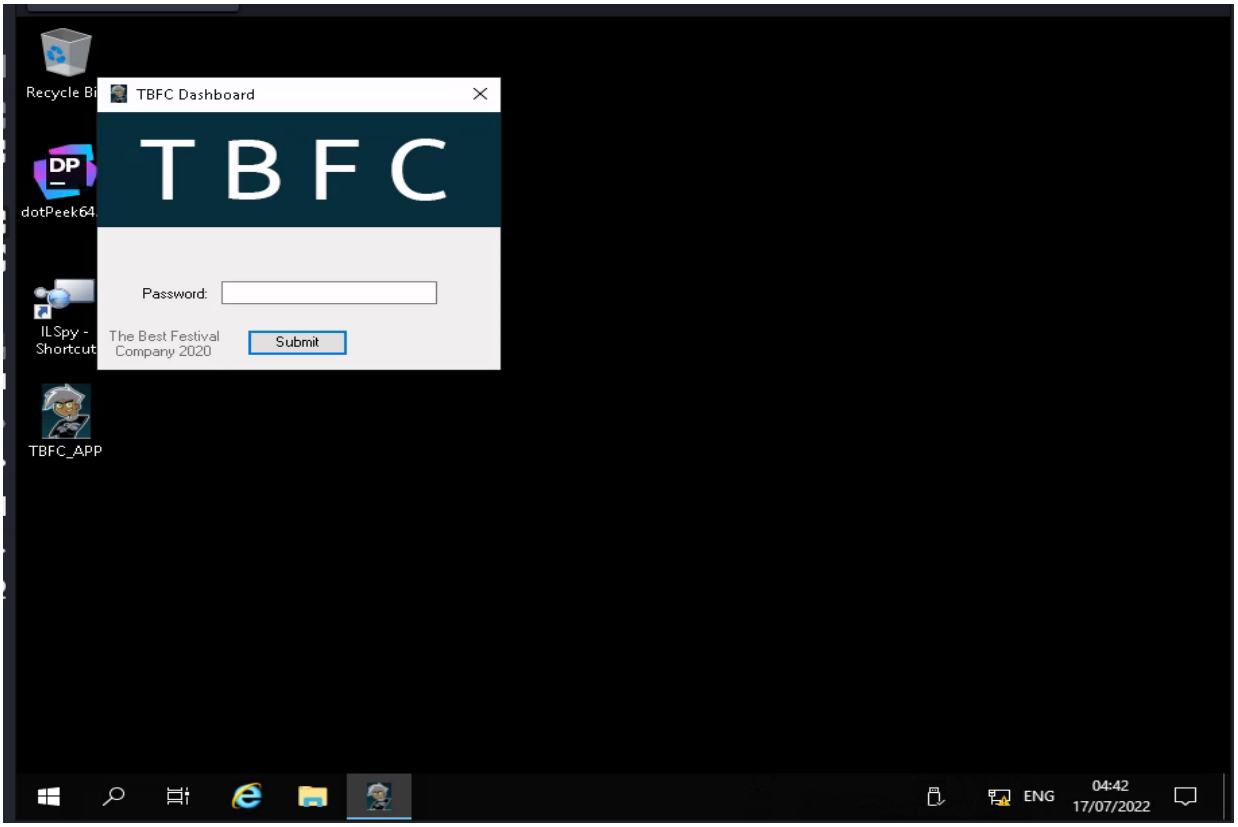


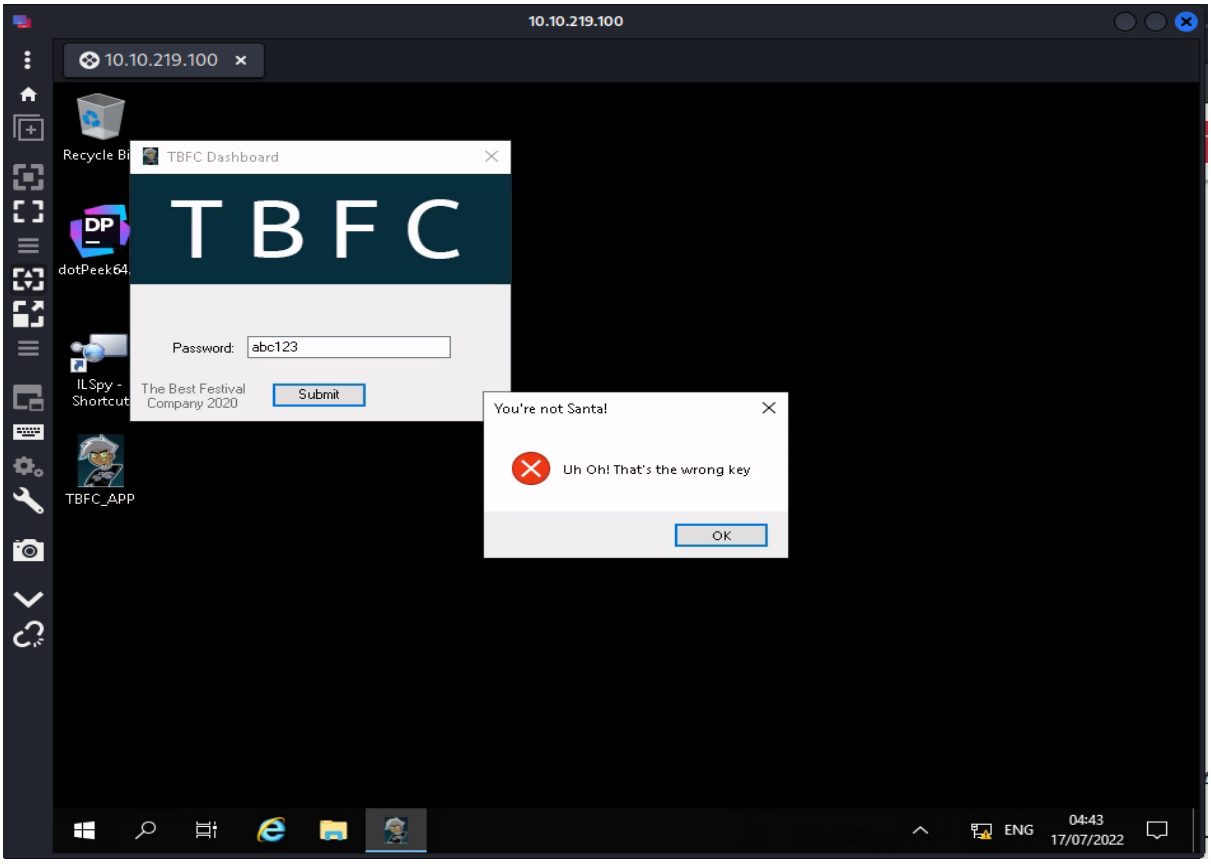
3. Enter the username and password that has been provided, then press OK.



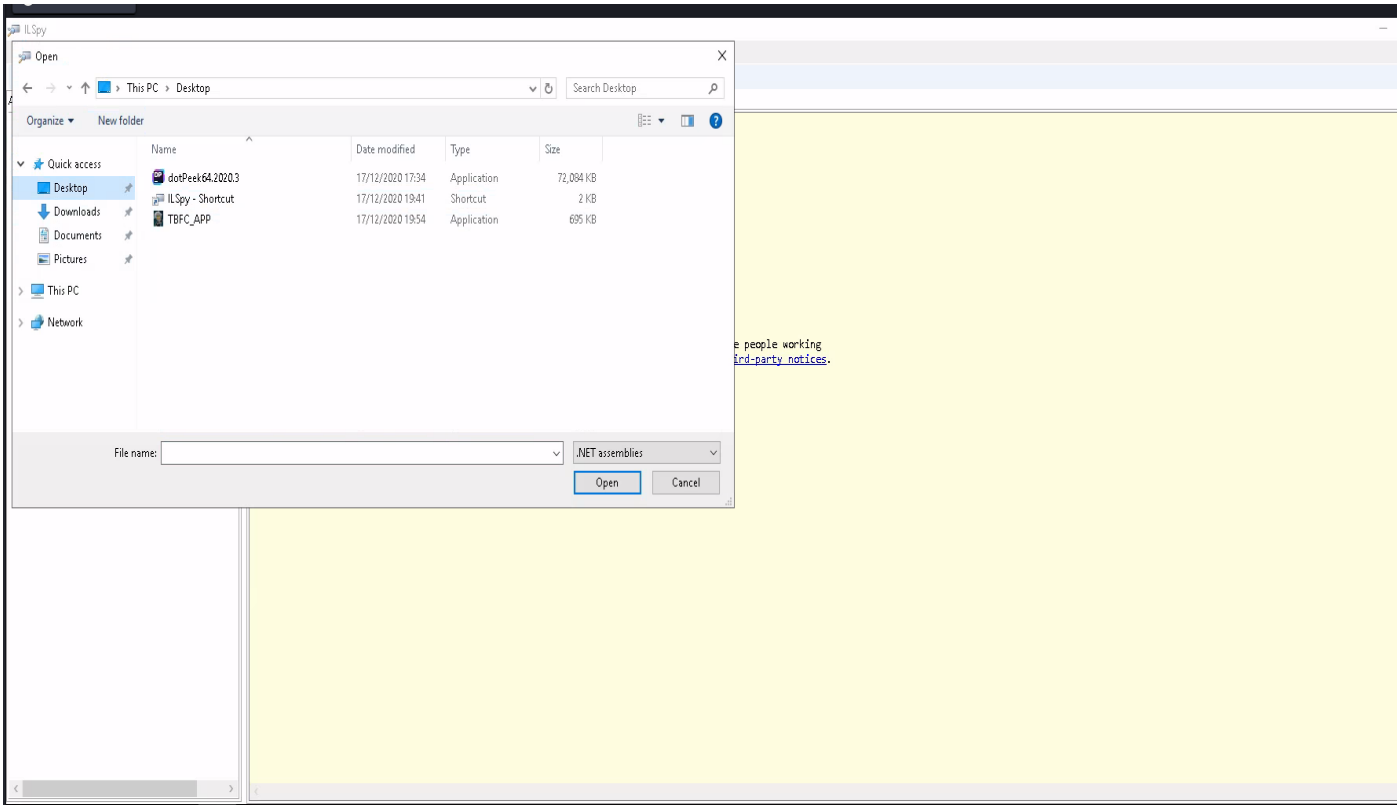
4. After login in, we look at the TBFC_APP and try entering a random password to check an output.



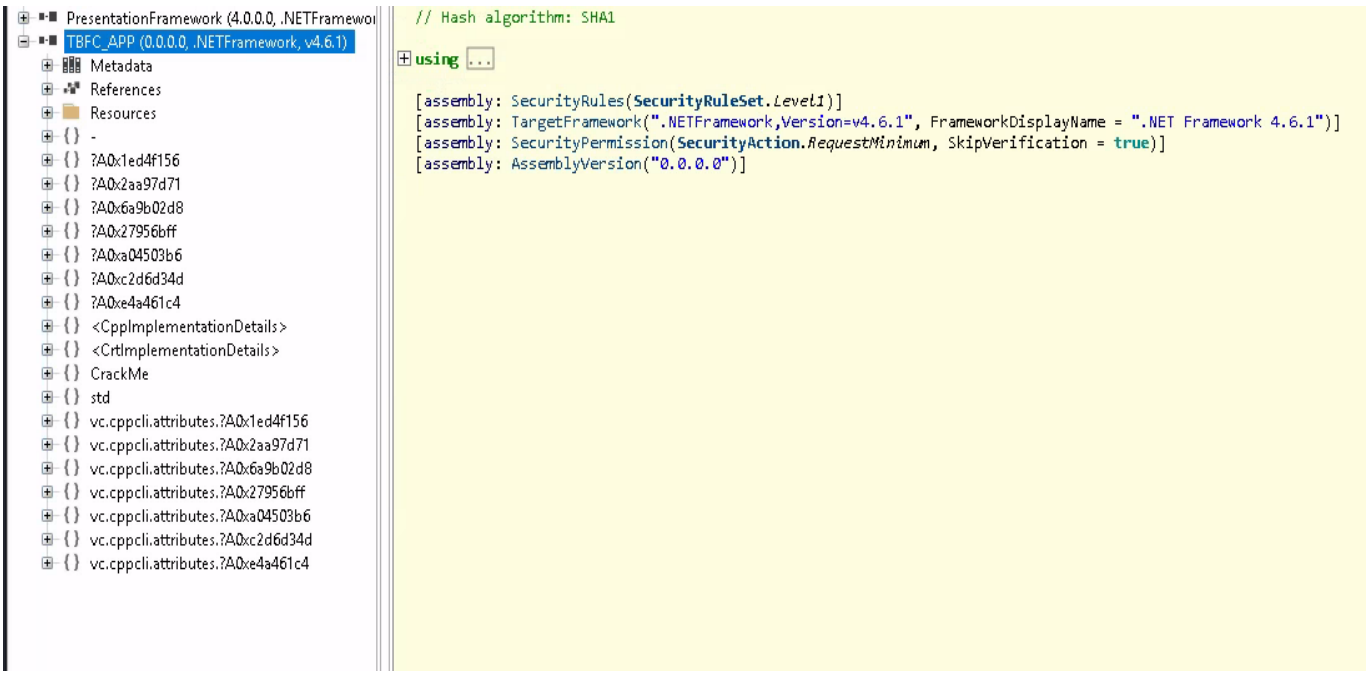

Recycle Bi
dotPeek64
ILSpy - Shortcut
TBFC_APP
TBFC Dashboard
T B F C
Password:
The Best Festival Company 2020
Submit
ENG
04:42
17/07/2022



5. As we can see, a message pops up stating that You're not Santa!
6. Now we are gonna use ILSpy to reverse engineer the password



7. Then, open the file that contains the TBFC_APP and begin decompiling

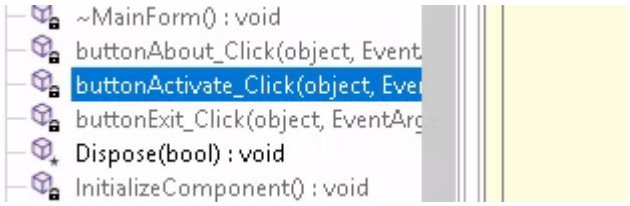


Q: What is Santa's password?

Answer: santapassword321

1. When the decompiling is completed, as we can see there is a tab call CrackMe
2. Look into CrackMe.
3. And there's a MainForm where we can see how the program works.

4. Afterwards, in MainForm there are many objects however the buttonActivate_Click() is what catches our attention.



5. Inside we can see C++ code and how the button works

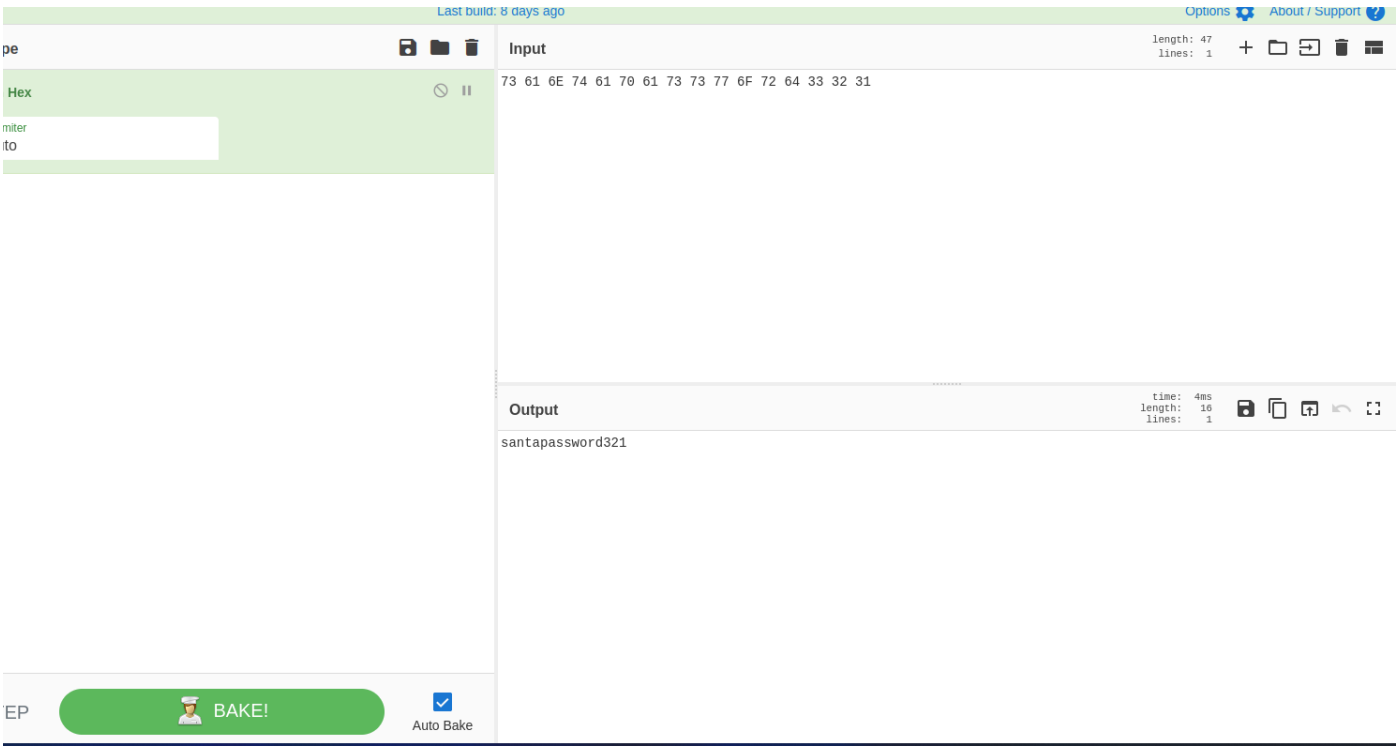
```
buttonActivate_Click(object, EventArgs) : void
// CrackMe.MainForm
using ...

private unsafe void buttonActivate_Click(object sender, EventArgs e)
{
    IntPtr value = Marshal.StringToHGlobalAnsi(textBoxKey.Text);
    sbyte* ptr = (sbyte*)System.Runtime.CompilerServices.Unsafe.AsPointer(ref <Module>.??_C@_0BB@IKKDFEPG@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);
}
```

6. There's a string containing the password however we aren't sure if that is the password and not a variable in the program, so we click on it since it's a module.

```
er(ref <Module>.??_C@_0BB@IKKDFEPG@santapassword321@);
```

7. That brought us to a tab where there is hexadecimal code.

8. We can head to cyberchef to decode the hexadecimal code.

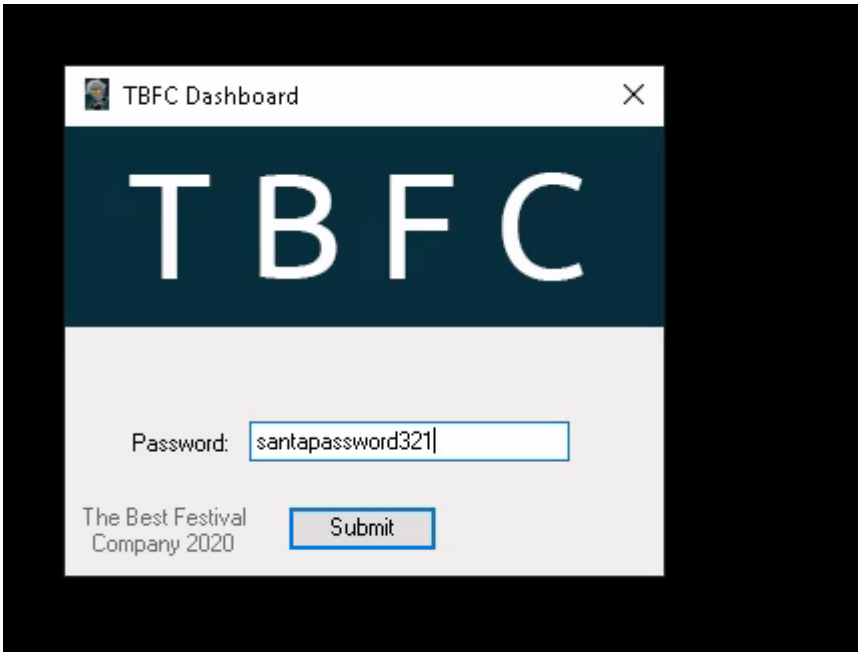


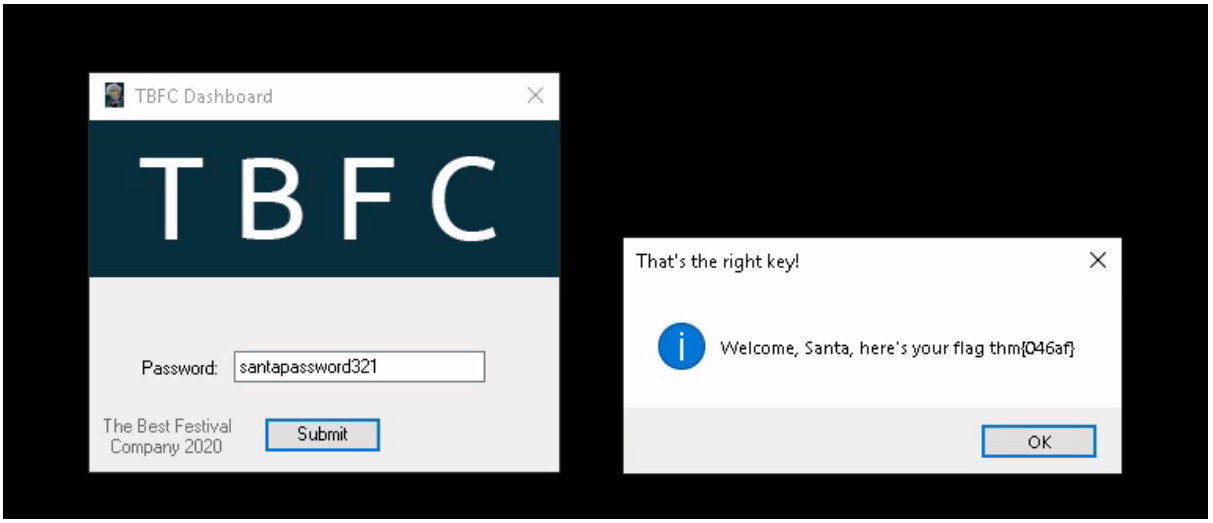
To get the password.

Q: Now that you've retrieved this password, try to login...What is the flag?

Answer: thm{046af}

1. After retrieving the password, now we can try submitting it to claim the flag





Thought Process/ Methodology:

We begin by launching Remmina while using the machine's ip address to log in. A prompt was shown and we entered the username and password that was given on THM. When we log in, we see a windows platform that contains several applications on the desktop, and there's an application called TBFC_APP that interests us. Then launching the application, there's a password textbox with the submit button and next to the button there's the full name of the application. We try typing random passwords to see if we can get in, however we are greeted with a message that we are not santa. We couldn't brute force our way in, so

we looked at the ILSpy application on the desktop and launched it to look at the code of the application. After opening up the file with the application on ILSpy, we begin looking around the code and try to find something that stands out. As we poke around, there's a tab called CrackMe with mainForm() that catches our eye with several objects inside of it. We see an object that is named buttonActivate_Click() and we suspect that this object is what happens when the submit button is clicked. Afterwards, there was C++ code and we identified how the code works, then we saw a password string but we aren't sure if it's the password, therefore we clicked to see the source of where it is from. There was hexadecimal code displayed, thus we can decode it using cyberchef and retrieve the password. After retrieving the password, we headed into the application and revealed the flag.

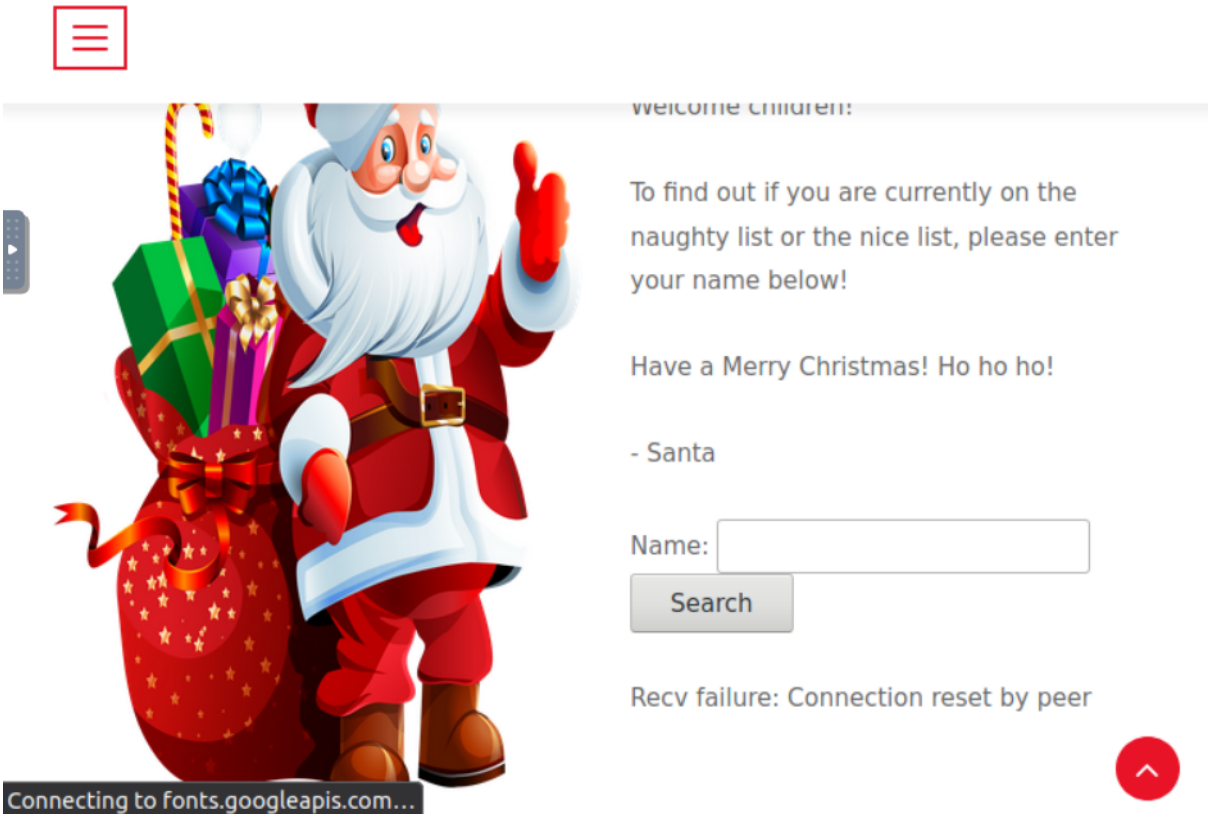
Day 19: The Naughty or Nice List

Solution and walkthrough:

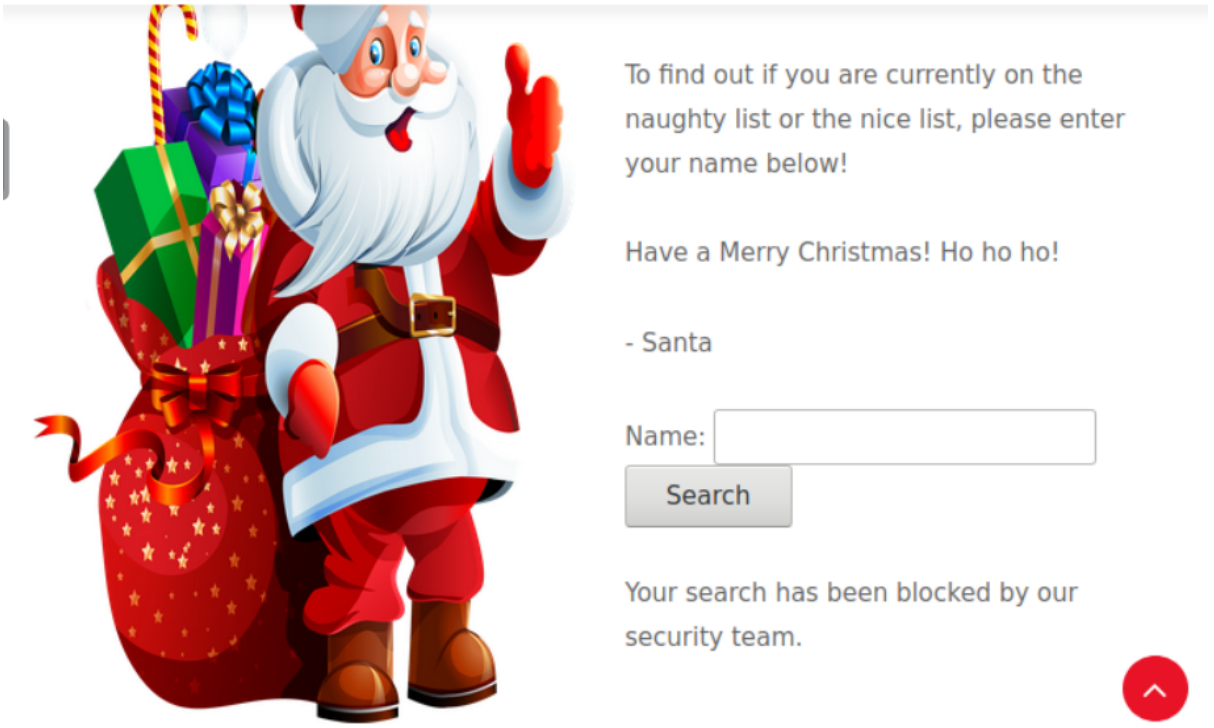
Q1: What is Santa's password?

Answer: Be good for goodness sake!

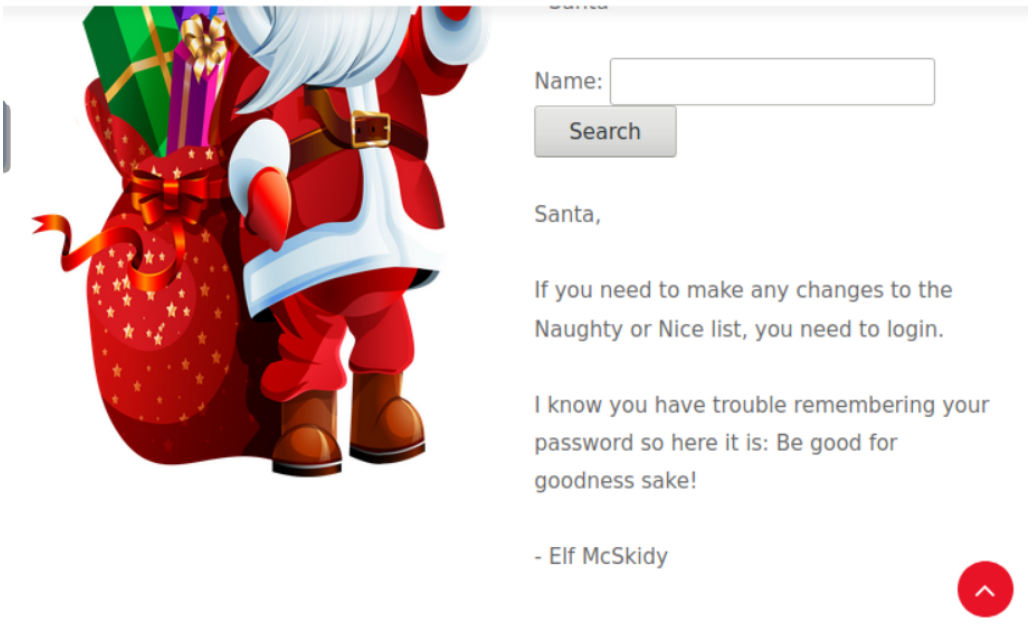
1. We should start by fetching the root of the same site, using the given URL, but we get an error message, "Not Found. The requested URL was not found on this server." This is generic 404 message, which shows we were able to make the server request our modified URL.
2. Next we try changing the port number, to which the message now changes to " Failed to connect to list.hohoho port 80: Connection refused"
3. Since both of those did not work, we try changing the port to 22, which is the default SSH port.



4. The message now changes to "Recv failure: Connection reset by peer". This means port 22 did open but did not understand what was sent.
5. We can also try replacing the list.hohoho hostname with "localhost" or "127.0.0.1", but that causes our search to be blocked.



6. This is because the developer has a check in place for this, which immediately blocks any hostname that does not start with list.hohoho.
7. Therefore, we just use a hostname which does start with list.hohoho, and in this case we use “list.hohoho.localtest.me”, to try and bypass that check.

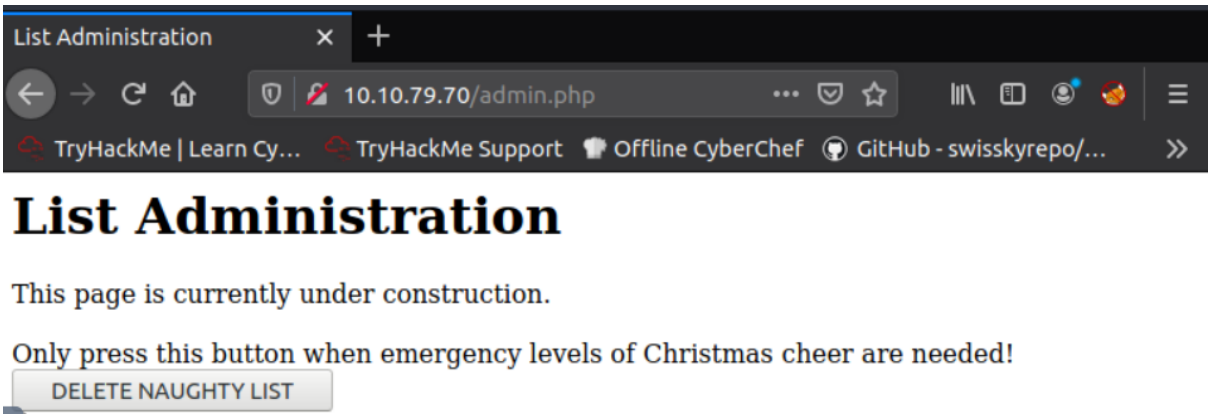


8. Success! It does work. Through this, we have recovered santa’s username since it is in Elf McSkidy’s message.

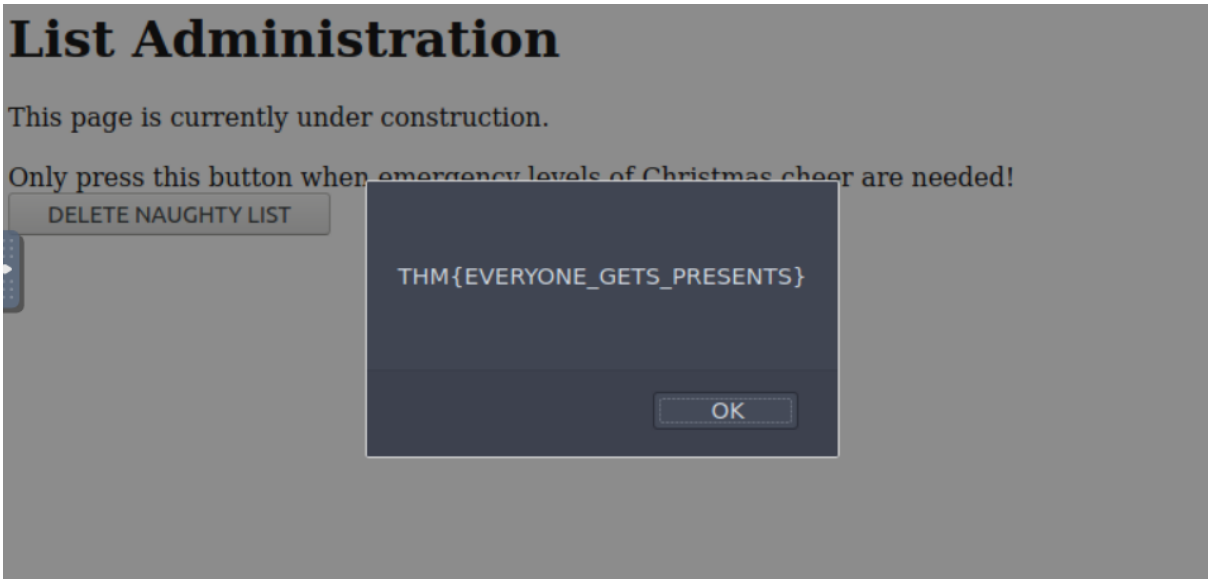
Q2: What is the challenge flag?

Answer: THM{EVERYONE_GETS_PRESENTS}

1. Since we have santa's password, all we need is the username, which shouldn't be hard to guess. It is Santa's account, and hence his username is Santa.
2. We log in with Santa's account details, and this screen shows up.



3. To get the challenge flag, all we have to do is delete the naughty list.



4. And voila! There it is, the answer for our question.

Day 20: PowershELIF to the rescue

Solution and Walkthrough :

Q1: Search for the first hidden elf file within the Documents folder. Read the contents of this file. What does Elf 1 want?

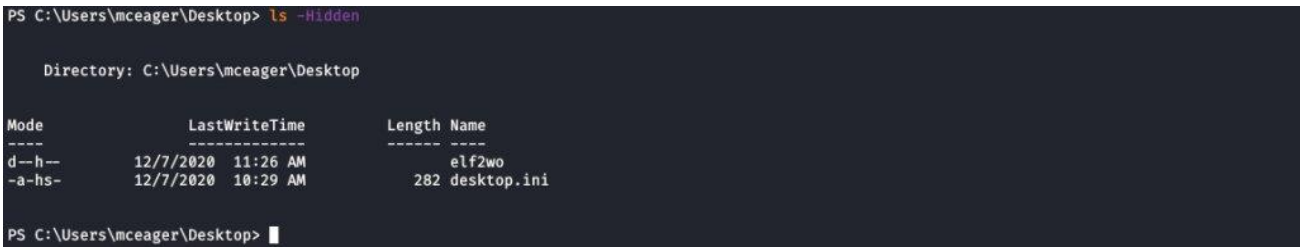
Answer: 2 front teeth

1. Navigate to our Documents directory with the command Set-Location .\Documents\. The first question asks us to find the contents of the hidden file within this directory. We can use ls to do this and add the flag –Hidden. The final command is ls -Hidden and will give us the following result.

2. Inside there is a hidden file called e1fone.txt. When we read the contents of the file with Get-Content -Path .\e1fone.txt, we see it contains a message about 2 front teeth.

Q: Search on the desktop for a hidden folder that contains the file for Elf 2. Read the contents of this file. What is the name of that movie that Elf 2 wants?
Answer: Scrooged

1. Search for the hidden directory on using ls -Hidden again. We see a hidden directory named elf2wo.



2. When we search the contents of this directory we see a file called e70smsW10Y4k.txt. When we read the contents of this file with Get-Content it reveals the movie Scrooged.

```
PS C:\Users\mceager\Desktop> cd .\elf2wo\
PS C:\Users\mceager\Desktop\elf2wo> ls -Hidden
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> Get-Content e70smsW10Y4k.txt
I want the movie Scrooged <3!
PS C:\Users\mceager\Desktop\elf2wo>
```

Q: Search the Windows directory for a hidden folder that contains files for Elf 3. What is the name of the hidden folder? (This command will take a while)

Answer: 3lfthr3e

1. Search the entire Windows directory for a hidden directory containing the third file. We can do this with the command Get-ChildItem -Path / -Recurse -Hidden -ErrorAction SilentlyContinue. After scrolling through the output, we see a hidden directory within System32 called 3lfthr3e.

```
    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
```

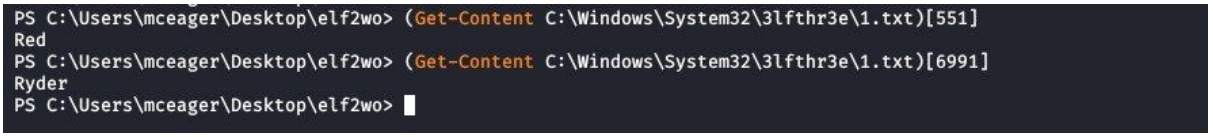
Q: How many words does the first file contain?
Answer: 9999

1. Use the command Get-Content C:\Windows\System32\3lfthr3e\1.txt | Measure-Object -Word and see that the file has 9999 words.

```
PS C:\Users\mceager\Desktop\elf2wo> Get-Content C:\Windows\System32\3lfthr3e\1.txt | Measure-Object -Word

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

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

Q: What 2 words are at index 551 and 6991 in the first file?
Answer: Red Ryder

1. After finding the total number of words, we want to find specific words in the file. We can do this with the command (Get-Content C:\Windows\System32\3lfthr3e\1.txt)[index] where the index is the location of the word we want to find. We can use this to find 551 and 6991 respectively.



Q: This is only half the answer. Search in the 2nd file for the phrase from the previous question to get the full answer. What does Elf 3 want? (use spaces when submitting the answer)
Answer: Red Ryder BB Gun

1. Finally, we want to search the second file for the phrase from the previous question. We can do this with the command Select-String -Path C:\Windows\System32\3lfthr3e\2.txt -Pattern ‘redryder’.