PSP0201

Week 5

Writeup

Group Name: uwu gang

Members

ID	Name	Role
1211101376	Isaiah Wong Terjie	Leader
1211101321	Muhammad Zafran Bin Mohd Anuar	Member
1211100857	Javier Austin Anak Jawa	Member
1211100824	Ahmad Danial Bin Ahmad Fauzi	Member

Day 16: Scripting – Help! Where Is Santa?

Tools used: Kali Linux, Firefox, NMAP, Python. Sublime text

Solution/walkthrough:

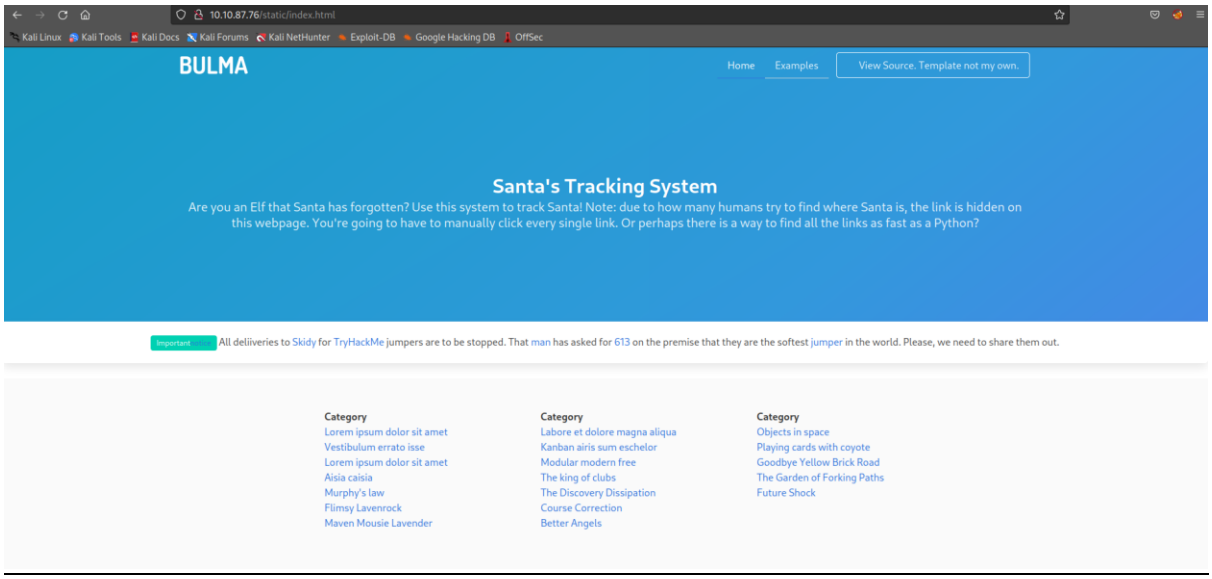
Question 1 :

We used nmap along with the IP to search for the web server port. Which is 80.

```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.87.76
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-16 04:38 EDT
Nmap scan report for 10.10.87.76
Host is up (0.20s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

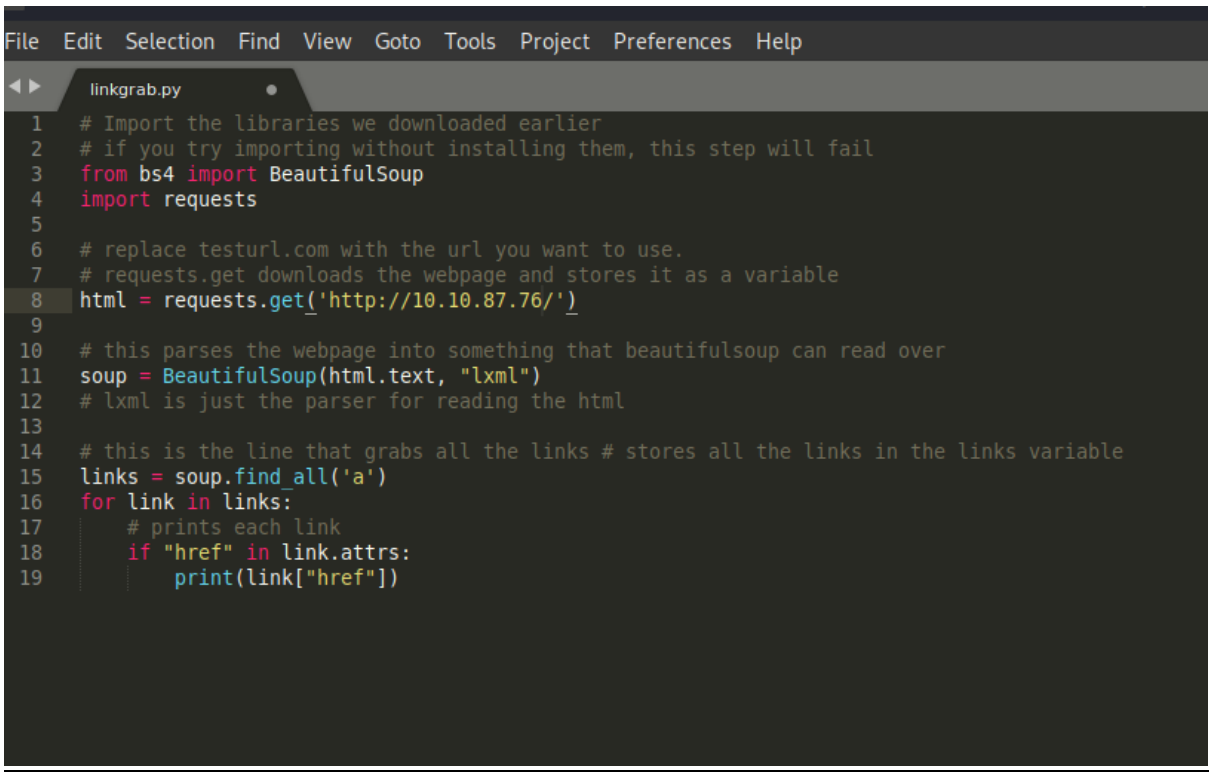
Question 2:

It is using the BULMA template.

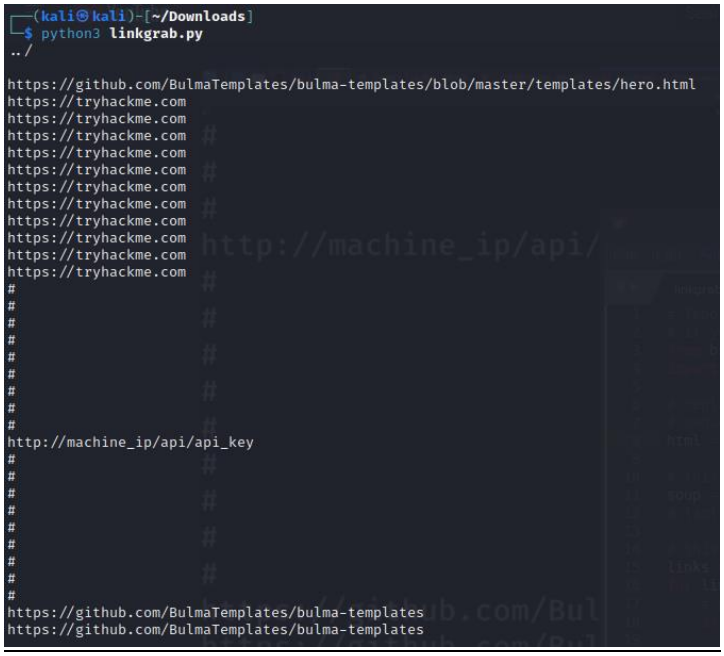


Question 3:

In order to find the API directory, we created a Python file and pasted the BeautifulSoup library.

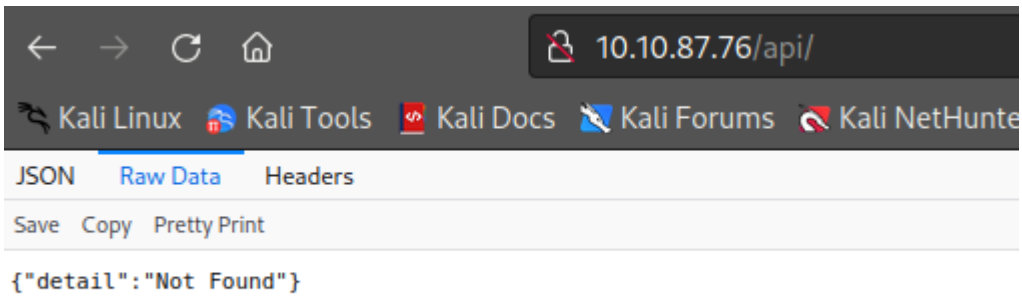


After making minor adjustments to the script, we ran the file and found the API directory.



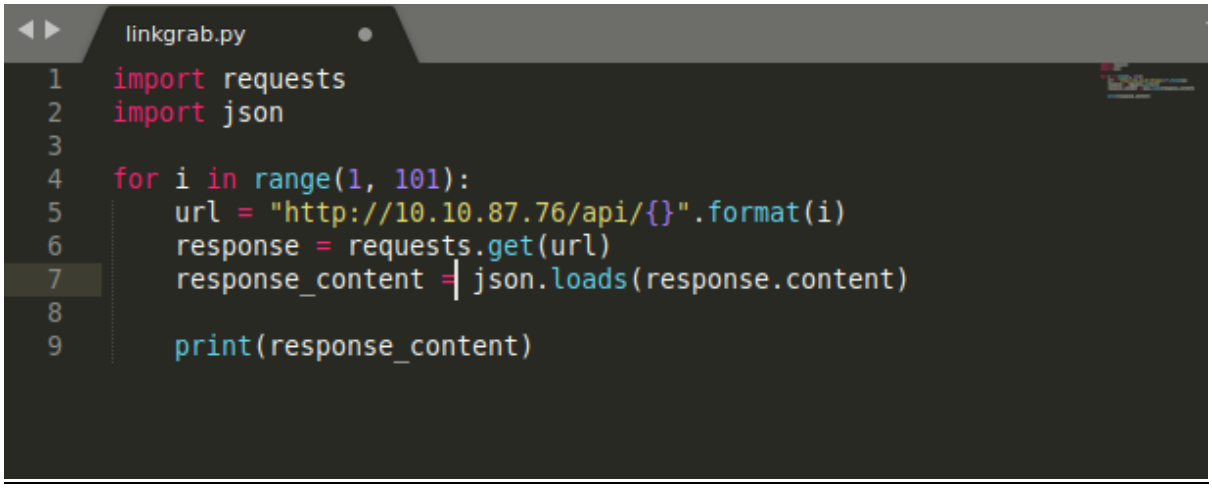
Question 4

We were asked to go to the API endpoint without any set of parameters.

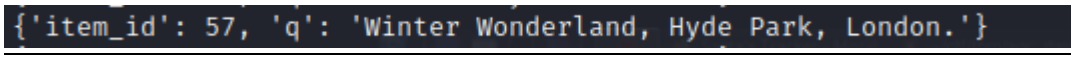


Question 5

We used the “requests” and “JSON” library



We ran the script and searched for the valid API key.



Thoughts/Methodology:

Firstly, we were tasked to find the port of the webserver that is currently running. Again, we should know that finding the port requires nmap. The port that was running for HTTP is 80 and we were brought to a webpage named BULMA, after we pasted the link that was provided in the browser. We were told to look up for the hidden link somewhere, so we decided to use Python in order to complete our task. We created a sublime text file and used the library that was provided on Day 15, which is the "BeautifulSoup" library. After making some changes to the script we ran the python file in the terminal, and we found a link in the format of "http://machine_ip/api/api_key". Then, we navigated the API endpoint without any parameters. Later, we see some results from our previous script that includes API key along with URL and our task requires us to use the number between 0 and 100 to look for the right key. But it would take quite some time, so we created another Python file using the "requests" and "JSON" library. Lastly, we ran the script and got our results.

Day 17: Reverse Engineering- ReverseELFneering

Tools used: Kali Linux, Firefox, NMAP, Python. Sublime text

Solution/walkthrough:

Question 1:

Provided in the thread

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:

Provided in the Radare2 cheat sheet

Analyse all	`aa[a[a]]`

Question 3:

Provided in the Radare2 cheat sheet

Set breakpoint	`db [addr]`

Question 4:

Provided in the Radare2 cheat sheet

Continue execution	`dc`

Question 5,6,7:

Scanned the IP using nmap.

```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.202.248
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-16 11:35 EDT
Nmap scan report for 10.10.202.248
Host is up (0.20s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 27.51 seconds
```

Login to the webserver using SSH.

```
┌──(kali㉿kali)-[~]
└─$ ssh elfmceager@10.10.202.248
elfmceager@10.10.202.248'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 Sat Jul 16 15:00:40 UTC 2022

  System load:  0.08               Processes:           90
  Usage of /:   39.4% of 11.75GB   Users logged in:     0
  Memory usage: 8%                 IP address for ens5: 10.10.202.248
  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 then open the “challenge1” file using Radare2 to debug.

```
Last login: Wed Dec 16 18:25:51 2020 from 192.168.190.1
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1582 started...
= attach 1582 1582
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
asm.bits 64
[0x00400a30]> aa
```

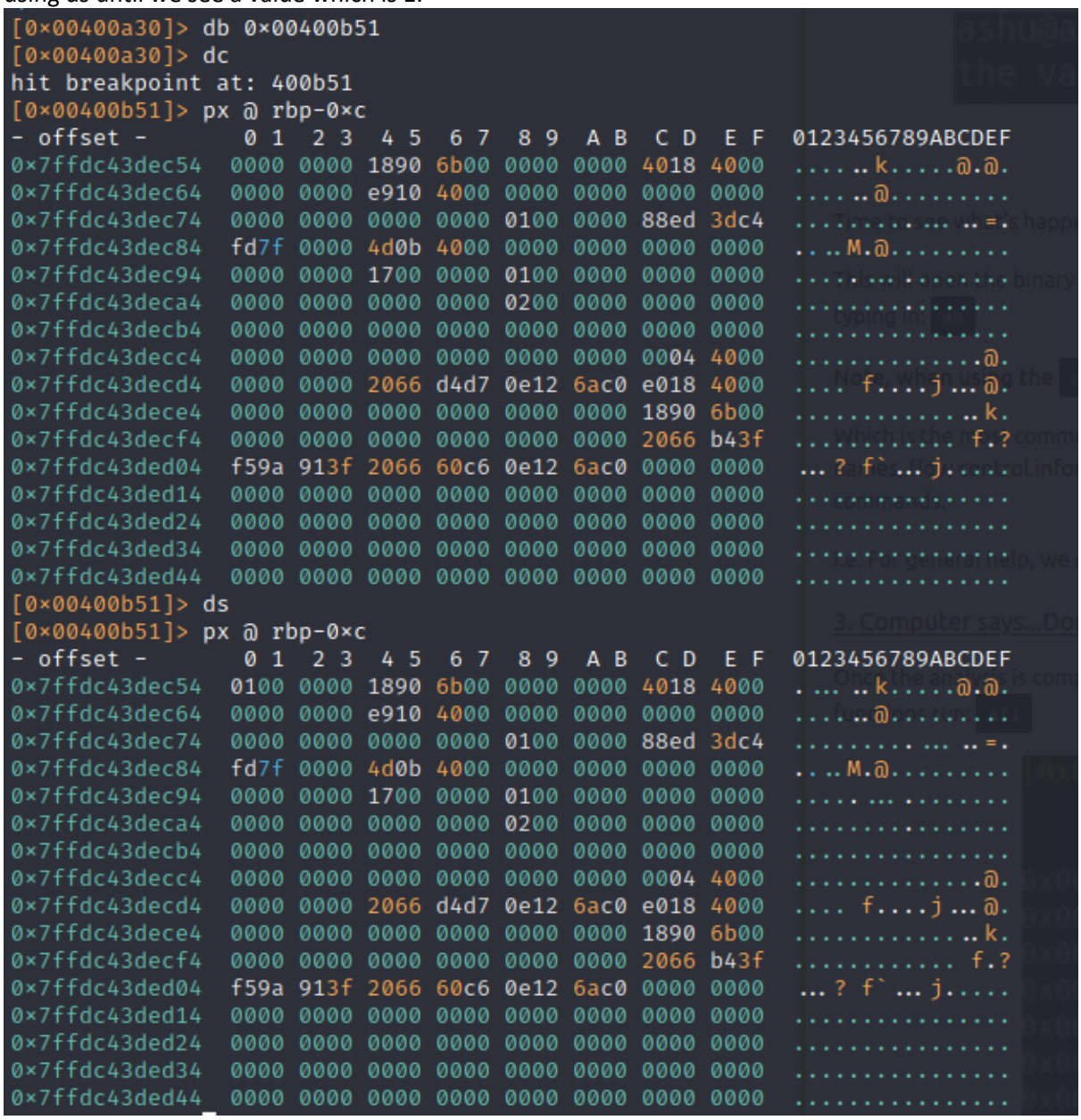
Then we analyse the file using aa command

```
[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]>
```

We examine the assembly code using “pdf @ main” .

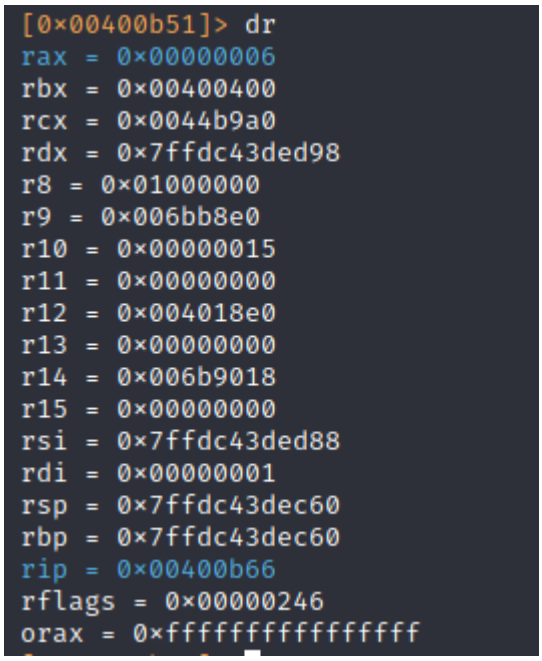
```
[0x00400a30]> pdf @ main
            ;-- main:
/ (fcn) sym.main 35
|   sym.main ();
|           ; var int local_ch @ rbp-0xc
|           ; var int local_8h @ rbp-0x8
|           ; var int local_4h @ rbp-0x4
|              ; DATA XREF from 0x00400a4d (entry0)
|           0x00400b4d      55              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
[0x00400a30]>
```

Use db command to create a break point for the selected address and then we use dc to run the program. We then analyse the contents of local_ch with “px @ rbp-0xc” and switch the directory using ds until we see a value which is 1.

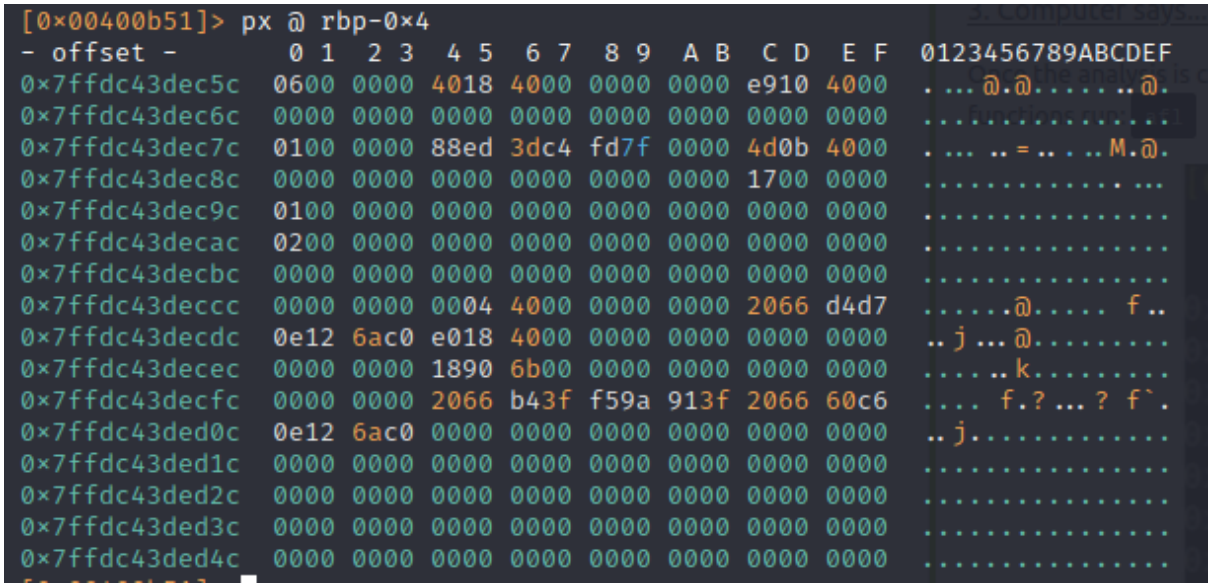


```
[0×00400a30]> db 0×00400b51
[0×00400a30]> dc
hit breakpoint at: 400b51
[0×00400b51]> px @ rbp-0×c
- offset -       0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0×7ffdc43dec54  0000 0000 1890 6b00 0000 0000 4018 4000  ....  k.....@.@.
0×7ffdc43dec64  0000 0000 e910 4000 0000 0000 0000 0000  ....  @.........
0×7ffdc43dec74  0000 0000 0000 0000 0100 0000 88ed 3dc4  ....  ..... ..  =.
0×7ffdc43dec84  fd7f 0000 4d0b 4000 0000 0000 0000 0000  ..  M.@.........
0×7ffdc43dec94  0000 0000 1700 0000 0100 0000 0000 0000  ....  ....  ....
0×7ffdc43deca4  0000 0000 0000 0000 0200 0000 0000 0000  ....  ....  ....
0×7ffdc43decb4  0000 0000 0000 0000 0000 0000 0000 0000  ....  ....  ....
0×7ffdc43decc4  0000 0000 0000 0000 0000 0000 0004 4000  ....  ....  ..@.
0×7ffdc43decd4  0000 0000 2066 d4d7 0e12 6ac0 e018 4000  ....  f...  j..  @.
0×7ffdc43dece4  0000 0000 0000 0000 0000 0000 1890 6b00  ....  ....  ..  k.
0×7ffdc43decf4  0000 0000 0000 0000 0000 0000 2066 b43f  ....  ....  f.  ?
0×7ffdc43ded04  f59a 913f 2066 60c6 0e12 6ac0 0000 0000  ...  ?  f`  ..  j.....
0×7ffdc43ded14  0000 0000 0000 0000 0000 0000 0000 0000  ....  ....  ....
0×7ffdc43ded24  0000 0000 0000 0000 0000 0000 0000 0000  ....  ....  ....
0×7ffdc43ded34  0000 0000 0000 0000 0000 0000 0000 0000  ....  ....  ....
0×7ffdc43ded44  0000 0000 0000 0000 0000 0000 0000 0000  ....  ....  ....
[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
0×7ffdc43dec54  0100 0000 1890 6b00 0000 0000 4018 4000  . ...  ..  k.....@.@.
0×7ffdc43dec64  0000 0000 e910 4000 0000 0000 0000 0000  ....  ..  @.....  ...
0×7ffdc43dec74  0000 0000 0000 0000 0100 0000 88ed 3dc4  ....  ....  ... ..  =.
0×7ffdc43dec84  fd7f 0000 4d0b 4000 0000 0000 0000 0000  ..  M.@.........
0×7ffdc43dec94  0000 0000 1700 0000 0100 0000 0000 0000  ....  ....  ....
0×7ffdc43deca4  0000 0000 0000 0000 0200 0000 0000 0000  ....  ....  ....
0×7ffdc43decb4  0000 0000 0000 0000 0000 0000 0000 0000  ....  ....  ....
0×7ffdc43decc4  0000 0000 0000 0000 0000 0000 0004 4000  ....  ....  ..@.
0×7ffdc43decd4  0000 0000 2066 d4d7 0e12 6ac0 e018 4000  ....  f...  j..  @.
0×7ffdc43dece4  0000 0000 0000 0000 0000 0000 1890 6b00  ....  ....  ..  k.
0×7ffdc43decf4  0000 0000 0000 0000 0000 0000 2066 b43f  ....  ....  f.  ?
0×7ffdc43ded04  f59a 913f 2066 60c6 0e12 6ac0 0000 0000  ...  ?  f`  ..  j.....
0×7ffdc43ded14  0000 0000 0000 0000 0000 0000 0000 0000  ....  ....  ....
0×7ffdc43ded24  0000 0000 0000 0000 0000 0000 0000 0000  ....  ....  ....
0×7ffdc43ded34  0000 0000 0000 0000 0000 0000 0000 0000  ....  ....  ....
0×7ffdc43ded44  0000 0000 0000 0000 0000 0000 0000 0000  ....  ....  ....
```

To find the eax value at the first imul, we used ds to move to the next line and then use dr to see the value of the eax.



Lastly, we want to find the value of the local_4h before it was set to 0. We then use the ds command to switch to the line where it shows local_4h and run “px @ rbp-0x4”. The value that we found was 6.



Thoughts/Methodology:

Firstly, we scanned the IP using nmap to determine the webserver of the IP. Later, we found out that the IP is only using the SSH port. So, we logged in to the target machine using "ssh elfmceager@10.10.202.248" and along with the password "adventofcyber". We know that there are two files naming "file1" and "challenge1". The task requires us to focus on the "challenge1" file and the first thing we have to do is using the radare2 mode, "r2 -d ./challenge1". Once we entered the debug mode, we can do a full analysis with "aa". Now, we want to dig deeper into the analysis we used "afl | grep main". Then, we ran "pdf @ main" to examine further into the assembly code. Now, we use the "db" command to create a break point for the selected address and then we use "dc" to run the program. We then analyse the contents of local_ch with "px @ rbp-0xc" and switch the directory using ds until we see a value which is 1. To find the eax value at the first imul, we used "ds" to move to the next line and then use "dr" to see the value of the eax. Lastly, we are required to find the value local_4h before it was set to 0. We then run the "ds" command to switch to the next line where it shows local_4h and run "px @ rbp-0x4". The value that was shown is 6.

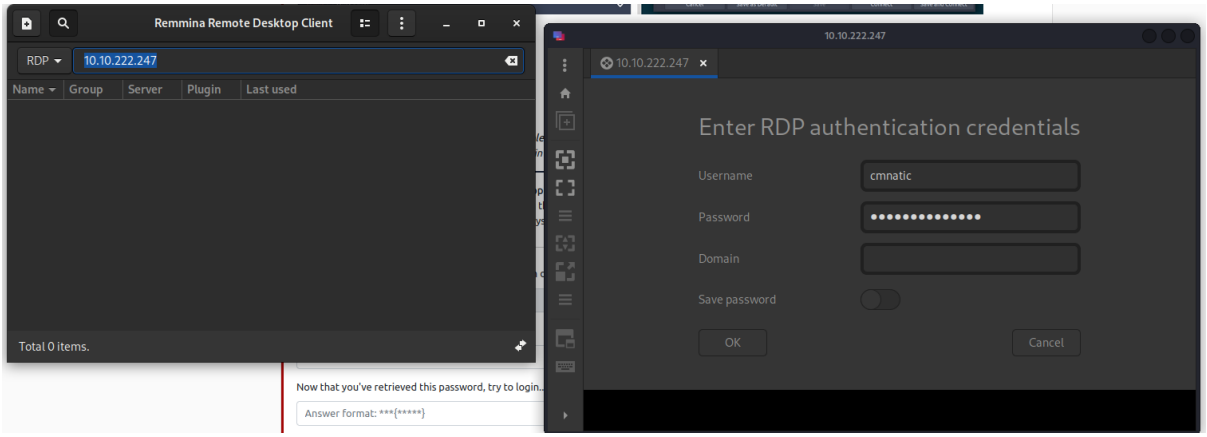
Day 18: Reverse Engineering – The Bits of Christmas

Tools: Kali Linux, FireFox, Remmina, IL SPY, Cyberchef

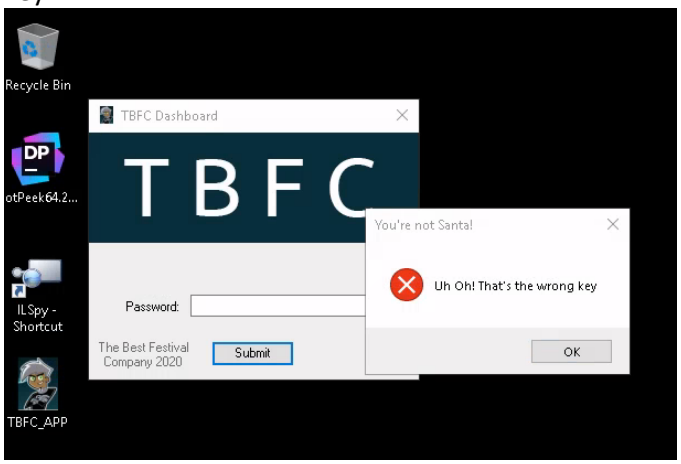
Solutions:

Question 1:

We entered the machine's IP address into the Remmina Client and run. Then, we enter the credentials.

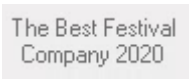


We try to enter the password for TBFC_APP and we received a message "Uh Oh! That's the wrong key".



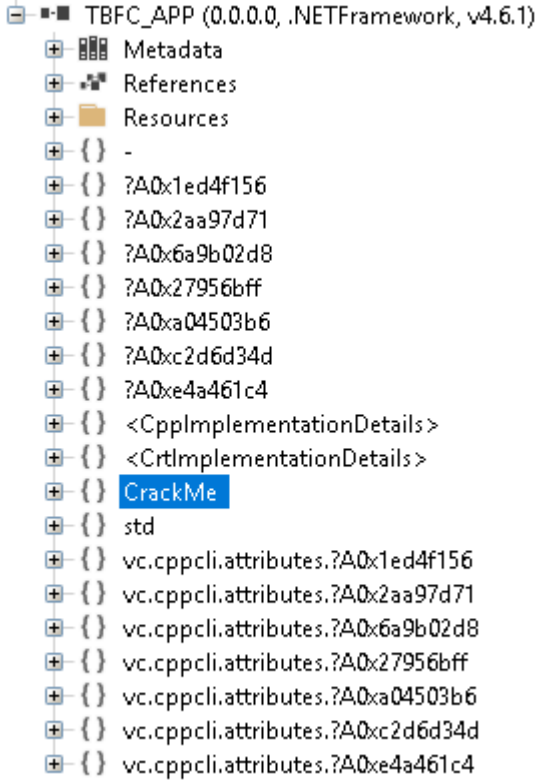
Question 2:

Bottom left corner of the TBFC_APP.



Question 3:

Once we launched IL Spy and open it with the TBFC_APP, to allow us to do some reverse engineering. Once we open it, we are required to find Santa's password. So, we were curious about the contents that is inside the CrackMe directory.



Question 4:

We entered the Mainform and found the password as well.

```
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;
```

Question 5:

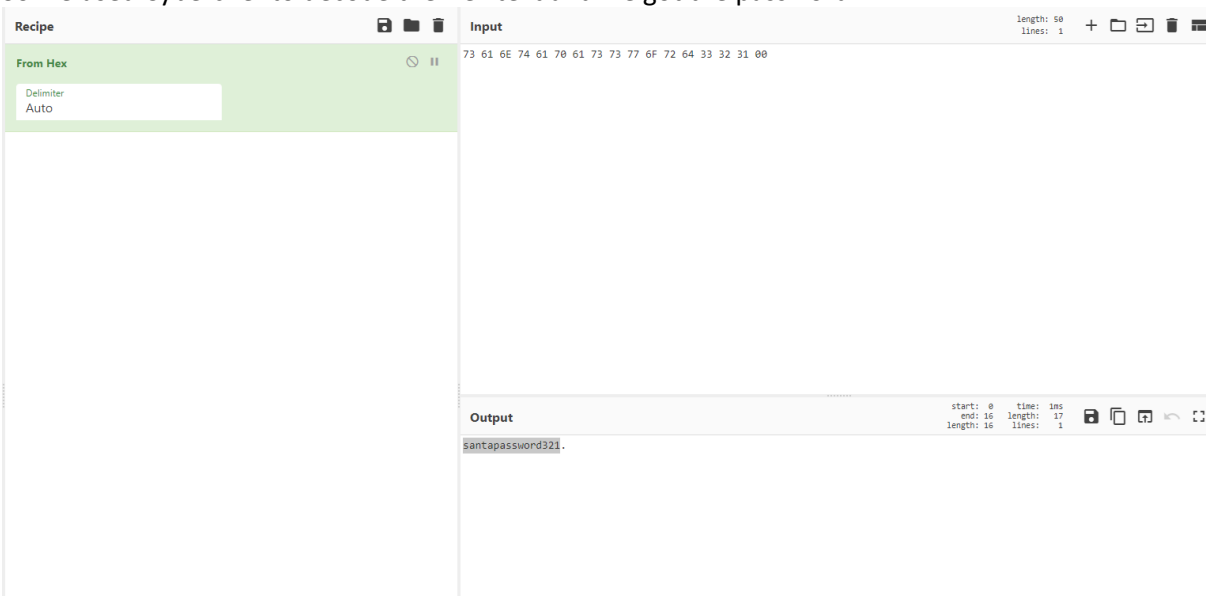
We double clicked on the password because it acted as a button, "buttonActivate_Click". But the data was not supported in IL Spy.

```
using ...

internal static $ArrayType$$$BY0BB@$$CBD ??_C@_0BB@IKKDFEPG@santapassword321@/* Not supported: data(73 61 6E 74 61 70 61 73 73 77 6F 72 64 33 32 31 00) */;
```

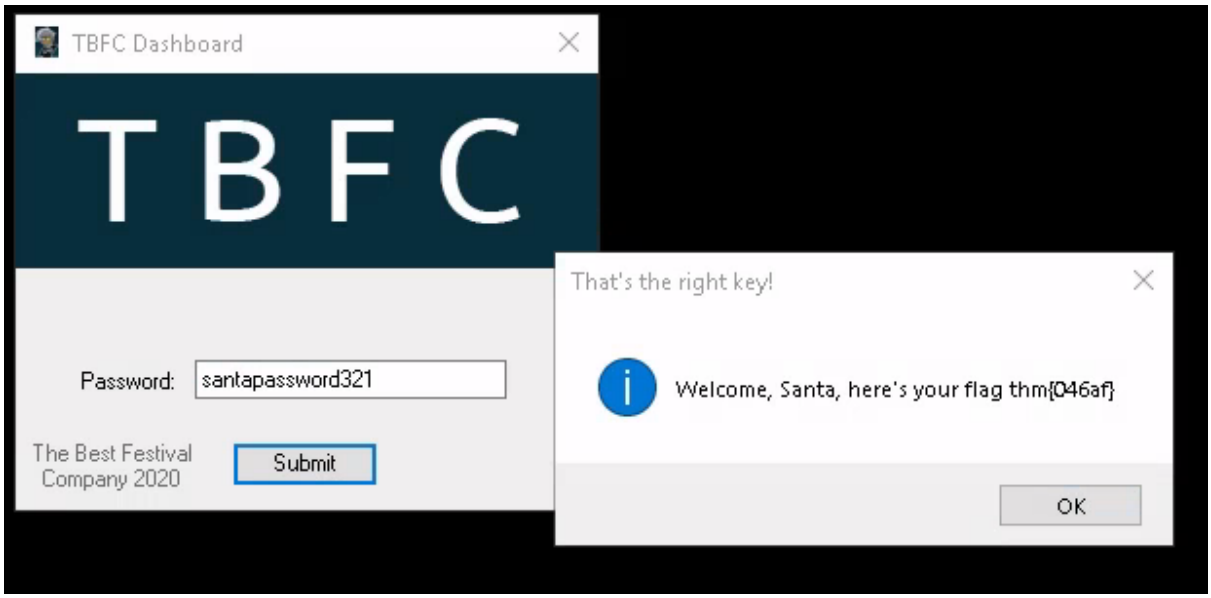
Question 6:

So we used Cyberchef to decode the Hex text and we got the password.



Question 7

We entered the password into the APP and received the flag.



Thoughts/Methodology:

Firstly, we are required to enter the IP into Remmina RDP Client and fill in the provided credentials. Once we are connected, we are presented with a Windows desktop. Now, we are required to open the TBFC_APP to understand what TBFC stands for and try to click the submit button without a password to find the message that shows up if we enter the wrong password. Later, we want to open TBFC_APP in IL Spy to do some reverse engineering. Then, we are asked to find Santa's password and this text named "CrackMe" was a little suspicious, so we proceed to click on it to look at the contents inside the directory. After some searching, we ended up inside the Mainform section and we found a line with "santapassword321". We were curious, so we double clicked on the line, and it acted as a "buttonActivate_Click". But unfortunately, the data was not supported in IL Spy. Then, we head to CyberChef to do some decoding for the Hex text, and we got our password. Lastly, we enter the password into TBFC_APP and we manage to get our flag.

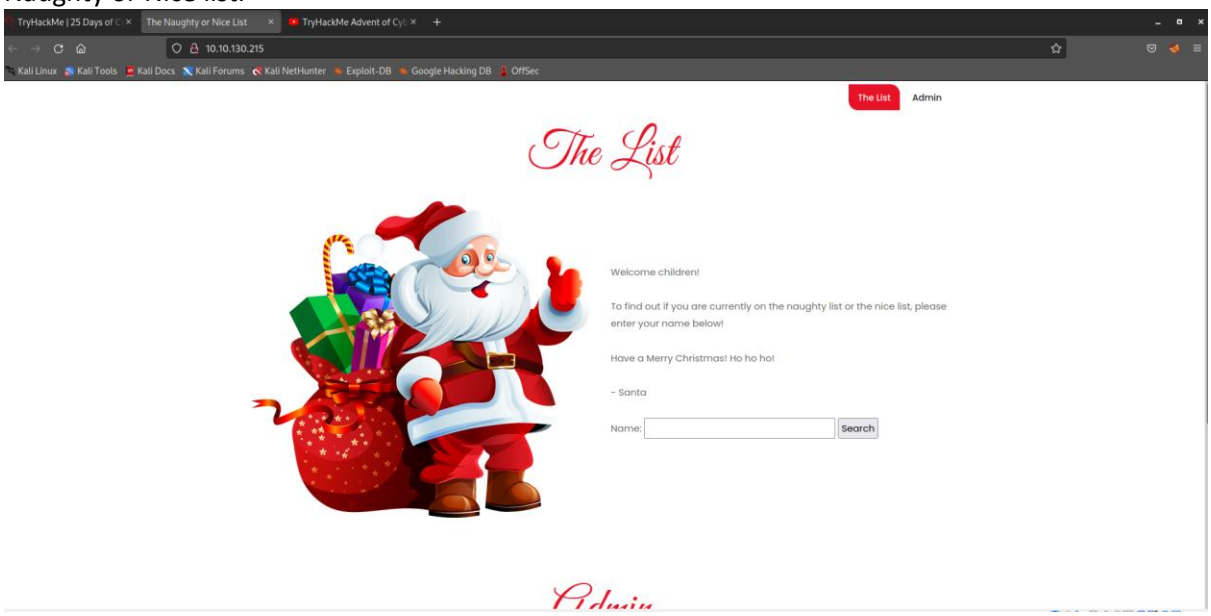
Day 19: Web Exploitation – The Naughty or Nice List

Tools: Kali Linux, FireFox,

Solutions:

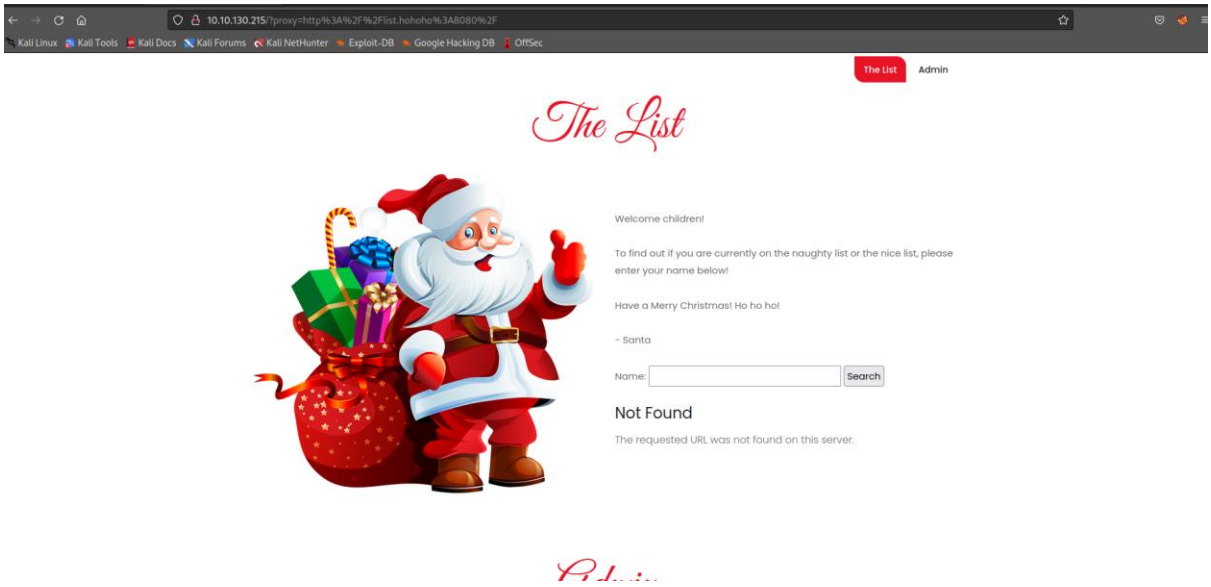
Question 1:

After booting up our target machine, we navigate the IP address in our address bar, and we are greeted with a homepage which allows us to enter different names to check if that person is on the Naughty or Nice list.



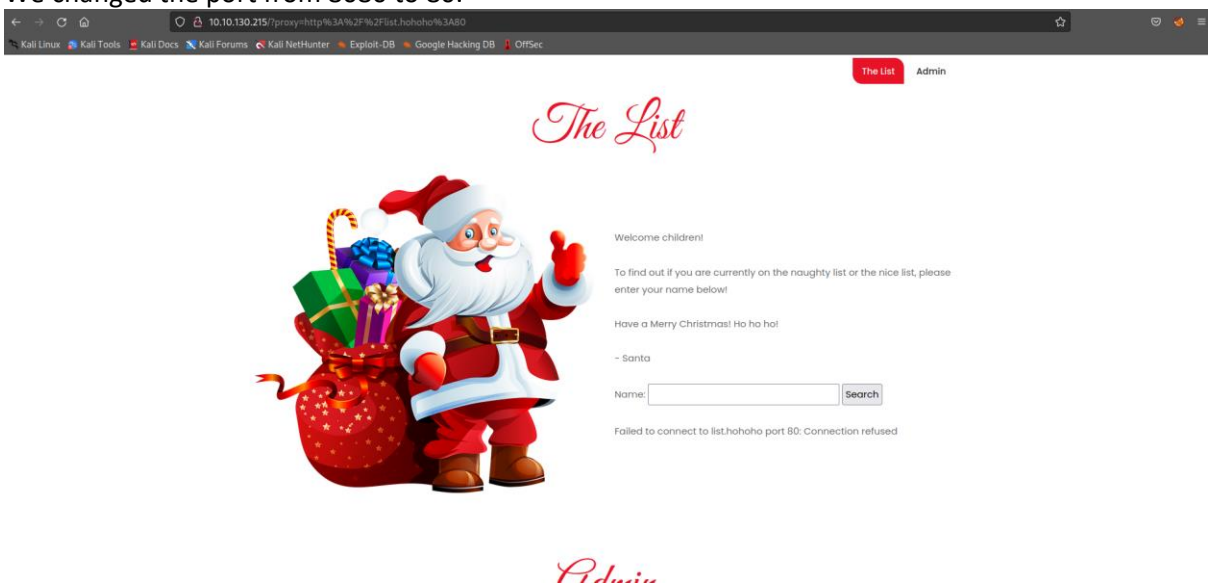
Question 2:

We entered “http://10.10.130.215//?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F” in the address bar.



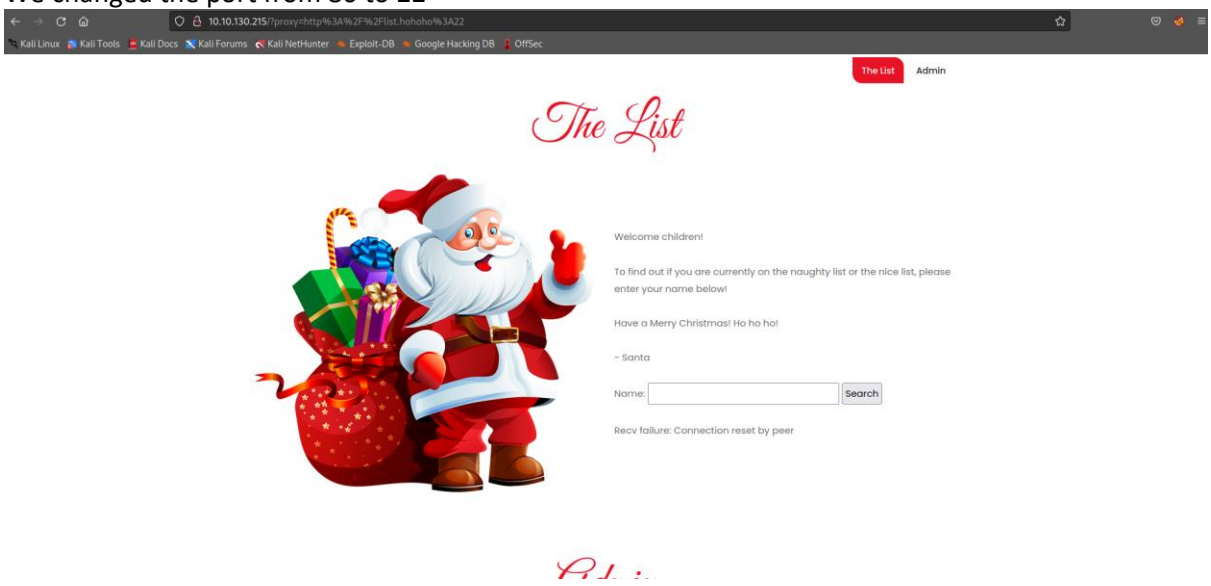
Question 3:

We changed the port from 8080 to 80.



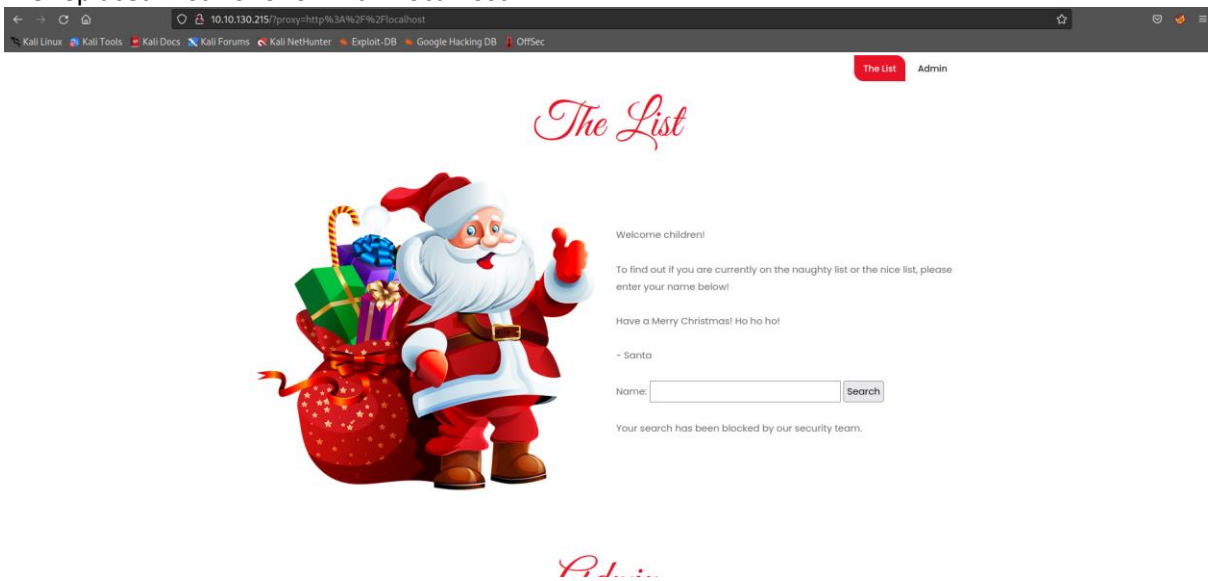
Question 4:

We changed the port from 80 to 22



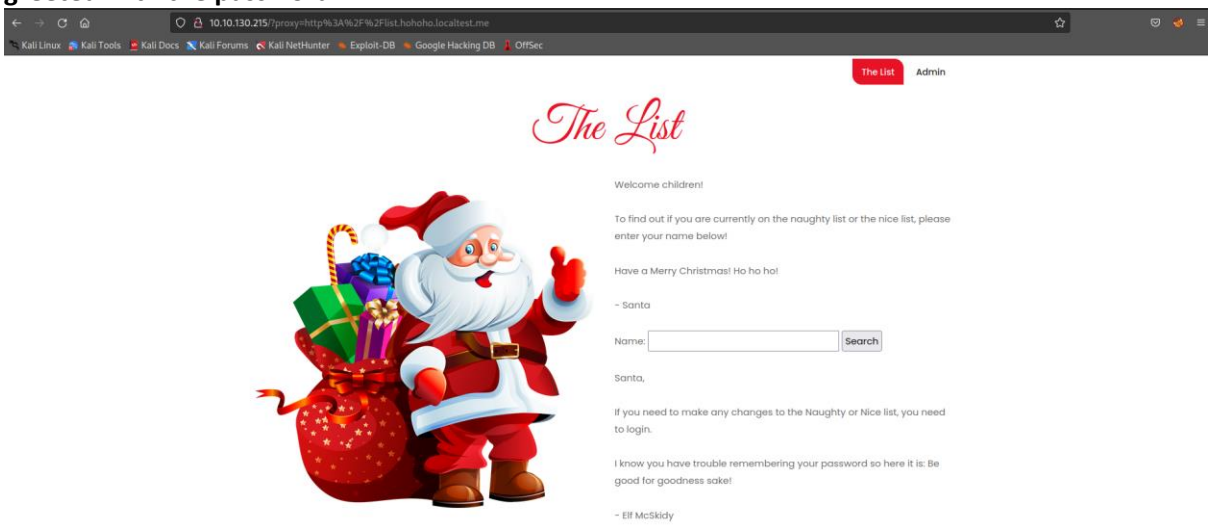
Question 5:

We replaced “list.hohoho” with “localhost”.



Question 6:

We know that the only host that is compatible with “list.hohoho” is “**localtest.me”. We were greeted with the password.**

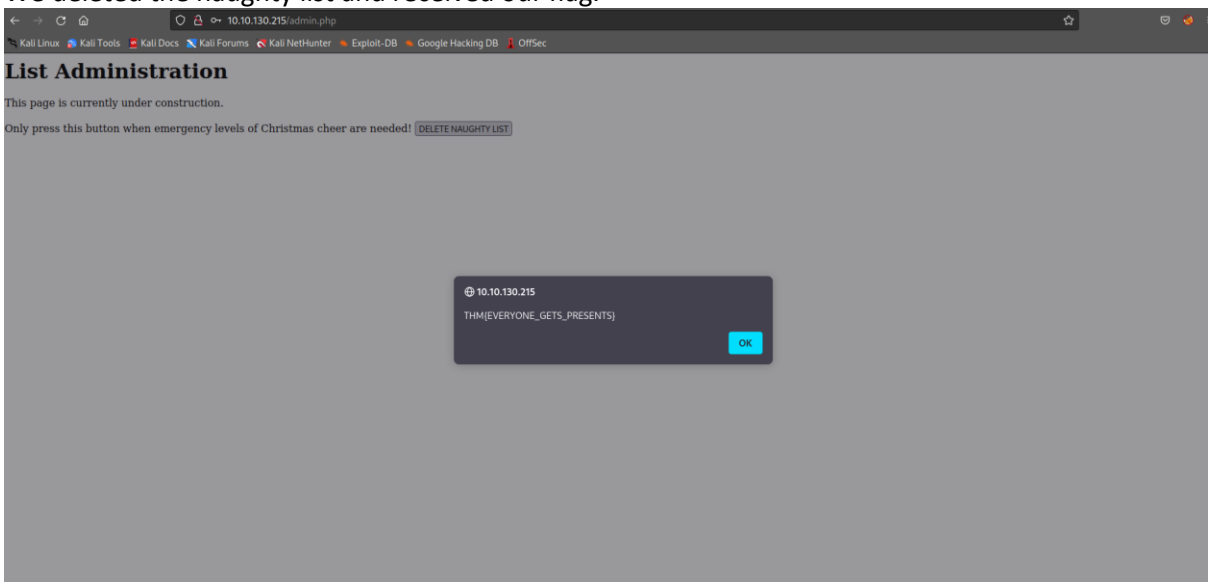


Question 7:

Enter the credentials.

Question 8:

We deleted the naughty list and received our flag.



Thoughts/Methodology:

After starting up our target system, we input the IP address in the address bar to see the homepage, where we may type in different names to see if they are on the Nice or Naughty list. Later, we were required to test a few additional "proxy" parameters that was provided. After some experimentation, we found out that the only subdomain that is compatible with "list.hohoho" is "localtest.me". Then, we tried using the full URL "**http://10.10.130.215//?proxy=http%3A%2F%2Flist.hohoho.localtest.me"** and we received a message from Elf McSkidy that contains a password to login. After logging into the admin panel, we are required to delete the naughty list and obtain the flag which is **THM{EVERYONE_GETS_PRESENTS}.**

Day 20: Blue Teaming – PowerELIF to the rescue

Tools: Kali Linux, FireFox,

Solutions:

Question 1:

Shown in the SSH manual.

```
-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:

After running "ssh -l mceager 10.10.27.122", we successfully got into the machine.

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

mceager@ELFSTATION1 C:\Users\mceager>
```

Launch the powershell.

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

Switch directory to Documents and list out the hidden contents using "ls -Hidden".

```
PS C:\Users\mceager> cd Documents
PS C:\Users\mceager\Documents> ls -Hidden


    Directory: C:\Users\mceager\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hsl         12/7/2020  10:28 AM                My Music
d--hsl         12/7/2020  10:28 AM                My Pictures
d--hsl         12/7/2020  10:28 AM                My Videos
-a-hs-         12/7/2020  10:29 AM            402 desktop.ini
-arh--        11/18/2020   5:05 PM             35 e1fone.txt
```

Open the file using the cat command to show us the hidden contents inside the file.

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

Question 3:

Switch directory to "..\Desktop\" and list out the hidden directories.

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

Then we switch directory to ".\elf2wo\" and list out the hidden contents.

```
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
```

Open the contents of "e70smsW10Y4k.txt"

```
PS C:\Users\mceager\Desktop\elf2wo> cat e70smsW10Y4k.txt
I want the movie Scrooged <3!
```

Question 4:

After scrolling through the directory, we found 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
```

Question 5:

Switch to the hidden directory and list out the hidden contents inside the directory.

```
PS C:\Users\mceager\Desktop\elf2wo> cd C:\Windows\System32\3lfthr3e
PS C:\Windows\System32\3lfthr3e> ls
PS C:\Windows\System32\3lfthr3e> ls -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
```

Open the file and measure the word length of the file.

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

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

Question 6:

We run "(cat 1.txt)[index]" to find the specific word in the file.

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

Question 7:

Search for the second file for the phrase from the previous file.

```
PS C:\Windows\System32\3lfthr3e> Select-String 2.txt -Pattern 'redryder'

2.txt:558704:redryderbbgun
```

Thoughts/Methodology:

Once the machine finished booting, we used SSH to connect to the machine along with the provided credentials. Then, we opened the PowerShell within the terminal and navigate to our Documents directory with the command "**cd Documents**", and we list out the hidden contents inside the directory. Inside the directory contains a file called "**e1fone.txt**". Then, we try reading the contents inside the file using "**cat e1fone.txt**" and it contains a message about " **All I want is my '2 front teeth'!!!**". Next, we head to the Desktop directory, and we list out the contents inside the Desktop directory. We found another hidden directory, so we search for the contents inside the elf2wo directory, and we saw this weird file with the naming "**e70smsW10Y4k.txt**". When we read the contents inside the file it reveals the movie named "**Scrooged**". Next , we want to list out the entire Windows directory to search for a hidden directory containing the third file. Once we listed out all the directory, we scroll through and found a hidden directory in **System32** called **3lfthr3e**. Then, we are required to find how many words the first file contains, so we used the command "**cat 1.txt | Measure-Object -Word**" and see that the file has 9999 words. After finding the number of words, we were asked to search for specific words in the file. With this we can use **"(cat 1.txt)[551,6991]**" to find the location of the word and what we found was "Red Ryder". Lastly, we want to search the second file for the phrases from the first file. What we did was "**Select-String 2.txt -Pattern 'redryder'**" and we got our final answer which is "Red Ryder bb Gun".