PSP0201

Week 5

Writeup

Group Name: Potatoes & Tomatoes

Members

ID	Name	Role
1211101125	Sayid Abdur-Rahman Al-Aidarus Bin Syed Abu Bakar Mashor Al-Idrus	Leader
1211101237	Mohammad Zulhilman Bin Mohd Hisham	Member
1211103699	Choo Qing Lam	Member
1211101234	Muhammad Zahin Adri	Member

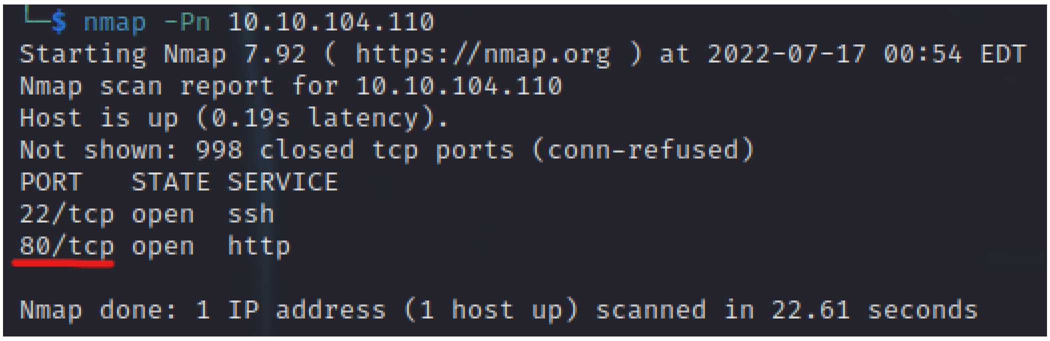
Day 16:

Tools used: Kali Linux (VirtualBox), Nmap, Firefox, Python3

Solution/walkthrough:

Question 1

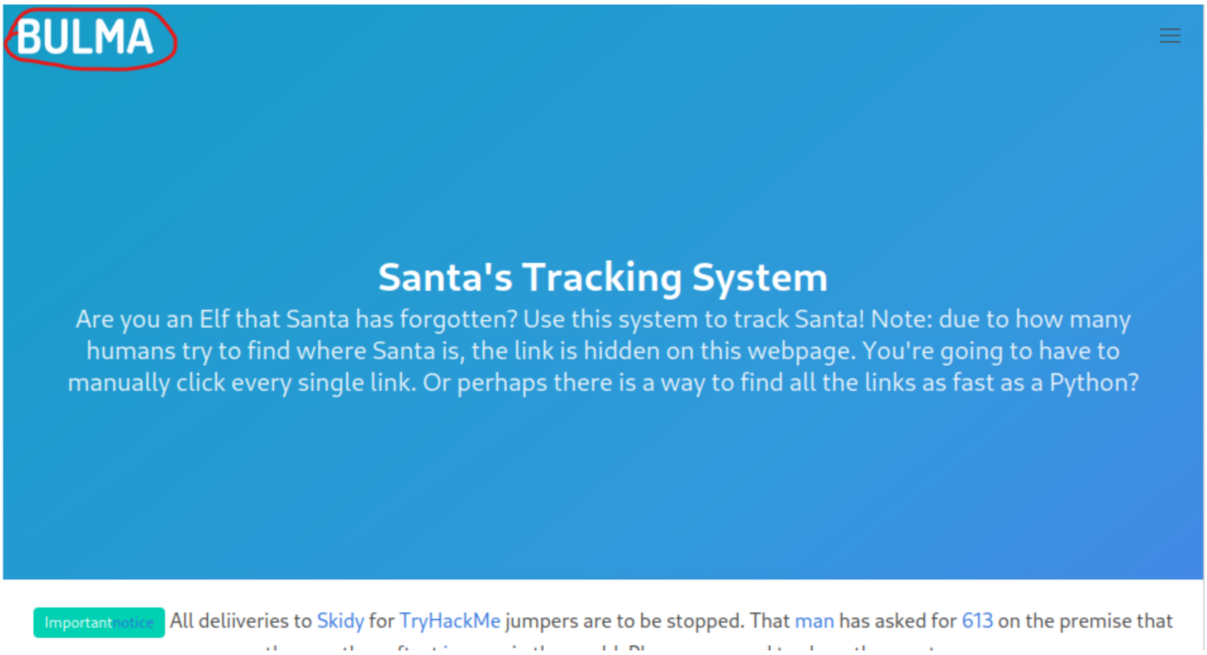
Use Nmap to scan the ports of the target machine to find the port of the web server.



```
└─$ nmap -Pn 10.10.104.110
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-17 00:54 EDT
Nmap scan report for 10.10.104.110
Host is up (0.19s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 22.61 seconds
```

Question 2

The template being used is located on the top left of the website page.



BULMA

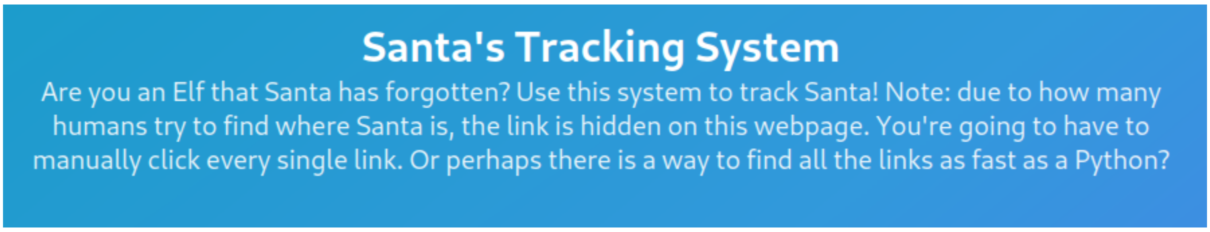
Santa's Tracking System

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

Question 3

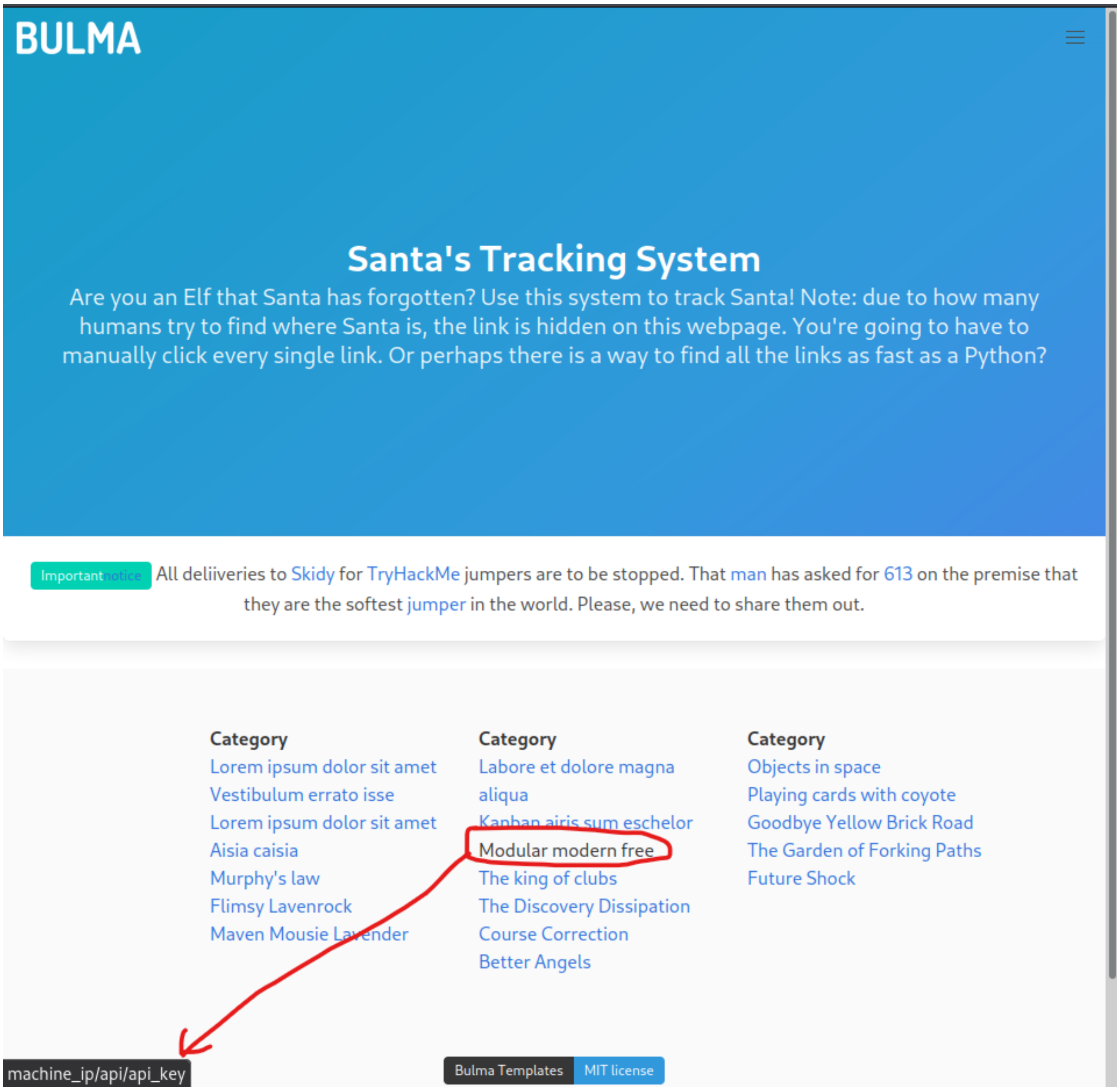
On the website, we get a clue on how to find the directory to the API.



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?

By looking at all the links on the page, we will be able to find the directory to the API. This specific link in the picture below reveals the directory to the API.



Santa's Tracking System

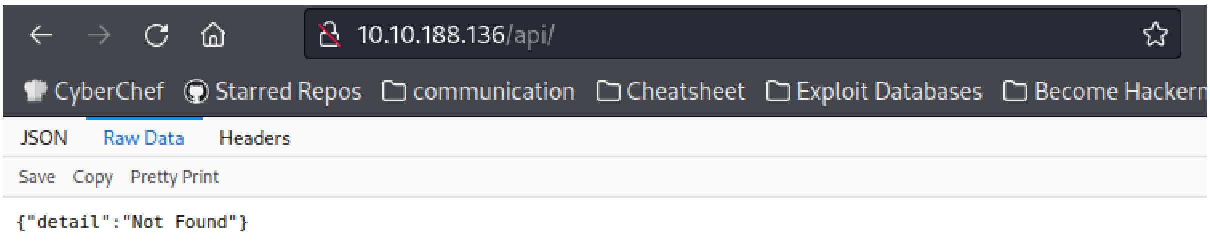
Are you an Elf that Santa has forgotten? Use this system to track Santa! Note: due to how many humans try to find where Santa is, the link is hidden on this webpage. You're going to have to manually click every single link. Or perhaps there is a way to find all the links as fast as a Python?

Importantnotice 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.

machine_ip/api/api_key

Question 4

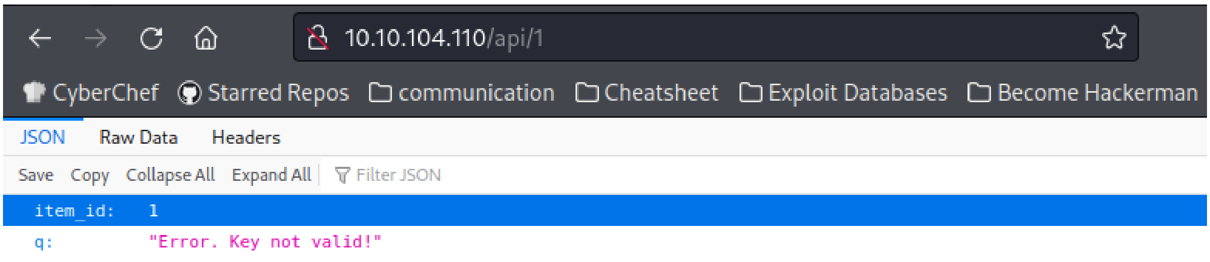
Open the link to the API directory without passing in an API key and look under the raw data tab.



```
{"detail":"Not Found"}
```

Question 5

We first try accessing the api directory using a random api key to see the output.

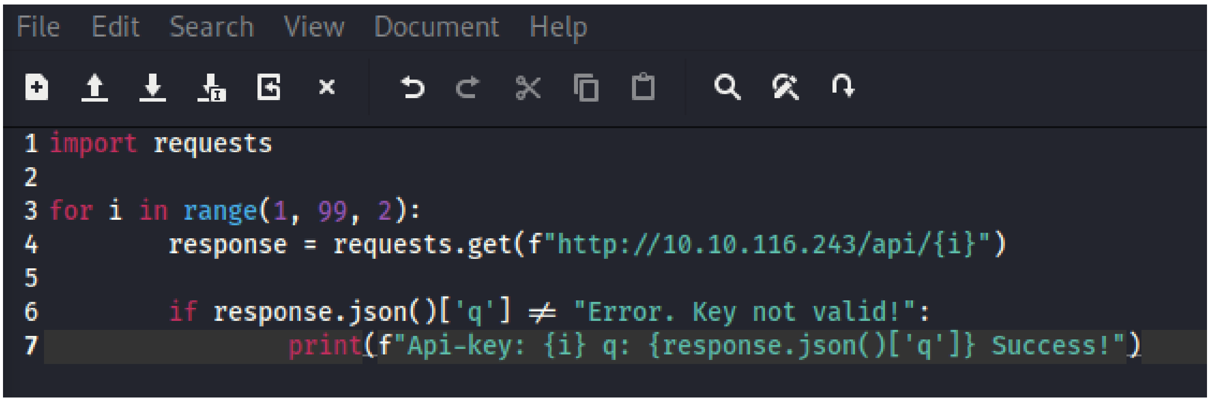


```
item_id:   1
q:         "Error. Key not valid!"
```

The output is in the form of JSON which we can parse using the python requests library.

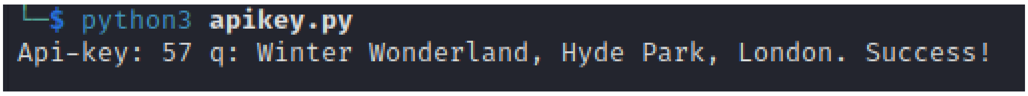
Now we have enough information to start writing the python script to find the api key.

A brief explanation of the script below, it will iterate through every odd number from 1 to 99 and use it as the api key when making a get request to the api directory. In the json of the response, if the value of ‘q’ is “Error. Key not valid!” then continue to the next iteration of the loop, but if the value of ‘q’ is something else then it will print out the api key used and the value of ‘q’.



```python
import requests

for i in range(1, 99, 2):
        response = requests.get(f"http://10.10.116.243/api/{i}")

        if response.json()['q'] != "Error. Key not valid!":
                print(f"Api-key: {i} q: {response.json()['q']} Success!")
```

The image below is the result after running the script.



```
└─$ python3 apikey.py
Api-key: 57 q: Winter Wonderland, Hyde Park, London. Success!
```

Thought Process/Methodology:

We first started by port scanning the machine to find the port number of the web server. We then opened the web page in firefox and used the hint on the page to find the directory the API is located in. After that, we created a python script to automate the process of iterating through all the possible API keys to find the right one. After finding the right API key, santa's location is revealed to us.

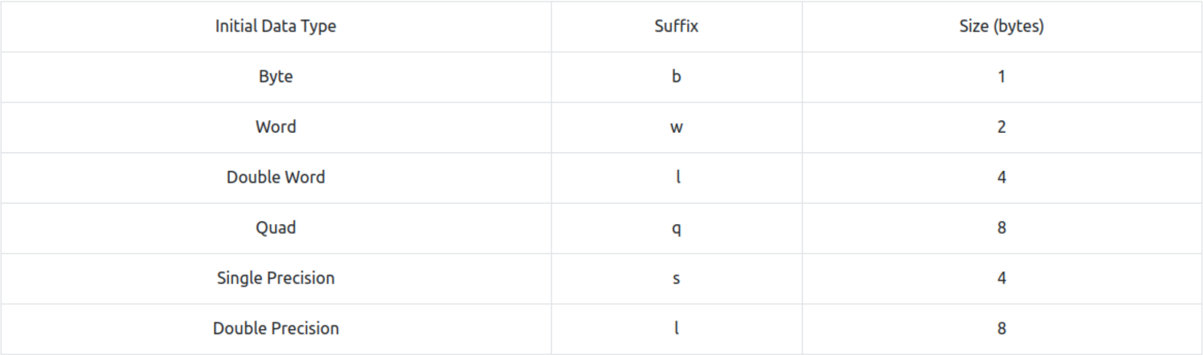
Day 17:

Tools used: *Kali Linux (VirtualBox), Radare2*

Solution/walkthrough:

Question 1

Just fill in the info from this table from THM



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

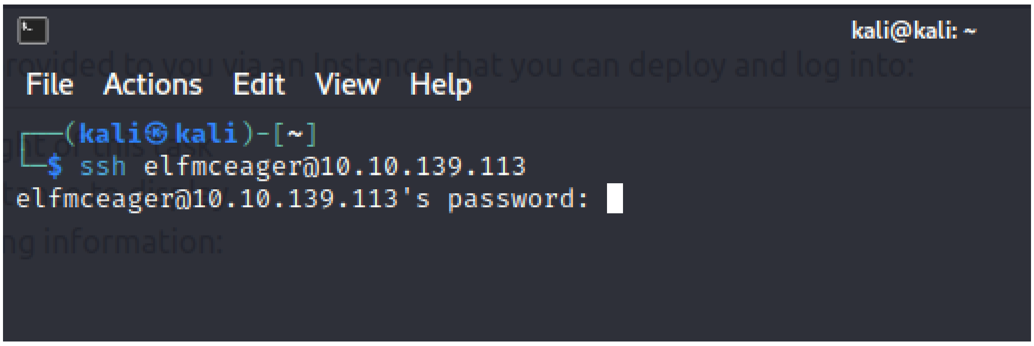
Questions 2-4

They can all be found by referring to the radare2 cheat sheet provided by THM

Questions 5-7

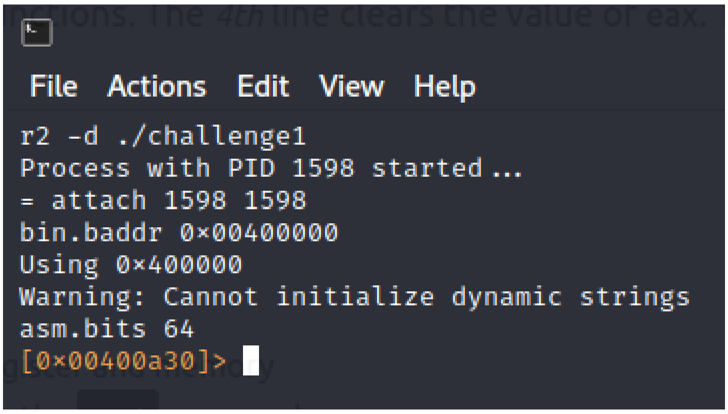
These questions have the same initial steps it's just a matter of understanding the info radare2 is giving

First start off by opening the Instance by inputting “ssh elfmceager@MACHINE_IP” mine was 10.10.139.113 so it’ll be “ssh elfmceager@10.10.139.113” then type in the password given



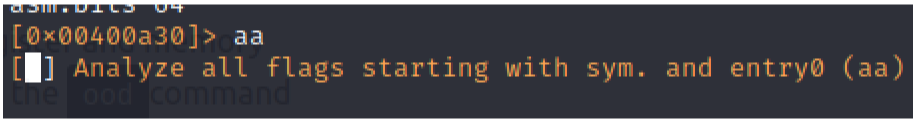
```
┌──(kali㉿kali)-[~]
└─$ ssh elfmceager@10.10.139.113
elfmceager@10.10.139.113's password: █
```

Then open up the challenge1 file in radare2 with “r2 -d ./challenge1”



```
r2 -d ./challenge1
Process with PID 1598 started...
= attach 1598 1598
bin.baddr 0×00400000
Using 0×400000
Warning: Cannot initialize dynamic strings
asm.bits 64
[0×00400a30]> 
```

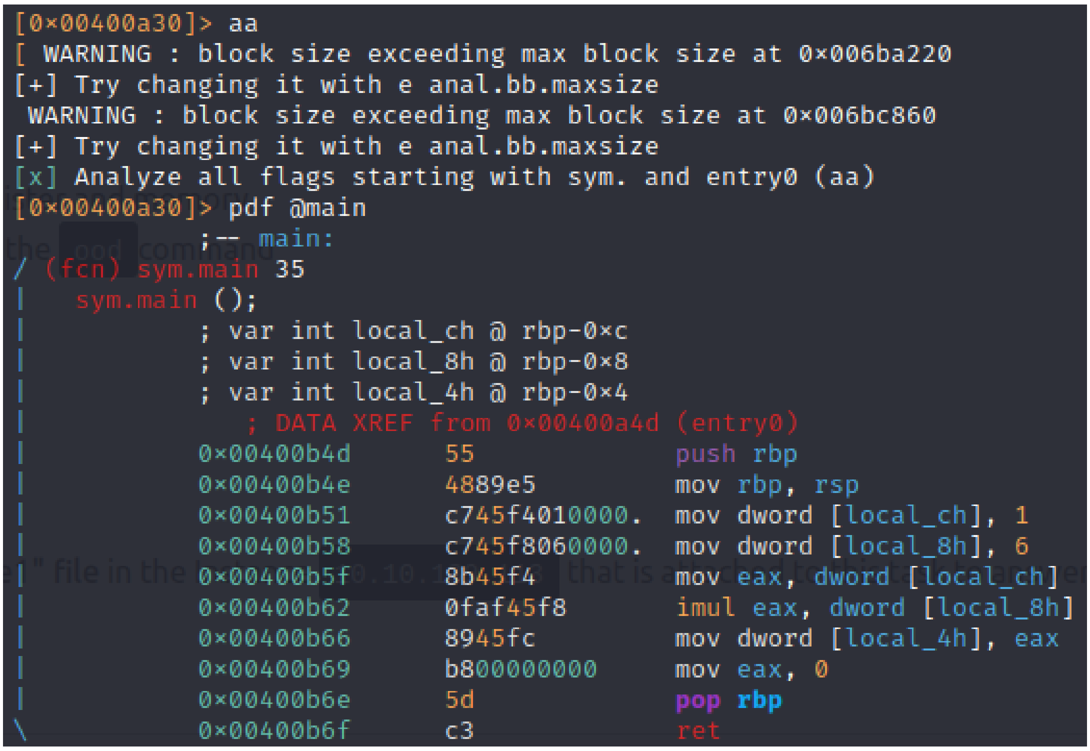
Proceed with running the “aa” command



```
[0×00400a30]> aa
[ ] Analyze all flags starting with sym. and entry0 (aa)
```

(this won’t take long but you have enough time for a bathroom break)

Now with that “aa” has ran, print out the disassembly of the main functions with “pdf @main”

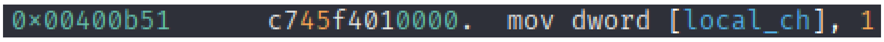


```
[0×00400a30]> aa
[ WARNING : block size exceeding max block size at 0×006ba220
[+] Try changing it with e anal.bb.maxsize
 WARNING : block size exceeding max block size at 0×006bc860
[+] Try changing it with e anal.bb.maxsize
[x] Analyze all flags starting with sym. and entry0 (aa)
[0×00400a30]>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

You can already get all the answers from these lines with some understanding of assembly

Question 5

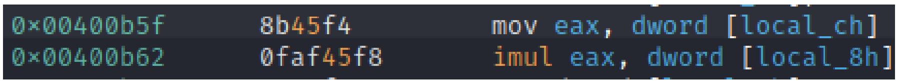
Mov simply copies the the value of the second operand “1” into the first operand “local_ch”



```
0x00400b51      c745f4010000.  mov dword [local_ch], 1
```

Question 6

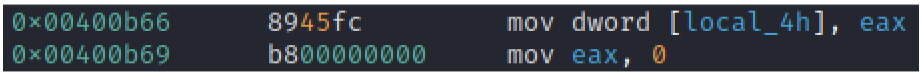
eax’s value becomes 1 in the first line then is multiplied (imul) with the second operand “local_8h” which has the value of 6



```
0x00400b5f      8b45f4         mov eax, dword [local_ch]
0x00400b62      0faf45f8       imul eax, dword [local_8h]
```

Question 7

Same concept as question 1



```
0x00400b66      8945fc         mov dword [local_4h], eax
0x00400b69      b800000000     mov eax, 0
```

Thought Process/Methodology:

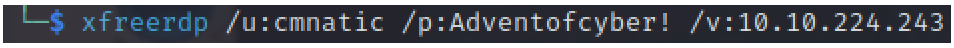
We log into the instance then open the binary in debugging mode. After doing a binary analysis we can open the file in disassembly code and figure out what data to use to find the answers.

Day 18:

***Tools used:** Kali Linux, ILspy, Freerdp2*

Solution/walkthrough:

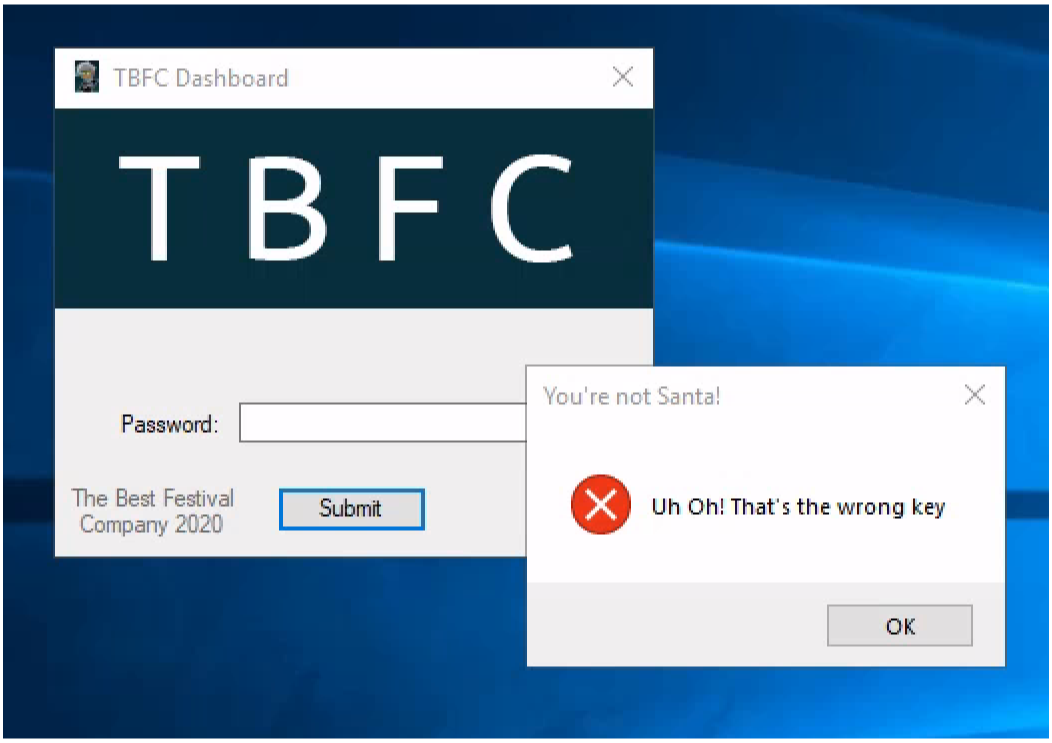
First of all, we connect to the instance using Remote Desktop Protocol (RDP). We connected using Freerdp2 with the following command:



```
└─$ xfreerdp /u:cmnatic /p:Adventofcyber! /v:10.10.224.243
```

Question 1

Launch TBFC_APP and attempt a login.

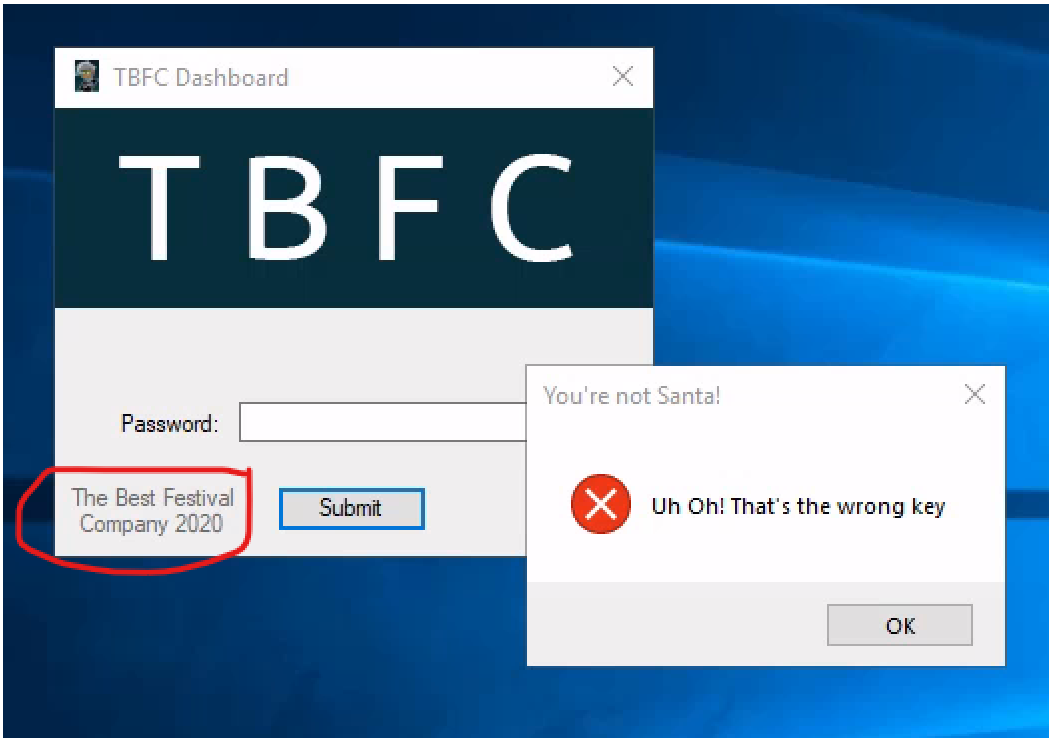


You're not Santa!

Uh Oh! That's the wrong key

Question 2

TBFC means The Best Festival Company as seen in the bottom left of the application UI

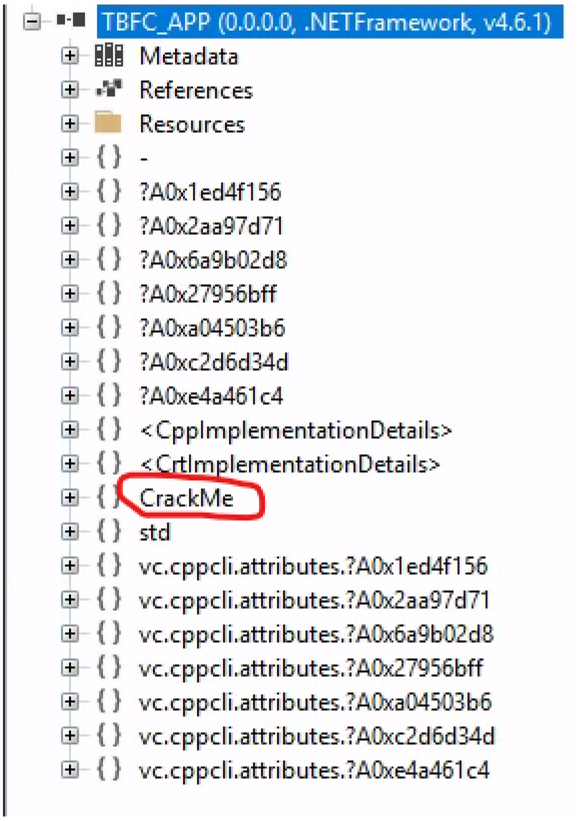


You're not Santa!

Uh Oh! That's the wrong key

Question 3

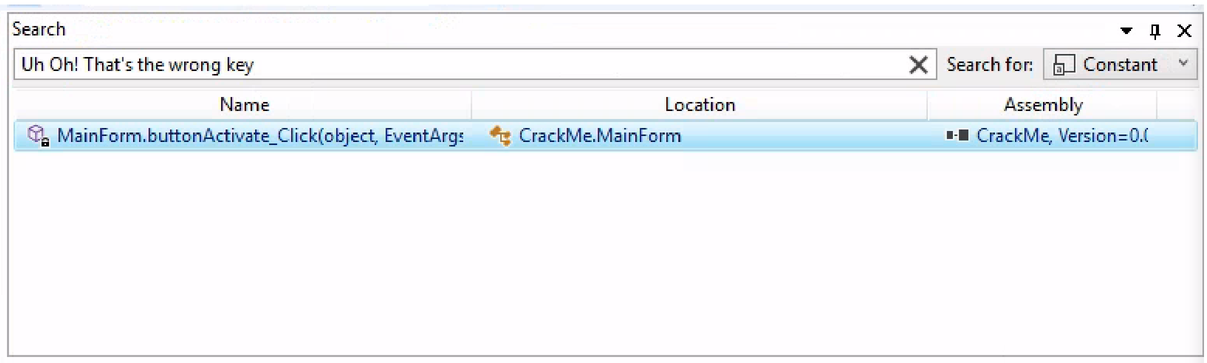
The name of the module circled in red is very eye catching.



```
TBFC_APP (0.0.0.0, .NETFramework, v4.6.1)
    Metadata
    References
    Resources
    { } -
    { } ?A0x1ed4f156
    { } ?A0x2aa97d71
    { } ?A0x6a9b02d8
    { } ?A0x27956bff
    { } ?A0xa04503b6
    { } ?A0xc2d6d34d
    { } ?A0xe4a461c4
    { } <CppImplementationDetails>
    { } <CrtImplementationDetails>
    { } CrackMe
    { } std
    { } vc.cppcli.attributes.?A0x1ed4f156
    { } vc.cppcli.attributes.?A0x2aa97d71
    { } vc.cppcli.attributes.?A0x6a9b02d8
    { } vc.cppcli.attributes.?A0x27956bff
    { } vc.cppcli.attributes.?A0xa04503b6
    { } vc.cppcli.attributes.?A0xc2d6d34d
    { } vc.cppcli.attributes.?A0xe4a461c4
```

Question 4 & 5

We can search the term “Uh Oh! That's the wrong key” to find where the code that handles the login is located.

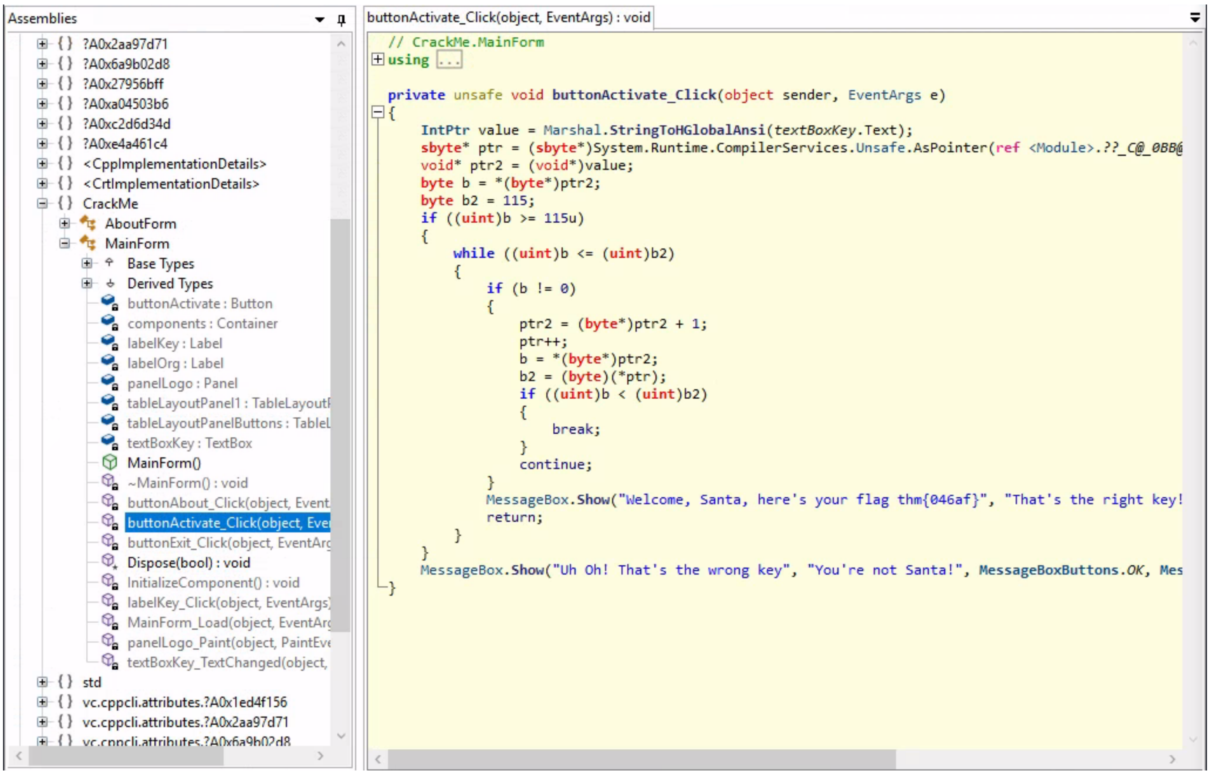


Uh Oh! That's the wrong key

Search for: Constant

Name	Location	Assembly
MainForm.buttonActivate_Click(object, EventArgs	CrackMe.MainForm	CrackMe, Version=0.(

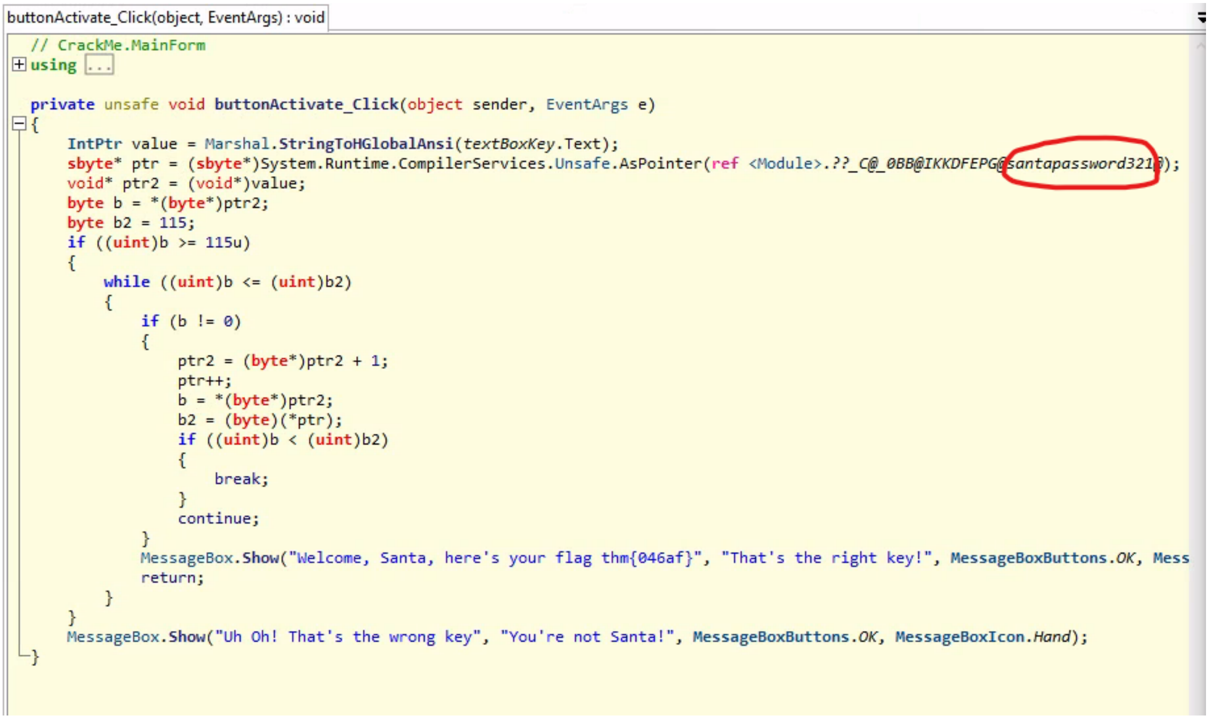
Looking into our search result, Mainform contains a function called “buttonActivate_Click” which is the function that handles the login.


Assemblies
{ } ?A0x2aa97d71
{ } ?A0x6a9b02d8
{ } ?A0x27956bff
{ } ?A0xa04503b6
{ } ?A0xc2d6d34d
{ } ?A0xe4a461c4
{ } <CppImplementationDetails>
{ } <CrtImplementationDetails>
{ } CrackMe
AboutForm
MainForm
Base Types
Derived Types
buttonActivate : Button
components : Container
labelKey : Label
labelOrg : Label
panelLogo : Panel
tableLayoutPanel1 : TableLayoutP
tableLayoutPanelButtons : TableL
textBoxKey : TextBox
MainForm()
~MainForm() : void
buttonAbout_Click(object, Event
buttonActivate_Click(object, Eve
buttonExit_Click(object, EventArg
Dispose(bool) : void
InitializeComponent() : void
labelKey_Click(object, EventArgs)
MainForm_Load(object, EventArg
panelLogo_Paint(object, PaintEve
textBoxKey_TextChanged(object,
{ } std
{ } vc.cppcli.attributes.?A0x1ed4f156
{ } vc.cppcli.attributes.?A0x2aa97d71
{ } vc.cppcli.attributes.?A0x6a9b02d8
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@
    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!
            return;
        }
    }
    MessageBox.Show("Uh Oh! That's the wrong key", "You're not Santa!", MessageBoxButtons.OK, Mes
}
```

Question 6

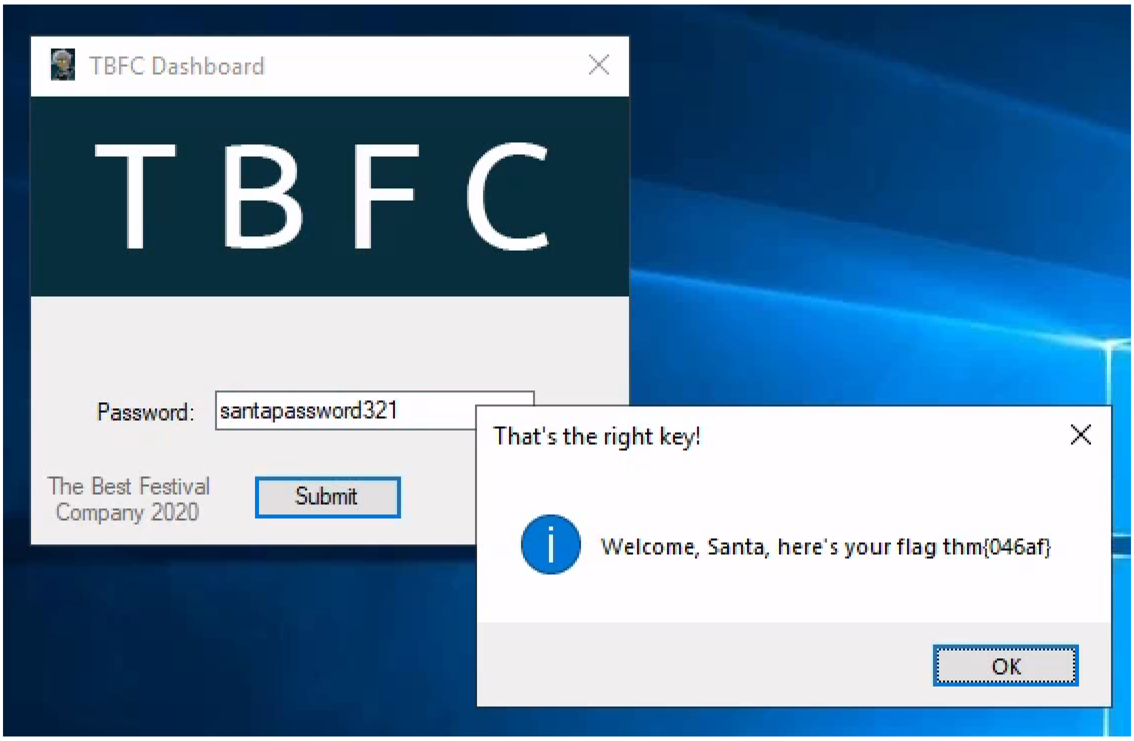
The login password (circled in red) can be seen in the code of the buttonActivate_Click function.



```
// 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, Mess
            return;
        }
    }
    MessageBox.Show("Uh Oh! That's the wrong key", "You're not Santa!", MessageBoxButtons.OK, MessageBoxIcon.Hand);
}
```

Question 7

Login using the password found earlier and a pop up will appear revealing the flag.



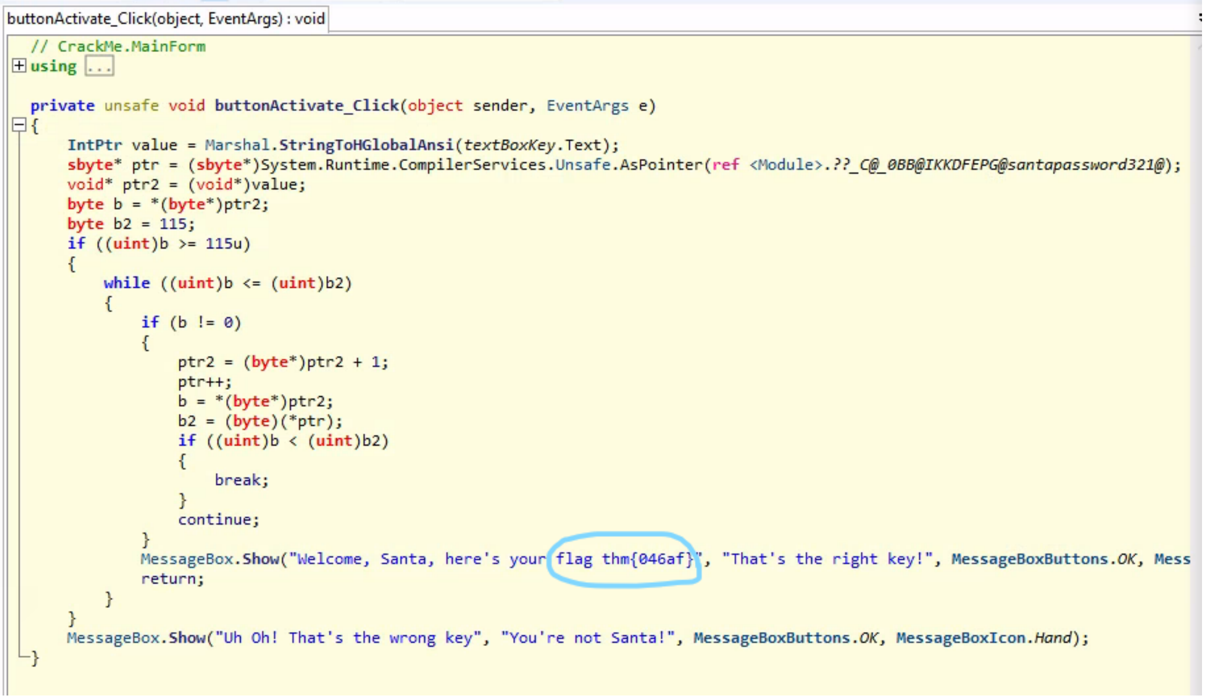
The Best Festival Company 2020

Password: santapassword321

That's the right key!

Welcome, Santa, here's your flag thm{046af}

Note: The flag can also be found in the code snippet that handles the login from earlier (circled in blue in the image below).



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, Mess
            return;
        }
    }
    MessageBox.Show("Uh Oh! That's the wrong key", "You're not Santa!", MessageBoxButtons.OK, MessageBoxIcon.Hand);
}
```

Thought Process/Methodology:

After logging into the instance using RDP, we began decompiling the code of the “TBFC_APP” application using ILspy. After attempting a failed login, we used the text that appeared in the pop up from the failed login to search through the decompiled code for the code snippet that handles the login. By analysing the code, we find Santa's password to login to the application. After a successful login, we’re prompted with the flag to complete the task in a pop up.

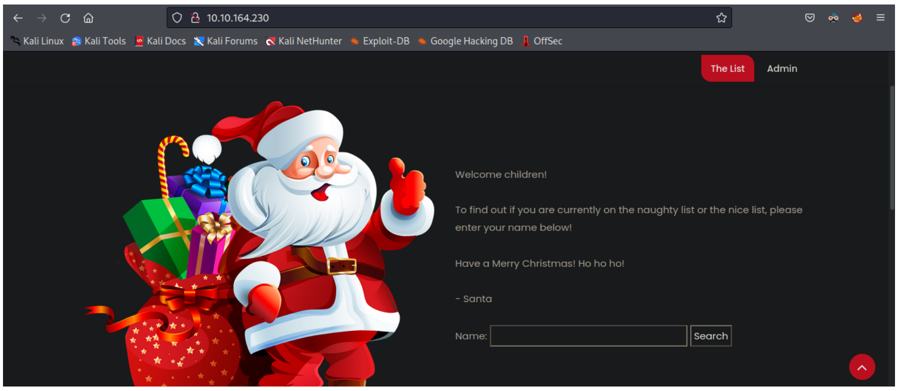
Day 19:

***Tools used:** Kali Linux (VirtualBox), Firefox, SSRF*

Solution/walkthrough:

Step 1

First, sign in with the machineIPaddress that is given in the start machine box that is 10.10.164.230 for the web.



Welcome children!

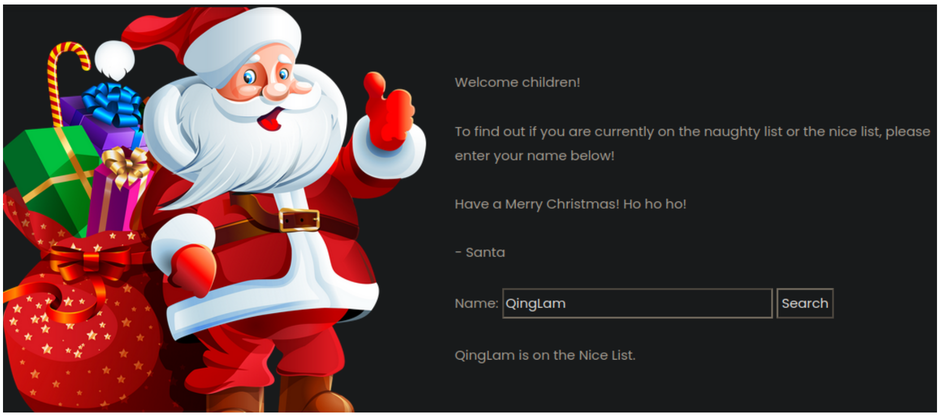
To find out if you are currently on the naughty list or the nice list, please enter your name below!

Have a Merry Christmas! Ho ho ho!

- Santa

Name: Search

Next up try using the search tag to see if your name is on the nice list or not



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!

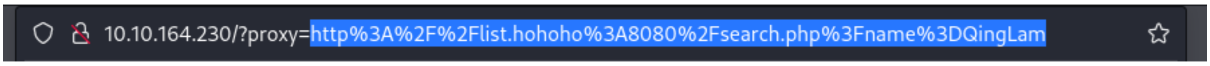
- Santa

Name: QingLam Search

QingLam is on the Nice List.

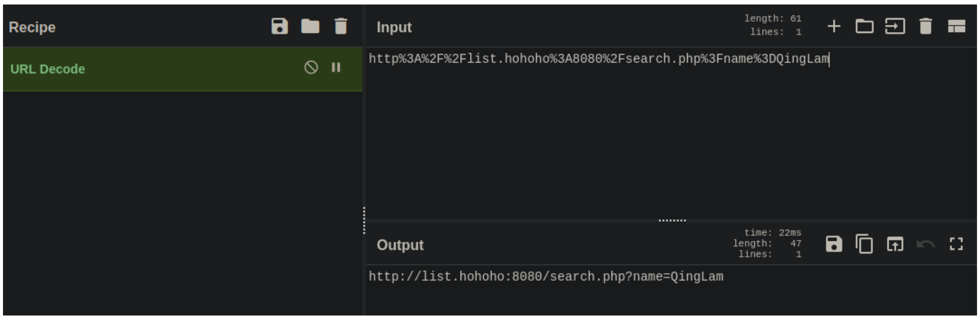
Step 2

Copy the URL part



```
10.10.164.230/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%3DQingLam
```

to translate in url decoder in cyberchef, that will lead us to



length: 61
lines: 1

```
http%3A%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%3DQingLam
```

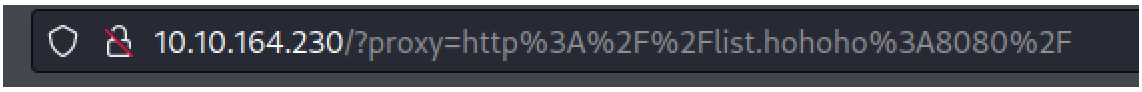
time: 22ms
length: 47
lines: 1

```
http://list.hohoho:8080/search.php?name=QingLam
```

Step 3

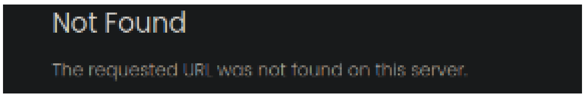
Next try using different root and port numbers

Such as changing the url without using the search.php



```
10.10.164.230/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F
```

The result you will get is Not Found

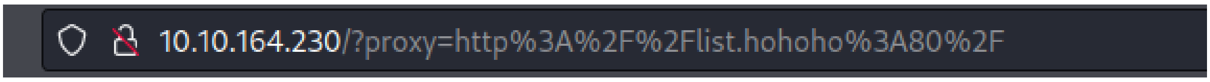


Not Found

The requested URL was not found on this server.

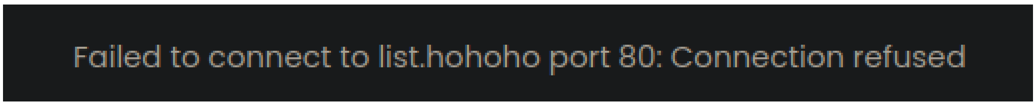
Step 4

We can try is changing the port numbers into 80 instead of 8080



```
10.10.164.230/?proxy=http%3A%2F%2Flist.hohoho%3A80%2F
```

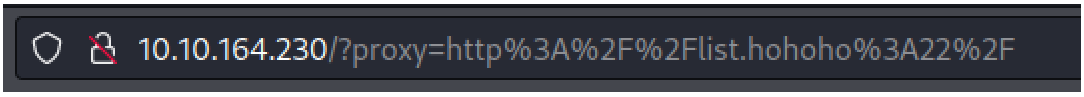
As you can see its the same it will be failed to connect to list



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

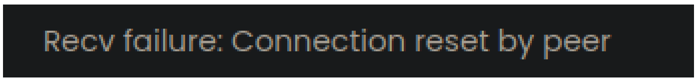
Step 5

Another thing we can try is to change the port number into 22 instead of 8080



```
10.10.164.230/?proxy=http%3A%2F%2Flist.hohoho%3A22%2F
```

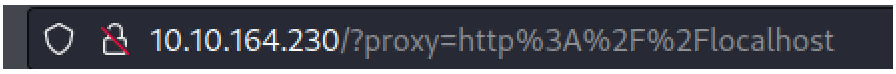
You will receive this Recv Failure



Recv failure: Connection reset by peer

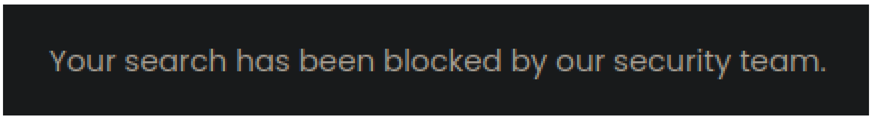
Step 6

We can try out the SSRF that is localhost and 127.0.0.1



```
10.10.164.230/?proxy=http%3A%2F%2Flocalhost
```

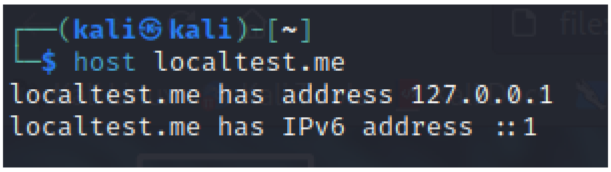
Both of these SSRF is blocked by the security team



Your search has been blocked by our security team.

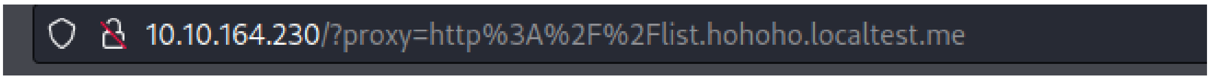
Step 7

As you type in host localtest.me on your terminal it will pop out 127.0.0.1 as your address



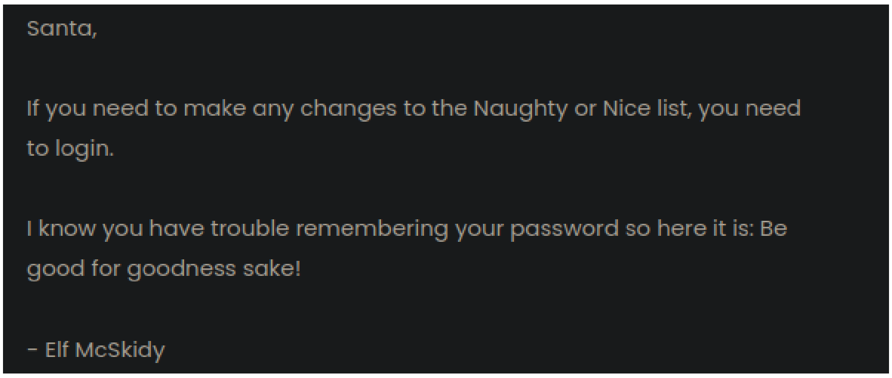
```
┌──(kali㉿kali)-[~]
└─$ host localtest.me
localtest.me has address 127.0.0.1
localtest.me has IPv6 address ::1
```

After typing localtest.me behind hohoho



```
10.10.164.230/?proxy=http%3A%2F%2Flist.hohoho.localtest.me
```

You will get a message from Elf McSkidy



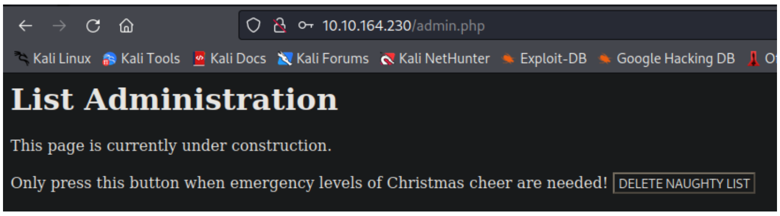
Santa,

If you need to make any changes to the Naughty or Nice list, you need to login.

I know you have trouble remembering your password so here it is: Be good for goodness sake!

- Elf McSkidy

We will try out the Login page for the admin by putting in santa as username and Be good for goodness sake! As the password and your in!



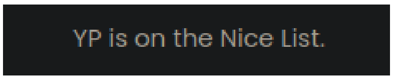
List Administration

This page is currently under construction.

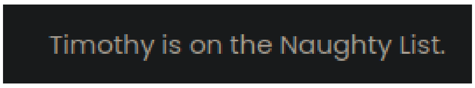
Only press this button when emergency levels of Christmas cheer are needed! DELETE NAUGHTY LIST

Question 1

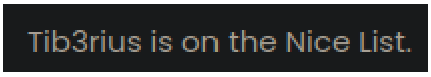
By using the step 1.2 we used above we can easily separate all of the members on list to see who is naughty or nice.



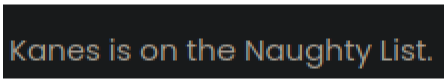
YP is on the Nice List.



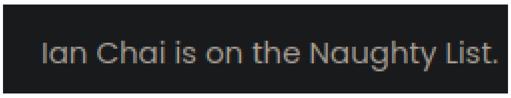
Timothy is on the Naughty List.



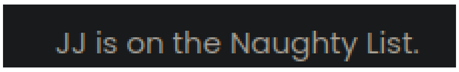
Tib3rius is on the Nice List.



Kanes is on the Naughty List.



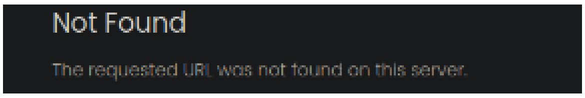
Ian Chai is on the Naughty List.



JJ is on the Naughty List.

Question 2

This is what you will get when you change “/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F

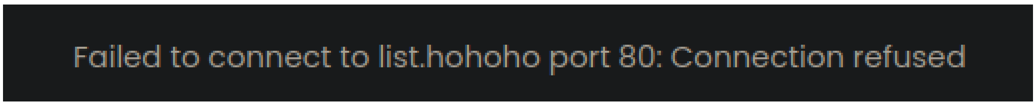


Not Found

The requested URL was not found on this server.

Question 3

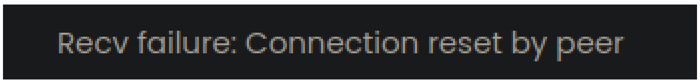
This is what you will get when you change "/?proxy=http%3A%2F%2Flist.hohoho%3A80"



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

Question 4

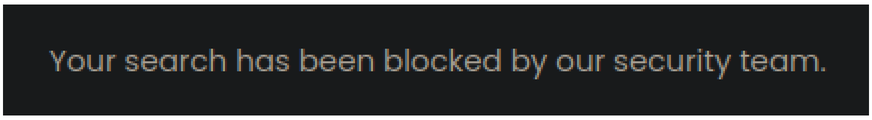
This is what you will get when you change proxy=http%3A%2F%2Flist.hohoho%3A22"



Recv failure: Connection reset by peer

Question 5

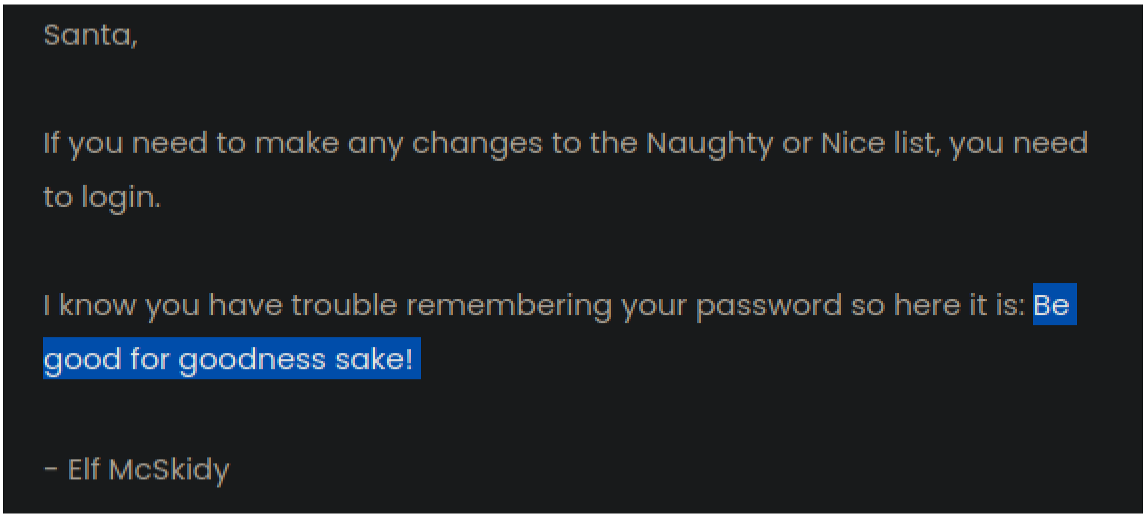
This is what you will get when you change "/?proxy=http%3A%2F%2Flocalhost"?



Your search has been blocked by our security team.

Question 6

For the password for Question 1 would be Be good for goodness sake! That is given by Elf McSkidy in the step 7 part as we use list.hohoho.localhost.me



Santa,

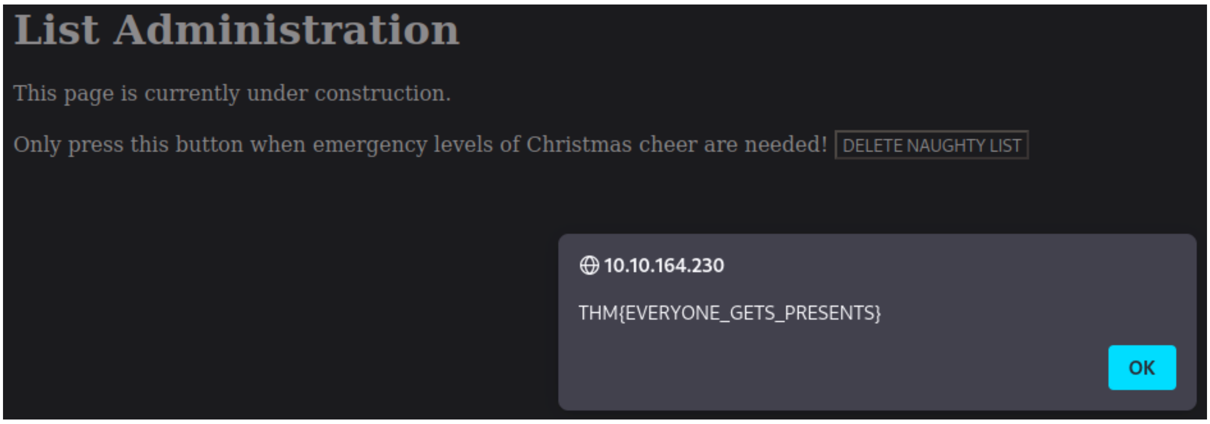
If you need to make any changes to the Naughty or Nice list, you need to login.

I know you have trouble remembering your password so here it is: Be good for goodness sake!

- Elf McSkidy

Question 7

After entering the admin page you will see a delete naugthy list button by clicking on in it will show you the tag for the answer for the flag



List Administration

This page is currently under construction.

Only press this button when emergency levels of Christmas cheer are needed! DELETE NAUGHTY LIST

10.10.164.230

THM{EVERYONE_GETS_PRESENTS}

Thought Process/Methodology

We try out a lot of ways doing this by using every single step in the walkthrough to find the solution and the answers for the questions. We learned that SSRF is the most useful for attackers to force web applications. Nevertheless after finding out the answer in one of the steps makes us realize that by doing every step that is included in the walkthrough we will get the answer in point.

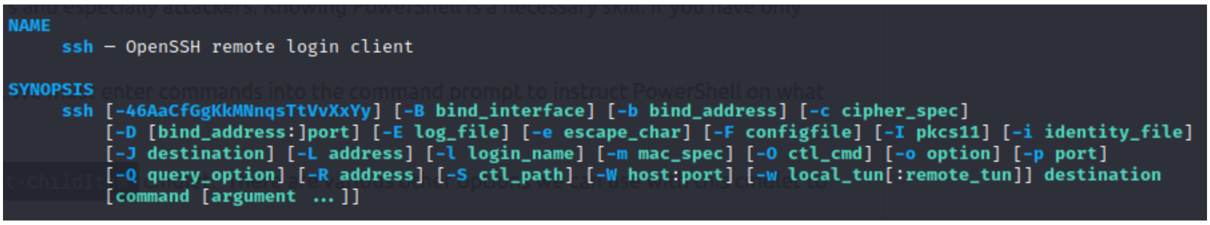
Day 20:

***Tools used:** Kali Linux (VirtualBox), Powershell*

Solution/walkthrough:

Question 1

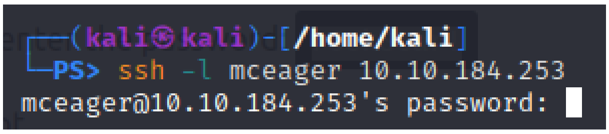
Input “man ssh” in a terminal



```
NAME
       ssh – OpenSSH remote login client

SYNOPSIS
       ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface] [-b bind_address] [-c cipher_spec]
           [-D [bind_address:]port] [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11] [-i identity_file]
           [-J destination] [-L address] [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]] destination
           [command [argument ...]]
```

Before we do the rest of the questions we must set up a couple things first

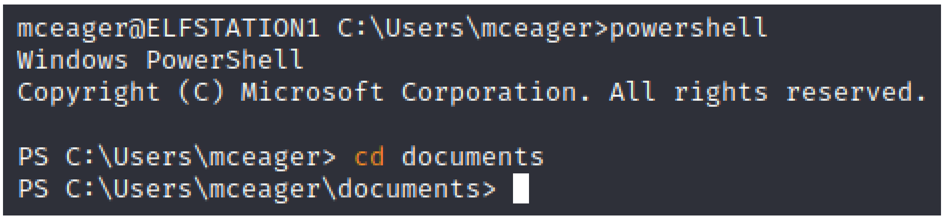
Start by connecting with the remote machine over SSH with “ssh -l mceager MACHINE_IP” and entering McEager’s password “r0ckStar!”



```
┌──(kali㉿kali)-[/home/kali]
└─PS> ssh -l mceager 10.10.184.253
mceager@10.10.184.253's password: 
```

(it might take a little while for the password prompt to appear)

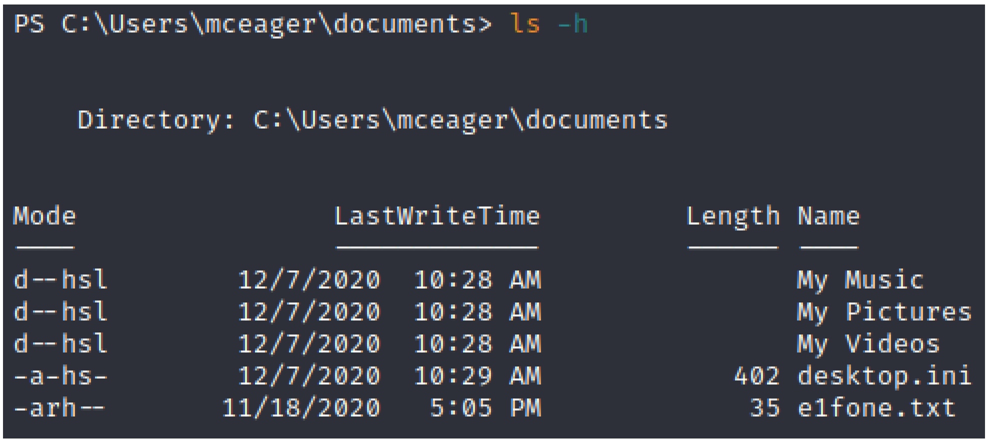
Go into powershell by inputting “powershell”, then change the directory to /documents with “cd documents”



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

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

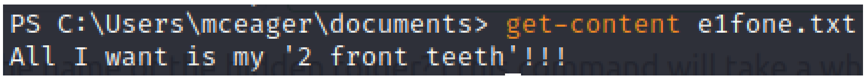
Question 2

View the hidden items in the directory with “ls -h”



```
PS C:\Users\mceager\documents> ls -h


    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
```

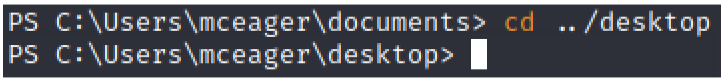
As we can see the non essential hidden file there is e1fone.txt so now we just view its contents with “get-content e1fone.txt”



```
PS C:\Users\mceager\documents> get-content e1fone.txt
All I want is my '2 front teeth'!!!
```

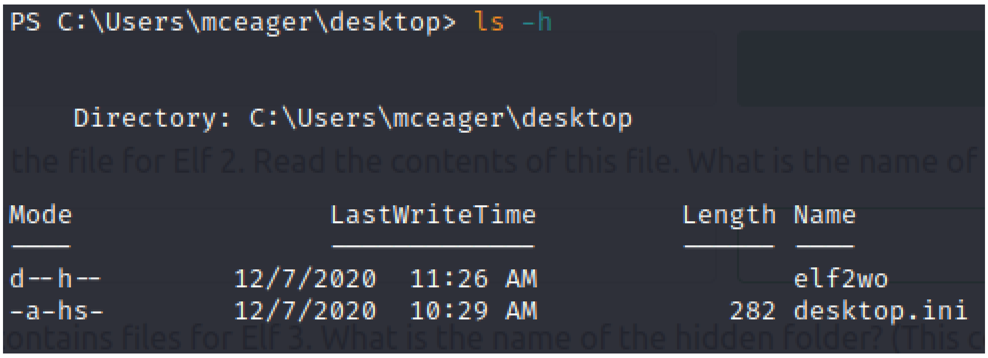
Question 3

Get out of the /documents folder and into the desktop folder with “cd ../desktop”



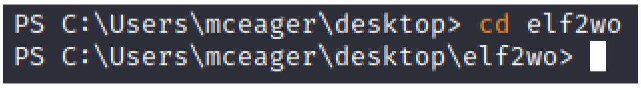
```
PS C:\Users\mceager\documents> cd ../desktop
PS C:\Users\mceager\desktop>
```

View hidden file with “ls -h”



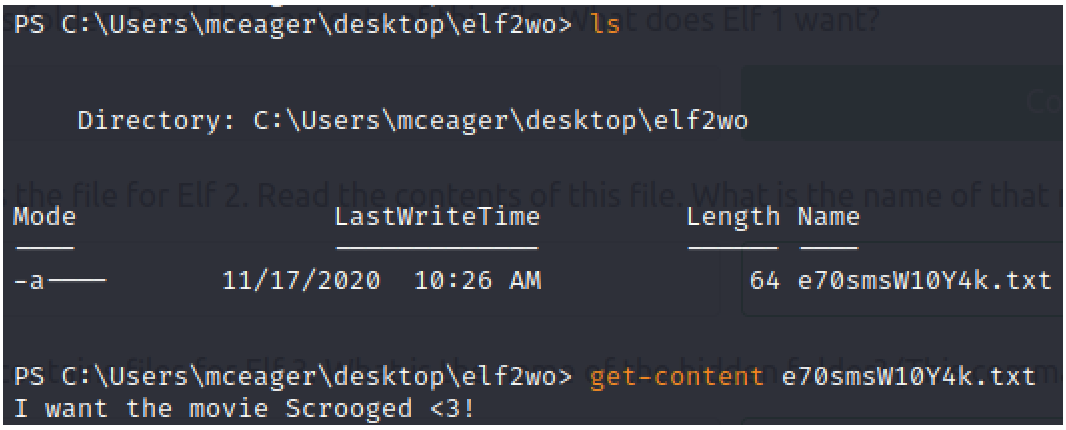
```
PS C:\Users\mceager\desktop> ls -h


    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 go into the hidden file with “cd elf2wo”



```
PS C:\Users\mceager\desktop> cd elf2wo
PS C:\Users\mceager\desktop\elf2wo>
```

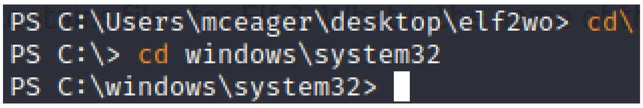
View the contents of txt file inside as the same way with question 1



```
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!
```

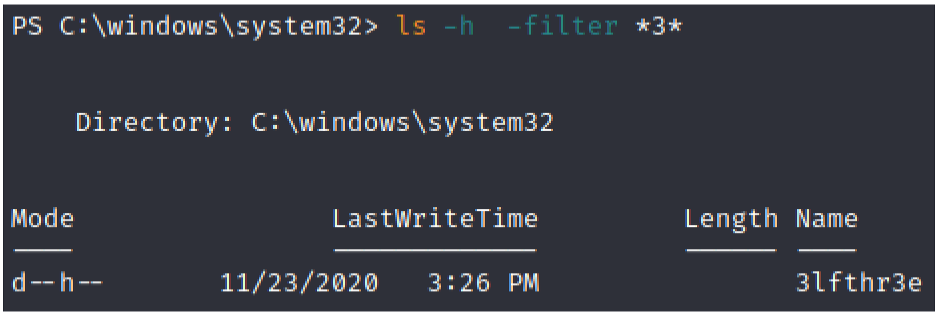
Question 4

Get of of the current directory with “cd\” and get into C:\windows\system32 with “cd windows\system32”



```
PS C:\Users\mceager\desktop\elf2wo> cd\
PS C:\> cd windows\system32
PS C:\windows\system32>
```

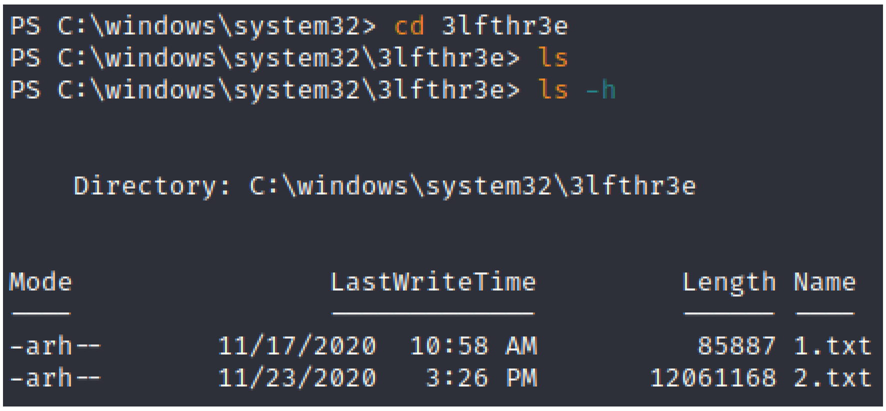
Then find the hidden file with “ls -h -filter *3*”



```
PS C:\windows\system32> ls -h  -filter *3*


    Directory: C:\windows\system32


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

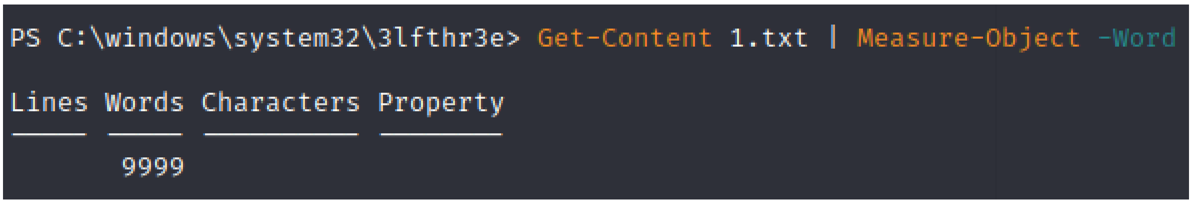
Question 5

Get into the file and view contents of the file with “ls -h”



```
PS C:\windows\system32> cd 3lfthr3e
PS C:\windows\system32\3lfthr3e> ls
PS C:\windows\system32\3lfthr3e> ls -h


    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
```

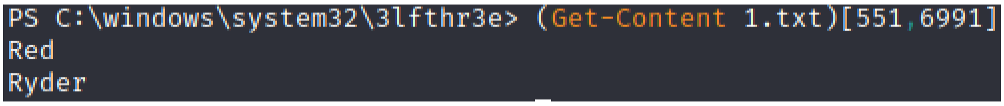
View the word count of the first file with “Get-Content 1.txt | Measure-Object -Word”



```
PS C:\windows\system32\3lfthr3e> Get-Content 1.txt | Measure-Object -Word

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

Question 6

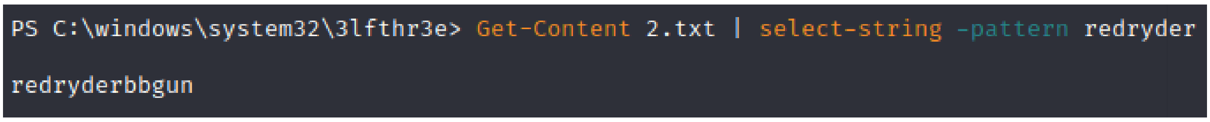
Find the words at index 551 and index 6991 with “(Get-Content 1.txt)[551,6991]”



```
PS C:\windows\system32\3lfthr3e> (Get-Content 1.txt)[551,6991]
Red
Ryder
```

Question 7

Find the second half of the phrase with “Get-Content 2.txt | select-string -pattern redryder”



```
PS C:\windows\system32\3lfthr3e> Get-Content 2.txt | select-string -pattern redryder

redryderbbgun
```

Thought Process/Methodology:

With basic understanding of powershell, we are able to utilise a couple commands for us to collect information from the files we needed to access the wanted gifts of all 3 elfs and their respective directories.