PSP0201

Week 2 Writeup

Group name: SOLO

ID	NAME	ROLE
1211100574	Ivan Liew Qi Hong	leader

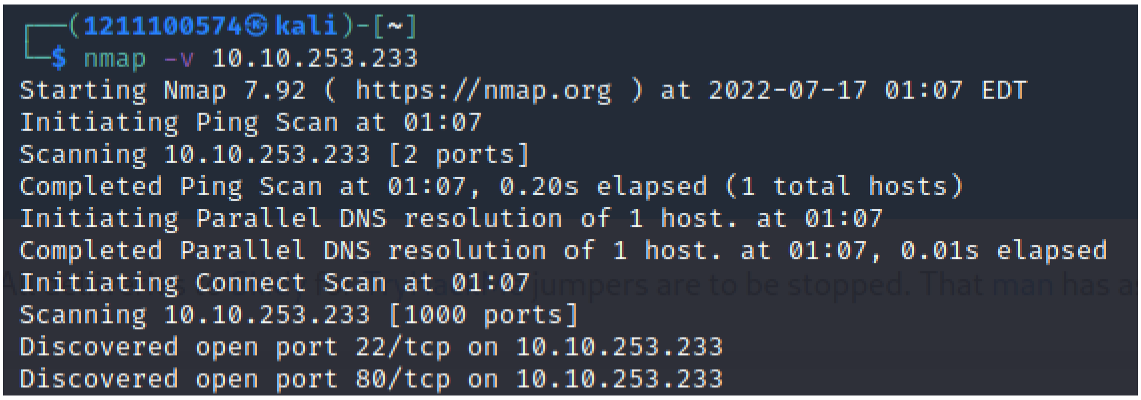
Day 16: [Scripting] Help! Where is Santa?

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

Question 1

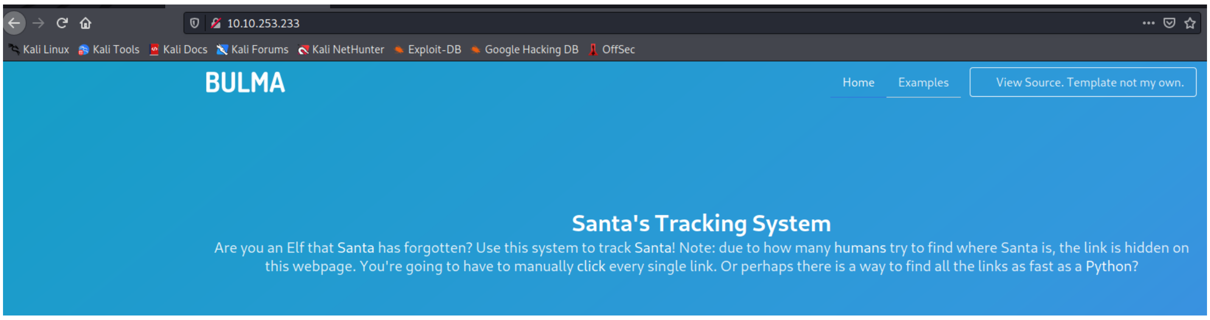
Use nmap to find open ports and use them to locate website with ip address



```
┌──(1211100574㉿kali)-[~]
└─$ nmap -v 10.10.253.233
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-17 01:07 EDT
Initiating Ping Scan at 01:07
Scanning 10.10.253.233 [2 ports]
Completed Ping Scan at 01:07, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:07
Completed Parallel DNS resolution of 1 host. at 01:07, 0.01s elapsed
Initiating Connect Scan at 01:07
Scanning 10.10.253.233 [1000 ports]
Discovered open port 22/tcp on 10.10.253.233
Discovered open port 80/tcp on 10.10.253.233
```

Question 2

Look at the top right of the webpage

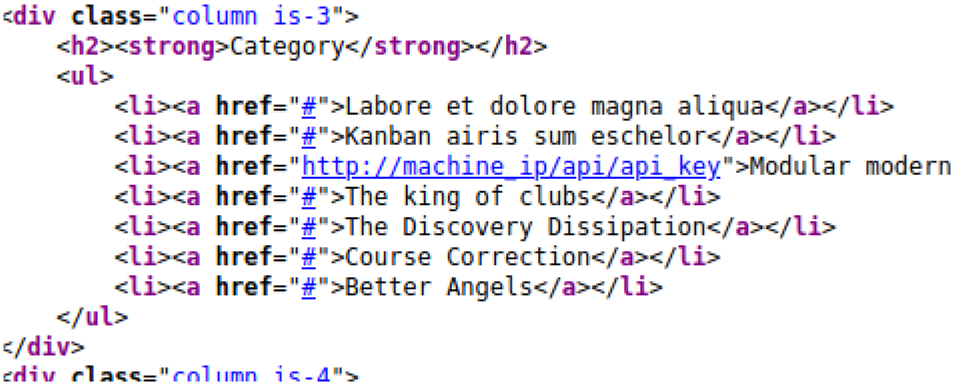


Santa's Tracking System

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?

Question 3

View page source to find directory of API

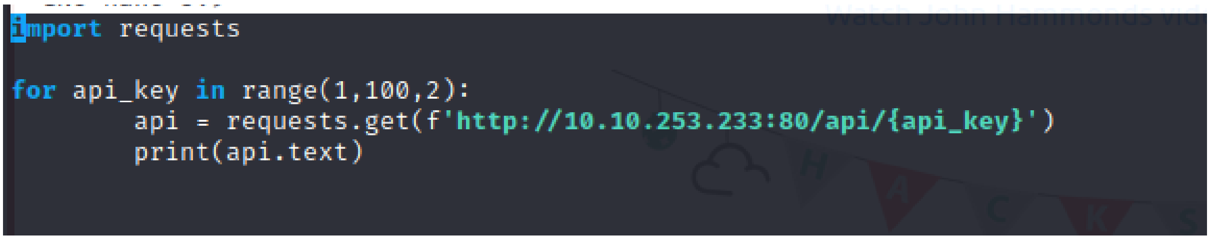


```
<div class="column is-3">
    <h2><strong>Category</strong></h2>
    <ul>
        <li><a href="#">Labore et dolore magna aliqua</a></li>
        <li><a href="#">Kanban airis sum eschelor</a></li>
        <li><a href="http://machine_ip/api/api_key">Modular modern
        <li><a href="#">The king of clubs</a></li>
        <li><a href="#">The Discovery Dissipation</a></li>
        <li><a href="#">Course Correction</a></li>
        <li><a href="#">Better Angels</a></li>
    </ul>
</div>
<div class="column is-4">
```

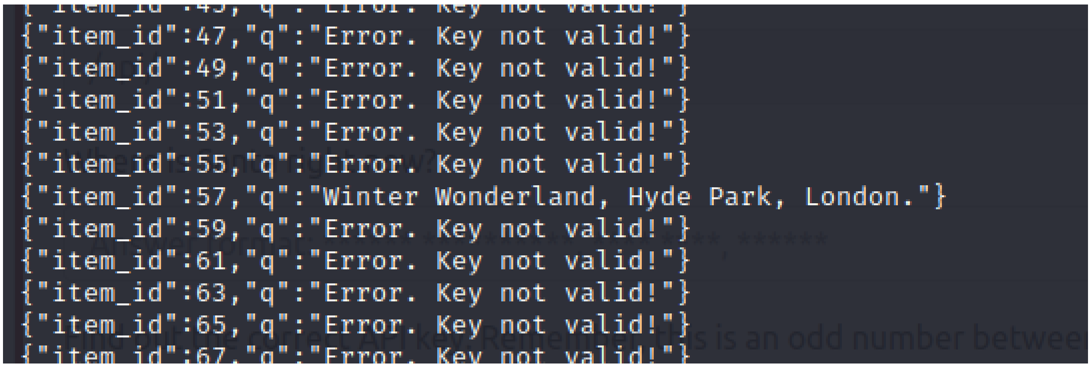
Question 4

Question 5

Use range in python to brute force through every possible answer to get the location



```
import requests

for api_key in range(1,100,2):
        api = requests.get(f'http://10.10.253.233:80/api/{api_key}')
        print(api.text)
```



```
{"item_id":45,"q":"Error. Key not valid!"}
{"item_id":47,"q":"Error. Key not valid!"}
{"item_id":49,"q":"Error. Key not valid!"}
{"item_id":51,"q":"Error. Key not valid!"}
{"item_id":53,"q":"Error. Key not valid!"}
{"item_id":55,"q":"Error. Key not valid!"}
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
{"item_id":59,"q":"Error. Key not valid!"}
{"item_id":61,"q":"Error. Key not valid!"}
{"item_id":63,"q":"Error. Key not valid!"}
{"item_id":65,"q":"Error. Key not valid!"}
{"item_id":67,"q":"Error. Key not valid!"}
```

Question 6

The number is the item id from the previous question

Thought Process/Methodology:

I nmapped the ip address to find the open ports that may be used to hold the webpage.I then found the webpage template at the top right of the web page. I followed it by inspecting the page source to find the api directory. I then used a python command the go through all possible numbers to find the right one as well as the location.

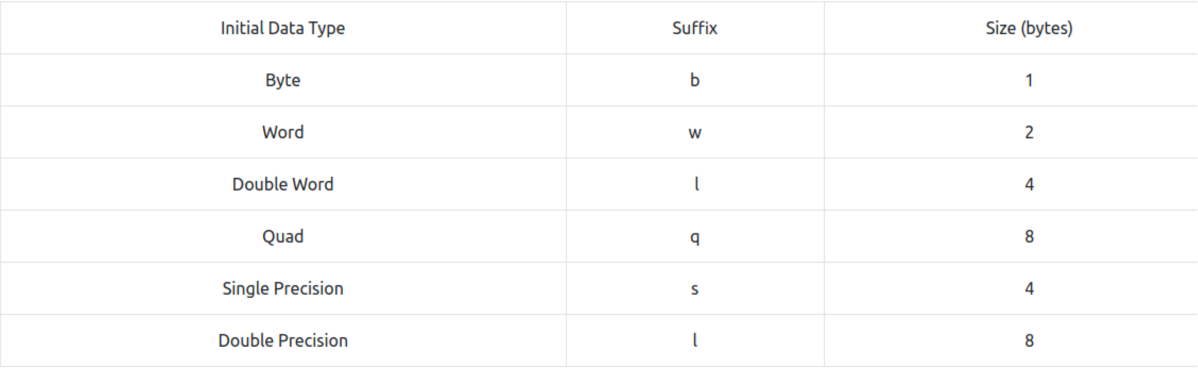
Day 17: [Reverse Engineering] ReverseELFneering

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

Question 1

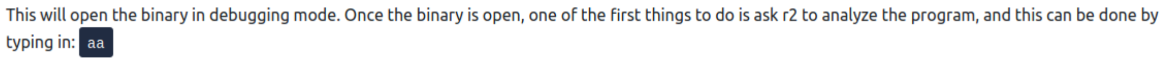
Follow the table



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

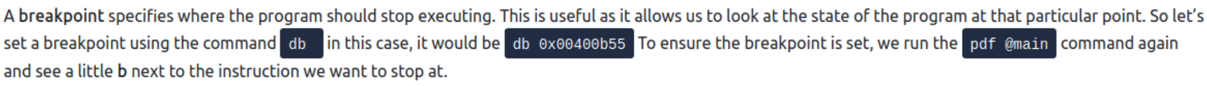
In the question



This will open the binary in debugging mode. Once the binary is open, one of the first things to do is ask r2 to analyze the program, and this can be done by typing in: `aa`

Question 3

In the question



A **breakpoint** specifies where the program should stop executing. This is useful as it allows us to look at the state of the program at that particular point. So let's set a breakpoint using the command `db` in this case, it would be `db 0x00400b55` To ensure the breakpoint is set, we run the `pdf @main` command again and see a little **b** next to the instruction we want to stop at.

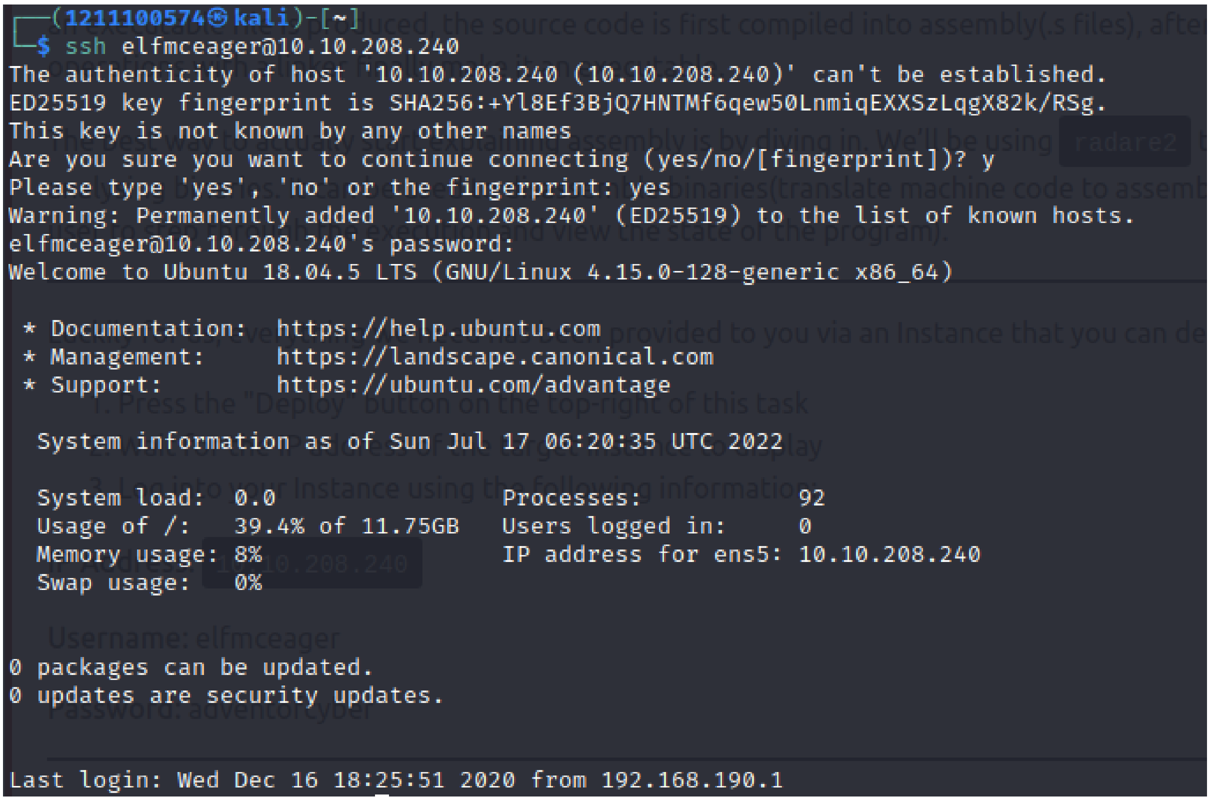
Question 4

In the question

Running dc will execute the program until we hit the breakpoint. (

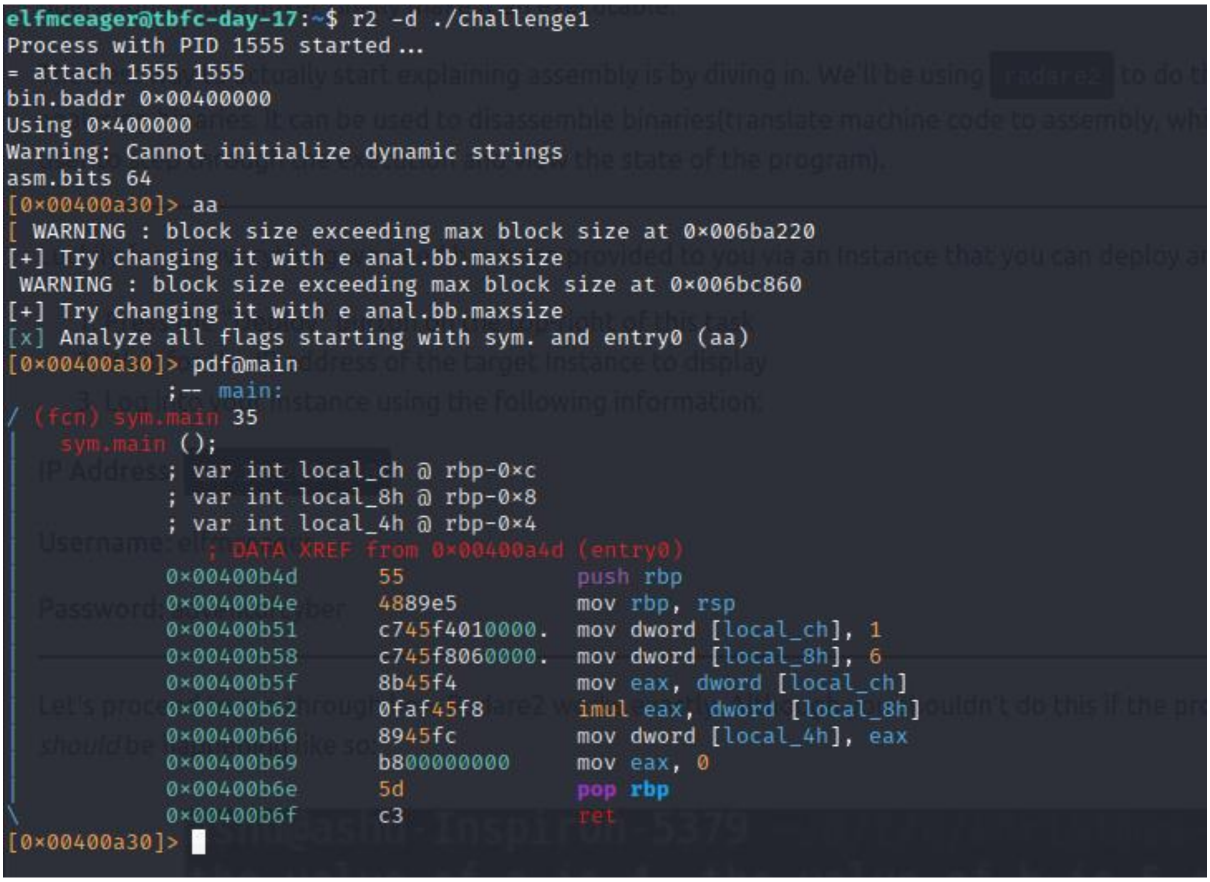
Question 5, 6 and 7

Login to elfmceager's instance



```
┌──(1211100574㉿kali)-[~]
└─$ ssh elfmceager@10.10.208.240
The authenticity of host '10.10.208.240 (10.10.208.240)' can't be established.
ED25519 key fingerprint is SHA256:+Yl8Ef3BjQ7HNTMf6qew50LnmiqEXXSzLqgX82k/RSg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.10.208.240' (ED25519) to the list of known hosts.
elfmceager@10.10.208.240'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 Sun Jul 17 06:20:35 UTC 2022

  System load:  0.0               Processes:           92
  Usage of /:   39.4% of 11.75GB  Users logged in:     0
  Memory usage: 8%                IP address for ens5: 10.10.208.240
  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
```

Open the challenge folder and analyse it. Then print the main to obtain answers



```
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1555 started...
= attach 1555 1555
bin.baddr 0×00400000
Using 0×400000
Warning: Cannot initialize dynamic strings
asm.bits 64
[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
[0×00400a30]>
```

Thought Process/Methodology:

I logged into elfmceager’s terminal using ssh command and the credentials given. I then opened the challenge file and analysed it with aa which I then followed it up by using pdf to print the main where I used to see the values of local_ch, eax and local_4h.

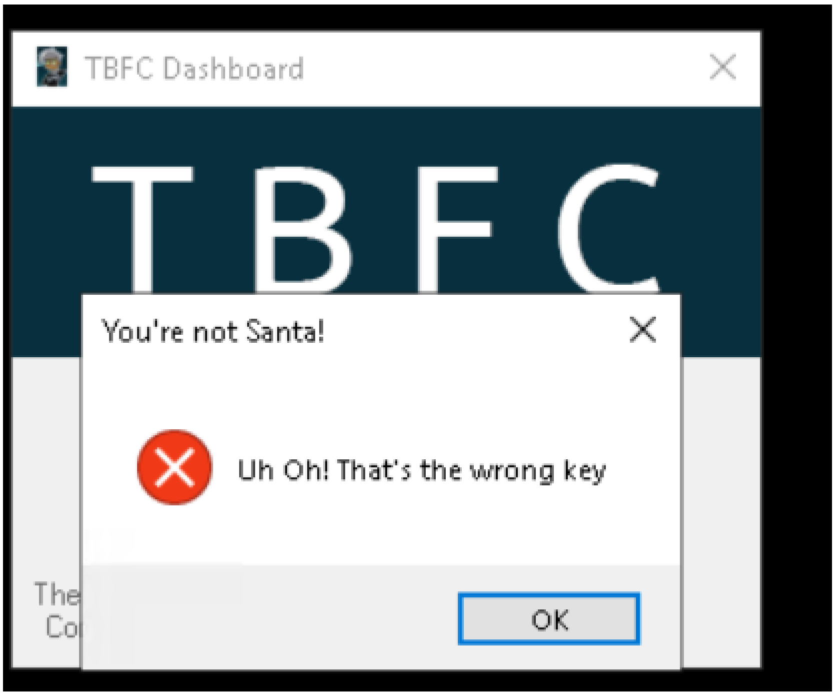
Day 18: [Reverse Engineering] The Bits of Christmas

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

Question 1

Type in anything that isn’t the correct password

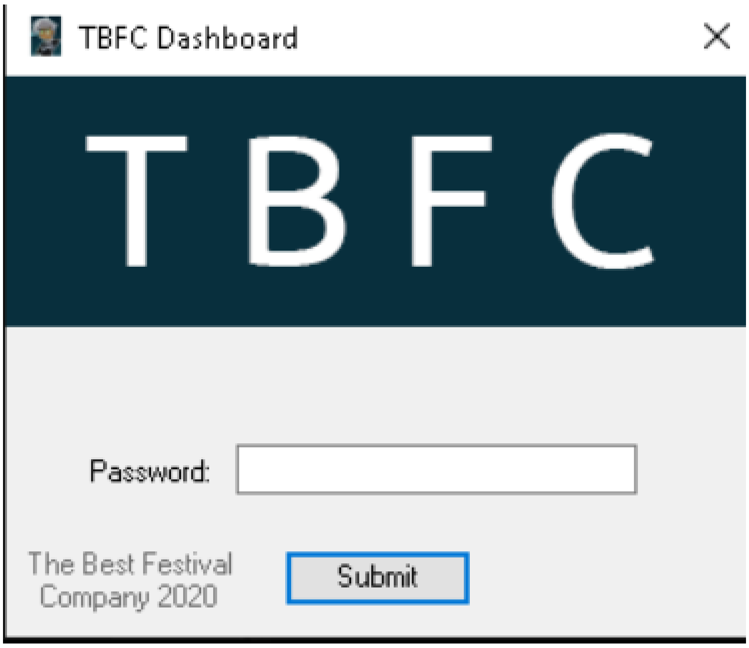


You're not Santa!

Uh Oh! That's the wrong key

Question 2

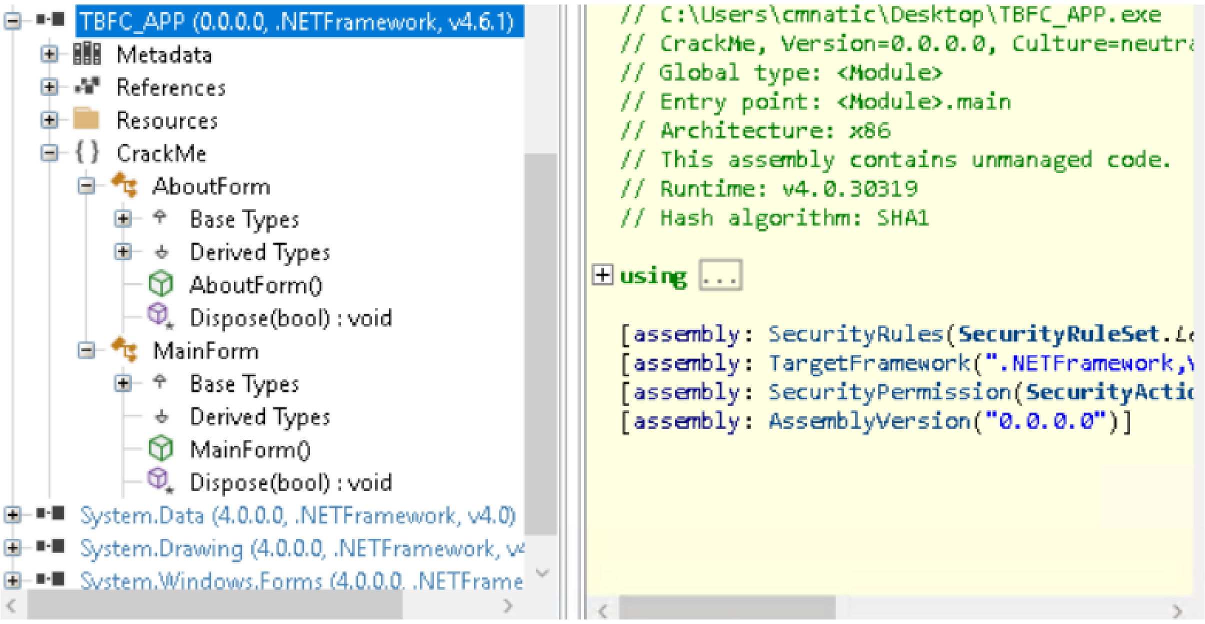
Bottom left of the app



The Best Festival Company 2020

Question 3

Decompile each class to see which has something interesting



```
TBFC_APP (0.0.0.0, .NETFramework, v4.6.1)
    Metadata
    References
    Resources
    { } CrackMe
        AboutForm
            Base Types
            Derived Types
            AboutForm()
            Dispose(bool) : void
        MainForm
            Base Types
            Derived Types
            MainForm()
            Dispose(bool) : void
System.Data (4.0.0.0, .NETFramework, v4.0)
System.Drawing (4.0.0.0, .NETFramework, v4
System.Windows.Forms (4.0.0.0, .NETFrame
```

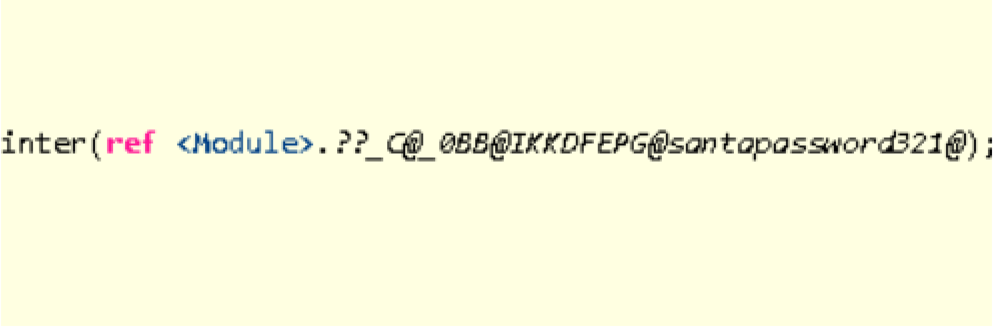
```
// C:\Users\cmnatic\Desktop\TBFC_APP.exe
// CrackMe, Version=0.0.0.0, Culture=neutra
// Global type: <Module>
// Entry point: <Module>.main
// Architecture: x86
// This assembly contains unmanaged code.
// Runtime: v4.0.30319
// Hash algorithm: SHA1

using ...

[assembly: SecurityRules(SecurityRuleSet.Le
[assembly: TargetFramework(".NETFramework,\
[assembly: SecurityPermission(SecurityActic
[assembly: AssemblyVersion("0.0.0.0")]
```

Question 4

The main form contains information about the button click when entering the app

Question 5

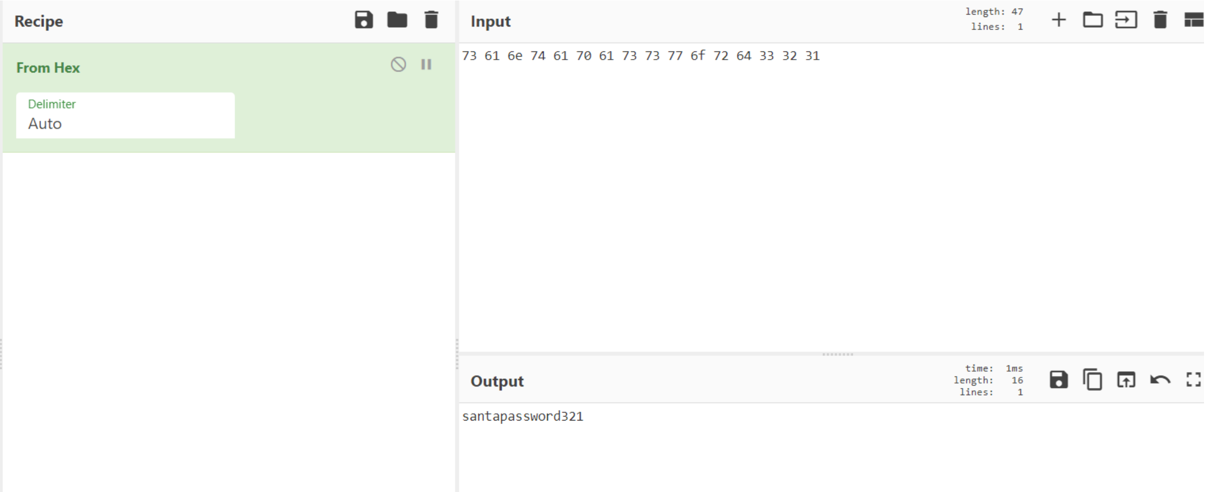
ButtonActivate_Click has a code that may be the password



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

Question 6

Get the code and decode it to see the password



Delimiter: Auto

length: 47 lines: 1

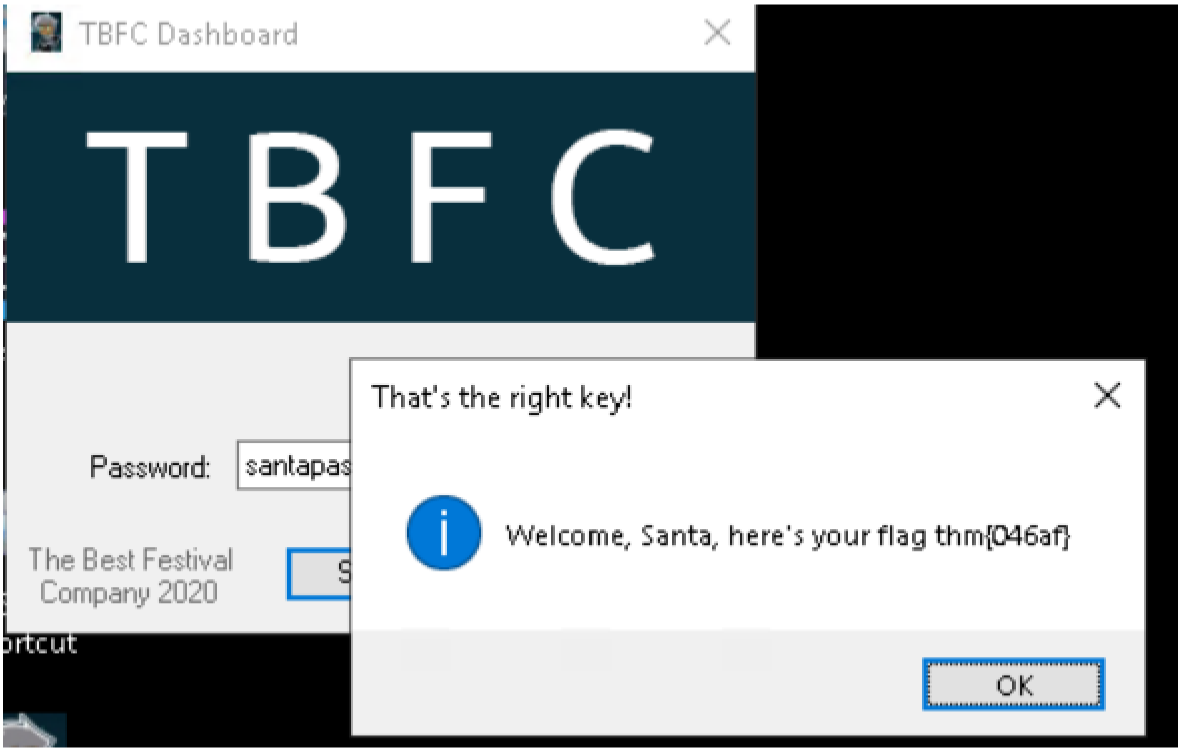
```
73 61 6e 74 61 70 61 73 73 77 6f 72 64 33 32 31
```

time: 1ms length: 16 lines: 1

```
santapassword321
```

Question 7

Enter password to get flag



Password: santapas

That's the right key!

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

Thought Process/Methodology:

I opened an rdp and entered in the given ip address. I then opened the TBFC app to determine what it stood for and to see what prompt was given when an incorrect password was given. I later opened ILspy and decompiled things from the TBFC app to see what contained the information I needed and managed to find the string from a code which would possibly be the password. I then converted the hexadecimal code back into words and entered the password to obtain the flag.

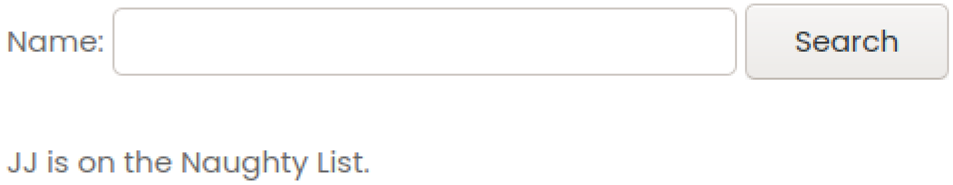
Day 19: [Web Exploitation] The Naughty or Nice list

Tools used: Kali Linux, Firefox

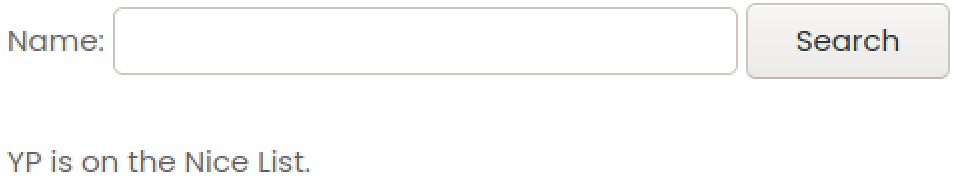
Solution/Walkthrough:

Question 1

Enter the names in the name bar to find whether they are in the naughty or nice list



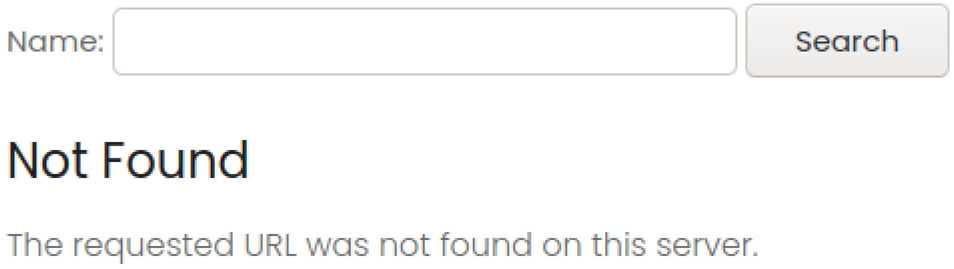
JJ is on the Naughty List.



YP is on the Nice List.

Question 2

Fetch the root of the site

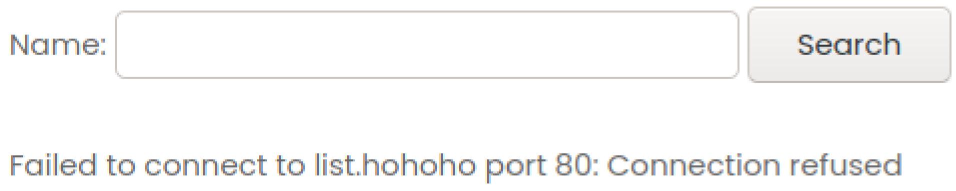


Not Found

The requested URL was not found on this server.

Question 3

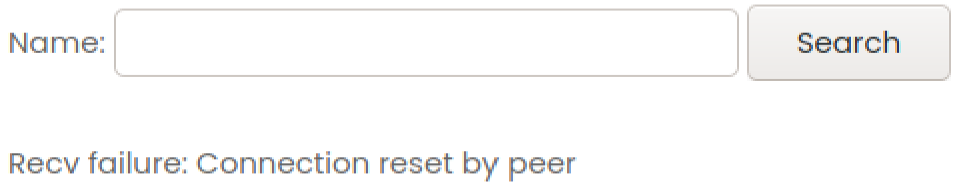
Change port from 8080 to 80



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

Question 4

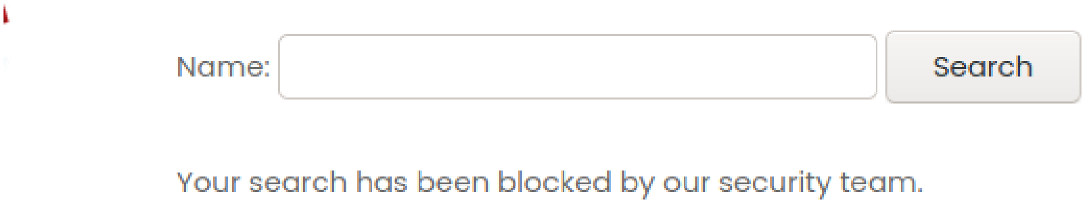
Change port to 22



Recv failure: Connection reset by peer

Question 5

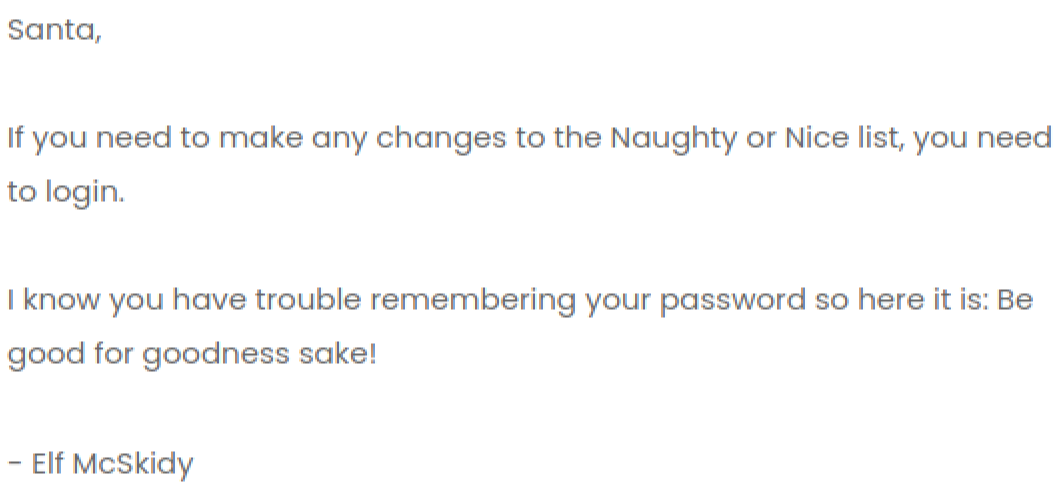
Replace list.hohoho with localhost



Your search has been blocked by our security team.

Question 6

Add localtest.me to the back of list.hohoho to bypass the check



Santa,

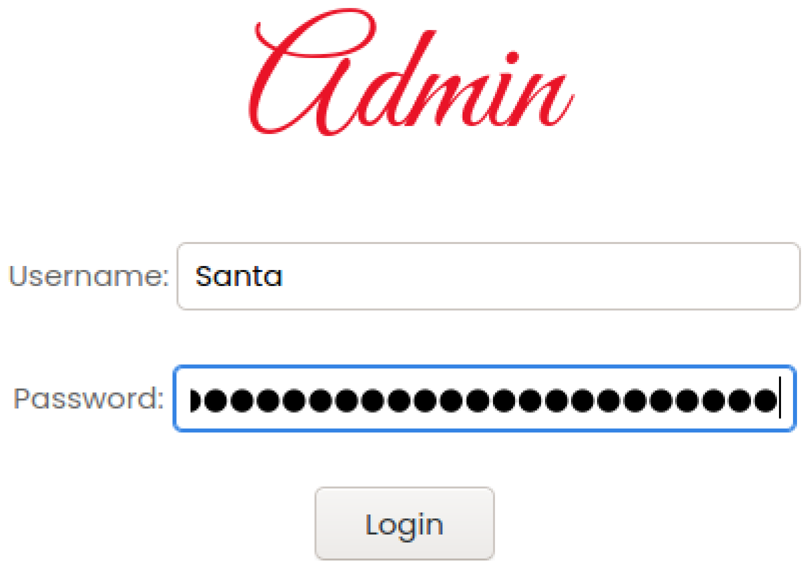
If you need to make any changes to the Naughty or Nice list, you need to login.

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

- Elf McSkidy

<u>Question 7</u>

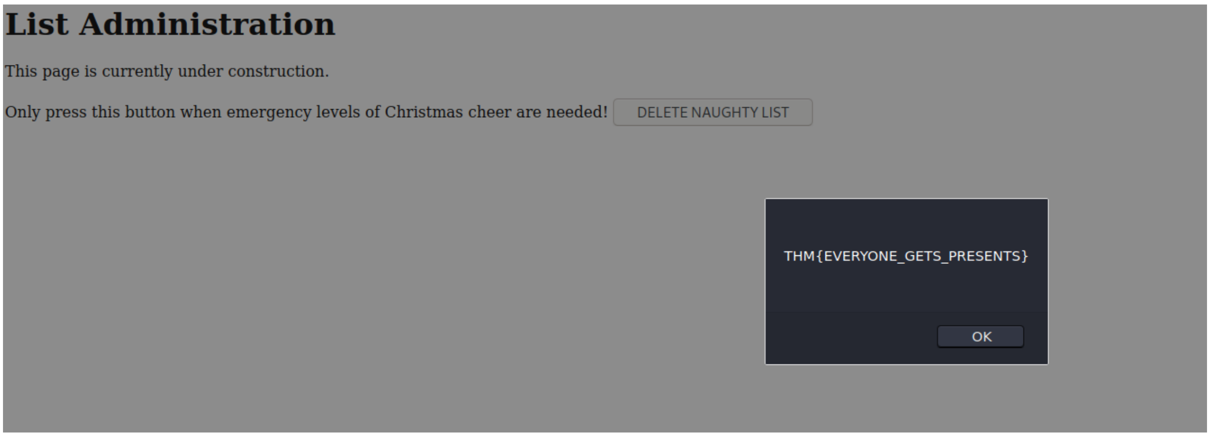
Login using santa and his password to get flag



Admin

Username: Santa

Password: ●●●●●●●●●●●●●●●●●●●●●●●



List Administration

This page is currently under construction.

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

THM{EVERYONE_GETS_PRESENTS}

Thought Process/Methodology:

I entered the website and entered names to determine whether they were in the naughty or nice list. I then attempted to fetch the root of the site. I also changed the port to 80 and 22 which both gave separate error messages. I then changed list.hohoho to localhost to gain access to the local site. I managed to bypass the security check by maintaining list.hohoho while adding localtest.me to gain santa’s password and the flag.

Day 20: [Blue Teaming] PowershELlF to the rescue

Tools used: Kali Linux, Firefox

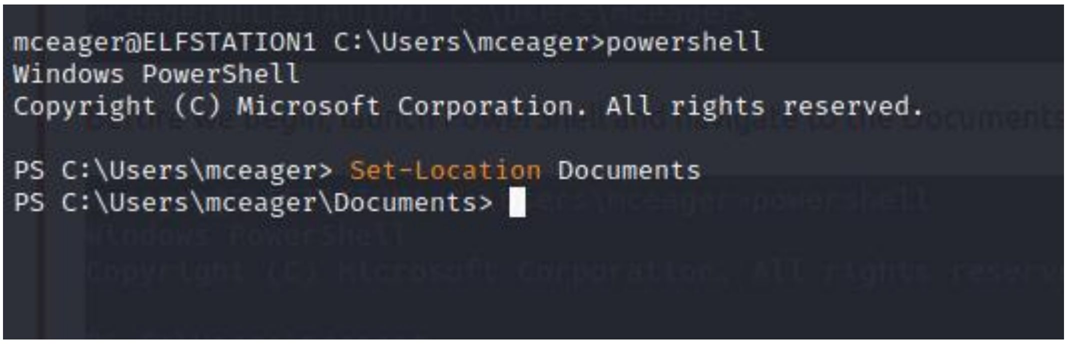
Solution/Walkthrough:

<u>Question 1</u>

Checked the manual and found -l is login name

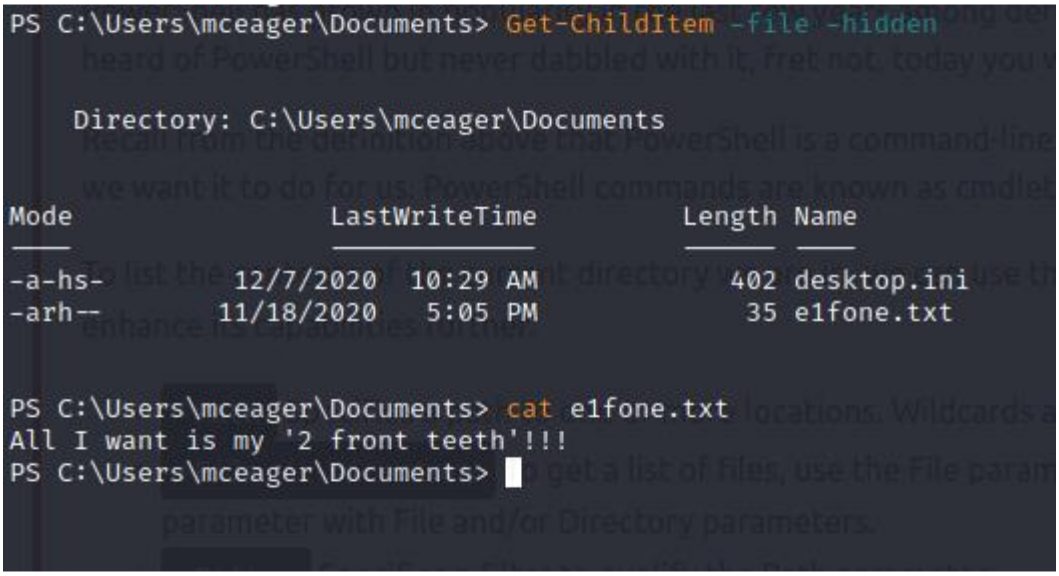
<u>Question 2</u>

Use ssh to login as elfmceager with the password given. Activate powershell and navigate to documents folder



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

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

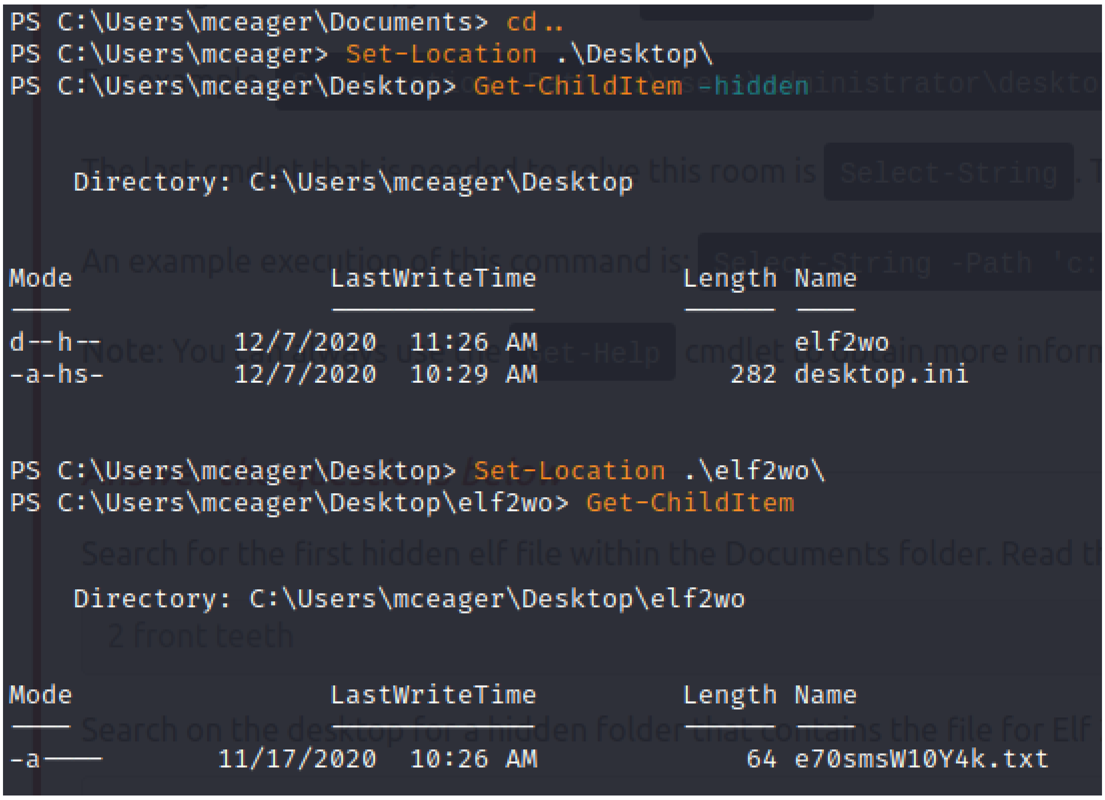
Use get-childitem with -file and -hidden to find hidden files and read the txt file



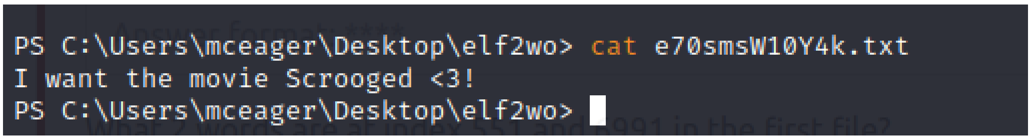
```
PS C:\Users\mceager\Documents> Get-ChildItem -file -hidden


    Directory: C:\Users\mceager\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-hs-         12/7/2020  10:29 AM            402 desktop.ini
-arh--        11/18/2020   5:05 PM             35 e1fone.txt


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

Question 3

Navigate to desktop and find hidden directory and navigate to it. Read text file found in directory



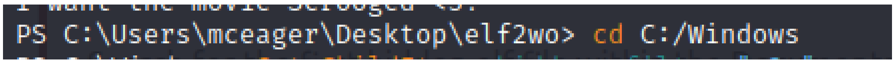
```
PS C:\Users\mceager\Documents> cd ..
PS C:\Users\mceager> Set-Location .\Desktop\
PS C:\Users\mceager\Desktop> Get-ChildItem -hidden


    Directory: C:\Users\mceager\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--h--         12/7/2020  11:26 AM                elf2wo
-a-hs-         12/7/2020  10:29 AM            282 desktop.ini


PS C:\Users\mceager\Desktop> Set-Location .\elf2wo\
PS C:\Users\mceager\Desktop\elf2wo> Get-ChildItem


    Directory: C:\Users\mceager\Desktop\elf2wo


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/17/2020  10:26 AM             64 e70smsW10Y4k.txt
```



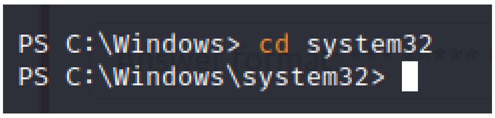
```
PS C:\Users\mceager\Desktop\elf2wo> cat e70smsW10Y4k.txt
I want the movie Scrooged <3!
PS C:\Users\mceager\Desktop\elf2wo>
```

Question 4

Navigate to windows, system 32 directory

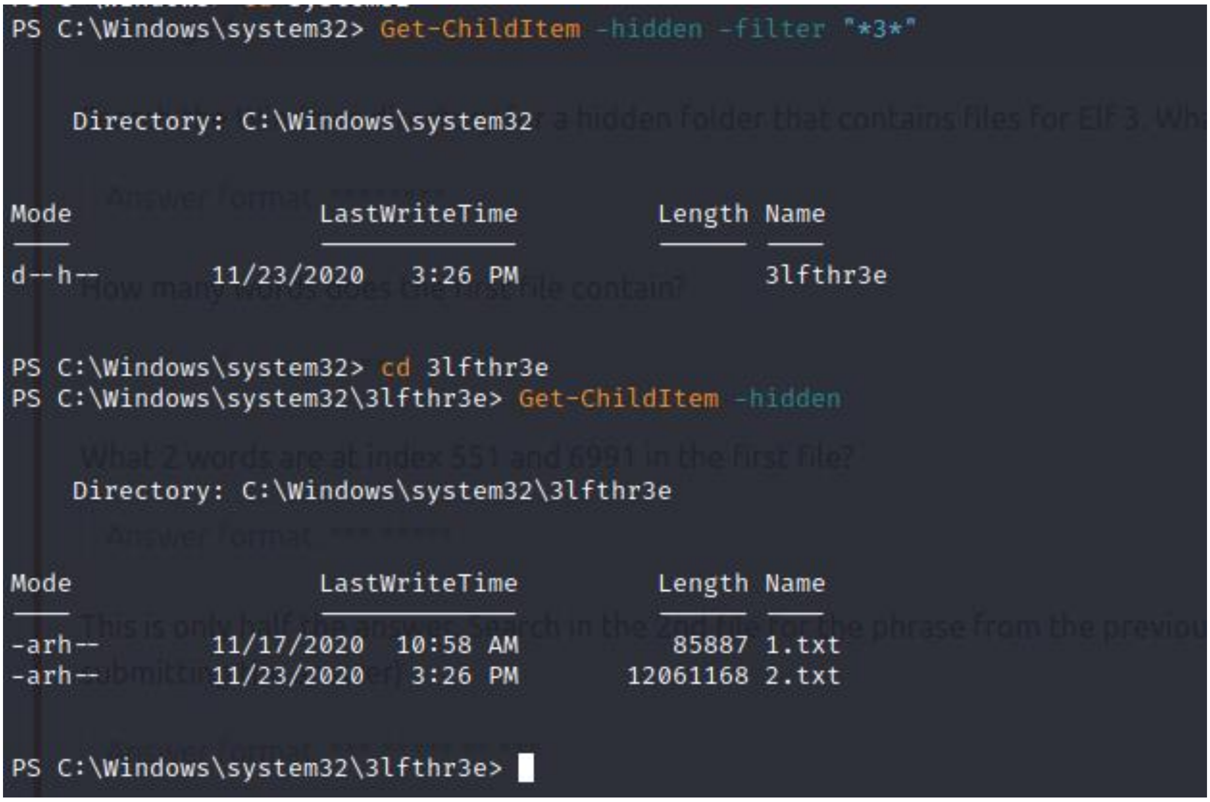


```
PS C:\Users\mceager\Desktop\elf2wo> cd C:/Windows
```



```
PS C:\Windows> cd system32
PS C:\Windows\system32>
```

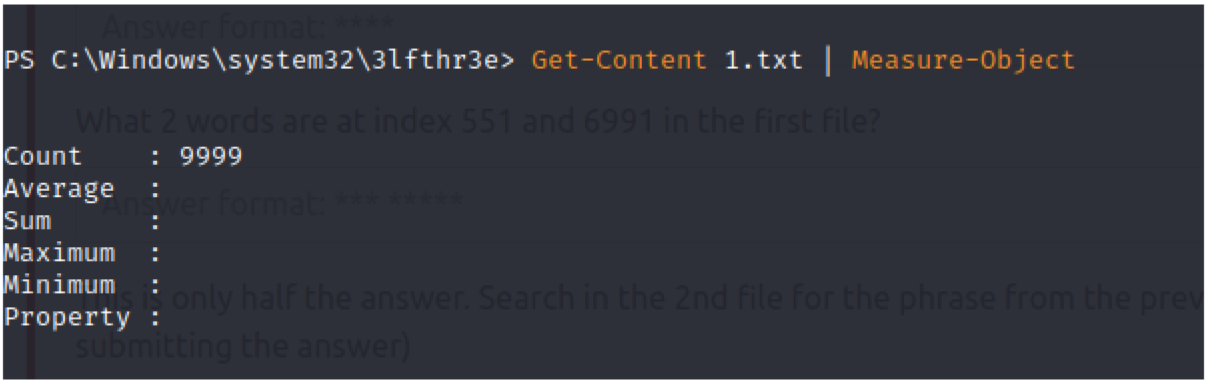
Use filter to find directory with the third files



```
PS C:\Windows\system32> Get-ChildItem -hidden -filter "*3*"


    Directory: C:\Windows\system32


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


PS C:\Windows\system32> cd 3lfthr3e
PS C:\Windows\system32\3lfthr3e> Get-ChildItem -hidden


    Directory: C:\Windows\system32\3lfthr3e


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-arh--        11/17/2020  10:58 AM          85887 1.txt
-arh--        11/23/2020   3:26 PM       12061168 2.txt


PS C:\Windows\system32\3lfthr3e>
```

Question 5

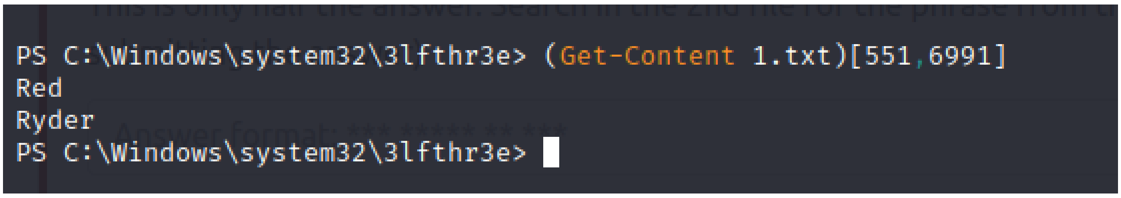
Use measure object to count how many words are in the file



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


Count    : 9999
Average  :
Sum      :
Maximum  :
Minimum  :
Property :
```

Question 6

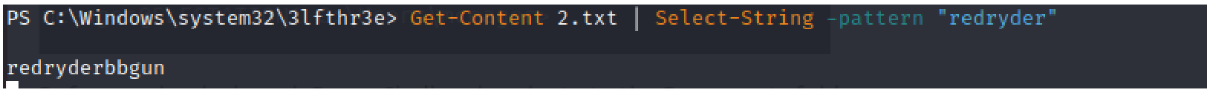
Use get-content with the index for the number needed



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

Question 7

Use select-string to get string with redryder in it



```
PS C:\Windows\system32\3lfthr3e> Get-Content 2.txt | Select-String -pattern "redryder"

redryderbbgun
```

Thought Process/Methodology:

I logged in into elfmceager using ssh and navigated to the documents directory. I then used a command to find hidden files that are in said directory and managed to identify what was the first request. I then repeated it by navigating to the desktop directory and finding a hidden directory that contained the text file for the second request. I then navigated to the windows/system32 directory and used a filter to find the directory that contained the third text files. I used the measure-object command to determine the number of words in the first text file as well as adding the index number needed to find the correct words. The words are then used as a filter to search for a string containing them in the second text file.