PSP0201

Week 5

Writeup

Group Name: suspicious

Member:

ID	Name	Role
1211104293	Noor Hannan Bin Noor Hamsuruddin	Leader
1211102270	Yap Choo Kath Moon	Member
1211103154	Wan Muhammad Atif Bin Taram Satiraksa	Member

Day 16: Help! Where is Santa?

Tool used: Firefox web browser, Kali Linux, Nmap, Python
Solution/walkthrough:

Step 1:

-First, we run a nmap -A scan to see what port to use, in this case the port is 80, using this we were able to access the website.

```
┌──(1211102270㉿kali)-[~]
└─$ nmap -A 10.10.148.37
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-12 01:23 EDT
Nmap scan report for 10.10.148.37
Host is up (0.20s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 31:4e:6f:1b:9b:4d:a6:9f:34:f0:ca:3e:96:31:a6:9e (RSA)
|   256 60:5d:1b:59:24:8b:b8:7a:5f:1c:75:55:5f:bf:e0:83 (ECDSA)
|_  256 05:08:d8:66:d1:04:cf:91:8c:6a:56:55:df:07:a4:d6 (ED25519)
80/tcp open  http    uvicorn
|_http-server-header: uvicorn
|_http-title: Santa's Tracker
| fingerprint-strings:
```

Step 2:

-We wrote a python script with the reference from day 15, to get it's api directory, which is /api/.

~/link.py - Mousepad

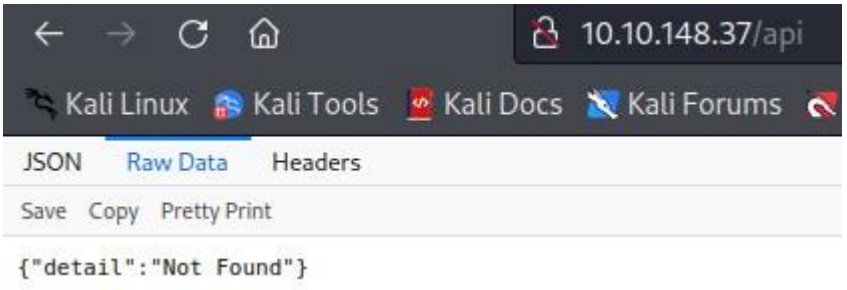
File Edit Search View Document Help

```python
# Import the libraries we downloaded earlier
# if you try importing without installing them, this step will fail
from bs4 import BeautifulSoup
import requests

# replace testurl.com with the url you want to use.
# requests.get downloads the webpage and stores it as a variable
html = requests.get('http://10.10.148.37:80/')

# this parses the webpage into something that beautifulsoup can read over
soup = BeautifulSoup(html.text, "lxml")
# lxml is just the parser for reading the html

# this is the line that grabs all the links # stores all the links in the links variable
links = soup.find_all('a')

for link in links:
    # prints each link
    if 'href' in link.attrs:
        print(link["href"])

```

```
┌──(1211102270㉿kali)-[~]
└─$ python3 link.py
../

https://github.com/BulmaTemplates/bulma-templates/blob/master/templates/hero.html
https://tryhackme.com
https://tryhackme.com
https://tryhackme.com
https://tryhackme.com
https://tryhackme.com
https://tryhackme.com
https://tryhackme.com
https://tryhackme.com
https://tryhackme.com
https://tryhackme.com
https://tryhackme.com
#
#
#
#
#
#
#
#
#
http://machine_ip/api/api_key
#
```

Step 3:

-With the directory of it's api we insert it into the search bar as such 10.10.148.37/api/. Then we goes to the raw data section to the {"detail":"Not Found"}



Step 4:

-We wrote another python script to get the api key, after running the script, we got the api key which 57 and also santa's location.

```
import requests

for api_key in range(1,100,2):
        print(f'api_key {api_key}')
        html = requests.get(f'http://10.10.148.37:80/api/{api_key}')
        print(html.text)

```

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

Thought process/methodology:

Using nmap we were able to get the port, which enable us to enter into the website, with this we were able to view api that we got from our python script that we wrote and run. We then run another script we wrote to get it's api key and also santa's location.

Day 17: ReverseELFneering:

Tool used: Kali Linux, radare2

Solution/walkthrough:

Step 1:

-First we save the ip address to a text file, then we login into the server as elfmceager.

```
┌──(1211102270㉿kali)-[~]
└─$ echo '10.10.4.32' > tg.txt
```

```
┌──(1211102270㉿kali)-[~]
└─$ ssh elfmceager@10.10.4.32
The authenticity of host '10.10.4.32 (10.10.4.32)' 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])? yes
Warning: Permanently added '10.10.4.32' (ED25519) to the list of known hosts.
elfmceager@10.10.4.32's password:
Permission denied, please try again.
elfmceager@10.10.4.32'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 Fri Jul 15 03:57:42 UTC 2022

  System load:  0.1                Processes:           100
  Usage of /:   39.4% of 11.75GB   Users logged in:     0
  Memory usage: 8%                 IP address for ens5: 10.10.4.32
  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
elfmceager@tbfc-day-17:~$
::1             ff02::2         ip6-localnet    tbfc-day-17
fe00::0         ip6-allnodes    ip6-loopback
ff00::0         ip6-allrouters  ip6-mcastprefix
ff02::1         ip6-localhost   localhost
elfmceager@tbfc-day-17:~$
```

Step 2:

-After that we open the challenge1 file in debug mode.

```
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1620 started...
= attach 1620 1620
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
asm.bits 64
[0x00400a30]>
```

Step 3:

-Then we ran the aa command to anaylse the file, after that we use the pdf command see it's main.

```
[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]>
```

```
[0x00400a30]> pdf @main
            ;-- main:
/ (fcn) sym.main 35
|   sym.main ();
|           ; var int local_ch @ rbp-0x c
|           ; var int local_8h @ rbp-0x 8
|           ; var int local_4h @ rbp-0x 4
|              ; 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]>
```

Step 4:

-After that we analyse the main find the value of the local_ch when its corresponding movl instruction is called, which is 1, then we find the value of eax when the imull instruction is called and local_4h before eax is set to 0, which both of them is 6.

```
[0x00400a30]> pdf @main
            ;-- main:
/ (fcn) sym.main 35
|   sym.main ();
|           ; var int local_ch @ rbp-0x c
|           ; var int local_8h @ rbp-0x 8
|           ; var int local_4h @ rbp-0x 4
|              ; 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]>
```

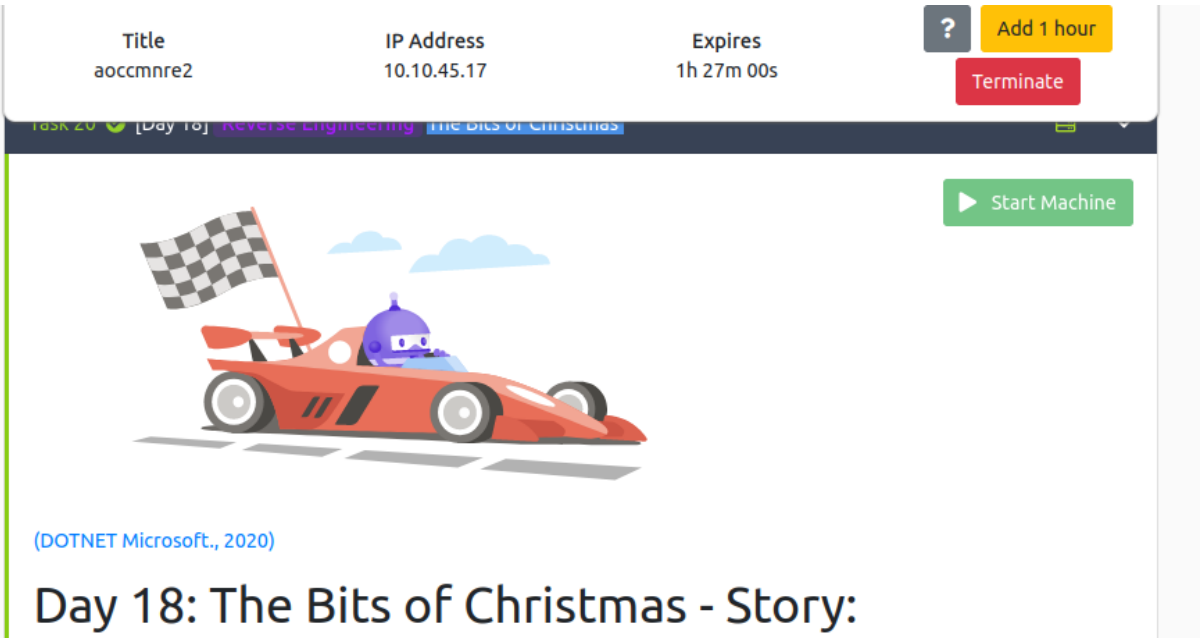
Thought process/methodology:
After logging into the server we use radare2 to find the information the server, we enter into the challenge1 in debug mode, and analyse the file with aa command then run pdf @main to see the information then get the value for local_ch when its corresponding movl instruction is called, which is 1. Then, eax when the imull instruction is called and local_4h before eax is set to 0, which both of them is 6.

Day 18: The Bits of Christmas

Tool used: Kali Linux, Remmina, ILSpy

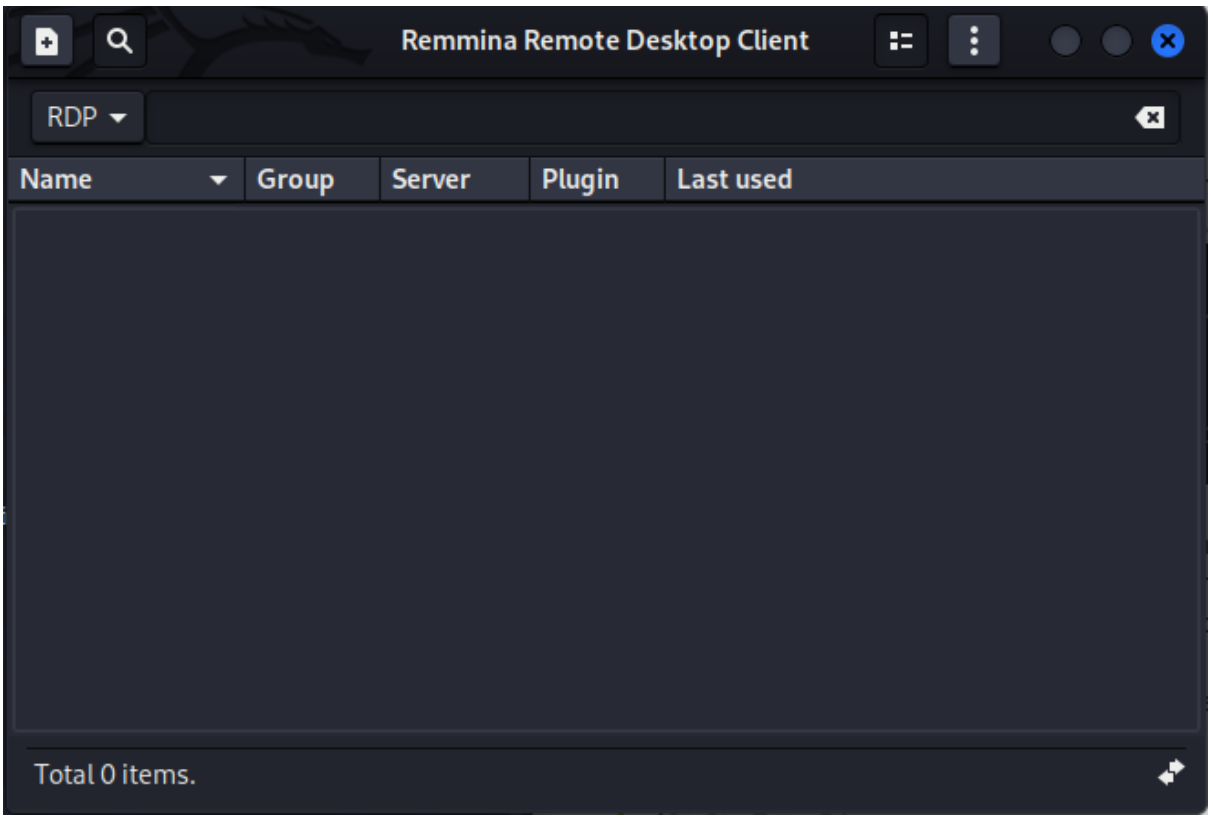
Solution/walkthrough:

Step 1: Start the machine on the day 18 task

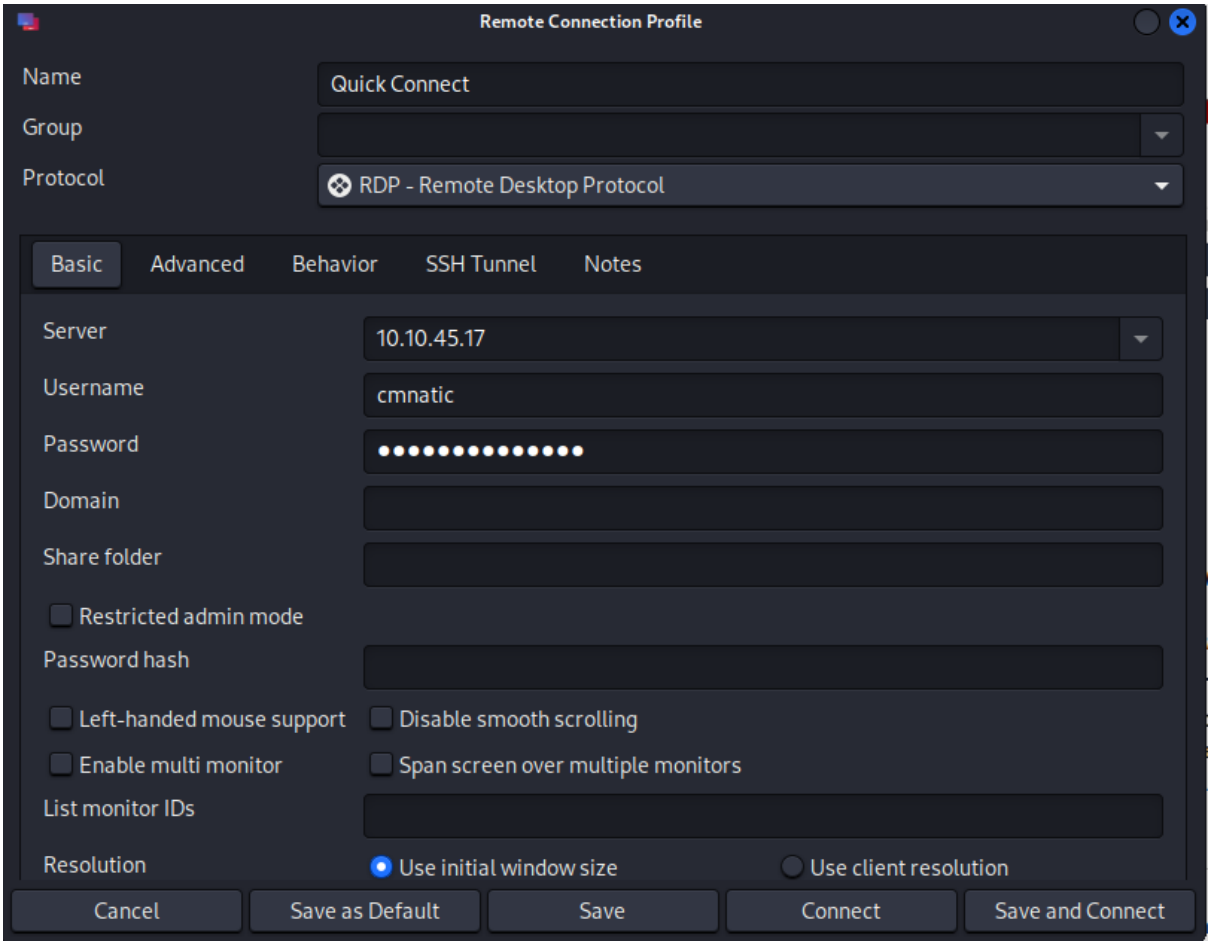


Step 2: Install and open Remmina by using the command sudo apt-get -y install remmina.

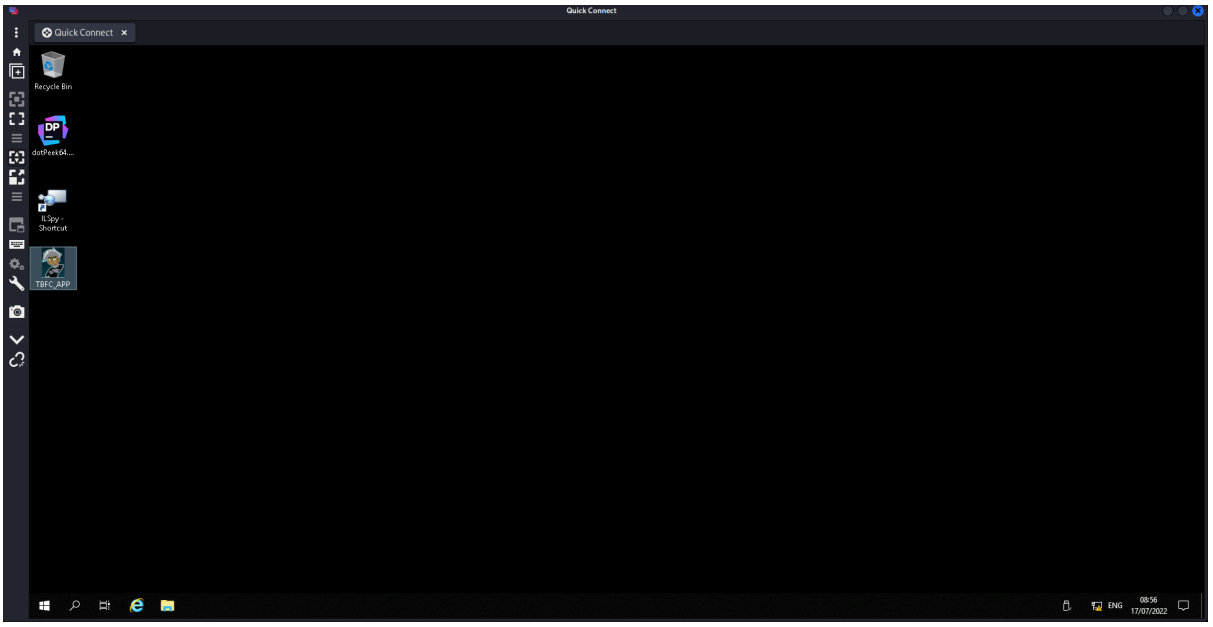
```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get -y install remmina
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  avahi-daemon libavahi-client3 libavahi-common-data libavahi-common3 libavahi-core7 libavahi-glib1
  libavahi-ui-gtk3-0 libvncclient1 remmina-common remmina-plugin-rdp remmina-plugin-secret remmina-plugin-vnc
Suggested packages:
  avahi-autoipd remmina-plugin-exec remmina-plugin-kwallet remmina-plugin-python remmina-plugin-spice
  remmina-plugin-www remmina-plugin-x2go
Recommended packages:
  libnss-mdns
The following NEW packages will be installed:
  libavahi-ui-gtk3-0 libvncclient1 remmina remmina-common remmina-plugin-rdp remmina-plugin-secret
  remmina-plugin-vnc
The following packages will be upgraded:
  avahi-daemon libavahi-client3 libavahi-common-data libavahi-common3 libavahi-core7 libavahi-glib1
6 upgraded, 7 newly installed, 0 to remove and 810 not upgraded.
```



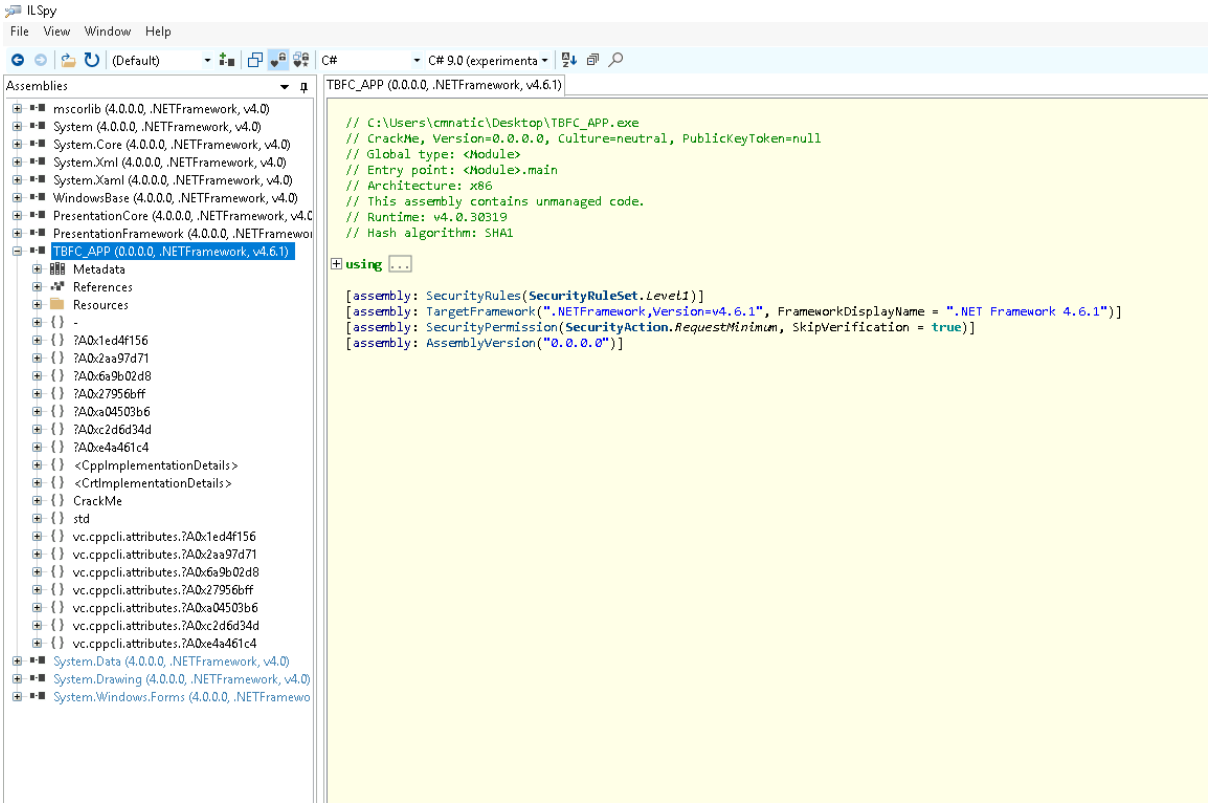
Step 3: Establish and RDP connection using Remmina and the login details provided by THM



Step 4: Once connected, a Windows Desktop should appear.



Step 5: Open ILSpy and open the TBFC_APP on said software. We can see that it runs on a .NET framework from here.

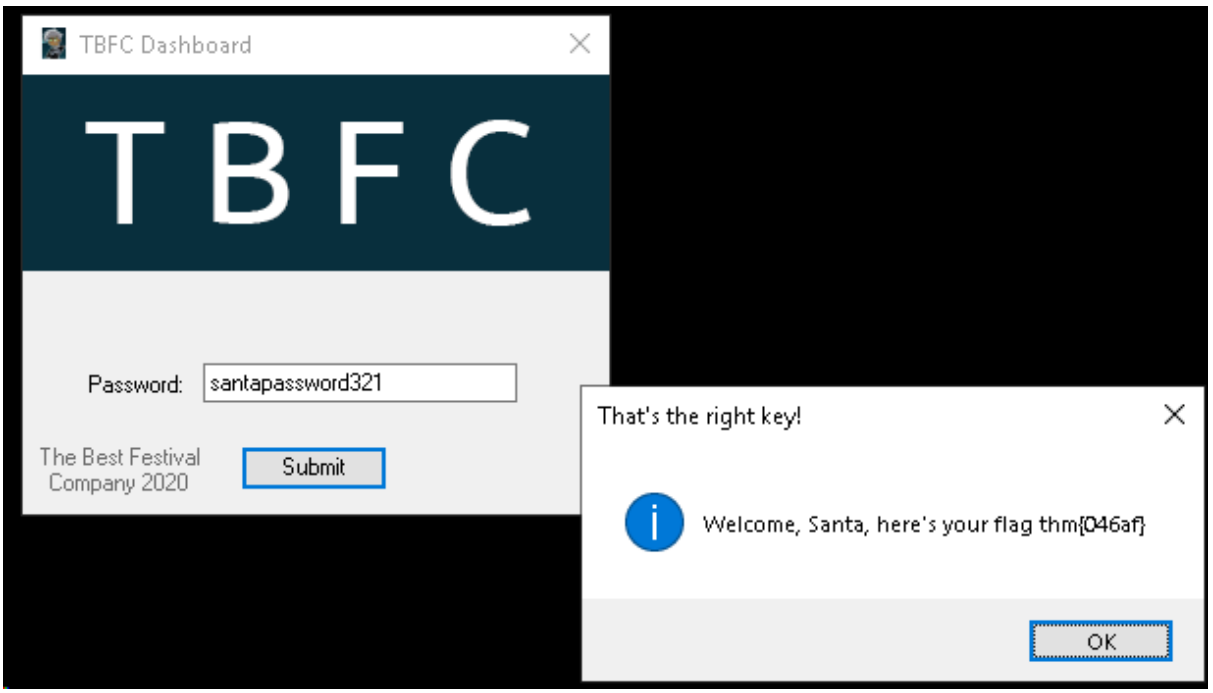


Step 6: Parse through the “CrackMe” to find source codes that are used in the login form. It should be located specifically at “CrackMe.MainForm.buttonActivate_Click” since the source code for that is used to activate the login button.

```
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
System.Data (4.0.0.0, .NETFramework, v4.0)
System.Drawing (4.0.0.0, .NETFramework, v4.0)
System.Windows.Forms (4.0.0.0, .NETFramewo
```

Step 7: From here we can find both a probable password as well as the flag that would show if we get the password correct. We can test this out by inputting the probable password into the TBFC App to find that it is indeed the password.

```
buttonActivate_Click(object, EventArgs) : void
// CrackMe.MainForm
using System;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
using System.Windows.Forms;

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);
}
```



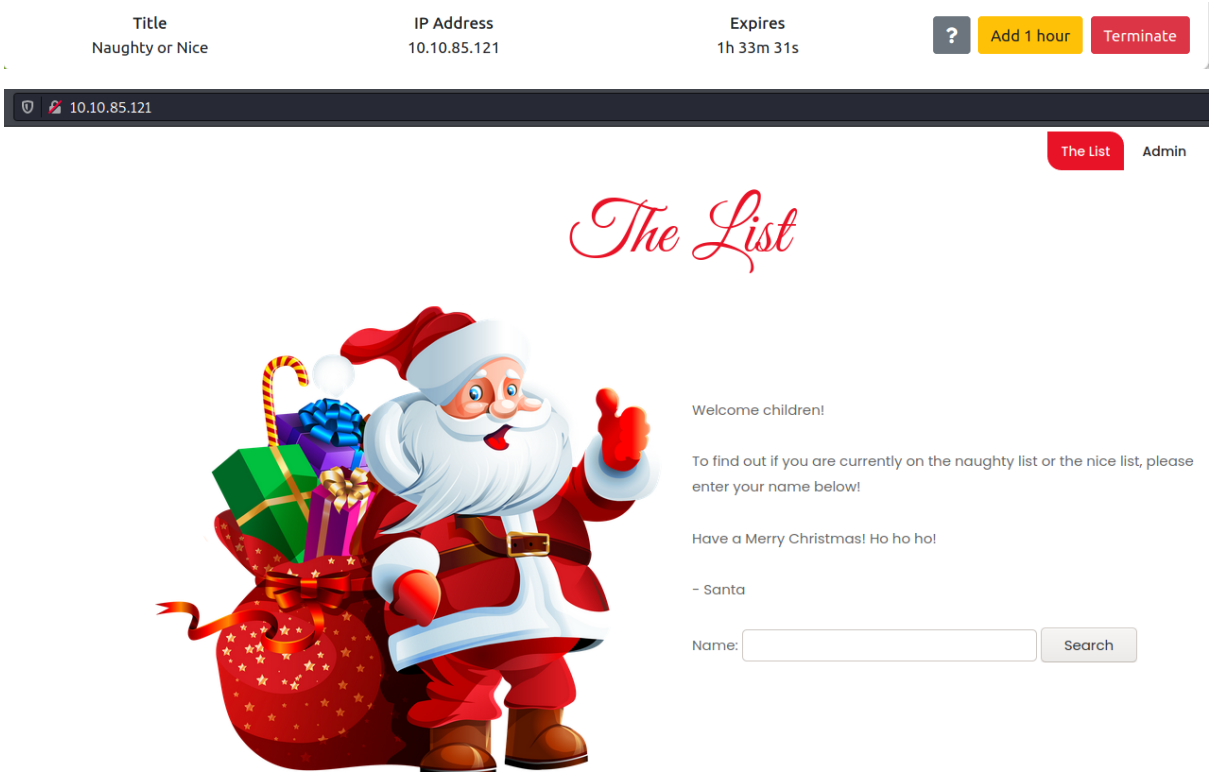
Thinking Process/Methodology :
By using Remmina, we can connect to the machine via Remote Desktop Protocol (RDP). From there we are tasked to find Santa's password that he had forgotten. Using ILSpy, we can disassemble the TBFC App as it uses a commonly used software development framework called .NET Framework. From there, all the application's source code can be compiled and reviewed. We then check each code until we find a relevant one in this case located at “CrackMe.MainForm.buttonActivate_Click”. We found a probable password as well as the flag that will be shown once we successfully logged in to the app. Testing it out in the app itself, we find that it is the correct password as it shows the flag popup.

Day 19:The Naughty or Nice List

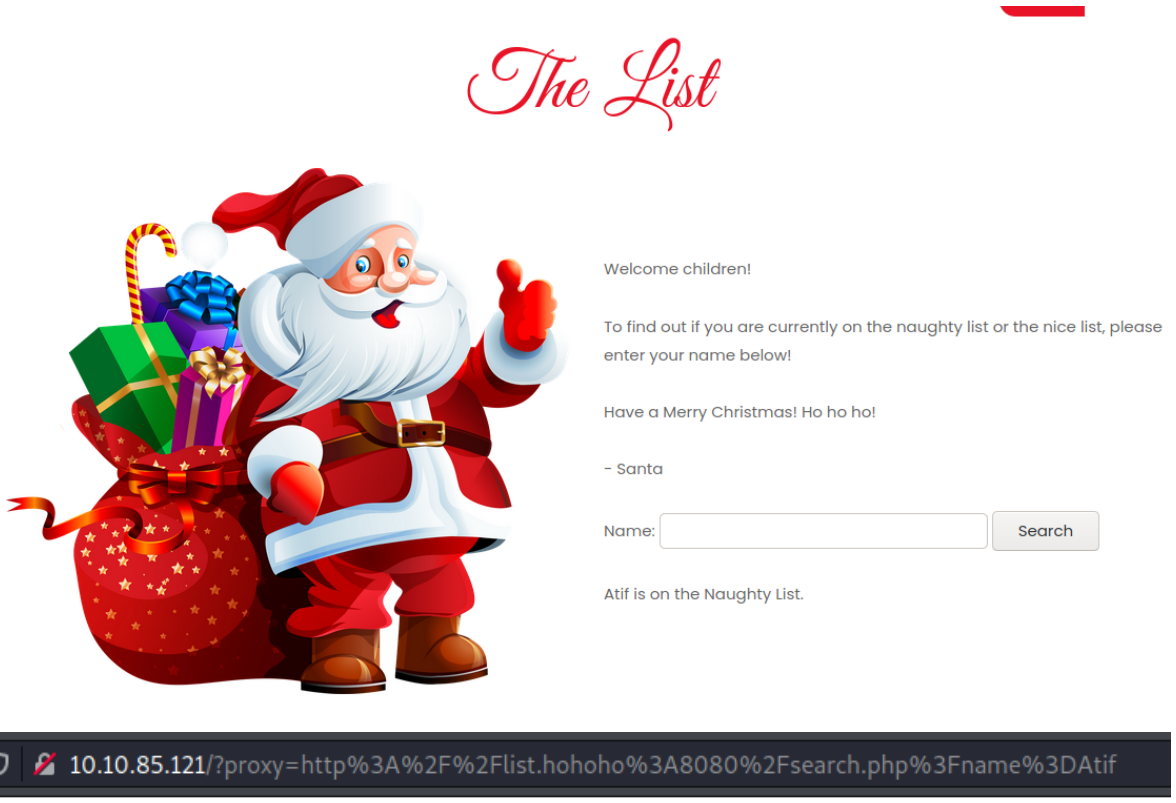
Tool used: Kali Linux

Solution/walkthrough:

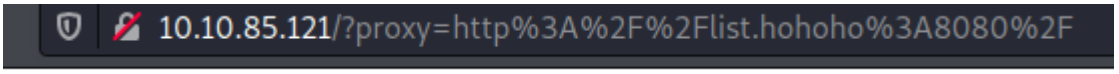
Step 1: Connect through the web app using the link http://(MACHINE IP)

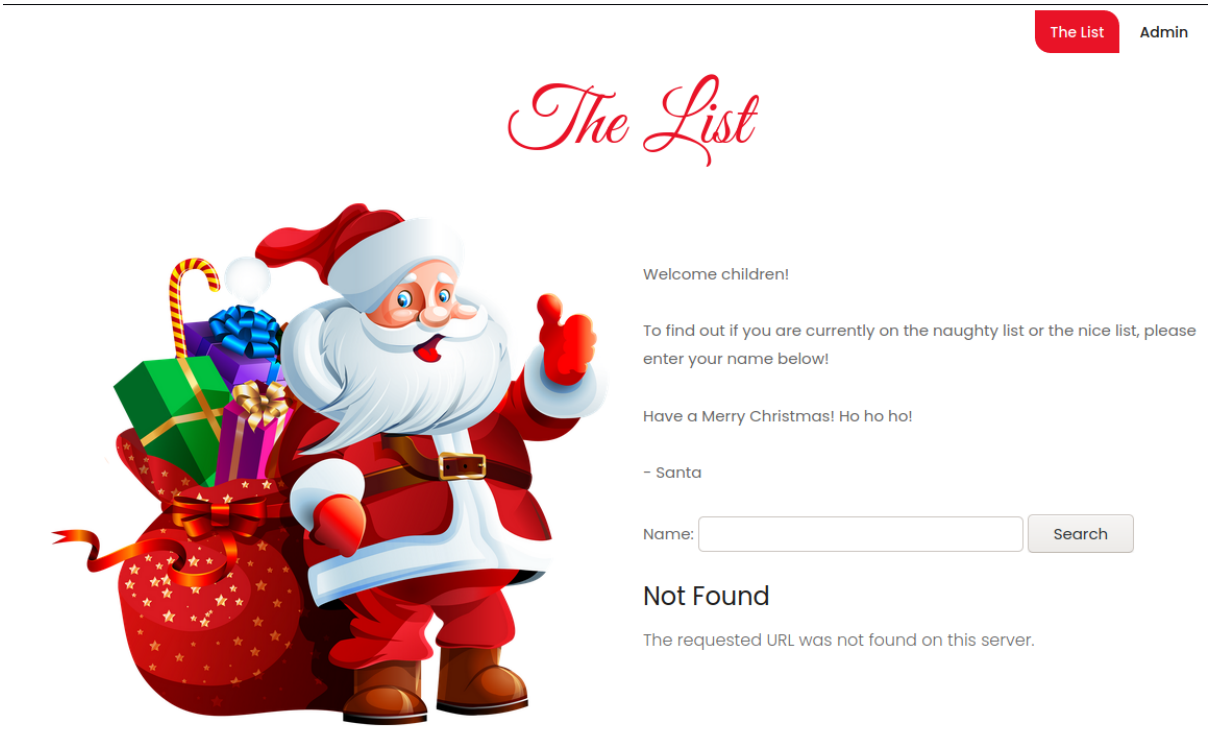


Step 2: Attempt to check the naughty or nice list by typing in a random name and observe the output.



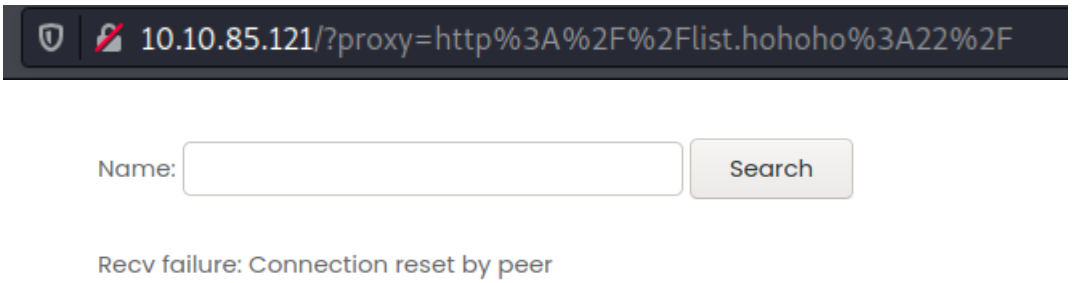
Step 3: Attempt to connect to the root of the same site by typing in the address, consisting of the ip as well as the proxy link up to the text before “search.php…”



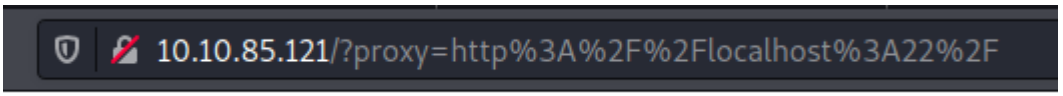


(Instead of the usual response, we receive a “not found”.)

Step 4: Change the 8080 in the link(a port number) to different ports to see if there are any other open ports. Attempt this by using common ports such as 80 or 22.



Step 5: Attempt to connect to a localhost to try and gain access to the Naughty/Nice list by replacing the “flist.hohoho” target with localhost in the link in the search bar.



Name: Search

Your search has been blocked by our security team.

Step 6: To attempt to bypass this, replace localhost with “flist.hohoho” and add “localtest.me” and remove the rest of the text in front.

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

Name: Search

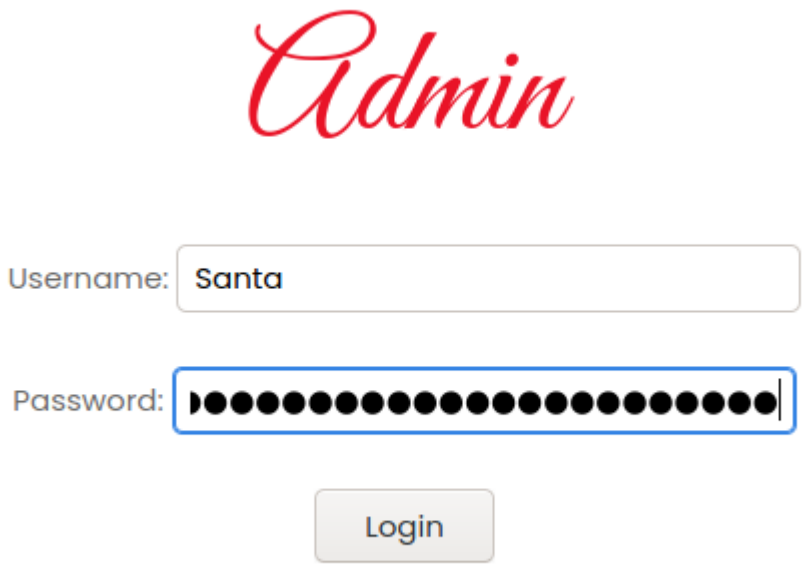
Santa,

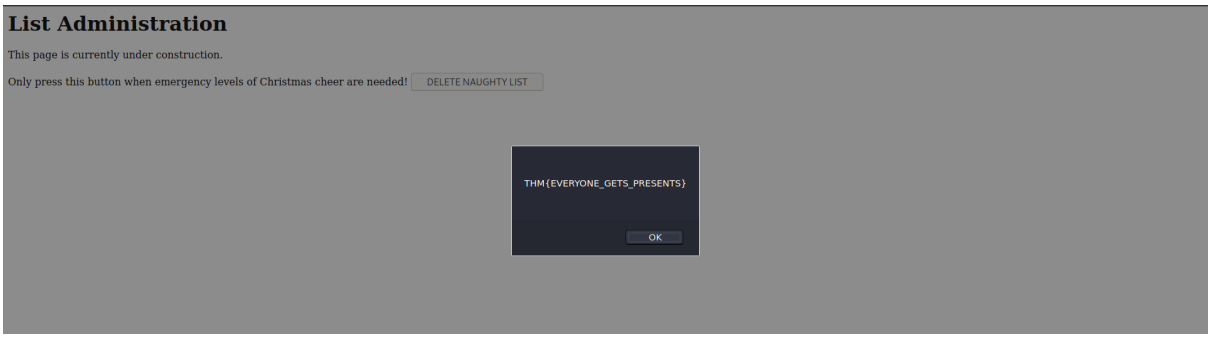
If you need to make any changes to the Naughty or Nice list, you need to login.

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

- Elf McSkidy

Step 7: Use the given password and guess santa’s username(in this case his username is simply Santa) to gain access and delete the naughty list and get the flag.





Thinking process/Methodology:
Our goal this time is to ensure every child gets a present from Santa, no matter how naughty or nice they have been. In order to achieve this, we will work to delete the naughty list and ensure that everyone has been “nice” this year. We start by first typing in a random name, and thanks to a vulnerability, we manage to see the proxy server inside the search bar. We then attempt to connect to the localhost server by switching out the back-end machine target “flist.hohoho” with “localhost”. After being denied by the enemy security’s check, we bypass it by adding “localtest.me” after the flist to ensure flist.hohoho is still

in the url while still accessing localhost. By doing this, we manage to obtain the password to Santa's account, allowing us to delete the naughty list and obtain a flag.

Day 20:PowersElf to the rescue

Tool used: Kali Linux

Solution/Walkthrough:

Step 1: Start and connect to the remote machine using SSH.

```
┌──(kali㉿kali)-[~]
└─$ ssh -l mceager 10.10.161.224
The authenticity of host '10.10.161.224 (10.10.161.224)' can't be established.
ED25519 key fingerprint is SHA256:X2ViBklLQoHmAsXFoem36jkL9faKH+Fr2lt2dd/kIWY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.161.224' (ED25519) to the list of known hosts.
mceager@10.10.161.224's password:
```

Step 2: Use the password provided(rOckStar!) to log in.

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

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

Step 3: Change the directory to become the Document directory.

```
mceager@ELFSTATION1 C:\Users\mceager>dir
 Volume in drive C has no label.
 Volume Serial Number is E82E-8322

 Directory of C:\Users\mceager

12/07/2020  11:29 AM    <DIR>          .
12/07/2020  11:29 AM    <DIR>          ..
12/07/2020  11:29 AM    <DIR>          3D Objects
12/07/2020  11:29 AM    <DIR>          Contacts
12/07/2020  12:26 PM    <DIR>          Desktop
12/07/2020  12:26 PM    <DIR>          Documents
12/07/2020  11:29 AM    <DIR>          Downloads
12/07/2020  11:29 AM    <DIR>          Favorites
12/07/2020  11:29 AM    <DIR>          Links
12/07/2020  11:29 AM    <DIR>          Music
12/07/2020  11:29 AM    <DIR>          Pictures
12/07/2020  11:29 AM    <DIR>          Saved Games
12/07/2020  11:29 AM    <DIR>          Searches
12/07/2020  11:29 AM    <DIR>          Videos
               0 File(s)              0 bytes
              14 Dir(s)   5,122,736,128 bytes free

mceager@ELFSTATION1 C:\Users\mceager>cd Documents

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

Step 4: Activate Powershell by typing in “powershell” then use a command in powershell to find hidden files in Documents. Use the “TYPE” command to read the text file found in Documents.

```
PS C:\Users\mceager\Documents> Get-ChildItem -File -Hidden -ErrorAction SilentlyContinue


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

Step 5: Return to the main directory of mceager and then change directory again to desktop. Use the command from earlier(with slight modifications from -File to -Directory) to find more hidden files left behind by other elves.

```
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-r---        12/7/2020  10:29 AM                3D Objects
d-r---        12/7/2020  10:29 AM                Contacts
d-r---        12/7/2020  11:26 AM                Desktop
d-r---        12/7/2020  11:26 AM                Documents
d-r---        12/7/2020  10:29 AM                Downloads
d-r---        12/7/2020  10:29 AM                Favorites
d-r---        12/7/2020  10:29 AM                Links
d-r---        12/7/2020  10:29 AM                Music
d-r---        12/7/2020  10:29 AM                Pictures
d-r---        12/7/2020  10:29 AM                Saved Games
d-r---        12/7/2020  10:29 AM                Searches
d-r---        12/7/2020  10:29 AM                Videos


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

```
PS C:\Users\mceager\Desktop> Get-ChildItem -Directory -Hidden -ErrorAction SilentlyContinue


    Directory: C:\Users\mceager\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d--h--        12/7/2020  11:26 AM                elf2wo


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


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

Step 6: Change directories until you find the windows directory, then navigate further into the System32 directory, then find the hidden file located inside the directory left behind by the third elf.

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

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

```
    Directory: C:\Windows

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        7/16/2022  10:50 PM                System32
d-----        9/15/2018  12:19 AM                twain_32

PS C:\Windows> cd System32
PS C:\Windows\System32> dir

    Directory: C:\Windows\System32
```

```
PS C:\Windows\System32> Get-ChildItem -Directory -Hidden -ErrorAction SilentlyContinue -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> dir
PS C:\Windows\System32\3lfthr3e> Get-ChildItem -File -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
```

Step 7: Use commands to find out how many words are there inside the file “1.txt”

```
poison
PS C:\Windows\System32\3lfthr3e> Get-Content -Path 1.txt | Measure-Object -Word

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

Step 8: Get the words located specifically at index 551 and 6991 using more commands.

```
PS C:\Windows\System32\3lfthr3e> (Get-Content -Path 1.txt)[551]
Red
```

```
PS C:\Windows\System32\3lfthr3e> (Get-Content -PAth 1.txt)[6991]
Ryder
PS C:\Windows\System32\3lfthr3e>
```

Step 9: Use the phrase obtained by combining the 2 words from step 8 as well as some other commands to find what Elf 3 wants in 2.txt.

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

2.txt:558704:redryderbbgun
```

Thought process/methodology:
To find the traces left behind by the misbehaving elves we first begin by looking into some directories and what hidden files they may contain. First, we connect to the remote Machine using SSH and then activate powershell. We then first use commands to find any hidden files inside of the Documents directory. Afterwards, we navigate to the Desktop directory and then search for hidden files there. And finally, we navigate all the way to the windows directory(or more specifically, the System32 directory) and search inside to find the hidden file left by Elf 3. Inside, we use commands to find specific keywords and phrases to find out what the elf wants and obtain the information we need.