

# PSP0201

## Week 5

# Writeup

Group Name: 404 Not Found

Members

ID	Name	Role
1211102687	Emily Phang Ru Ying	Leader
1211102975	Loi Xinyi	Member
1211102751	Teo Yu Jie	Member
1211102753	Lim Cai Qing	Member

## Day 16 [Scripting] – Help! Where is Santa?

Tools used: Kali Linux, Firefox, nmap, python 3

Solution/walkthrough:

### Question 1

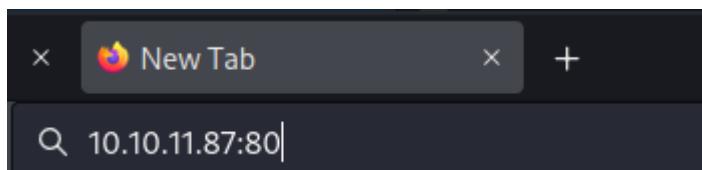
We found two open ports after using nmap to scan the IP address. Port 80 is the answer as it is used to host an HTTP service.

```
(1211102687㉿kali)-[~]
$ nmap -v 10.10.11.87
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-14 08:48 EDT
Initiating Ping Scan at 08:48
Scanning 10.10.11.87 [2 ports]
Completed Ping Scan at 08:49, 0.19s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 08:49
Completed Parallel DNS resolution of 1 host. at 08:49, 0.08s elapsed
Initiating Connect Scan at 08:49
Scanning 10.10.11.87 [1000 ports]
Discovered open port 80/tcp on 10.10.11.87
Discovered open port 22/tcp on 10.10.11.87
Increasing send delay for 10.10.11.87 from 0 to 5 due to 73 out of 241 dropped probes since last.
Completed Connect Scan at 08:49, 14.67s elapsed (1000 total ports)
Nmap scan report for 10.10.11.87
Host is up (0.21s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE      SERVICE
22/tcp    open       ssh
80/tcp    open       http
5357/tcp  filtered  wsdapi

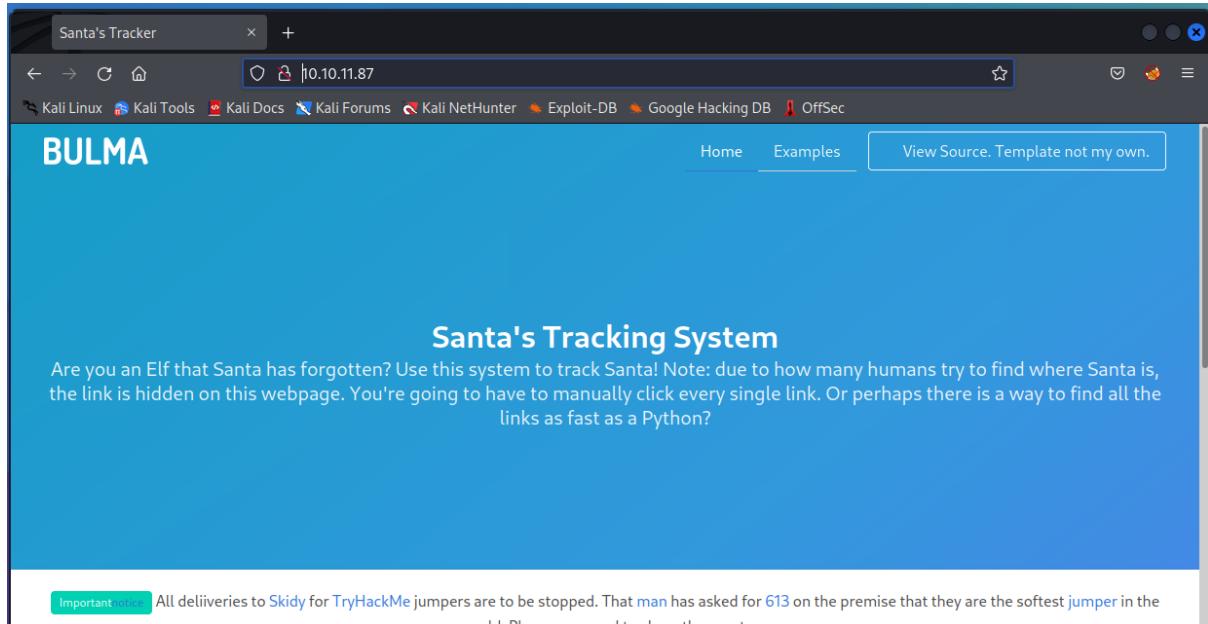
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 14.98 seconds
```

### Question 2

We entered the IP address with the port number on firefox's search bar.



We successfully entered Santa's Tracking System. The template used is BULMA which is shown on the top left of the page.



### Question 3

We view the page source of the page and found the directory for the API.

```
<li><a href="#">Flimsy Lavenrock</a></li>
<li><a href="#">Maven Mousie Lavender</a></li>
</ul>
</div>
<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 free</a></li>
    <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>
```

### Question 4

We created a python script to loop through the possible odd number API key and run it in the terminal.

```
#!/usr/bin/python3
# apibruter.py - A simple tool to brute force API keys
# Usage: python3 apibruter.py <target> <start> <end>
# Example: python3 apibruter.py http://10.10.11.87:80/api/ 1 100
# This script is for educational purposes only.

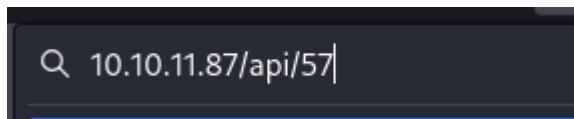
import requests
for api_key in range(1,100,2):
    print(f"api_key {api_key}")
    html = requests.get(f"http://10.10.11.87:80/api/{api_key}")
    print(html.text)
```

```
[root@kali ~]# python3 apibruter.py http://10.10.11.87:80/api/ 1 100
[...]
api_key 57
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
api_key 59
{"item_id":59,"q":"Error. Key not valid!"}
```

We got a response from API key 57.

```
{"item_id":55,"q":"Error. Key not valid!"}
api_key 57
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
api_key 59
{"item_id":59,"q":"Error. Key not valid!"}
```

We visited the API directory and use 57 as our API key.



The Raw Data was shown after pressing the raw data tab in Firefox.

File Edit View Go Help

File Actions Edit View

Santa's Tracker x http://10.10.11.87/ x 10.10.11.87/api/57

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

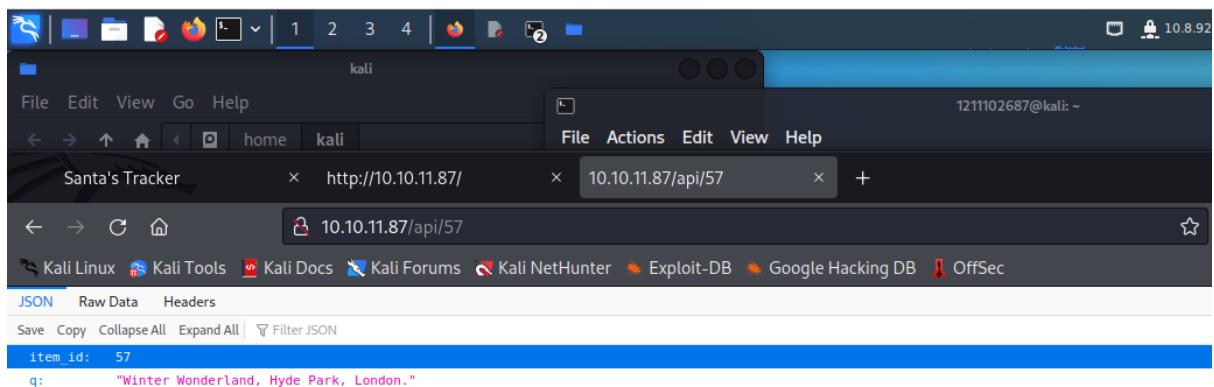
JSON Raw Data Headers

Save Copy Pretty Print

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

### Question 5

Santa's location was shown after we successfully entered the API directory with 57 as the API key.



### Question 6

We created a python script to loop through the possible odd number API key on the API directory from the range of 1-100 and run it in the terminal.

```
—(1211102687㉿kali)-[~]
$ python3 apibruter.py
```

```
{"item_id":55,"q":"Error. Key not valid!"}
api_key 57
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
api_key 59
{"item_id":59,"q":"Error. Key not valid!"}
```

#### **Thought Process/Methodology:**

First, we ran a nmap scan on the IP address to find open ports, and two open ports were found. We decided to investigate port 80 as it is used to host an HTTP service. We entered the IP address with the port number on firefox's search bar and successfully entered Santa's Tracking System. The template used which is BULMA was shown on the top left of the page. We right-clicked and view the page source of the page. We found the directory for the API in the page source. We created a python script to loop through the possible odd number API key on the API directory from the range of 1-100 and run it in the terminal. We got a response from API key 57. We visited the API directory and use 57 as our API key. Santa's location was shown after we successfully entered the API directory with 57 as the API key. The Raw Data was also shown after pressing the raw data tab in Firefox.

### Day 17[Reverse Engineering]- ReverseELFneering

**Tools used:** Kali Linux, Firefox, ssh, radare2

**Solution/walkthrough:**

#### Question 1

The size in bytes for the data type can be obtained from the table in the task description in THM.

##### 3. Register me this, register me that...

The core of assembly language involves using registers to do the following:

- Transfer data between memory and register, and vice versa
- Perform arithmetic operations on registers and data
- Transfer control to other parts of the program Since the architecture is x86-64, the registers are 64 bit and Intel has a list of 16 registers:

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

The command to analyze the program in radare2 can be seen in the task description in THM.

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

The command to set a breakpoint in radare2 can be seen in the task description in THM.

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

The command to execute the program until we hit a breakpoint can be seen in the task description in THM.

Running `dc` will execute the program until we hit the breakpoint. Once we hit the breakpoint and print out the main function, the rip which is the current instruction shows where execution has stopped. From the notes above, we know that the **mov** instruction is used to

#### Question 5

We SSH into our target machine with the username `elfmceager` and the password `adventofcyber`

```
File Actions Edit View Help
└── (1211102753㉿kali)-[~] └───SSH: timers and/or timeouts modified
└── $ ssh elfmceager@10.10.253.253 └───Compression parms modified
The authenticity of host '10.10.253.253 (10.10.253.253)' can't be established.
d.
ED25519 key fingerprint is SHA256:+Yl8Ef3BjQ7HNTMf6qew50LnmiqEXXSzLqgX82k/RS
g.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.253.253' (ED25519) to the list of known ho
sts.
elfmceager@10.10.253.253's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-128-generic x86_64) HWADDR=08:00:27:00:00:00
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
System information as of Thu Jul 14 15:44:36 UTC 2022
 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 use radare2 to enter debug mode so that we can learn more about the challenge1 file. We do a full analysis of the file using the aa command. We run the afl command to find a list of the functions and then filter our results with grep.

```
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1497 started ...
= attach 1497 1497
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
asm.bits 64
[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]> afl | grep main
0x00400b4d    1 35          sym.main
0x00400de0   10 1007 → 219  sym._libc_start_main
0x00403840   39 661   → 629  sym._nl_find_domain
0x00403ae0   308 5366 → 5301 sym._nl_load_domain
0x00415ef0    1 43          sym._IO_switch_to_main_get_area
0x0044ce10    1 8           sym._dl_get_dl_main_map
0x00470430    1 49         sym._IO_switch_to_main_wget_area
0x0048f9f0    7 73   → 69  sym._nl_fnddomain_subfreeres
0x0048fa40   16 247   → 237 sym._nl_unload_domain
```

There is a function at main. We examine the assembly code at main by running the command pdf @main.

```
[0x00400a30]> pdf @ main
;-- main: initialization Sequence Completed
/ (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
```

We use db to put a breakpoint at the address of local\_ch. We use dc to run the program and we received a message that we have stopped at the breakpoint we set. We use the command px @rbp-0xc to view the contents of the local\_ch variable.

```

[0x00400a30]> db 0x00400b51  this configuration may cache passwords in memory →
[0x00400a30]> dc 11 Initialization Sequence Completed
hit breakpoint at: 400b51
[0x00400b51]> px @ rbp-0xc
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0x7ffea2ce51b4 0000 0000 1890 6b00 0000 0000 4018 4000 ....k..@.@
0x7ffea2ce51c4 0000 0000 e910 4000 0000 0000 0000 0000 ....@...
0x7ffea2ce51d4 0000 0000 0000 0000 0100 0000 e852 cea2 .....R..
0x7ffea2ce51e4 fe7f 0000 4d0b 4000 0000 0000 0000 0000 ....M.@
0x7ffea2ce51f4 0000 0000 1700 0000 0100 0000 0000 0000 .....
0x7ffea2ce5204 0000 0000 0000 0000 0200 0000 0000 0000 .....
0x7ffea2ce5214 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffea2ce5224 0000 0000 0000 0000 0000 0000 0004 4000 .....@.
0x7ffea2ce5234 0000 0000 74a9 3af8 964d eaf2 e018 4000 ....t.:..M..@.
0x7ffea2ce5244 0000 0000 0000 0000 0000 0000 1890 6b00 .....k.
0x7ffea2ce5254 0000 0000 0000 0000 0000 0000 74a9 1a6b .....t..k
0x7ffea2ce5264 8a08 170d 74a9 8ee9 964d eaf2 0000 0000 ....t...M..
0x7ffea2ce5274 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffea2ce5284 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffea2ce5294 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffea2ce52a4 0000 0000 0000 0000 0000 0000 0000 0000 .....

```

We use ds to go to the next one and we obtained the value of 1.

```

[0x00400b51]> ds 11 net_iface_up set tun0 up
[0x00400b51]> px @rbp-0xc
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF table 0 metric
0x7ffea2ce51b4 0100 0000 1890 6b00 0000 0000 4018 4000 ....k..@.@
0x7ffea2ce51c4 0000 0000 e910 4000 0000 0000 0000 0000 ....@...
0x7ffea2ce51d4 0000 0000 0000 0000 0100 0000 e852 cea2 .....R..
0x7ffea2ce51e4 fe7f 0000 4d0b 4000 0000 0000 0000 0000 ....M.@
0x7ffea2ce51f4 0000 0000 1700 0000 0100 0000 0000 0000 .....
0x7ffea2ce5204 0000 0000 0000 0000 0200 0000 0000 0000 .....
0x7ffea2ce5214 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffea2ce5224 0000 0000 0000 0000 0000 0000 0004 4000 .....@.
0x7ffea2ce5234 0000 0000 74a9 3af8 964d eaf2 e018 4000 ....t.:..M..@.
0x7ffea2ce5244 0000 0000 0000 0000 0000 0000 1890 6b00 .....k.
0x7ffea2ce5254 0000 0000 0000 0000 0000 0000 74a9 1a6b .....t..k
0x7ffea2ce5264 8a08 170d 74a9 8ee9 964d eaf2 0000 0000 ....t...M..
0x7ffea2ce5274 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffea2ce5284 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffea2ce5294 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffea2ce52a4 0000 0000 0000 0000 0000 0000 0000 0000 .....

```

## Question 6

We use ds to move to the next line and then use dr to see the value of eax.

```

[0x004000b51]> ds    ; Outgoing Data Channel: Using 512 bit message hash 'SHA512' for HM
[0x004000b51]> dr    ; Incoming Data Channel: Cipher 'AES-256-CBC' initialized with 256
rax = 0x00000006    ; Incoming Data Channel: Using 512 bit message hash 'SHA512' for HM
rbx = 0x00400400    ; net_route_v4_best_gw query: dst 0.0.0.0
rcx = 0x0044b9a0    ; net_route_v4_best_gw result: via 10.0.2.2 dev eth0
rdx = 0x7ffea2ce52f8 ; ROUTE_GATEWAY 10.0.2.2/255.255.255.0 IFACE=eth0 HWADDR=08:00:27:13
r8 = 0x01000000    ; TUN/TAP device tun0 opened
r9 = 0x006bb8e0    ; net_iface_mtu_set: mtu 1500 for tun0
r10 = 0x00000015   ; net_iface_up: set tun0 up
r11 = 0x00000000   ; net_addr_v4_add: 10.8.99.93/16 dev tun0
r12 = 0x004018e0   ; net_route_v4_add: 10.10.0.0/16 via 10.8.0.1 dev [NULL] table 0 metric 0
r13 = 0x00000000   ; WARNING: this configuration may cache passwords in memory -- use
r14 = 0x006b9018   ; Initialization Sequence Completed
r15 = 0x00000000
rsi = 0x7ffea2ce52e8
rdi = 0x00000001
rsp = 0x7ffea2ce51c0
rbp = 0x7ffea2ce51c0
rip = 0x00400b66
rflags = 0x000000246
orax = 0xfffffffffffffff
[0x004000b51]> 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 b    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]
    ;-- rip:
  0x00400b66      8945fc      mov dword [local_4h], eax
  0x00400b69      b800000000  mov eax, 0
  0x00400b6e      5d          pop rbp
  0x00400b6f      c3          ret
[0x004000b51]> ds
[0x004000b51]> px @ rbp-0x4
- offset -      0 1 2 3 4 5 6 7 8 9 A B C D E F  0123456789ABCDEF

```

### Question 7

We use the ds command to get to 0x00400b66 and then check the value of local\_4h with px @ rbp-0x4. The value is 6.

```

0x00400b5f 8b45f4      mov eax, dword [local_ch]    SHA512 for HMAC
0x00400b62 0faf45f8    imul eax, dword [local_8h]   initialized with 256 bi
0x00400b61 ;-- rip:   8945fc    cast_gw_q    mov dword [local_4h], eax
0x00400b60 b800000000  mov eax, 0
0x00400b6e 5d          pop rbp    255.0 IP=FACE=eth0 HWADDR=08:00:27:50:
0x00400b6f c3          ret

[0x00400b51]> ds ll net_iface mtu set: mtu 1500 for tun0
[0x00400b51]> px @ rbp-0x4
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0x7ffea2ce51bc 0600 0000 4018 4000 0000 0000 e910 4000 . . . @ . . . @ .
0x7ffea2ce51cc 0000 0000 0000 0000 0000 0000 0000 0000 . . . . . . . . .
0x7ffea2ce51dc 0100 0000 e852 cea2 fe7f 0000 4d0b 4000 . . . R . . . M @ .
0x7ffea2ce51ec 0000 0000 0000 0000 0000 0000 1700 0000 . . . . . . .
0x7ffea2ce51fc 0100 0000 0000 0000 0000 0000 0000 0000 . . .
0x7ffea2ce520c 0200 0000 0000 0000 0000 0000 0000 0000 . . .
0x7ffea2ce521c 0000 0000 0000 0000 0000 0000 0000 0000 . . .
0x7ffea2ce522c 0000 0000 0004 4000 0000 0000 74a9 3af8 . . . @ . t : .
0x7ffea2ce523c 964d eaf2 e018 4000 0000 0000 0000 0000 . M . . @ .
0x7ffea2ce524c 0000 0000 1890 6b00 0000 0000 0000 0000 . . . k .
0x7ffea2ce525c 0000 0000 74a9 1a6b 8a08 170d 74a9 8ee9 . . . t . k . . t ...
0x7ffea2ce526c 964d eaf2 0000 0000 0000 0000 0000 0000 . M ..
0x7ffea2ce527c 0000 0000 0000 0000 0000 0000 0000 0000 . . .
0x7ffea2ce528c 0000 0000 0000 0000 0000 0000 0000 0000 . . .
0x7ffea2ce529c 0000 0000 0000 0000 0000 0000 0000 0000 . . .
0x7ffea2ce52ac 0000 0000 0000 0000 0000 0000 0000 0000 . . .

[0x00400b51]>

```

### Thought Process/Methodology:

After reading the task description in THM, the size in bytes for the data type can be obtained from the table in the task description. The command aa is used to analyze the program in radare2. The command db is used to set a breakpoint in radare2. The command dc is used to execute the program until we hit a breakpoint. We SSH into our target machine with the username elfmceager and the password adventofcyber. We use radare2 to enter debug mode so that we can learn more about the challenge1 file. We do a full analysis of the file using the aa command. As most programs have an entry point defined as main, we run the afl command to find a list of the functions and then filter our results with grep. It turns out that there is a function at main. We examine the assembly code at main by running the command pdf @main. We use db to put a breakpoint at the address of local\_ch. Then, we use dc to run the program and we received a message that we have stopped at the breakpoint we set. We use the command px @rbp-0xc to view the contents of the local\_ch variable and then used ds to go to the next one. We obtained the value of 1. We use ds to move to the next line and then use dr to see the value of eax. We use the ds command to get to 0x00400b66 and then check the value of local\_4h with px @ rbp-0x4. The value is 6.

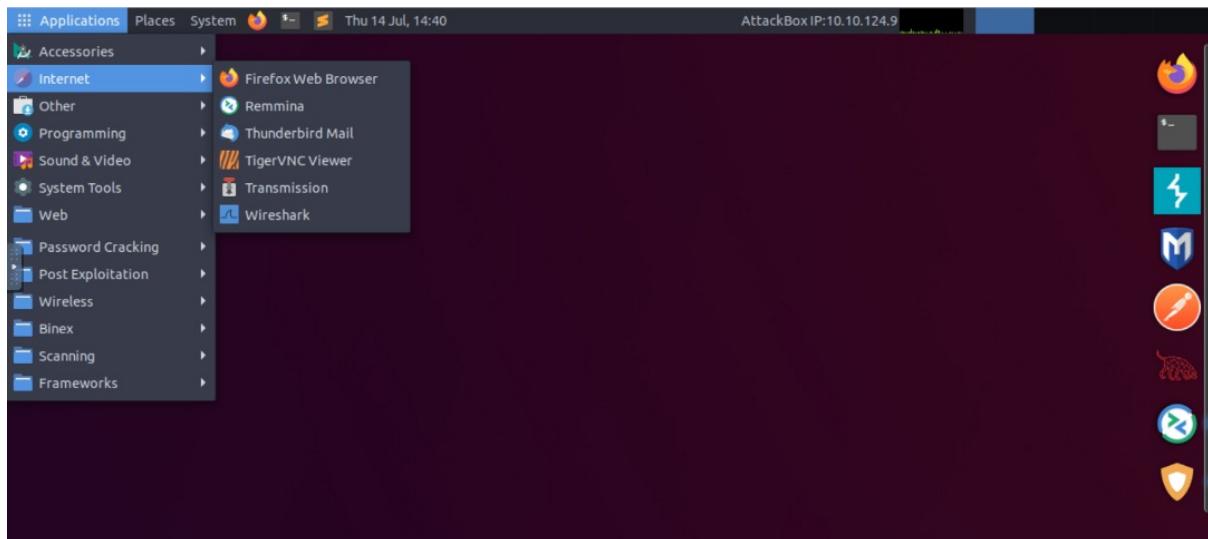
## **Day 18 [Reverse Engineering] - The Bits of Christmas**

**Tools used:** Kali Linux, Remmina, CyberChef

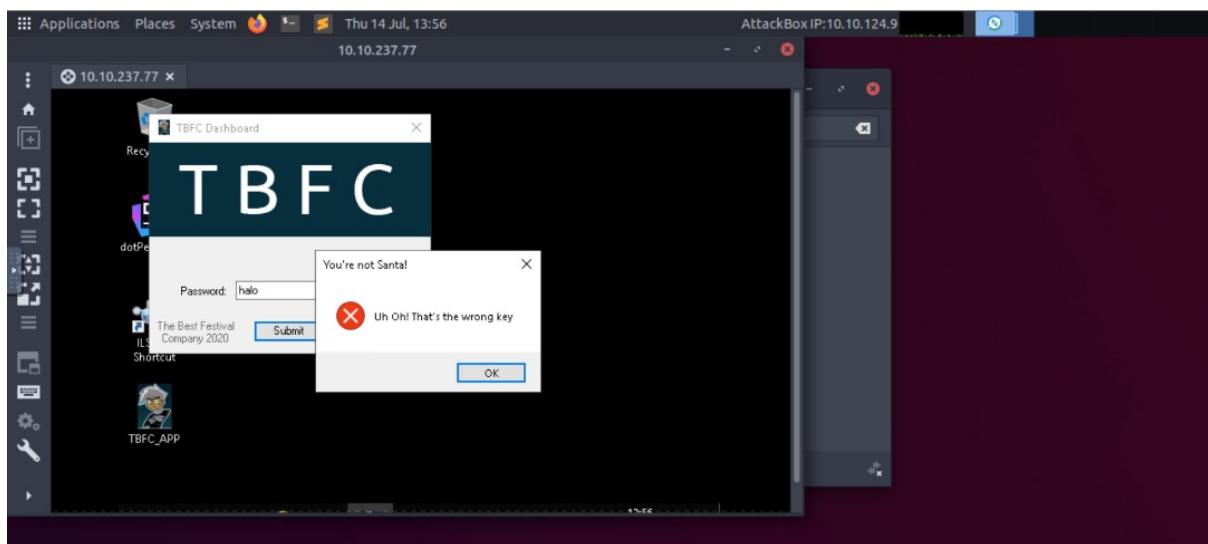
**Solution/walkthrough:**

### **Question 1**

First we open Remmina. We key in our IP address, username, and password which were provided in the task description in THM.

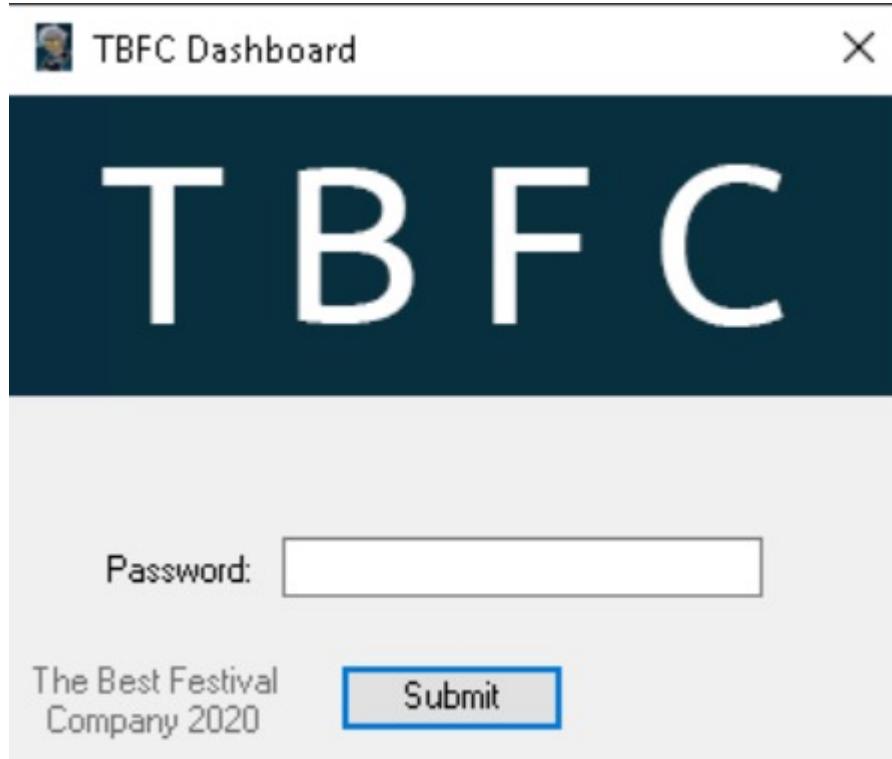


We tried entering the wrong password to see what will be shown.



### **Question 2**

The full name of TBFC can be seen on the bottom left of the TBFC Dashboard.



### Question 3

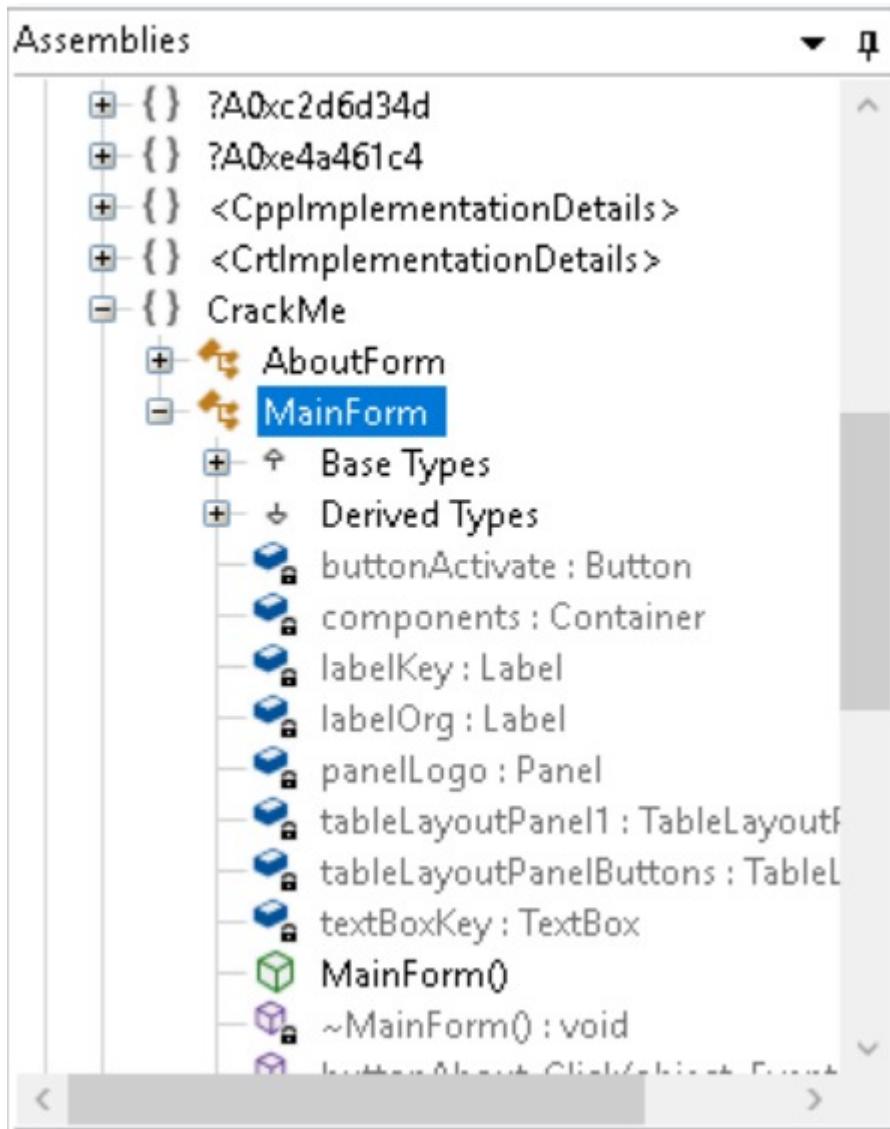
We opened TBFC\_APP in ILSpy and decompile it. We are asked to find Santa's password.

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

using System.Reflection;
using System.Runtime.Versioning;
using System.Security;
using System.Security.Permissions;

[assembly: SecurityRules(SecurityRuleSet.Level1)]
[assembly: TargetFramework(".NETFramework,Version=v4.6.1", FrameworkDisplayName = ".NET Framework 4.6.1")]
[assembly: SecurityPermission(SecurityAction.RequestMinimum, SkipVerification = true)]
[assembly: AssemblyVersion("0.0.0.0")]
```

The module that catches our attention is CrackMe as it contains a lot of information.



#### Question 4

It was shown that MainForm contains the information we are looking for.

```

private void InitializeComponent()
{
    System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(CrackMe));
    labelKey = new System.Windows.Forms.Label();
    textBoxKey = new System.Windows.Forms.TextBox();
    panelLogo = new System.Windows.Forms.Panel();
    tableLayoutPanelPanel1 = new System.Windows.Forms.TableLayoutPanel();
    buttonActivate = new System.Windows.Forms.Button();
    tableLayoutPanelButtons = new System.Windows.Forms.TableLayoutPanel();
    tableLayoutPanelPanelButtons = new System.Windows.Forms.TableLayoutPanel();
    labelOrg = new System.Windows.Forms.Label();
    tableLayoutPanelPanel1.SuspendLayout();
    tableLayoutPanelButtons.SuspendLayout();
    tableLayoutPanelPanelButtons.SuspendLayout();
    SuspendLayout();
    labelKey.Anchor = System.Windows.Forms.AnchorStyles.Right;
    labelKey.AutoSize = true;
    System.Drawing.Point location = new System.Drawing.Point(30, 14);
    labelKey.Location = location;
    labelKey.Name = "labelKey";
    System.Drawing.Size size = new System.Drawing.Size(56, 13);
    labelKey.Size = size;
    labelKey.TabIndex = 0;
    labelKey.Text = "Password:";
}

```

#### Question 5

After a bit of searching, we ended up finding something interesting in buttonActivate\_Click, it contains the word password.

```

    file ((uint)b <= (uint)b2)
    {
        if (b != 0)
        {
            ptr2 = (byte*)ptr2 + 1;
            ptr++;
            b = *(byte*)ptr;
            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.Information);
        return;
    }

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

```

## Question 6

We double-clicked the password string to check.

```

private unsafe void buttonActivate_Click(object sender, EventArgs e)
{
    IntPtr value = Marshal.StringToGlobalAnsi(textBoxKey.Text);
    sbyte* ptr = (sbyte*)System.Runtime.CompilerServices.Unsafe.AsPointer(ref <Module>._??_C@_0B@IKKDFEPG@santapasswordB21@);

    internal static <CppImplementationDetails>.$ArrayType$$BV0BR$$_$CDB global::<Module>._??_C@_0B@IKKDFEPG@santapasswordB21@

    if ((uint)b >= 1150)
    {
        while ((uint)b <= (uint)b2)
        {
            if (b != 0)
            {
                ptr2 = (byte*)ptr2 + 1;
                ptr++;
                b = *(byte*)ptr2;
                b2 = *(byte*)(*ptr);
                if ((uint)b < (uint)b2)
                {
                    .
                    .
                    .
                }
            }
        }
    }
}

```

We copied the hexadecimal and convert it to string using CyberChef. We left out the 00 as 00 is a null byte that terminates the string.

```
?_C@_0BB@IKKDFEPG@santapassword321@ : $ArrayType$$$BY0BB@$CBD
```

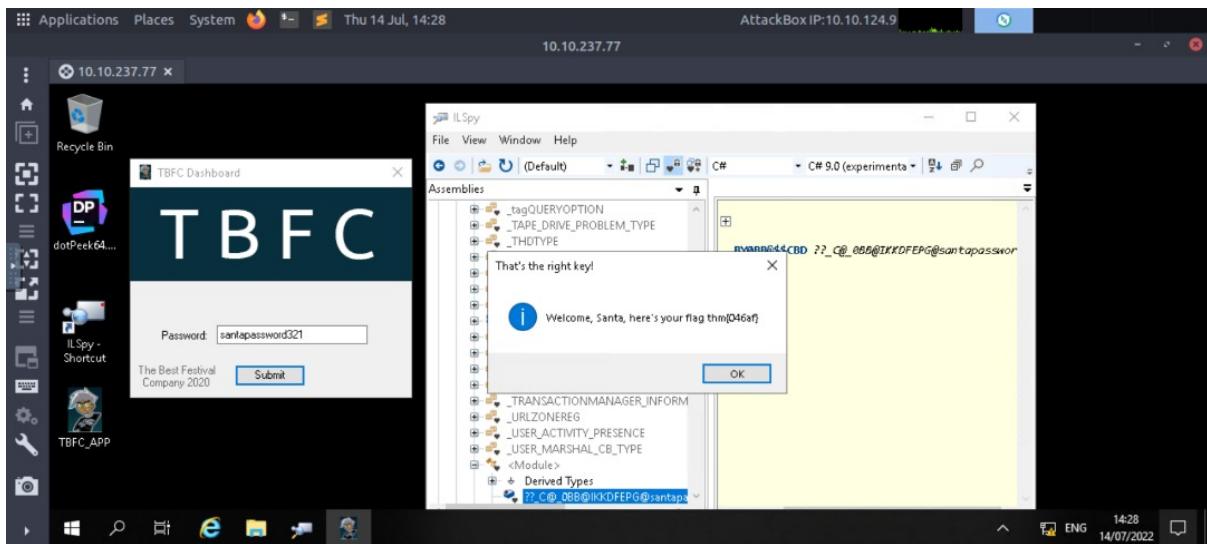
After converting, Santa's password was shown.

[https://gchq.github.io/CyberChef/#recipe=From\\_Hex\('Auto'\)&input=NzMgNjEgNkUgNzQgNjEgNzAgNjEgNzMgNzMgNzcgNkYgNzlgNjQgMzMgMzlzMzE](https://gchq.github.io/CyberChef/#recipe=From_Hex('Auto')&input=NzMgNjEgNkUgNzQgNjEgNzAgNjEgNzMgNzMgNzcgNkYgNzlgNjQgMzMgMzlzMzE)

Operations	Recipe	Input	Output
Search...	From Hex	73 61 6E 74 61 70 61 73 73 77 6F 72 64 33 32 31	santapassword321

## Question 7

After keying in the correct password, the flag was shown.



### Thought Process/Methodology:

First we open Remmina. We key in our IP address, username, and password which were provided in the task description in THM. Before entering the correct password, we tried entering the wrong password to see what will be shown. The full name of TBFC was shown on the bottom left of the TBFC Dashboard. We opened TBFC\_APP in ILSpy and decompile it. We are asked to find Santa's password. The module CrackMe caught our attention as it contains a lot of information. It was shown that MainForm contains the information we are looking for. After a bit of searching, we ended up finding something interesting in buttonActivate\_Click. It contains the word password. We double-clicked the password string to check it. We copied the hexadecimal and convert it to string using CyberChef. We left out the 00 as 00 is a null byte that terminates the string. After converting, Santa's password was shown. We keyed in the password and the flag was shown.

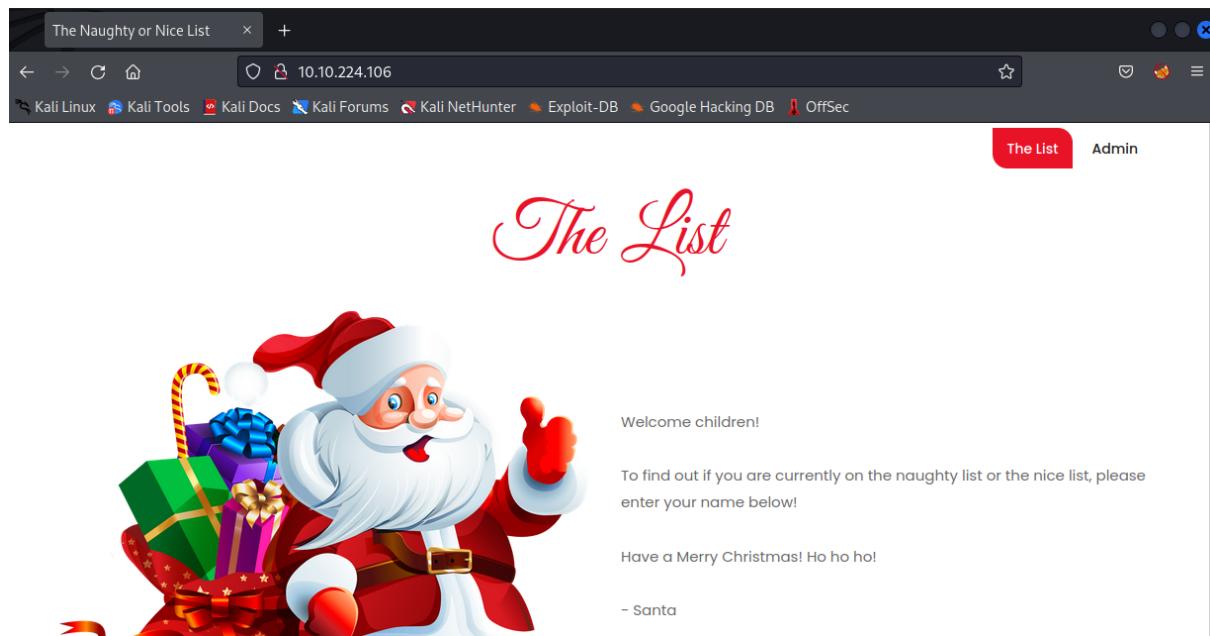
### Day 19 [Web Exploitation] - The Naughty or Nice List

**Tools used:** Kali Linux, Firefox, CyberChef

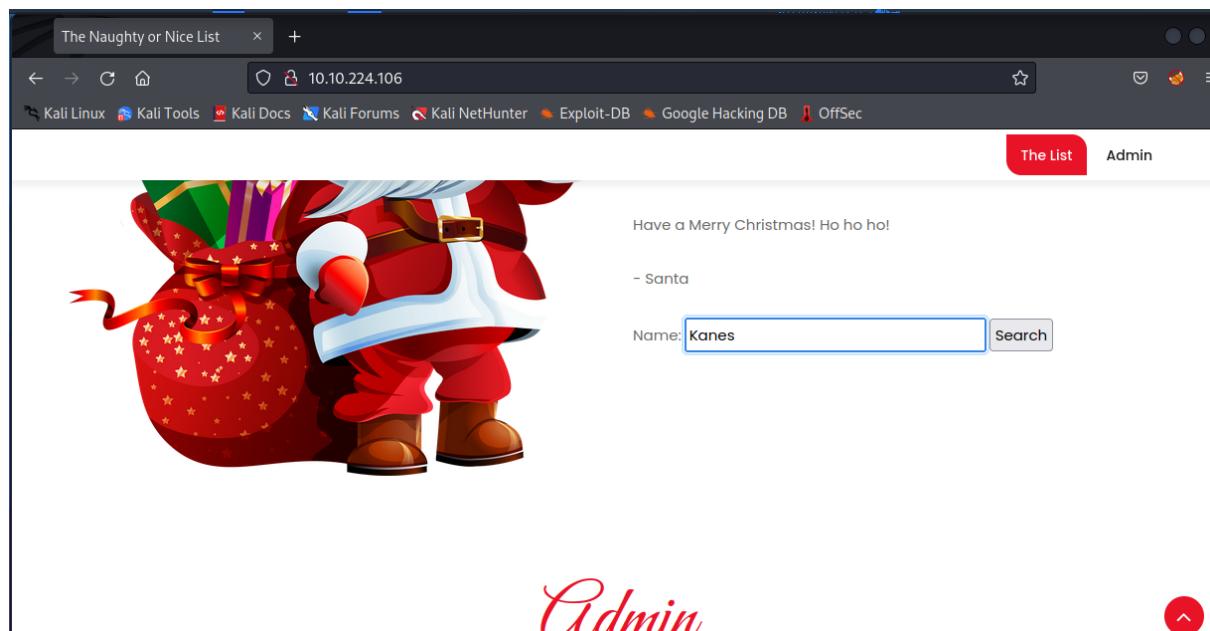
**Solution/walkthrough:**

#### Question 1

We entered the IP address and were shown a list.



We entered the given names in the form to determine if they are on the naughty or nice list.



We use URL Decoder on the value of the "proxy" parameter, and we get

<http://list.hohoho:8080/search.php?name=Tib3rius>

The screenshot shows the CyberChef interface. On the left, there's a sidebar with various operations like To Base64, From Base64, To Hex, From Hex, To Hexdump, From Hexdump, URL Decode, Regular expression, Entropy, and Fork. The main area has tabs for Recipe, Input, and Output. The Recipe tab shows 'URL Decode'. The Input field contains the URL `http%3A%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%3DTib3rius`. The Output field shows the decoded URL `http://list.hohoho:8080/search.php?name=Tib3rius`. At the bottom, there are buttons for 'STEP', 'BAKE!', and 'Auto Bake'.

## Question 2

We fetch the root of the same site and browse it. Not Found was displayed.

The screenshot shows a browser window with the URL `10.10.224.106/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%3DTib3rius`. The page content includes a large cartoon illustration of Santa Claus carrying a sack full of wrapped gifts. Text on the page says: "To find out if you are currently on the naughty list or the nice list, please enter your name below!" and "Have a Merry Christmas! Ho ho ho!". Below this, it says "- Santa". There is a search form with a 'Name:' input field and a 'Search' button. A blue banner at the bottom says "Not Found". Below the banner, a message states "The requested URL was not found on this server."

### Question 3

We try changing the port number from 8080 to just 80.



The page has a red header bar with "The List" and "Admin" buttons. The main content area includes a welcome message, a search field, and an error message.

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!

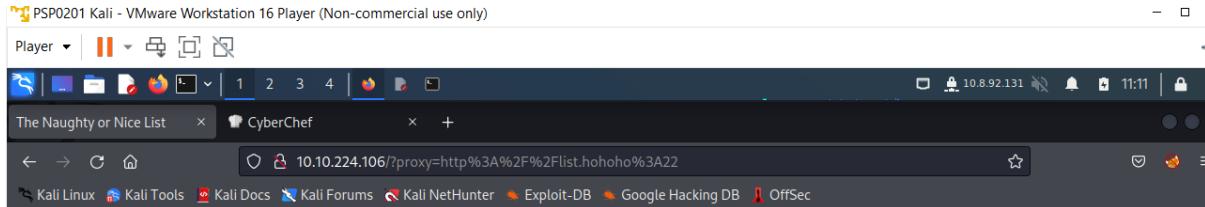
- Santa

Name:  Search

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

### Question 4

We try changing the port number to 22.



The page has a red header bar with "The List" and "Admin" buttons. The main content area includes a welcome message, a search field, and an error message.

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!

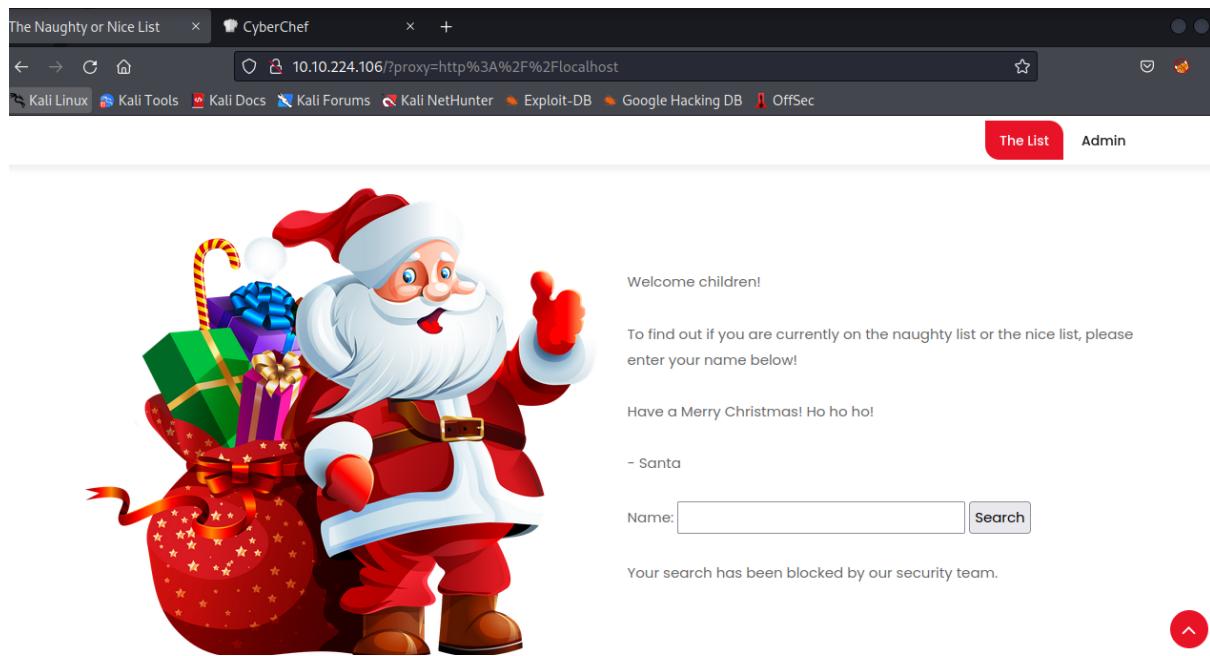
- Santa

Name:  Search

Recv failure: Connection reset by peer

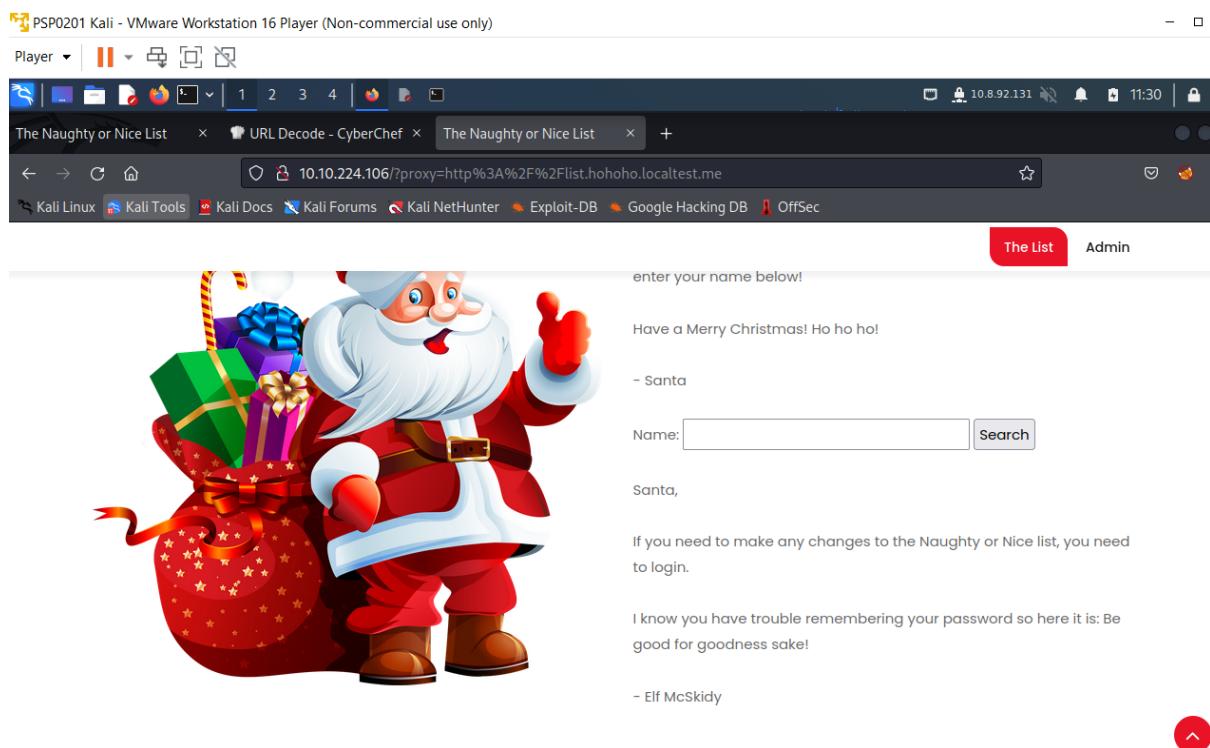
### Question 5

We access services running locally on the server by replacing the list.hohoho hostname with "localhost".



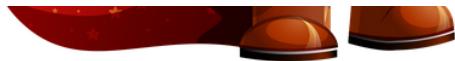
### Question 6

We set the hostname in the URL to "list.hohoho.localtest.me", bypass the check, and access local services. Santa's password was shown.



### Question 7

We guessed Santa's username as Santa and keyed in the password.



# Admin

Username:

Password:

We were then shown the administration list.

← → C ⌂ 10.10.224.106/admin.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

## List Administration

This page is currently under construction.

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

After deleting the naughty list, the flag was shown.

⊕ 10.10.224.106

THM{EVERYONE\_GETS\_PRESENTS}

OK

## **Thought Process/Methodology:**

We entered the IP address and were shown a naughty or nice list. We entered the given names in the form to determine if they are on the naughty or nice list. We use URL Decoder on the value of the "proxy" parameter, and we get <http://list.hohoho:8080/search.php?name=Tib3rius>. We fetch the root of the same site and browse it. Not Found was then displayed. We try changing the port number from 8080 to just 80 but it was not open. We try changing the port number to 22, port 22 is open but did not understand what was sent. We access services running locally on the server by replacing the list.hohoho hostname with "localhost" but our search was blocked. We set the hostname in the URL to "list.hohoho.localtest.me", bypass the check, and access local services. Santa's password was then shown. We guessed Santa's username as Santa and keyed in the password. We were then shown the administration list and the option to delete the naughty list. After deleting the naughty list, the flag was shown.

## Day 20 [Blue Teaming] - PowershellIF to the rescue

**Tools used:** Kali Linux, Firefox, Powershell, SSH

## Solution/walkthrough:

### Question 1

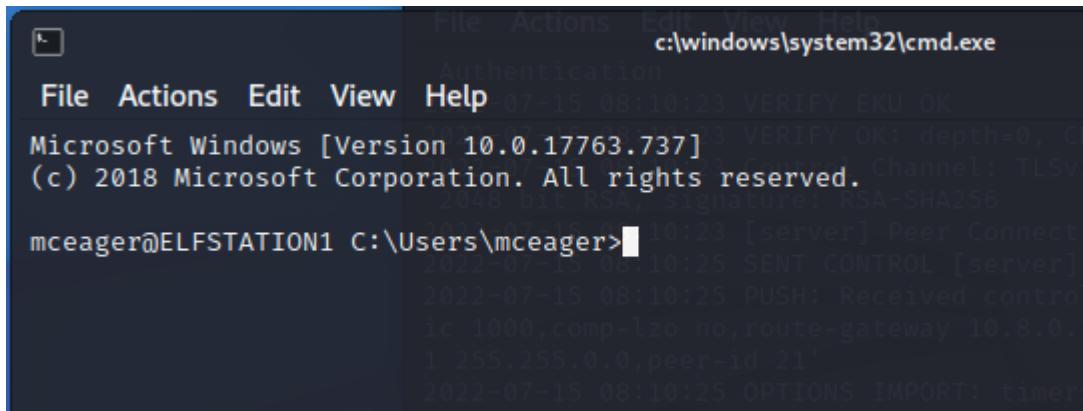
We entered the ssh manual using SSH command and got the answer.

```
File Actions Edit View Help  
Authentication 1211102687@kali: ~  
2022-07-15 08:10:23 VERIFY EKU OK  
2022-07-15 08:10:23 VERIFY OK depth=0, CN=server  
(1211102687@kali)-[~] 2022-07-15 08:10:23 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, 2048 bit RSA, signature RSA-SHA256  
$ ssh usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface] Connection Initiated with [AF_INET]1  
[-b bind_address] [-c cipher_spec] [-D [bind_address:]port] JSSH_REQUEST5' (status=1)  
[-E log_file] [-e escape_char] [-F configfile] [-I pkcs11] ssgage: 'PUSH_REPLY,route  
[-i identity_file] [-J [user@]host[:port]] [-L address]  
[-l login_name] [-m mac_spec] [-O ctrl_cmd] [-o option] [-p port]  
[-Q query_option] [-R address] [-S ctl_path] [-W host:port] or timeouts modified  
[-w local_tun[:remote_tun]] destination [command [argument ...]] ms modified  
(1211102687@kali)-[~] 2022-07-15 08:10:25 OPTIONS IMPORT: --ifconfig/up options modified  
(1211102687@kali)-[~] 2022-07-15 08:10:25 OPTIONS IMPORT: route options modified
```

## Question 2

We use SSH to connect to the remote machine and key in the password.

```
File Actions Edit View  
Authentication 2022-07-15 08:10:23 VER  
1211102687@kali) ~ [~]  
$ ssh -l mceager 10.10.86.106  
mceager@10.10.86.106's password:
```



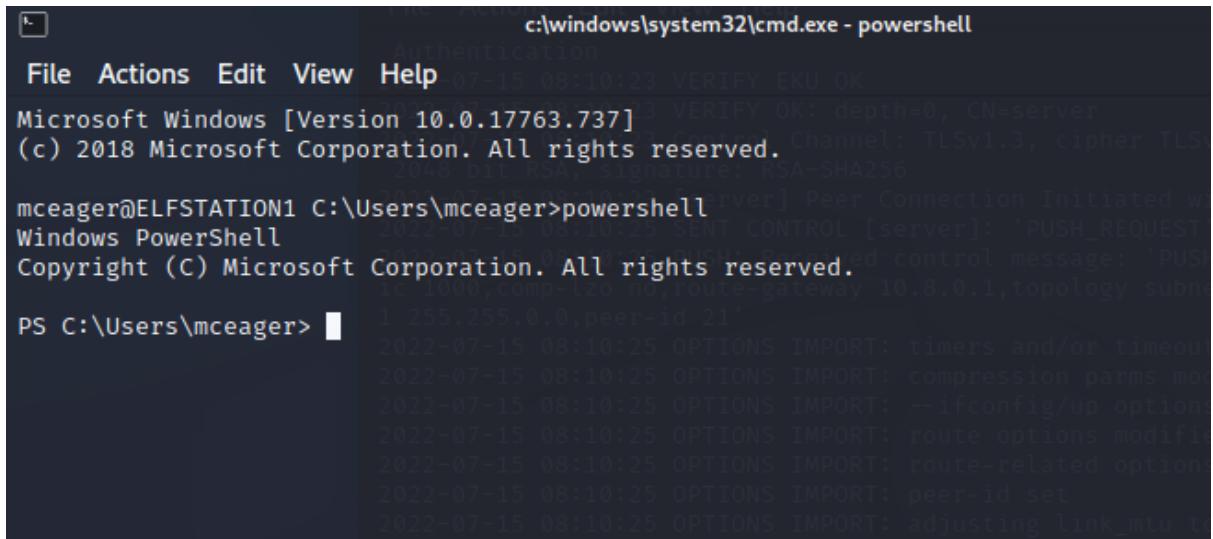
A screenshot of a Windows Command Prompt window titled "File Actions Edit View Help" and "c:\windows\system32\cmd.exe". The window displays a log of network traffic. Key lines include:

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

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

The log shows various network events such as "VERIFY OK", "Peer Connect", "SENT CONTROL", "PUSH", and "OPTIONS IMPORT". The timestamp for the log entries is 2022-07-15 08:10:23.

We launched Powershell.



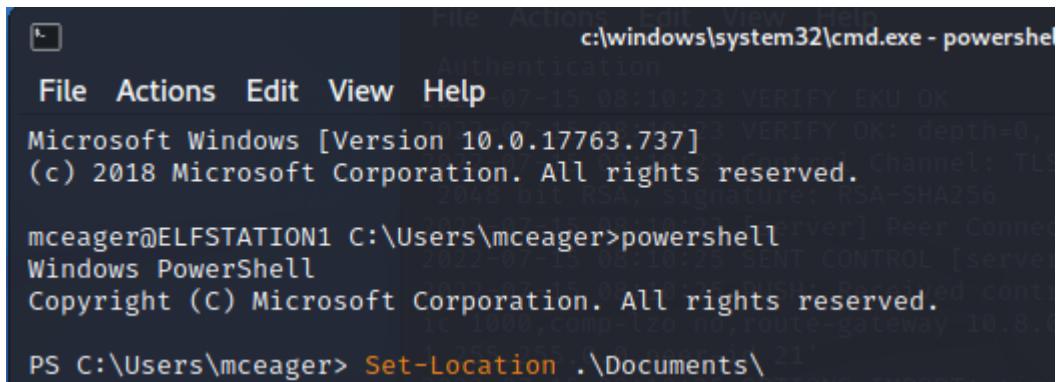
A screenshot of a Windows PowerShell window titled "File Actions Edit View Help" and "c:\windows\system32\cmd.exe - powershell". The window displays a log of network traffic. Key lines include:

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

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

The log shows various network events such as "Peer Connection Initiated", "SENT CONTROL", "PUSH\_REQUEST", and "OPTIONS IMPORT". The timestamp for the log entries is 2022-07-15 08:10:25.

We navigate to the Documents folder.



A screenshot of a Windows PowerShell window titled "File Actions Edit View Help" and "c:\windows\system32\cmd.exe - powershell". The window displays a log of network traffic. Key lines include:

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

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

The log shows the command "Set-Location .\Documents\" being run. The timestamp for the log entries is 2022-07-15 08:10:25.

We use GET to obtain the hidden elf file within the Documents folder and read the content.

```

PS C:\Users\mceager\Documents> Get-ChildItem -File -Hidden
2022-07-15 08:10:25 OPTIONS IMPORT: compression parms modified
2022-07-15 08:10:25 OPTIONS IMPORT: --ifconfig/up options modified
2022-07-15 08:10:25 OPTIONS IMPORT: route options modified
2022-07-15 08:10:25 OPTIONS IMPORT: route-related options modified
2022-07-15 08:10:25 OPTIONS IMPORT: peer-id set
2022-07-15 08:10:25 Using link_mtu to 1625
2022-07-15 08:10:25 Using cipher 'AES-256-CBC'
2022-07-15 08:10:25 Cipher 'AES-256-CBC' initialized
2022-07-15 08:10:25 Outgoing Data Channel: Using 512 bit message has
2022-07-15 08:10:25 Incoming Data Channel: Cipher 'AES-256-CBC' initialized
2022-07-15 08:10:25 net_route_v4_best_gw query: dst 0.0.0.0
2022-07-15 08:10:25 net_route_v4_best_gw result: via 192.168.80.2 dev tun0
2022-07-15 08:10:25 ROUTE_GATEWAY 192.168.80.2/255.255.255.255.0 IFACE=tun0
2022-07-15 08:10:25 TUN/TAP device tun0 opened
2022-07-15 08:10:25 net_iface_mtu set: mtu 1500 for tun0
2022-07-15 08:10:25 net_iface_up: set tun0 up
2022-07-15 08:10:25 net_v4_add: 10.8.92.131/16 dev tun0
2022-07-15 08:10:25 net_v4_add: 10.10.0.0/16 via 10.8.0.1 dev tun0
2022-07-15 08:10:25 WAZERO configuration may cache passwords option to prevent this
2022-07-15 08:10:25 Initialization Sequence Completed

Directory: C:\Users\mceager\Documents
Mode                LastWriteTime      Length Name
----              -----          -----      -----    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> Get-Content e1fone.txt
All I want is my '2 front teeth'!!!
PS C:\Users\mceager\Documents>

```

### Question 3

We change the directory to Desktop.

```

All I want is my '2 front teeth'!!!
PS C:\Users\mceager\Documents> cd ..
PS C:\Users\mceager> Set-Location .\Desktop\

```

We searched for the hidden folder in Desktop. We change our directory to the folder and proceeded to read the contents inside the text file. The movie elf 2 wants was shown.

```

ic 1000,comp-lzo,no,route-gateway 10.8.0.1,topology subnet,ping 5,p
PS C:\Users\mceager\Desktop> Get-ChildItem -Hidden
2022-07-15 08:10:25 OPTIONS IMPORT: timers and/or timeouts modified
2022-07-15 08:10:25 OPTIONS IMPORT: compression parms modified
2022-07-15 08:10:25 OPTIONS IMPORT: --ifconfig/up options modified
2022-07-15 08:10:25 OPTIONS IMPORT: route options modified
2022-07-15 08:10:25 OPTIONS IMPORT: route-related options modified
2022-07-15 08:10:25 OPTIONS IMPORT: peer-id set
2022-07-15 08:10:25 Using link_mtu to 1625
2022-07-15 08:10:25 Using cipher 'AES-256-CBC'
2022-07-15 08:10:25 Cipher 'AES-256-CBC' initialized
2022-07-15 08:10:25 Outgoing Data Channel: Using 512 bit message has
2022-07-15 08:10:25 Incoming Data Channel: Cipher 'AES-256-CBC' initialized
2022-07-15 08:10:25 net_route_v4_best_gw query: dst 0.0.0.0
2022-07-15 08:10:25 net_route_v4_best_gw result: via 192.168.80.2 dev tun0
2022-07-15 08:10:25 ROUTE_GATEWAY 192.168.80.2/255.255.255.255.0 IFACE=tun0
2022-07-15 08:10:25 TUN/TAP device tun0 opened
2022-07-15 08:10:25 net_iface_mtu set: mtu 1500 for tun0
2022-07-15 08:10:25 net_iface_up: set tun0 up
2022-07-15 08:10:25 net_v4_add: 10.8.92.131/16 dev tun0
2022-07-15 08:10:25 net_v4_add: 10.10.0.0/16 via 10.8.0.1 dev tun0
2022-07-15 08:10:25 WAZERO configuration may cache passwords option to prevent this
2022-07-15 08:10:25 Initialization Sequence Completed

Directory: C:\Users\mceager\Desktop\elf2wo
Mode                LastWriteTime      Length Name
----              -----          -----      -----    Name
-d--h--           12/7/2020  11:26 AM        102 elf2wo
-a-hs-           12/7/2020  10:29 AM        282 desktop.ini

PS C:\Users\mceager\Desktop> cd elf2wo
PS C:\Users\mceager\Desktop\elf2wo> ls
2022-07-15 08:10:25 net_route_v4_best_gw query: dst 0.0.0.0
2022-07-15 08:10:25 net_route_v4_best_gw result: via 192.168.80.2 dev tun0
2022-07-15 08:10:25 ROUTE_GATEWAY 192.168.80.2/255.255.255.255.0 IFACE=tun0
2022-07-15 08:10:25 TUN/TAP device tun0 opened
2022-07-15 08:10:25 net_iface_mtu set: mtu 1500 for tun0
2022-07-15 08:10:25 net_iface_up: set tun0 up
2022-07-15 08:10:25 net_v4_add: 10.8.92.131/16 dev tun0
2022-07-15 08:10:25 net_v4_add: 10.10.0.0/16 via 10.8.0.1 dev tun0
2022-07-15 08:10:25 WAZERO configuration may cache passwords option to prevent this
2022-07-15 08:10:25 Initialization Sequence Completed

Directory: C:\Users\mceager\Desktop\elf2wo
Mode                LastWriteTime      Length Name
----              -----          -----      -----    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

We change the directory to Windows and proceeded to System32 as it contains most of the important files of the Operating System.

```
PS C:\Users\mceager\Desktop\elf2wo> cd C:/Windows
PS C:\Windows> ls
2022-07-15 08:10:25 SENT CONTROL [server]: 'PUSH_REQUEST'
2022-07-15 08:10:25 PUSH: Received control message: 'PUS
ic 1000,comp-lzo no,route-gateway 10.8.0.1,topology subn
1 255.255.0.0,peer-id 21'
Directory: C:\Windows
2022-07-15 08:10:25 OPTIONS IMPORT: timers and/or timeout
2022-07-15 08:10:25 OPTIONS IMPORT: compression parms mo
Mode          LastWriteTime    Length Name
_____
d----        9/15/2018  12:19 AM      0 ADFS
d----        9/15/2018  12:19 AM      0 appcompat
d----        9/6/2019   5:31 PM      0 apppatch
d----        12/7/2020  10:50 AM      0 AppReadiness
d-r--        9/15/2018  2:11 AM      0 assembly
d----        9/15/2018  12:19 AM      0 bcastdvr
d----        9/15/2018  12:19 AM      0 Boot
```

```
PS C:\Windows> cd System32
PS C:\Windows\System32> █
```

We filtered the contents of System 32 and set it so that we find directories that contain the number 3.

```
PS C:\Windows\System32> Get-ChildItem -Hidden -Directory -Filter "*3*"
2022-07-15 08:10:25 dst 0.0.0.0
2022-07-15 08:10:25 net_route_v4_best_gw result: via 192.168.
2022-07-15 08:10:25 ROUTE_GATEWAY 192.168.80.2/255.255.255.0
Directory: C:\Windows\System32
2022-07-15 08:10:25 TUN/TAP device tun0 opened
2022-07-15 08:10:25 net_iface_mtu_set: mtu 1500 for tun0
2022-07-15 08:10:25 net_iface_up: set tun0 up
2022-07-15 08:10:25 net_iface_v4_add: 10.8.92.131/16 dev tun0
2022-07-15 08:10:25 net_iface_v4_add: 10.10.0.0/16 via 10.8.0
2022-07-15 08:10:25 WARN 3lfthr3e configuration may cache pas
2022-07-15 08:10:25 Initialization Sequence Completed
PS C:\Windows\System32> cd 3lfthr3e
```

### Question 5

There is no list of contents in the folder which means the files are hidden, so we tried to get the hidden files.

```

PS C:\Windows\System32> Get-ChildItem -Hidden -Directory -Filter "*3*"
2022-07-15 08:10:25 OPTIONS IMPORT: timers and/or timeouts modified
2022-07-15 08:10:25 OPTIONS IMPORT: compression parms modified
2022-07-15 08:10:25 OPTIONS IMPORT: --ifconfig/up options modified
2022-07-15 08:10:25 OPTIONS IMPORT: route options modified
2022-07-15 08:10:25 OPTIONS IMPORT: route-related options modified
Mode          LastWriteTime    Length Name
-->--        11/23/2020   3:26 PM      3lfthr3e
PS C:\Windows\System32> cd 3lfthr3e
PS C:\Windows\System32\3lfthr3e> ls
PS C:\Windows\System32\3lfthr3e> dir
PS C:\Windows\System32\3lfthr3e> Get-ChildItem -Hidden
2022-07-15 08:10:25 net_route_v4_best_gw result: via 192.168.80.2 dev eth0
2022-07-15 08:10:25 ROUTE_GATEWAY 192.168.80.2/255.255.255.0 IFACE=eth0 HWA
2022-07-15 08:10:25 TUN/TAP device tun0 opened
Directory: C:\Windows\System32\3lfthr3e
net_iface_mtu_set: mtu 1500 for tun0
2022-07-15 08:10:25 net_iface_up: set tun0 up
2022-07-15 08:10:25 net_addr_v4_add: 10.8.92.131/16 dev tun0
Mode          LastWriteTime    Length Name
-->--        11/17/2020   10:58 AM      85887 1.txt
-->--        11/23/2020   3:26 PM      12061168 2.txt
PS C:\Windows\System32\3lfthr3e>

```

There are too many words in the file so we used the measure-object command to count the total number of words in the file.

```

arbor
mediawiki
configurations
poison
PS C:\Windows\System32\3lfthr3e> cat 1.txt | Measure-Object -Word
2022-07-15 08:10:25 ROUTE_GATEWAY 192.168.80.2/255.255.255.0 IFACE=eth0 HWA
2022-07-15 08:10:25 TUN/TAP device tun0 opened
2022-07-15 08:10:25 net_iface_mtu_set: mtu 1500 for tun0
2022-07-15 08:10:25 net_iface_up: set tun0 up
2022-07-15 08:10:25 net_addr_v4_add: 10.8.92.131/16 dev tun0
2022-07-15 08:10:25 net_route_v4_add: 10.10.0.0/16 via 10.8.0.1
2022-07-15 08:10:25 WARNING: this configuration may cache password
Lines Words Characters Property
--- --- --- --- 
9999
PS C:\Windows\System32\3lfthr3e>

```

## Question 6

We try to find what words are at the indexes of 551 and 6991 in the 1 text file.

```

option to prevent this
PS C:\Windows\System32\3lfthr3e> (cat 1.txt)[551,6991]
Red
Ryder
PS C:\Windows\System32\3lfthr3e>

```

## Question 7

We try to find the string that has the pattern of “redryder” in the 2 text file.

```

yhunkadqujzeo
cqbrbonednnd
PS C:\Windows\System32\3lfthr3e> cat 2.txt | Select-String -Pattern "redryder"
option to prevent this
redryderbbgun
2022-07-15 08:10:25 Initialization Sequence Completed
PS C:\Windows\System32\3lfthr3e>

```

### **Thought Process/Methodology:**

First, we entered the ssh manual using SSH command. We use SSH to connect to the remote machine and key in the password. Then we launched Powershell and navigated to the Documents folder. We use GET to obtain the hidden elf file within the Documents folder and read the content. Elf 1 wants 2 front teeth. We change the directory to Desktop. We searched for the hidden folder in Desktop. Then, we change our directory to the folder and proceeded to read the contents inside the text file. We change the directory to Windows and proceeded to System32 as it contains most of the important files of the Operating System. We filtered the contents of System 32 and set it so that we find directories that contain the number 3. There is no list of contents in the folder which means the files are hidden, so we tried to get the hidden files. There are too many words in the file so we used the measure-object command to count the total number of words in the file. We try to find what words are at the indexes of 551 and 6991 in text file 1. We try to find the string that has the pattern of “redryder” in text file 2 and found redryderbbgun.

