

PSP0201

Week 5

Writeup

Group Name: Blessing Software

Members

| ID | Name | Role |
|------------|--------------|--------|
| 1211103213 | Uwais | Leader |
| 1211103184 | Muzaffar | Member |
| 1211103149 | Dzakry Hariz | Member |
| 1211102082 | Thanussha | Member |

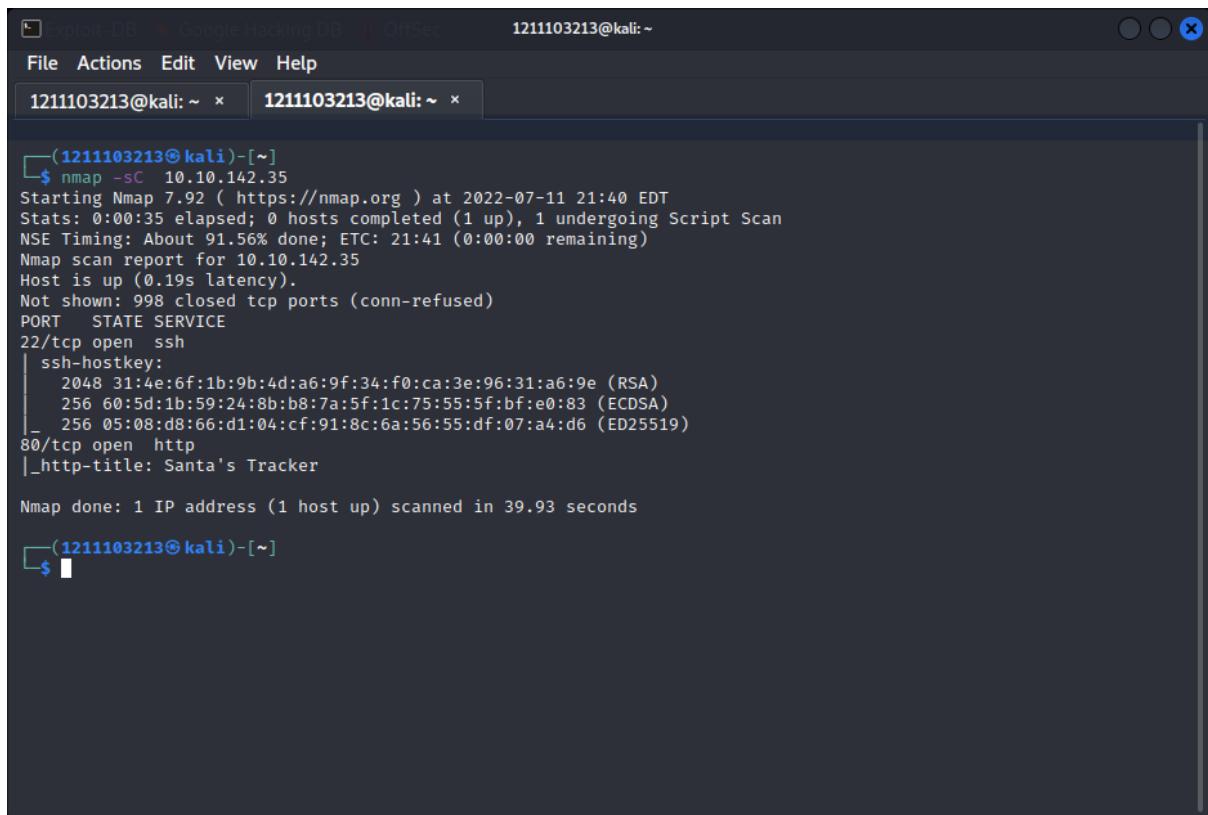
Day 16: Scripting – Help! Where is Santa?

Tools used: Kali Linux, Firefox, nmap, python3

Solution/walkthrough:

Question 1

Searched through the open ports using nmap.



The screenshot shows a terminal window with two tabs. The current tab displays the output of an nmap script scan on host 10.10.142.35. The output shows port 22/tcp is open and listening for ssh, and port 80/tcp is open and listening for http. The http service is identified as "Santa's Tracker".

```
(1211103213㉿kali)-[~]
$ nmap -sC 10.10.142.35
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-11 21:40 EDT
Stats: 0:00:35 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 91.56% done; ETC: 21:41 (0:00:00 remaining)
Nmap scan report for 10.10.142.35
Host is up (0.19s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
          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
          |_http-title: Santa's Tracker

Nmap done: 1 IP address (1 host up) scanned in 39.93 seconds

(1211103213㉿kali)-[~]
$
```

Question 2

Went to the website and looked for the template.

| Category | Category | Category |
|--|---|---|
| Lorem ipsum dolor sit amet | Labore et dolore magna aliqua | Objects in space |
| Vestibulum erratoisse | Kanban airis sum eschelor | Playing cards with coyote |
| Lorem ipsum dolor sit amet | Modular modern free | Goodbye Yellow Brick Road |
| Aisia caisia | The king of clubs | The Garden of Forking Paths |
| Murphy's law | The Discovery Dissipation | Future Shock |
| Flimsy Laverock | Course Correction | |
| Maven Mousie Lavender | Better Angels | |

Question 3

Viewed the page source and looked for the directory.

```
<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>
</div>
```

OR

Used python scripting to search through the website for links.

1211103213@kali:~ x 1211103213@kali:~ x 1211103213@kali:~ x

GNU nano 5.9 script.py

```
#!/usr/bin/env python3

from bs4 import BeautifulSoup
import requests

html = requests.get('http://10.10.142.35:80')

soup = BeautifulSoup(html.text, "lxml")

links = soup.find_all('a')

for link in links:
    print (link)
```

[Read 14 lines]

[Read 14 lines]

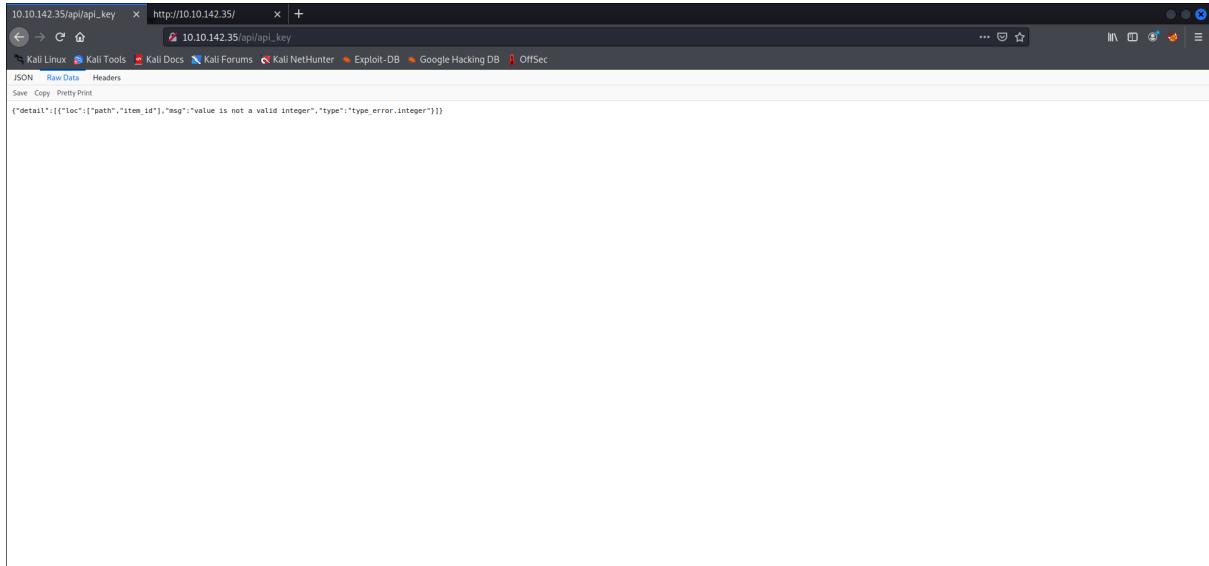
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

```
1211103213@kali:~ x 1211103213@kali:~ x 1211103213@kali:~ x
(1211103213@kali)-[~]
$ python3 script.py
<a class="navbar-item" href="/">

</a>
<a href="#">Home</a>
<a href="#">Examples</a>
<a class="button is-white is-outlined" href="https://github.com/BulmaTemplates/bulma-templates/blob/master/templates/hero.html">
<span class="icon">
<i class="fa fa-github"></i>
</span>
<span title="Hello from the other side">View Source. Template not my own.</span>
</a>
<a href="https://tryhackme.com">Santa</a>
<a href="https://tryhackme.com">Santa</a>
<a href="https://tryhackme.com">humans</a>
<a href="https://tryhackme.com">click</a>
<a href="https://tryhackme.com">Python</a>
<a href="https://tryhackme.com">notice</a>
<a href="https://tryhackme.com">Skidy</a>
<a href="https://tryhackme.com">TryHackMe</a>
<a href="https://tryhackme.com">man</a>
<a href="https://tryhackme.com">613</a>
<a href="https://tryhackme.com">jumper</a>
<a href="#">Lorem ipsum dolor sit amet</a>
<a href="#">Vestibulum errato isse</a>
<a href="#">Lorem ipsum dolor sit amet</a>
<a href="#">Aisia caisia</a>
<a href="#">Murphy's law</a>
<a href="#">Flimsy Lavenrock</a>
<a href="#">Maven Mousie Lavender</a>
<a href="#">Labore et dolore magna aliqua</a>
<a href="#">Kanban airis sum eschelor</a>
<a href="http://machine_ip/api/api_key">Modular modern free</a>
<a href="#">The king of clubs</a>
<a href="#">The Discovery Dissipation</a>
<a href="#">Course Correction</a>
<a href="#">Better Angels</a>
<a href="#">Objects in space</a>
<a href="#">Playing cards with coyote</a>
<a href="#">Goodbye Yellow Brick Road</a>
<a href="#">The Garden of Forking Paths</a>
```

Question 4

Went to the API endpoint from the last question, and looked to the Raw Data section.

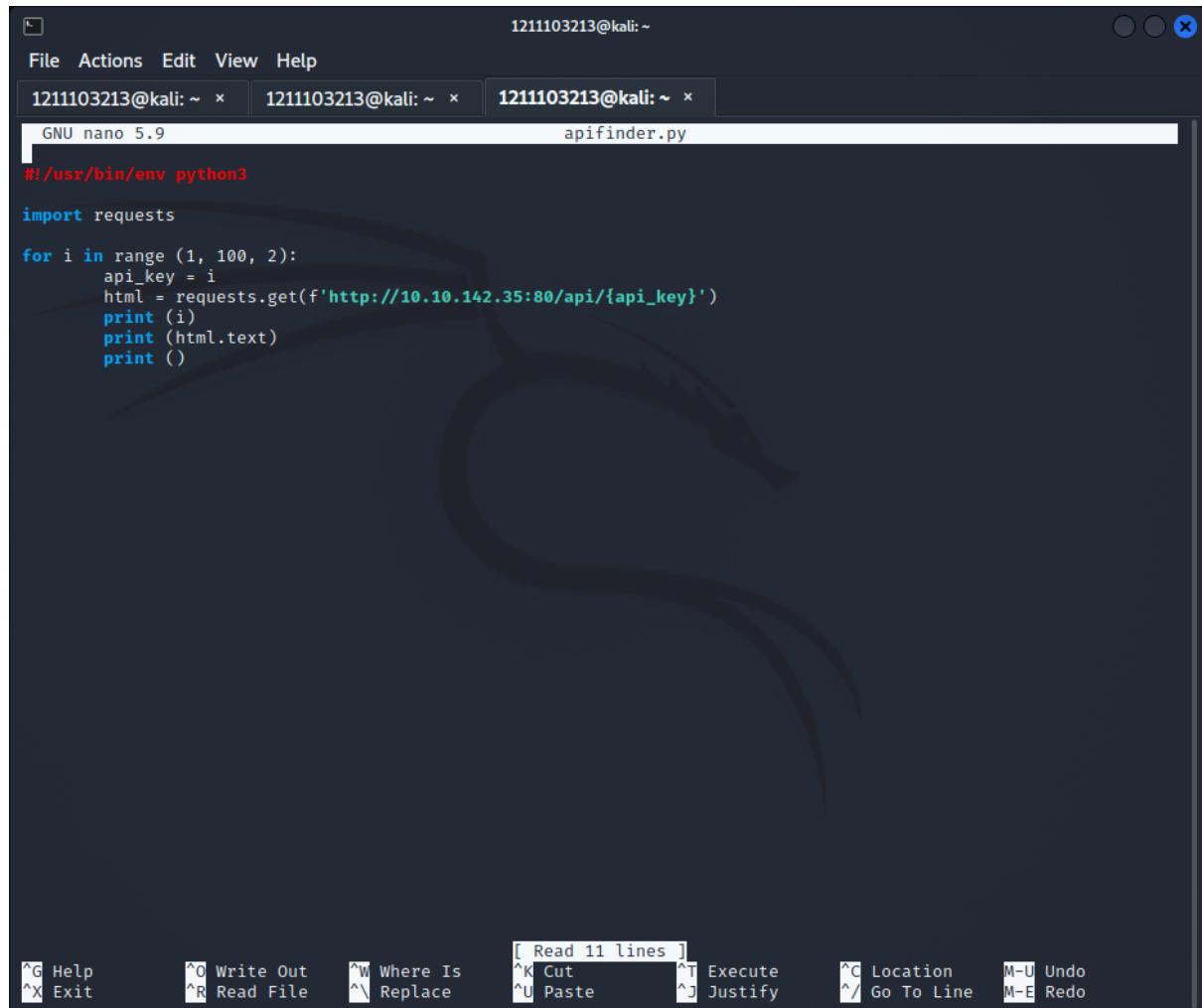


The screenshot shows a browser window with the URL `http://10.10.142.35/api/api_key`. The page title is "10.10.142.35/api/api_key". The browser interface includes tabs for Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec. Below the address bar, there are buttons for Save, Copy, and Pretty Print. The main content area displays a JSON object:

```
{"detail": [{"loc": ["path", "item_id"], "msg": "value is not a valid integer", "type": "type_error.integer"}]}
```

Question 5 and 6

Used python scripting to find the API key.



```
1211103213@kali:~
```

```
File Actions Edit View Help
```

```
1211103213@kali: ~ x 1211103213@kali: ~ x 1211103213@kali: ~ x
```

```
GNU nano 5.9 apifinder.py
```

```
#!/usr/bin/env python3

import requests

for i in range (1, 100, 2):
    api_key = i
    html = requests.get(f'http://10.10.142.35:80/api/{api_key}')
    print (i)
    print (html.text)
    print ()
```

```
[ Read 11 lines ]
```

```
^A Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo
```

```
1211103213@kali:~
```

```
File Actions Edit View Help  
1211103213@kali:~ x 1211103213@kali:~ x 1211103213@kali:~ x
```

```
39 {"item_id":39,"q":"Error. Key not valid!"}  
41 {"item_id":41,"q":"Error. Key not valid!"}  
43 {"item_id":43,"q":"Error. Key not valid!"}  
45 {"item_id":45,"q":"Error. Key not valid!"}  
47 {"item_id":47,"q":"Error. Key not valid!"}  
49 {"item_id":49,"q":"Error. Key not valid!"}  
51 {"item_id":51,"q":"Error. Key not valid!"}  
53 {"item_id":53,"q":"Error. Key not valid!"}  
55 {"item_id":55,"q":"Error. Key not valid!"}  
57 {"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}  
59 {"item_id":59,"q":"Error. Key not valid!"}  
61 {"item_id":61,"q":"Error. Key not valid!"}  
63 {"item_id":63,"q":"Error. Key not valid!"}  
65 {"item_id":65,"q":"Error. Key not valid!"}
```

Got Santa's location using the correct API key.

The screenshot shows a browser window with the URL `http://10.10.142.35/api/57`. The page displays a JSON object with one item:

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

Thought Process/Methodology:

We used nmap scripts to search for any open ports to look for the website. After getting it, we looked for the template and went into the page source for hints on the API key. We decided to use python scripting to automate the search for the API directory, and then to automate the search for the API keys. Through this, our IP address wouldn't get blocked. Finally, we got the API key and Santa's location.

Day 17: Reverse Engineering – ReverseELFneering

Tools used: Kali Linux, radare2

Solution/walkthrough:

Question 1

Referred to the notes given in TryHackMe.

| 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

Referred to the notes given in TryHackMe.

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

Referred to the notes given in TryHackMe.

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

Referred to the notes given in TryHackMe.

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 transfer values. This statement is

Question 5

Analyzed the file, and examined the code.

The screenshot shows a terminal window with two tabs. The left tab is titled '1211103213@kali: ~' and the right tab is titled 'elfmceager@tbfc-day-17: ~'. The terminal content displays assembly code for the 'main' function:

```
[0x00400a30]> pdf @main
;-- main:
/ (fcn) sym.main 35
sym.main ();
    ; var int local_ch @ rbp-0xc
    ; var int local_8h @ rbp-0x8
    ; var int local_4h @ rbp-0x4
        ; DATA XREF from 0x00400a4d (entry0)
0x00400b4d      55          push rbp
0x00400b4e      4889e5      mov rbp, rsp
0x00400b51      c745f4010000. mov dword [local_ch], 1
0x00400b58      c745f8060000. mov dword [local_8h], 6
0x00400b5f      8b45f4      mov eax, dword [local_ch]
0x00400b62      0faf45f8    imul eax, dword [local_8h]
0x00400b66      8945fc      mov dword [local_4h], eax
0x00400b69      b800000000  mov eax, 0
0x00400b6e      5d          pop rbp
0x00400b6f      c3          ret
```

Set a breakpoint at the corresponding movl instruction to see the value change within local_ch.

```
elfmceager@tbfc-day-17:~
```

```
File Actions Edit View Help
```

```
1211103213@kali: ~ x elfmceager@tbfc-day-17: ~ x
```

```
\ 0x00400b6f c3 ret
[0x00400a30]> db 0x00400b51
[0x00400a30]> dc
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
0>ffd12ce9224 0000 0000 1890 6b00 0000 0000 4018 4000 . . . k . . . @. @.
0>ffd12ce9234 0000 0000 e910 4000 0000 0000 0000 0000 . . . @. .
0>ffd12ce9244 0000 0000 0000 0000 0100 0000 5893 ce12 . . . X ...
0>ffd12ce9254 fd7f 0000 4d0b 4000 0000 0000 0000 0000 . . . M. @. .
0>ffd12ce9264 0000 0000 1700 0000 0100 0000 0000 0000 . . . .
0>ffd12ce9274 0000 0000 0000 0000 0200 0000 0000 0000 . . . .
0>ffd12ce9284 0000 0000 0000 0000 0000 0000 0000 0000 . . . .
0>ffd12ce9294 0000 0000 0000 0000 0000 0000 0004 4000 . . . @. .
0>ffd12ce92a4 0000 0000 f6ef 6646 1781 3f8c e018 4000 . . . fF . ? . @. .
0>ffd12ce92b4 0000 0000 0000 0000 0000 0000 1890 6b00 . . . k. .
0>ffd12ce92c4 0000 0000 0000 0000 0000 0000 f6ef 6652 . . . fR .
0>ffd12ce92d4 0aa4 c573 f6ef d257 1781 3f8c 0000 0000 . . . s . w . ? . .
0>ffd12ce92e4 0000 0000 0000 0000 0000 0000 0000 0000 . . . .
0>ffd12ce92f4 0000 0000 0000 0000 0000 0000 0000 0000 . . . .
0>ffd12ce9304 0000 0000 0000 0000 0000 0000 0000 0000 . . . .
0>ffd12ce9314 0000 0000 0000 0000 0000 0000 0000 0000 . . . .
[0x00400b51]> ds
[0x00400b51]> ds
[0x00400b51]> px @ rbp-0xc
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0>ffd12ce9224 0100 0000 0600 0000 0000 0000 4018 4000 . . . . . . . @. @.
0>ffd12ce9234 0000 0000 e910 4000 0000 0000 0000 0000 . . . @. .
0>ffd12ce9244 0000 0000 0000 0000 0100 0000 5893 ce12 . . . X ...
0>ffd12ce9254 fd7f 0000 4d0b 4000 0000 0000 0000 0000 . . . M. @. .
0>ffd12ce9264 0000 0000 1700 0000 0100 0000 0000 0000 . . . .
0>ffd12ce9274 0000 0000 0000 0000 0200 0000 0000 0000 . . . .
0>ffd12ce9284 0000 0000 0000 0000 0000 0000 0000 0000 . . . .
0>ffd12ce9294 0000 0000 0000 0000 0000 0000 0004 4000 . . . @. .
0>ffd12ce92a4 0000 0000 f6ef 6646 1781 3f8c e018 4000 . . . fF . ? . @. .
0>ffd12ce92b4 0000 0000 0000 0000 0000 0000 1890 6b00 . . . k. .
0>ffd12ce92c4 0000 0000 0000 0000 0000 0000 f6ef 6652 . . . fR .
0>ffd12ce92d4 0aa4 c573 f6ef d257 1781 3f8c 0000 0000 . . . s . w . ? . .
0>ffd12ce92e4 0000 0000 0000 0000 0000 0000 0000 0000 . . . .
0>ffd12ce92f4 0000 0000 0000 0000 0000 0000 0000 0000 . . . .
0>ffd12ce9304 0000 0000 0000 0000 0000 0000 0000 0000 . . . .
0>ffd12ce9314 0000 0000 0000 0000 0000 0000 0000 0000 . . . .
[0x00400b51]> 
```

Question 6

Went to the step that has the imull instruction to pass it, and then checked the value of eax.

```
elfmceager@tbfc-day-17:~
```

```
File Actions Edit View Help
1211103213@kali: ~ x elfmceager@tbfc-day-17: ~ x
```

```
[0x00400b51]> ds
[0x00400b51]> ds
[0x00400b51]> 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

[0x00400b51]> dr
rax = 0x00000006
rbx = 0x00400400
rcx = 0x0044b9a0
rdx = 0x7ffd12ce9368
r8 = 0x01000000
r9 = 0x006bb8e0
r10 = 0x00000015
r11 = 0x00000000
r12 = 0x004018e0
r13 = 0x00000000
r14 = 0x006b9018
r15 = 0x00000000
rsi = 0x7ffd12ce9358
rdi = 0x00000001
rsp = 0x7ffd12ce9230
rbp = 0x7ffd12ce9230
rip = 0x00400b66
rflags = 0x00000246
orax = 0xfffffffffffffff
[0x00400b51]>
```

Question 7

Went before the step that sets eax as 0, and checked the value of local_4h.

The screenshot shows the radare2 debugger interface with two tabs: 'elfmceager@kali: ~' and 'elfmceager@tbfc-day-17: ~'. The assembly code for the main function is displayed in the terminal window.

```

File Actions Edit View Help
1211103213@kali: ~ x elfmceager@tbfc-day-17: ~ x
rsp = 0x7ffd12ce9230
rbp = 0x7ffd12ce9230
rip = 0x00400b69
rflags = 0x00000246
orax = 0xffffffffffff
[0x00400b51]> 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]
0x00400b66 8945fc  mov dword [local_4h], eax
;-- rip:
0x00400b69 b800000000  mov eax, 0
0x00400b6e 5d      pop rbp
0x00400b6f c3      ret
[0x00400b51]> px @ rbp-0x4
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0x7ffd12ce922c 0600 0000 4018 4000 0000 0000 e910 4000 . ... @. @. .... @.
0x7ffd12ce923c 0000 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
0x7ffd12ce924c 0100 0000 5893 ce12 fd7f 0000 40db 4000 . ... X. .... M. @.
0x7ffd12ce925c 0000 0000 0000 0000 0000 0000 1700 0000 ..... .
0x7ffd12ce926c 0100 0000 0000 0000 0000 0000 0000 0000 ..... .
0x7ffd12ce927c 0200 0000 0000 0000 0000 0000 0000 0000 ..... .
0x7ffd12ce928c 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
0x7ffd12ce929c 0000 0000 0004 4000 0000 0000 f6ef 6646 . ... @. .... FF
0x7ffd12ce92ac 1781 3f8c e018 4000 0000 0000 0000 0000 .. ? ... @.
0x7ffd12ce92bc 0000 0000 1890 6b00 0000 0000 0000 0000 .. . k. .
0x7ffd12ce92cc 0000 0000 f6ef 6652 0aa4 c573 f6ef d257 .. . fR. ... s. ... W
0x7ffd12ce92dc 1781 3f8c 0000 0000 0000 0000 0000 0000 .. ?. .
0x7ffd12ce92ec 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
0x7ffd12ce92fc 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
0x7ffd12ce930c 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
0x7ffd12ce931c 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
[0x00400b51]> []

```

Thought Process/Methodology:

Using the radare2 commands, we analysed the challenge1 file and its code. From pdf @main, we can see the steps and disassemble the functions step by step. We saw the 3rd instruction moved the value to local_ch. Using breakpoint, we ran the code right after that instruction and stopped it, followed by looking into the value of local_ch to see its current value. Then, using ds to proceed to the next steps, we went to the imull instruction. There, we ran dr to see the eax value. Finally, we went to the 8th instruction where we saw the eax value changing to 0. By not passing it, we could check the value of local_4h before that instruction is passed.

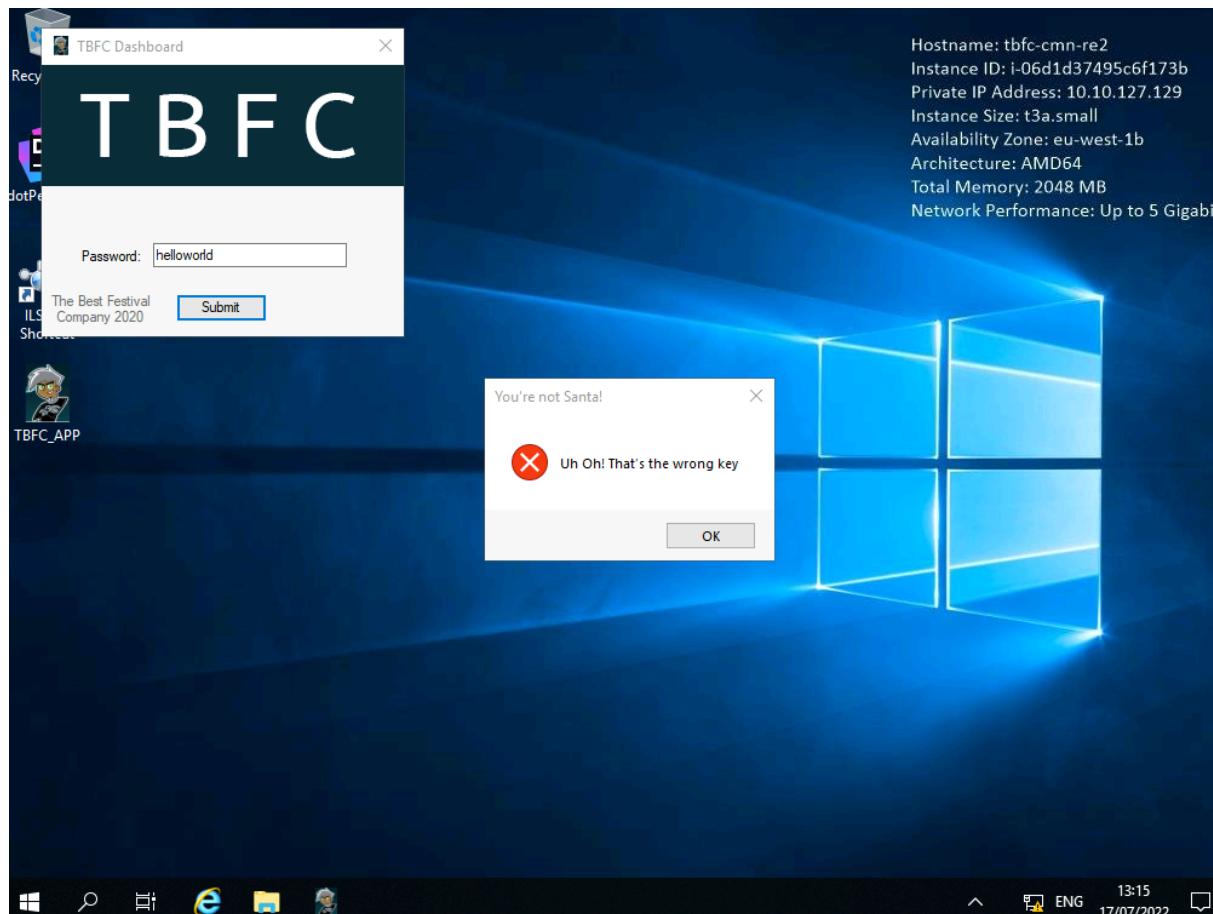
Day 18

Tools used: Kali Linux, Firefox

Solution/Walkthrough:

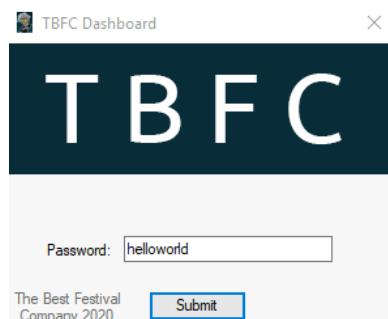
Question 1

Open the machine and click on TBFC_APP. Then, enter the wrong password.



Question 2

Look at the bottom left corner of the application.



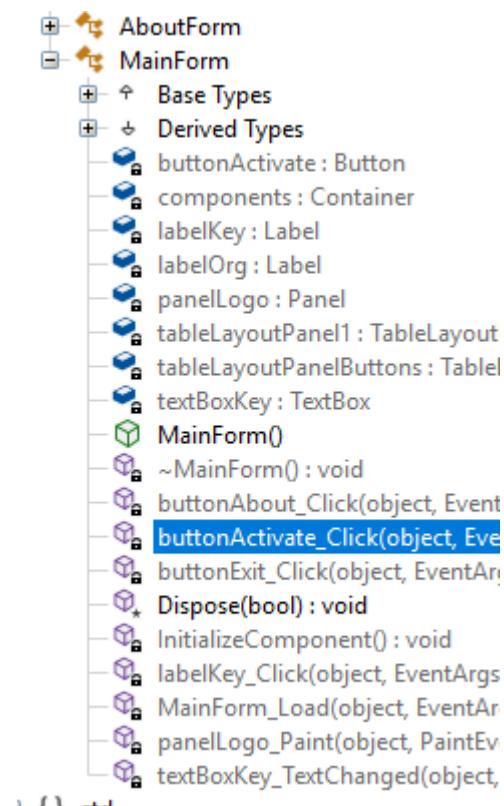
Question 3

Click on ILSpy. Look through TBFC_APP.

```
+ {} <CtrlImplementationDetails>
+ {} CrackMe
+ {} std
```

Question 4

Look on both forms.



```
+ AboutForm
+ MainForm
  + Base Types
  + Derived Types
    + buttonActivate : Button
    + components : Container
    + labelKey : Label
    + labelOrg : Label
    + panelLogo : Panel
    + tableLayoutPanelPanel1 : TableLayout
    + tableLayoutPanelButtons : TableLayoutPanel
    + textBoxKey : TextBox
  + MainForm()
  + ~MainForm() : void
  + buttonAbout_Click(object, EventArgs)
  + buttonActivate_Click(object, EventArgs)
  + buttonExit_Click(object, EventArgs)
  + Dispose(bool) : void
  + InitializeComponent() : void
  + labelKey_Click(object, EventArgs)
  + MainForm_Load(object, EventArgs)
  + panelLogo_Paint(object, PaintEventArgs)
  + textBoxKey_TextChanged(object, TextChangedEventArgs)
```

Question 5

Click on the file that has connection to the password.

The screenshot shows a debugger interface with two panes. The left pane displays a class hierarchy for a file named 'CrackMe'. The right pane shows the assembly code for the 'buttonActivate_Click' event handler.

```
// CrackMe.MainForm
using ...

private unsafe void buttonActivate_Click(object sender, EventArgs e)
{
    IntPtr value = Marshal.StringToHGlobalAnsi(textBoxKey.Text);
    sbyte* ptr = (sbyte*)System.Runtime.CompilerServices.Unsafe.AsPointer(ref <Module>._C@_0BB);
    void* ptr2 = (void*)value;
    byte b = *(byte*)ptr2;
    byte b2 = 115;
    if ((uint)b >= 115u)
    {
        while ((uint)b <= (uint)b2)
        {
            if (b != 0)
            {
                ptr2 = (byte*)ptr2 + 1;
                ptr++;
                b = *(byte*)ptr2;
                b2 = (byte)(*ptr);
                if ((uint)b < (uint)b2)
                {
                    break;
                }
                continue;
            }
            MessageBox.Show("Welcome, Santa, here's your flag thm{046af}", "That's the right key");
            return;
        }
    }
    MessageBox.Show("Uh Oh! That's the wrong key", "You're not Santa!", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

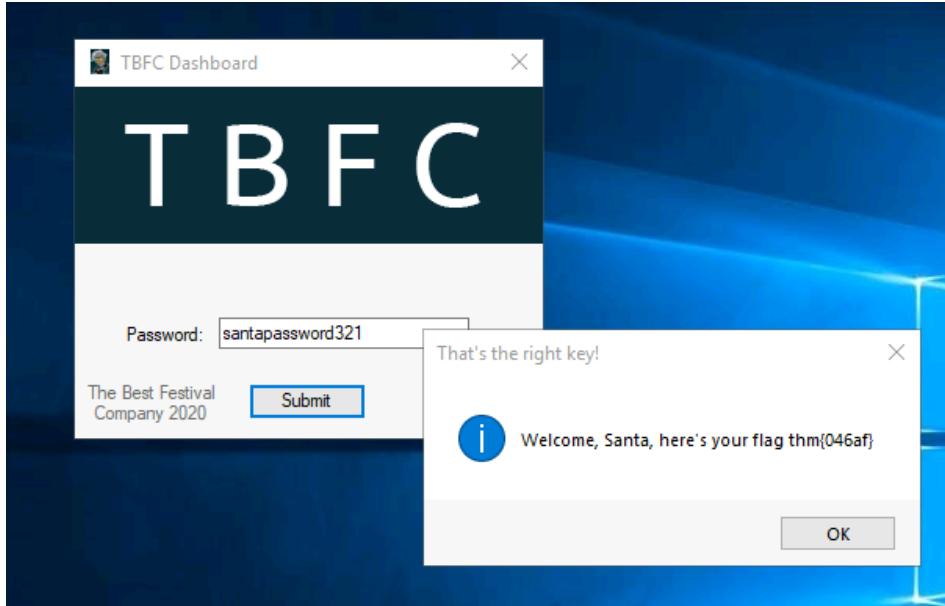
Question 6

Look for the password.

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

Question 7

Enter the password on the TBFC_APP.



Thought process/Methodology:

Once we found the way to the machine, click on ILSpy and search for TBFC_APP. There, we found the file, CrackMe. We thought there is some connection of submit button with the password. Then, we searched for the button activation file and found the password. After entering the password on the application, we finally obtained the flag.

Day 19: Web Exploitation - The Naughty or Nice List

Tools used: Kali Linux, FireFox

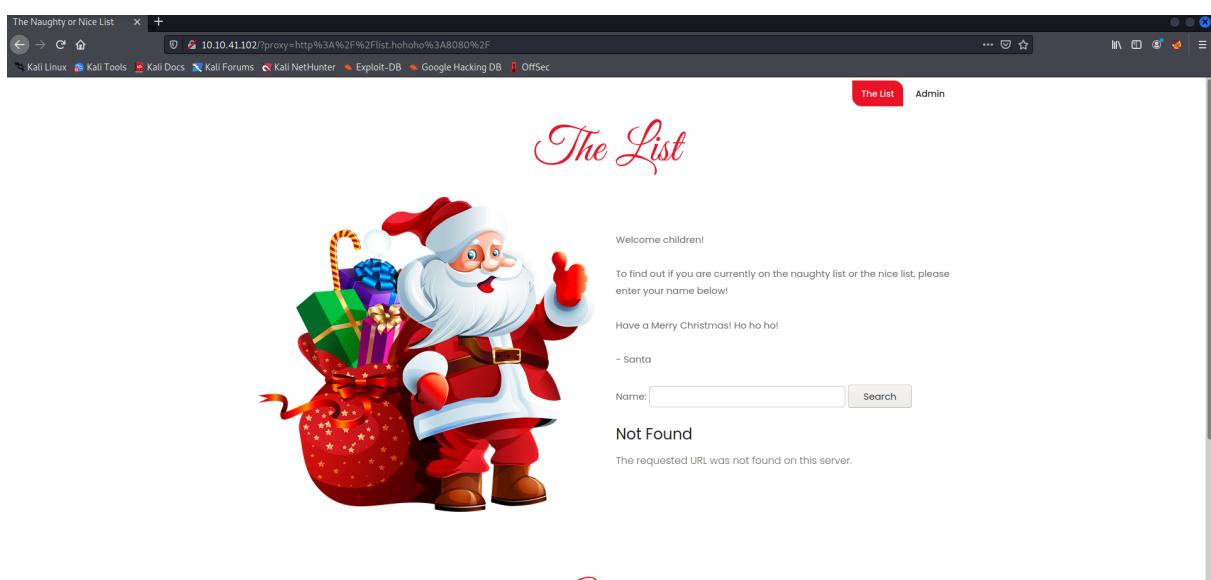
Solution/ Walkthrough:

Question 1

Search for every name in the list on the searchbar.

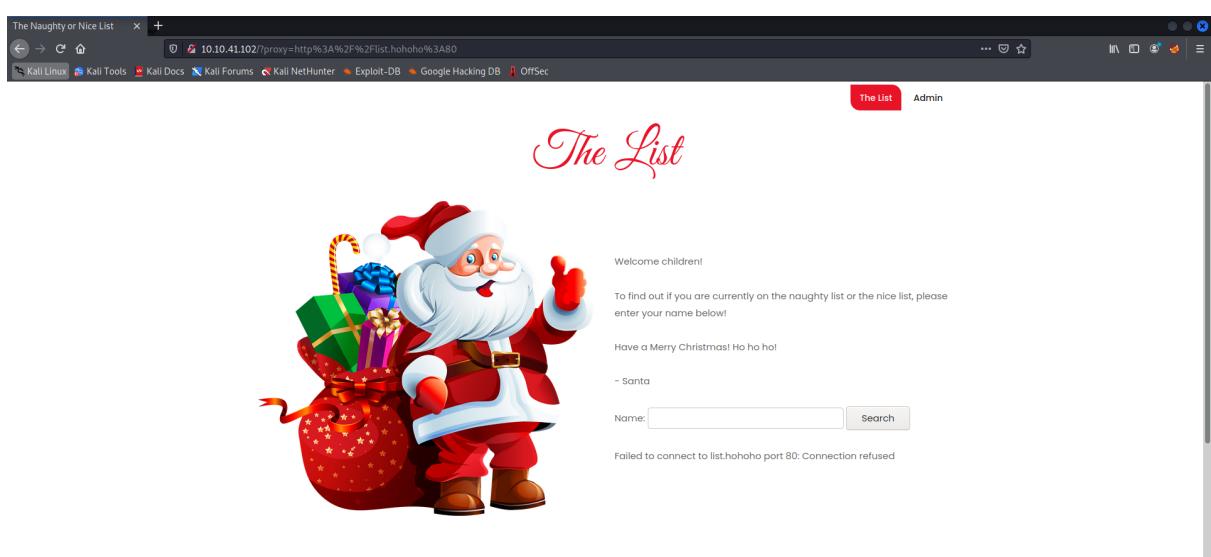
Question 2

Copy and paste “/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F” on the browser.



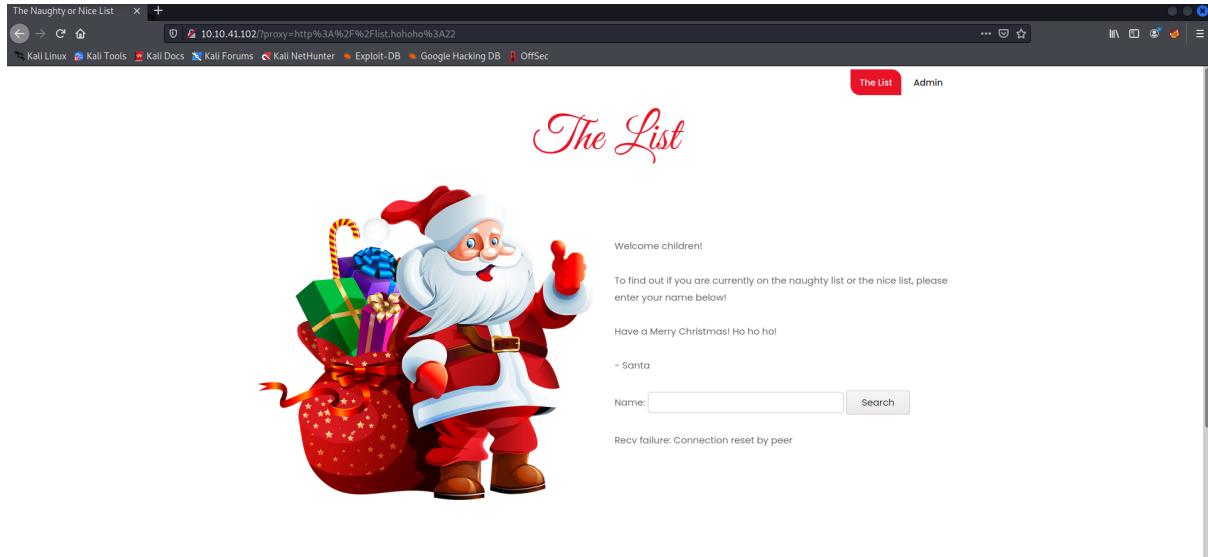
Question 3

Paste "?proxy=http%3A%2F%2Flist.hohoho%3A80"? on the browser.



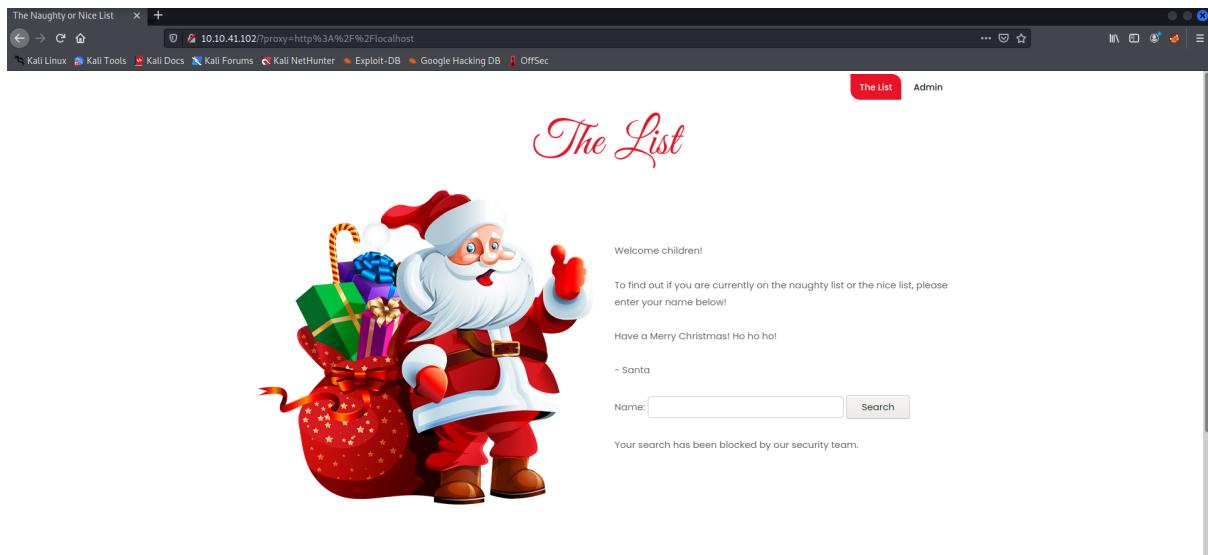
Question 4

Paste "/?proxy=http%3A%2F%2Flist.hohoho%3A22" on the browser.



Question 5

Paste "/?proxy=http%3A%2F%2Flocalhost" on the browser.



Question 6

Set the hostname in the URL to "list.hohoho.localtest.me". Look for the password.

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

Santa,
If you need to make any changes to the Naughty or Nice list, you need to login.
I know you have trouble remembering your password so here it is: Be good for goodness sake!

- Elf McSkidy

Question 7

Guess the username and enter the password.

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

Delete the naughty list.

List Administration

This page is currently under construction.

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

Thought process/Methodology:

To begin with, we entered the Ip address given and directed to the website. When we search a name in the search bar, the page is loaded and told whether the name is on the Naughty list or Nice list. We noticed the URL had changed and we tried to fetch the root, but it gave us a 404 message. We tried to change the port to 80 but the connection was refused. When we switched the port to 22, we got this message; "Recv failure: Connection reset by peer". We tried to run the server locally, but it got blocked by the security team. Although it was unfortunate, we noticed that we can take advantage of DNS subdomains and create our own domain. So, we used the localtest.me and it brought us to the password. After guessing the username and entering the password at the admin login page, it brought us to the list administration. There, we can delete the Naughty list and get the flag.

Day 20 : Blue Teaming - PowershELF to the rescue

Tools used: Kali Linux, Firefox, SSH

Solution/walkthrough:

Question 1

Search for -l in the SSH manual

```
By default, the local port is bound in accordance with the GatewayPorts setting. However, an explicit bind_address may be used to bind the connection to a specific address. The bind_address of "localhost" indicates that the listening port be bound for local use only, while an empty address or "*" indicates that the port should be available from all interfaces.

-l login_name
    Specifies the user to log in as on the remote machine. This also may be specified on a per-host basis in the configuration file.

-M
    Places the ssh client into "master" mode for connection sharing. Multiple -M options places ssh into "master" mode but with confirmation required using ssh-askpass(1) before each operation that changes the multiplexing state (e.g. opening a
```

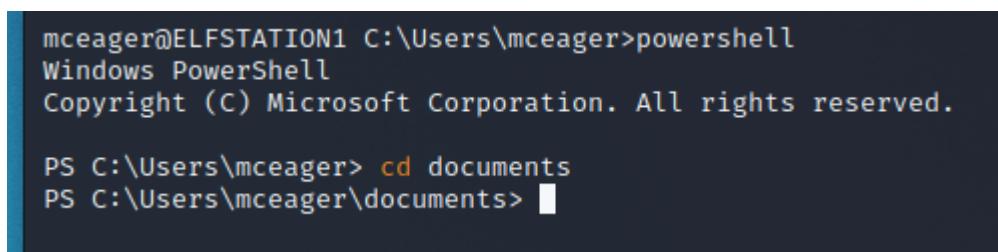
Question 2

Enter the command and password given to login.



A screenshot of a Windows Command Prompt window titled 'c:\windows\system32\cmd.exe'. The window shows a successful login from a Kali Linux host to a Windows 10 target. The command entered was 'ssh -l mceager 1211103149@kali'. The output shows the user 'mceager' at 'ELFSTATION1' with a Windows 10 desktop background.

Startup powershell and got to documents



A screenshot of a Windows PowerShell window titled 'c:\windows\system32\cmd.exe'. The user has started a PowerShell session and navigated to the 'documents' folder in their user directory. The command 'cd documents' was run, and the current directory is shown as 'C:\Users\mceager\documents'.

Using some commands we can get to the hidden files. cat scan the contents to get the answer

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

Question 3

Go into desktop to find the elf2wo file

```
PS C:\Users\mceager\documents> cd ..
PS C:\Users\mceager> cd desktop
PS C:\Users\mceager\desktop> ls -hidden

Directory: C:\Users\mceager\desktop

Mode                LastWriteTime         Length Name
—
d-- h--          12/7/2020 11:26 AM            0 elf2wo
-a-hs-          12/7/2020 10:29 AM            282 desktop.ini

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

Cat the file to get the answer

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

Go into system 32 in windows

```
PS C:\Users\mceager> cd C:/Windows  
PS C:\Windows> cd system32  
PS C:\Windows\system32> █
```

Using filter to easily find the hidden file

```
PS C:\Windows\system32> get-childitem -hidden -directory -filter "*3*"  
  
Directory: C:\Windows\system32  
  
Mode LastWriteTime Length Name  
-- -- -- --  
d--h-- 11/23/2020 3:26 PM 3lfthr3e
```

Question 5

change the directory to the file and get the contents with -hidden

```
PS C:\Windows\system32> cd 3lfthr3e  
PS C:\Windows\system32\3lfthr3e> ls -hidden  
  
Directory: C:\Windows\system32\3lfthr3e  
  
Mode LastWriteTime Length Name  
-- -- -- --  
-arh-- 11/17/2020 10:58 AM 85887 1.txt  
-arh-- 11/23/2020 3:26 PM 12061168 2.txt
```

Use the command from THM to get the number of words in the first file

```
PS C:\Windows\system32\3lfthr3e> cat 1.txt | Measure-object -Word  
Lines Words Characters Property  
-- -- -- --  
9999  
  
PS C:\Windows\system32\3lfthr3e> █
```

Question 6

Use command from THM to get the exact position we want

```
PS C:\Windows\system32\3lfthr3e> (cat 1.txt)[551]
Red
PS C:\Windows\system32\3lfthr3e> (cat 1.txt)[6991]
Ryder
PS C:\Windows\system32\3lfthr3e> █
```

Question 7

Use the command from THM to search for string only with redryder in them to get the answer

```
PS C:\Windows\system32\3lfthr3e> cat 2.txt | select-string -Pattern "redryder"
redryderbbgun

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

Thought Process/Methodology:

Using the command given in THM, we were able to get into the windows machine. Using commands we can search for items in a file to get the answer and using filters we can narrow down the search to hidden items, the number of the index and a specific string