

PSP0201

Week 5

Writeup

Group Name: CyberQuest

Members

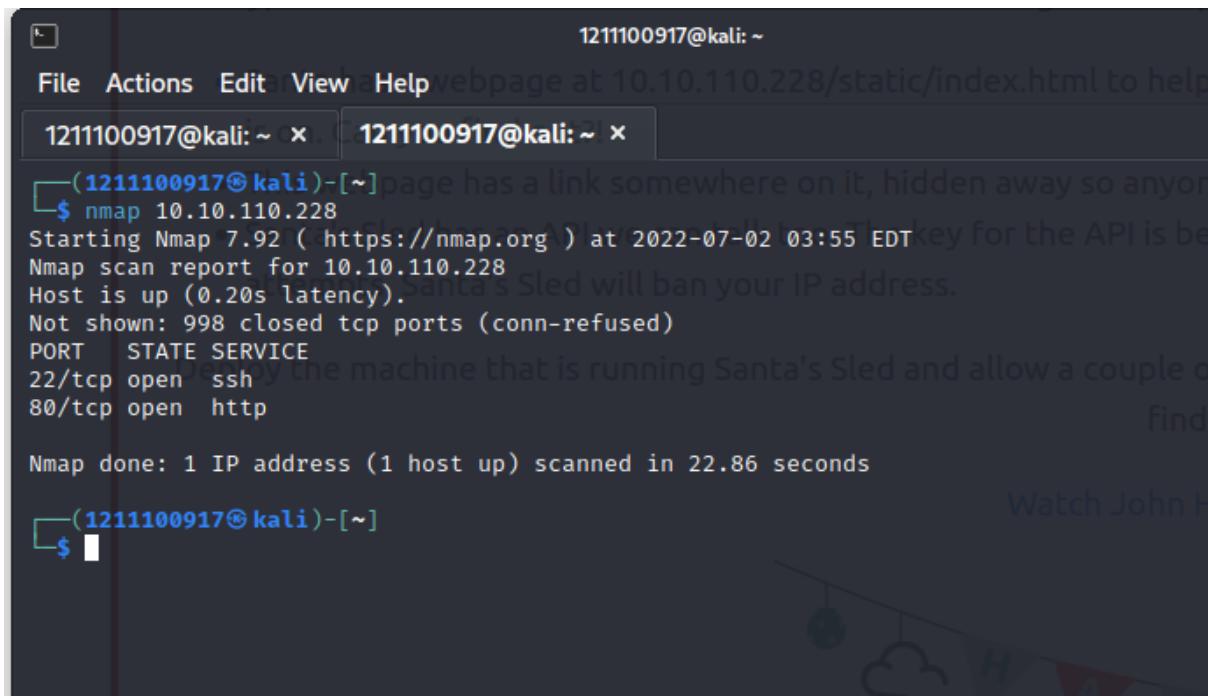
ID	Name	Role
1211102409	CHUA KAI ZHENG	Leader
1211102696	LEE JIA MENG	Member
1211100917	NATALIE TAN LI YI	Member

Day 16 : Scripting - Help! Where is Santa?

Tools : Sublime Text, Kali Linux, FireFox, Terminal

Question 1

Find port number for the web server using nmap

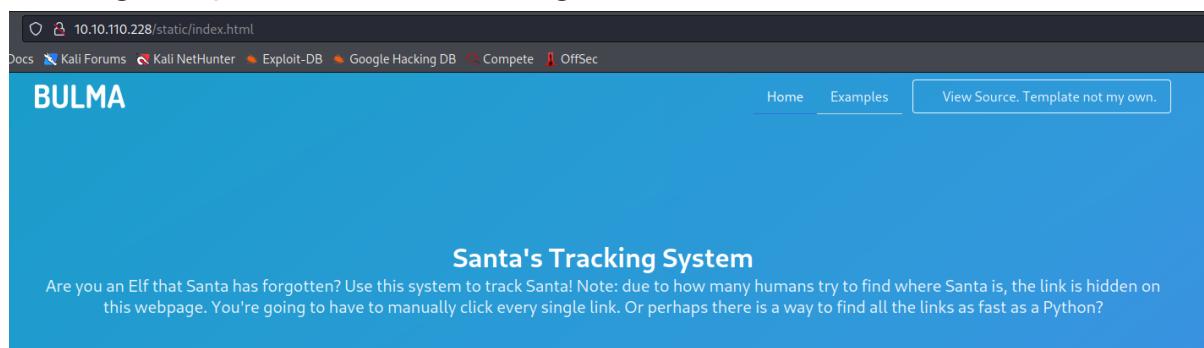


```
1211100917@kali: ~
File Actions Edit View Help webpage at 10.10.110.228/static/index.html to help
1211100917@kali: ~ x 1211100917@kali: ~ x
(1211100917㉿kali)-[~]page has a link somewhere on it, hidden away so anyone can find it.
$ nmap 10.10.110.228
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-02 03:55 EDT
Nmap scan report for 10.10.110.228
Host is up (0.20s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 22.86 seconds
(1211100917㉿kali)-[~]
$
```

Question 2

Getting templates that are being used from website



The screenshot shows a Firefox browser window with the URL `10.10.110.228/static/index.html` in the address bar. The page title is "BULMA". At the top, there are navigation links: "Home", "Examples", and a button labeled "View Source. Template not my own.". The main content area features a heading "Santa's Tracking System" and a paragraph of text: "Are you an Elf that Santa has forgotten? Use this system to track Santa! Note: due to how many humans try to find where Santa is, the link is hidden on this webpage. You're going to have to manually click every single link. Or perhaps there is a way to find all the links as fast as a Python?".

Question 3

Finding api of website using “/api”

A screenshot of a web browser window. The address bar shows the URL `10.10.110.228/api/`. The page content is a JSON response with the following structure:

```
detail: "Not Found"
```

Question 4

Find raw data returned if no parameters are entered

A screenshot of a web browser window. The address bar shows the URL `10.10.110.228/api/`. The page content is a JSON response with the following structure:

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

Question 5

Use sublime text to create 'script.py' to find Santa's position

A screenshot of a terminal window. The command `$ python3 script.py` is run, followed by a series of failed API requests. The output shows multiple iterations of the same error message:

```
api_key1
{"item_id":1,"q":"Error. Key not valid!"}
api_key3
{"item_id":3,"q":"Error. Key not valid!"}
api_key5
{"item_id":5,"q":"Error. Key not valid!"}
api_key7
{"item_id":7,"q":"Error. Key not valid!"}
api_key9
{"item_id":9,"q":"Error. Key not valid!"}
api_key11
{"item_id":11,"q":"Error. Key not valid!"}
api_key13
{"item_id":13,"q":"Error. Key not valid!"}
api_key15
```

```
i item_id :53, q : Error. Key not valid! }
api_key55
{"item_id":55,"q":"Error. Key not valid!"}
api_key57
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
api_key59
{"item_id":59,"q":"Error. Key not valid!"}
api_key61
{"item_id":61,"q":"Error. Key not valid!"}
api_key63
{"item_id":63,"q":"Error. Key not valid!"}
```

Thought Process/Methodology

We first copy the ip address and paste it at another tab. We then use nmap to find out the port number of the web server (80). The templates being used can be found at the left upper corner named as BULMA. Then, we try “/api/” behind the ip address to find out the directory for the API. We click at the Raw Data section to get answer for return when no parameters are entered Next, we download Sublime Text using the terminal and paste the example code from Day 15. Then, we edit some parts of the code to let it loop the api in order to find Santa's placing and API key, which we save it as ‘script.py’. We then use ‘script.py’ to find Santa's location.

Day 17: Reverse Engineering: ReverseELFneering

Tools used: Kali, Firefox, Terminal

Question 1

Get the answer from THM table

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

Use aa

```
[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), we could
[0x00400a30]> █
```

Question 3

Use db

```
[0x00400a30]> db 0x00400b55
[0x00400a30]> pdf @main
      3rd column specifies the name that r2 uses to reference them and the 4th column shows the actual
      ;-- main:
/ (Fcn) sym.main 68
  sym.main ();
  to allocate ; var int local_ch @ rbp-0xc
  ; there's enough room for variables to be allocated and more). We'll start looking at
  fiction (mov) ; var int local_8h @ rbp-0x8
  ; var int local_4h @ rbp-0x4
  ; DATA XREF from 0x00400a4d (entry0)
  0x00400b4d      55          push rbp
  the program is now executing. This is useful for us to look at the state of the program at that particular point. So let's
  hand db 0x00400b51 4883ec10 sub rsp, 0x10
  ; the breakpoint is set, we run the pdf @main command again
  struction we 0x00400b55 b c745f4040000. mov dword [local_ch], 4
  0x00400b5c at. c745f8050000. mov dword [local_8h], 5
  0x00400b63 8b55f4        mov edx, dword [local_ch]
  0x00400b66 8b45f8        mov eax, dword [local_8h]
  0x00400b69 01d0          add eax, edx
  0x00400b6b 8945fc        mov dword [local_4h], eax
  0x00400b6e 8b4dfc        mov ecx, dword [local_4h]
  0x00400b71 8b55f8        mov edx, dword [local_8h]
  0x00400b74 8b45f4        mov eax, dword [local_ch]
  0x00400b77 89c6          mov esi, eax
  0x00400b79 488d3d881409. lea rdi, qword str.the_value_of_a_is_%d_the_value_of_b_is_%d_and_the_val
ue_of_c_is_%d ; 0x492008 ; "the value of a is %d, the value of b is %d and the value of c is %d"
  0x00400b80 b800000000    mov eax, 0
  0x00400b85 e8f6ea0000    call sym.__printf
  0x00400b8a b800000000    mov eax, 0
  0x00400b8f c9             leave
  0x00400b90 c3             ret
```

Question 4

Use dc

```
[0x00400a30]> dc  
hit breakpoint at: 400b55  
[0x00400b55]> █
```

Question 5

Use pdf@main to assembly code

```
elfmceager@tbfc-day-17: ~  
File Actions Edit View Help  
1211102409@kali: ~> r2 -d ./challenge1  
Do you want to quit? (Y/n) nd  
Do you want to kill the process? (Y/n)  
elfmceager@tbfc-day-17: ~$ r2 -d ./challenge1  
Process with PID 1586 started ...  
= attach 1586 1586  
bin.baddr 0x00400000  
Using 0x400000  
Warning: Cannot initialize dynamic strings  
asm.bits 64  
[0x00400a30]> aa  
[ W A R N I N G : 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]> pdf @main  
;-- main:  
/ (fcn) sym.main 35  
| insns=35 nops=0 total=114 bytes  
| 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 Submit  
|? 0x00400b5f    b845f4      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  
Correct Answer  
Submit
```

Question 6

Set the breakpoint at the 0x00400b62 and execute it. Find the value with px @rbp-0x8.

```
[0x00400a30]> db 0x00400b62
[0x00400a30]> dc
hit breakpoint at: 400b62
[0x00400b62]> px @rbp-0x8
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0x7ffd4c245ee8 0600 0000 0000 0000 4018 4000 0000 0000 . . . . . . . .
0x7ffd4c245ef8 e910 4000 0000 0000 0000 0000 0000 0000 .. . . . .
0x7ffd4c245f08 0000 0000 0100 0000 1860 244c fd7f 0000 . . . . . ` $L .
0x7ffd4c245f18 4d0b 4000 0000 0000 0000 0000 0000 0000 M. . . .
0x7ffd4c245f28 1700 0000 0100 0000 0000 0000 0000 0000 . . . . .
0x7ffd4c245f38 0000 0000 0200 0000 0000 0000 0000 0000 . . . . .
0x7ffd4c245f48 0000 0000 0000 0000 0000 0000 0000 0000 . . . . .
0x7ffd4c245f58 0000 0000 0000 0000 0004 4000 0000 0000 . . . . .
0x7ffd4c245f68 7e5a ec32 9b51 e61c e018 4000 0000 0000 ~Z.2.Q. . . .
0x7ffd4c245f78 0000 0000 0000 0000 1890 6b00 0000 0000 . . . . k .
0x7ffd4c245f88 0000 0000 0000 0000 7e5a 6cbc 53c9 1ce3 . . . . ~Z1.S ...
0x7ffd4c245f98 7e5a 5823 9b51 e61c 0000 0000 0000 0000 ~ZX#.Q. . .
0x7ffd4c245fa8 0000 0000 0000 0000 0000 0000 0000 0000 . . . . .
0x7ffd4c245fb8 0000 0000 0000 0000 0000 0000 0000 0000 . . . . .
0x7ffd4c245fc8 0000 0000 0000 0000 0000 0000 0000 0000 . . . . .
0x7ffd4c245fd8 0000 0000 0000 0000 0000 0000 0000 0000 . . . . 
```

Question 7

Set the breakpoint at the 0x00400b69 and execute it. Find the value with px @rbp-0x4.

```
[0x00400b69]> px @rbp-0x4
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0x7ffc0fa2e97c 0600 0000 4018 4000 0000 0000 e910 4000 . . . . . . . .
0x7ffc0fa2e98c 0000 0000 0000 0000 0000 0000 0000 0000 . . . . .
0x7ffc0fa2e99c 0100 0000 a8ea a20f fc7f 0000 4d0b 4000 . . . . M. . 
```

Thought Process/Methodology

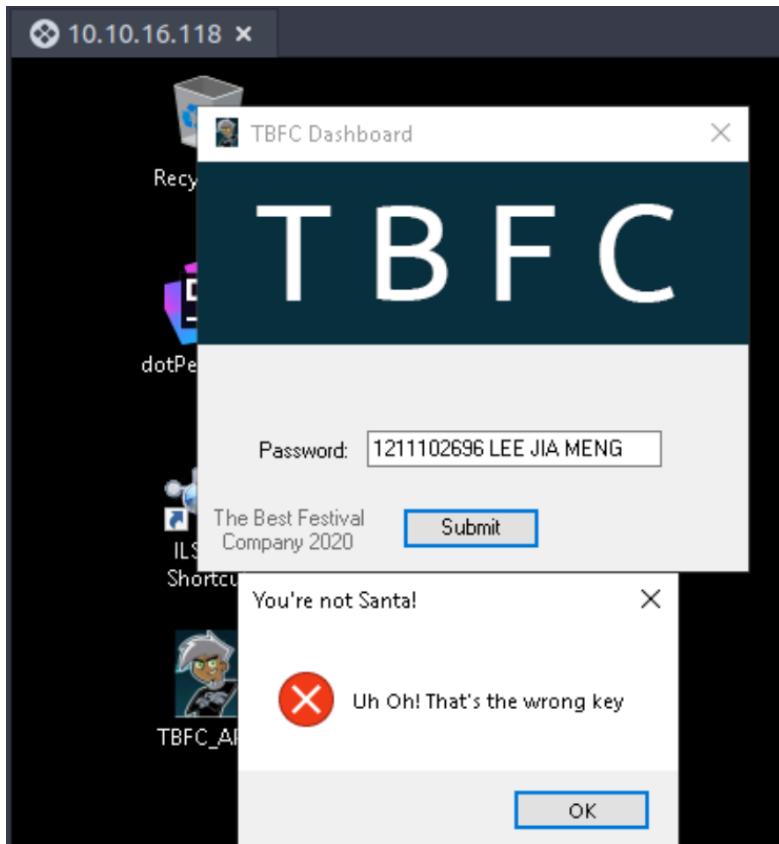
First log in using ssh username(elfmceager)@MACHINE_IP with the password(adventofcyber) that is provided in THM.Run through Radare 2 in debugging mode with the command r2 -d ./challenge1 .Ask r2 to analyze the program with ‘aa’.Use ‘afl’ to find the list of the functions. After that, we assembly code at main by running the command pdf @main.We can find the answer for Q5 by looking at the output. Then, we set the breakpoint at the imul eax to find the value with px @rbp-0x8.Last, we exit and re-enter debugging mode. This time we set the breakpoint at 0x00400b69 and execute it to find the value of local_4h before eax set to 0 with px @rbp-0x4.

Day 18: [Reverse Engineering] The Bits of Christmas

Tools used: THM AttackBox, Remmina, CyberChef.

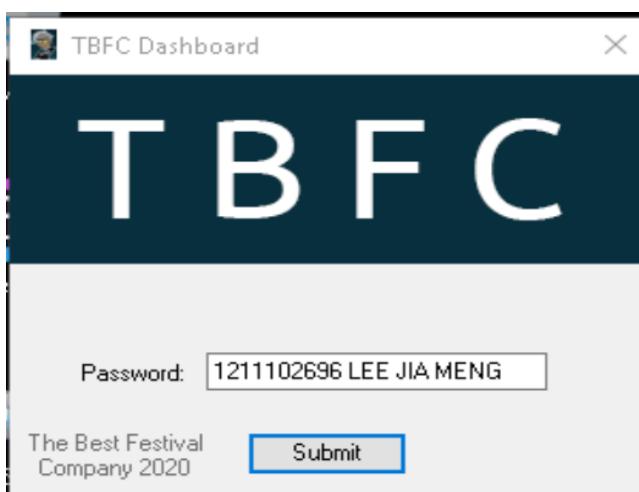
Question 1

Open up TBFC_app after getting into Remmina and entering anything into the password box.



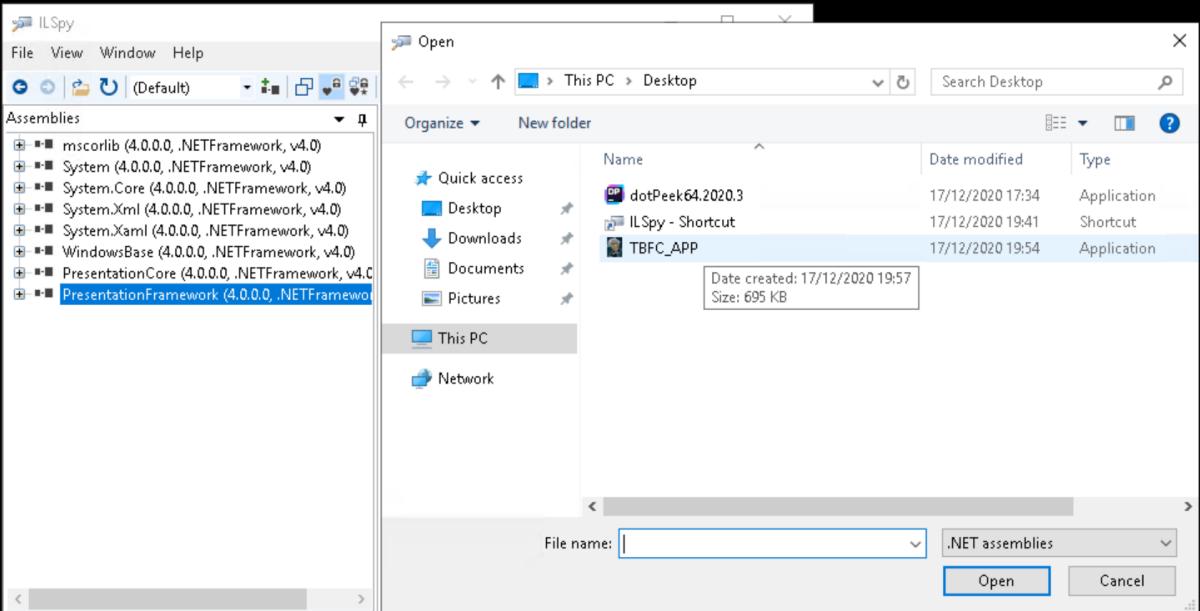
Question 2

TBFC stands for



Question 3

Module that catches my attention is

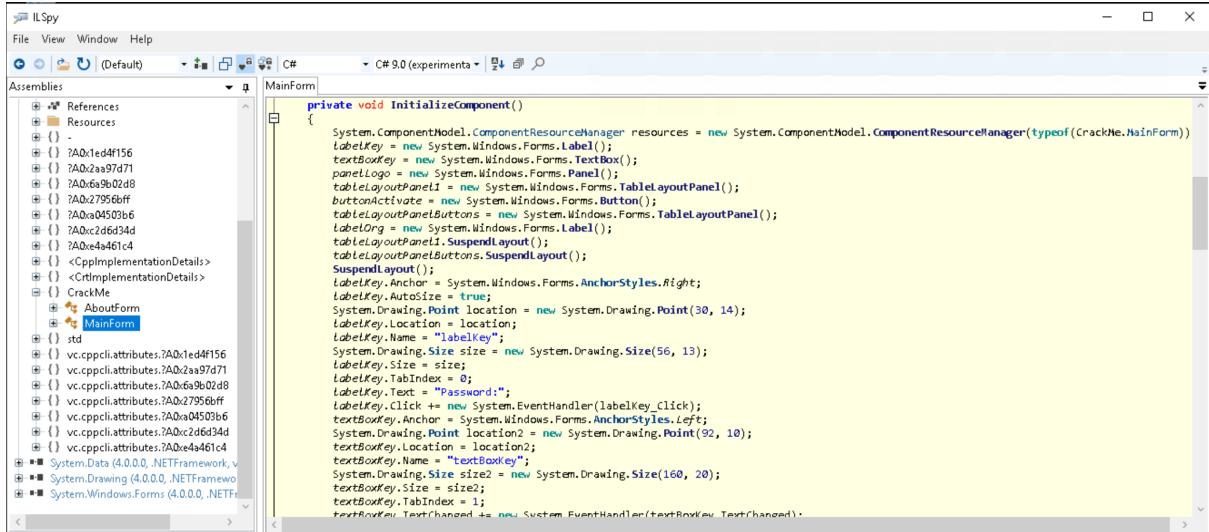


The screenshot shows the ILSpy interface. In the top-left pane, under 'Assemblies', there is a list of standard .NET Framework assemblies such as msclr (4.0.0.0, .NETFramework, v4.0), System (4.0.0.0, .NETFramework, v4.0), System.Core (4.0.0.0, .NETFramework, v4.0), System.Xml (4.0.0.0, .NETFramework, v4.0), System.Xaml (4.0.0.0, .NETFramework, v4.0), WindowsBase (4.0.0.0, .NETFramework, v4.0), PresentationCore (4.0.0.0, .NETFramework, v4.0), and PresentationFramework (4.0.0.0, .NETFramework, v4.0). In the bottom-left pane, the assembly tree for 'TBFC_APP (0.0.0.0, .NETFramework, v4.6.1)' is displayed. The tree includes nodes for Metadata, References, Resources, and several memory addresses starting with ?A0x. One node, 'CrackMe', is highlighted with a yellow background. The bottom-right pane shows a file browser window titled 'Open' with the path 'This PC > Desktop'. It lists files: dotPeek64.2020.3 (Application, 17/12/2020 17:34), ILSPY - Shortcut (Shortcut, 17/12/2020 19:41), and TBFC_APP (Application, 17/12/2020 19:54). The 'TBFC_APP' file is selected.

```
graph TD; TBFC_APP[TBFC_APP (0.0.0.0, .NETFramework, v4.6.1)] --> Metadata[Metadata]; TBFC_APP --> References[References]; TBFC_APP --> Resources[Resources]; TBFC_APP --> A1["?A0x1ed4f156"]; TBFC_APP --> A2["?A0x2aa97d71"]; TBFC_APP --> A3["?A0x6a9b02d8"]; TBFC_APP --> A4["?A0x27956bff"]; TBFC_APP --> A5["?A0xa04503b6"]; TBFC_APP --> A6["?A0xc2d6d34d"]; TBFC_APP --> A7["?A0xe4a461c4"]; TBFC_APP --> CppImplementationDetails[<CppImplementationDetails>]; TBFC_APP --> CrtImplementationDetails[<CrtImplementationDetails>]; TBFC_APP --> CrackMe[CrackMe]; TBFC_APP --> std[std]; TBFC_APP --> vc_cppcli_attributes_1["vc.cppcli.attributes.?A0x1ed4f156"]; TBFC_APP --> vc_cppcli_attributes_2["vc.cppcli.attributes.?A0x2aa97d71"]; TBFC_APP --> vc_cppcli_attributes_3["vc.cppcli.attributes.?A0x6a9b02d8"]; TBFC_APP --> vc_cppcli_attributes_4["vc.cppcli.attributes.?A0x27956bff"]; TBFC_APP --> vc_cppcli_attributes_5["vc.cppcli.attributes.?A0xa04503b6"]; TBFC_APP --> vc_cppcli_attributes_6["vc.cppcli.attributes.?A0xc2d6d34d"]; TBFC_APP --> vc_cppcli_attributes_7["vc.cppcli.attributes.?A0xe4a461c4"];
```

Question 4

The form that contain the information are looking for is

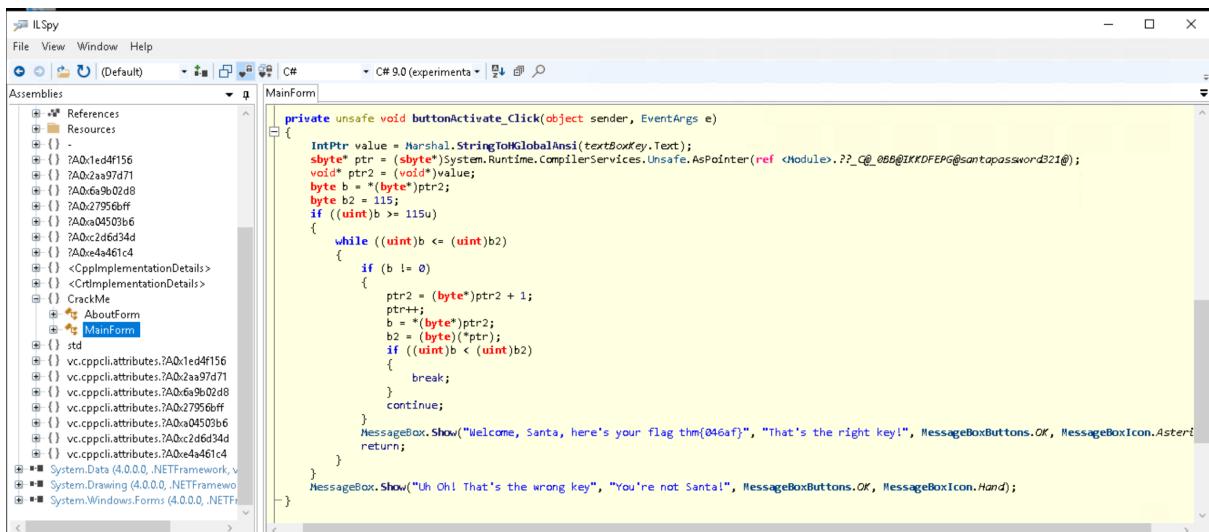


```

ILSpy
File View Window Help
C# C# 9.0 (experimental) 
Assemblies MainForm
private void InitializeComponent()
{
    System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(CrackMe.MainForm));
    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();
    labelOrg = new System.Windows.Forms.Label();
    tableLayoutPanelPanel1.SuspendLayout();
    tableLayoutPanelButtons.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";
    labelKey.Size = new System.Drawing.Size(56, 13);
    labelKey.TabIndex = 0;
    labelKey.Text = "Password:";
    labelKey.Click += new System.EventHandler(labelKey_Click);
    textBoxKey.Anchor = System.Windows.Forms.AnchorStyles.Left;
    System.Drawing.Point location2 = new System.Drawing.Point(92, 10);
    textBoxKey.Location = location2;
    textBoxKey.Name = "textBoxKey";
    System.Drawing.Size size2 = new System.Drawing.Size(160, 20);
    textBoxKey.Size = size2;
    textBoxKey.TabIndex = 1;
    textBoxKey.TextChanged += new System.EventHandler(textBoxKey_TextChanged);
}

```

Question 5



```

ILSpy
File View Window Help
C# C# 9.0 (experimental)
Assemblies MainForm
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@_0BB@IKKDFEPG@santapassword321@);
    void* ptr2 = *(void**)ptr;
    byte b = *(byte*)ptr2;
    byte b2 = *(byte*)(ptr);
    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);
    }
}

```

Question 6

Expand the buttonActivate_Click and look through for password.



```

MainForm
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@_0BB@IKKDFEPG@santapassword321@);
    void* ptr2 = *(void**)ptr;
    byte b = *(byte*)ptr2;
    byte b2 = *(byte*)(ptr);
    if ((uint)b >= 115u)
    {
        ??_C@_0BB@IKKDFEPG@santapassword321@:$ArrayType$$BY0BB@$$CBD global:<Module>._C@_0BB@IKKDFEPG@santapassword321@
        // <Module>
        using ...
        internal static $ArrayType$$BY0BB@$$CBD ??_C@_0BB@IKKDFEPG@santapassword321@/* Not supported: data(73 61 6E 74 61 70 61 73 73 77 6F 72 64 33 32 31 00)
    }
}

```

The screenshot shows a hex editor interface. At the top, there is a status bar with the text "internal static \$ArrayType\$\$\$BY0BB@\$\$CBD ??_C@_0BB@IKKDFEPG@santapassword321/* Not supported: data(73 61 6E 74 61 70 61 73 73 77 6F 72 64 33 32 31 00) Copy". Below the status bar, there are three icons: a save icon, a folder icon, and a trash bin icon.

Recipe		Input
From Hex	✖️	73 61 6E 74 61 70 61 73 73 77 6F 72 64 33 32 31
Delimiter		
Auto		

At the bottom, there is an "Output" section containing the ASCII string "santapassword321".

Question 7

Use the password that was obtained using CyberChef to login.



Thought Process/Methodology

Start the AttackBox and open up Remmina. Connect to it using the MACHINE_IP, username , and password provided in THM. Open the TBFC_app and randomly type in the password to obtain the message. From the TBFC dashboard, it shows that it stands for The Best Festival Company. Moving forward, close the TBFC_app and open up ILSPY. Now, open up TBFC_app in ILSPY and look through the TBFC_app by expanding it. The module that catches my eyes is the ‘CrackMe’ module that was standing out. From then on, expand the ‘CrackMe’ module to explore the two forms under it. The form that will give us information is the MainForm as it lets the user interact with. Look through the method within the MainForm for Santa's password. By expanding the buttonActivate_Click method, a string that seems to be a clue to Santa’s password would appear. Double click on it and ILSPY will take us to a dedicated page of it. In the new page, there is a set of data that seems to be Santa’s password, copy the data and convert the hexadecimal by using CyberChef. Santa’s password will be obtained and used to login to the TBFC_app. Lastly, we will obtain the flag when we key in the correct password.

Day 19: Web Exploitation: The Naughty or Nice List

Tools: Kali Linux

Question 1

Finding which list are the person on

The List



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:

Timothy is on the Naughty List.

Question 2

Finding what is displayed on the page when you use
"/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F"

The List



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:

Not Found

[The requested URL was not found on this server.](#)

Question 3

Finding what is displayed on the page when you use
"/?proxy=http%3A%2F%2Flist.hohoho%3A80"

The List



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:

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

Question 4

Finding what is displayed on the page when you use
"/?proxy=http%3A%2F%2Flist.hohoho%3A22"

The List



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:

Recv failure: Connection reset by peer

Question 5

Finding what is displayed on the page when you use
"/?proxy=http%3A%2F%2Flocalhost"

The List



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:

Your search has been blocked by our security team.

Question 6

Finding Santa's password

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

The List Admin

The List



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:

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

Getting the challenge flag



Thought Process/ Methodology

Paste the IP address to a new page, then start by searching the names from the google form to see which list are they in. Then, paste all the links from google form (for Q2- Q5) in order to find what does the page display. Next, we add

"list.hohoho.localtest.me" to the end of link to get the password for the admin. After getting the password, we log in using Santa as username and the password we got. Then we delete the name list to get the flag.

Day20:Blue Teaming: Powershell to the rescue

Tools used: Kali, Firefox, Terminal, PowerShell

Question 1

Use ssh -h to check the parameter

```
(1211102409㉿kali)-[~]
$ ssh -h remote machine: ssh -l mceager 10.10.61.59
unknown option -- h
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command [argument ...]]
```

Question 2

Finding the hidden file with Get-ChildItem -Hidden. Get the content using Get-Content -Path e1fone.txt

```
PS C:\Users\mceager\Documents> Get-ChildItem -Hidden  
Directory: C:\Users\mceager\Documents  
Information about a specific cmdlet. For example, Get-Help Select-  
  
Mode          LastWriteTime      Length Name  
--hsl        12/7/2020 10:28 AM           My Music  
--hsl        12/7/2020 10:28 AM           My Pictures  
--hsl        12/7/2020 10:28 AM           My Videos  
-arh--       11/18/2020 5:05 PM            35 e1fone.txt  
  
PS C:\Users\mceager\Documents> Get-Content -Path e1fone.txt  
All I want is my '2 front teeth'!!!
```

Question 3

Find the hidden file in C:\Users\mceager\Desktop\elf2wo and read the content with 'Get-Content -Path e70smsW10Y4k.txt'

```
PS C:\Users\mceager\Desktop> Set-Location elf2wo  
PS C:\Users\mceager\Desktop\elf2wo> ls  
  
Directory: C:\Users\mceager\Desktop\elf2wo  
Information about a specific cmdlet. For example, Get-Help Select-  
  
Mode          LastWriteTime      Length Name  
-a---       11/17/2020 10:26 AM            64 e70smsW10Y4k.txt  
  
PS C:\Users\mceager\Desktop\elf2wo> Get-Content .\e70smsW10Y4k.txt  
I want the movie Scrooged <3!
```

Question 4

Go to the Windows directory using 'cd C:\Windows'. Use 'Get-ChildItem -Directory -Hidden -Recurse -Filter "*3*" -ErrorAction SilentlyContinue' to find the hidden folder that contains files for Elf 3

```
PS C:\Windows> Get-ChildItem -Directory -Hidden -Recurse -Filter '*3*' -ErrorAction SilentlyContinue
[+]index 551 and 6991 in the first file?
    Directory: C:\Windows\System32
    < Submit

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

Question 5

PS C:\Windows\System32\3lfthr3e> Get-ChildItem -hidden

[+]What is the hidden folder? (This command will call the hidden folder)

```
    Directory: C:\Windows\System32\3lfthr3e
    < Submit

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

[+]in the first file?

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

Lines	Words	Characters	Property
9999			

[+]the 2nd file for the phrase from the previous question to get the full answer. What does Elf 3 want? (use s

Use 'Get-Content -Path 1.txt | Measure-object -Word' to get the number for the words.

Question 6

Find the 2 words using '(Get-Content -Path 1.txt)[551/6991]'

```
PS C:\Windows\System32\3lfthr3e> (Get-Content -Path 1.txt)[551]
aces W
Red
PS C:\Windows\System32\3lfthr3e> (Get-Content -Path 1.txt)[6991]
Ryder
```

Question 7

Search using 'Select-String -Path
'C:\Windows\System32\3lfthr3e\2.txt' -Pattern 'redryder'

```
PS C:\Windows\System32\3lfthr3e> Select-String -Path 'C:\Windows\System32\3lfthr3e\2.txt' -Pattern 'redryder'
2.txt:558704:redryderbbgun
```

Thought Process/Methodology

First, we log in using ‘ssh -l mceager MACHINE_IP’ with the password ‘r0ckStar!’. After that, we launch the PowerShell and navigate to the Documents folder with ‘Set-Location .\Documents\’. ‘Get-ChildItem -Hidden’ had been used to find the hidden elf file and ‘Get-Content -path e1fone.txt’ to read the content of e1fone.txt. Then, we got back to \mceager by using ‘cd ..’ and navigating to the Desktop file with ‘Set-Location .\Desktop\’. The e70smsW10Y4k.txt has been found and read in the same way as the e1fone.txt. To find the name of the hidden folder for Elf 3, we need to go back to the Windows directory using ‘cd C:\Windows\’ and find the file with ‘Get-ChildItem -Directory -Hidden -Recurse -Filter “*3*” -ErrorAction SilentlyContinue’. Next, we use ‘Get-ChildItem -hidden’ to get the hidden file in the folder and use ‘Get-Content -Path 1.txt | Measure-object -Word’ to get the number of words contained in the first file. We also find the two words at index 551 and 6991 with ‘(Get-Content -Path 1.txt)[551/6991]’. Lastly, we use ‘Select-String -Path ‘C:\Windows\System32\3lfthr3e\2.txt’ -Pattern ‘redryder’ ’ to search the phrase “redryder” in the 2nd file.