PSP0201
WEEKLY WRITE UP
WEEK 5

By: Amilia Nadzeera Binti Baharudin , 1211102162

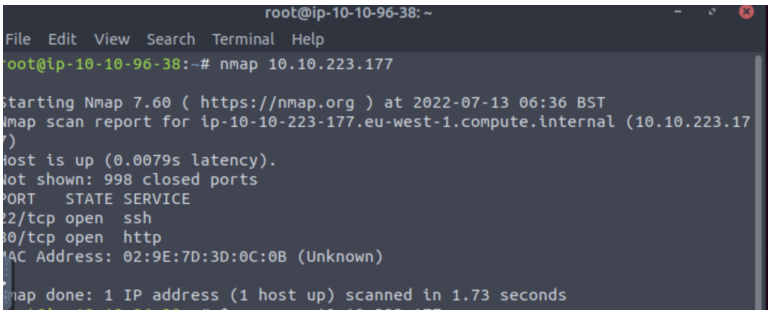
Day 16: Scripting ; Help! Where is Santa?

Tools used : THM AttackBox , FireFox , Python

Question 1: What is the port number for the web server?

The port number can be found using nmap and syntax , host number is shown.

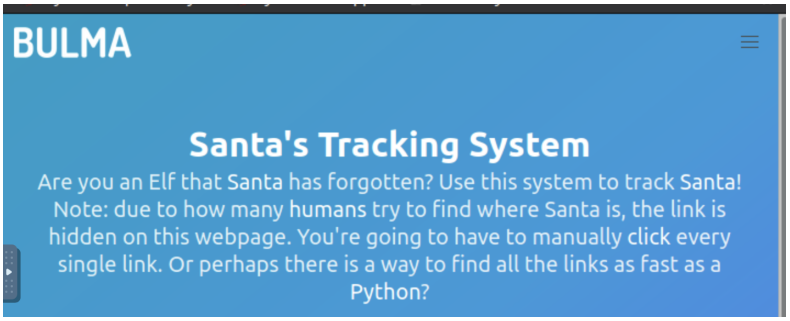
Answer 1: 80



Question 2: What templates are being used?

For this , we checked the top left of the website

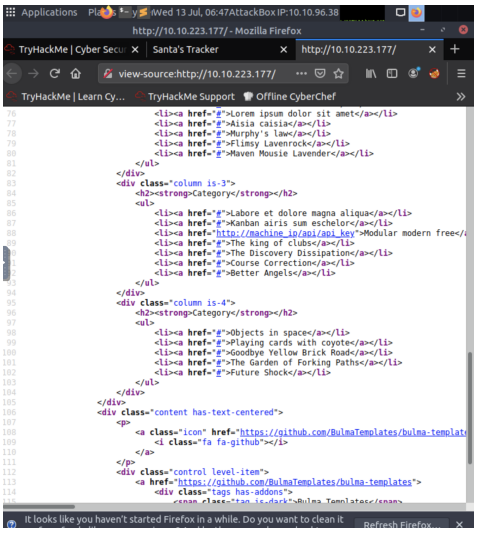
Answer 2: BULMA



Question 3: Without using enumeration tools such as Dirbuster, what is the directory for the API?

Open the developers tools in your browser , then open your elements tab. Scroll down and check the footer to find the HTML tag

Answer 3: /api/



Question 4: Go to the API endpoint. What is the Raw Data returned if no parameters are entered?

the raw data returned was retrieved from the endpoint of the API.

Answer 4: {"detail":[{"loc":["path","item_id"],"msg":"value is not a valid integer","type":"type_error.integer"}]}

Question 5: Where is Santa right now?

Answer 5: Winter Wonderland , Hyde Park , London

Question 6: Find out the correct API key. Remember, this is an odd number between 0-100.

Answer 6: 57

For both questions , we had to try our luck and key in an odd number between 0-100 to know where Santa is right now. 57 is

an odd number and is fortunately the answer for Question 6 and they key to finding out where Santa is

Thought process / Methodology :
Firstly, we're using nmap in the terminal to find the open ports using syntax
nmap <ip address>. Next, we opened the developer tools in th web browser followed by tab elements.We look for the hint through the source code; http://machine_ip/api/api_key.
Then, we're using python python3 <script name> to look out for all the HTML links in the website. From there, we modified the script and continued to run it. Then using the library request, we have created a script and saved it as "apibrute.py" and run python3 apibrute.py there. We found Santa's location and the API key.

Day 17 : Reverse Engineering ; ReverseELFngineering

Tools used : THM AttackBox , FireFox

Question 1: Match the data type with the size in bytes

For this question , we followed the tale provided in TryHackMe.

Answer 1:

- Byte - 1
- Word - 2
- Double Word - 4
- Quad - 8
- Single Precision - 4
- Double Precision - 8

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: What is the command to analyse the program in radare2?

The answer was provided in the text shown below.
Answer 2: pdf@main

As seen here, there actually is a function at main. Let's examine the assembly code at main by running the command `pdf @main` Where **pdf** means print disassembly function. Doing so will give us the following view:

Question 3: What is the command to set a breakpoint in radare2?

The answer was provided in the text shown below.

Answer 3: db

The line starting with **sym.main** indicates we're looking at the **main function**. The next 3 lines are used to represent the variables stored in the function. The second column indicates that they are integers(*int*), the 3rd column specifies the name that `r2` uses to reference them and the 4th column shows the actual memory location.

The first 3 instructions are used to allocate space on that stack (ensures that there's enough room for variables to be allocated and more). We'll start looking at the program from the 4th instruction (`movl $4`). We want to analyse the program while it runs and the best way to do this is by using **breakpoints**.

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: What is the command to execute the program until we hit a breakpoint?

The answer was provided in the text shown below.

Answer 4: dc

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 transferring the value 4 into the `local_ch` variable. To view the contents of the `local_ch` variable, we use the following instruction `px @memory-address` In this case, the corresponding memory address for `local_ch` will be `rbp-0xc` (from the first few lines of **@pdf main**) This instruction prints the values of memory in hex:

```
[0x00400b55]> px @ rbp-0xc
- offset -       0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
```

Question 5: What is the value of local_ch when its corresponding movl instruction is called (first if multiple)?

The value is 1 as the instruction stated mov dwod [local_ch],1
Answer 5: 1

```
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1615 started...
= attach 1615 1615
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]> 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
[0x00400a30]>
```

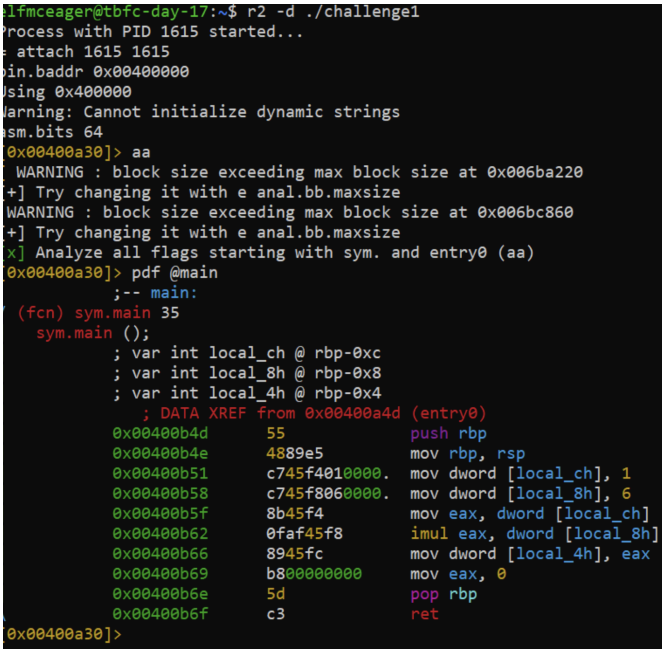
Question 6: What is the value of eax when the imull instruction is called?

Answer 6: 6


```
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1615 started...
= attach 1615 1615
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]> 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
[0x00400a30]>
```


Question 7: What is the value of local_4h before eax is set to 0?

Answer 7: 6



Thought process / methodology :

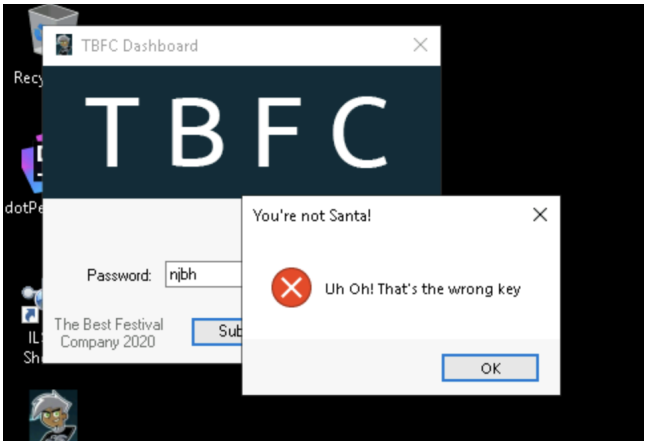
We used a secure shell(ssh) in the terminal with an ip address given such as <ssh elfmceager@ip address>and inserted the password given which is adventofcyber. Then, Run the command r2 -d ./file1. This will open the binary in debugging mode. Once the binary is open, one of the first things to do is ask r2 to analyse the program, and this can be done by typing in: aa. Once the analysis is complete, you would want to know where to start analysing from - most programs have an entry point defined as main. To find a list of the functions run: afl. As seen here, there actually is a function at main. Let’s examine the assembly code at main by running the command pdf @main Where pdf means print disassembly function.

Day 18:Reverse Engineering - The Bits of Christmas

Tools used: THM AttackBox, Firefox, Cyberchef, Remmina, TBFC_APP

Question 1 :What is the message that shows up if you enter the wrong password for TBFC_APP?

Answer 1 : Uh Oh! That's the wrong key

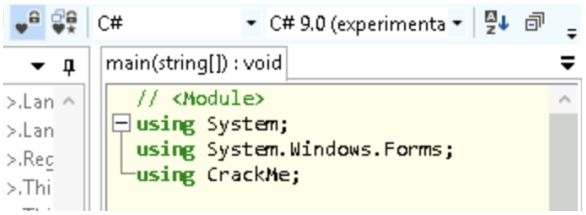


Question 2 :What does TBFC stand for?

Answer 2 : The Best Festival Company

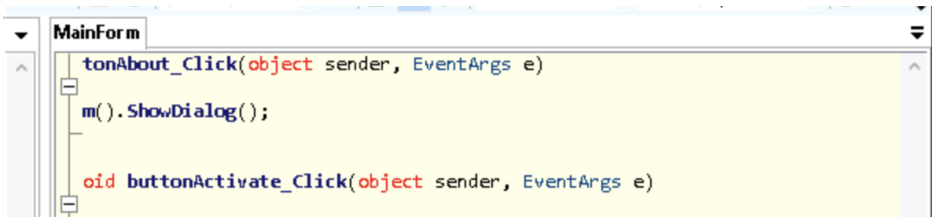
Question 3 : Decompile the TBFC_APP with ILSpy. What is the module that catches your attention?

Answer 3 : CrackMe



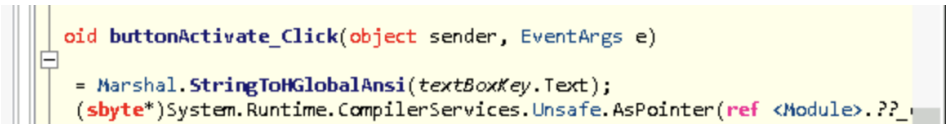
Question 4: Within the module, there are two forms. Which contains the information we are looking for?

Answer 4: MainForm



Question 5 :Which method within the form from Q4 will contain the information we are seeking?

Answer 5: buttonActivate_Click



Question 6: What is Santa's password?

Answer 6: santapassword321

Question 7: Now that you've retrieved this password, try to login...What is the flag?

Answer 7: thm{046af}

Thought Process/ Methodology:

After we opened TBF_APP, we tried entering the wrong password to find the message. Then, we decompile TBFC_APP with ILSpy. Now we can see some of the source codes behind the application. We can find all sorts of information and even passwords. After we found santa's password, we can now decode using cyberchef. Then we can get the flag by successfully login into the TBFC_APP.

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

Tools used: THM Attackbox, Firefox

Question 1: Which list is this person on?

Answer 1: Ian Chai=Nice, JJ=Naughty, Timothy=Naughty, Tib3rius=Nice, YP=Nice, Kanes=Naughty

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

Answer 2: Not Found. The requested URL was not found on this server.

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

Answer 3: Failed to connect to list.hohoho port 80: Connection refused

Question 4: What is displayed on the page when you use "/?proxy=http%3A%2F%2Flist.hohoho%3A22"?

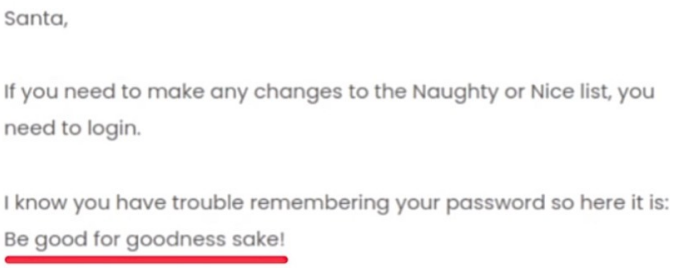
Answer 4: Recv failure: Connection reset by peer

Question 5: What is displayed on the page when you use "/?proxy=http%3A%2F%2Flocalhost"?

Answer 5: Your search has been blocked by our security team.

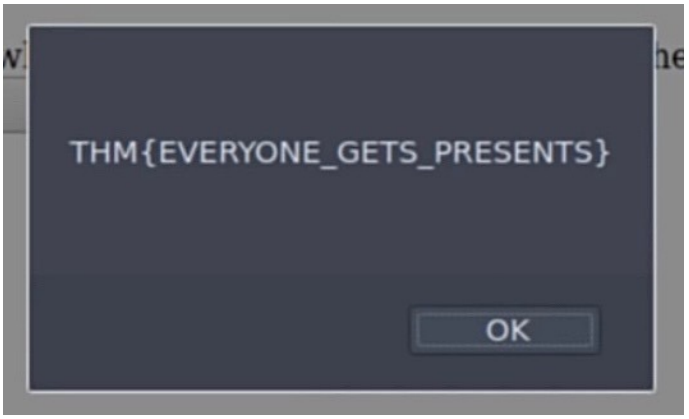
Question 6: What is Santa's password?

Answer 6: Be good for goodness sake!



Question 7: What is the challenge flag?

Answer 7: THM{EVERYONE_GETS_PRESENTS}



Thought Process/ Methodology:

We start by entering the given IP Address in the search bar which will lead us to access Santa's Naughty List or Nice List where we can enter names to check whether they are in Santa's Naughty List or Nice List. Then, we replace the URL with the given replacements which will lead us to different pages. Next, to find out Santa's password, we start by modifying the URL with localtest.me which will lead us to a message left by Santas's Elf that contains Santa's password. Lastly, we are able to login to Admin which will display the challenge flag.

Day 20: [Blue Teaming] PowershELlF to the rescue

Tools used: THM AttackBox, PowerShell, SSH

Question 1: Check the ssh manual. What does the parameter -1 do

Answer 1 : login name

```
└─$ ssh -h
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: Search for the first hidden elf file within the Documents folder. Read the contents of this file. What does Elf 1 want?

Answer 2 : 2 front teeth

```
PS C:\Users\mceager\Documents> Get-Content elfone.txt
Nothing to see here...
PS C:\Users\mceager\Documents> cat e1fone.txt
All I want is my '2 front teeth'!!!
PS C:\Users\mceager\Documents>
```

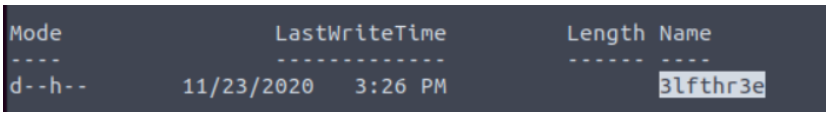
Question 3: Search on the desktop for a hidden folder that contains the file for Elf 2. Read the contents of this file. What is the name of that movie that Elf 2 wants?

Answer 3: Scrooged

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

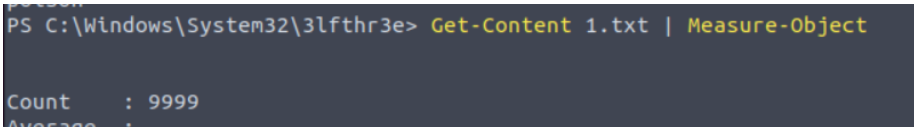
Question 4: Search the Windows directory for a hidden folder that contains files for Elf 3. What is the name of the hidden folder? (This command will take a while)

Answer 4: 3lfthr3e



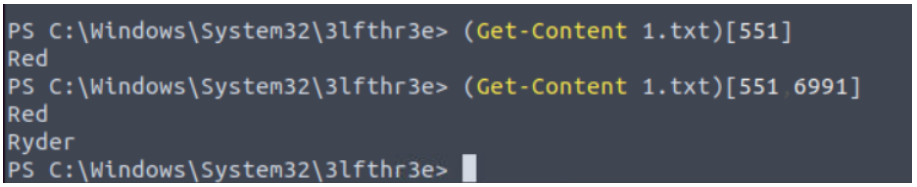
Question 5: How many words does the first file contain?

Answer 5: 9999



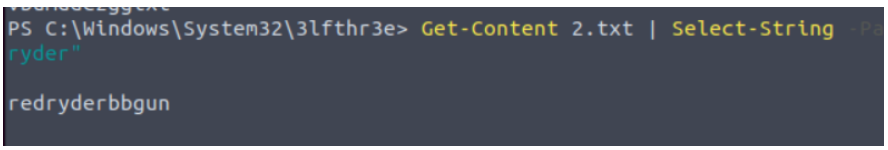
Question 6: What 2 words are at index 551 and 6991 in the first file?

Answer 6: Red Ryder



Question 7: This is only half the answer. Search in the 2nd file for the phrase from the previous question to get the full answer. What does Elf 3 want?(use spaces when submitting the answer)

Answer 7: red ryder bbgun



Thought Process/ Methodology:

First, we launched PowerShell and navigated to the Documents folder.We use powershell command to access the ‘Documents’ directory. After we navigate into the directory, we can see a list of the directory contents with Get-ChildItem but the result of this command doesn’t require the results we want. Then, we add additional flags; Get-ChildItem -Hidden -File to specify the command. Then, we use the command Get-Content to see the content of the file and find what elf1 wants. Next, we navigate into the Desktop directory where it lists all the contents inside it using powershell command; Get-ChildItem -Hidden -Directory and navigate into ‘elf2wo’ directory. Next, we’re looking for a hidden folder that contains files for Elf 3 using command Get-ChildItem -Hidden -Filter ‘*3*’ We navigate into the file we found earlier and list the file inside. Using Measure-Object cmdlet with flag -Word to count all the words. Then we use the common that has been provided (Get-Content .\1.txt)[551, 6991] to find the 2 words in the first file.For the last question, we use command Select-String <path/filename> -Pattern ‘redryder’ to get the full answer.