

PSP0201

Week 5

Writeup

Group Name: Sunny

Members:

ID	Name	Role
1211104248	Lew Chun Men	Leader
1211102048	Nur Aqilah Marsya Binti Abdul Halim	Member
1211103274	Nur Insyirah Binti Abd Jalin	Member
1211101070	Hazrel Idlan bin Hafizal	Member

DAY 16: Scripting - Help! Where is Santa?

Tools Used: Python, Firefox, THM Attackbox

Tutorial/Walkthrough:

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

To know the port number of the server, we will be using nmap.

Run command, "nmap -v machine_ip"

```
root@ip-10-10-83-223:~# nmap -v 10.10.89.87
```

From the nmap scans it was evident that the web server was running on tcp **port 80**.

```
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
```

Q2 - What templates are being used?

```
<span class="navbar-item">
  <a class="button is-white is-outlined" href="https://github.com/BulmaTemplates/bulma-templates/blob/master/templates/hero.html">
    <span class="icon">
```

For some unknown reason, there was nothing at the top left of the website on my end. So, I scrolled through the page source and found that the templates used were **BULMA**.

Q3: Without using enumerations tools such as Dirbuster, what is the directory for the API?

```

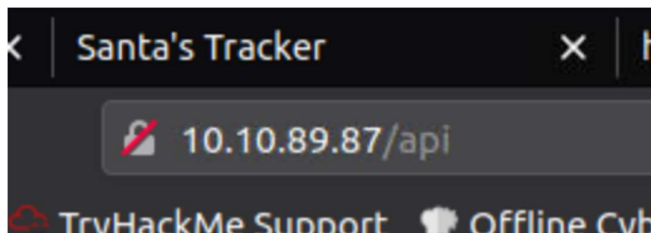
<ul>
  <li><a href="#">Labore et dolore magna aliqua</a></li>
  <li><a href="#">Kanban airis sum eschelor</a></li>
  <li><a href="http://machine_ip/api/api_key">Modular modern free</a></li>
  <li><a href="#">The king of clubs</a></li>
  <li><a href="#">The Discovery Dissipation</a></li>

```

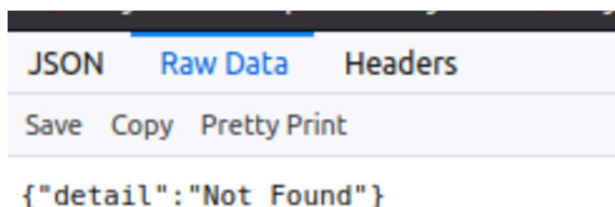
In the middle unordered list, we see one of the links contains a reference URL, which is `http://machine_ip/api/api_key`.

Q4: Go the API endpoint. What is the Raw Data returned if no parameters are entered?

To go the the API endpoint, navigate “machine_ip/api” at Firefox.



We will see a page with three tabs on top. Click the Raw Data tab to get the returning.



The Raw Data returned is `{"detail":"Not Found"}`.

Q5: Where is Santa right now?

We have to first get the correct API key in order to know the location of Santa.

```
root@ip-10-10-83-223:~# subl track.py
```

Run the command “subl filename.py” to open the Sublime text editor.

```
1 import requests
2
3 for api_key in range(1,100,2):
4     print(f'api_key {api_key}')
5     html = requests.get(f'http://10.10.89.87/api/{api_key}')
6     print(html.text)
```

Write a Python script to run through odd numbers appended to the URL and print out if we get a hit.

Run the script by using the command “python3 filename.py”

```
root@ip-10-10-83-223:~# python3 track.py
```

We can now see that santa is at **Winter Wonderland, Hyde Park, London.**

```
api_key 57
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
api_key 59
```

Q6: Find out the correct API key. Remember, this is an odd number between 0-100. After too many attempts, Santa's Sled will block you. To unblock yourself, simply terminate and re-deploy the target instance.

57, the location of santa is returned.

Thought process/ Methodology:

For today's task, santa is leaving and we are required to find the location of his location. First, we need to open the server's http interface, to do that we need to first see which port the server is running on. To do that, we will be using nmap by running the command "nmap machine_ip" and nmap will show us the list of which port and what kind of services are running on it. In our case, the http is running on port 80. Then to open the server interface, we can go to "machine_ip:80" on Firefox. We then view the page source to know the templates used, which is Bulma, and the directory for the API, /api/. Now that we know the directory, we can go to the endpoint by navigating "machine_ip/api" at Firefox. There will be three tabs on top of the page, go to the second tab and we will see the Raw Data returned when no parameters were used which is {"detail":"Not Found"}. To know the location of santa, we have to write a python script all possible API keys and append them to the API URL. After running the script, we can see that the correct API key is 57 and santa is at Winter Wonderland, Hyde Park, London

Day 17 : ReverseELFneering.

Tools used : Attackbox

Tutorial /walkthrough :

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

Byte = **1**

Word = **2**

Double word = **4**

Quad = **8**

Single Precision = **4**

Double Precision = **8**

Question 2 : What is the command to analyse the program in radare2?

The command to analyse the program in radare2 is **aa**. This means to "analyse everything." That applies to all symbols and entry points.

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

Breakpoint can be defined using **db** address/flag >. The database will simply display a list over all breakpoints.

Question 4 : What is the command to execute the program until we hit a breakpoint?

Running **dc** will run the programme until we reach the breakpoint. When we approach the breakpoint and print the main function, the rip, which is the current instruction, indicates where execution has paused.

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

After we deploy the machine, we will log into the instance using the information given.

IP address : 10.10.178.162

Username: elfmceager

Password: adventofcyber

Then, we will run **./file1** and shall see

```
elfmceager@tbfc-day-17:~$ ./file1
the value of a is 4, the value of b is 5 and the value of c is 9elfmceager@tbfc-day-17:~$ r2 -d ./file1
Process with PID 1589 started...
= attach 1589 1589
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
asm.bits 64
[0x00400a30]> █
```

Next, run **r2 -d ./file1** and run **aa** for the machine to analyse all the flags. We also can see the function list by running **afl**.

Afterwards, we will run **pdf @main** and it may disassembly the code.

```
elfmceager@tbfc-day-17: ~
File Edit View Search Terminal Help
;-- main:
(fix) sym.main 68
sym.main ();
; var int local_ch @ rbp-0xc
; var int local_8h @ rbp-0x8
; var int local_4h @ rbp-0x4
; DATA XREF from 0x00400b4d (entry0)
0x00400b4d 55      push rbp
0x00400b4e 4889e5   mov rbp, rsp
0x00400b51 4883ec10 sub rsp, 0x10
0x00400b55 c745f4040000 mov dword [local_ch], 4
0x00400b5c 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 8b4d0fc  mov ecx, dword [local_4h]
0x00400b71 8b55f8   mov edx, dword [local_ch]
0x00400b74 8b45f4   mov eax, dword [local_8h]
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_value_of_c_is_d ; 0x002000
; "the value of a is %d, the value of b is %d and the value of c is %d"
0x00400b80 b8000000 mov eax, 0
0x00400b85 e8f6ea0000 call sym.__printf
0x00400b8a b800000000 mov eax, 0
0x00400b8f c9       leave
0x00400b90 c3       ret
```

sym.main indicates that we are currently in the main function. So now, we are going to set a breakpoint to decide where the program should be executed. We will run **db 0x00400b55** and afterwards, we may run the program using **dc** and it shall run until the breakpoint that has been decided.

Next, we can see that **local_ch** is **rbp-0xc**.

```
0x00400b90 c3 ret
[0x00400b55]> px @rbp-0xc
offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
7ffc0ffc9644 0000 0000 1890 6b00 0000 0000 7018 4000 .....k.....p.@.
7ffc0ffc9654 0000 0000 1911 4000 0000 0000 0000 0000 .....@.....
0x7ffc0ffc9664 0000 0000 0000 0000 0000 0100 0000 7897 fc0f .....X...
0x7ffc0ffc9674 fc7f 0000 4d0b 4000 0000 0000 0000 0000 ....M.@.....
0x7ffc0ffc9684 0000 0000 1700 0000 0100 0000 0000 0000 .....
0x7ffc0ffc9694 0000 0000 0000 0000 0200 0000 0000 0000 .....
0x7ffc0ffc96a4 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffc0ffc96b4 0000 0000 0000 0000 0000 0000 0004 4000 .....@.
0x7ffc0ffc96c4 0000 0000 be2e da04 a9ca e096 1019 4000 .....@.
0x7ffc0ffc96d4 0000 0000 0000 0000 0000 0000 1890 6b00 .....k.
0x7ffc0ffc96e4 0000 0000 0000 0000 0000 0000 be2e fa18 .....
0x7ffc0ffc96f4 d0d5 1869 be2e ae15 a9ca e096 0000 0000 ...i.....
0x7ffc0ffc9704 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffc0ffc9714 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffc0ffc9724 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffc0ffc9734 0000 0000 0000 0000 0000 0000 0000 0000 .....
[0x00400b55]>
```

If we take a look at the memory location after running the command **ds**, it shall appear like this;


```

[0x00400b55]> ds
[0x00400b55]> px @rbp-0xc
offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
7ffc0ffc9644 0400 0000 1890 6b00 0000 0000 7018 4000 .....k.....p.@.
7ffc0ffc9654 0000 0000 1911 4000 0000 0000 0000 0000 .....@.....
0x7ffc0ffc9664 0000 0000 0000 0000 0100 0000 7897 fc0f .....x...
0x7ffc0ffc9674 fc7f 0000 4d0b 4000 0000 0000 0000 0000 ....M.@.....
0x7ffc0ffc9684 0000 0000 1700 0000 0100 0000 0000 0000 .....
0x7ffc0ffc9694 0000 0000 0000 0000 0200 0000 0000 0000 .....
0x7ffc0ffc96a4 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffc0ffc96b4 0000 0000 0000 0000 0000 0000 0004 4000 .....@.
0x7ffc0ffc96c4 0000 0000 be2e da04 a9ca e096 1019 4000 .....@.
0x7ffc0ffc96d4 0000 0000 0000 0000 0000 0000 1890 6b00 .....k.
0x7ffc0ffc96e4 0000 0000 0000 0000 0000 0000 be2e fa18 .....
0x7ffc0ffc96f4 d0d5 1869 be2e ae15 a9ca e096 0000 0000 ...t.....
0x7ffc0ffc9704 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffc0ffc9714 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffc0ffc9724 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffc0ffc9734 0000 0000 0000 0000 0000 0000 0000 0000 .....
[0x00400b55]>

```

The first two bytes are 4, so if we do the same thing for the next instruction, local_8h will become 5.

To see the product from register %eax, we will use the command dr.

```

[0x00400b55]> ds
[0x00400b55]> ds
[0x00400b55]> ds
[0x00400b55]> dr
rax = 0x00000009
rbx = 0x00400400
rcx = 0x0044ba90
rdx = 0x00000004
rsi = 0x01000000
r9 = 0x006bb8e0
r10 = 0x00000015
r11 = 0x00000000
r12 = 0x00401910
r13 = 0x00000000
r14 = 0x006b9018
r15 = 0x00000000
rsi = 0x7ffc0ffc9778
rdi = 0x00000001
rsp = 0x7ffc0ffc9640
rbp = 0x7ffc0ffc9650
rip = 0x00400b6b
rflags = 0x00000206
orax = 0xffffffffffffffff
[0x00400b55]>

```

Then we will run pdf @main to see our latest file.


```

(func) 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

```

Based on the screenshot, we can see that `mov dword [local_ch], 1` is stated showing that the variable `local_ch` is set to 1.

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

Based on the definition;

`imulq source, destination: destination = destination * source`

We can say that;

`mov eax, dword [local_ch]` (set `eax` to 1)

`imul eax, dword [local_8h]` (destination * source or $6(1) = 6$)

Thus, the answer is **6**.

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

Based on question 6, we are sure that `eax` was set to 6. So, `mov dword [local_4h], eax` sets `[local_4h]` to `eax`'s value of **6**.

Thought process / Methodology :

For today's task, we are introduced to x86-64 Assembly where machine code is typically represented by assembly code, a more understandable version of the code. We'll be doing today's task with radare2, which is a framework for reverse engineering and analysing binaries. It is capable of disassembling binaries. First, we'll use `r2 -d./file1` to access the binary files in debug mode. The radare will then analyse the file `aa`. When it's finished, we can use `afl | grep main` to find the file's entry point. At this point, we can see where the programme begins in memory. We may also look at it using `pdf@main`. The breakpoint may then be set using `db`. We may run `pdf@main` again and observe a letter `b` immediately after the break point position. Then, using the `dc` command, we may execute the programme until it reaches the break point. We can use `px @rbp-0cx` to examine the contents of the local `ch` variable, but we still need to use `ds` since we can't see variable 4. When it is returned, we may proceed to the challenge file and repeat the first stages. We may now see the `pdf @main` file.

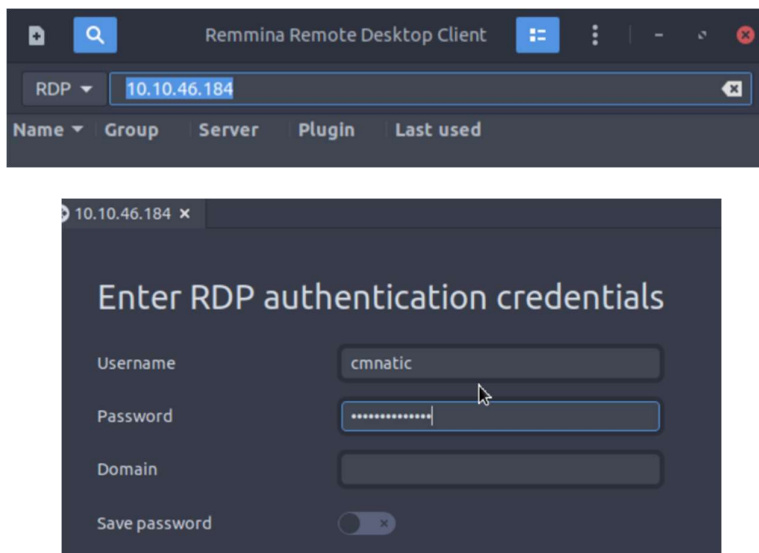
Day 18: Reverse Engineering The Bits of Christmas

Tools Used: THM Attackbox, ILspy, Remmina

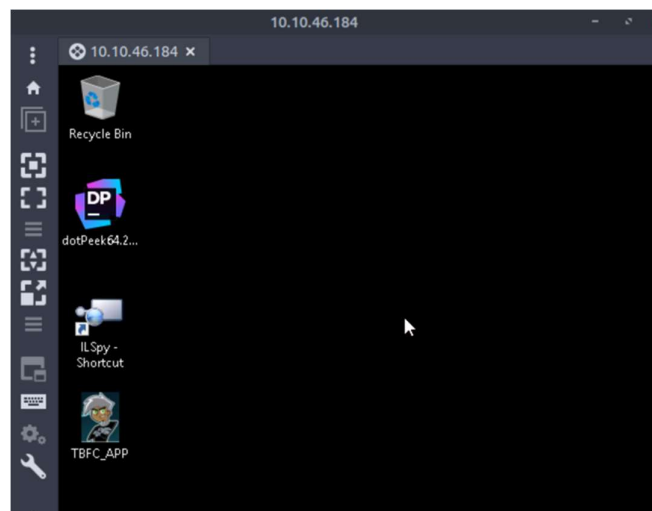
Tutorial/Walkthrough:

Q1: What is the message that shows up if you enter the wrong password for TBFC_APP?

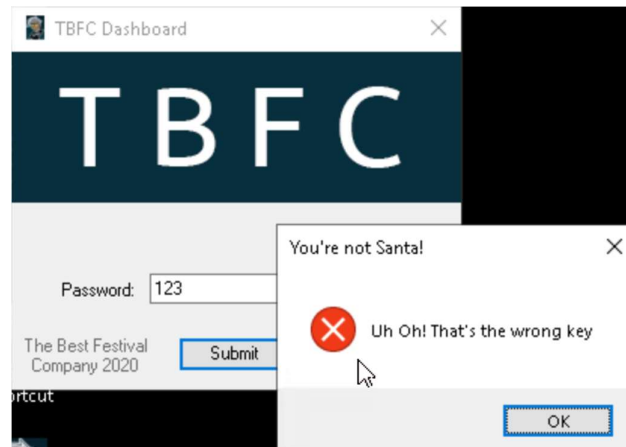
Enter our machine_ip and connect to the machine instance, Remmina using the username cmnatic and password Adventofcyber!



We will see the TBFC_APP at the left side of the Windows Remote Desktop.

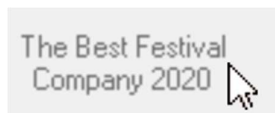


Open the file by clicking twice and enter a random password to get the message. There will be a pop up window with “You’re not Santa” as the title and “ Uh Oh! That’s the wrong key” as the message.



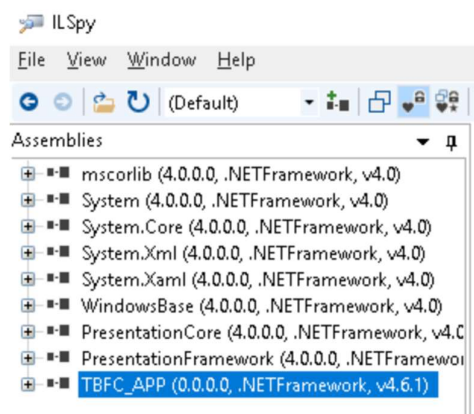
Q2: What does TBFC stand for?

The Best Festival Company, this is stated at the bottom left corner of the app dashboard.

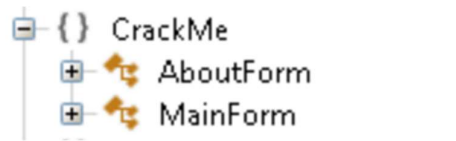


Q3: Decompile the TBFC APP with ILSpy. What is the module that catches your attention?

Open TBFC_APP in ILSpy by clicking the File > Open to navigate to the file app.

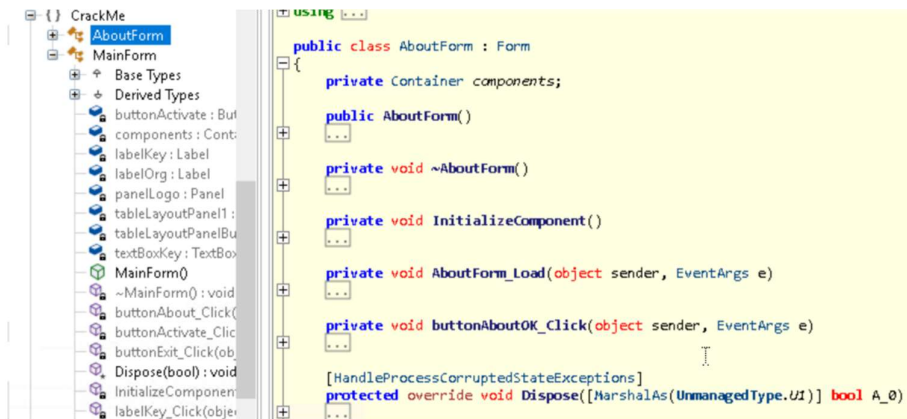


The file that catches our attention was **CrackMe** as it looks like this is where the form is handled.



Q4: Within the module, there are two forms. Which contains the information we are looking for?

After skimming through the AboutForm, we noticed that there is no information related to what we are looking for.



MainForm is the answer as there is a method that has a password check.



Q5: Which method within the form from Q4 will contain the information we are seeking?

```
te 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*)value;
```

buttonActivate_Click, it has a sensitive value and references the correct password.

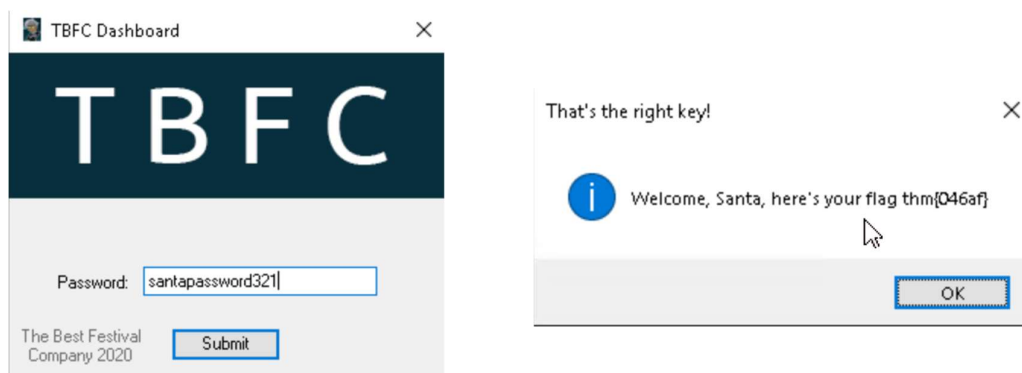
Q6: What is Santa's password?

```
es (sbyte*)System.Runtime.CompilerServices.Unsafe.AsPointer(ref <Module>._?_C@_0BB@IKKDFEPG@santapassword321@);  
internal static <CppImplementationDetails>.$ArrayType$$$8Y0BB@$$CBD global::<Module>._?_C@_0BB@IKKDFEPG@santapassword321@
```

From the code, it is obvious that the password to the app is **santapassword321**.

Q7: Now that you've retrieved this password, try to login...What is the flag?

Go to the app window again and enter the password that we have just retrieved, santapassword321.



After submitting, a pop up window appeared showing the flag **thm{046af}**.

Thought process/ Methodology:

Santa forgot his password again and today we are assigned to retrieve the password. Connect our machine to Remmina by entering the IP address, username and password provided. Once we are connected, a Windows desktop should appear with the TBFC_APP file at the left side. We can see the app dashboard. We then entered a random password to get the message if we enter the wrong password. A window appeared with "Uh Oh! That's the wrong key" as its message. At the bottom of the dashboard, we can get the full name of the app which is The Best Festival Company 2020. After decompiling the app file on ILspy, we found that the module CrackMe is interesting as it is where the form were handled. We can see two forms but only one of it has the information related to what we were looking for, which is the second form MainForm. After a bit of digging, we found that there is a sensitive value in buttonActivate_Click. We saw a password check in there and it references the correct password, santapassword321. To get the flag, we submitted the password at the app dashboard and a pop up window appeared with the flag thm{046af}.

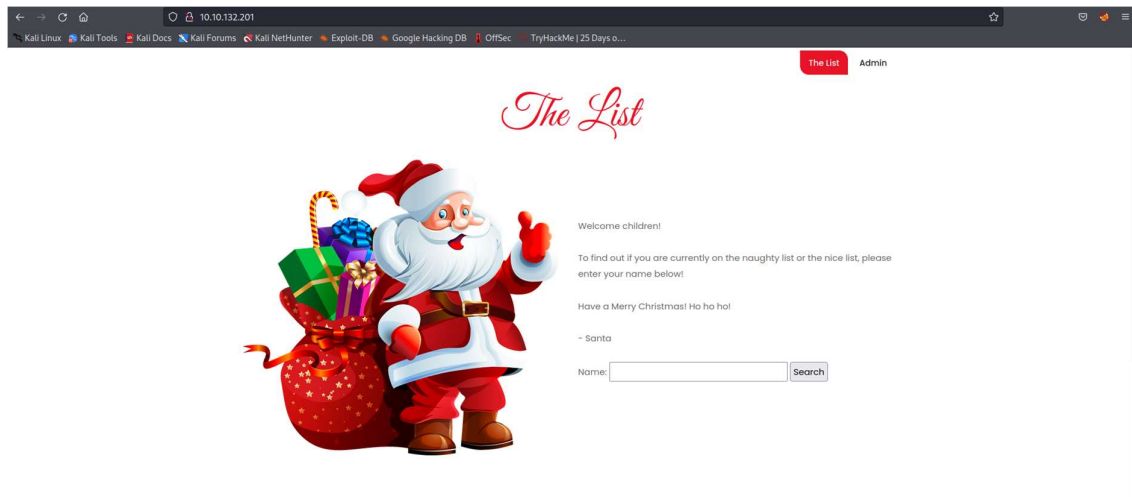
Day 19: Web Exploitation – The Naughty or Nice List

Tools used: Kali Linux, Firefox, Cyberchef

Tutorial/Walkthrough:

Question 1 – Which list is this person on?

Go to “machine.ip” in the browser in order to go to the web app.



And then to see if the person is on the “Naughty” list or in the “Nice” list, we can enter their name into the search bar one by one.

If we enter the name “Tib3rius”, we can see that this person is in the “Nice” list from the prompt “Tib3rius is on the Nice List.”

Name:

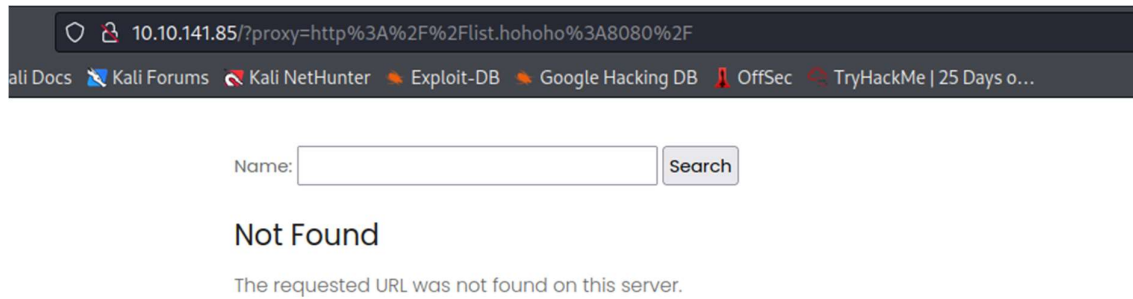
Tib3rius is on the Nice List.

However if the person is in the “Naughty” list, we should know by the prompt “‘person’ is on the Naughty List.”

Name:

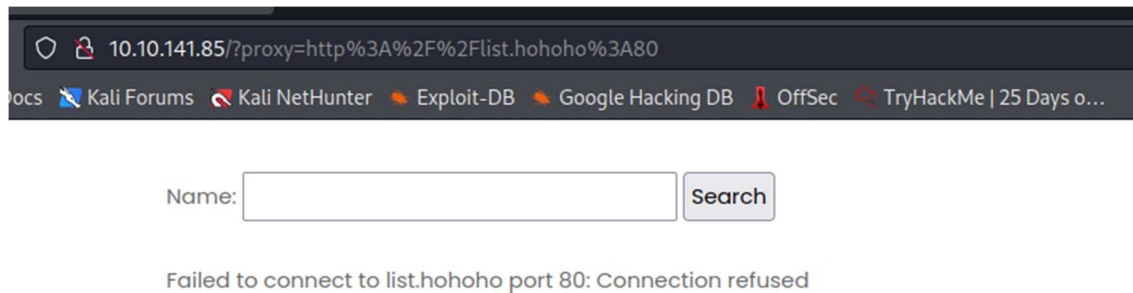
Kanes is on the Naughty List.

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



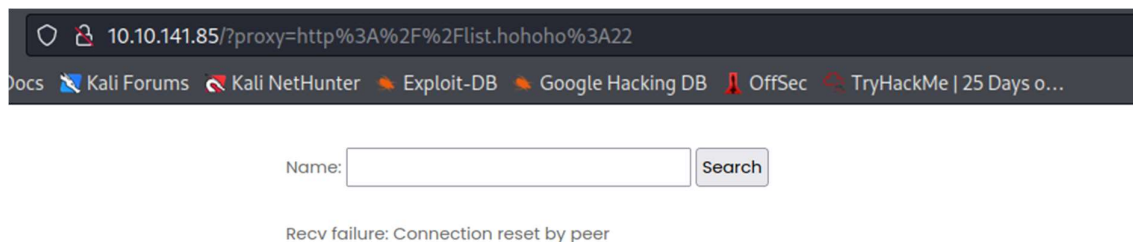
The page displays, “Not Found. The requested URL was not found on this server.”

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



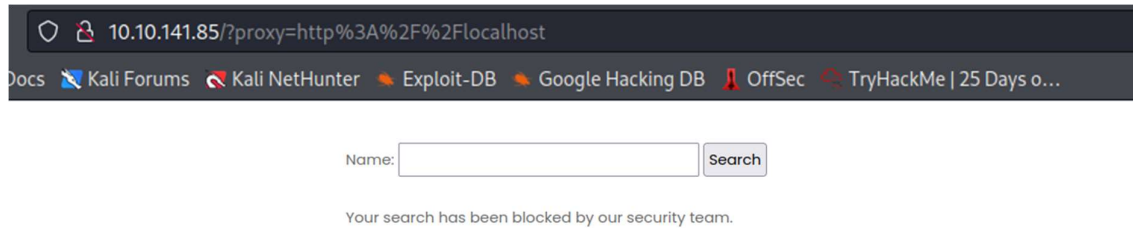
The page shows, “Failed to connect to list.hohoho port 80. Connection refused.”

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



The page shows, “Recv failure: Connection reset by peer”

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

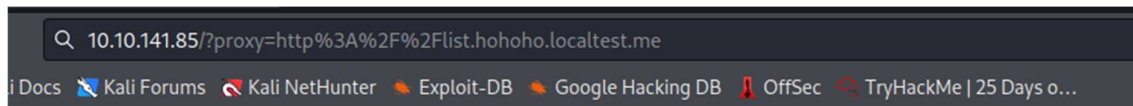


The screenshot shows a web browser window with the address bar displaying `10.10.141.85/?proxy=http%3A%2F%2Flocalhost`. The browser's bookmark bar includes links to Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, OffSec, and TryHackMe | 25 Days o... Below the browser, there is a search form with a text input labeled "Name:" and a "Search" button. Below the search form, a message states: "Your search has been blocked by our security team."

The page says, “Your search has been blocked by our security team.”

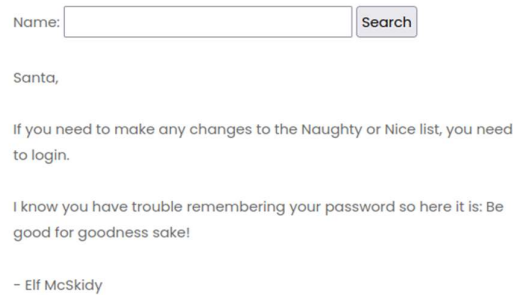
Question 6 – What is Santa’s password?

In order to get the password, we need to use “/?proxy=http%3A%2F%2Flist.hohoho.localtest.me”.



The screenshot shows a web browser window with the address bar displaying `10.10.141.85/?proxy=http%3A%2F%2Flist.hohoho.localtest.me`. The browser's bookmark bar includes links to Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, OffSec, and TryHackMe | 25 Days o... Below the browser, there is a search form with a text input labeled "Name:" and a "Search" button. Below the search form, the text "Santa," is displayed. Below that, a message states: "If you need to make any changes to the Naughty or Nice list, you need to login." Below that, a message states: "I know you have trouble remembering your password so here it is: Be good for goodness sake!" Below that, the text "- Elf McSkidy" is displayed.

And when it is used, we can see that Elf McSkidy has left the password, which is,



The screenshot shows a web browser window with the address bar displaying `10.10.141.85/?proxy=http%3A%2F%2Flist.hohoho.localtest.me`. The browser's bookmark bar includes links to Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, OffSec, and TryHackMe | 25 Days o... Below the browser, there is a search form with a text input labeled "Name:" and a "Search" button. Below the search form, the text "Santa," is displayed. Below that, a message states: "If you need to make any changes to the Naughty or Nice list, you need to login." Below that, a message states: "I know you have trouble remembering your password so here it is: Be good for goodness sake!" Below that, the text "- Elf McSkidy" is displayed.

“Be good for goodness sake!”

Question 7 – What is the challenge flag?

To get the challenge flag to complete the task, which is to access the administration list and delete the naughty list.

In order to do that, we need to login as admin. As we already know the password, now we just need to guess the username.

By trial and error and by inputting common usernames like “Admin”, “user” or “Santa”, we found the username to be “Santa”.



A login form with a red cursive title "Admin" above it. The form contains two input fields: "Username:" with the value "Santa" and "Password:" with a masked password of 15 dots. Below the fields is a "Login" button.

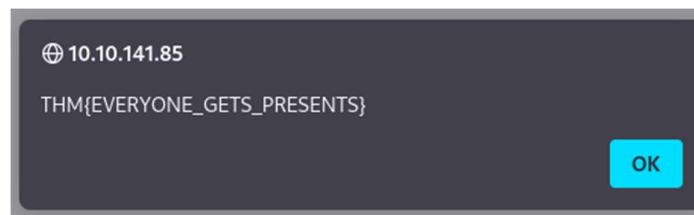
After we click login, we are brought to the list administration.

List Administration

This page is currently under construction.

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

After clicking on the button “DELETE NAUGHTY LIST”,



An alert with the flag pops up.

Thought Process/Methodology:

In today's task, we are to use Server-Side Request Forgery to access resources of a web app that we are normally not able to access and clear the naughty list. When we first open the web app, a page with the search bar and an admin login appears. When we enter the name of a person into the search bar, we can see if this person is on the nice list or on the naughty list. The URL of the webapp becomes

"http://machine.ip/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%3DTib3rius", so if we use Cyberchef to decode the URL we get,

"http://machine.ip/?proxy= http://list.hohoho:8080/search.php?name=Tib3rius". It seems the web app acts like a proxy and takes this URL, then makes a request at a back-end machine and outputs the result back to the web app. We will be trying to exploit this function using Server-Side Request Forgery. First, we will try to access the root of the site by going to "http://machine.ip/?proxy=http://list.hohoho:8080/" but in its encoded form so that the website interprets the URL properly. Upon doing that, a message shows us "Not Found. The requested URL was not found on this server." This method did not work and now we try to change the port to the default HTTP port which is port 80 by using the URL "http:// machine.ip/?proxy= http://list.hohoho:80/" in its encoded form. A message shows "Failed to connect to list.hohoho port 80: Connection refused", which means that port 80 is not open on the back-end server. Now we try to change the port to the default SSH port which is port 22 by using the URL "http://machine.ip/? proxy=http://list.hohoho:22/". A message shows "Recv failure: Connection reset by peer" which means that an SSH server is open on port 22, but since the SSH server cannot interpret a HTTP request, this error shows up. This also did not work, and now we will try to access services running locally on the server by changing the hostname, "list.hohoho", to "localhost" by using the URL, "http://machine_ip/?proxy=http://localhost" in the encoded form. A message shows, "Your search has been blocked by our security team." This means that the developer added a function that checks to ensure that the hostname provided start with "list.hohoho", if not the request will be blocked. To bypass this checking function, we can make use of DNS subdomains and use a subdomain that resolves to 127.0.0.1, an example of that is "localtest.me". Which means we can replace the hostname with "list.hohoho.localtest.me" to bypass the check and access the local services. The URL becomes, "http://machine_ip/?proxy=http://list.hohoho.localtest.me". Upon going to that URL, a message containing the password is shown. Now that we have the password, we are missing the username. We can guess the correct username by trial and error using common usernames or in this case santa's username. Ultimately, we found the username to be "Santa" and when we log in to the admin page, a button saying "Delete naughty list" is shown. When we click on the button, the challenge flag is shown.

Day 20: Blue Teaming – PowershELF to the rescue

Tools used: Kali Linux, Terminal, SSH, PowerShell

Tutorial/Walkthrough:

Question 1 – Check the ssh manual. What does the parameter -l do?

To check the SSH manual, we can use the command “ssh” in the terminal.

```
(1211104248@kali)-[~]  
$ ssh  
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 ... ]]
```

The -l parameter allows us to specify the login name.

Question 2 – Search for the first hidden elf file within the Documents folder. Read the contents of this file. What does Elf 1 want?

First, we need to login to the server.

To do that, we will use the command “ssh -l mceager machine.ip”, then enter the password that we are given.

```
(1211104248@kali)-[~]  
$ ssh -l mceager 10.10.180.69  
The authenticity of host '10.10.180.69 (10.10.180.69)' can't be established.  
ED25519 key fingerprint is SHA256:X2ViBkLLQoHmAsXFoem36jkl9faKH+Fr2lt2dd/kIWY.  
This host key is known by the following other names/addresses:  
  ~/.ssh/known_hosts:10: [hashed name]  
Are you sure you want to continue connecting (yes/no/[fingerprint])? ye  
Please type 'yes', 'no' or the fingerprint: yes  
Warning: Permanently added '10.10.180.69' (ED25519) to the list of known hosts.  
mceager@10.10.180.69's password:
```

When we are logged on, we arrive at the Command Prompt.

Then we use the command “powershell” to launch PowerShell.

```
mceager@ELFSTATION1 C:\Users\mceager>powershell  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
PS C:\Users\mceager> █
```

Now we can search for the first hidden elf file in the “Documents” folder.

First, we need to go to the "Documents" folder. To do that use the command "cd Documents".

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

Now we can search for hidden files in this directory. To do that use the command "Get-ChildItem -Hidden -File -ErrorAction SilentlyContinue".

"Get-ChildItem" is to specify that we are using the "Get-ChildItem" cmdlet. "ls" and "dir" are aliases of this command and they can be used in place of the "Get-ChildItem" part.

"-Hidden" is to specify that only hidden files are shown.

"-File" is to specify that we want to get a list of files instead of directories. If we want to get a list of directories/folders we can replace "-File" with "-Directories".

"-ErrorAction SilentlyContinue" is to specify the action carried out by PowerShell when there is an error during the "Get-ChildItem" operation.

```
PS C:\Users\mceager\Documents> ls -Hidden -File -ErrorAction SilentlyContinue
Directory: C:\Users\mceager\Documents

Mode                LastWriteTime         Length Name
----                -
-a-hs-            12/7/2020  10:29 AM           402 desktop.ini
-arh--            11/18/2020   5:05 PM           35 elfone.txt
```

As shown, the file of name of the first hidden elf file is "elfone.txt".

To read the contents of this file, we can use the command "Get-Content elfone.txt" or use "cat" or "type" in place of "Get-Content" as they are aliases of this cmdlet.

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

As we can see Elf 1 wants "2 front teeth".

Question 2 – 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?

First, we need to go to the "Desktop" directory.

To do that, we need to go up one directory back to "\mceager" using the command "cd .." then we can go to the "Desktop" directory using the command "cd Desktop".

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

Now we can search for the hidden folder using the command "ls -Hidden -Directory -ErrorAction SilentlyContinue".

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

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

Then we can use “cd elf2wo” to go into this folder and use “ls” to find the file.

Then to read the contents of this file, use the command “type e70smsW10Y4k.txt”.

```
PS C:\Users\mceager\Desktop\elf2wo> type e70smsW10Y4k.txt
I want the movie Scrooged <3!
PS C:\Users\mceager\Desktop\elf2wo> ls
Mode                LastWriteTime         Length Name
----                -
-a                11/17/2020  10:26 AM             64 e70smsW10Y4k.txt
```

As we can see, elf 2 want the movie “**Scrooged**”.

Question 4 – Search the Windows directory for a hidden folder that contains files for Elf 3. What is the name of the hidden folder?

First, we need to go into the “Windows” directory, which is located at the root directory.

To go to the root directory, use the command “cd /” then we can go to the “Windows” directory using the command “cd Windows”.

```
PS C:\Users\mceager\Desktop\elf2wo> cd /
PS C:\> cd Windows
PS C:\Windows>
```

Now that we are at the “Windows” directory, we can start the search by using the command,

“ls -Hidden -Directory -ErrorAction SilentlyContinue -Recurse -Filter ‘*3*’”

“-Recurse” is to specify the folders in this directory and the childitems in the folders.

```
PS C:\Windows> ls -Hidden -Directory -ErrorAction SilentlyContinue -Recurse -Filter '*3*'
This will allow you to read the contents of a file.
Directory: C:\Windows\System32
Get-Content -Path file.txt
Mode                LastWriteTime         Length Name
----                -
d--h--            11/23/2020   3:26 PM             3lfthr3e
```

“-Filter ‘*3*’” is to filter the search and specify that the name of the folder contains the character ‘3’.

As we can see, the name of the folder is “3lfthr3e”.

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

First, we need to navigate to elf three's folder.

```
PS C:\Windows> cd System32
PS C:\Windows\System32> cd 3lfthr3e
PS C:\Windows\System32\3lfthr3e>
```

We can use the command “ls -Hidden” to see the file.

```
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
```

Then to see how many words it contains, we will use the command, “Get-Content -Path 1.txt | Measure-Object -Word”.

“Get-Content” gets the contents from the file specified in “Path” and “|” pipes the result to another function which in this case is “Measure-Object” which counts the amount of object specified.

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

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

As we can see the number of words in the first file is “9999”.

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

To do this we use the command “(Get-Content -Path 1.txt)[551]” and then “(Get-Content -Path 1.txt)[6991]”

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

As we can see the 2 words are “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?

To do this we will use the command “Select-String -Path ‘2.txt’ -Pattern ‘redryder’”.

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

Hence, we get the answer “redryderbbgun”.

Thought Process/Methodology:

In today's task, we accessed a Windows machine through SSH and use PowerShell and cmdlets to find hidden files and to do certain functions.

First, we need to find a hidden elf file in the Documents folder and read its contents. So we navigated to the Documents folder and search for hidden files, and the name of elf one's file show up, allowing us to read its contents.

Second, we need to find a hidden folder that contain the file for elf two in the desktop directory. To do this, navigated to the desktop directory and searched for hidden folders. A folder by the name "elf2wo" shows up. The name of the folder indicated that it is elf two's folder, and when we navigated into the folder and get the child item, we get the file name, allowing us to read its contents.

Third, we need to search the Windows directory for a hidden folder that contains the files for Elf three. To do this, we navigated to the Windows directory searched for hidden folders but we added a few more function which is "Recurse" allowing us to search further through and "Filter" allowing us to narrow down the search results. Upon doing so, a folder named, "3lfthr3e" shows up. In the folder we found two hidden files named "1.txt" and "2.txt". To find how many words the first file contained by using the command "Get-Content -Path 1.txt | Measure-Object -Word", upon which the number of words is shown. Next, we are to find two word at index 511 and index 6991. To do this we used the command, "(Get-Content -Path 1.txt)[511]" and "(Get-Content -Path 1.txt)[6991]". The word at index 511 is "Red" and the word at index 6991 is "Ryder". Next, we are to search for the phrase "Red Ryder" to get what Elf three wants. To do this, we used the command "Select-String -Path '2.txt' -Pattern 'redryder'" upon which the word that shows up is "redryderbbgun".