PSP0201

Week 5

Writeup

Group Name: Modus Potent

Members

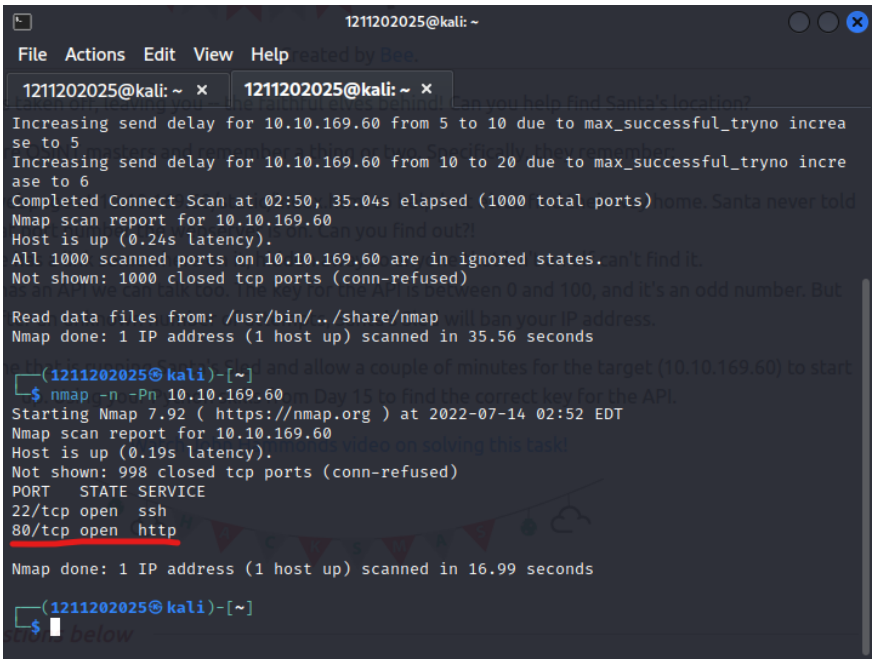
ID	Name	Role
1211200107	Afiezar Ilyaz bin Alfie Iskandar	Leader
1211202025	Abdullah bin Kamaruddin	Member
1211103649	Nur Qistina binti Roslan	Member

Day 16: Scripting - Help! Where is Santa?

Tools Used: Firefox, Terminal

Question 1

Using the command **nmap 10.10.169.60** (not using the parameter -n and -Pn is also okay, since we used for example -n to simply speed up the scanning process), we discovered the port number for the web server, which is **80**.

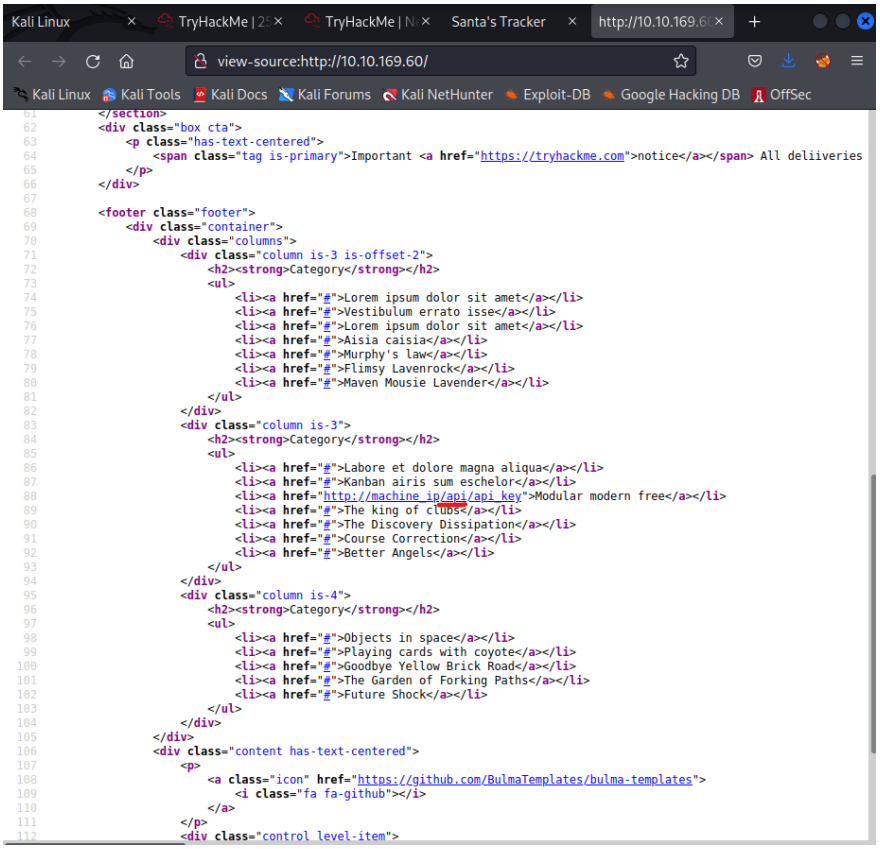


Question 2

After going to the IP address with the port, **10.10.169.60:80**, we found that the template that is being used is **BULMA**, indicated at the top left corner of the website.

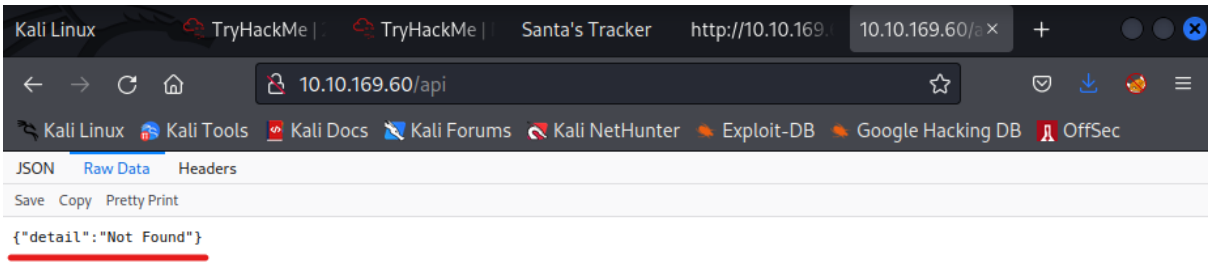
Question 3

By clicking and viewing the page source of the website, we were able to locate the directory for the API, which is **/api/**

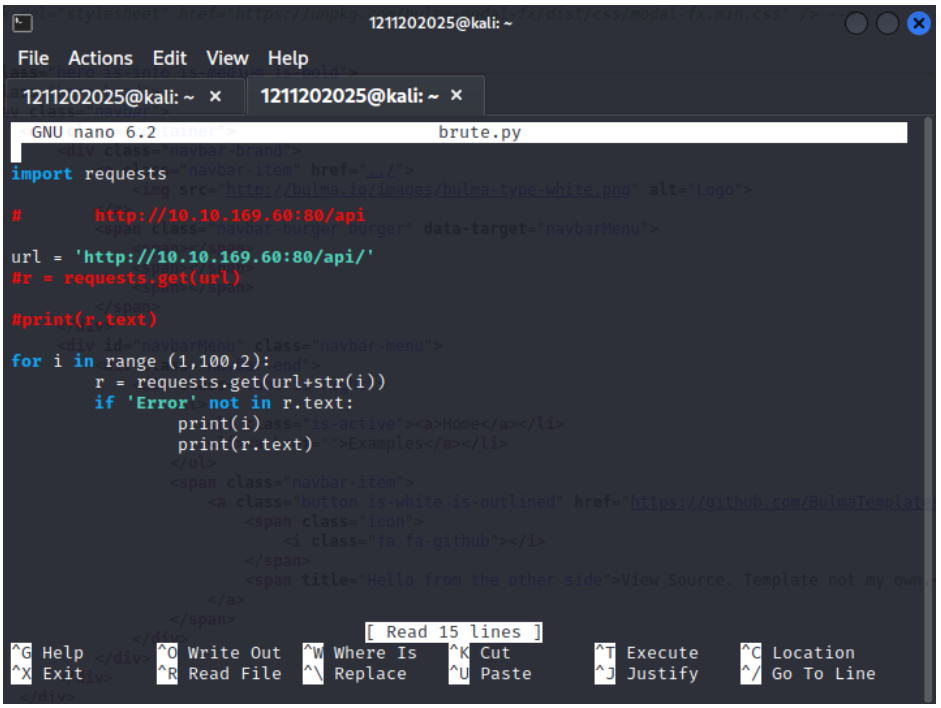


Question 4

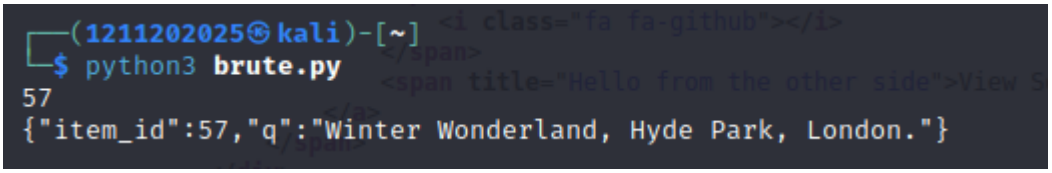
After obtaining the API directory, we continued and went to the API endpoint by including the /api/ directory into the IP address of the machine, **10.10.169.60/api**. From there, we clicked on the raw data tab and found the website response, which is **{"detail":"Not Found"}**



Question 5&6



Using the code above after creating a file named brute.py and putting it in there, we were able to find the location of Santa right now, which is **Winter Wonderland, Hyde Park, London** as well as the correct API key which is **57**.



Thought Process/Methodology:

For question 1, having access to the machine's IP address, we use the nmap command to find the port number used for the web server, which is 80. Question 2, we knew that BULMA was the template that was used for the website after we went to the website by putting in the IP address as well as the port, which in our case would be 10.10.169.69:80. BULMA was shown at the top left corner of the website. Question 3, the directory for the API was found in the page source of the website, that is /api/. Question 4, After knowing the directory for the API, we went straight to its endpoint and looked for the raw data to find the answer. Located there, is the website response, which is {"detail":"Not Found"}. Question 5 and 6, we located the answers for the 2 questions simultaneously by using a code with the help of the information given from both TryHackMe and the internet. But before that, we used the command nano brute.py, allowing us to both create a file named brute.py and straight away edit it, putting in the code that we obtained. Using the knowledge and information at hand, we adjusted the code to just give us the answers right away without all the errors and stuff, by letting it print out the outcome if and only if 'Error' is not in the text. After all things done, we got the answers, Winter Wonderland, Hyde Park, London for the location of Santa and 57 for the correct API key.

Day 17: Reverse Engineering - ReverseELFngeering

Tools Used: THM Attackbox, Terminal

Question 1

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 answer is the command aa

Question 3

What is the command to set a breakpoint in radare2?

The answer is the command db

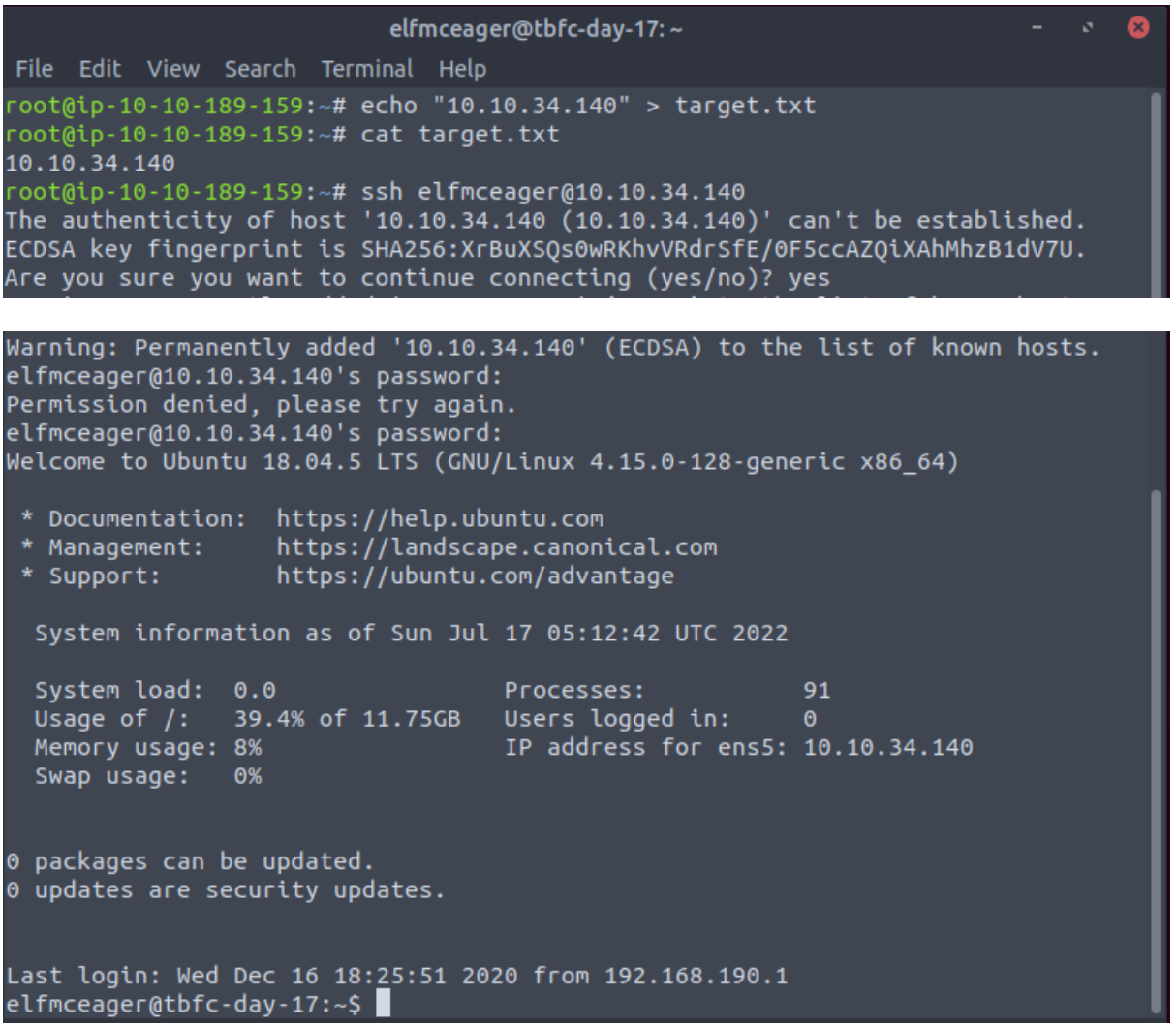
Question 4

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

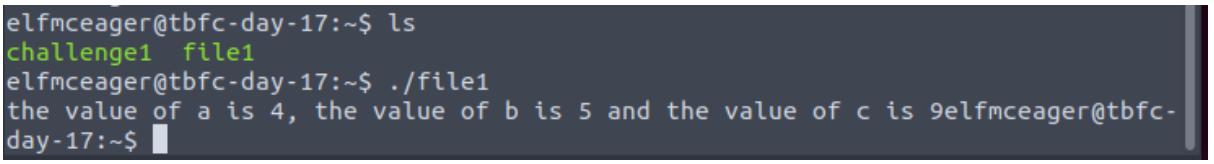
The answer is the command dc

Question 5

1. We will firstly set the target to 10.10.34.140 using the echo command and we will start a ssh with elfmceager and the ip address 10.10.34.140. Type in the password “adventofcyber” to get access



2. Type in ls command to see what's inside. we find out that there are 2 files inside, challenge1 and file1



3. using the radare2, we will open the challenge1 file using the command *r2 -d ./challenge1* then we will analyze using the *aa* command

```
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1781 started...
= attach 1781 1781
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
asm.bits 64
[0x00400a30]> aa
[ ] Analyze all flags starting with sym. and entry0 (aa)
```

4. Examine the assembly code at the main by running the *pdf @main* command. For the question on what is the value of local_ch when its corresponding movl instruction is called (first if multiple)? The answer to this is 1.

```
[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?

We will move value 1 into eax which is 6, meaning 1 is multiplied by 6, giving us the answer 6

```
[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?

mov eax 0 is after the imull instruction is called, meaning that before it sets to 0, the value of local_4h is 6.

```
[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]>
```

Thought Process/Methodology:

For the first question, we know that a byte is 1, a word is 2, a double word is 4, a quad is 8, single precision is 4 and double precision is 8. For question 2, what is the command to analyze the program in radare2? The answer is the command aa. Now on question 3, what is the command to set a breakpoint in radare2? The answer is the command db. Moving on to question 4, what is the command to execute the program until we hit a breakpoint? The answer is the command dc. Next, question 5 asks what is the value of local_ch when its corresponding movl instruction is called (first if multiple). To get the answer, first, we will set the target to 10.10.34.140 using the echo command and start a ssh with elfmceager and the IP address 10.10.34.140. Type in the password "adventofcyber" to get access. Then type in ls command to see what's inside. We find out that there are 2 files inside which are called challenge1 and file1. we will now be using the radare2 to open the challenge1 file using the command *r2 -d ./challenge1* then we will analyze using the *aa* command. After that, examine the assembly code at the main by running the *pdf @main* command. We will find the answer to the fifth question, which is 1. Moving on to question 6, what is the value of eax when the imull instruction is called? To get the answer, we will move value 1 into eax which is 6, meaning 1 is multiplied by 6, giving us the answer 6. Last but not least, question 7 asks what is the value of local_4h before eax is set to 0? looking at the main, mov eax 0 is after the imull instruction is called, meaning that before it sets to 0, the value of local_4h is 6.

Day 18: Reverse Engineering - The Bits of Christmas

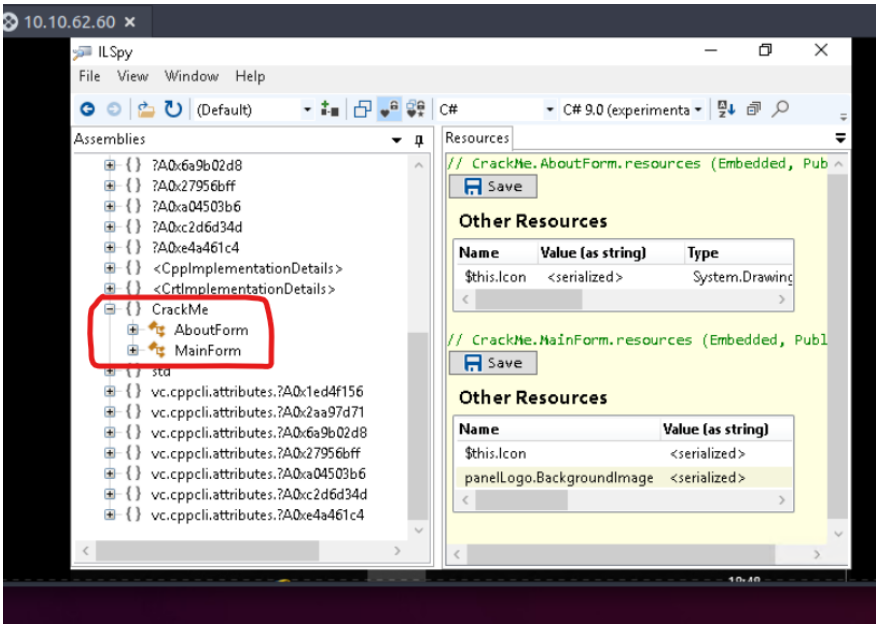
Tools Used: AttackBox, Remmina, ILSpy, TBFC, CyberChef

Question 1&2

By putting in a random password, we got the message saying "**Uh Oh! That's the wrong key**" which is the answer for question 1. For question 2, we knew what TBFC stands for by the name indicated at the bottom left corner of the TBFC Dashboard, that is **The Best Festival Company**.

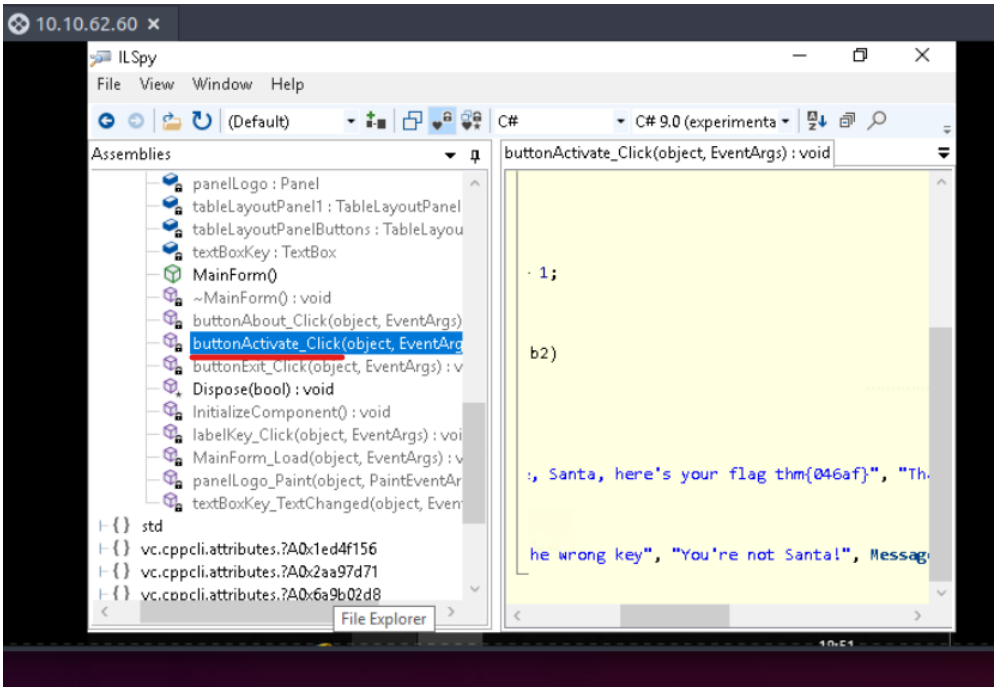
Question 3&4

The module that caught our attention was the **CrackMe** module because it is named that way. For question 2, between the forms AboutForm and MainForm, **MainForm** is the one that contains the information that we're looking for.



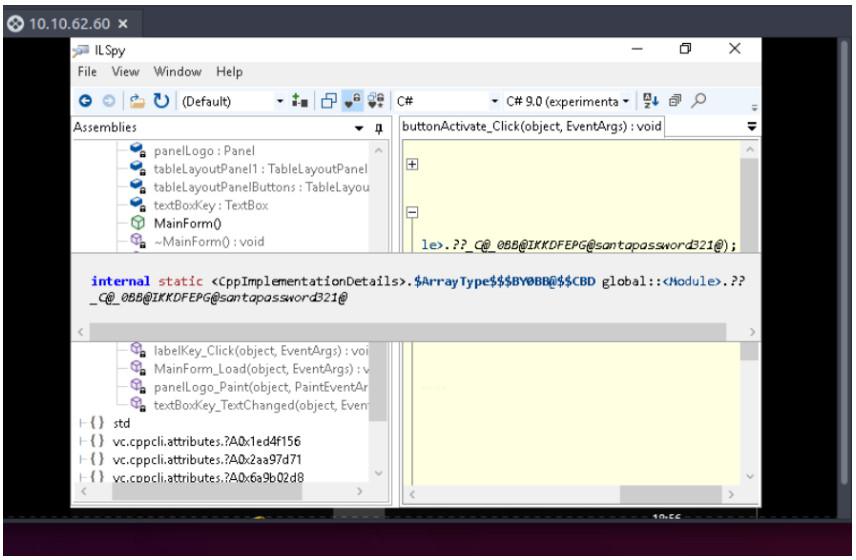
Question 5

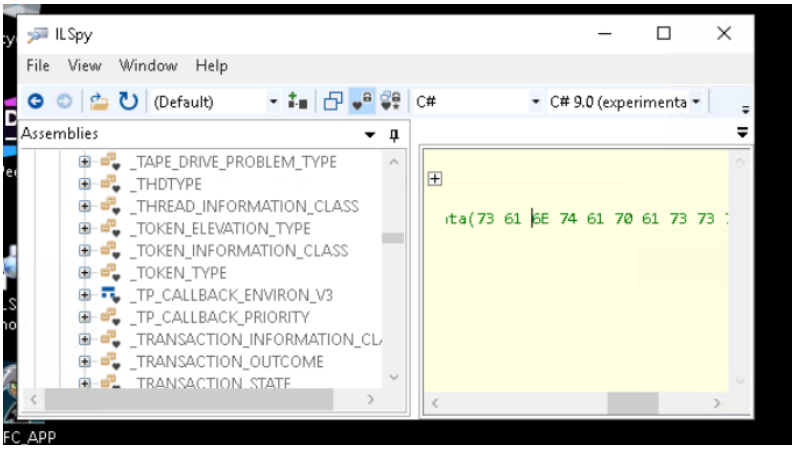
The method that was used within the MainForm is the **buttonActivate_Click** since from there, we can find not only the flag, but also Santa's password as well.

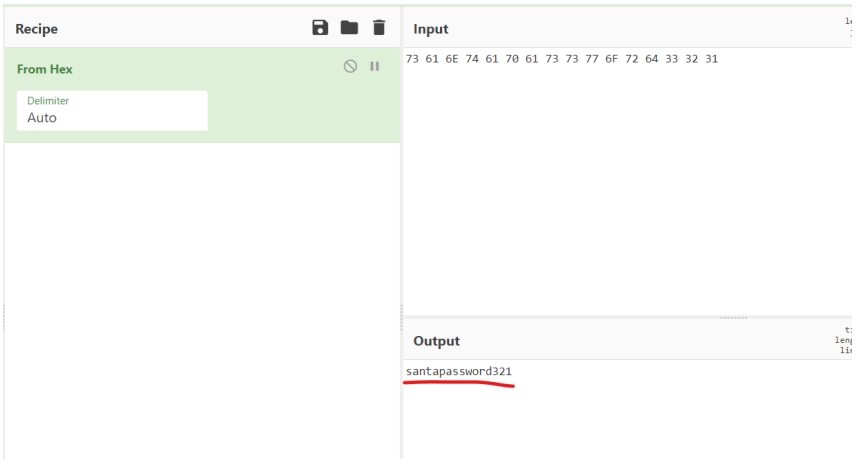


Question 6

Go to the upper part of the buttonActivate_Click and we'll see Santa's password, which is **santapasswrod321.**

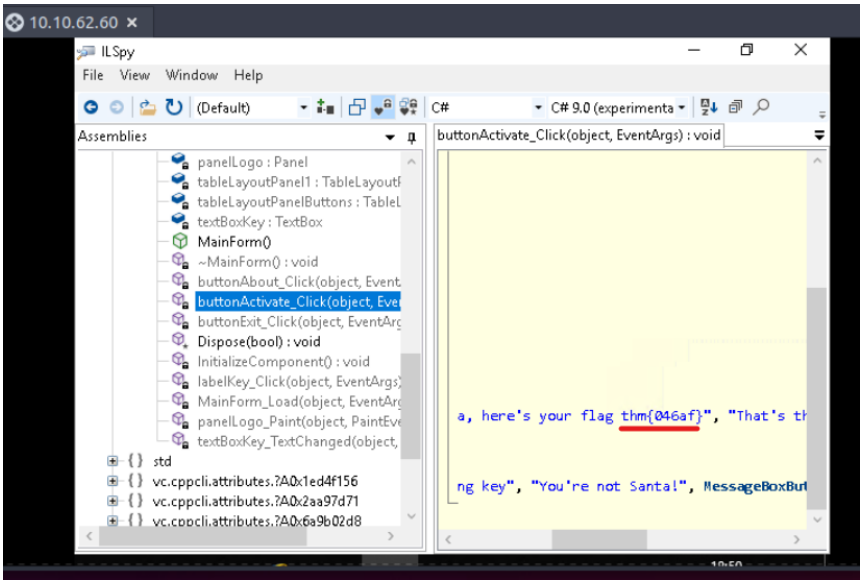






Question 7

As said before, we got the flag for day 18 in the buttonActivate_Click, that is the flag **thm{046af}**.



Thought Process/Methodology:

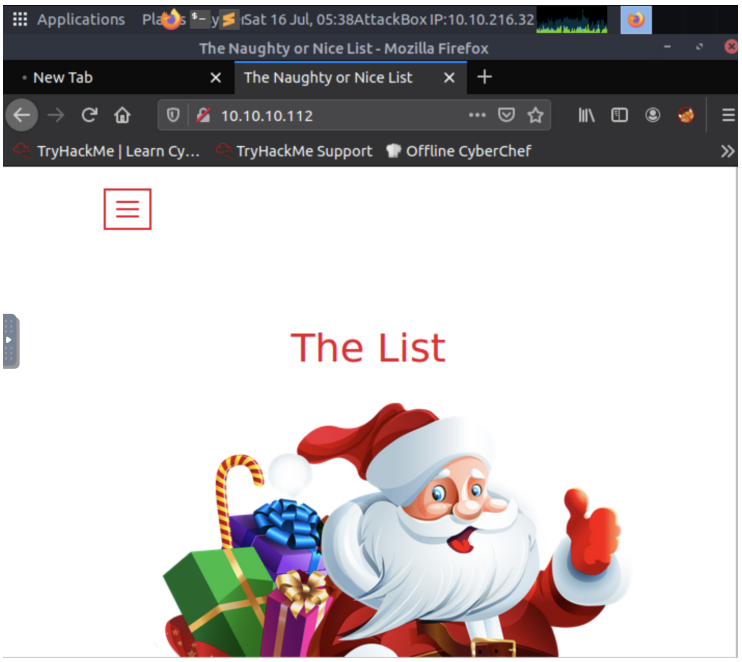
Having the AttackBox launched in TryHackMe and the machine's IP address ready, we headed to Remmina under the Internet section. There, we copy the IP address 10.10.62.60 into the Remmina Remote Desktop Client (RDP) and using the credentials given, cmnatic as the username and Adventofcyber! as the password, we were able to log in. After that, we opened the TBFC app to find the answer for question 1 and 2. Question 1, we put in a random password, in this case Mini IT best, to know what message will be displayed if an incorrect password is put. And there we got our answer. For question 2, we also got the answer by opening the app but more easily, because what TBFC stands for is already stated at the bottom left corner of the app dashboard. For question 3, after we decompiled the TBFC_APP with ILSpy, the module that caught our attention was the CrackMe module because it is suspiciously named, as if it wanted to be cracked. Question 4, between the two forms AboutForm and MainForm that exist within the module CrackMe, MainForm is the one which contains the information that we were looking for. We knew this since we checked it first before the AboutForm and close enough, we were able to find both the answers to question 6 and 7 under the method within the form named buttonActivate_Click. There we got the answer for question 5 as to which method from MainForm will contain the information we are seeking. From here, we can see in the buttonActivate_Click straight away the flag for day 18. Question 7 is done as the flag is received. As for question 6, we looked just a bit to the upper part in the buttonActivate_Click and saw Santa's password being santapassword321 but we were skeptical, so we made sure of it by double-clicking it. Then, it brought us to a place and we saw a 2-digit integer array, and knew that it was the bytes, hexadecimal at it. Knowing that, we instantly copy all of it except the 00 since it is just the null byte that terminates the string. We then pasted it into the input in CyberChef and by putting hexadecimal as the recipe and baking it, we easily achieved Santa's password, being true that the password is indeed santapassword321.

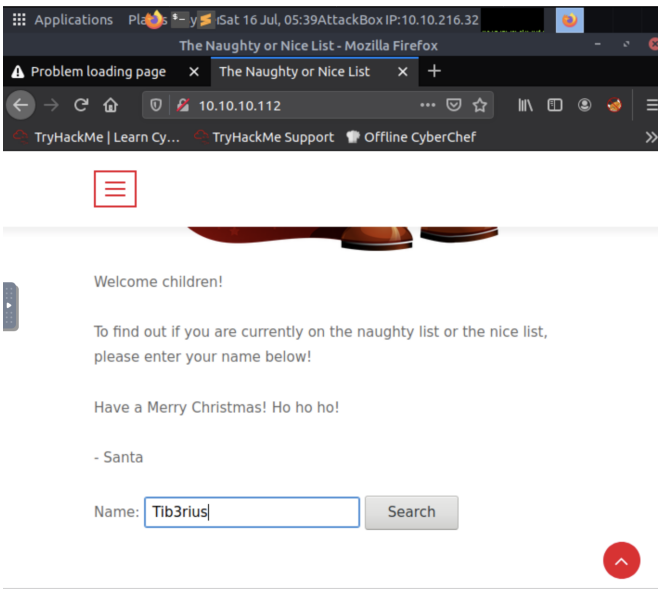
Day 19: Web Exploitation - The Naughty or Nice List

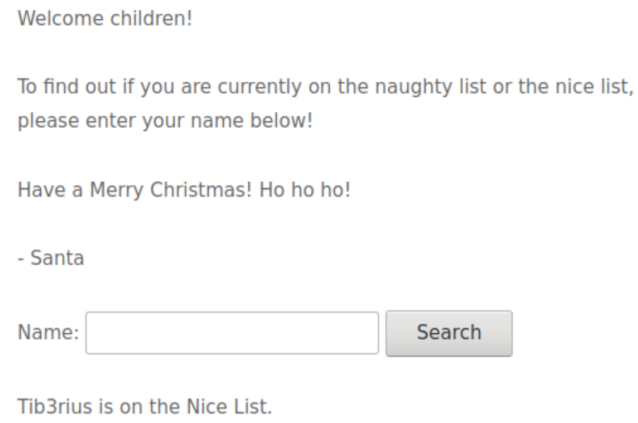
Tools Used: Firefox, THM Attackbox

Question 1

Using the IP address given we get to the website that will tell us which child has been put on the naughty or nice list.

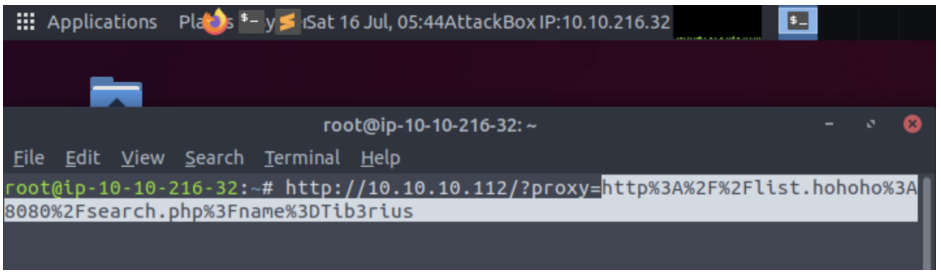




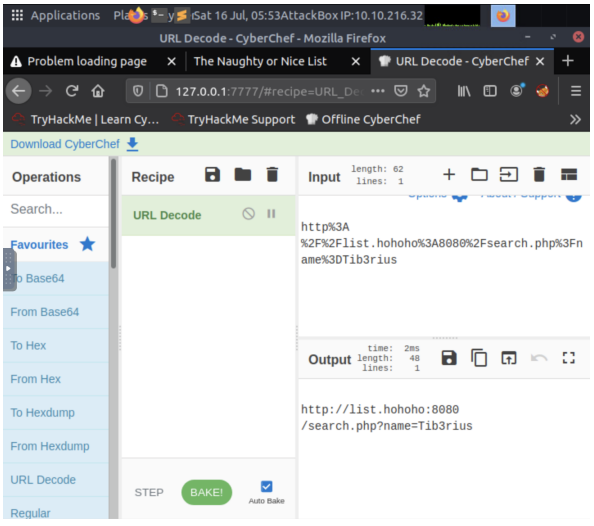


Question 2

When putting a name in the search bar the URL for the page will look like this http://10.10.10.112/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%3DTib3rius. Using CyberChef to decode URL encoded from the link

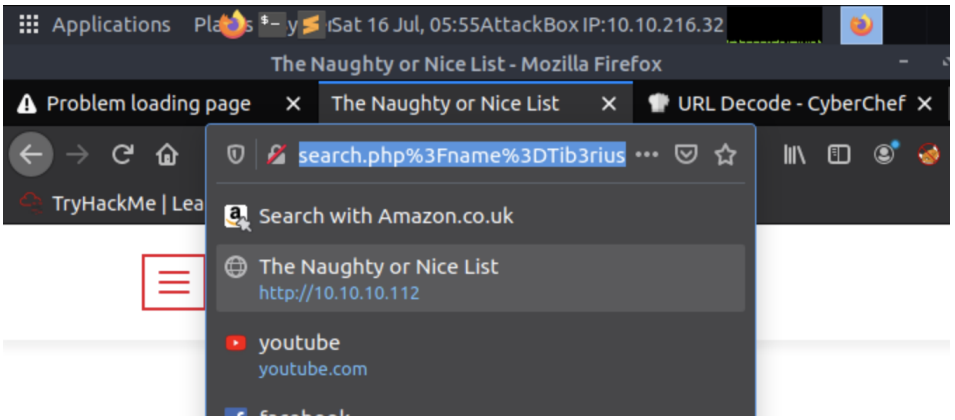


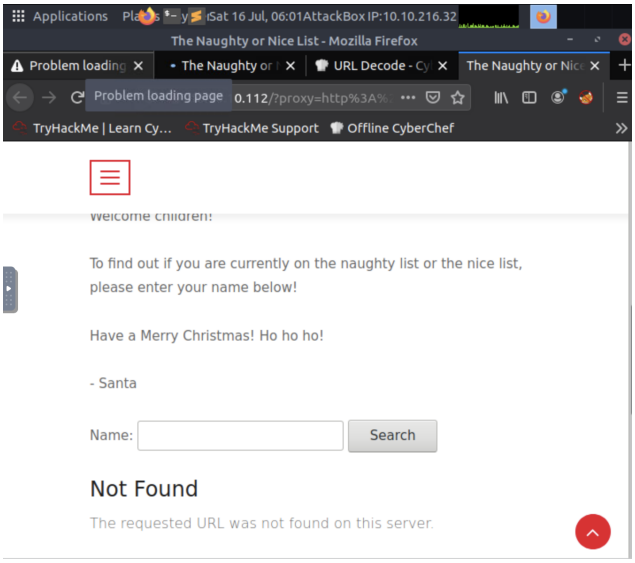
From there you will get the decoded output below. It is clear that list.hohoho is not a valid hostname and is likely making request at the back end of the machine and is displaying the request at the front end



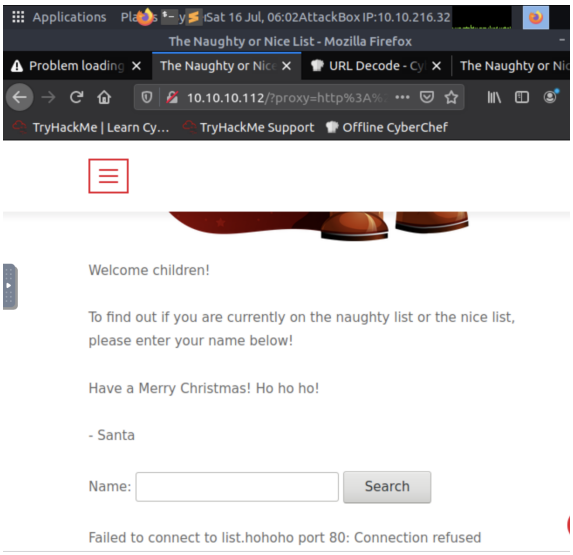
Question 4

Next thing we'll do is fetch the root of the URL. We'll see that the response of the website has changed to "Not Found. The requested URL was not found on this server."





Other things we could do is changing the port number from 8080 to 80 and the message displayed will also be changed to "Failed to connect to list.hohoho port 80: Connection refused". We also tried port number 22 and it displayed “Recv failure: Connection reset by peer” indicating that the port is open but could not entirely understand. It is also mentioned that if we were to try any other hostname that does not start with list.hohoho will automatically be blocked by their security team indicating that anything starting with list.hohoho will be allowed through. When we replaced list.hohoho from the URL with localhost only, it responded with “ Your search has been blocked by our security team."



Question 5

Using the information we gain with knowing that the hostname needs to start with list.hohoho. We can use the DNS subdomains to our advantage and what we will be using rather than buying a domain or configuring the DNS is localtest.me, which resolves every subdomain to 127.0.0.1.

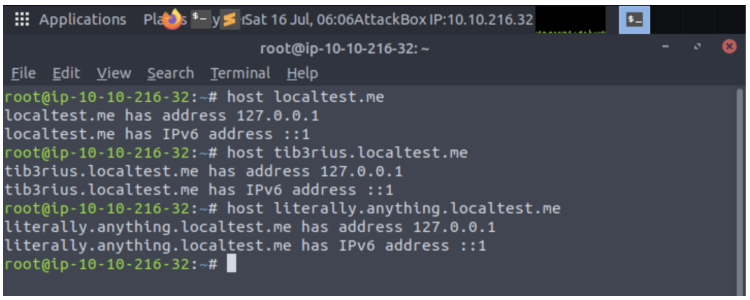
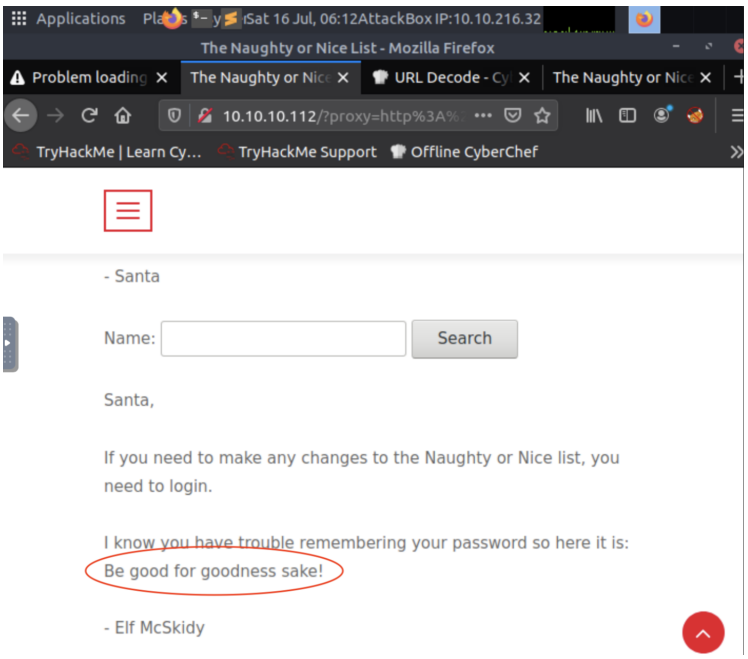


image above proves that every subdomains of localtest.me resolves to 127.0.0.1

Question 6

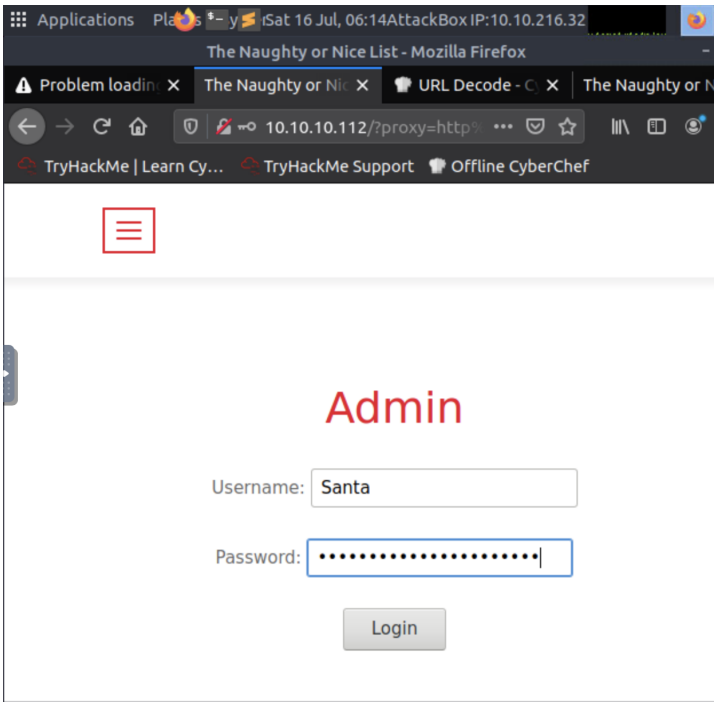
we will alter the url link adding localtest.me at the very end of the url http://10.10.10.112/?proxy=http%3A%2F%2Flist.hohoho.localtest.me. The Screen will then display a new message that seems like the elf has left in case if santa had forgotten all the information including his password



The question asking “What is santa’s password” has been answered and it is Be good for goodness sake!

Question 7

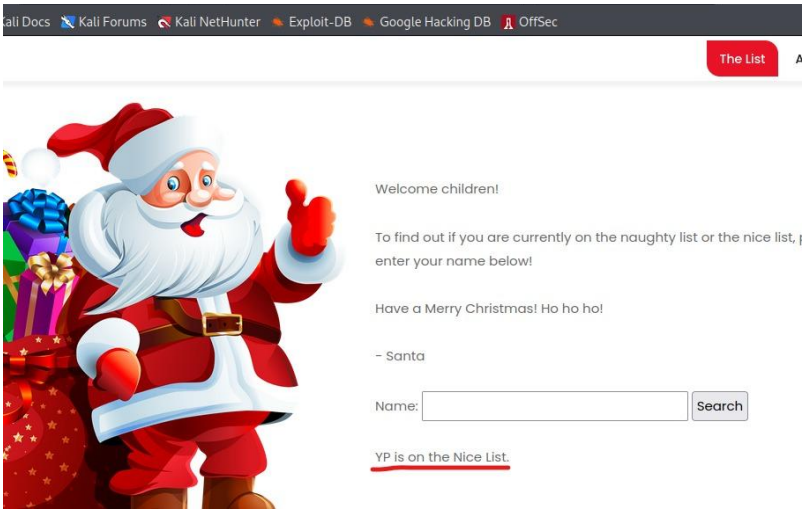
In order to fulfill our original tasks and make sure every child gets a present we will have to login as administrator to make it happen. We can login using the username that is often mentioned which is ‘santa’ as well as using the password we just discovered

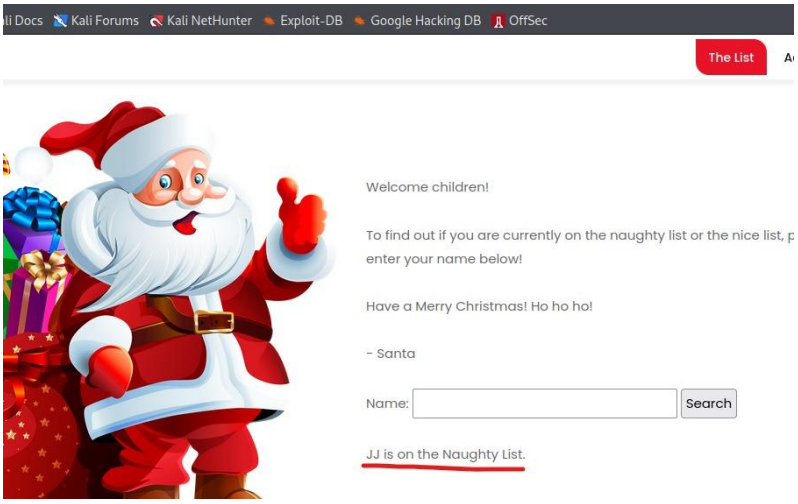


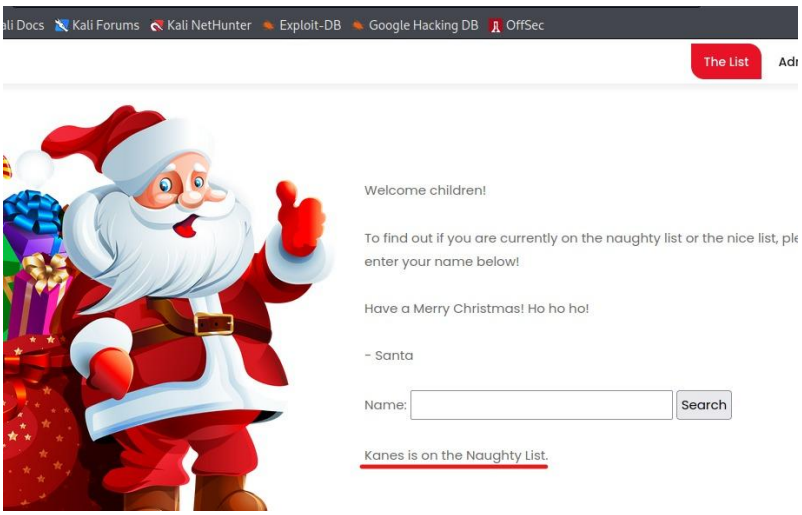
we get to the page where we are in control in whether we want to delete the naughty list or not and as click the button the flag THM{EVERYONE_GETS_PRESENTS} will pop up.

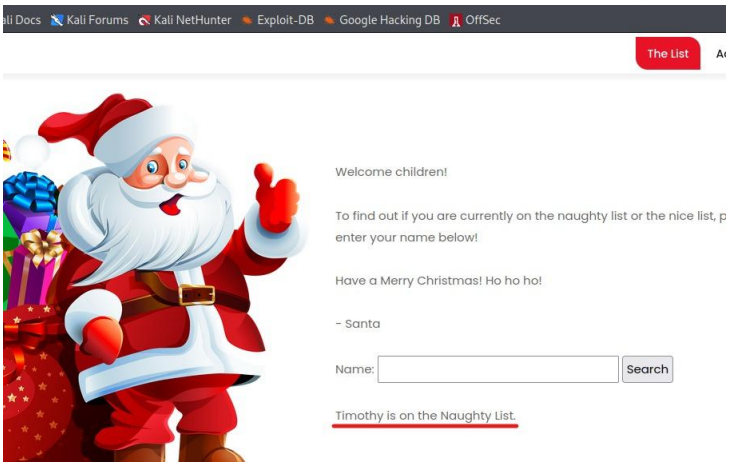
Question 8

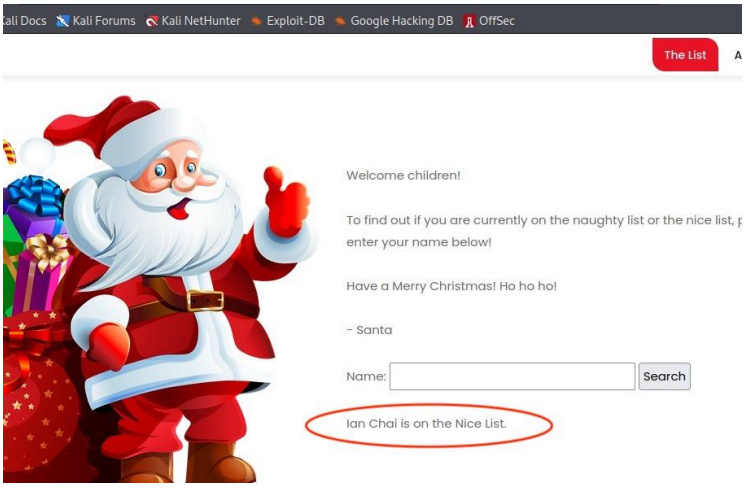
Which list is this person on?






Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
The List
Welcome children!
To find out if you are currently on the naughty list or the nice list, ple
enter your name below!
Have a Merry Christmas! Ho ho ho!
- Santa
Name:
Search
Kanes is on the Naughty List.


Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
The List
Welcome children!
To find out if you are currently on the naughty list or the nice list, p
enter your name below!
Have a Merry Christmas! Ho ho ho!
- Santa
Name:
Search
Timothy is on the Naughty List.


Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
The List
Welcome children!
To find out if you are currently on the naughty list or the nice list, p
enter your name below!
Have a Merry Christmas! Ho ho ho!
- Santa
Name:
Search
Ian Chai is on the Nice List.

Thought Process/Methodology:

From the beginning we could conclude that the original URL after using cyberchef to decode did not have a well grounded hostname. Making it easier to customize the result of the front- end web app by making a request at the back end. From numerous attempts of using other hostnames for the link and getting responses such as “The requested URL is not found in the server” or “Your search has been blocked by our security team” we can assume that the developer had implemented a check ensuring that list.hohoho must be the starting of the hostname. After acknowledging that check, we can easily bypass it by using a domain such as localtest.me that will work out in every subdomain in 127.0.0.1. After using the URL we can get a hold of Santa’s login information thus enabling us to delete the naughty list.

Day 20: Blue Teaming - PowershELIF to the rescue

Tools Used: Terminal

Question 1

The parameter -l specifies the **login name**.

-i identity_file A file from which the identity key (private key) for public key authentication is read.

-J [user@]host[:port] Connect to the target host by first making a ssh connection to the pjump host[(/iam/jump-host) and then establishing a TCP forwarding to the ultimate destination from there.

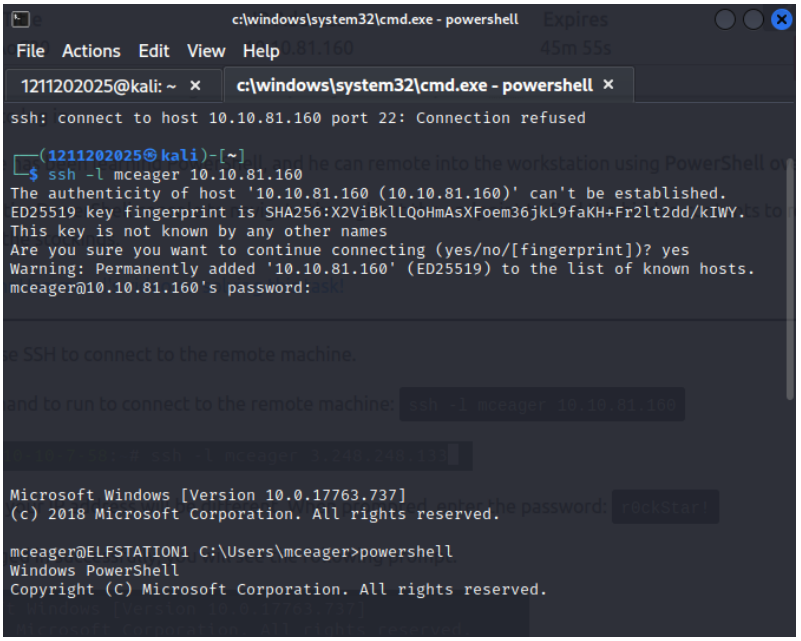
-l login_name Specifies the user to log in as on the remote machine.

-p port Port to connect to on the remote host.

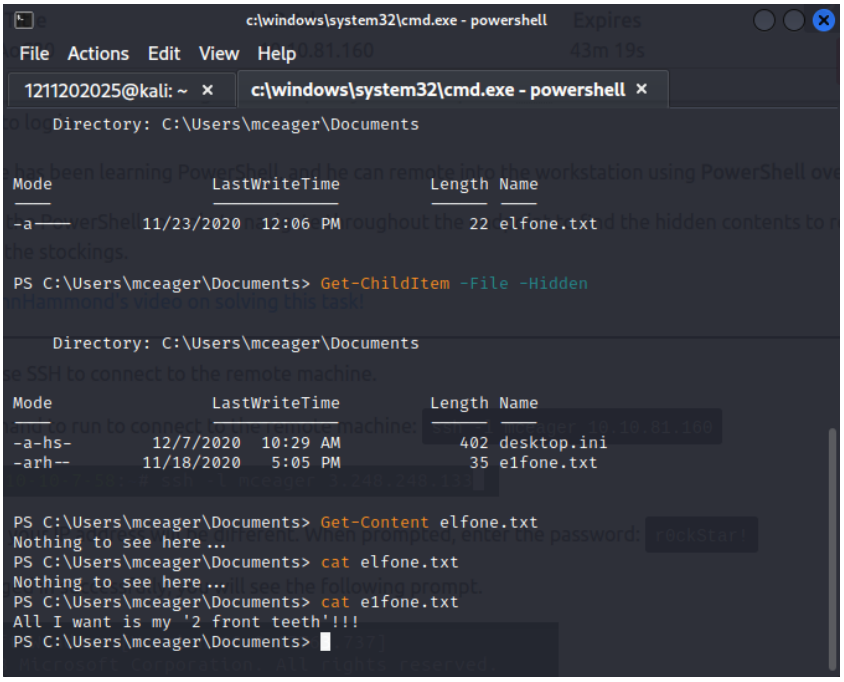
-q Quiet mode.

Question 2

After we were able to log in to the account **mceager** with the password given as **r0ckStar!**,

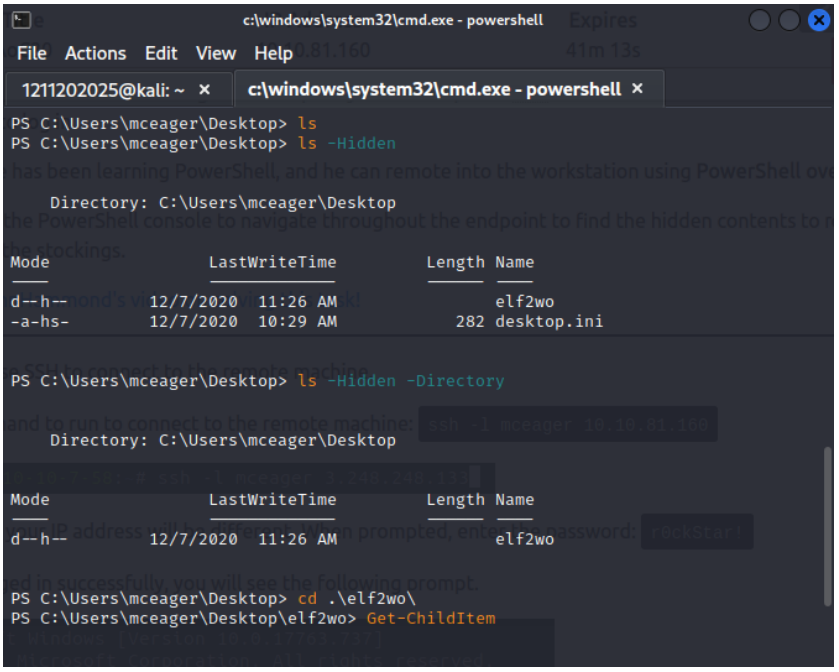


We went to the Documents directory by using the cmdlet **Set-Location .\Documents** . After getting to the Documents directory, we use the cmdlet **Get-ChildItem -File -Hidden** and are able to find the file **e1fone.txt**. Knowing the file name, we used the command **cat e1fone.txt** to print out the content of the file, and there it showed what Elf 1 wants, **2 front teeth**.

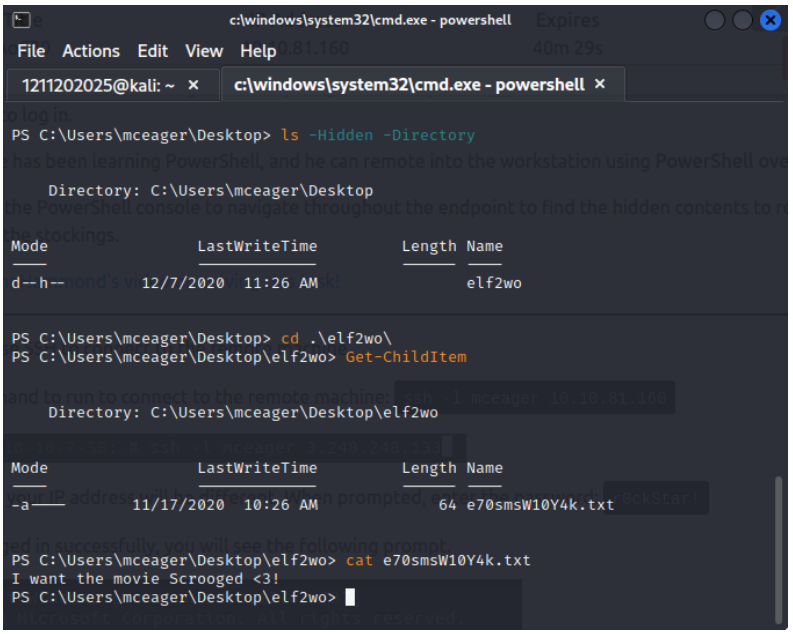


Question 3

We went to the Desktop directory and using the command **ls -Hidden**, we found that there is a directory named **elf2wo**.



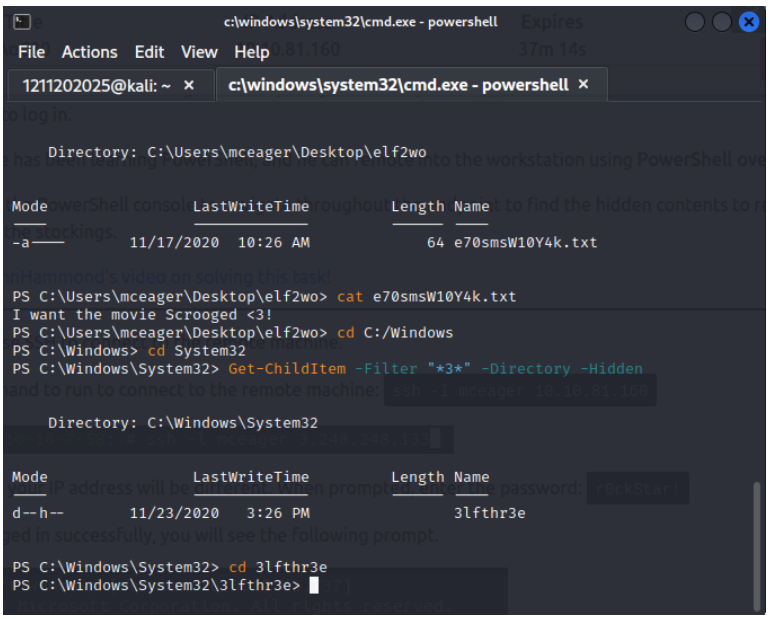
Going into the **elf2wo** directory, we used the command **Get-ChildItem** to list the contents of the directory, and the file **e70smsW10Y4k.txt** showed up.



Using the command **cat e70smsW10Y4k.txt**, we were able to obtain the name of the movie that Elf 2 wants, and that is the movie Scrooged.

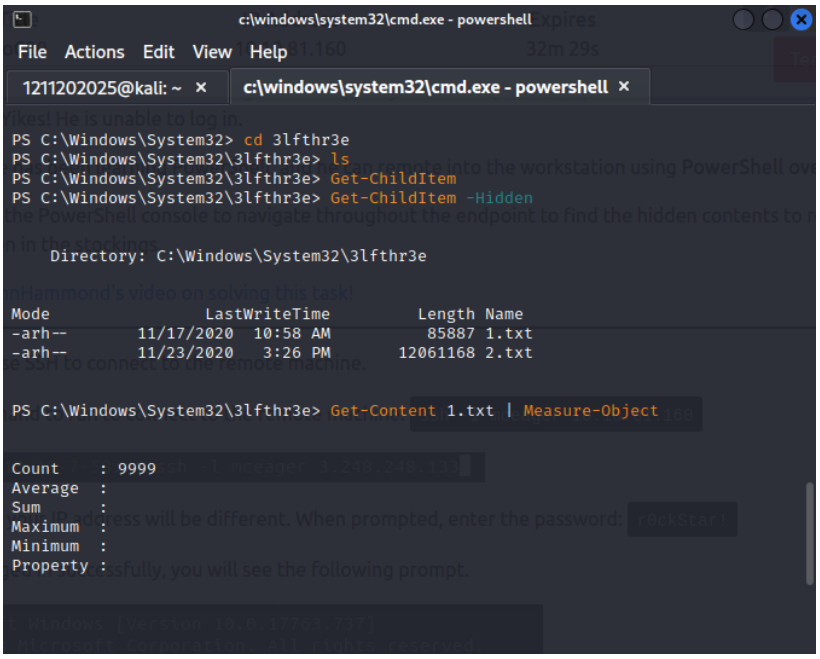
Question 4

We went back and headed into the Windows\System32 directory. There, we used the command **Get-ChildItem -Filter "*3*" -Directory -Hidden** and able to acquire the name of the hidden folder, that is **3lfthr3e**.

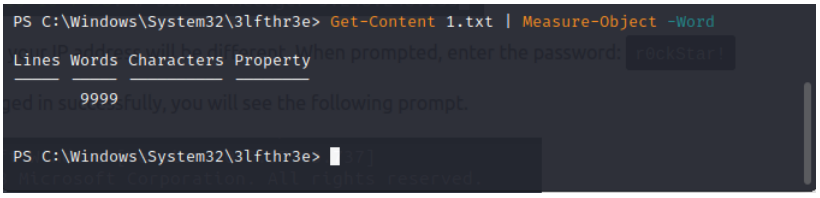


Question 5

We continued and went into the **3lfthr3e** folder, and used the command **Get-ChildItem -Hidden**. Knowing the files name and what the question wanted, which is the first file, we used the command **Get-Content 1.txt | Measure-Object** and found the word count for the first file, which is **9999**.

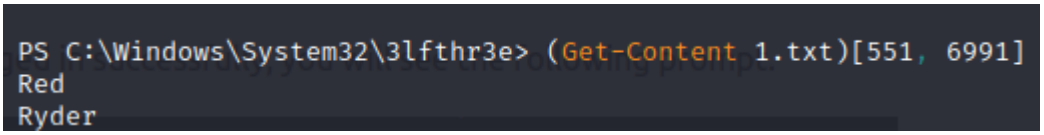


But we hesitated, since it only stated there "Count" and not word count, so we rechecked it by adding in the command **-Word** after the **Measure-Object**, and from there, we confirmed that the **9999** was indeed the word count.



Question 6

To know what 2 words are at index 551 and 6991 in the first file, we used the command **(Get-Content 1.txt)[551, 6991]** and were able to get the answers right away, that is Red at 551 and Ryder at 6991.



Question 7

Finding for the full answer by searching in the second file and using the hint from TryHackMe, we used the command **Get-Content 2.txt | Select-String -Pattern "redryder"**, we were able to successfully obtain the full answer as to what Elf 3 wants, which is **redryderbbgun**.

```
PS C:\Windows\System32\3lfthr3e> Get-Content 2.txt | Select-String -Pattern "redryder"

redryderbbgun
```

Thought Process/Methodology:

After obtaining the machine's IP address, we used the command ssh -l mceager 10.10.81.160 to log into the account of mceager, and also by using the password given r0ckStar! . We were able to successfully log in but before that for question 1, we were able to know that the parameter -l specify the login name by doing some research. From here onwards, using all the information that was given to us by TryHackMe, for example the command Get-ChildItem, -Hidden, -File, etc. we use it to our advantage for all the remaining questions. Question 2, we went to the Documents directory by using the command Set-Location .\Documents\ . From there, we were able to obtain what Elf 1 wants by using the command Get-ChildItem -File -Hidden and are able to find the file e1fone.txt. We used the command cat e1fone.txt to get the contents inside the file, and it contains what they want, their 2 front teeth. Question 3, we went to the Desktop directory and using the command -ls - Hidden, we found that there is a directory named elf2wo. We confirmed it to be a directory by adding the command -Directory and it still showed up, indicating that it is a directory. A file named file e70smsW10Y4k.txt showed up after we used the command Get-ChildItem. After that, using the command cat e70smsW10Y4k.txt, we get to know what Elf 2 wants, which is the movie Scrooged. Question 4, we went to Windows directory and used the same command to search for the hidden folder, but this time we added the command -Filter "*3*" -Directory -Hidden to let the output show hidden directories that only have the number 3 in them, that way it is easier to find the Elf 3 file since the numbers are what makes them different from each other. True enough, we were able to find the folder, 3lfthr3e. Question 5, we used a command and able to locate both the first file and the second file, 1.txt and 2.txt, and using the command Get-Content 1.txt | Measure-Object -Word to find the word count for the first file, we got the answer stated as 9999, meaning there are 9999 words in the first file. Question 6, using the command (Get-Content 1.txt)[551, 6991], we were able to find the words at the exact index, and that is Red and Ryder. Question 7, searching for the full answer in file 2, we used the command Get-Content 2.txt | Select-String -Pattern "redryder" to find a particular file with the word redryder in it, and fair enough, we were able to get the full answer, that is redryderbbgun.