



PSP0201

Week 5

Write-up

Group Name: **PennCake**

ID	Name	Role
1211103144	Vaarindran Nyenasegran	Leader
1211103222	Asyran Syazwan Yuhanis	Member
1211104230	Nur Aisyah Nabilah Nahar	Member
1211101169	Tengku Alyssa Sabrina Tengku Erwin Martino	Member

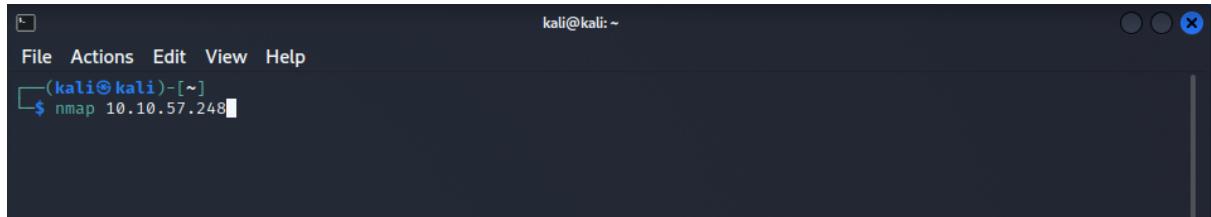
Day 16: Scripting - Help! Where is Santa?

Tools used: Kali Linux, Terminal, Mozilla Firefox

Solution/walkthrough:

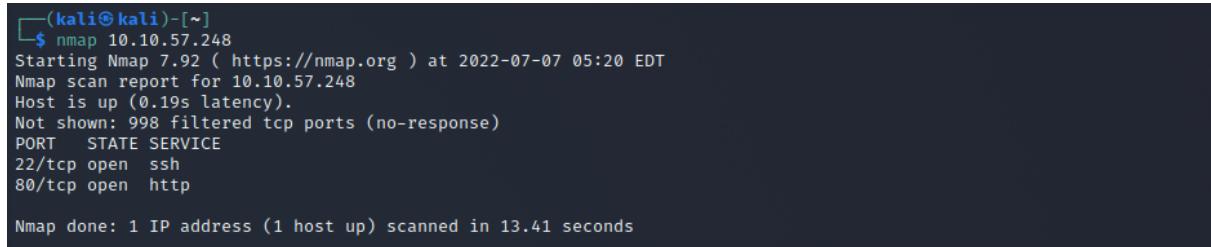
Question 1: What is the port number for the web server?

Firstly, we need to set up nmap for the target IP Address. We can set it up by using the following command.



A screenshot of a terminal window titled "kali@kali: ~". The window has a dark background with light-colored text. At the top, there's a menu bar with "File", "Actions", "Edit", "View", and "Help". Below the menu is a command line interface. The prompt shows "(kali㉿kali)-[~]" followed by the command "\$ nmap 10.10.57.248". The rest of the window is mostly blank, indicating the scan is still in progress.

Wait for a bit and the port will be shown to us. From there, we can see that port **80** is for a **http** service.



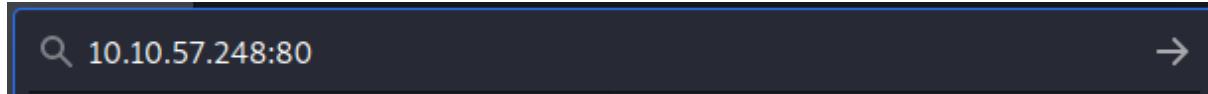
```
(kali㉿kali)-[~]
$ nmap 10.10.57.248
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-07 05:20 EDT
Nmap scan report for 10.10.57.248
Host is up (0.19s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 13.41 seconds
```

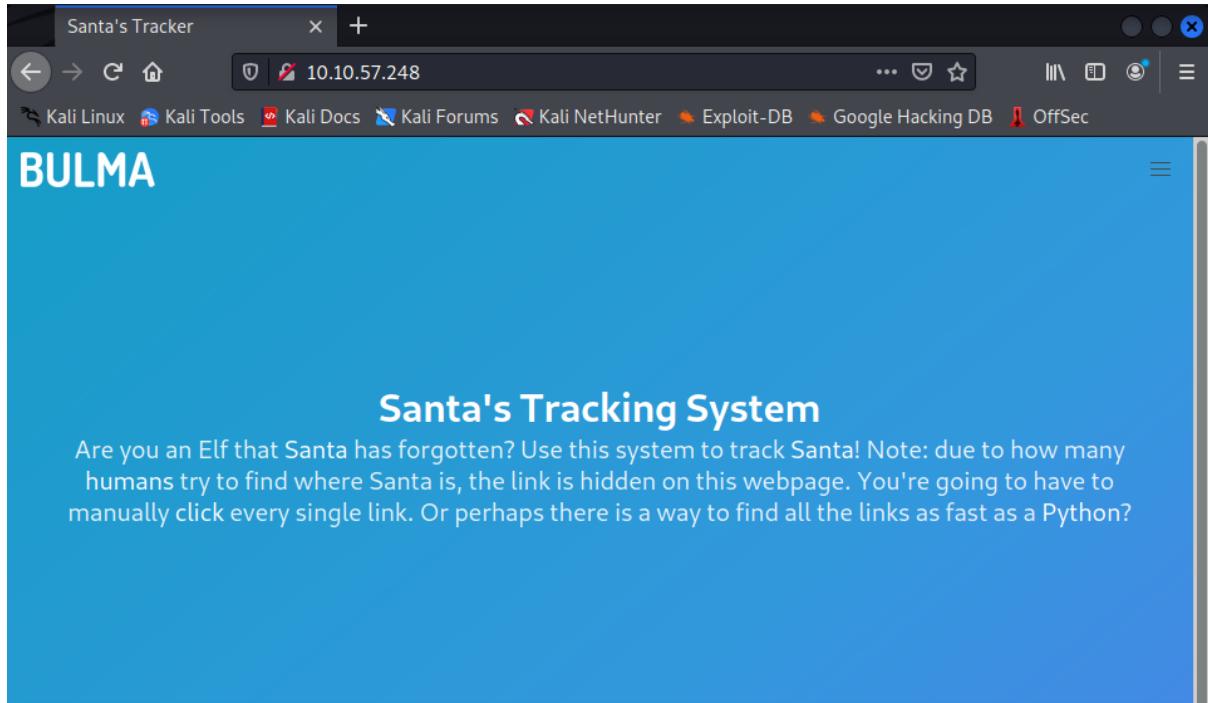
Answer: 80

Question 2: What templates are being used?

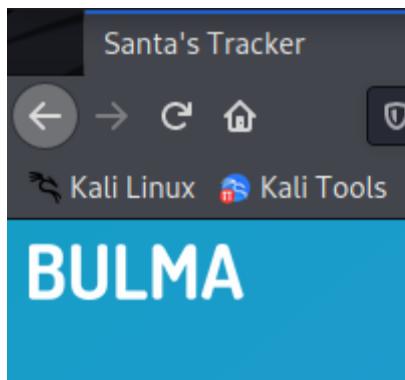
Enter the IP address with the port we want to use to access the website that we want to go to.



We will be brought to a website called "Santa's Tracker".



On the corner of the website, the template that it used is **BULMA**.



Answer: BULMA

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

From the Python script from Day 15, we can reuse it and help us find the directory.

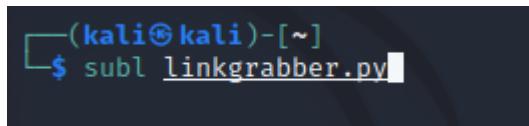
```
# Import the libraries we downloaded earlier
# if you try importing without installing them, this step will fail
from bs4 import BeautifulSoup
import requests

# replace testurl.com with the url you want to use.
# requests.get downloads the webpage and stores it as a variable
html = requests.get('testurl.com')

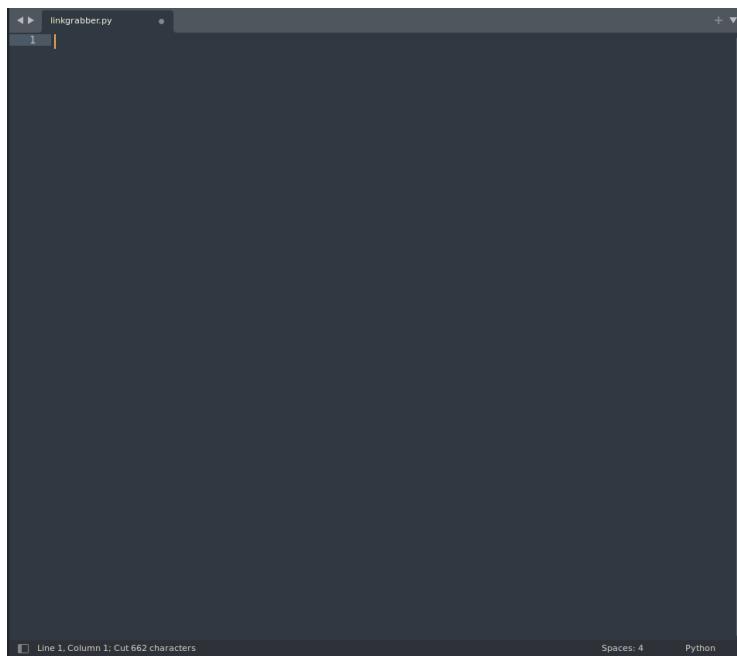
# this parses the webpage into something that beautifulsoup can read over
soup = BeautifulSoup(html, "lxml")
# lxml is just the parser for reading the html

# this is the line that grabs all the links # stores all the links in the links variable
links = soup.find_all('a href')
for link in links:
    # prints each link
    print(link)
```

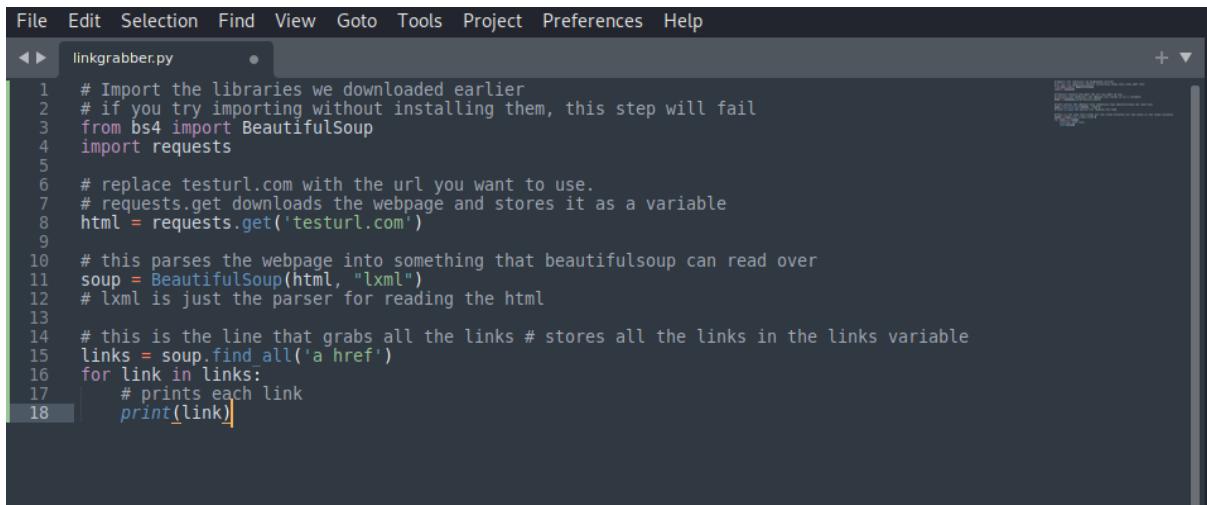
We need to paste the script into a sublime text editor. So we can use the command **subl** and the name, we also need to make sure it's a Python file by putting **.py** in the end of the name.



It will open a sublime text editor tab.

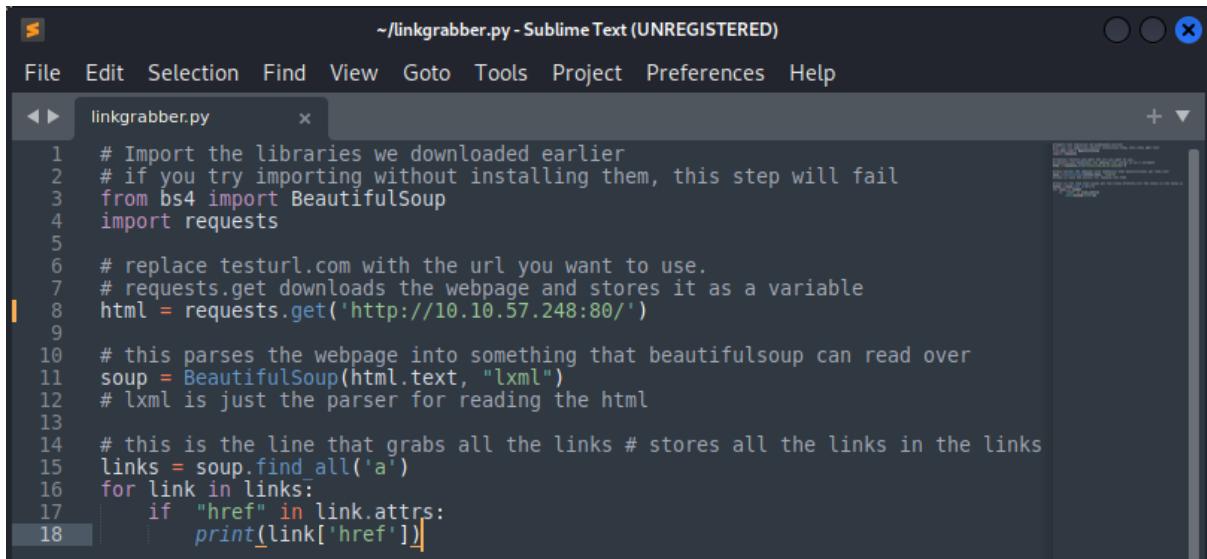


Then, we can paste the Python script into it.



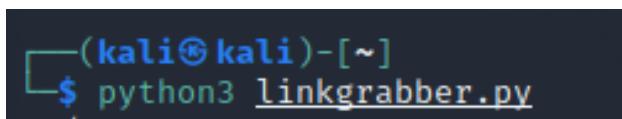
```
File Edit Selection Find View Goto Tools Project Preferences Help
linkgrabber.py •
1 # Import the libraries we downloaded earlier
2 # if you try importing without installing them, this step will fail
3 from bs4 import BeautifulSoup
4 import requests
5
6 # replace testurl.com with the url you want to use.
7 # requests.get downloads the webpage and stores it as a variable
8 html = requests.get('testurl.com')
9
10 # this parses the webpage into something that beautifulsoup can read over
11 soup = BeautifulSoup(html, "lxml")
12 # lxml is just the parser for reading the html
13
14 # this is the line that grabs all the links # stores all the links in the links variable
15 links = soup.find_all('a href')
16 for link in links:
17     # prints each link
18     print(link)
```

Now, we need to change the script a bit so that it can find the link for us.



```
File Edit Selection Find View Goto Tools Project Preferences Help
linkgrabber.py •
1 # Import the libraries we downloaded earlier
2 # if you try importing without installing them, this step will fail
3 from bs4 import BeautifulSoup
4 import requests
5
6 # replace testurl.com with the url you want to use.
7 # requests.get downloads the webpage and stores it as a variable
8 html = requests.get('http://10.10.57.248:80/')
9
10 # this parses the webpage into something that beautifulsoup can read over
11 soup = BeautifulSoup(html.text, "lxml")
12 # lxml is just the parser for reading the html
13
14 # this is the line that grabs all the links # stores all the links in the links
15 links = soup.find_all('a')
16 for link in links:
17     if "href" in link.attrs:
18         print(link['href'])
```

Next, we need to go to the terminal to execute the script. We can execute it by using the command **python3 name.py**.



```
└─(kali㉿kali)-[~]
$ python3 linkgrabber.py
```

Thus, it will run and list out all the links from the Santa's Tracker.

There, we can find the link we want to obtain. We can see that the directory of the API is `/api/`.

```
#  
http://machine_ip/api/api_key  
#
```

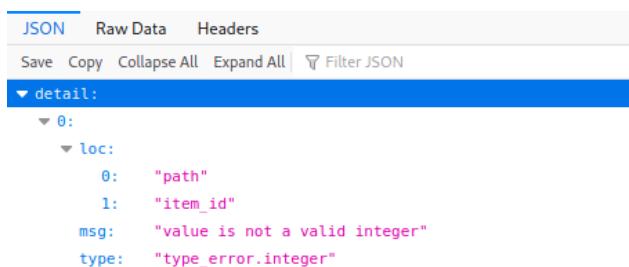
Answer: /api/

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

From the link, we can use it to find the raw data.

```
Q 10.10.57.248/api/api_key
```

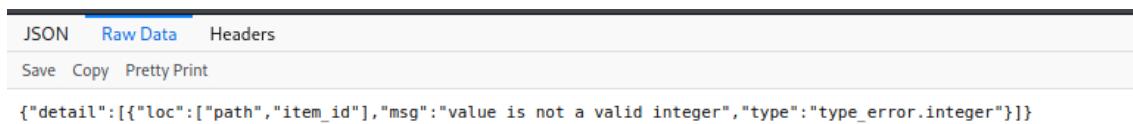
The link will lead to a page that lists out the JSON and Raw Data.



A screenshot of a JSON viewer interface. At the top, there are tabs for "JSON", "Raw Data", and "Headers". Below the tabs are buttons for "Save", "Copy", "Collapse All", "Expand All", and "Filter JSON". The main area shows a single object under the "detail" key. The "detail" key has one child object, "0", which has a child "loc" object. The "loc" object contains three keys: "0" (value "path"), "1" (value "item_id"), "msg" (value "value is not a valid integer"), and "type" (value "type_error.integer").

```
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
▼ detail:
  ▼ 0:
    ▼ loc:
      0: "path"
      1: "item_id"
      msg: "value is not a valid integer"
      type: "type_error.integer"
```

Go to the **Raw Data** tab to get the answer.



A screenshot of a JSON viewer interface. At the top, there are tabs for "JSON", "Raw Data", and "Headers". Below the tabs are buttons for "Save", "Copy", and "Pretty Print". The "Raw Data" tab is selected. The main area displays the following JSON code:

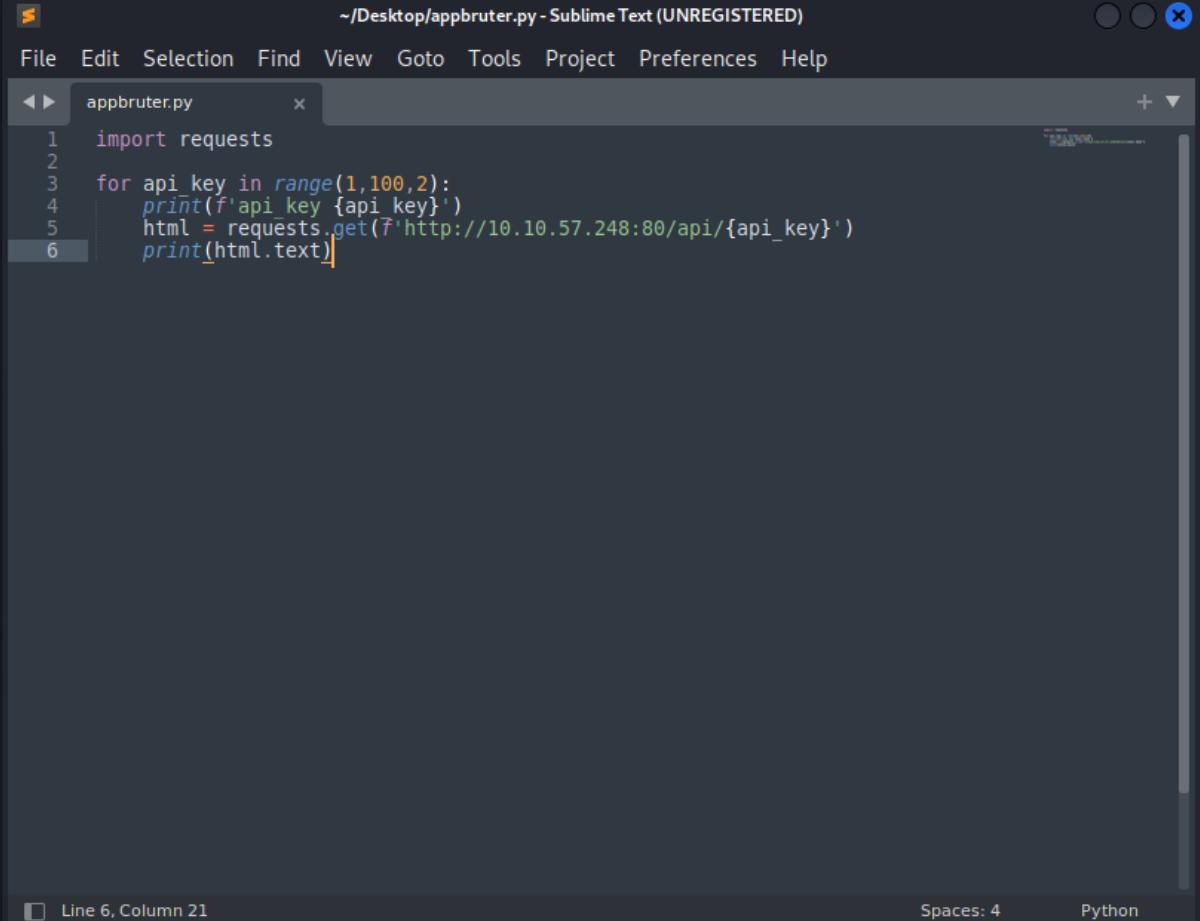
```
{"detail": [{"loc": ["path", "item_id"], "msg": "value is not a valid integer", "type": "type_error.integer"}]}
```

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

Question 5: Where is Santa right now?

To find Santa's location, we need to make a Python programme that can detect the location of Santa.

We need to get another Sublime Text Editor. We need to have a script that continuously repeats itself with a new number until it finds Santa's location.

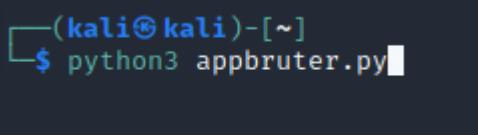


The screenshot shows a Sublime Text window with a dark theme. The title bar reads "S ~/Desktop/appbruter.py - Sublime Text (UNREGISTERED)". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. A tab bar at the top has "appbruter.py" selected. The code editor contains the following Python script:

```
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://10.10.57.248:80/api/{api_key}')
6     print(html.text)
```

At the bottom left, there is a status bar with "Line 6, Column 21". At the bottom right, it says "Spaces: 4" and "Python".

Now, we can run it by using the command **python3 name.py**.



The screenshot shows a terminal window with a black background and white text. The prompt is "(kali㉿kali)-[~]". The user has typed the command "\$ python3 appbruter.py" and is awaiting the output.

When it is run, it will repeat itself with a new number until it finds the location.

```
(kali㉿kali)-[~]
$ python3 appbruter.py
api_key 1
{"item_id":1,"q":"Error. Key not valid!"}
api_key 3
{"item_id":3,"q":"Error. Key not valid!"}
api_key 5
{"item_id":5,"q":"Error. Key not valid!"}
api_key 7
{"item_id":7,"q":"Error. Key not valid!"}
api_key 9
{"item_id":9,"q":"Error. Key not valid!"}
api_key 11
{"item_id":11,"q":"Error. Key not valid!"}
api_key 13
{"item_id":13,"q":"Error. Key not valid!"}
api_key 15
{"item_id":15,"q":"Error. Key not valid!"}
api_key 17
{"item_id":17,"q":"Error. Key not valid!"}
api_key 19
{"item_id":19,"q":"Error. Key not valid!"}
api_key 21
{"item_id":21,"q":"Error. Key not valid!"}
api_key 23
{"item_id":23,"q":"Error. Key not valid!"}
api_key 25
{"item_id":25,"q":"Error. Key not valid!"}
api_key 27
{"item_id":27,"q":"Error. Key not valid!"}
api_key 29
{"item_id":29,"q":"Error. Key not valid!"}
```

Then, when the correct number is keyed in, it will list Santa's location.

```
api_key 53
{"item_id":53,"q":"Error. Key not valid!"}
api_key 55
{"item_id":55,"q":"Error. Key not valid!"}
api_key 57
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
api_key 59
{"item_id":59,"q":"Error. Key not valid!"}
api_key 61
{"item_id":61,"q":"Error. Key not valid!"}
api_key 63
```

Answer: Winter Wonderland, Hyde Park, London

Question 6: 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

From the last question, the correct API Key is **57**.

```
api_key 53
{"item_id":53,"q":"Error. Key not valid!"}
api_key 55
{"item_id":55,"q":"Error. Key not valid!"}
api_key 57
{"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
api_key 59
{"item_id":59,"q":"Error. Key not valid!"}
api_key 61
{"item_id":61,"q":"Error. Key not valid!"}
api_key 63
```

Answer: 57

Thought Process/Methodology:

With the given IP Address, we can find the port that is open by using **nmap**. There, only 2 ports are open and one of them has a service of **http**. Now, we know the correct port is **80**. Hence, we use the port along with the IP Address to access the website which is **Santa's Tracker**. The first thing we can see in the website is the template used which is **BULMA**. Next, we need to find the directory of the API. We can easily go to “**View Page Source**” and find the link that leads to the directory. However, we can also use Python to obtain the directory. Firstly, we can use the script that has been given to us in the TryHackMe website on **Day 15**. Secondly, we need to paste the script in a **Sublime Text Editor** and make it a **Python file** by using the **.py** on the end of the name of the file. Next, we need to modify the script a bit to have a programme that can help us filter out all the links that are in the website. Now, we can run the Python script by using the command **python3 name.py**. It will filter out the links and we can identify the correct link that has the directory we need which is **/api/**. From the link we obtained from the Python script, we can also obtain the **Raw Data**. Use the link with the IP Address that TryHackMe has provided. There, we can obtain the **Raw Data** which is **{"detail": [{"loc": ["path", "item_id"], "msg": "value is not a valid integer", "type": "type_error.integer"}]}**. After that, we need to find **Santa's location**. We can find the location by using **Python**. We need to make a Python script that can continuously repeat itself until it finds the correct API key and lists out the location. We can do this by making a loop. After finishing the script, run it and it will continuously repeat the programme until it finds the correct API key. The correct API key is **57** and Santa's location is **Winter Wonderland, Hyde Park, London**.

Day 17: Reverse Engineering - ReverseELFneering

Tools used: AttackBox, Terminal

Solution/walkthrough:

Question 1: Match the data type with size of bytes

Answer :

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?

Echo “10.10.78.161” > target.txt command was used to display lines of text or string which are passed as arguments on the command line. Then, **cat target.txt** command and **ssh elfmceager@10.10.78.161** was used.

The screenshot shows a terminal window titled "AttackBoxIP:10.10.26.192". The terminal session starts with the user root@ip-10-10-26-192. The user runs "echo '10.10.78.161' > target.txt" and "cat target.txt", both of which output "10.10.78.161". Then, the user runs "ssh elfmceager@10.10.78.161", which prompts for confirmation due to a new host fingerprint. The user types "yes" and the connection is established.

```
File Edit View Search Terminal Help
root@ip-10-10-26-192:~# echo "10.10.78.161" > target.txt
root@ip-10-10-26-192:~# cat target.txt
10.10.78.161
root@ip-10-10-26-192:~# ssh elfmceager@10.10.78.161
The authenticity of host '10.10.78.161 (10.10.78.161)' can't be established.
ECDSA key fingerprint is SHA256:XrBuSQs0wRKhvVrdrSfE/0F5ccAZQiXAhMhzB1dV7U.
Are you sure you want to continue connecting (yes/no)? yes
```

The screenshot shows a terminal window titled "AttackBoxIP:10.10.26.192". The user is now connected as root at the host "elfmceager@tbfc-day-17". The session starts with the user running "echo '10.10.78.161' > target.txt" and "cat target.txt", both of which output "10.10.78.161". Then, the user runs "ssh elfmceager@10.10.78.161", which prompts for confirmation due to a new host fingerprint. The user types "yes" and the connection is established. The terminal then displays the Ubuntu 18.04.5 LTS welcome message and system information. It shows 0 packages and updates available, and the last login was on Wednesday, December 16, 2020.

```
File Edit View Search Terminal Help
elfmceager@tbfc-day-17:~#
root@ip-10-10-26-192:~# echo "10.10.78.161" > target.txt
root@ip-10-10-26-192:~# cat target.txt
10.10.78.161
root@ip-10-10-26-192:~# ssh elfmceager@10.10.78.161
The authenticity of host '10.10.78.161 (10.10.78.161)' can't be established.
ECDSA key fingerprint is SHA256:XrBuSQs0wRKhvVrdrSfE/0F5ccAZQiXAhMhzB1dV7U.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.78.161' (ECDSA) to the list of known hosts.
elfmceager@10.10.78.161's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-128-generic x86_64)

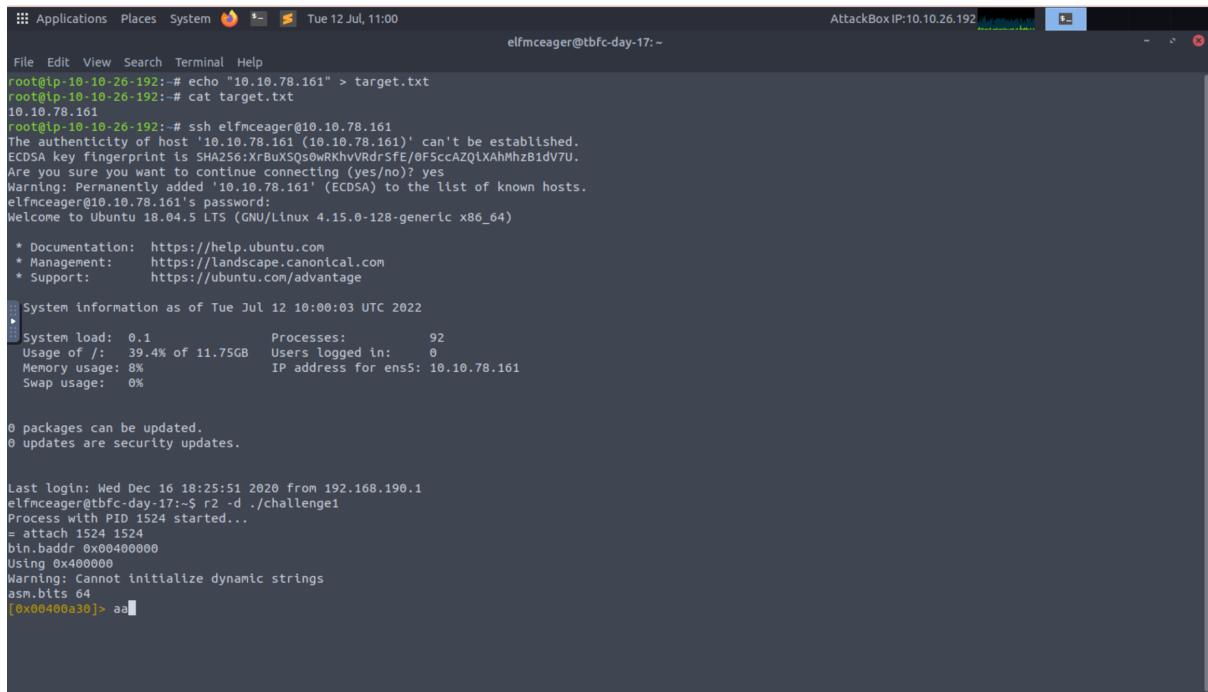
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Tue Jul 12 10:00:03 UTC 2022
System load: 0.1      Processes:         92
Usage of /: 39.4% of 11.75GB  Users logged in:     0
Memory usage: 8%
Swap usage:  0%

0 packages can be updated.
0 updates are security updates.

Last login: Wed Dec 16 18:25:51 2020 from 192.168.190.1
elfmceager@tbfc-day-17:~$
```

r2 -d ./challenge1 was used to open the binary in debugging mode.



```
File Edit View Search Terminal Help
root@ip-10-10-26-192:~# echo "10.10.78.161" > target.txt
root@ip-10-10-26-192:~# cat target.txt
10.10.78.161
root@ip-10-10-26-192:~# ssh elfmceager@10.10.78.161
The authenticity of host '10.10.78.161 (10.10.78.161)' can't be established.
ECDSA key fingerprint is SHA256:XRBUxSqs0wRKhvVrdrSF/E/0F5ccAZQlXAhMhzB1dV7U.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.78.161' (ECDSA) to the list of known hosts.
elfmceager@10.10.78.161's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-128-generic x86_64)

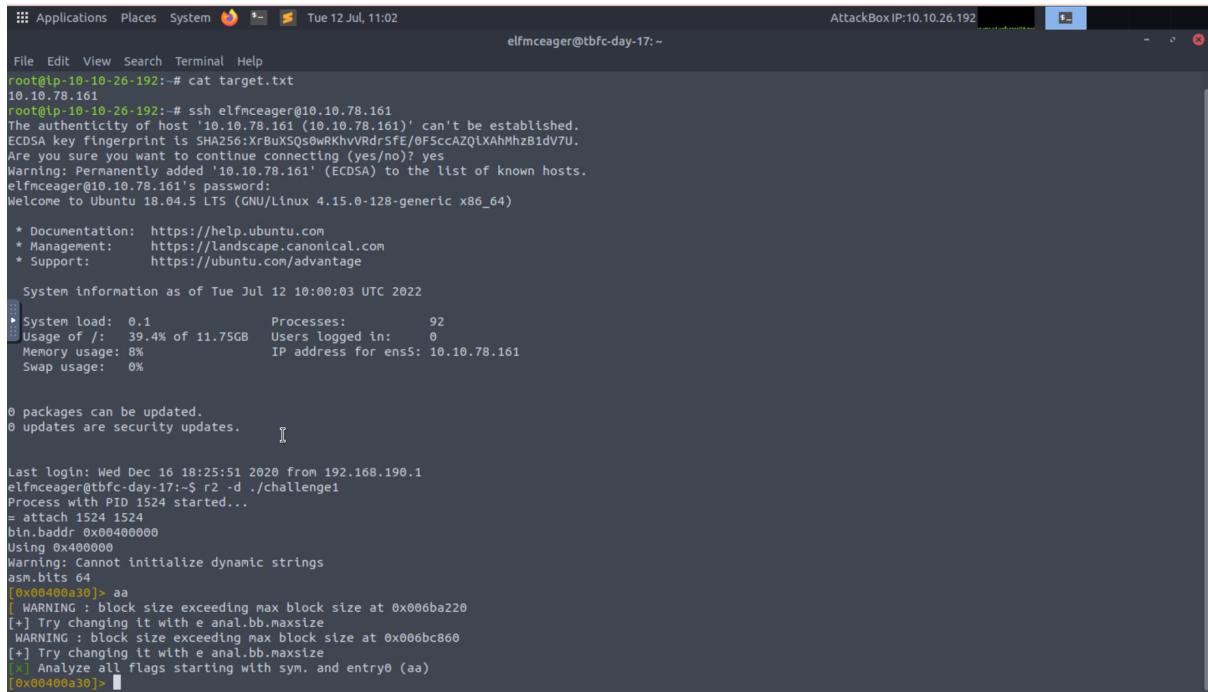
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Tue Jul 12 10:00:03 UTC 2022
System load: 0.1 Processes: 92
Usage of /: 39.4% of 11.75GB Users logged in: 0
Memory usage: 8% IP address for ens5: 10.10.78.161
Swap usage: 0%

0 packages can be updated.
0 updates are security updates.

Last login: Wed Dec 16 18:25:51 2020 from 192.168.190.1
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1524 started...
= attach 1524 1524
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
asm.bits 64
[0x00400a30]> aa
```

aa command was used to analyze the program.



```
File Edit View Search Terminal Help
root@ip-10-10-26-192:~# cat target.txt
10.10.78.161
root@ip-10-10-26-192:~# ssh elfmceager@10.10.78.161
The authenticity of host '10.10.78.161 (10.10.78.161)' can't be established.
ECDSA key fingerprint is SHA256:XRBUxSqs0wRKhvVrdrSF/E/0F5ccAZQlXAhMhzB1dV7U.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.78.161' (ECDSA) to the list of known hosts.
elfmceager@10.10.78.161's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-128-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Tue Jul 12 10:00:03 UTC 2022
System load: 0.1 Processes: 92
Usage of /: 39.4% of 11.75GB Users logged in: 0
Memory usage: 8% IP address for ens5: 10.10.78.161
Swap usage: 0%

0 packages can be updated.
0 updates are security updates.

Last login: Wed Dec 16 18:25:51 2020 from 192.168.190.1
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1524 started...
= attach 1524 1524
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
[+] Analyze all flags starting with sym. and entry0 (aa)
[0x00400a30]>
```

Answer: aa

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

The **db** command was used to set a breakpoint.

The screenshot shows the radare2 interface with the assembly code for the main function. A breakpoint is set at address 0x00400a30. The assembly code includes instructions for local variable declarations, a printf call, and the main loop logic. The assembly code is color-coded for readability.

```
File Edit View Search Terminal Help
File Applications Places System Tue 12 Jul, 08:39
elfmceager@tbfc-day-17:~ AttackBoxIP:10.57.255
[0x00400a30]> db 0x00400b55
[0x00400a30]> pdf @main
... main:
/ (fcn) sym.main 68
sym.main ()
    ; var int local_ch @ rbp-0xc
    ; var int local_8h @ rbp-0x8
    ; var int local_4h @ rbp-0x4
    ; DATA XFER FROM 0x00400a30d (entry)
0x00400b4d      55          push rbp
0x00400b4e      4889e5     mov rbp, rsp
0x00400b51      4883c10    sub rsp, 0x10
0x00400b55 b   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      8b4dfc     mov ecx, dword [local_4h]
0x00400b71      8b55f8     mov edx, dword [local_8h]
0x00400b74      8b45f4     mov eax, dword [local_ch]
0x00400b77      89c6       mov est, 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 ; 0x402008 ; "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      ebf6ea0000  call sym.__printf
0x00400b8a      b800000000  mov eax, 0
0x00400b8f      c9          leave
0x00400b90      c3          ret
[0x00400a30]>
```

Answer: db

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

We used **dc** command to execute the program until it hit the breakpoint that was set previously.

The screenshot shows the radare2 interface with the assembly code for the main function. The **dc** command is being run to start the program execution. The assembly code includes instructions for local variable declarations, a printf call, and the main loop logic. The assembly code is color-coded for readability.

```
File Edit View Search Terminal Help
File Applications Places System Tue 12 Jul, 08:49
elfmceager@tbfc-day-17:~ AttackBoxIP:10.57.255
[0x00400a30]> dc
child stopped with signal 28
[*] SIGNAL 28 errno=0 addr=0x00000000 code=128 ret=0
[0x00400a30]> dc
hit breakpoint at: 400b55
[0x00400b55]> px @rbp-0x8
- offset -          0 1 2 3 4 5 6 7 8 9  A B  C D  E F  0123456789ABCDEF
0x7ffffd2432e44  0000 0008 1890 6b00 0000 0000 7018 4000 .....k... p.@

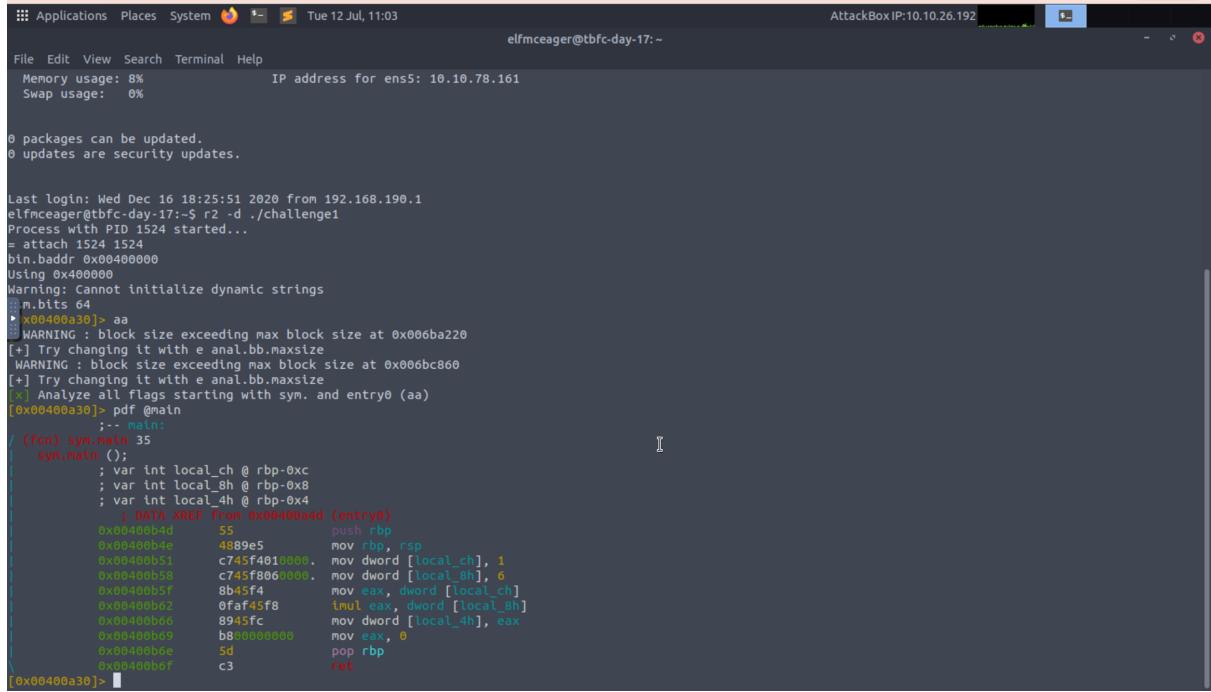
0x7ffffd2432e54  0000 0008 1911 4000 0000 0000 0000 0000 .....@.
0x7ffffd2432e64  0000 0000 0000 0000 0000 0100 0000 782f 43d2 .....x/c.
0x7ffffd2432e74  fdff 0000 400b 4000 0000 0000 0000 0000 .....M.@

0x7ffffd2432e84  0000 0000 1700 0000 0100 0000 0000 0000 .....@.
0x7ffffd2432e94  0000 0000 0000 0000 0200 0000 0000 0000 .....@.
0x7ffffd2432ea4  0000 0000 0000 0000 0000 0000 0000 0000 .....@.
0x7ffffd2432eb4  0000 0000 0000 0000 0000 0000 0004 4000 .....@.
0x7ffffd2432ec4  0000 0000 0ba6 7a4a 1c38 3b97 1019 4000 .....z3.8;...@.
0x7ffffd2432ed4  0000 0000 0000 0000 0000 0000 1890 0ba6 .....k.
0x7ffffd2432ee4  0000 0000 0000 0000 0000 0000 0000 0ba6 5a26 .....Z8.
0x7ffffd2432ef4  1a9c c068 0ba6 0e5b 1c38 3b97 0000 0000 ...h...[.8;.
0x7ffffd2432f04  0000 0000 0000 0000 0000 0000 0000 0000 .....@.
0x7ffffd2432f14  0000 0000 0000 0000 0000 0000 0000 0000 .....@.
0x7ffffd2432f24  0000 0000 0000 0000 0000 0000 0000 0000 .....@.
0x7ffffd2432f34  0000 0000 0000 0000 0000 0000 0000 0000 .....@.
[0x00400b55]>
```

Answer: dc

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

By using pdf @main, the value of local_ch could be seen.



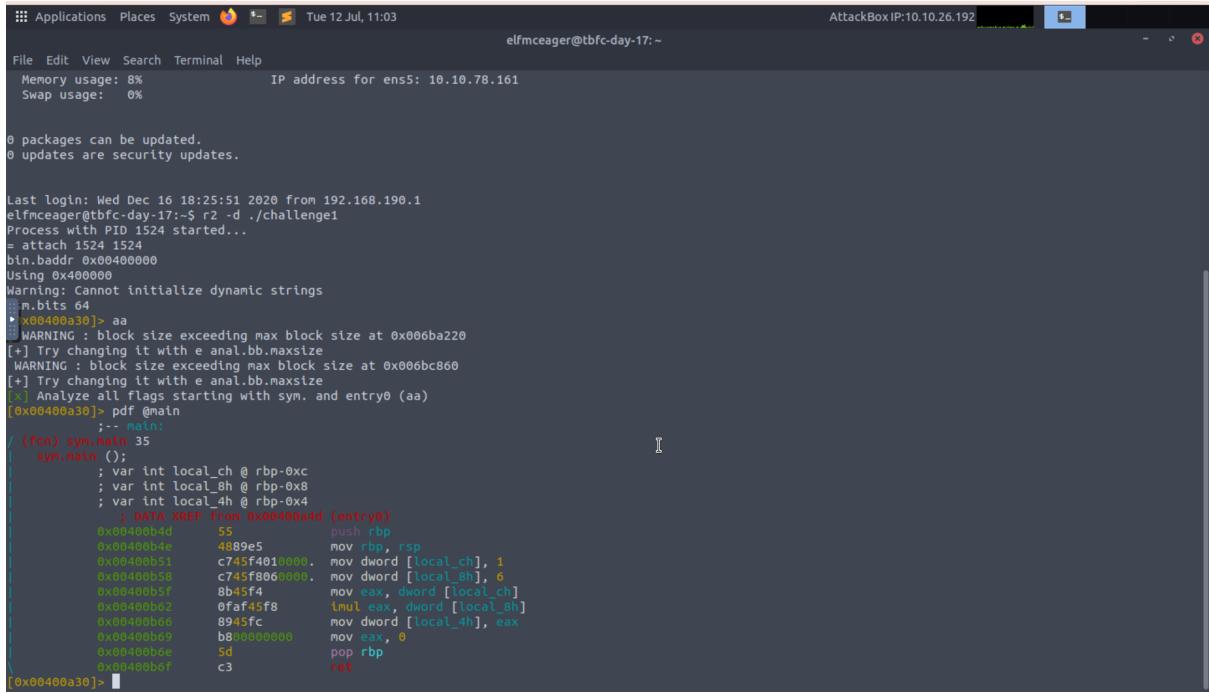
```
File Edit View Search Terminal Help
Memory usage: 8% IP address for ens5: 10.10.78.161
Swap usage: 0%
0 packages can be updated.
0 updates are security updates.

Last login: Wed Dec 16 18:25:51 2020 from 192.168.190.1
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1524 started...
= attach 1524 1524
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
:m.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:
((tcn) 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 0x00400a3d (entry0)
0x00400b4d 55          push rbp
0x00400b4e 4899e5      mov rbp, rsp
0x00400b51 c745f4010000 mov dword [local_ch], 1
0x00400b58 c745f8060000 mov dword [local_8h], 6
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
0x00400b7f c3          ret
[0x00400a30]>
```

Answer: 1

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

dword = 6 , eax = 1. Thus, $6 \times 1 = 6$



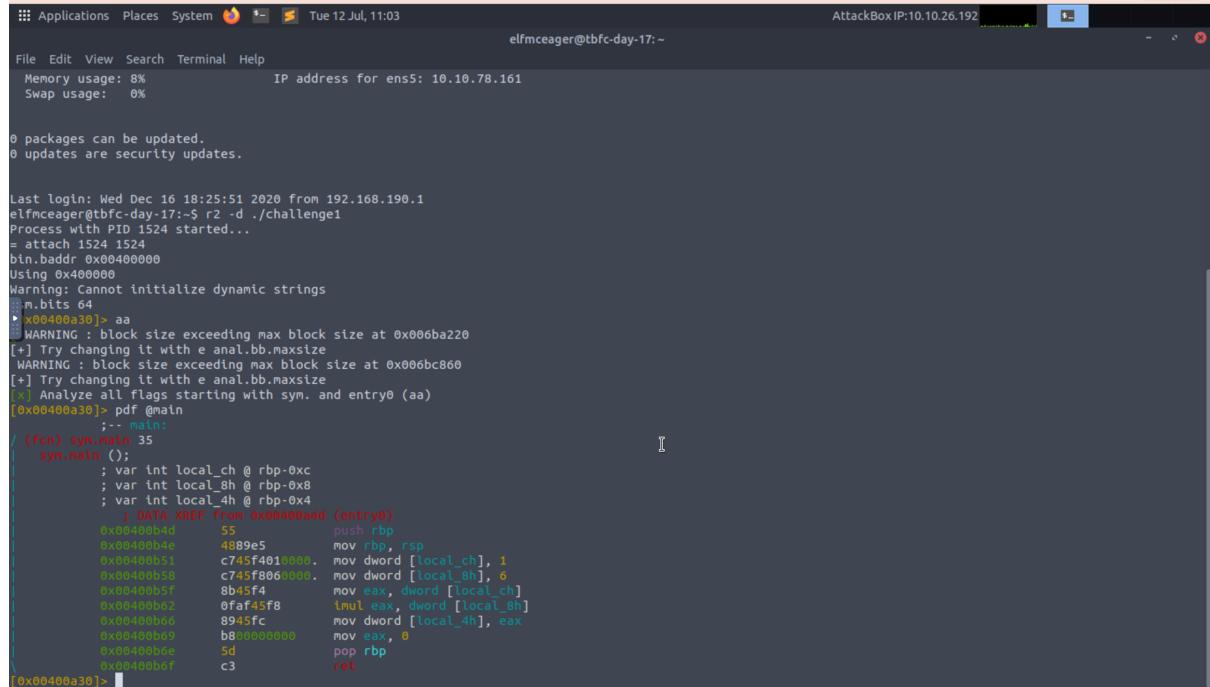
The screenshot shows a terminal window titled "elfmceager@tbfc-day-17:~". The window displays a command-line interface with various system status and log messages. At the bottom, there is a debugger session showing assembly code. The assembly code is as follows:

```
Last login: Wed Dec 16 18:25:51 2020 from 192.168.190.1
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1524 started...
= attach 1524 1524
bin.baddr 0x04000000
Using 0x400000
Warning: Cannot initialize dynamic strings
::m.bits 64
> x00400a30> 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
[!] Analyze all blocks starting with syn. 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 ebx, 0
0x00400b6e 5d          pop rbp
0x00400b6f c3          ret
[0x00400a30]>
```

Answer: 6

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

dword = 6 , eax = 1. Thus, 6x1=6



The screenshot shows a terminal window on a Linux system. The title bar indicates it's running on an AttackBox IP: 10.10.26.192. The terminal window displays the following text:

```
Last login: Wed Dec 16 18:25:51 2020 from 192.168.190.1
elfmceager@tbfc-day-17:~$ r2 -d ./challenge1
Process with PID 1524 started...
= attach 1524 1524
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
m.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
[*] Analyze all flags starting with sym. and entry0 (aa)
[0x00400a30]> pdf @main
    ;-- main:
    / (7c) sym.main 35
    sym.main () {
        ; var int local_ch @ rbp-0xc
        ; var int local_bh @ rbp-0x8
        ; var int local_4h @ rbp-0x4
        ; DATA XREF From 0x00400aad (entry0)
        0x00400b4d 55 push rbp
        0x00400b4e 4889e5 mov rbp, rsp
        0x00400b51 c745f4010000. mov dword [local_ch], 1
        0x00400b58 c745f8060000. mov dword [local_bh], 6
        0x00400b5f b845f4 mov eax, dword [local_ch]
        0x00400b62 0faf45f8 imul eax, dword [local_bh]
        0x00400b66 8945fc mov dword [local_4h], eax
        0x00400b69 b800000000 mov eax, 0
        0x00400b6e 5d pop rbp
        0x00400b6f c3 ret
[0x00400a30]>
```

Answer: 6

Thought Process/Methodology:

To start the task, **echo “10.10.78.161” > target.txt** and **cat target.txt** was used. Then, **ssh elfmceager@10.10.78.161** was used to log in. Then, password is required to connect to Ubuntu which we used **adventofcyber**. Next to start the challenge, **r2 -d ./challenge** was used. Then, we used **aa** command to analyze the program. After analyzing the program, **pdf @main** was used to examine the assembly code at main. Then, we used **db** command to set a breakpoint and **dc** command to execute the program until it hit the breakpoint.

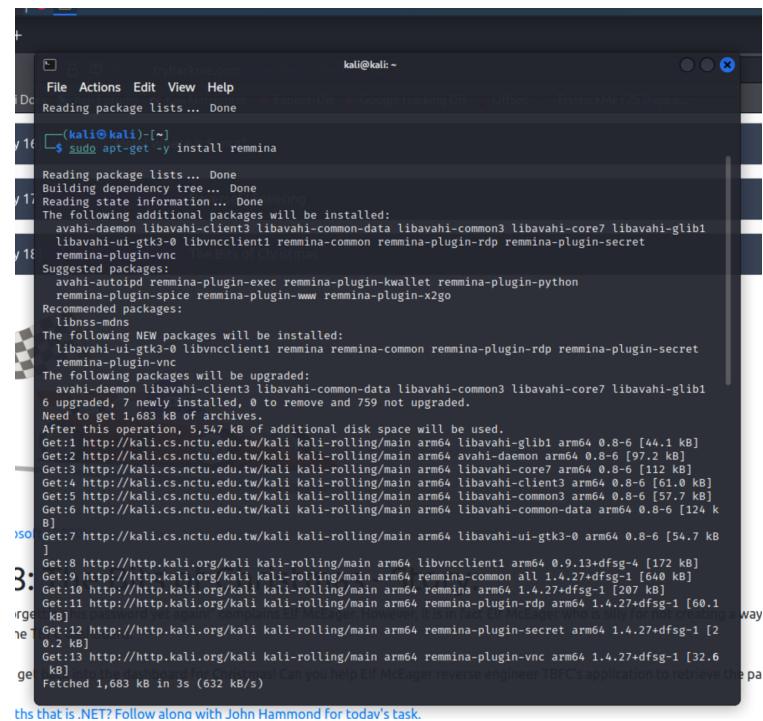
Day 18 : Reverse Engineering - The Bits of Christmas

Tools used: Kali Linux, Remmina, ILSpy, TBFC_APP, OpenVPN, FireFox, Command Prompt

Solution/walkthrough:

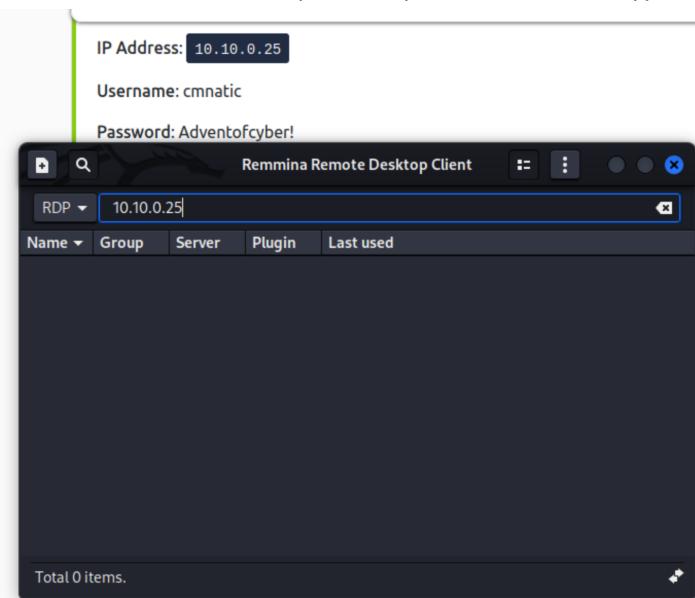
Question 1: What is the message that shows up if you enter the wrong password for TBFC_APP?

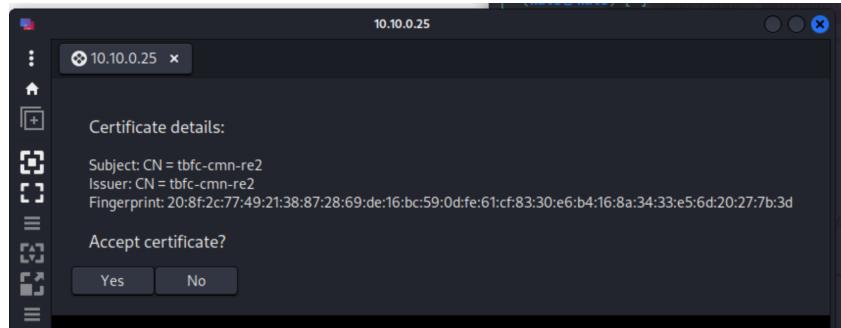
Install remmina using the command **sudo apt-get -y install remmina**.



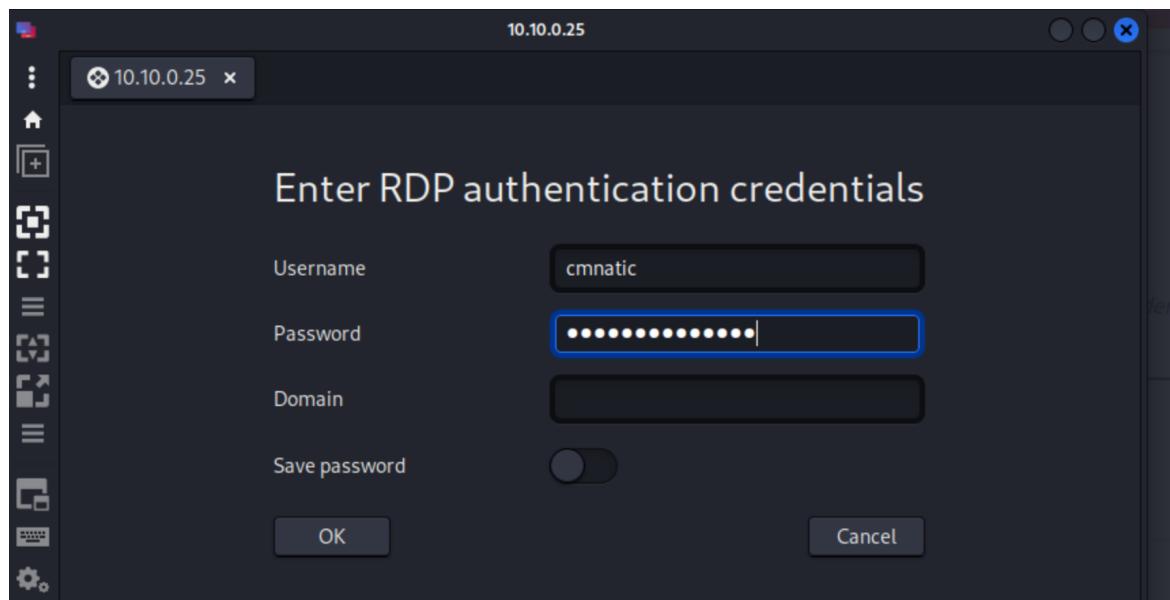
```
kali@kali: ~
File Actions Edit View Help
Reading package lists... Done
Reading state information... Done
The following additional packages will be installed:
  avahi-daemon libavahi-client3 libavahi-common-data libavahi-common libavahi-core7 libavahi-glib1
  libavahi-ui-gtk3-0 libvncclient1 remmina-common remmina-plugin-rdp remmina-plugin-secret
Suggested packages:
  avahi-daemon libremmina-plugin-exec remmina-plugin-kwallet remmina-plugin-python
  remmina-plugin-spice remmina-plugin-www remmina-plugin-x2go
Recommended packages:
  libnss-mdns
The following NEW packages will be installed:
  libavahi-ui-gtk3-0 libvncclient1 remmina remmina-common remmina-plugin-rdp remmina-plugin-secret
  remmina-plugin-vnc
The following packages will be upgraded:
  avahi-daemon libavahi-client3 libavahi-common-data libavahi-common3 libavahi-core7 libavahi-glib1
  6 upgraded, 7 newly installed, 0 to remove and 759 not upgraded.
Need to get 1,683 kB of archives.
After this operation, 5,547 kB of additional disk space will be used.
Get:1 http://kali.cs.nctu.edu.tw/kali kali-rolling/main arm64 libavahi-glib1 arm64 0.8-6 [44.1 kB]
Get:2 http://kali.cs.nctu.edu.tw/kali kali-rolling/main arm64 avahi-daemon arm64 0.8-6 [97.2 kB]
Get:3 http://kali.cs.nctu.edu.tw/kali kali-rolling/main arm64 libavahi-core7 arm64 0.8-6 [112 kB]
Get:4 http://kali.cs.nctu.edu.tw/kali kali-rolling/main arm64 libavahi-client3 arm64 0.8-6 [61.0 kB]
Get:5 http://kali.cs.nctu.edu.tw/kali kali-rolling/main arm64 libavahi-common3 arm64 0.8-6 [57.7 kB]
Get:6 http://kali.cs.nctu.edu.tw/kali kali-rolling/main arm64 libavahi-common-data arm64 0.8-6 [124 kB]
Get:7 http://kali.cs.nctu.edu.tw/kali kali-rolling/main arm64 libavahi-ui-gtk3-0 arm64 0.8-6 [54.7 kB]
Get:8 http://http.kali.org/kali kali-rolling/main arm64 libvncclient1 arm64 0.9.13+dfsg-4 [172 kB]
Get:9 http://http.kali.org/kali kali-rolling/main arm64 remmina-common all 1.4.27+dfsg-1 [640 kB]
Get:10 http://http.kali.org/kali kali-rolling/main arm64 remmina arm64 1.4.27+dfsg-1 [207 kB]
Get:11 http://http.kali.org/kali kali-rolling/main arm64 remmina-plugin-rdp arm64 1.4.27+dfsg-1 [60.1 kB]
Get:12 http://http.kali.org/kali kali-rolling/main arm64 remmina-plugin-secret arm64 1.4.27+dfsg-1 [20.2 kB]
Get:13 http://http.kali.org/kali kali-rolling/main arm64 remmina-plugin-vnc arm64 1.4.27+dfsg-1 [32.6 kB]
get  Fetched 1,683 kB in 3s (632 kB/s)
This is .NET? Follow along with John Hammond for today's task.
```

After installation has completed, open remmina and type in the machine instance IP address.

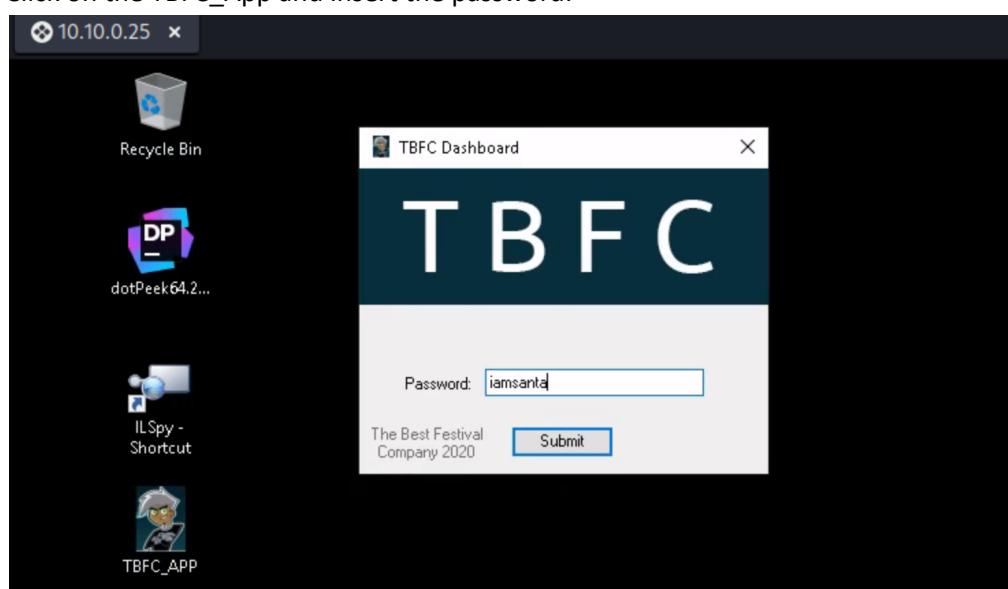




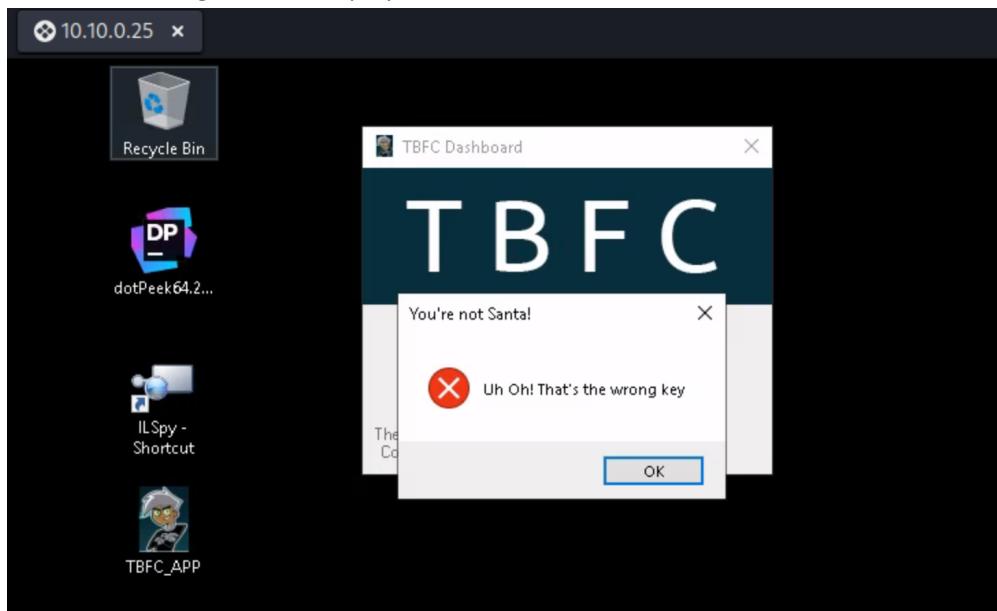
Insert the username and password given in tryhackme.



Click on the TBFC_App and insert the password.



The error message will be displayed on the screen.



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

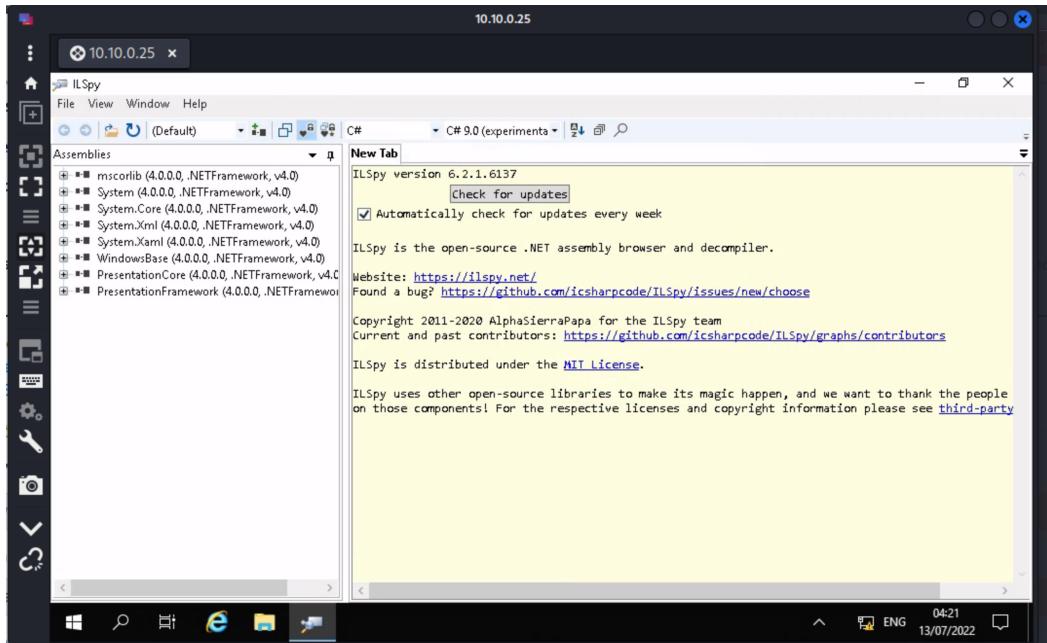
Question 2: What does TBFC stand for?



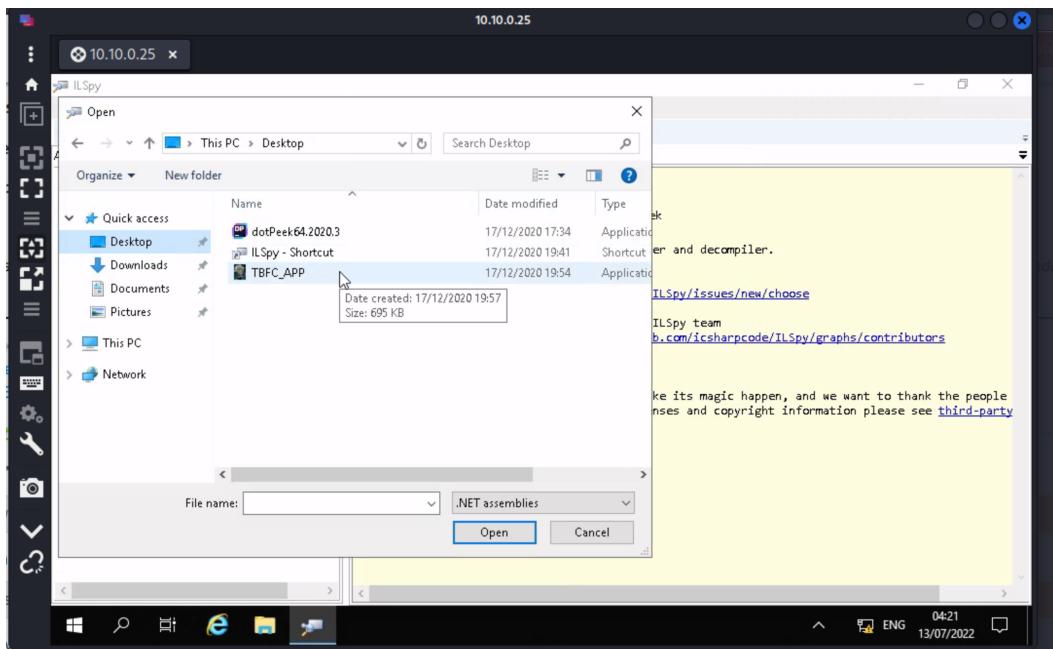
Answer: **The Best Festival Company**

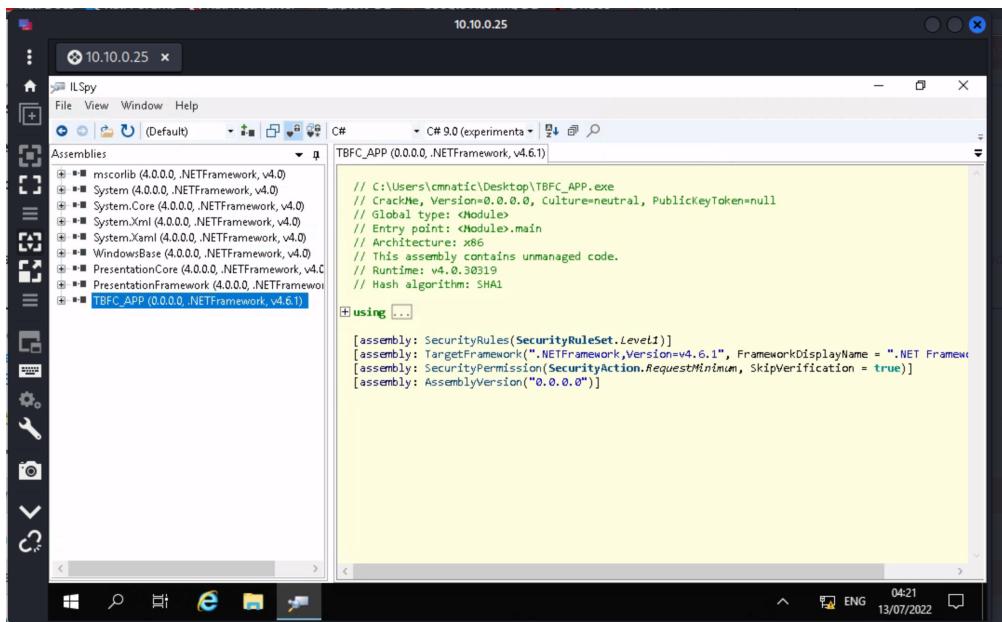
Question 3: Decompile the TBFC_APP with ILSpy. What is the module that catches your attention?

Click on ILSpy from the desktop.

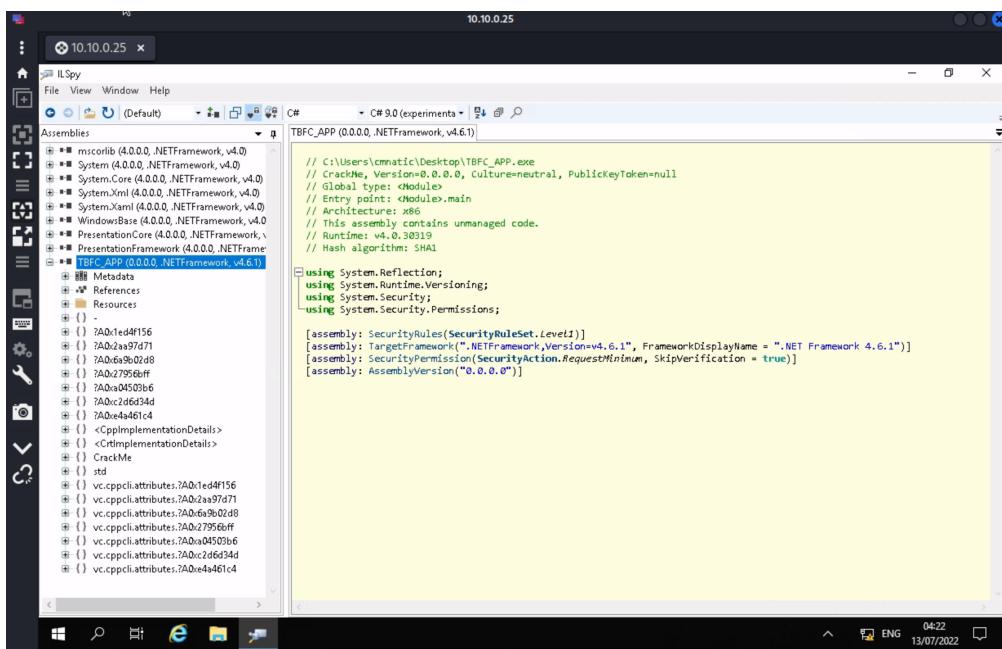


Open the TBFC_APP from ILSpy.

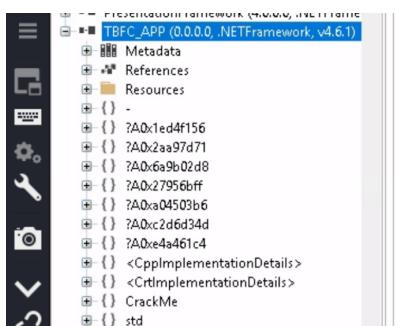




Click on the plus button to expand the TBFC_APP assemblies.



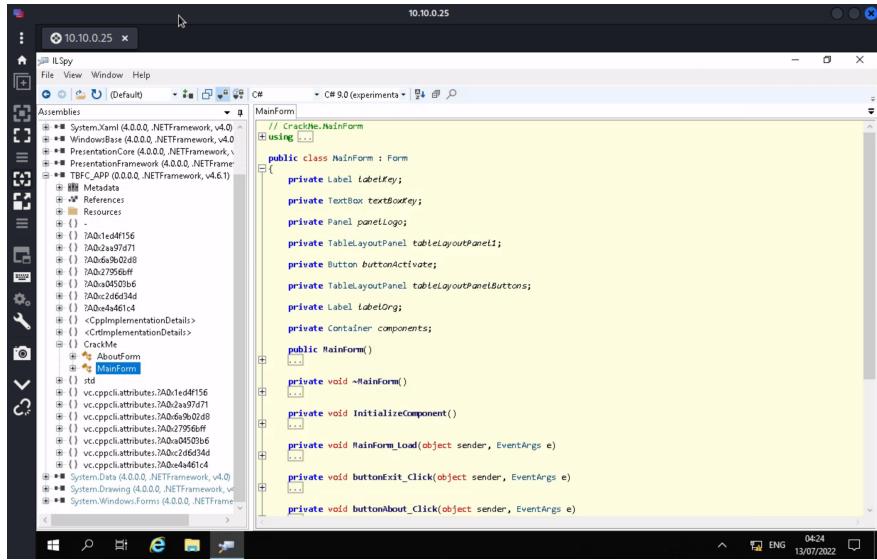
The module **CrackMe** can be seen.



Answer: CrackMe

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

Expand the CrackMe module by clicking on the plus sign. The MainForm contains information that we are looking for.



The screenshot shows the IL Spy interface with the assembly structure on the left and the code editor on the right. The assembly tree shows the following structure:

- 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)
- PresentationFramework (4.0.0.0, .NETFramework, v4.0)
- TBFC_APP (0.0.0.0, .NETFramework, v4.6.1)
- Metadata
- References
- Resources
- CrackMe (containing CrackMe, AboutForm, and MainForm)
- System (4.0.0.0, .NETFramework, v4.0)
- System.Drawing (4.0.0.0, .NETFramework, v4.0)
- System.Windows.Forms (4.0.0.0, .NETFramework, v4.0)

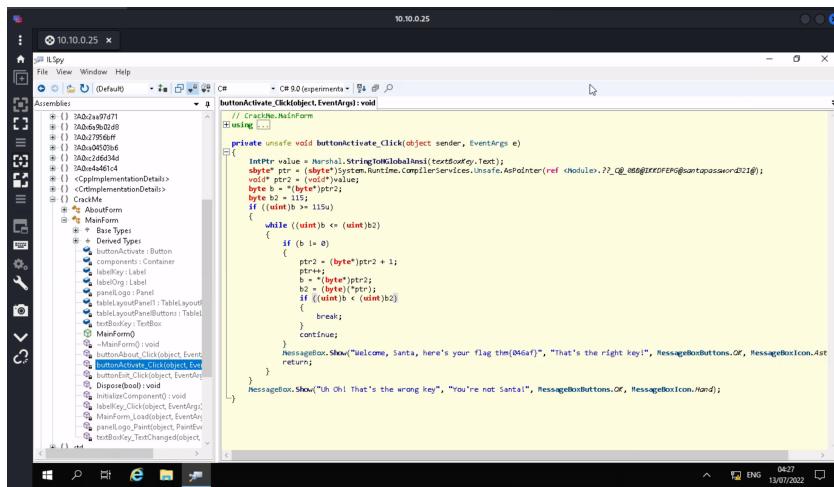
The code editor displays the MainForm.cs file:

```
// CrackMe.MainForm
using ...
public class MainForm : Form
{
    private Label labelKey;
    private TextBox textBoxKey;
    private Panel panelLogo;
    private TableLayoutPanel tableLayoutPanelButtons;
    private Label labelOrg;
    private Container components;
    public MainForm()
    {
        ...
    }
    private void Rainform()
    {
        ...
    }
    private void InitializeComponent()
    {
        ...
    }
    private void MainForm_load(object sender, EventArgs e)
    {
        ...
    }
    private void buttonExit_Click(object sender, EventArgs e)
    {
        ...
    }
    private void buttonAbout_Click(object sender, EventArgs e)
    {
        ...
    }
}
```

Answer: MainForm

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

Expand the MainForm by clicking on the plus sign. The buttonActivate method will contain the information.



The screenshot shows the IL Spy interface with the assembly structure on the left and the code editor on the right. The assembly tree shows the same structure as before, but the MainForm node is expanded to show its internal components.

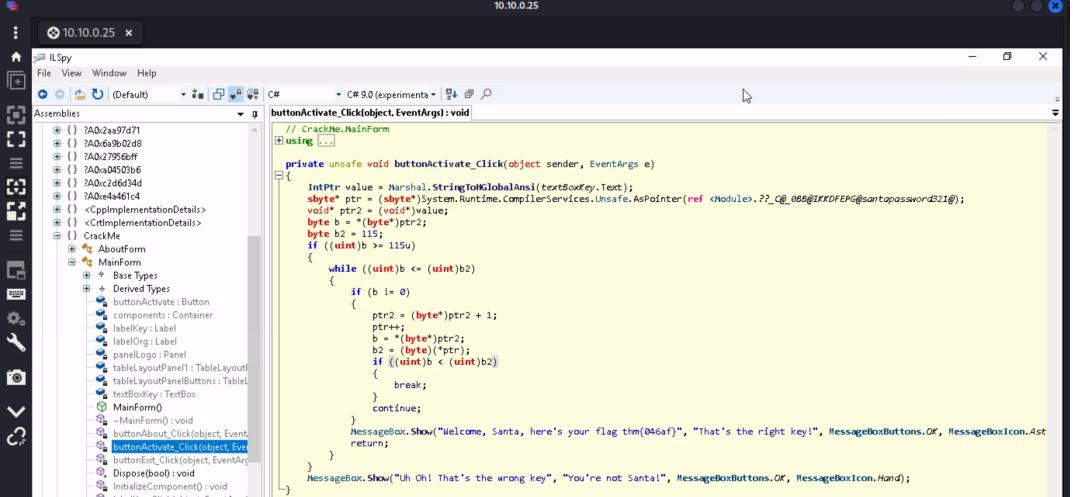
The code editor displays the buttonActivate_Click method:

```
private unsafe void buttonActivate_Click(object sender, EventArgs e)
{
    IntPtr value = Marshal.StringToGlobalAllocText(textBoxKey.Text);
    byte* ptr = (byte*)Program.Runtime.CompilerServices.Unsafe.AsPointer(ref module._2_C_0B@0B@0B@0B@0B@0B@0B@0B@0B);
    byte* ptr2 = (byte*)value;
    byte b = *(byte*)ptr2;
    if ((uint)b > 115)
    {
        while ((uint)b < (uint)b2)
        {
            if (b != 0)
            {
                ptr = (byte*)ptr2 + 1;
                ptr2 = (byte*)ptr;
                b2 = *(byte*)ptr;
                if ((uint)b < (uint)b2)
                {
                    break;
                }
                continue;
            }
            MessageBox.Show("Welcome, Santa, here's your flag tm(Miaa)", "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);
}
```

Answer: buttonActivate_click

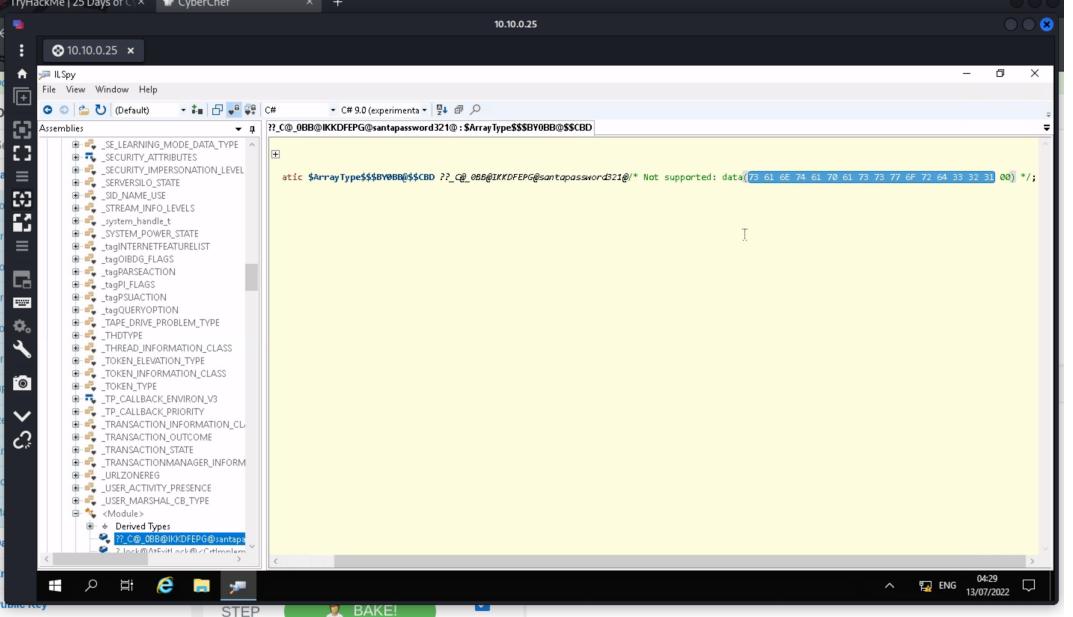
Question 6: What is Santa's password?

Click on the santapassword to retrieve the hexadecimal form.



```
10.10.0.25 x
ILSpy File View Window Help
C# C# 9.0 (experimental) 10.10.0.25
Assemblies
+ () 0AD2a397d71
+ () 0AD6a9b02d48
+ () 0AD27956ff
+ () 0ADa0450b6
+ () 0ADcxd634d
+ () 0ADxe4d4c4
+ <ImplementationDetails>
+ <ImplementationDetails>
+ <Module>
+ AboutForm
+ MainForm
+ + Base Types
+ + Derived Types
+ buttonActivate : Button
+ components : Container
+ labelKey : Label
+ labelOrg : Label
+ panelLogo : Panel
+ tableLayoutPanel1 : TableLayoutPanel
+ tableLayoutPanelPanelButtons : TableLayoutPanel
+ textBoxKey : TextBox
+ MainForm0
+ -MainForm() : void
+ buttonAbout_Click(object, EventArgs)
+ buttonActivate_Click(object, EventArgs)
+ Dispose(bool) : void
+ InitializeComponent() : void
buttonActivate_Click(object, EventArgs)
private unsafe void buttonActivate_Click(object sender, EventArgs e)
{
    IntPtr value = Marshal.StringToGlobalAnsi(textBoxKey.Text);
    byte* ptr = (byte*)System.Runtime.CompilerServices.Unsafe.AsPointer<ref Module>().??_C@_0B@IKKDFEPG@santapassword321@;
    void* ptr2 = (void*)value;
    byte b = *(byte*)ptr2;
    byte b2 = 115;
    if ((uint)b >= 115)
    {
        while ((uint)b <= (uint)b2)
        {
            if (b != 0)
            {
                ptr2 = (byte*)ptr2 + 1;
                ptr++;
                b = *(byte*)ptr;
                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);
    }
}
```

Copy the hexadecimal value and paste it into cyberchef.com .



```
TryHackMe | 25 Days of C | CyberChef x 10.10.0.25
ILSpy File View Window Help
C# C# 9.0 (experimental) 10.10.0.25
Assemblies
+ SE_LEARNING_MODE_DATA_TYPE
+ SECURITY_ATTRIBUTES
+ SECURITY_IMPERSONATION_LEVEL
+ SERVERSILE_STATE
+ SID_NAME_USE
+ STREAM_INFO_LEVELS
+ _SYSTEM_HANDLE_T
+ SYSTEM_POWER_STATE
+ _tagINTERNETFEATURELIST
+ _tagIPL_FLAGS
+ _tagPAREACTION
+ _tagP_FLAGS
+ _tagPSUCTION
+ _tagQUIFROPTION
+ _TAP_DRIVE_PROBLEM_TYPE
+ _THDTYPE
+ _THREAD_INFORMATION_CLASS
+ _TOKEN_ELEVATION_TYPE
+ _TOKEN_INFORMATION_CLASS
+ _TOKEN_PRIVILEGES
+ _TOKEN_RELABEL_CALLBACK_ENVIRON_V3
+ _TP_CALLBACK_PRIORITY
+ _TRANSACTION_INFORMATION_CL
+ _TRANSACTION_OUTCOME
+ _TRANSACTION_STATE
+ _TRANSACTIONMANAGER_INFORM
+ _URLZONEREG
+ _USER_ACTIVITY_PRESENCE
+ _USER_MARSHAL_CB_TYPE
+ <Module>
+ + Derived Types
+ ?_C@_0B@IKKDFEPG@santapassword321@: $ArrayType$SBV0BB@$SCBD
static $ArrayType$SBV0BB@$SCBD ?_C@_0B@IKKDFEPG@santapassword321@/* Not supported: data([3:61 66 74 61 70 61 73 77 6F 72 64 33 32 31 00] */;
```

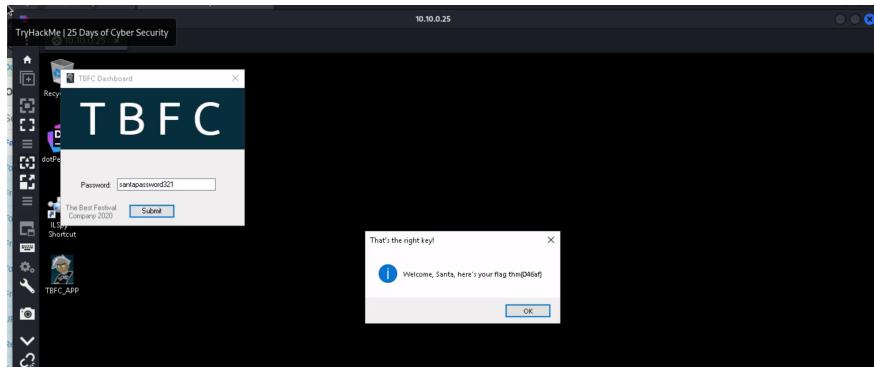
Convert the hexadecimal value.

The screenshot shows the CyberChef interface. On the left, there's a sidebar with various operations like 'To Base64', 'From Hex', etc. The main area has a 'Recipe' section titled 'From Hex' with 'Delimiter' set to 'Auto'. The 'Input' field contains the hex string: 73 61 6E 74 61 70 61 73 73 77 6F 72 64 33 32 31. Below it, the 'Output' section shows the ASCII string: santapassword321. At the bottom, there's a 'BAKE!' button and an 'Auto Bake' checkbox.

Answer: santapassword321

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

Insert the password into the TBFC_APP. The flag will be displayed in the message box.



Answer: thm{046af}

Thought Process/Methodology:

To start this task, we installed remmina into Kali Linux by using the command **sudo apt-get install remmina**. Once the tool has been installed, we typed in the machine instance IP Address and we connected to the machine instance by using the username (**cmnatic**) and password (**Adventofcyber!**) given from the tryhackme website. After successfully connecting to the machine instance, we opened **TBFC_APP** and tried to guess the password. Unfortunately, an error message was displayed on the screen telling us we were not Santa after we entered the wrong password. Thus, we used **reverse engineering** using **ILSpy** to allow us to look at the source code of the application. Firstly, we opened the **TBFC_APP** in **ILSpy**. Once we opened it, we saw an interesting module named **CrackMe**. We clicked the plus sign to expand the module and found two forms (**AboutForm** and **MainForm**). After looking through the two forms, we eventually found that the **MainForm** had a **buttonActivate_Click**. When we clicked on it, we saw a susceptible string which could potentially be Santa's password. However, we were not sure of the specific string for the password. Hence, to confirm our suspicions, we clicked on the string (**santapassword**) and it brought us to the hexadecimal value of the string. We copied this value, and pasted it into **cyberchef** to convert it. After successfully converting the value, it gave us **santapassword321**. Therefore, we opened the **TBFC_APP** and entered the password. After successfully logging in, we were shown the flag **thm{046af}**.

Day 19 : Web Exploitation—The Naughty or Nice List

Tools used: WSL, Firefox, Notepad, Kali Linux

Solution/walkthrough:

Question 1: Which list is this person on?

Start the machine and input the IP address given to your browser. Once you have successfully opened the website, input each name given into the list and observe the output given

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:

Search

Kones is on the Naughty List.

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:

Search

YP is on the Nice List.

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:

Search

Timothy is on the Naughty List.

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:

Search

JJ is on the Naughty List.

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:

Search

Ian Chai is on the Nice List.

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:

Search

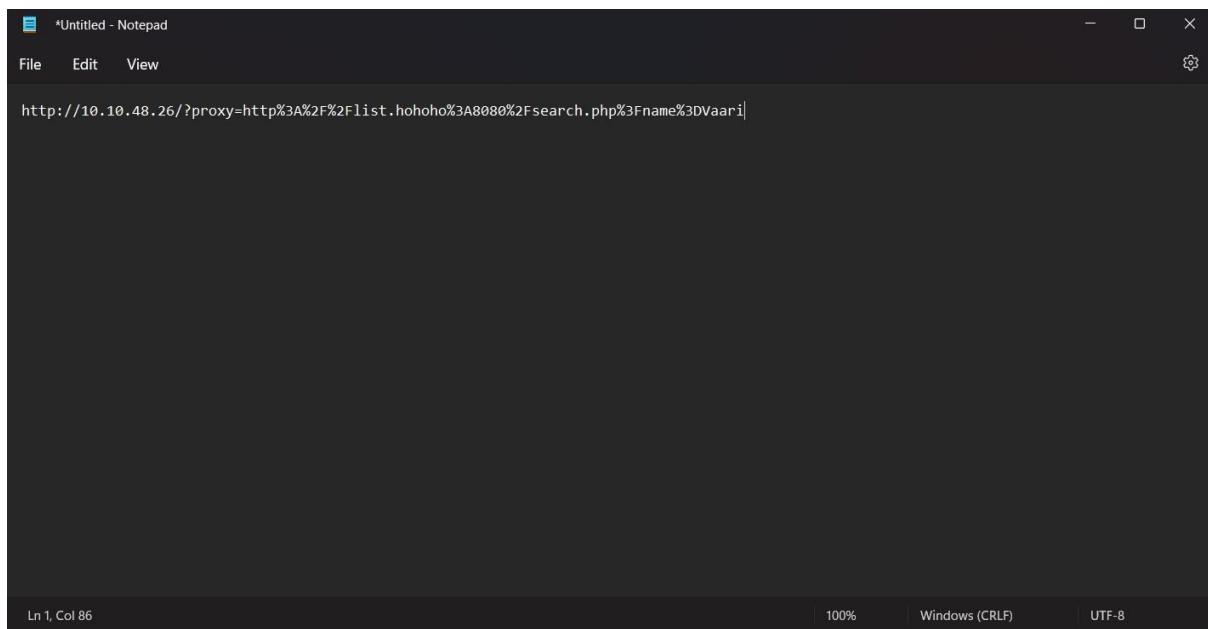
Tib3rius is on the Nice List.

Answer:

Name	Naughty or Nice
Ian Chai	Nice
Tib3rius	Nice
JJ	Naughty
Timothy	Naughty
YP	Nice
Kanes	Naughty

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

Copy the URL and paste it at your desired text editor



Open a new tab at your browser and search cyberchef, then decode it. Observe and understand the output given.

The screenshot shows the CyberChef interface in Mozilla Firefox. The URL is `https://gchq.github.io/CyberChef/#recipe=URL_Decode()&input=aHR0%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%3DVaari`. The input is decoded to `http://list.hohoho:8080/search.php?name=Vaari`.

Now we got the idea on how the website fetch the data, alter the URL by replacing the “/?proxy...”

to

“/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F”? and observe the output.

The screenshot shows a web browser window titled "The Naughty or Nice List". The URL is `10.10.48.26/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F`. The page features a cartoon Santa Claus holding a sack full of gifts. The text on the page reads:

Welcome children!

To find out if you are currently on the naughty list or the nice list, please enter your name below!

Have a Merry Christmas! Ho ho ho!

- Santa

Name:

Search

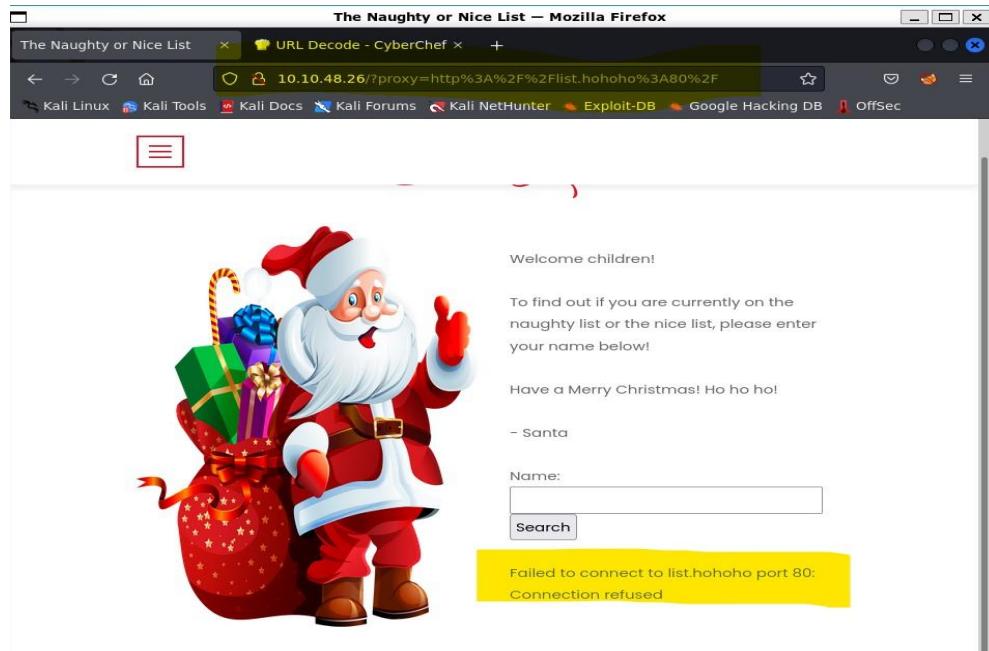
Not Found

The requested URL was not found on this server.

Answer: 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"?

Replace ".../?proxy..." to "/?proxy=http%3A%2F%2Flist.hohoho%3A80" and observe the output.



Answer: 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"

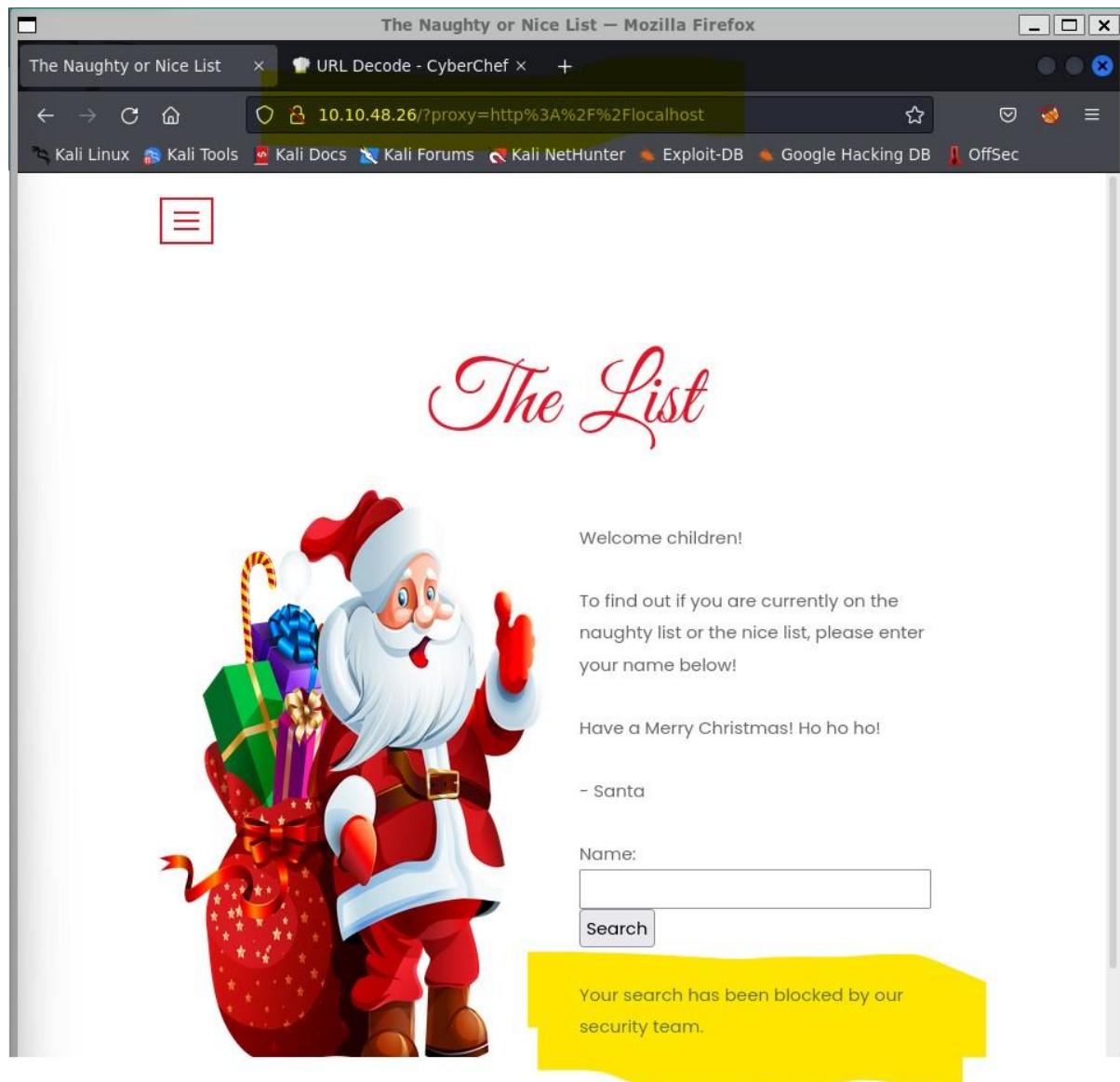
Replace ".../?proxy..." to "?proxy=http%3A%2F%2Flist.hohoho%3A22" and observe the output.

The screenshot shows a Mozilla Firefox browser window titled "The Naughty or Nice List – Mozilla Firefox". The address bar displays the URL "10.10.48.26/?proxy=http%3A%2F%2Flist.hohoho%3A22%2F". The page content features a large red banner at the top with the text "The List" in a cursive font. Below the banner is a cartoon illustration of Santa Claus carrying a large sack filled with wrapped gifts. To the right of the illustration, there is text: "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!", and "- Santa". Below this text is a form with a "Name:" label and a text input field, followed by a "Search" button. A yellow callout box at the bottom right contains the error message "Recv failure: Connection reset by peer".

Answer: Recv failure: Connection reset by peer

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

Replace ".../?proxy..." to "/?proxy=http%3A%2F%2Flocalhost" and observe the output.



Answer: Your search had been blocked by our security team.

Question 6: What is Santa's password?

By taking advantage of DNS subdomain, we be using localtest.me as our DNS subdomain, list.hohoho.localtest.me

```
kali@Vaari_HP: ~      x | + | v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\vaari> kali
[kali@Vaari_HP]~
$ host localtest.me
localtest.me has address 127.0.0.1
localtest.me has IPv6 address ::1

[kali@Vaari_HP]~
$ host vaari.localtest.me
vaari.localtest.me has address 127.0.0.1
vaari.localtest.me has IPv6 address ::1

[kali@Vaari_HP]~
$
```

Replace .../?proxy... to /?proxy=http%3A%2F%2Flist.hohoho.localtest.me and analyse the output.

The Naughty or Nice List — Mozilla Firefox

Welcome children!

To find out if you are currently on the naughty list or the nice list, please enter your name below!

Have a Merry Christmas! Ho ho ho!

~ Santa

Name:

Search

Santa:

If you need to make any changes to the Naughty or Nice list, you need to login.

I know you have trouble remembering your password so here it is: Be good for goodness sake!

~ Elf McSkidly

Answer: Be good for goodness sake!

Question 7: What is the challenge flag?

Go to admin page and enter Santa's credentials.

The Naughty or Nice List — Mozilla Firefox

The Naughty or Nice List

10.10.8.50

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



your name below!

Have a Merry Christmas! Ho ho ho!

- Santa

Name:

Search

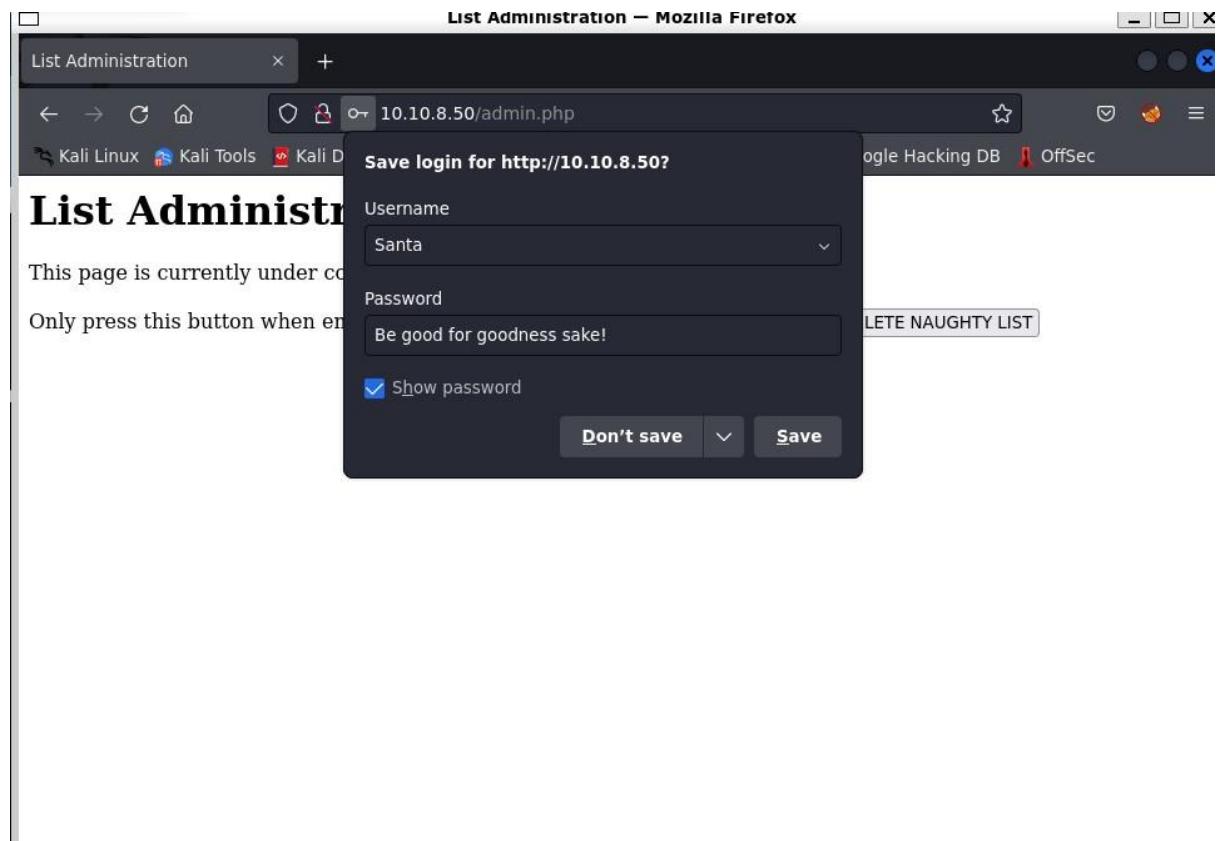
Admin

Username:

Password:

Login

Once we have successfully figured out and logged into the web application as Santa, we can delete the naughty list.



Click the 'Delete Naughty List' button to delete the list

List Administration

This page is currently under construction.

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

Observe the output once you have successfully deleted the naughty list.



Answer: THM{EVERYONE GETS PRESENTS}

Thought Process/Methodology:

Once we have successfully started the machine, input the IP address of the machine into our browser. We are greeted with a web application where we can check whether we are in Santa's Nice or Naughty list. From Google forms, we can input the names given into the search box and observe the output. By entering the names given, we can observe that the URL changes. Then, we copied the entire IP address and pasted it into our desired text editor. Afterwards, we decided to copy `"/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2Fsearch.php%3Fname%..."` and paste it into CyberChef and decode the URL. By decoding the URL, we can observe that there is a back-end machine (`list.hohoho`) where the data is fetched and sent to the web app. We can find ports and valid URLs for the back-end machine. By changing the port number to **80** from altering the URL to `".../?proxy=http%3A%2F%2Flist.hohoho%3A80"`, We can observe that the message states "**Failed to connect to list.hohoho port 80: Connection refused**" which means **port 80** is not open on the back-end machine. By changing the port number to **22** we can observe that the message changes to "**Recv failure: Connection reset by peer**" which means **port 22 is open** but it did not understand what was being sent because sending an **HTTP request** to an **SSH server** isn't a viable option. We can also try changing the `list.hohoho` hostname with localhost and observe the output. By changing it, we can observe that there is a message returned states "**Your search has been blocked by our security team.**" To find out Santa's credentials, we can simply bypass the backend machine by taking advantage of **DNS subdomains** in which we will be using `localtest.me` which resolves every subdomain to **127.0.0.1**. To do so, we can alter the website to `".../?proxy=http%3A%2F%2Flist.hohoho.localtest.me"`. When we observe the output, we can see that we have successfully discovered Santa's password. All we have to do is, go to the admin login and type the username: **Santa** and password: **Be good for goodness sake!** . Once we have successfully logged into the backend system we can finally delete the naughty list.

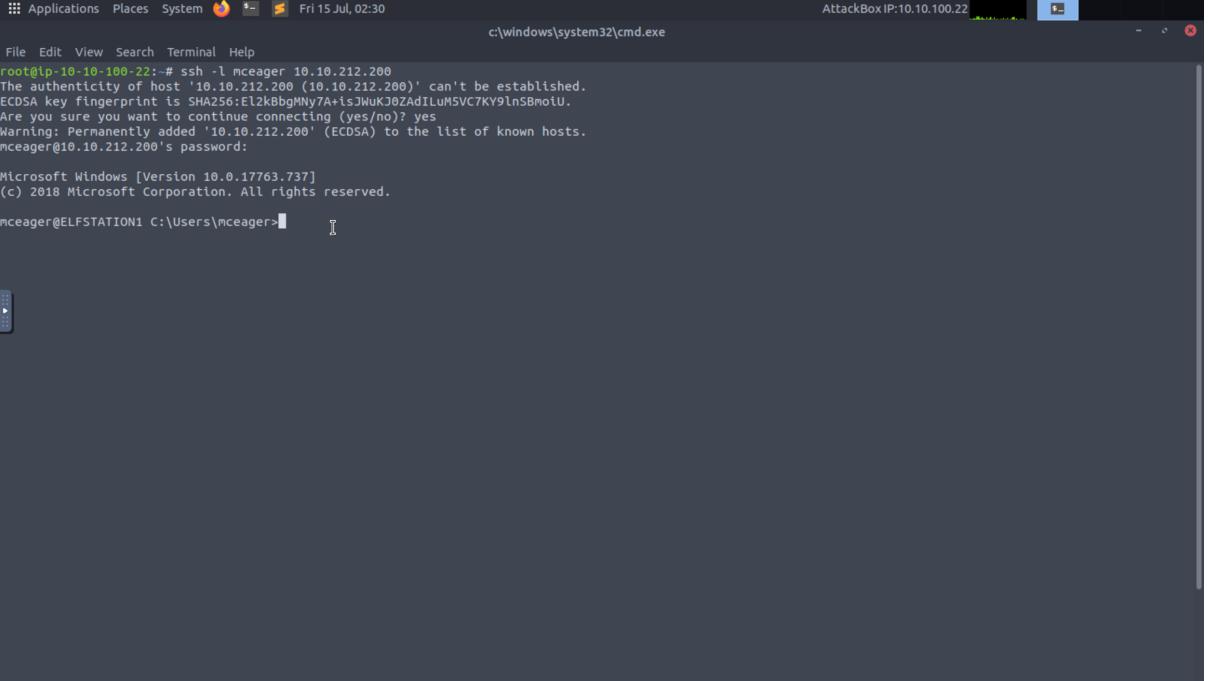
Day 20: Blue Teaming - Powershell to the rescue

Tools used: AttackBox, Terminal

Solution/walkthrough:

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

The -l command is used to specify our username and the hostname that we are connecting to.



A screenshot of a Windows terminal window titled 'cmd.exe' with the path 'c:\windows\system32'. The window shows the following text:

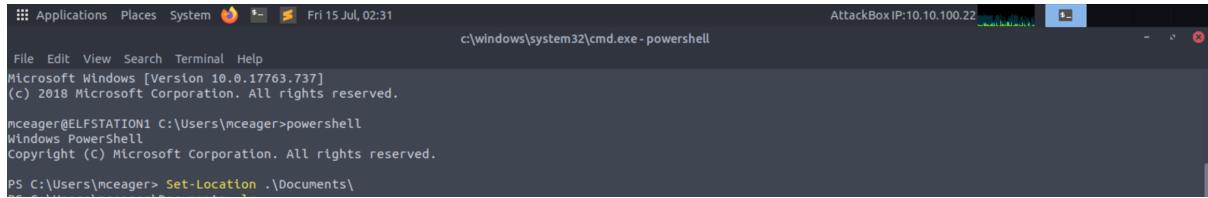
```
File Edit View Search Terminal Help
root@lp-10-10-100-22:~# ssh -l mceager 10.10.212.200
The authenticity of host '10.10.212.200 (10.10.212.200)' can't be established.
ECDSA key fingerprint is SHA256:ElzkbgbMny7A+i5JWuKj0ZAdILuMSVC7KYlnSBmoiU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.212.200' (ECDSA) to the list of known hosts.
mceager@10.10.212.200's password:
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

mceager@ELFSTATION1 C:\Users\mceager>
```

Answer: login name

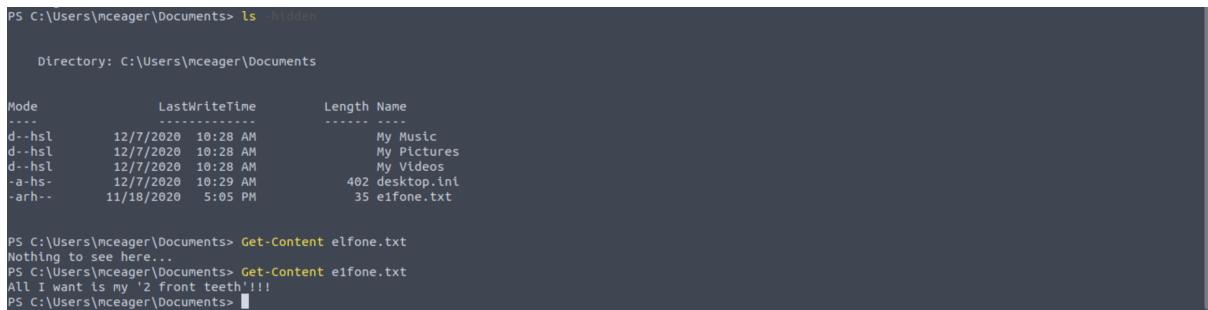
Question 2: Search for the first hidden elf file within the Documents folder. Read the contents of this file. What does Elf 1 want?

Set-Location is used to set the working location to the Documents location.



```
PS C:\Users\mceager> Set-Location .\Documents
```

Get-Content is used to see the contents of the file.



```
PS C:\Users\mceager\Documents> ls -hidden
Directory: C:\Users\mceager\Documents

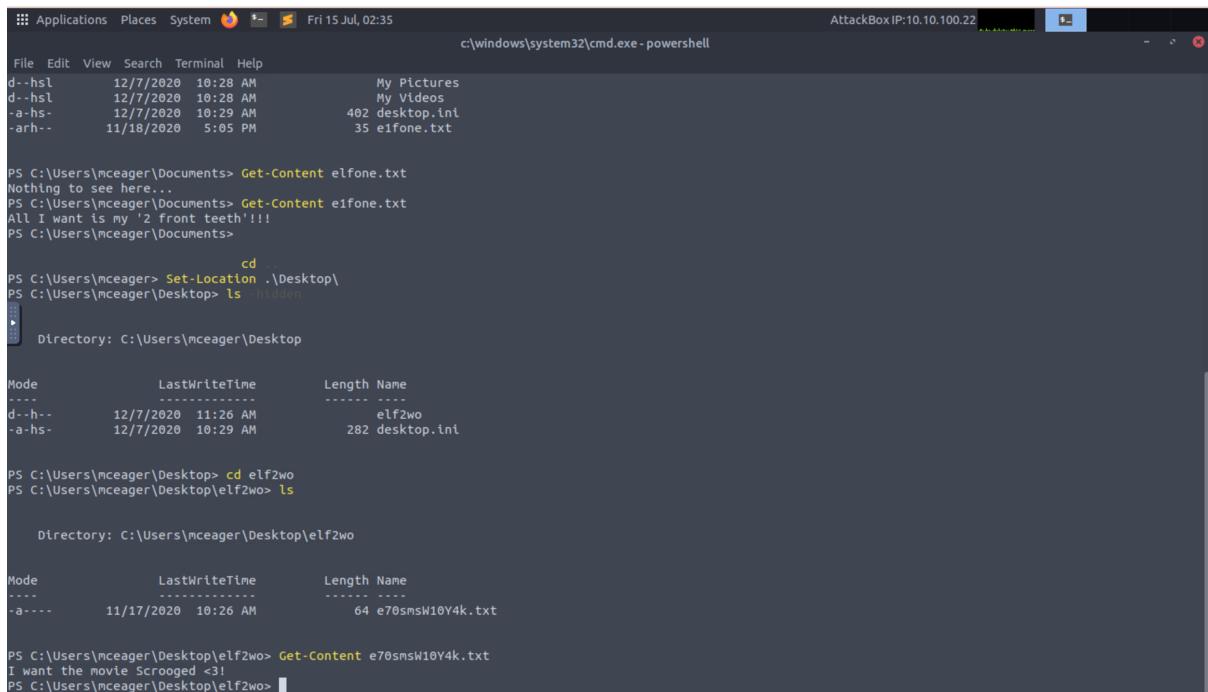
Mode                LastWriteTime      Length Name
----                -----        ---- 
d--hs1           12/7/2020 10:28 AM          0 My Music
d--hs1           12/7/2020 10:28 AM          0 My Pictures
d--hs1           12/7/2020 10:28 AM          0 My Videos
-a-hs-          12/7/2020 10:29 AM       402 desktop.ini
-ahr--         11/18/2020 5:05 PM        35 e1fone.txt

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

Answer: 2 front teeth

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?

Set-Location is used again to set working location to Desktop. **Is -Hidden** is used to get the list of the hidden folders. The content of the hidden folder can be found using **cd** command. Then, **Get-Content** is used to see the contents of the file.



```
PS C:\Users\mceager> Set-Location .\Desktop
PS C:\Users\mceager\Desktop> ls -hidden
Directory: C:\Users\mceager\Desktop

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

PS C:\Users\mceager\Desktop> cd elf2wo
PS C:\Users\mceager\Desktop\elf2wo> ls
Directory: C:\Users\mceager\Desktop\elf2wo

Mode                LastWriteTime      Length Name
----                -----        ---- 
-a---           11/17/2020 10:26 AM         64 e70smsW10Y4k.txt

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

Answer: 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? (This command will take a while)

From Users directory, we changed to Windows directory by using **cd C:/Windows** command.

```
PS C:\Users\mceager\Desktop\elf2wo> cd C:/Windows  
PS C:\Windows> System32
```

By using **cd System32**, we went into the System32 directory to get the list of hidden folders

```
PS C:\Windows> cd System32  
PS C:\Windows\System32> Get-ChildItem -Hidden -Directory -Filter "*3"  
PS C:\Windows\System32>
```

Using the **ls -Hidden** command, the hidden folders can be found.

```
PS C:\Windows\System32> ls -hidden  
  
Directory: C:\Windows\System32  
  
Mode LastWriteTime Length Name  
---- ----- ---- -  
d-h-- 11/23/2020 3:26 PM 3lfthr3e  
d-h-- 11/23/2020 2:26 PM GroupPolicy  
  
PS C:\Windows\System32>
```

Answer: 3lfthr3e

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

By using **cd 3lfthr3e**, we get to change directory to the hidden folder which is the 3lfthr3e. Then, we used **Get-ChildItem -hidden** command to get the hidden items in the folder. The list can be found then. For this question, **Get-Content 1.txt | Measure-Object -Word** is used to get the number of words that the file contains.

```
PS C:\Windows\System32>  
  
cd 3lfthr3e  
C:\Windows\System32\3lfthr3e> Get-ChildItem -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  
  
PS C:\Windows\System32\3lfthr3e>  
  
> Get-Content 1.txt | Measure-Object -Word  
Lines Words Characters Property  
----- ----- ----- -----  
9999
```

Answer: 9999

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

For this question, **(Get-Content 1.txt)[551,6991]** is used to get the words at index 551 and 6691 in the 1.txt file. Then, the words can be seen.

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

Answer: 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)

We used `Get-Content 2.txt | Select-String -Pattern "redryder"` to see what does Elf 3 want. Then, the words can be seen.

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

    > Get-Content 2.txt | Select-String -Pattern

    > Get-Content 2.txt | Select-String -Pattern "redryder"

redryderbbgun

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

Answer: redryderbbgun

Thought Process/Methodology:

To start the task, we used `ssh -l mceager 10.10.212.200` to login name. Then, we used **Set-Location .\Documents** to set working locations to Documents. After that, to list out the hidden files in the Documents, **Is -Hidden** command was used and the list could be seen. Next, we used **Get-Content e1fone.txt** to get the content of the file to answer the question given. For the next task, the working location was changed from Documents to Desktop by using **Set-Location .\Desktop** and **Is -Hidden** was used to get the list of hidden folders. By using **cd** command, we got into the folder and **Is** command was used to list out the files from the folder. To see the content of the folder, **Get-Content e70smsW10Y4k.txt**. The answer for the question can be seen which is in the form of sentence **I want the movie Scrooged <3!**. Next task, to change from Users to Windows, **cd C:/Windows** was used and **cd System32** to get into the System32 folder. The **Is -Hidden** command again was used to list out the hidden folders and **3lfthr3e** could be seen. We got into the folder by using **cd 3lfthr3e** command and **Get-ChildItem -Hidden** to get the list of the files inside the folder. Then, to get the number of words contained in the **1.txt** file, **Get-Content 1.txt | Measure-Object -word** was used. The answer **9999** can be seen. After that, **(Get-Content 1.txt)[551,6991]** was used to see the words at index 551 and 6991 which are Red and Ryder. Next, **Get-Content 2.txt | Select-String -Pattern "redryder"** was used to get the items that Elf 3 wants which is **redryderbbgun** that could be seen next.