PSP0201

Week 5

Writeup

Group Name: Woohoo

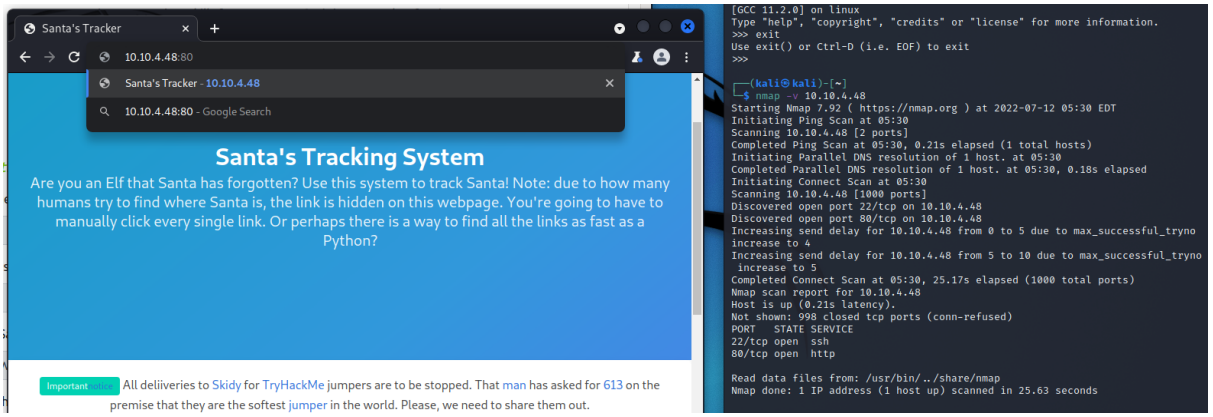
Members

ID	Name	Role
1211100312	CHAN HAO YANG	Leader
1211101506	LEONG JIA YI	Member
1211101961	CHAI DI SHENG	Member
1211101726	TAI JIN PEI	Member

Day 16 - [Scripting] Help! Where is Santa?

Tools used: Kali Linux, Firefox, Terminal

Solution/walkthrough:



*First, we used the command: **nmap -v 10.10.4.48** to find the port number for the web server.*

Q1: What is the port number for the web server?

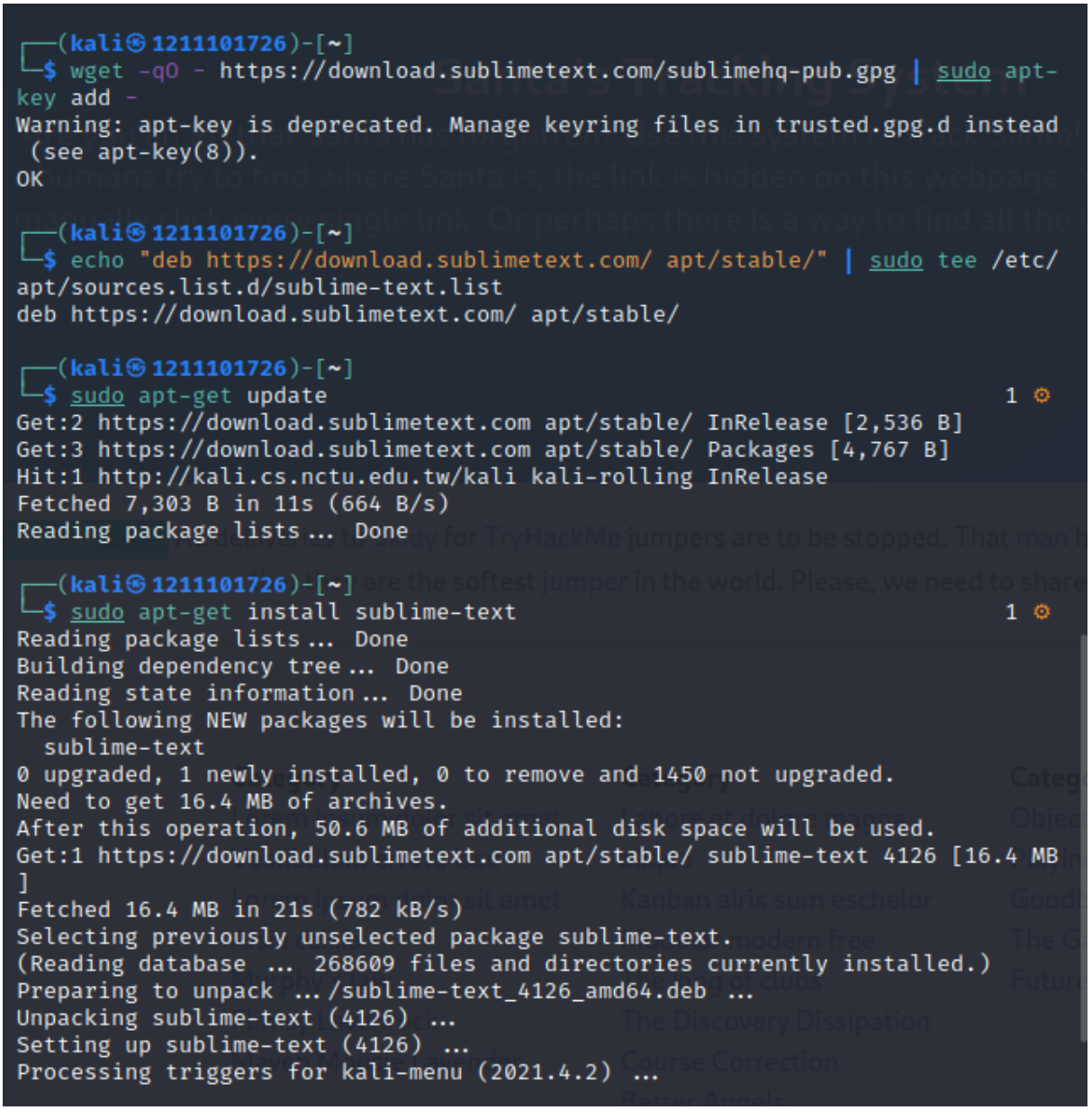
Answer: 80

```
</ul>
<span class="navbar-item">
    <a class="button is-white is-outlined" href="https://github.com/BulmaTemplates/bulma-templates/blob/master/templates/hero.html">
        <span class="icon">
            <i class="fa fa-github"></i>
        </span>
        <span title="Hello from the other side">View Source. Template not my own.</span>
    </a>
</span>
```

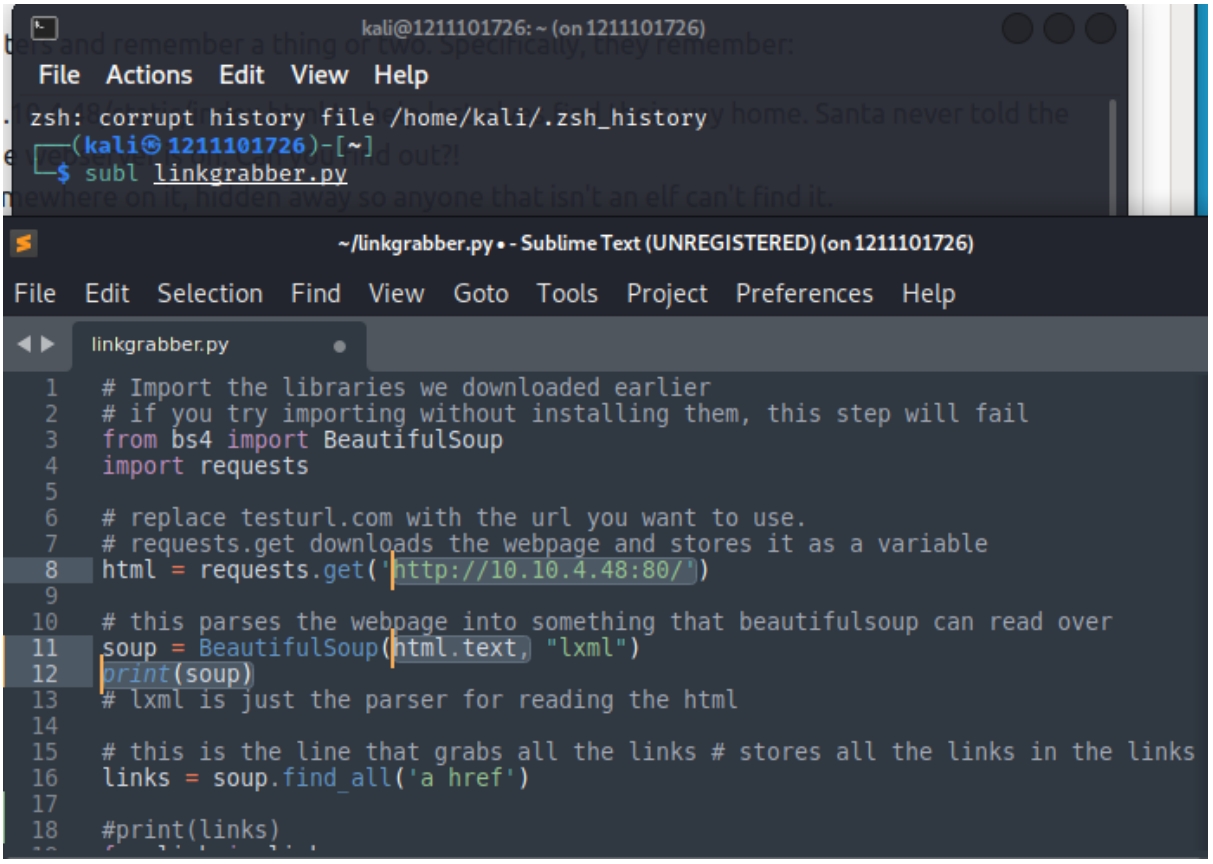
Second, we viewed the page source and found that Bulma templates is being used.

Q2: What templates are being used?

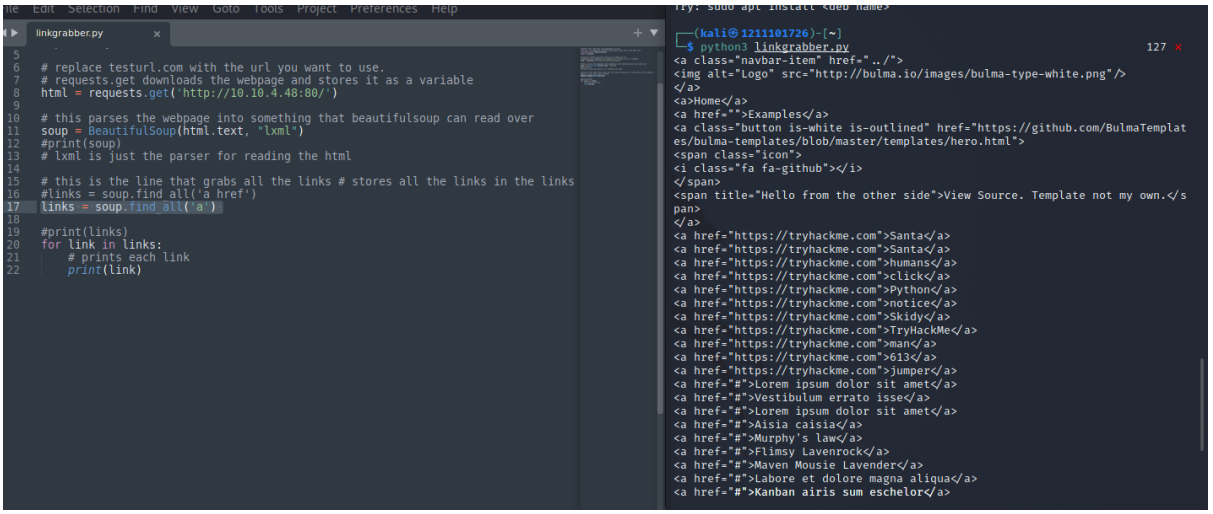
Answer: Bulma



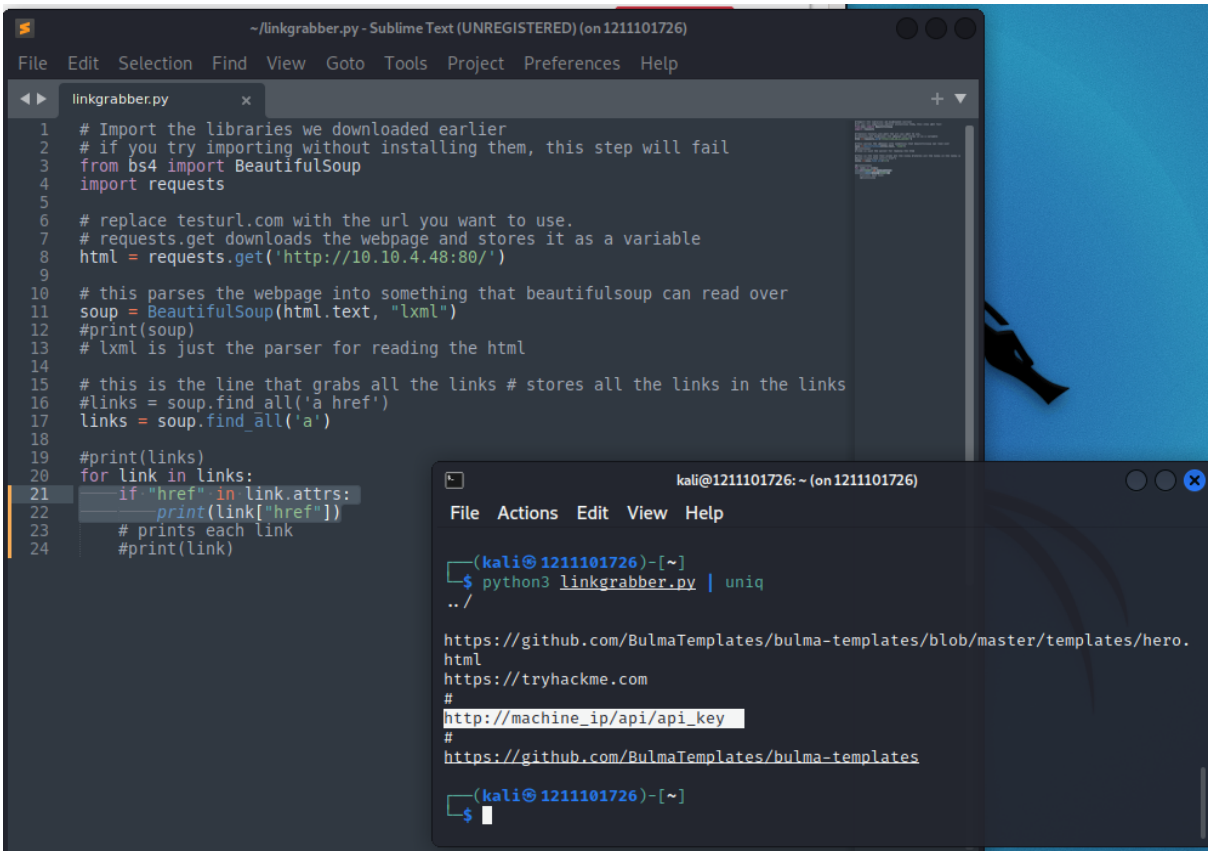
Third, we downloaded the sublimetext which includes a command-line helper called ***subl****.*



We then used the command: ***subl filename.py*** *to create a python cheat that can grab the lines.*



We then grabbed all the lines.

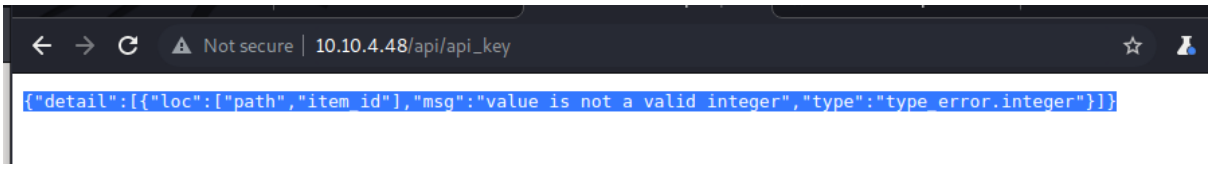


We used the command: | uniq which only showed the unique lines((non-duplicate lines).

We found that the directory for the API is at **/api/.**

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

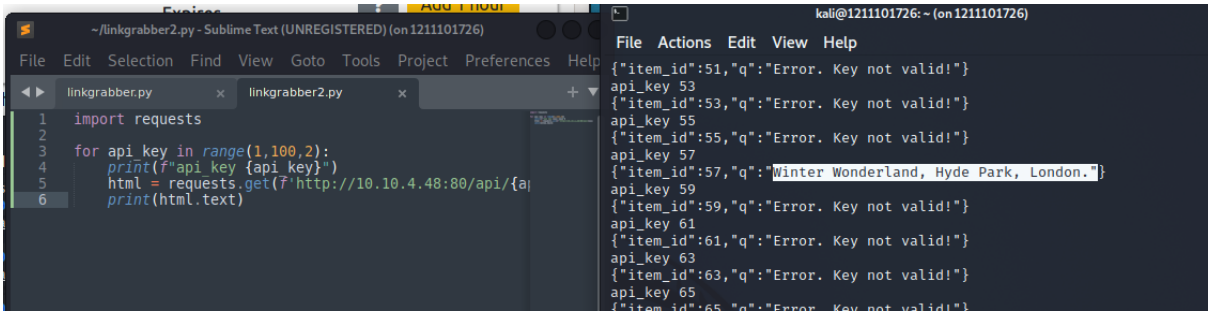
Answer: /api/



Forth, we went to API endpoint and the raw data appeared.

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

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



Fifth, we also used the cheat to print out all the API lines by looping and the places with the API key were shown.

Q5: Where is Santa right now? (Tick all correct answers.)

Answer: Winter Wonderland, Hyde Park, London

Q6: Find out the correct API key. Remember, this is an odd number between 0-100. After too many attempts, Santa's Sled will block you. To unblock yourself, simply terminate and re-deploy the target instance (10.10.94.92)

Answer: 57

Thought Process/Methodology:

First, we used the command: **nmap -v 10.10.4.48** to find the port number for the web server. Second, we viewed the page source and found that Bulma templates is being used. Third, we downloaded the sublimetext which includes a command-line helper called **subl**. We then used the command: **subl filename.py** to create a python cheat that can grab the lines. We then grabbed all the lines. We used the command: | uniq which only showed the unique lines((non-duplicate lines). Forth, we went to API endpoint and the raw data appeared. Fifth,

we also used the cheat to print out all the API lines by looping and the places with the API key were shown.

Day 17 - [Reverse Engineering] ReverseELFneering

Tools used: Kali Linux, Firefox, Terminal

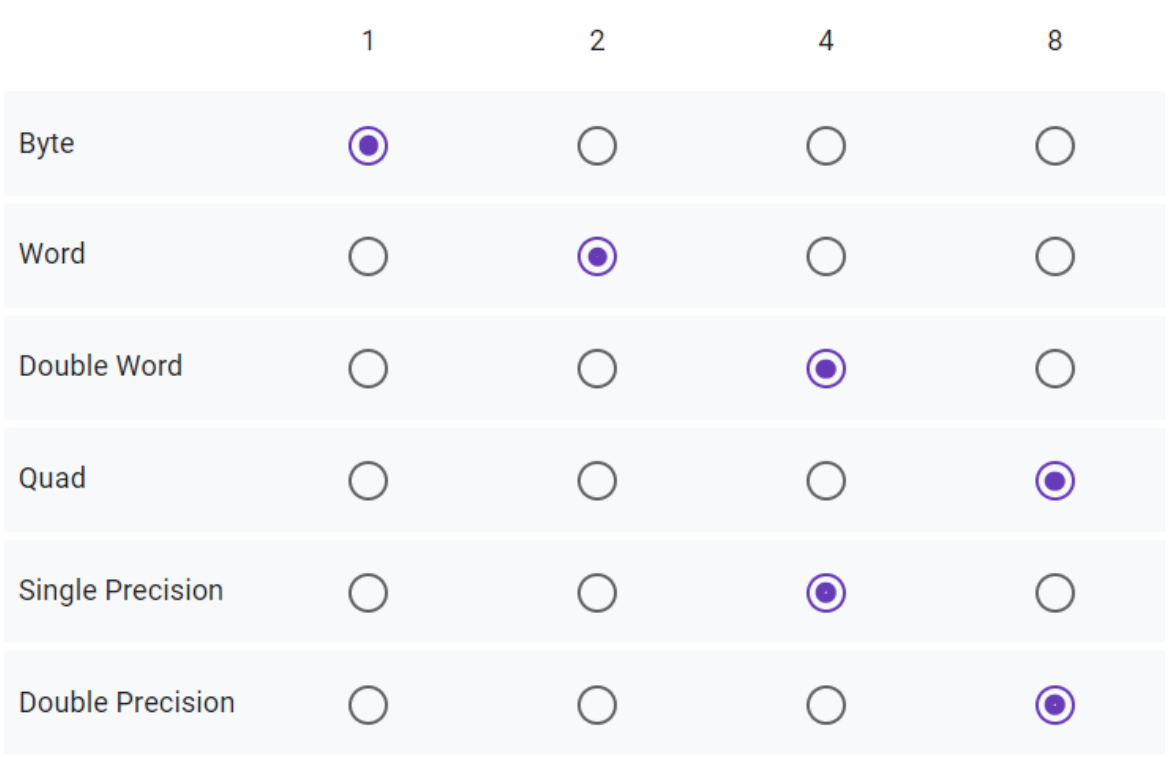
Solution/walkthrough:

- Transfer control to other parts of the program Since the architecture is x86-64, the registers are 64 bit and Intel has a list of 16 registers:

Initial Data Type	Suffix	Size (bytes)
Byte	b	1
Word	w	2
Double Word	l	4
Quad	q	8
Single Precision	s	4
Double Precision	l	8

Q1: Match the data type with the size in bytes:

Answer:



```
└─$ ssh elfmceager@10.10.6.157                                    255 ×
The authenticity of host '10.10.6.157 (10.10.6.157)' can't be established.
ED25519 key fingerprint is SHA256:+Yl8Ef3BjQ7HNTMf6qew50LnmiqEXXSzLqgX82k/RS
g.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.6.157' (ED25519) to the list of known host
s.
elfmceager@10.10.6.157's password:
Permission denied, please try again.
elfmceager@10.10.6.157'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 Wed Jul 13 03:29:37 UTC 2022

  System load:  0.0               Processes:           93
  Usage of /:   39.4% of 11.75GB  Users logged in:     0
  Memory usage: 8%                IP address for ens5: 10.10.6.157
  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:~$
```

First, we logged in to the target's server.

```
elfmceager@tbfc-day-17:~$ r2 -d ./file1
Process with PID 1598 started...
= attach 1598 1598
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
asm.bits 64
[0x00400a30]>
```

*Second, we used the command: **r2 -d ./file1** to open the binary in debugging mode.*

```
[0×00400a30]> aa
[ WARNING : block size exceeding max block size at 0×006ba220
[+] Try changing it with e anal.bb.maxsize
 WARNING : block size exceeding max block size at 0×006bc860
[+] Try changing it with e anal.bb.maxsize
[x] Analyze all flags starting with sym. and entry0 (aa)
[0×00400a30]>
```

We then used the command: ***aa*** *to ask r2 to analyze the program.*

```
[x] Analyze all flags starting with sym. and entry0 (aa)
[0×00400a30]> afl
0×00400400    3 23           sym._init
0×00400428    1 6            fcn.00400428
0×00400430    1 6            fcn.00400430
0×00400438    1 6            fcn.00400438
0×00400448    1 6            fcn.00400448
0×00400450    1 6            fcn.00400450
0×00400460    1 6            fcn.00400460
0×00400488    1 6            fcn.00400488
0×00400498    1 6            fcn.00400498
0×004004a8    1 6            fcn.004004a8
0×004004b0    1 6            fcn.004004b0
0×004004c0    1 6            fcn.004004c0
0×004004d0    1 1            sym.backtrace_and_maps.constprop.1
0×004004d1    1 86           sym.__malloc_assert.constprop.13
0×00400527    1 35           sym.__gconv_release_step.part.1
0×0040054a    1 33           sym.oom
0×00400570    4 128688 → 41    entry4.fini
0×004005a0   90 1157         entry2.init
0×00400a30    1 42           entry0
0×00400a60    1 2            sym._dl_relocate_static_pie
0×00400a70    3 35           sym.deregister_tm_clones
0×00400aa0    3 53           sym.register_tm_clones
0×00400ae0    5 50   → 49    sym.__do_global_dtors_aux
0×00400b20    3 45   → 40    entry1.init
0×00400b4d    1 68           sym.main
0×00400ba0   31 613  → 608   sym.get_common_indeces.constprop.1
0×00400e10   10 1007 → 219   sym.__libc_start_main
0×00401490   25 385  → 381   sym.__libc_check_standard_fds
0×00401620   15 581  → 568   sym.__libc_setup_tls
0×00401870    7 148          sym.__libc_csu_init
0×00401910    5 65   → 74    sym.__libc_csu_fini
0×00401960   11 339          sym.__assert_fail_base
0×00401ac0    1 66           sym.__assert_fail
0×00401b10   94 7517 → 2244  sym.__dcgettext
0×00401b20   10 118  → 101   sym.transcmp
0×00401ba0   93 1988 → 1129  sym.plural_eval
0×00402370  170 3064 → 2981  sym._nl_find_msg
0×00403870   39 661  → 629   sym._nl_find_domain
0×00403b10  308 5366 → 5301  sym._nl_load_domain
0×00405010    2 18   → 24    sym.alias_compare
0×00405030   54 1236 → 1219  sym.read_alias_file
0×00405510   39 453  → 447   sym._nl_expand_alias
0×004056e0   70 1488 → 1402  sym._nl_make_l10nflist
0×00405cb0   21 286  → 255   sym._nl_normalize_codeset
0×00405dd0   40 574  → 536   sym._nl_explode_name
```

*After completing the analysis, we used the command: **afl** to find a list of the functions.*

```
0x00400b20    3 45    → 40    entry1.init
0x00400b4d    1 68            sym.main
0x00400ba0   31 613   → 608   sym.get_common_ind
```

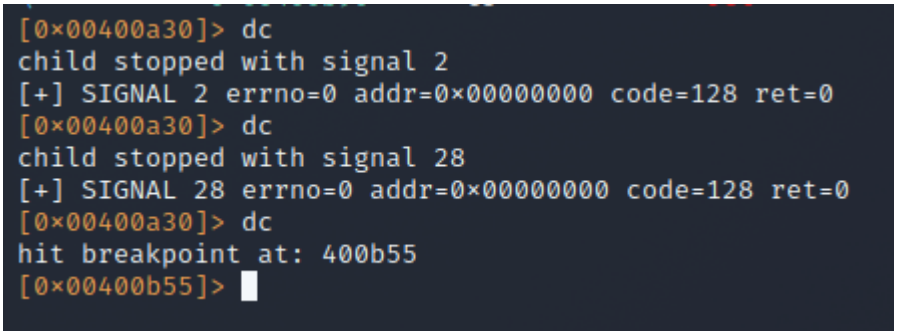
Third, We looked for the sym.main which is the main function.

```
[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 XREF from 0x00400a4d (entry0)
|           0x00400b4d      55              push rbp
|           0x00400b4e      4889e5          mov rbp, rsp
|           0x00400b51      4883ec10        sub rsp, 0x10
|           0x00400b55      c745f4040000.   mov dword [local_ch], 4
|           0x00400b5c      c745f8050000.   mov dword [local_8h], 5
|           0x00400b63      8b55f4          mov edx, dword [local_ch]
|           0x00400b66      8b45f8          mov eax, dword [local_8h]
|           0x00400b69      01d0            add eax, edx
|           0x00400b6b      8945fc          mov dword [local_4h], eax
|           0x00400b6e      8b4dfc          mov ecx, dword [local_4h]
|           0x00400b71      8b55f8          mov edx, dword [local_8h]
|           0x00400b74      8b45f4          mov eax, dword [local_ch]
|           0x00400b77      89c6            mov esi, eax
|           0x00400b79      488d3d881409.   lea rdi, qword str.the_value_of_a_is__d__the_value_of_b_is__d_and_the_val
ue_of_c_is__d ; 0x492008 ; "the value of a is %d, the value of b is %d and the value of c is %d"
|           0x00400b80      b800000000      mov eax, 0
|           0x00400b85      e8f6ea0000      call sym.__printf
|           0x00400b8a      b800000000      mov eax, 0
|           0x00400b8f      c9              leave
\           0x00400b90      c3              ret
[0x00400a30]>
```

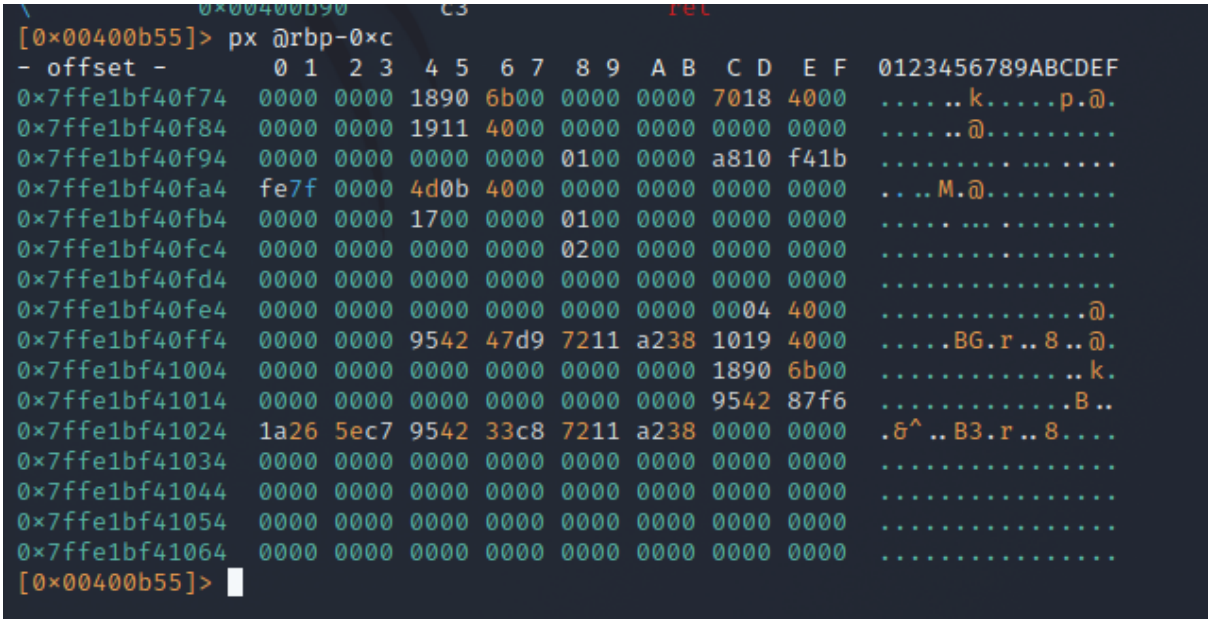
*Forth, we run the command: **pdf @main** to print the disassembly function at main.*

```
[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 XREF from 0x00400a4d (entry0)
|           0x00400b4d      55              push rbp
|           0x00400b4e      4889e5          mov rbp, rsp
|           0x00400b51      4883ec10        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 esi, eax
|           0x00400b79      488d3d881409.   lea rdi, qword str.the_value_of_a_is__d__the_value_of_b_is__d_and_the_val
ue_of_c_is__d ; 0x492008 ; "the value of a is %d, the value of b is %d and the value of c is %d"
|           0x00400b80      b800000000      mov eax, 0
|           0x00400b85      e8f6ea0000      call sym.__printf
|           0x00400b8a      b800000000      mov eax, 0
|           0x00400b8f      c9              leave
\           0x00400b90      c3              ret
[0x00400a30]>
```

Fifth, we set the breakpoint using the command: ***db 0x00400b55*** *and then we reopen the* ***pdf @main.*** *There will be a small capital* ***b (breakpoint indicator)*** *which we wanted to stop it at this point.*



We then ran command: ***dc*** *to run the program.*



Sixth, we ran the command: ***px @rbp-0xc(memory-address,rom the first few lines of @pdf main)*** *to view the contents of the local_ch but it shows that the variable currently doesn't have anything stored in it (it's just 0000).*

```
0x7ffe1bf41064  0000 0000 0000 0000 0000 0000 0000 0000  ................
[0x00400b55]> ds
[0x00400b55]> px @rbp-0xc
- offset -       0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0x7ffe1bf40f74  0400 0000 1890 6b00 0000 0000 7018 4000  ......k.....p.@.
0x7ffe1bf40f84  0000 0000 1911 4000 0000 0000 0000 0000  ......@.........
0x7ffe1bf40f94  0000 0000 0000 0000 0100 0000 a810 f41b  ................
0x7ffe1bf40fa4  fe7f 0000 4d0b 4000 0000 0000 0000 0000  ....M.@.........
0x7ffe1bf40fb4  0000 0000 1700 0000 0100 0000 0000 0000  ................
0x7ffe1bf40fc4  0000 0000 0000 0000 0200 0000 0000 0000  ................
0x7ffe1bf40fd4  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x7ffe1bf40fe4  0000 0000 0000 0000 0000 0000 0004 4000  ..............@.
0x7ffe1bf40ff4  0000 0000 9542 47d9 7211 a238 1019 4000  .....BG.r..8..@.
0x7ffe1bf41004  0000 0000 0000 0000 0000 0000 1890 6b00  ..............k.
0x7ffe1bf41014  0000 0000 0000 0000 0000 0000 9542 87f6  ..............B.
0x7ffe1bf41024  1a26 5ec7 9542 33c8 7211 a238 0000 0000  .&^..B3.r..8....
0x7ffe1bf41034  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x7ffe1bf41044  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x7ffe1bf41054  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x7ffe1bf41064  0000 0000 0000 0000 0000 0000 0000 0000  ................
[0x00400b55]>
```

We then used the command: ***ds*** *to goes to the next instruction. We ran again the command:* ***px @rbp-0xc,*** *and the variables were shown.*

```
0x7ffe1bf41064  0000 0000 00
[0x00400b55]> ds
[0x00400b55]> ds
[0x00400b55]> ds
[0x00400b55]> ds
[0x00400b55]> dr
rax = 0x00000009
rbx = 0x00400400
rcx = 0x0044ba90
rdx = 0x00000004
r8 = 0x01000000
r9 = 0x006bb8e0
r10 = 0x00000015
r11 = 0x00000000
r12 = 0x00401910
r13 = 0x00000000
r14 = 0x006b9018
r15 = 0x00000000
rsi = 0x7ffe1bf410a8
rdi = 0x00000001
rsp = 0x7ffe1bf40f70
rbp = 0x7ffe1bf40f80
rip = 0x00400b6b
rflags = 0x00000206
orax = 0xffffffffffffffff
[0x00400b55]>
```

Q2: What is the command to analyse the program in radare2?

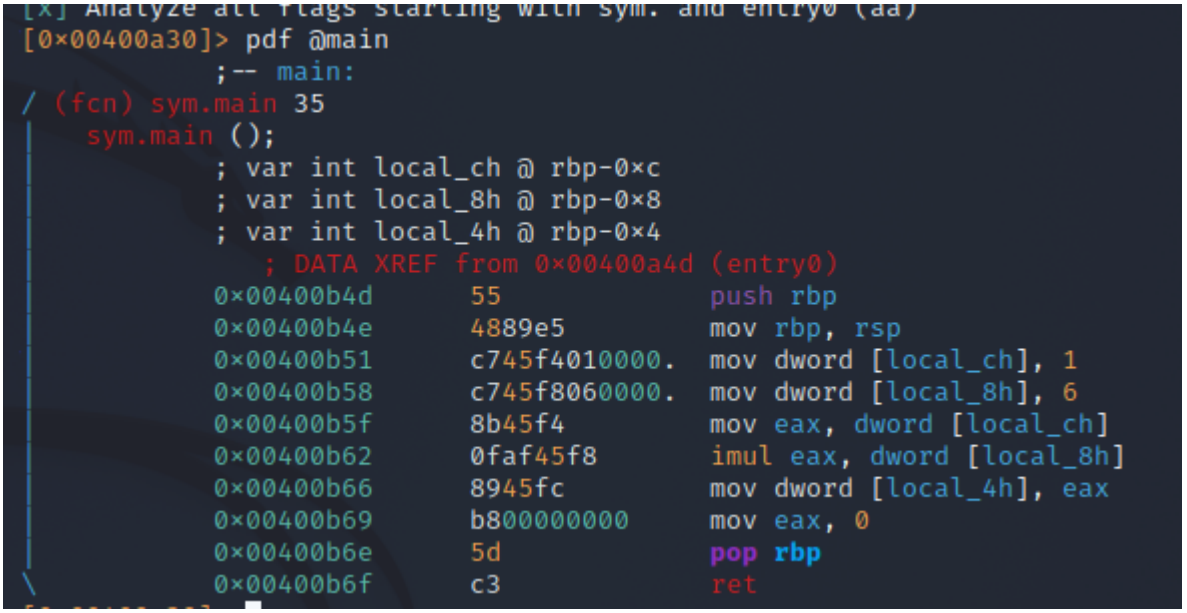
Answer: aa

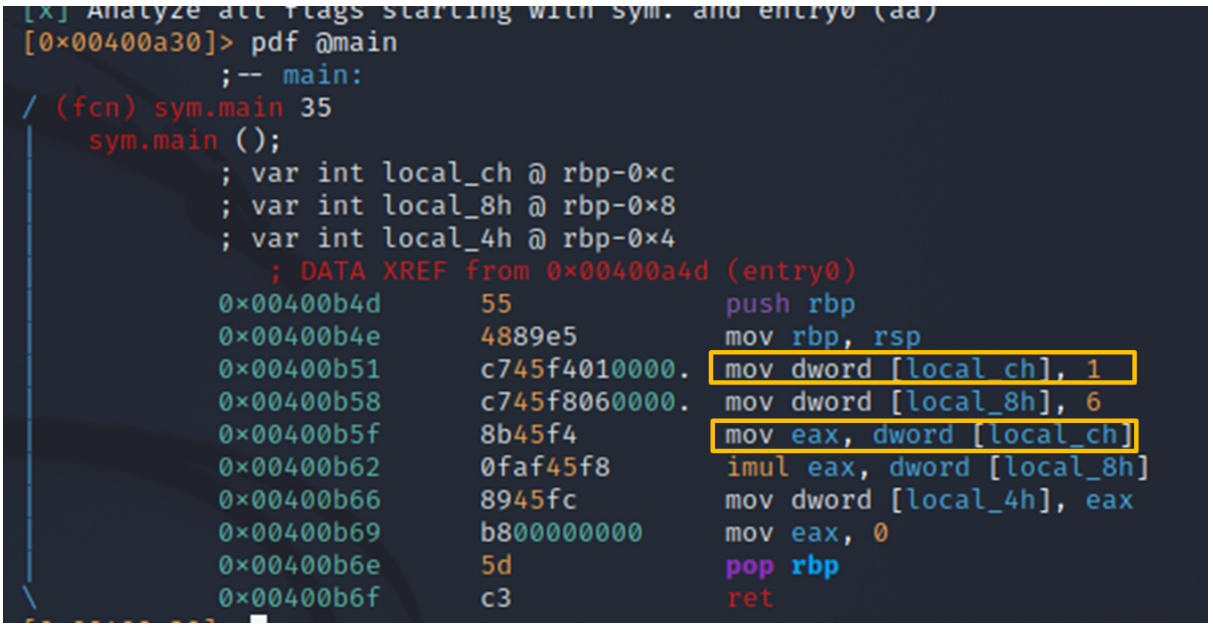
Q3: What is the command to set a breakpoint in radare2?

Answer: db

Q4: What is the command to execute the program until we hit a breakpoint?

Answer: dc





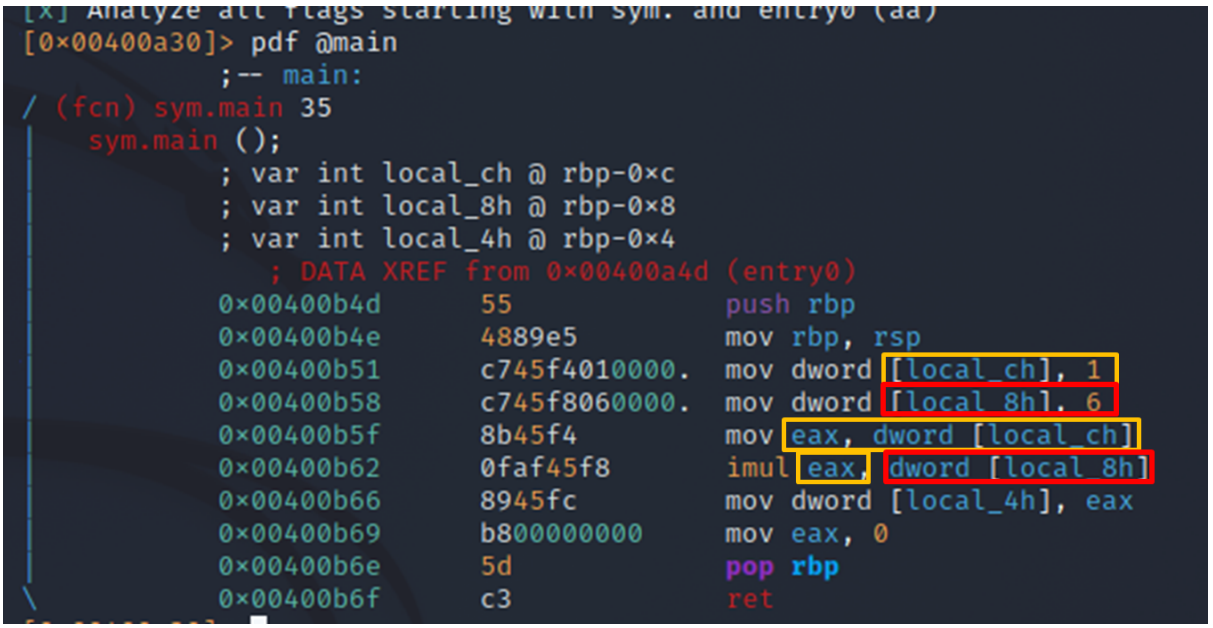
mov *instruction is used to transfer values.*

eax *is sometimes used to store results from functions.*

local_ch=1, eax = 1

Q5: What is the value of local_ch when its corresponding movl instruction is called (first if multiple)?

Answer: 1



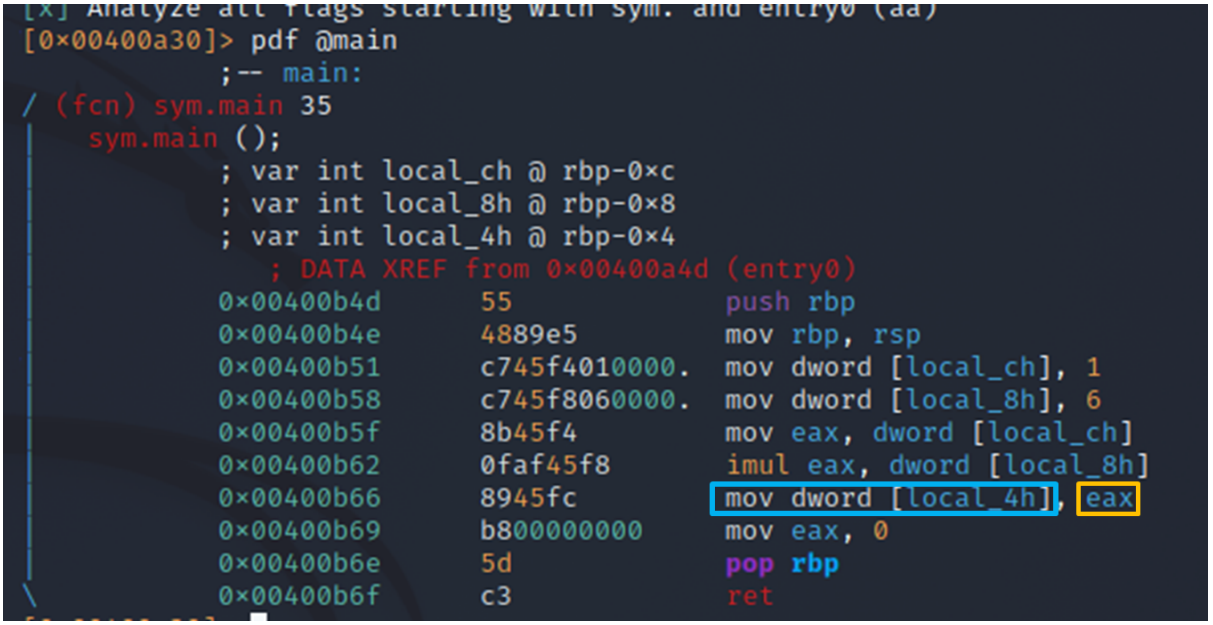
***imull source, destination**: destination = destination * source*

previous eax = 1, Local_8h = 6

*eax = 1*6 = 6*

Q6: What is the value of eax when the imull instruction is called?

Answer: 6



previous eax = 6; local_4h = 6

Q7: What is the value of local_4h before eax is set to 0?

Answer: 6

Thought Process/Methodology:

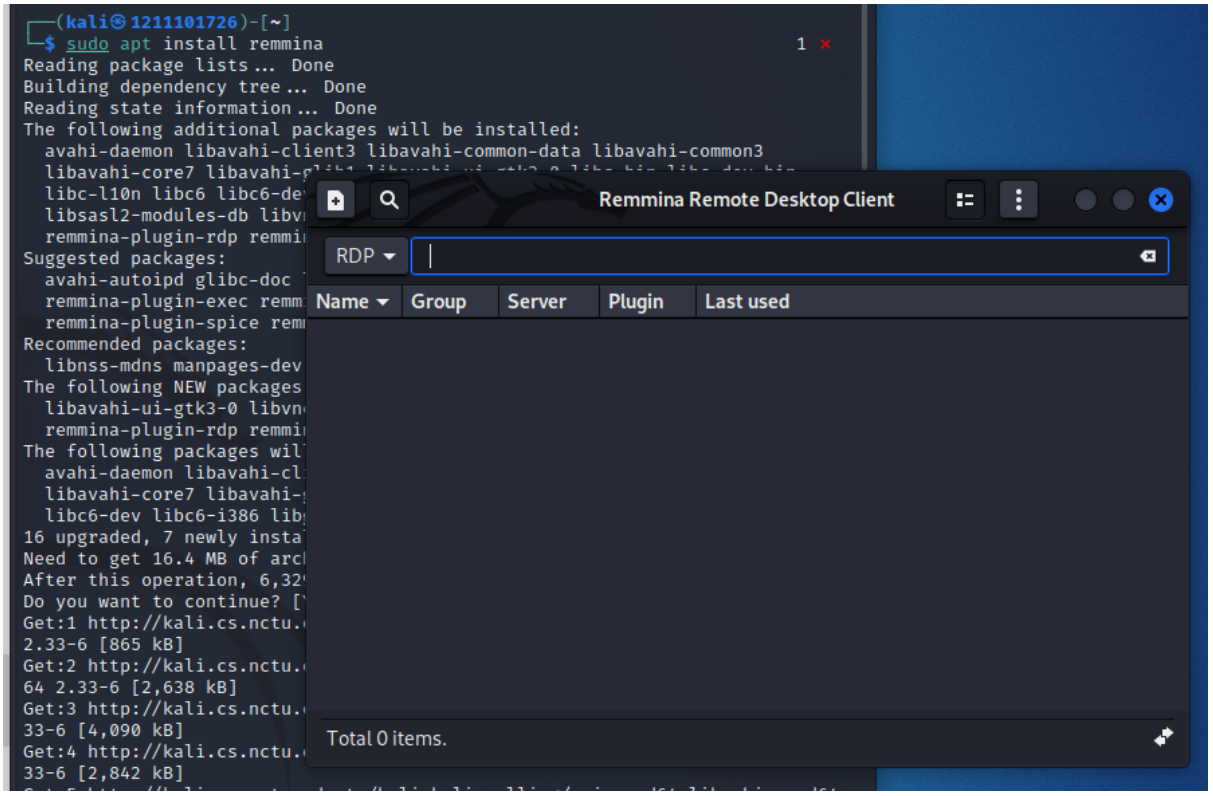
First, we logged in to the target's server. Second, we used the command: **r2 -d ./file1** to open the binary in debugging mode. We then used the command: **aa** to ask r2 to analyze the program. After completing the analysis, we used the command: **afl** to find a list of the functions. Third, We looked for the sym.main which is the main function. Forth, we run the command: **pdf @main** to print the disassembly function at main. Fifth, we set the breakpoint using the command: **db 0x00400b55** and then we reopen the **pdf @main.** There will be a small capital **b (breakpoint indicator)** which we wanted to stop it at this point. We then ran command: **dc** to ran the program. Sixth, we run the command: **px @rbp-0xc(memory-address,rom the first few lines of @pdf main)** to view the contents of the local_ch but it shows that the variable currently doesn't have anything stored in it (it's just

0000). We then used the command: **ds** to goes to the next instruction. We ran again the command: **px @rbp-0xc,** and the variables were shown.

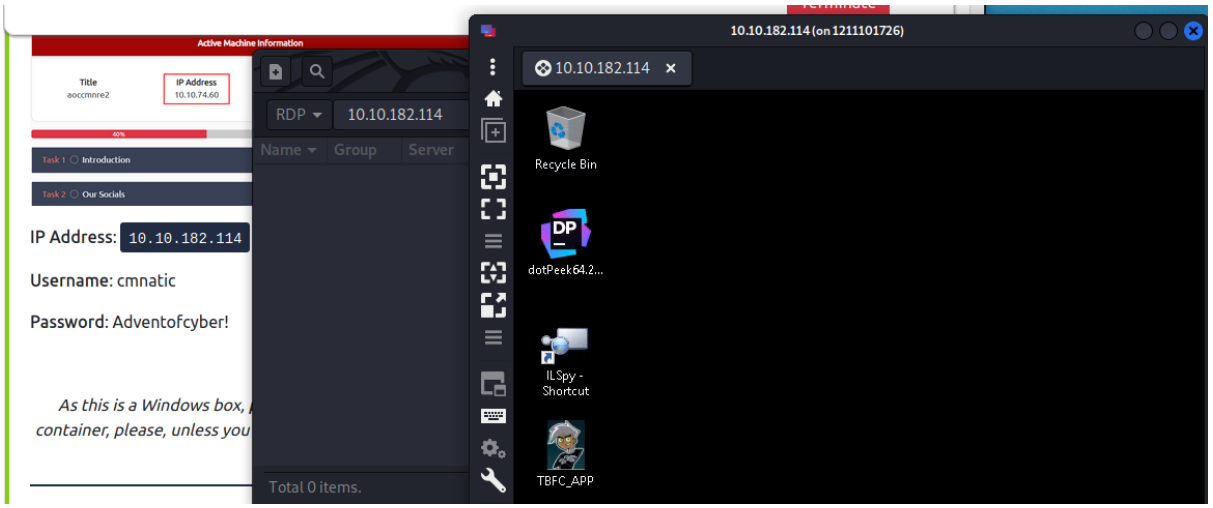
Day 18 - [Reverse Engineering] The Bits of Christmas

Tools used: Kali Linux, Firefox, Terminal, Remmina

Solution/walkthrough:



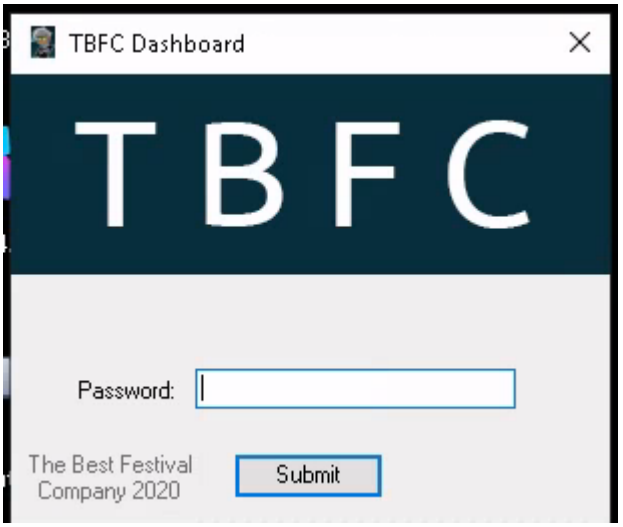
First, we used the command: ***sudo apt install remmina*** *to install remmina.*



Second, we filled out the IP address of the target Instance with the Username and password provided.

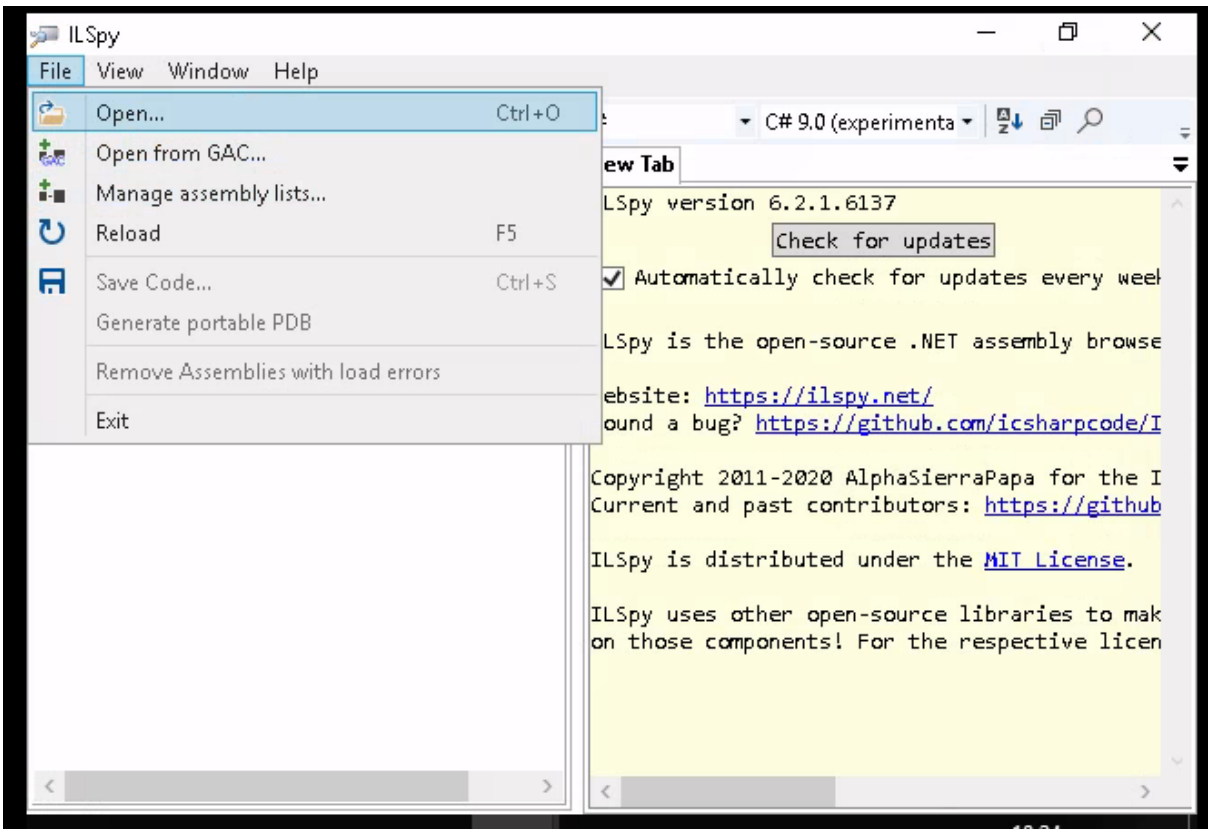
Q1: What is the message that shows up if you enter the wrong password for TBFC_APP?

Answer: Uh Oh! That's the wrong key

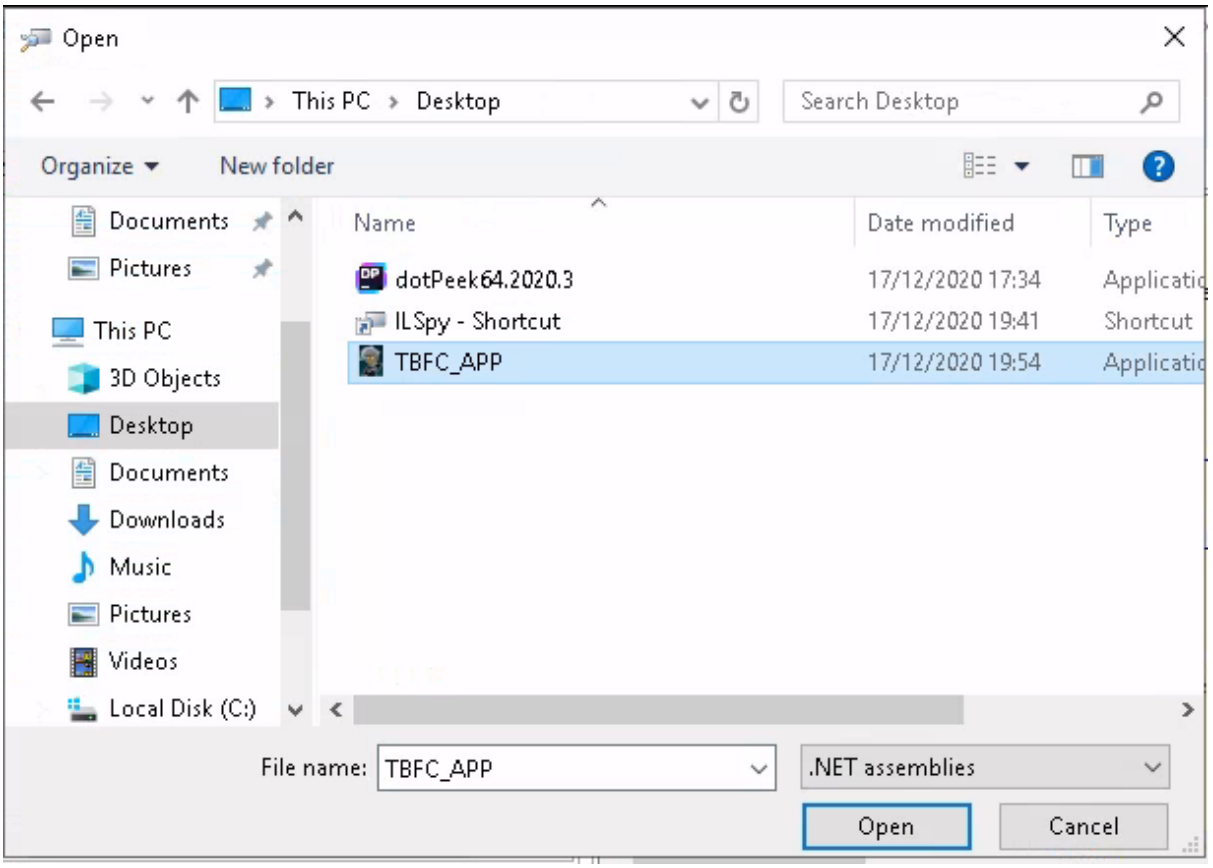


Q2: What does TBFC stand for?

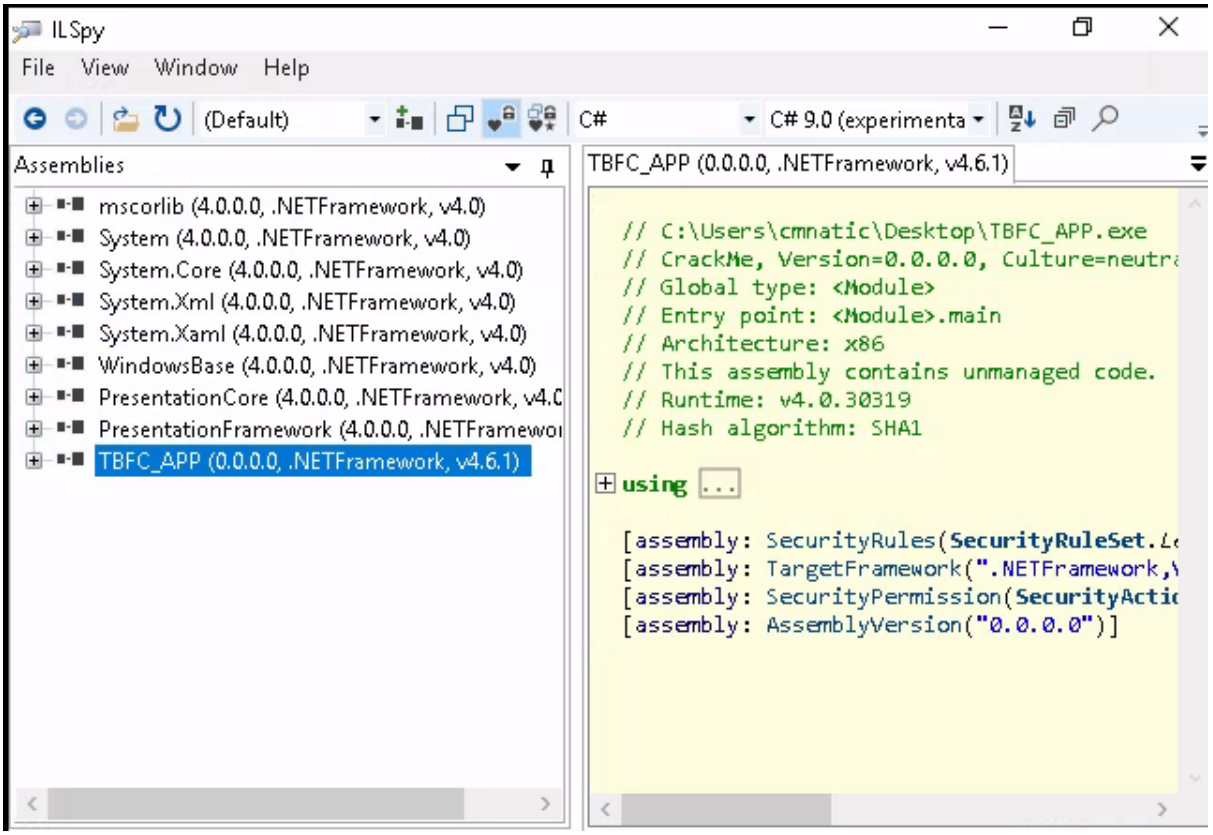
Answer: The Best Festival Company



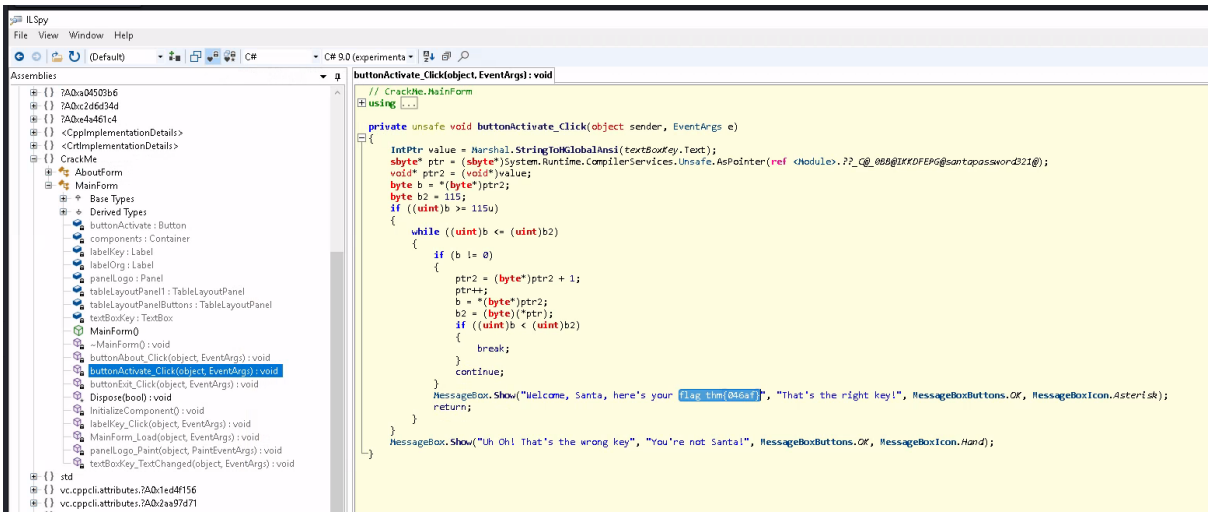
*Third, we opened the ILSpy, and we went to **File>Open…***



*And we opened **TBFC_APP**.*



ILSpy decompiled the code.



After deompiling, we went to CrackMe>MainForm>buttonActivate_Click, and the flag was shown.

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

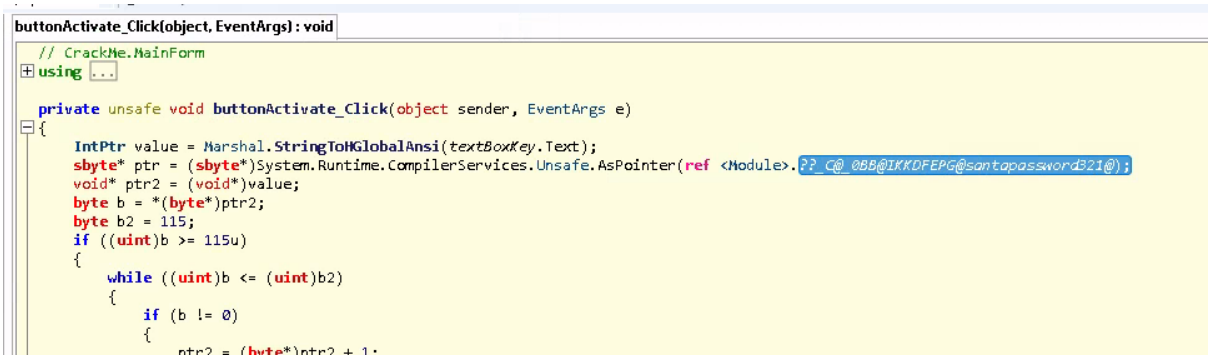
Answer: CrackMe

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

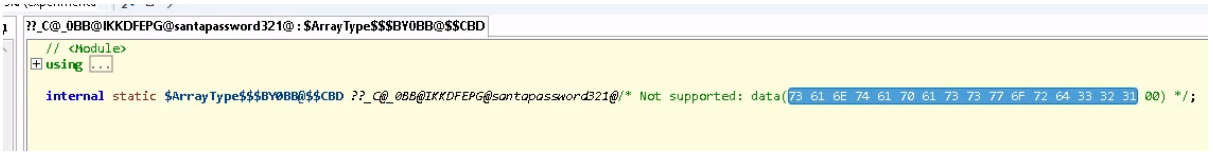
Answer: MainForm

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

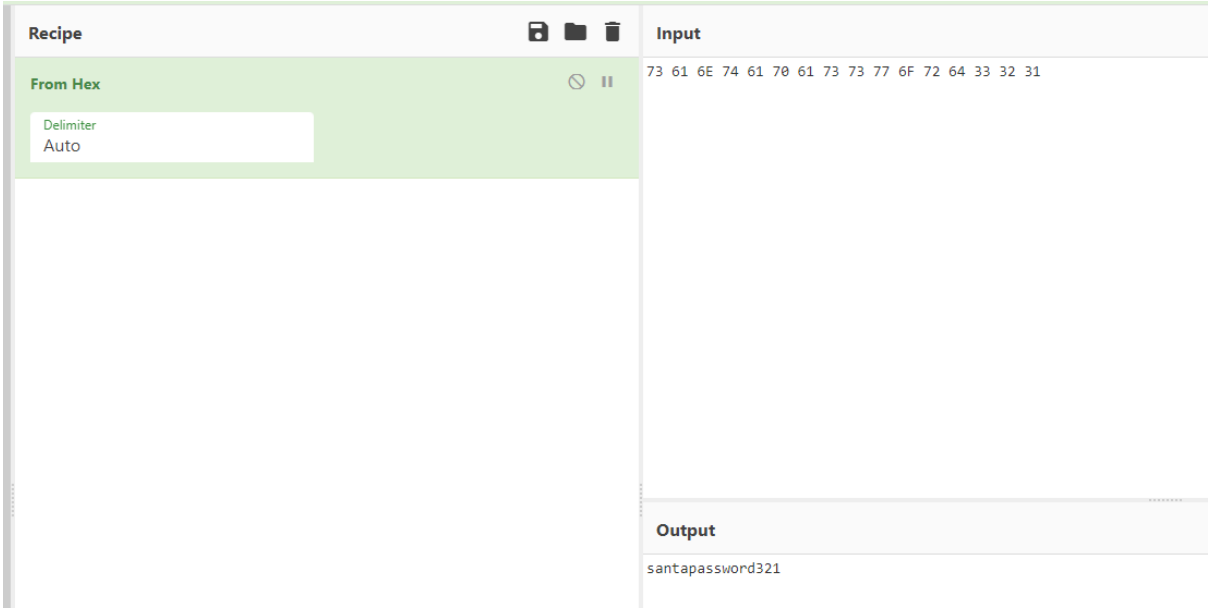
Answer: buttonActivate_Click



Forth, we double clicked on the ***??_C@_0BB@IKKDFEPG@santapassword321@*** *and it brought us to the location.*



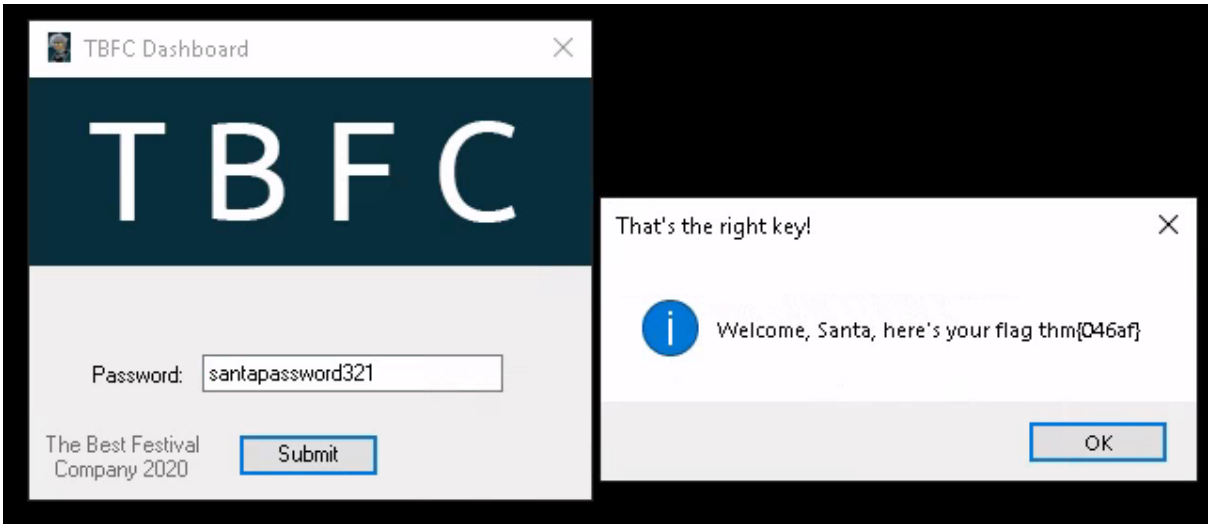
We copied the hexadecinal codes and pasted them into cyberchef to convert them into delimeter.



After converting, we got the password: **santapassword321.**

Q6: What is Santa's password?

Answer: santapassword321



Finally, we submitted the password into the TBFC_APP, and we logged in and got the flag!

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

Answer: thm{046af}

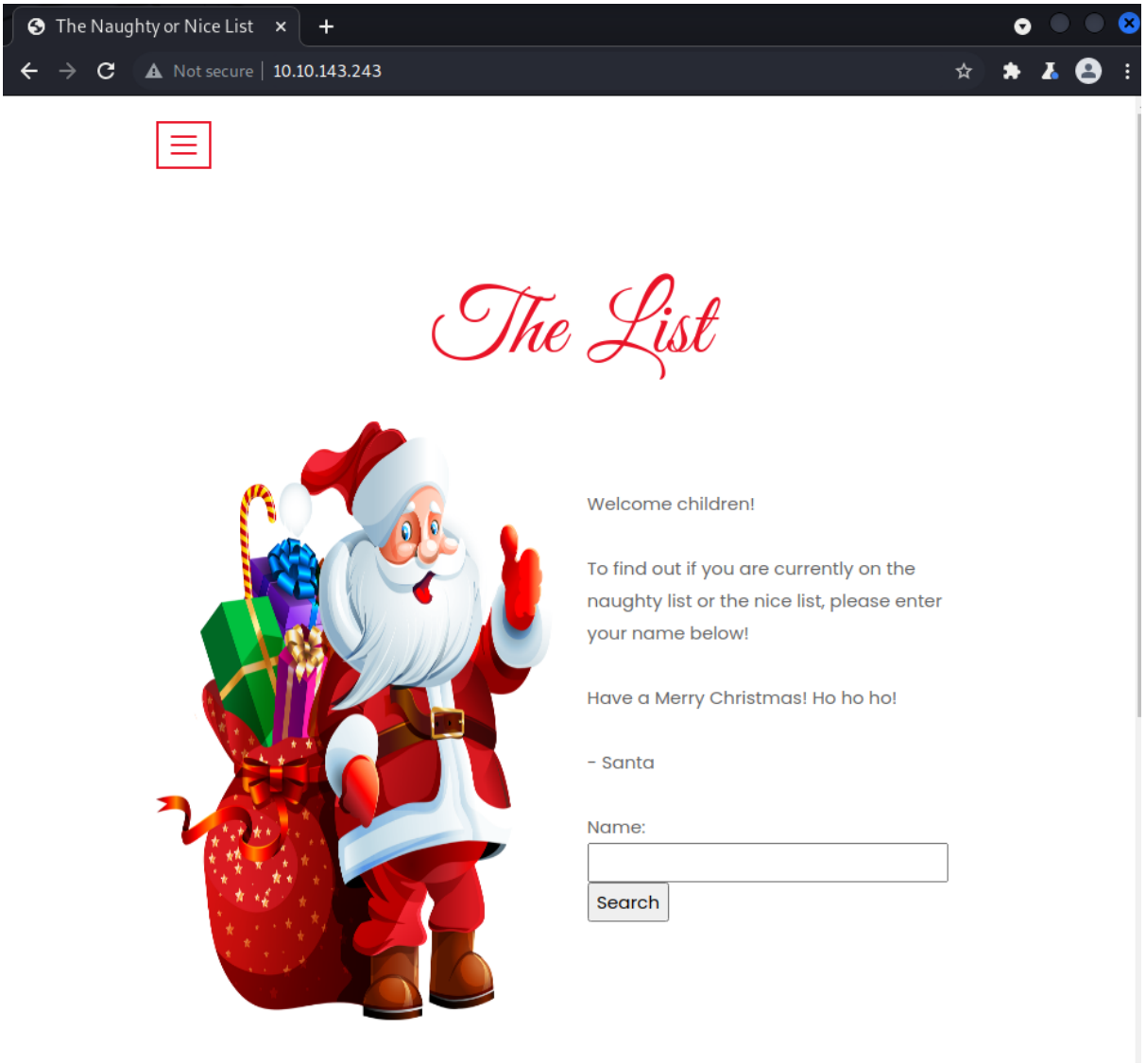
Thought Process/Methodology:

First, we used the command: **sudo apt install remmina** to install remmina. Second, we filled out the IP address of the target Instance with the Username and password provided. Third, we opened the ILSpy, and we went to **File>Open…** And we opened **TBFC_APP**. ILSpy decompiled the code. After deompiling, we went to CrackMe>MainForm>buttonActivate_Click, and the flag was shown. Forth, we double clicked on the **??_C@_0BB@IKKDFEPG@santapassword321@** and it brought us to the location. We copied the hexadecinal codes and pasted them into cyberchef to convert them into delimeter. After converting, we got the password: **santapassword321.** Finally, we submitted the password into the TBFC_APP, and we logged in and got the flag!

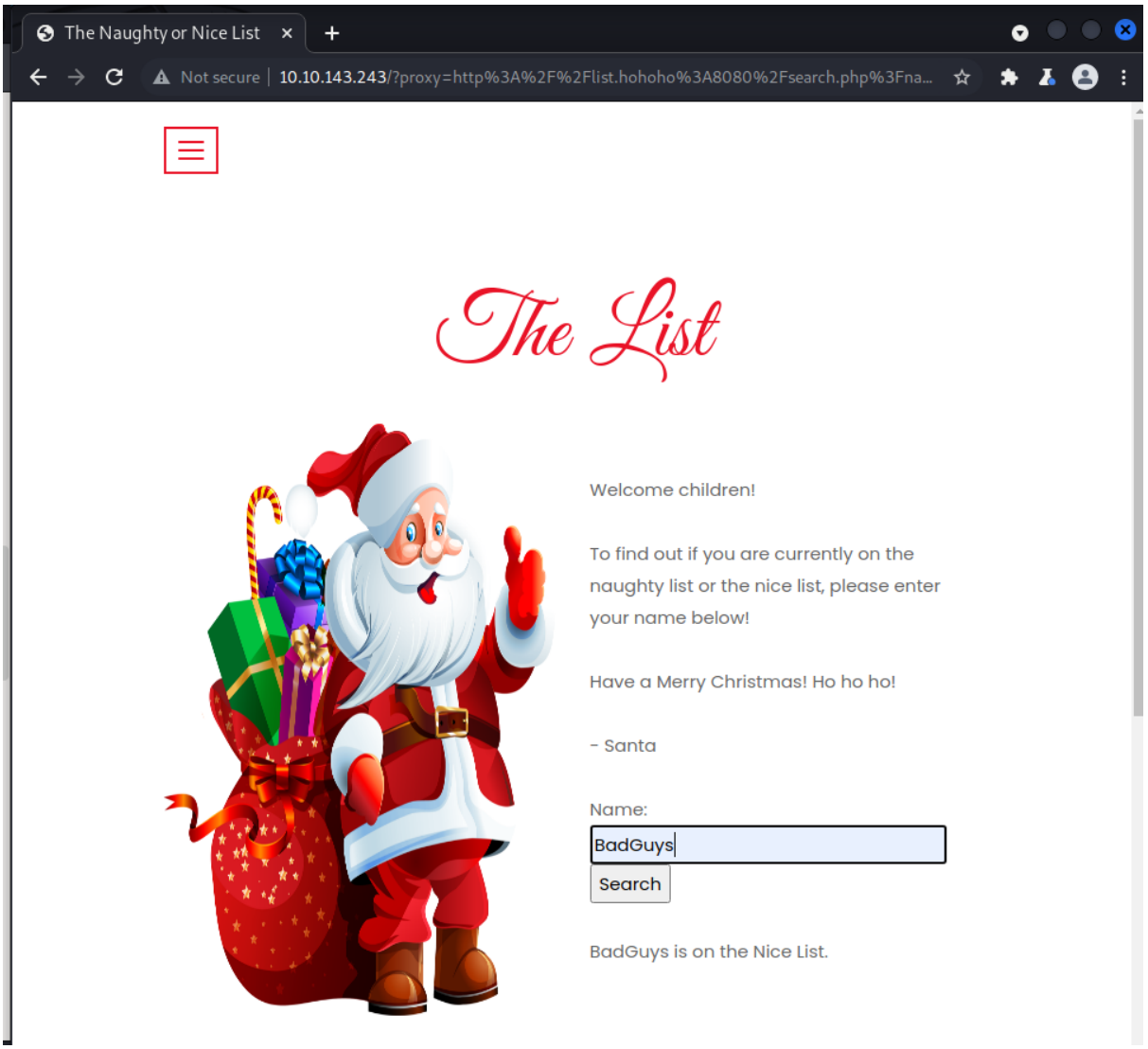
Day 19 - [Web Exploitation] The Naughty or Nice List

Tools used: Kali Linux, Firefox, Terminal, Burp Suite Community Edition

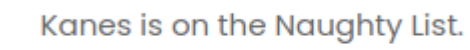
Solution/walkthrough:

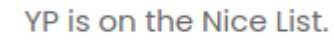


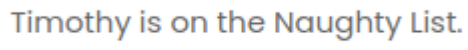
First, having accessed the target machine, we were shown a searching page.

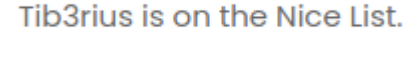


Second, we entered the names in the form and clicked the "Search" button. It told us whether that name is on the Naughty List or the Nice List.









JJ is on the Naughty List.

Ian Chai is on the Nice List.

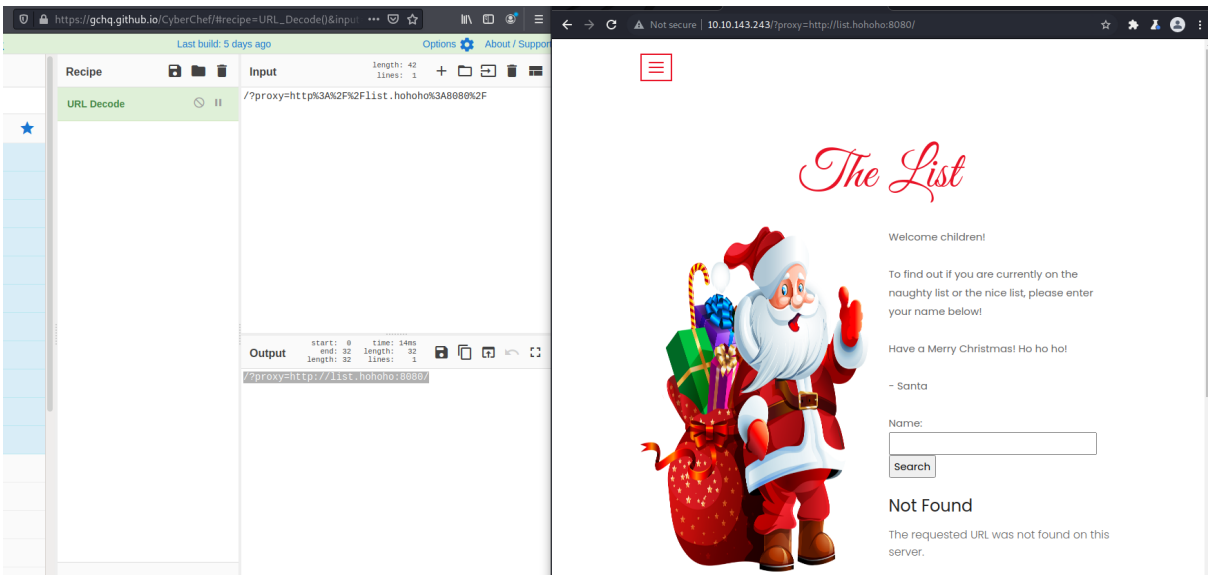
Q1: Which list is this person on

Answer:

Q1: Which list is this person on? * 6 points

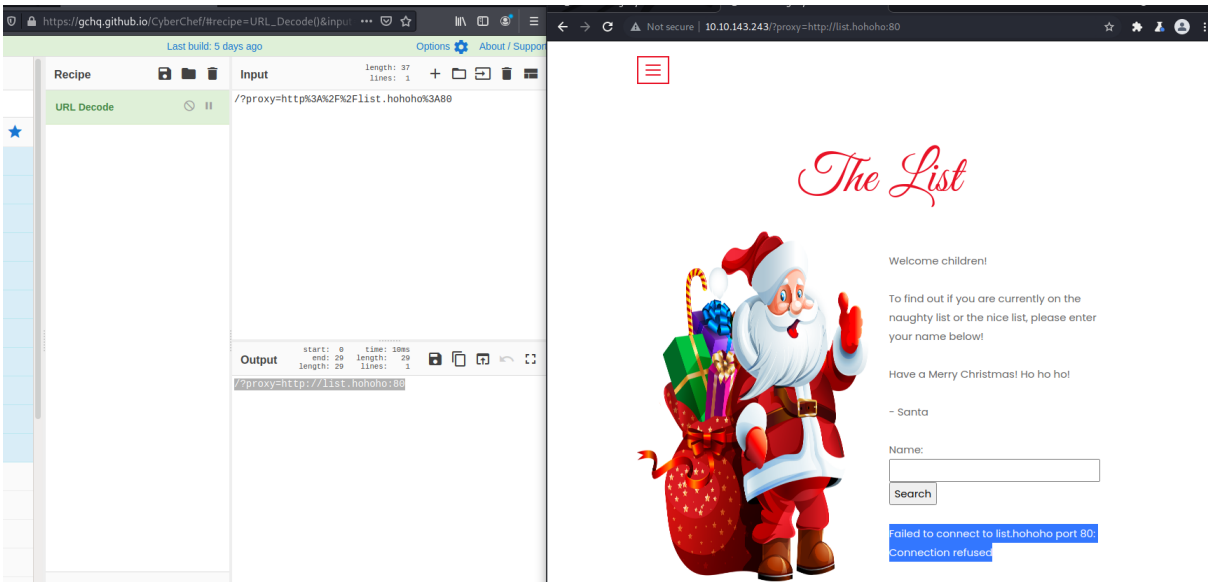
Select the proper words in the proper place of the command: [a] -c -z file,[b] http://[c].xyz/api.[d]?[e]=FUZZ

	Naughty	Nice
Kanes	◉	○
YP	○	◉
Timothy	◉	○
Tib3rius	○	◉
JJ	◉	○
Ian Chai	○	◉



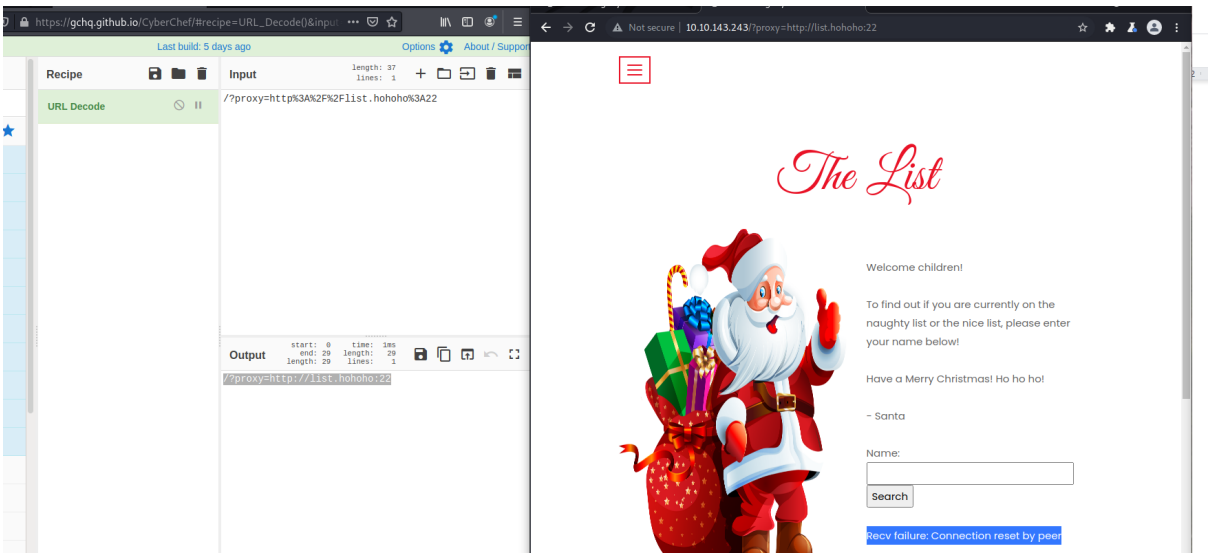
Q2: What is displayed on the page when you use "/?proxy=http%3A%2F%2Flist.hohoho%3A8080%2F"?

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



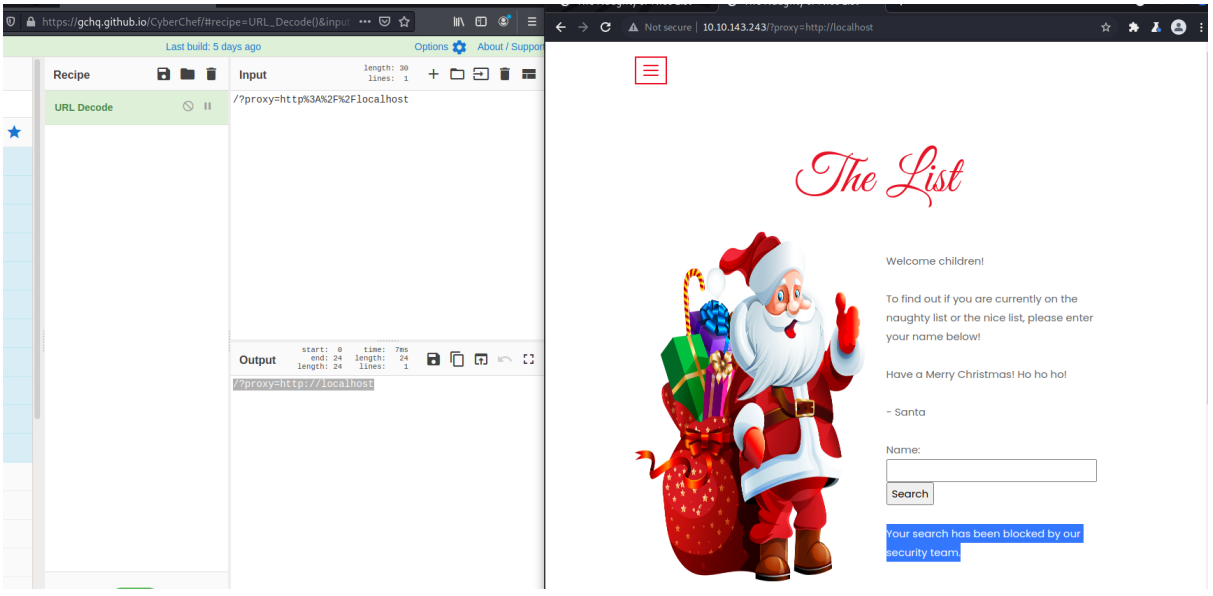
Q3: What is displayed on the page when you use "/?proxy=http%3A%2F%2Flist.hohoho%3A80"?

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



Q4: What is displayed on the page when you use "/?proxy=http%3A%2F%2Flist.hohoho%3A22"?

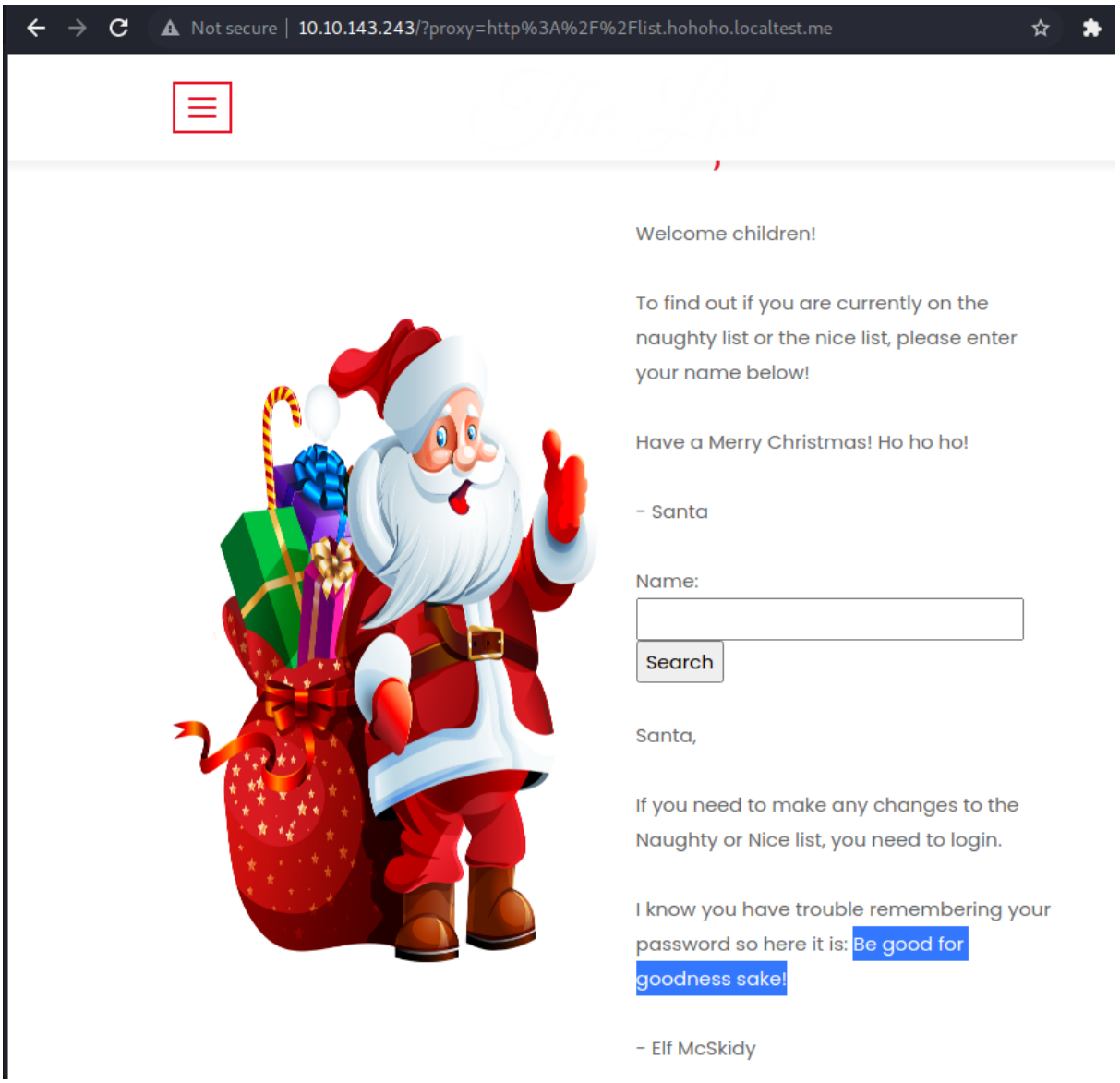
Answer: Recv failure: Connection reset by peer



We tried several hostnames and we found that the developer has implemented a check to ensure that the hostname provided starts with "list.hohoho", and will block any hostnames that don't.

Q5: What is displayed on the page when you use "/?proxy=http%3A%2F%2Flocalhost"?

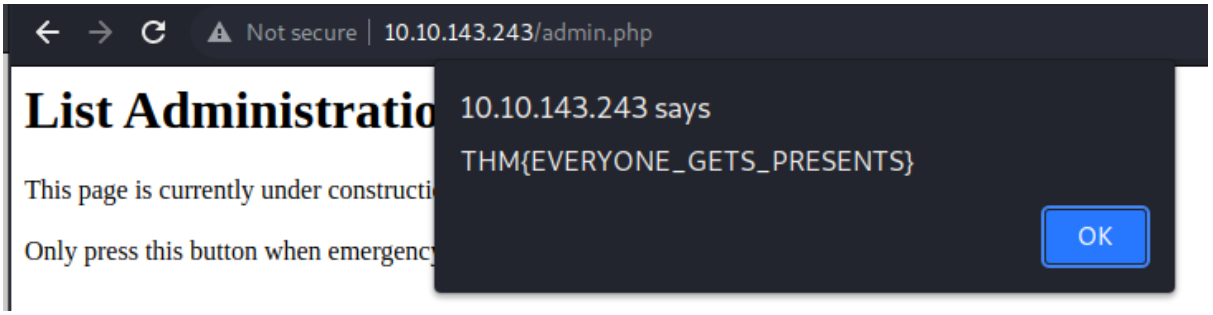
Answer: Your search has been blocked by our security team.



We then used the configured list.hohoho.localtest.me. The password were shown.

Q3: What is Santa's password?

Answer: Be good for goodness sake!



After logging in, the List Administration page was shown. We pressed the DELETE NAUGHTY LIST and the flag appeared.

Q4: What is the challenge flag?

Answer: THM{EVERYONE_GETS_PRESENTS}

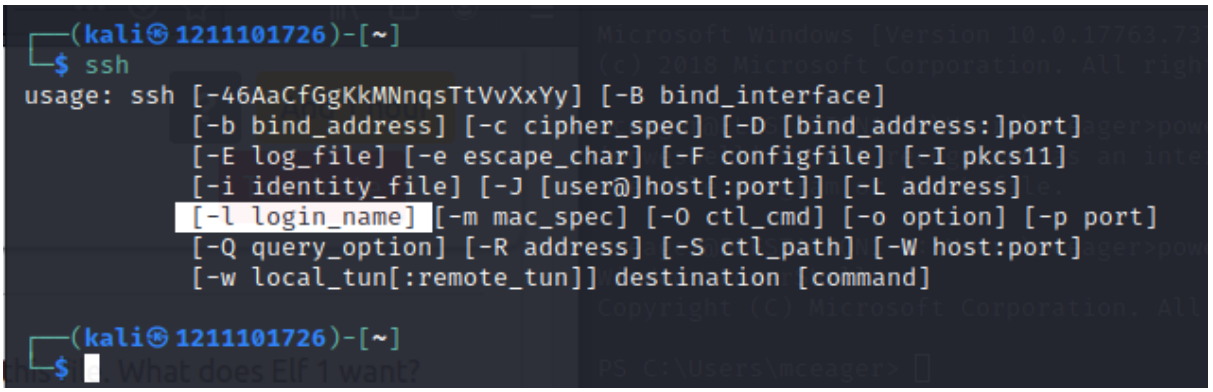
Thought Process/Methodology:

First, having accessed the target machine, we were shown a searching page. Second, we entered the names in the form and clicked the "Search" button. It told us whether that name is on the Naughty List or the Nice List. We tried several hostnames and we found that the developer has implemented a check to ensure that the hostname provided starts with "list.hohoho", and will block any hostnames that don't. We then used the configured list.hohoho.localtest.me. The password were shown. After logging in, the List Administration page was shown. We pressed the DELETE NAUGHTY LIST and the flag appeared.

Day 20 - [Blue Teaming] PowershELlF to the rescue

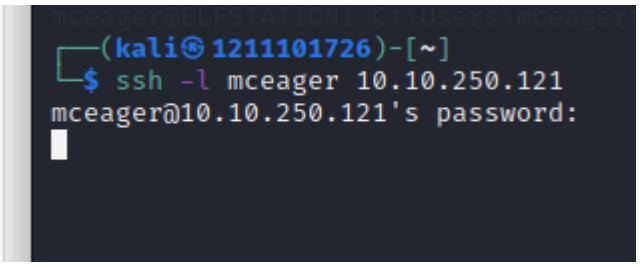
Tools used: Kali Linux, Firefox, Terminal

Solution/walkthrough:

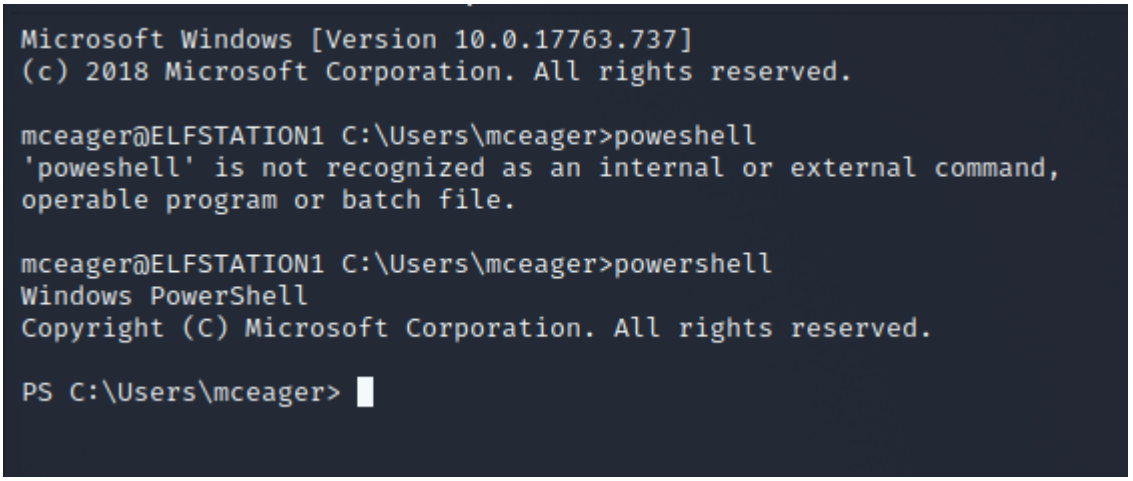


Q1: Check the ssh manual. What does the parameter -l do?

Answer: login name



First, we ran the command: ***ssh -l mceager 10.10.250.121*** *to connect to the remote machine with the password:* ***r0ckStar!***



After logged in, we used the command: ***powershell*** *to launch PoweShell to navigate to the Documents folder.*

```
PS C:\Users\mceager> ls


    Directory: C:\Users\mceager


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-r---        12/7/2020  10:29 AM                3D Objects
d-r---        12/7/2020  10:29 AM                Contacts
d-r---        12/7/2020  11:26 AM                Desktop
d-r---        12/7/2020  11:26 AM                Documents
d-r---        12/7/2020  10:29 AM                Downloads
d-r---        12/7/2020  10:29 AM                Favorites
d-r---        12/7/2020  10:29 AM                Links
d-r---        12/7/2020  10:29 AM                Music
d-r---        12/7/2020  10:29 AM                Pictures
d-r---        12/7/2020  10:29 AM                Saved Games
d-r---        12/7/2020  10:29 AM                Searches
d-r---        12/7/2020  10:29 AM                Videos


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

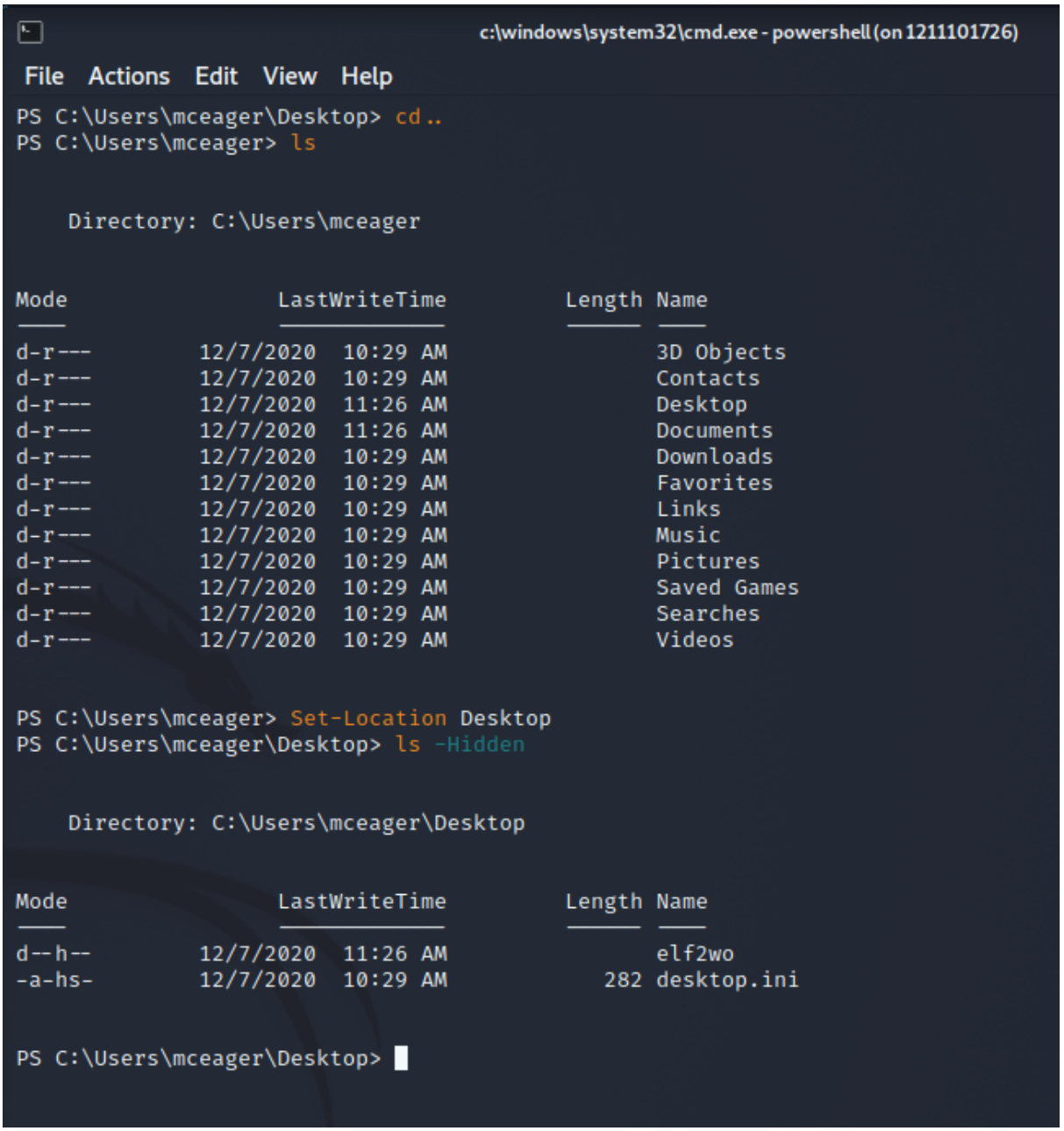
Second, We used the command: ***ls/Get-Content*** *to list out the directories and used the command:* ***Set-Location Documents*** *to change to Documents.*

```
c:\windows\system32\cmd.exe - powershell (on 1211101726)
File  Actions  Edit  View  Help
PS C:\Users\mceager\Documents> ls -File -Hidden


    Directory: C:\Users\mceager\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-hs-        12/7/2020  10:29 AM            402 desktop.ini
-arh--       11/18/2020   5:05 PM             35 e1fone.txt


PS C:\Users\mceager\Documents> cat e1fone.txt
All I want is my '2 front teeth'!!!
PS C:\Users\mceager\Documents>
```

Third, We used the command: ***ls/Get-Content -File -Hidden*** *to list out all the directories and documents and we used the command:* ***cat e1fone.txt*** *to open the e1fone.txt file. There was a message that Elf 1 wants 2 front teeth.*

Q2: Search for the first hidden elf file within the Documents folder. Read the contents of this file. What does Elf 1 want?

Answer: 2 front teeth



Forth, we used the command: ***cd..*** *to return back to the mceager directory. Then, we changed directory to Desktop and in Desktop directory, we used the command:* ***ls/Get-Content -Hidden*** *to get hidden items.*

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


    Directory: C:\Users\mceager\Desktop


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

We also used the command: ***ls -Hidden -Directory*** *to know that elf2wo is a directory and we used the command:* ***cd elf2wo*** *to change directory.*

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

We listed the files and then opened the e70smsW10Y4k.txt using the command: ***cat e70smsW10Y4k.txt.*** *There was a message that Scrooged was the movie that Elf 2 wanted.*

Q3: Search on the desktop for a hidden folder that contains the file for Elf 2. Read the contents of this file. What is the name of that movie that Elf 2 wants?

Answer: Scrooged

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


    Directory: C:\Windows\System32


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

*Fifth, we changed directory to **System32** and in **System32** directory, we used the command: **ls/Get-Content -Hidden -Directory -Filter "*3*"** to get hidden items.*

Q4: Search the Windows directory for a hidden folder that contains files for Elf 3. What is the name of the hidden folder? (This command will take a while)

Answer: 3lfthr3e

```
PS C:\Windows\System32\3lfthr3e> ls 1.txt | Measure-Object
ls : Could not find item C:\Windows\System32\3lfthr3e\1.txt.
At line:1 char:1
+ ls 1.txt | Measure-Object
+ ~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (C:\Windows\System32\3lfthr3e\1.txt:String) [Get-ChildItem], IOExcep
   tion
    + FullyQualifiedErrorId : ItemNotFound,Microsoft.PowerShell.Commands.GetChildItemCommand


Count    : 0
Average  :
Sum      :
Maximum  :
Minimum  :
Property :


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


Count    : 9999
Average  :
Sum      :
Maximum  :
Minimum  :
Property :
```

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

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

*We then used the command: **Get-Content 1.txt | Measure-Object -Word** to get the number of words contained within the 1.txt. It showed that 1.txt had **9999** words.*

Q5: How many words does the first file contain?

Answer: 9999

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

*Besides that. we used the command: **(Get-Content -Path 1.txt)[551,6991]** to get the exact position of a string within the 1.txt which the words are **Red** and **Ryder**.*

Q6: What 2 words are at index 551 and 6991 in the first file?

Answer: Red Ryder

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

redryderbbgun
```

Finally, we used the command: ***Get-Content 2.txt | Select-String -Pattern "redryder"*** *to search a 2.txt for a pattern "redryder". We then found the full answer: Elf 3 want a* ***redryderbbgun****.*

Q7: This is only half the answer. Search in the 2nd file for the phrase from the previous question to get the full answer. What does Elf 3 want? (use spaces when submitting the answer)

Answer: redryderbbgun

Thought Process/Methodology:

First, we ran the command: **ssh -l mceager 10.10.250.121** to connect to the remote machine with the password: **r0ckStar! .** After logged in, we used the command: **powershell** to launch PoweShell to navigate to the Documents folder. Second, We used the command: **ls/Get-Content** to list out the directories and used the command: **Set-Location Documents** to change to Documents. Third, We used the command: **ls/Get-Content -File -Hidden** to list out all the directories and documents and we used the command: **cat e1fone.txt** to open the e1fone.txt file. There was a message that Elf 1 wants 2 front teeth. Forth, we used the command: **cd..** to return back to the mceager directory. Then, we changed directory to Desktop and in Desktop directory, we used the command: **ls/Get-Content -Hidden** to get hidden items. We also used the command: **ls -Hidden -Directory** to know that elf2wo is a directory and we used the command: **cd elf2wo** to change directory. We listed the files and then opened the e70smsW10Y4k.txt using the command: **cat e70smsW10Y4k.txt.** There was a message that Scrooged was the movie that Elf 2 wanted. Fifth, we changed directory to **System32** and in **System32** directory, we used the command: **ls/Get-Content -Hidden -Directory -Filter "*3*"** to get hidden items. We then used the command: **Get-Content 1.txt | Measure-Object -Word** to get the number of words contained within the 1.txt. It showed that 1.txt had **9999** words. Besides that. we used the command: **(Get-Content -Path 1.txt)[551,6991]** to get the exact position of a string within the 1.txt which the words are **Red** and **Ryder**. Finally, we used the command: **Get-Content 2.txt | Select-String -Pattern "redryder"** to search a 2.txt for a pattern "redryder". We then found the full answer: Elf 3 want a **redryderbbgun**.