PSP0201 Week 4 Writeup

Group Name: Hacktocrats

Members

ID	Name	Role
1211103194	NUR FARAHIYA AIDA BINTI ABD RAZAK	Leader
1211103602	NUR ALIA AMELISA SYAZREEN BINTI MOHD SULEI	Member
1211103430	AINA SOFEA BINTI AMIER HAMZAH	Member
1211103237	NURUL AIN BINTI KAMARUDIN	Member

DAY 11 : Networking -The Rogue Gnome: Prelude
Tools used: Terminal, Google Chrome

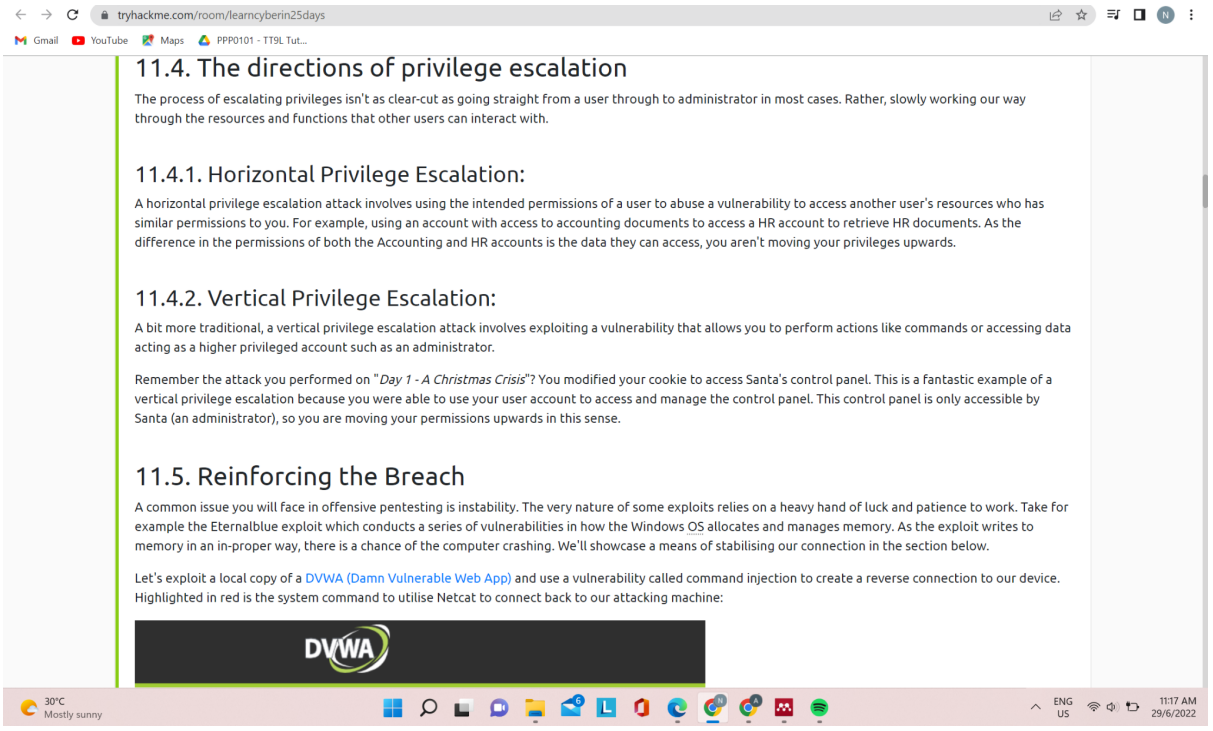
Solution/Walkthrough:

Question 1, question 2 and question 3 are all referring to the same screenshot provided below.

Question 1:
The privilege escalation that is used is vertical escalation as we used a user account to access a higher-positioned account which is the administrator.

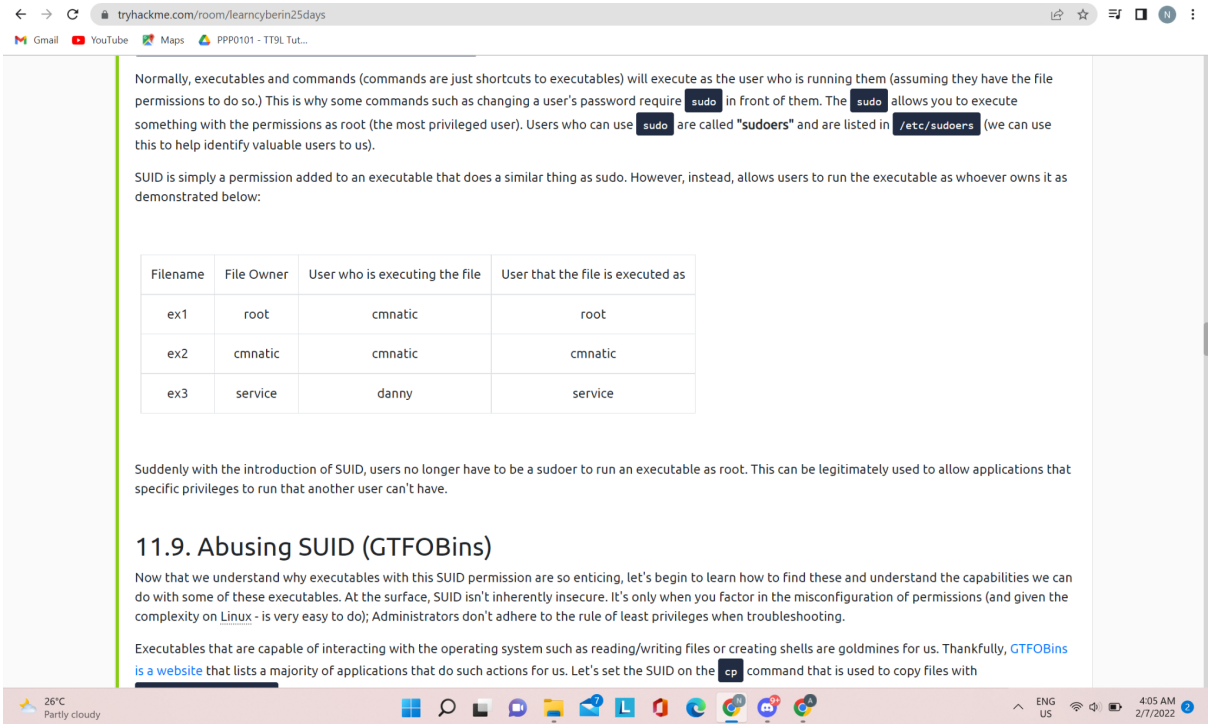
Question 2:
It is a vertical privilege escalation as we managed to pivot to an account which can use sudo commands. We know that only superusers can use sudo commands. Superuser here is the same as administrator in WIndows.

Question 3:
It is a horizontal privilege escalation as we pivot to an account which has the similar permissions and access to an account with almost similar privileges.



Question 4(a):

The users who can use sudo are called “sudoers” in the explanation in TryHackMe.



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).

SUID is simply a permission added to an executable that does a similar thing as sudo. However, instead, allows users to run the executable as whoever owns it as demonstrated below:

Filename	File Owner	User who is executing the file	User that the file is executed as
ex1	root	cmnatic	root
ex2	cmnatic	cmnatic	cmnatic
ex3	service	danny	service

Suddenly with the introduction of SUID, users no longer have to be a sudoer to run an executable as root. This can be legitimately used to allow applications that specific privileges to run that another user can't have.

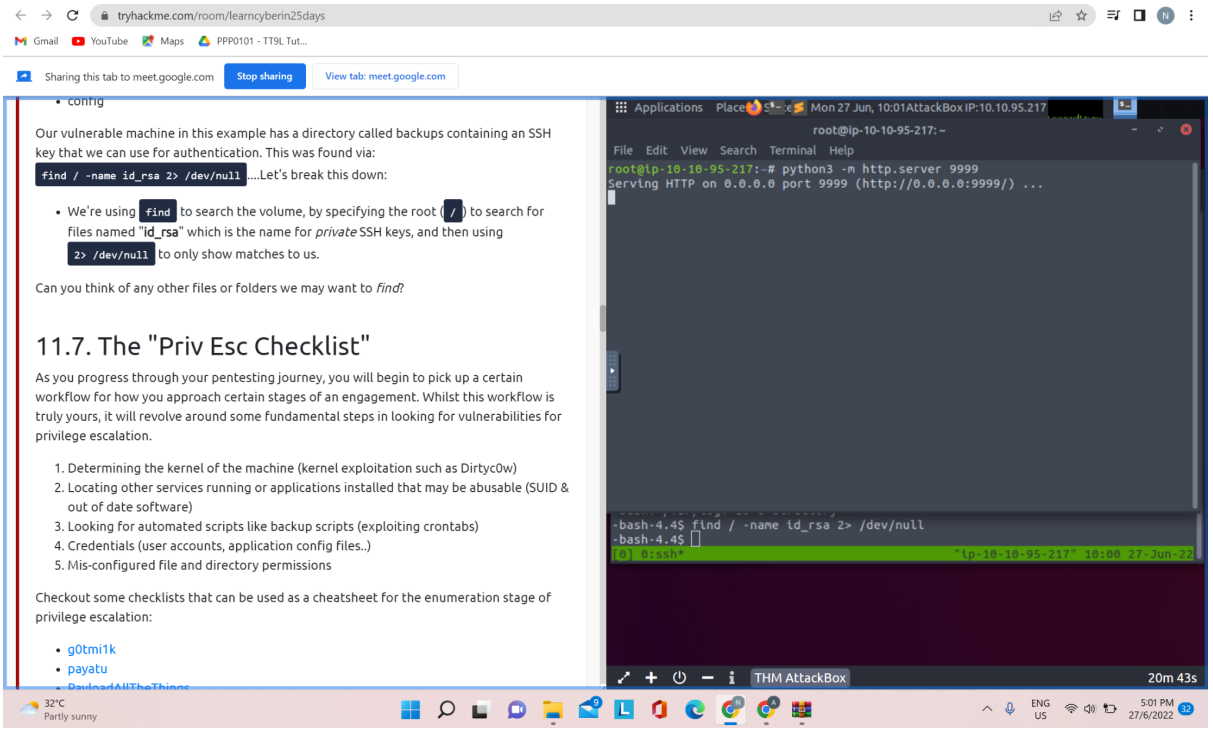
11.9. Abusing SUID (GTFOBins)

Now that we understand why executables with this SUID permission are so enticing, let's begin to learn how to find these and understand the capabilities we can do with some of these executables. At the surface, SUID isn't inherently insecure. It's only when you factor in the misconfiguration of permissions (and given the complexity on Linux - is very easy to do); Administrators don't adhere to the rule of least privileges when troubleshooting.

Executables that are capable of interacting with the operating system such as reading/writing files or creating shells are goldmines for us. Thankfully, GTFOBins is a website that lists a majority of applications that do such actions for us. Let's set the SUID on the `cp` command that is used to copy files with

Question 4(b):

Use the command recommended in the explanation on TryHackMe.



Our vulnerable machine in this example has a directory called backups containing an SSH key that we can use for authentication. This was found via:
`find / -name id_rsa 2> /dev/null` ....Let's break this down:

- We're using `find` to search the volume, by specifying the root (`/`) to search for files named "**id_rsa**" which is the name for *private* SSH keys, and then using `2> /dev/null` to only show matches to us.

Can you think of any other files or folders we may want to *find*?

11.7. The "Priv Esc Checklist"

As you progress through your pentesting journey, you will begin to pick up a certain workflow for how you approach certain stages of an engagement. Whilst this workflow is truly yours, it will revolve around some fundamental steps in looking for vulnerabilities for privilege escalation.

1. Determining the kernel of the machine (kernel exploitation such as Dirtyc0w)
2. Locating other services running or applications installed that may be abusable (SUID & out of date software)
3. Looking for automated scripts like backup scripts (exploiting crontabs)
4. Credentials (user accounts, application config files..)
5. Mis-configured file and directory permissions

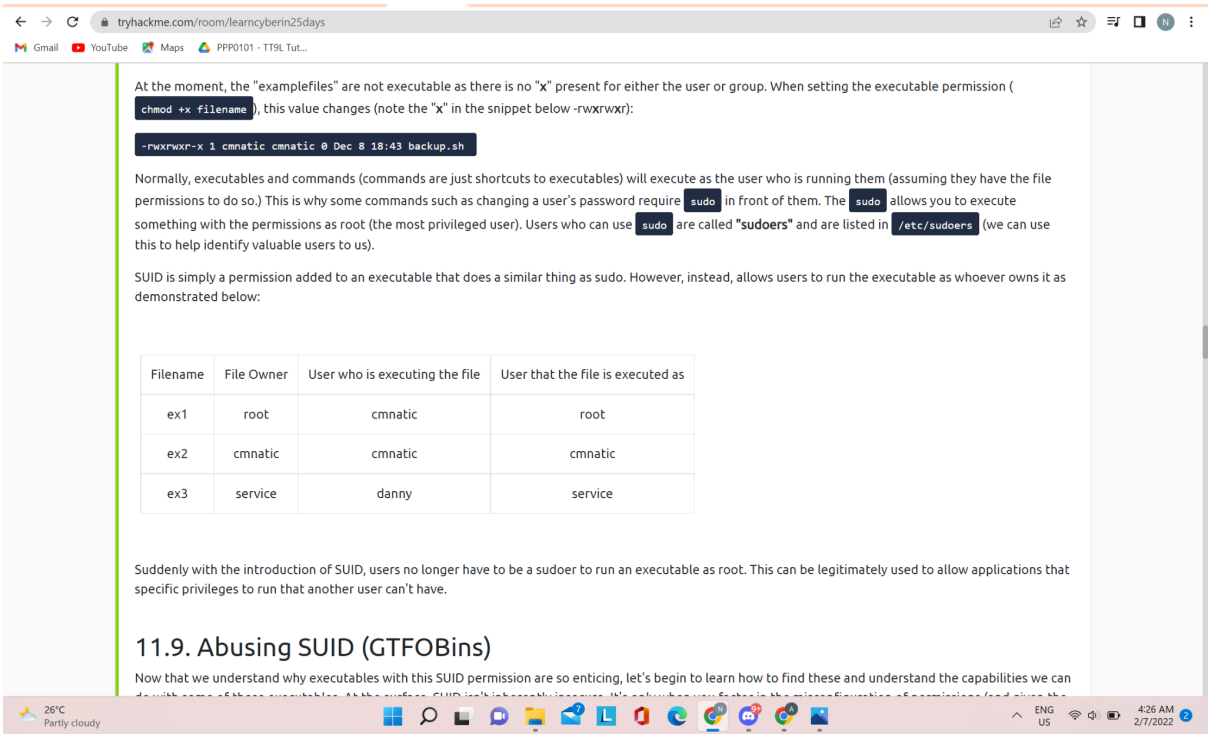
Checkout some checklists that can be used as a cheatsheet for the enumeration stage of privilege escalation:

- g0tmi1k
- payatu

```
root@ip-10-10-95-217:~# python3 -m http.server 9999
Serving HTTP on 0.0.0.0 port 9999 (http://0.0.0.0:9999/) ...
-bash-4.4$ find / -name id_rsa 2> /dev/null
-bash-4.4$
```

Question 5:

We use the command chmod +x find.sh to execute.



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 -rwxrwxr):

```
-rwxrwxr-x 1 cmnatic cmnatic 0 Dec 8 18:43 backup.sh
```

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).

SUID is simply a permission added to an executable that does a similar thing as sudo. However, instead, allows users to run the executable as whoever owns it as demonstrated below:

Filename	File Owner	User who is executing the file	User that the file is executed as
ex1	root	cmnatic	root
ex2	cmnatic	cmnatic	cmnatic
ex3	service	danny	service

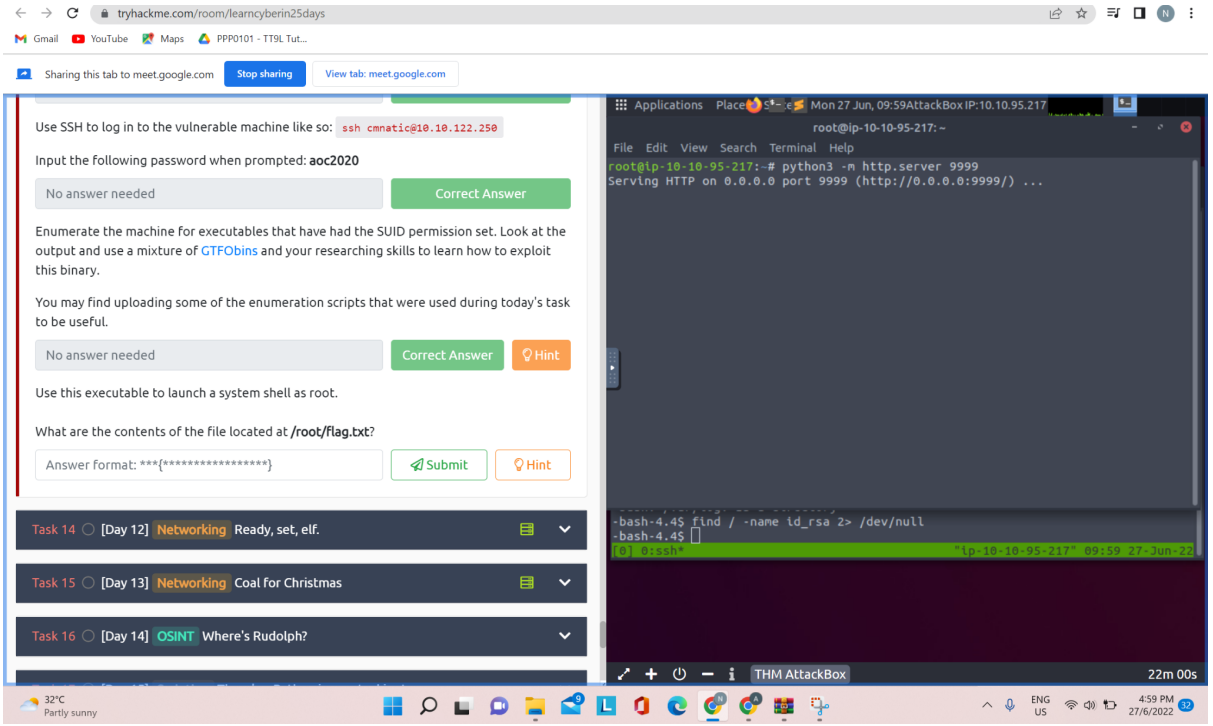
Suddenly with the introduction of SUID, users no longer have to be a sudoer to run an executable as root. This can be legitimately used to allow applications that specific privileges to run that another user can't have.

11.9. Abusing SUID (GTFOBins)

Now that we understand why executables with this SUID permission are so enticing, let's begin to learn how to find these and understand the capabilities we can

Question 7:

We use python3 -m http.server 9999 since our port number is 9999.



Use SSH to log in to the vulnerable machine like so: `ssh cmnatic@10.10.122.250`

Input the following password when prompted: **aoc2020**

No answer needed — Correct Answer

Enumerate the machine for executables that have had the SUID permission set. Look at the output and use a mixture of GTFOBins and your researching skills to learn how to exploit this binary.

You may find uploading some of the enumeration scripts that were used during today's task to be useful.

No answer needed — Correct Answer — Hint

Use this executable to launch a system shell as root.

What are the contents of the file located at **/root/flag.txt**?

Answer format: ***{**********} — Submit — Hint

Task 14 [Day 12] Networking Ready, set, elf.

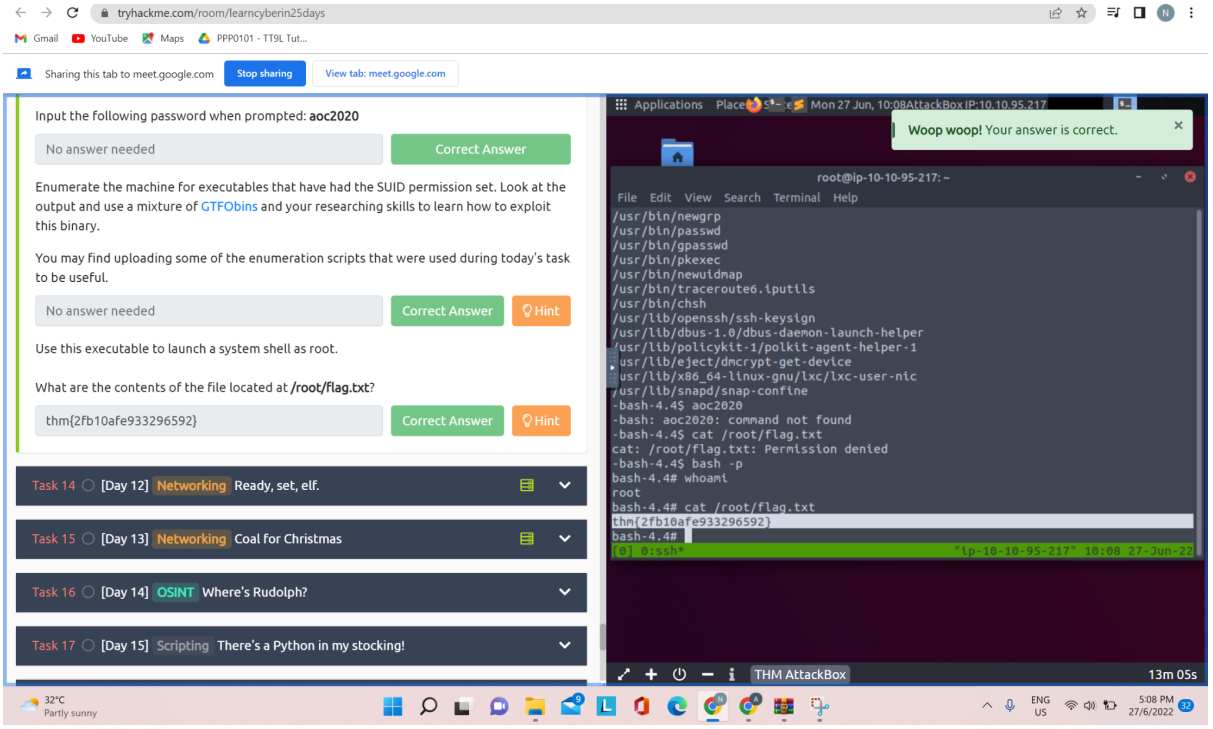
Task 15 [Day 13] Networking Coal for Christmas

Task 16 [Day 14] OSINT Where's Rudolph?

```
root@ip-10-10-95-217:~# python3 -m http.server 9999
Serving HTTP on 0.0.0.0 port 9999 (http://0.0.0.0:9999/) ...
-bash-4.4$ find / -name id_rsa 2> /dev/null
-bash-4.4$
```

Question 8:

After identifying who we are logging in as, we can be sure to read the contents of the file located at/root/flag.txt. We will discover the flag here.



Thought Process/ Methodology:

In this task, we will be escalating our privilege as pentester. To escalate the privilege, we will use a user account to access data at higher positioned accounts such as administrators. Firstly, open terminal and type ssh cmnatic@10.10.234.170 to log in to the machine. If a password is required, simply enter aoc2020. After logging in, we can use whoami to see who we are logging in as. It will show that we are cmnatic. Type id and we can see that our user id is also cmnatic. We have to use SUID so that we can add permission to run as administrator. Enter find / -perm -4000 2>/dev/null to find SUID. We will use/bash to run as root. Go to GTFOBins and search bash. Under the SUID section, we can find the command to be used for SUID for bash. As we enter ./bash -p on the terminal, we are now escalating our privilege. To confirm, enter id to see our user id has changed from cmnatic to root. Now we can go to the root directory /root/flag.txt to see the flag.

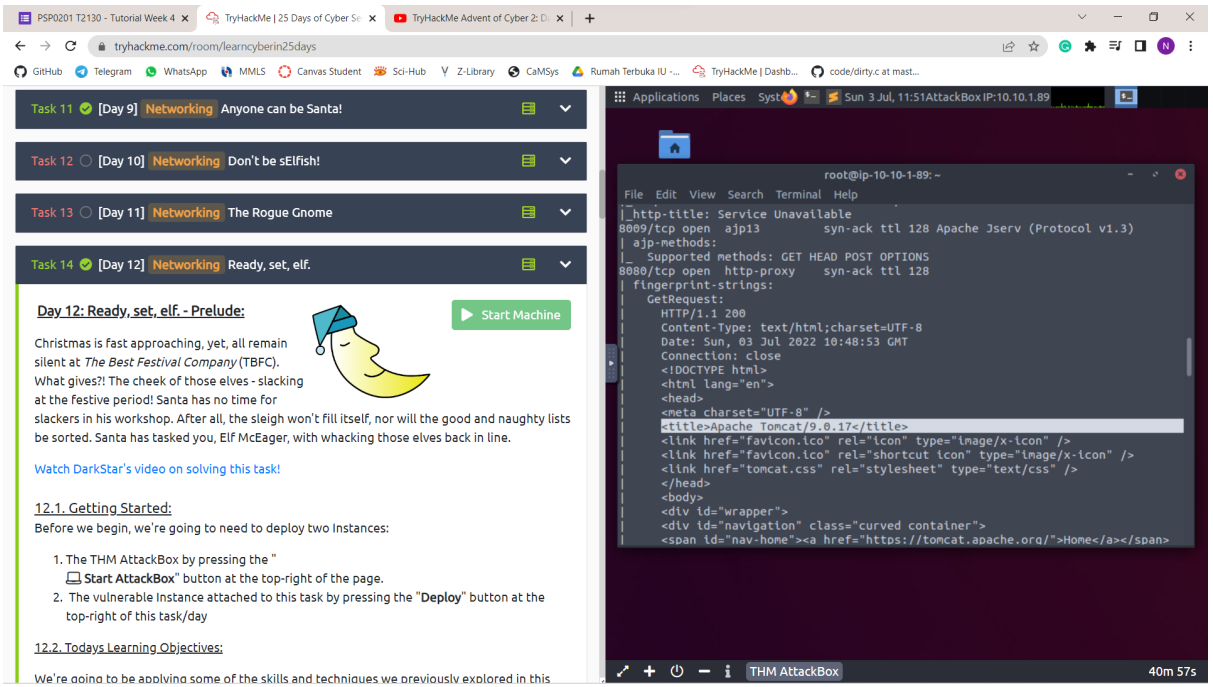
DAY 12 : Ready,set,elf - Prelude:

Tools used: terminal, firefox,google chrome

Solution/walkthrough:

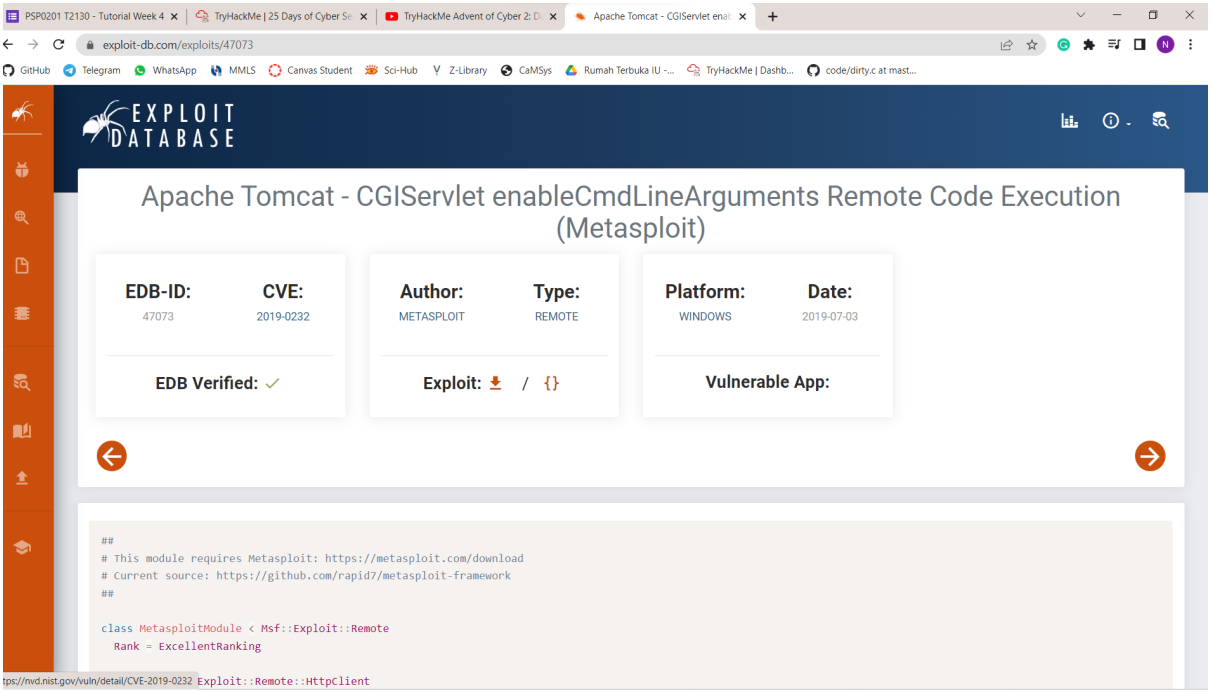
Question 1:

We used nmap -sVC command to detect service version and get common script. -vv to increase the verbosity and -iL to scan targets from a file. After commands execution, we successfully obtain version number of the web server



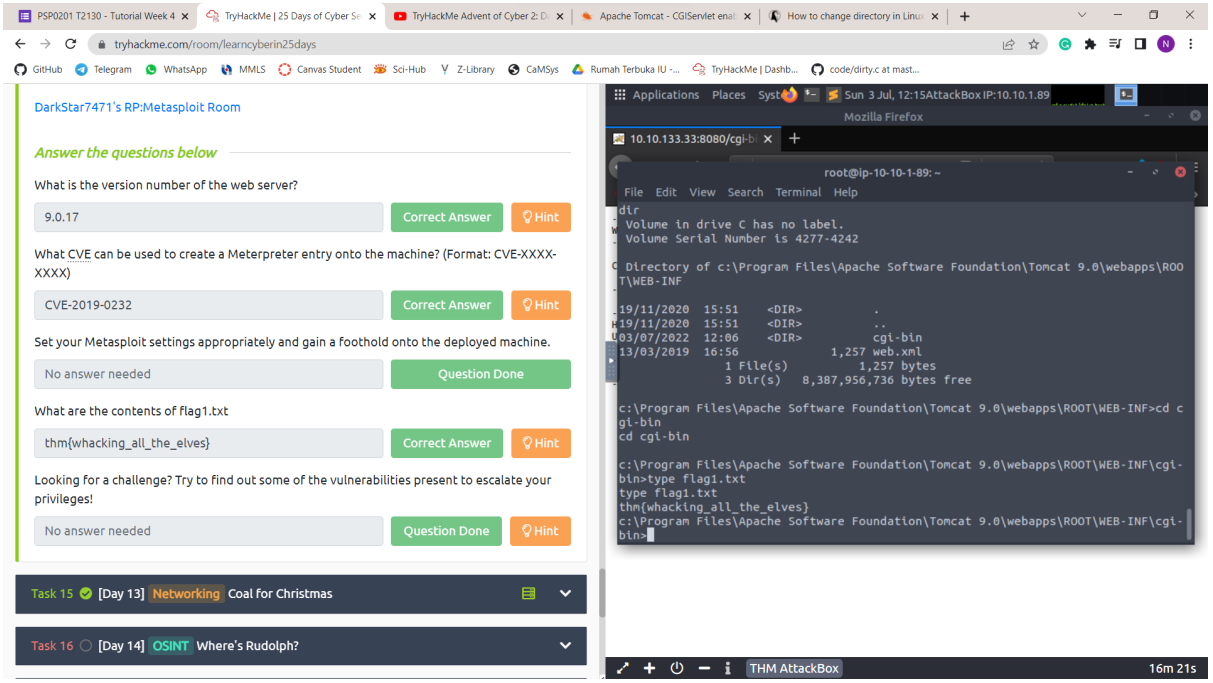
Question 2:

We use google chrome and browse to exploit-db.com to find the CVE number for Apache Tomcat metasploit.



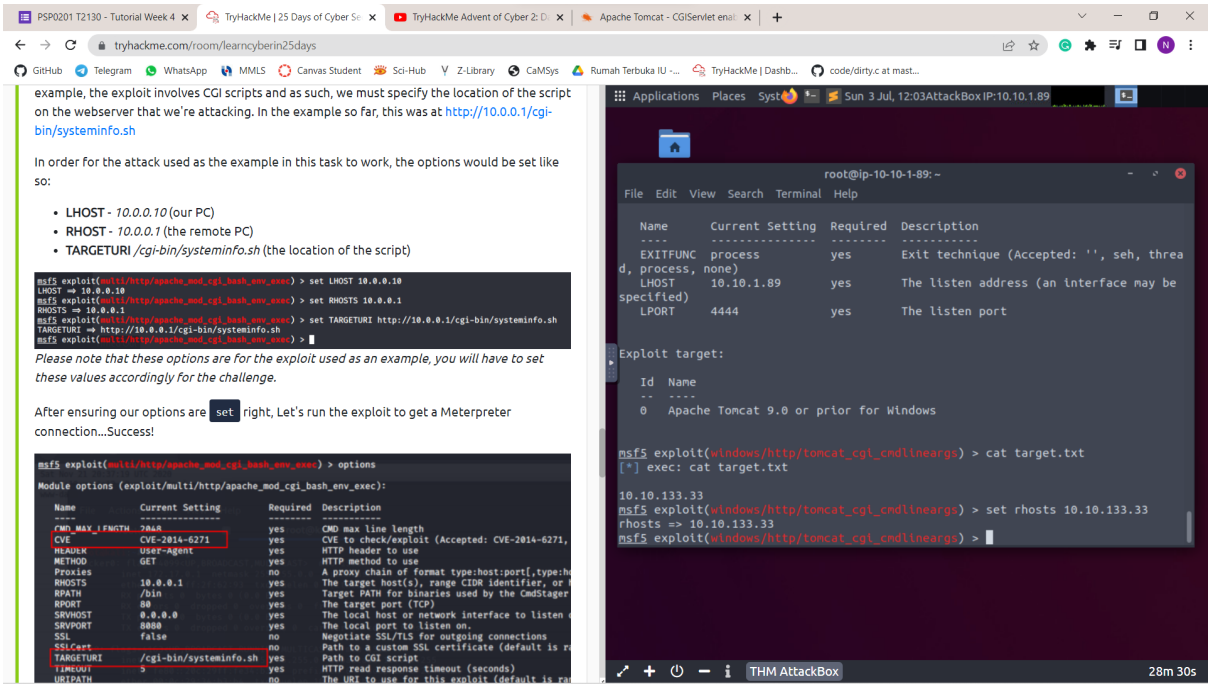
Question 3:

In order to get flag1.txt content, we need access the target machine using metasploit. After that, we need to access cgi-bin file inside Tomcat 9.0 directory



Question 4:

We need to set the rhosts and targeturi for metasploit to work



Thought Process/ Methodology:

In order to get the version number of the target webserver, we can use nmap -sVC -vv -iL commands. -sVC command was used to detect service version and get common script. -vv was used to increase the verbosity and -iL to scan targets from a file. After that, we need to find the CVE number for the webserver to know the vulnerability that the webserver version has. Next, insert the CVE number inside metasploit and setup the compulsory options. If metasploit execution works, we will then able to access to target machine using shell command and redirect to any files in the machine.

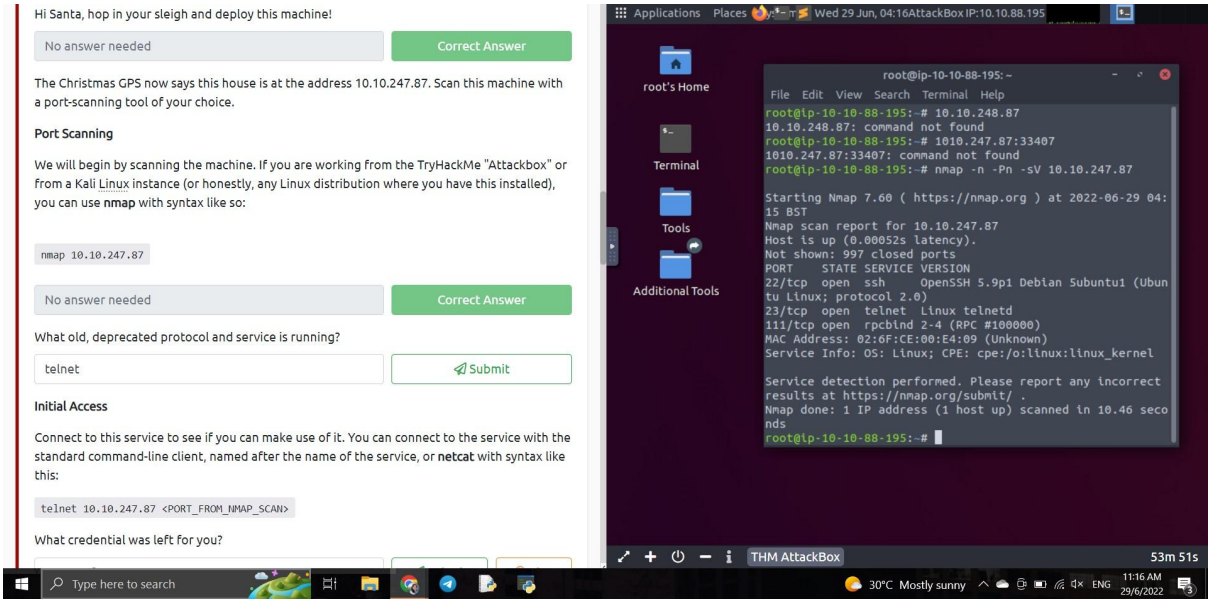
DAY 13 : Coal For Christmas

Tools used: terminal, firefox, google chrome

Solution/Walkthrough:

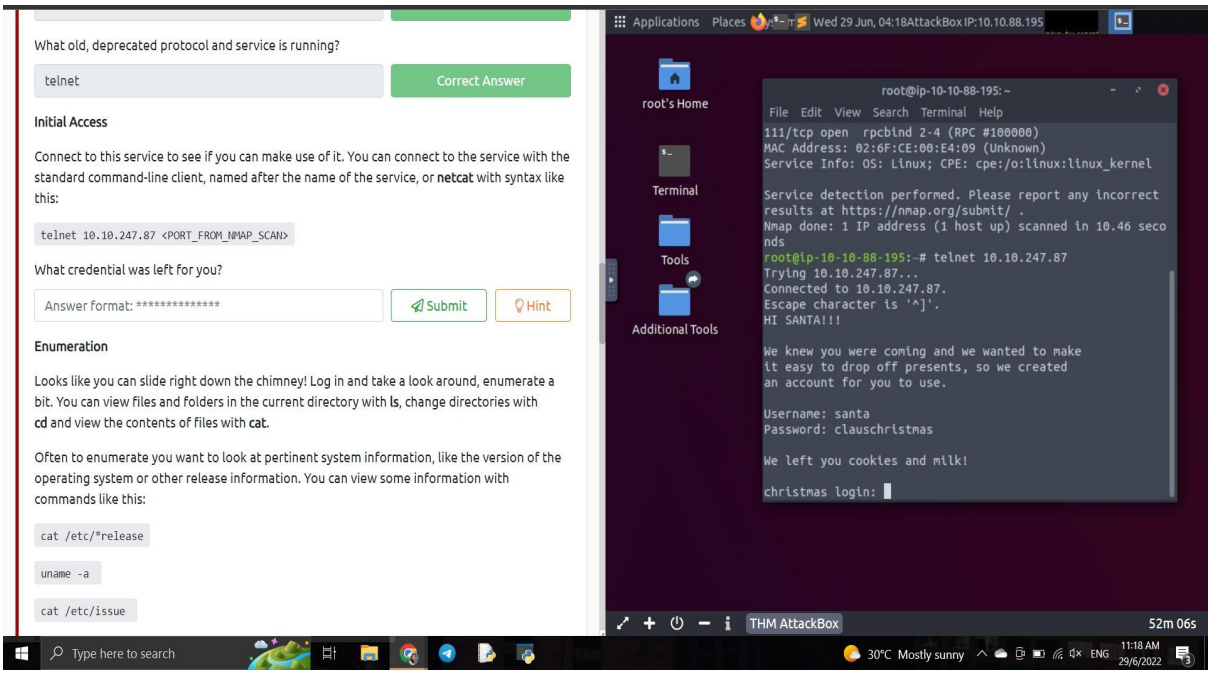
Question 1:

We searched for nmap using nmap -n -Pn -sV 10.10.247.87 in order to obtained the old, deprecated protocol and service that is currently running



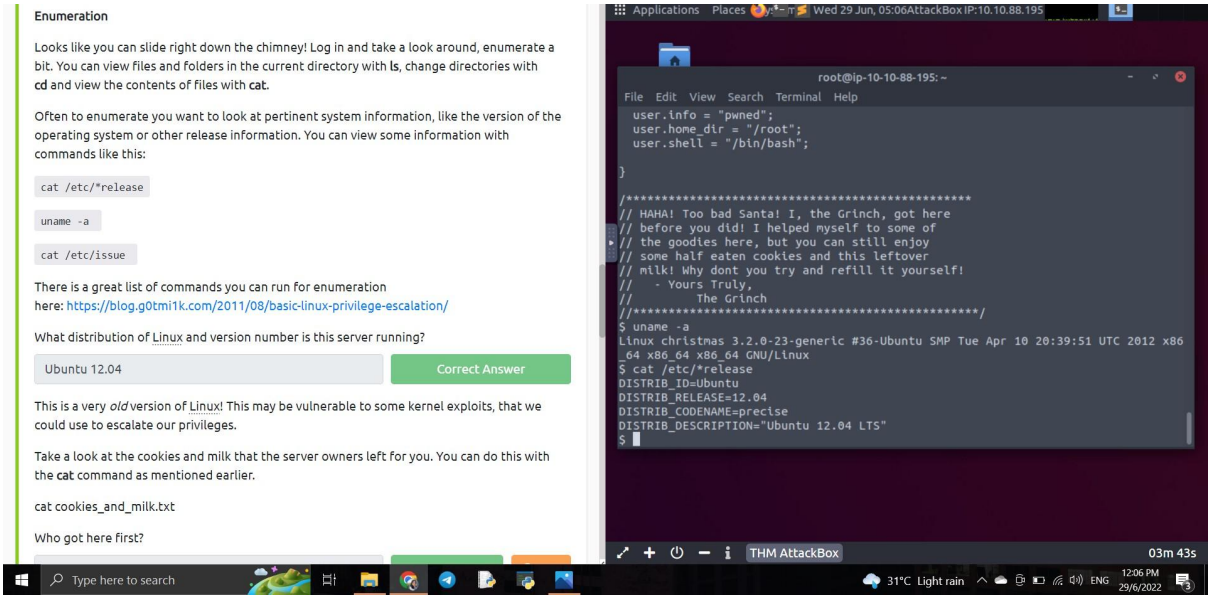
Question 2:

by using telnet 10.10.247.87 on terminal, we have found our credential left for us which was **clauschristmas**



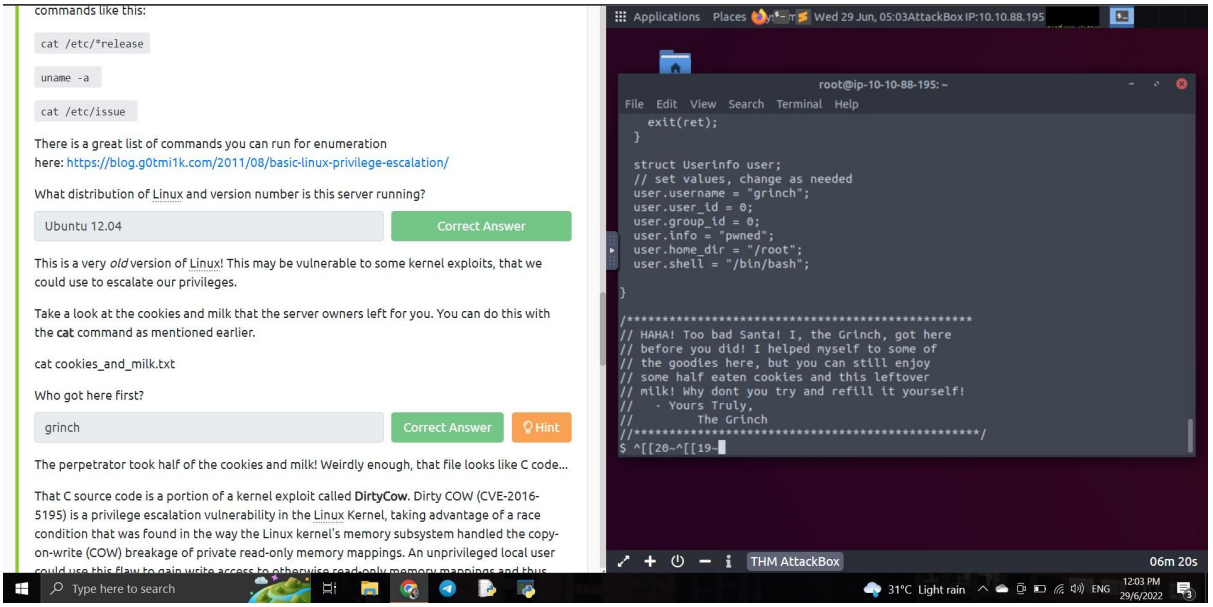
Question 3:

After that, we used the cat /etc/*release in order to obtained the distribution and version number that the server was running which was Ubuntu 12.04



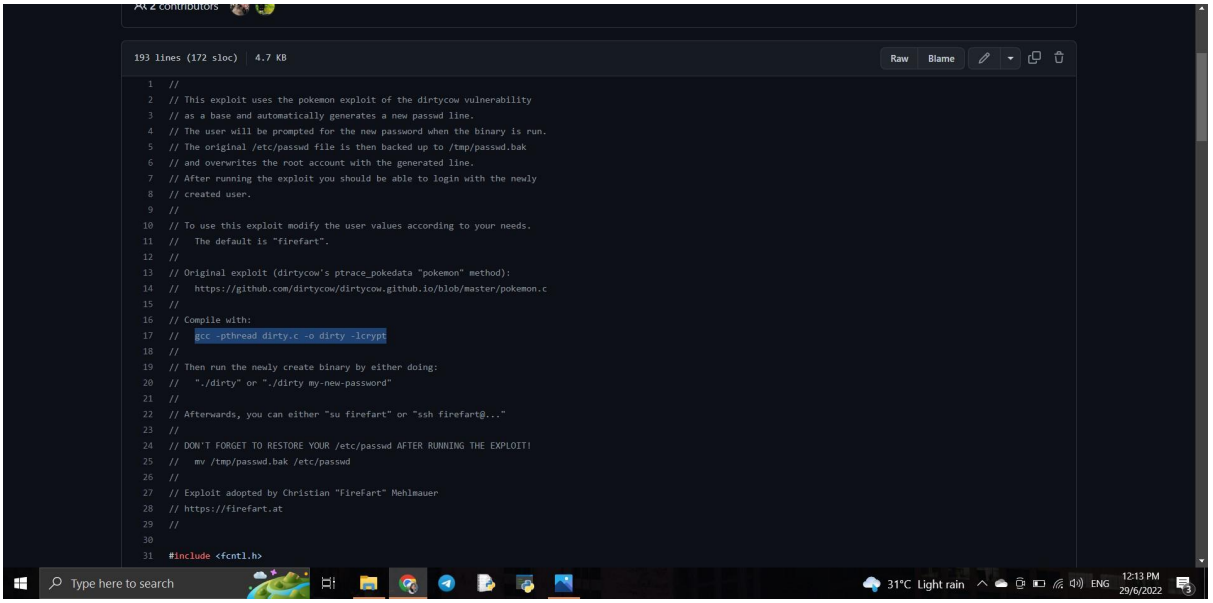
Question 4:

Then, by using the “cat cookies_and_milk.txt” command, the terminal has shown us the “person” who got there which was the **GRINCH**



Question 5:

After that, we search for the specific website which led us to github, after exploring the codes, we discovered the verbatim syntax we could use, taken from the real C source code comments.



```
193 lines (172 sloc)  4.7 KB                                   Raw  Blame

1   //
2   // This exploit uses the pokemon exploit of the dirtycow vulnerability
3   // as a base and automatically generates a new passwd line.
4   // The user will be prompted for the new password when the binary is run.
5   // The original /etc/passwd file is then backed up to /tmp/passwd.bak
6   // and overwrites the root account with the generated line.
7   // After running the exploit you should be able to login with the newly
8   // created user.
9   //
10  // To use this exploit modify the user values according to your needs.
11  //   The default is "firefart".
12  //
13  // Original exploit (dirtycow's ptrace_pokedata "pokemon" method):
14  //   https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c
15  //
16  // Compile with:
17  //   gcc -pthread dirty.c -o dirty -lcrypt
18  //
19  // Then run the newly create binary by either doing:
20  //   "./dirty" or "./dirty my-new-password"
21  //
22  // Afterwards, you can either "su firefart" or "ssh firefart@..."
23  //
24  // DON'T FORGET TO RESTORE YOUR /etc/passwd AFTER RUNNING THE EXPLOIT!
25  //   mv /tmp/passwd.bak /etc/passwd
26  //
27  // Exploit adopted by Christian "FireFart" Mehlmauer
28  // https://firefart.at
29  //
30
31  #include <fcntl.h>
```

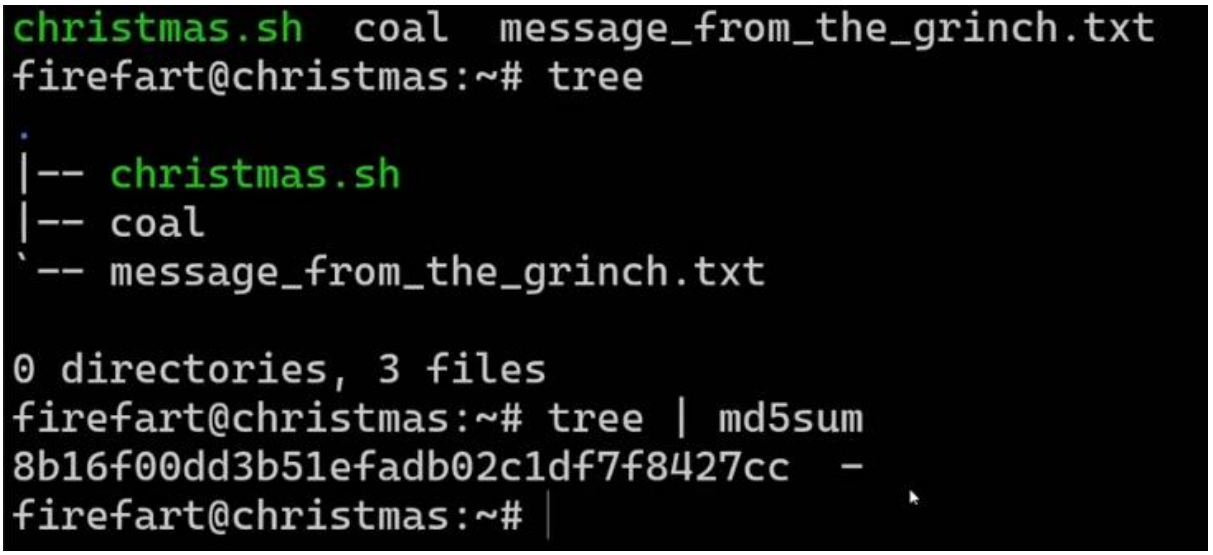

Question 6:

Then, by using terminal we have found the “new” username that was created with the default operations of the real C source code which was **firefart**.

```
┌──(1211103423㉿kali)-[~]
└─$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fiRzXJGixlD0Q:0:0:pwned:/root:/bin/bash

mmap: 7f75899d8000
madvise 0

ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'mr151003'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'mr151003'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
```

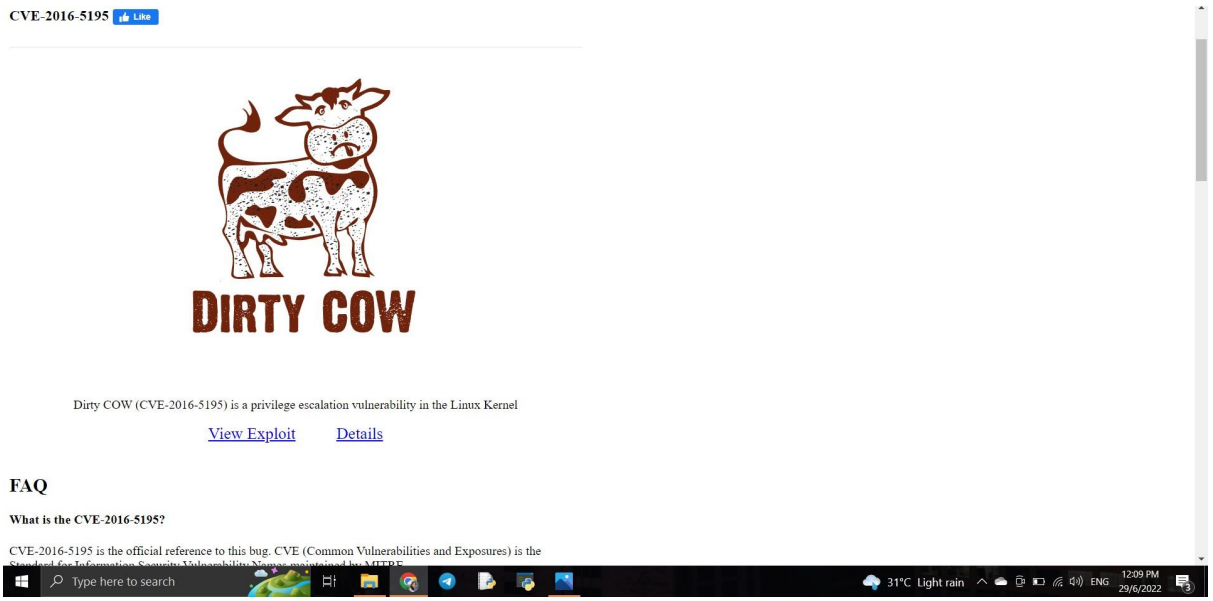
Question 7:

Then, after we got in the server in the terminal, by using “# tree” command, we successfully generated the MD5 hash output which was **8b16f00dd3b51efadb02c1df7f8427cc**



Question 8:

Then, we went to google chrome to find the CVE for DirtyCow. It was shown obviously when we loaded it on the web.



Thought process / methodology:

Firstly, we start the machine and get the IP address of the machine. Then, we open the terminal and start scanning the IP address of the target machine using nmap. After that, we can see that telnet is running on port 23.Then, we continue with typing the syntax telnet <machine_ip> <port>. After connecting, we are given some credentials which are the username and password to login. After logging in, we are now inside the telnet service. Then, we find some information about the OS distribution type and version by using the command cat/etc/*release and we got the distribution of Linux and version number for the server. After we got the information, we went for a look at the file named cookies_and_milk.txt using the cat command. After that we can see some C programming language code with the Grinch message at the top. Then, we find the original file of the exploited DirtyCow code and create a file for the code using netcat. We also get the syntax used to compile the C program file. After we run the program, it creates a username 'firefart'. We also asked to create a password for the user. After that, we switch our user account using su. We are given a txt file named message_from_grinch. Lastly we use coal command to open the content of the file and we successfully get the MD5 hash output

DAY 14 : OSINT- Where's Rudolph?

Tools used: Reddit, Twitter, Google Chrome

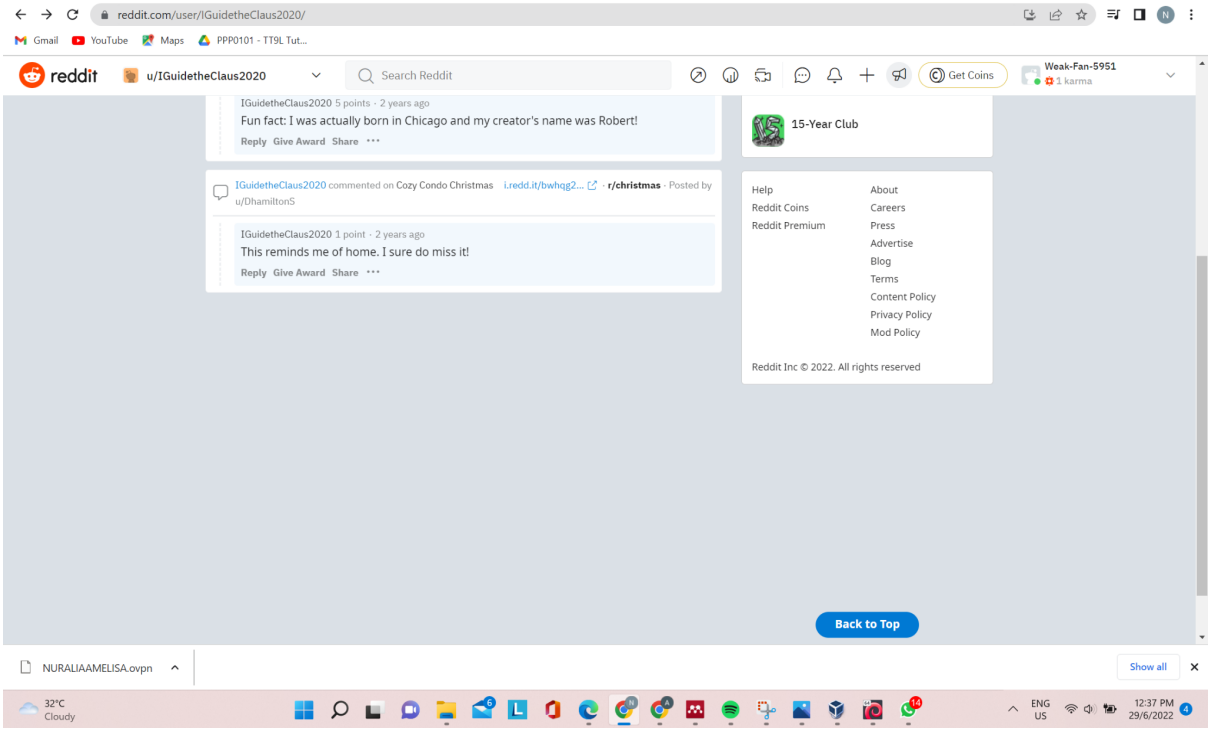
Solution/Walkthrough:

Question 1:

Type Rudolph's username on Reddit. Once we have found Rudolph's account, navigate to the comment tab and copy the URL.

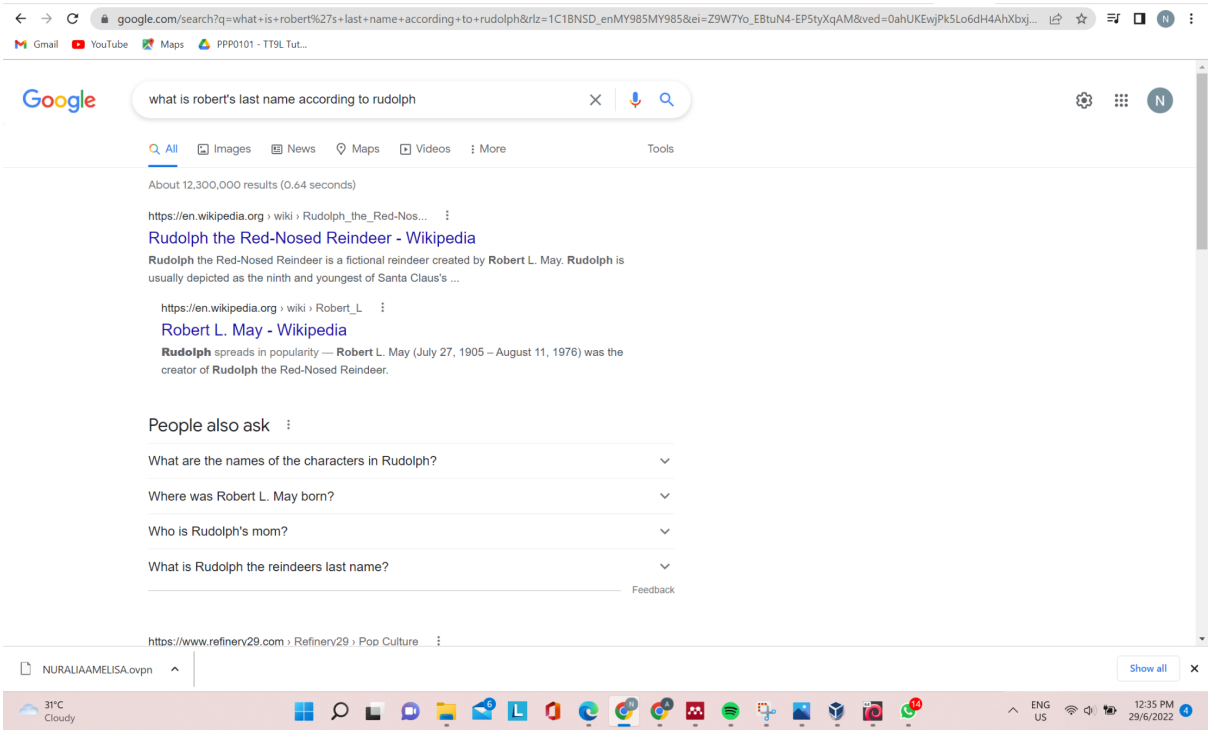
Question 2:

In one Reddit post, Rudolph commented that he was born in Chicago.



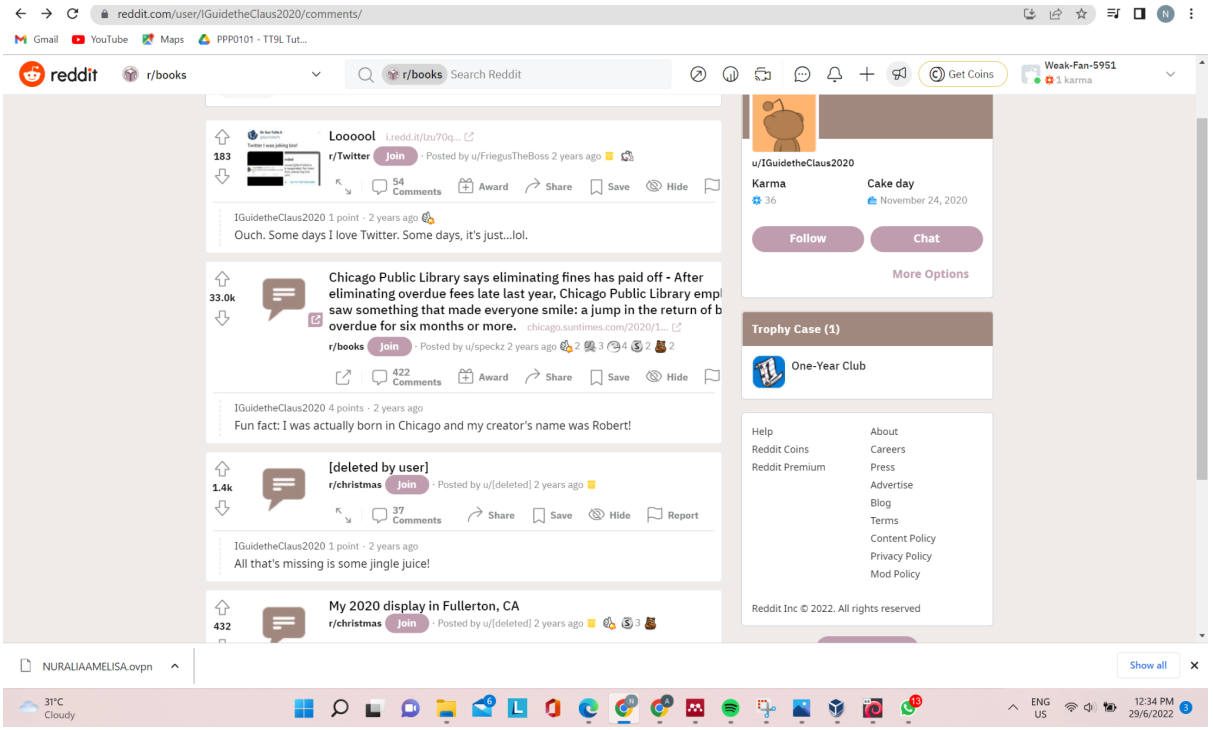
Question 3:

Rudolph mentioned that his creator's name is Robert. Use Google Chrome to find his creator. The one that we found shows that his last name is May.



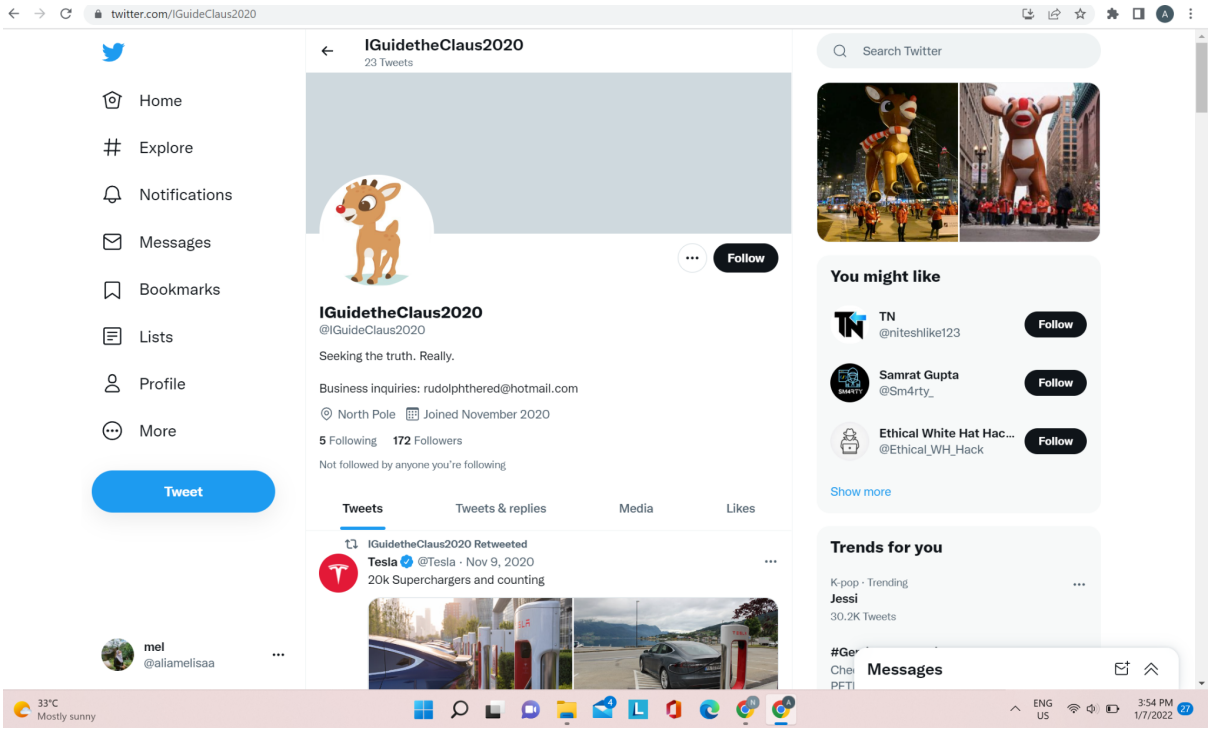
Question 4:

Rudolph commented in a Reddit post that some days he loves Twitter. So this shows that Rudolph must be on Twitter.



Question 5:

We have to type Rudolph's username on Reddit in Twitter to find his Twitter account. Once found, we can now see his Twitter username.

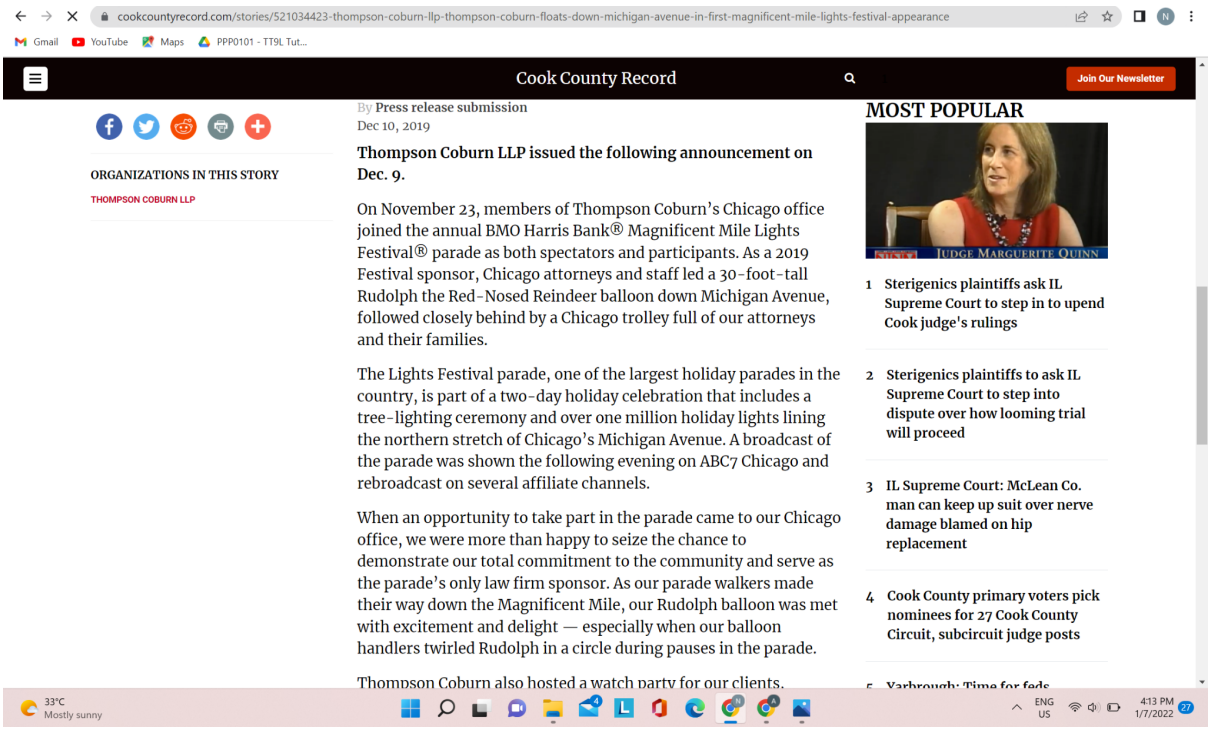


Question 6:

Rudolph tweeted that he loves Bachelorette. Since The Bachelorette is the name of a TV show, it must be Rudolph's favourite TV show as well.

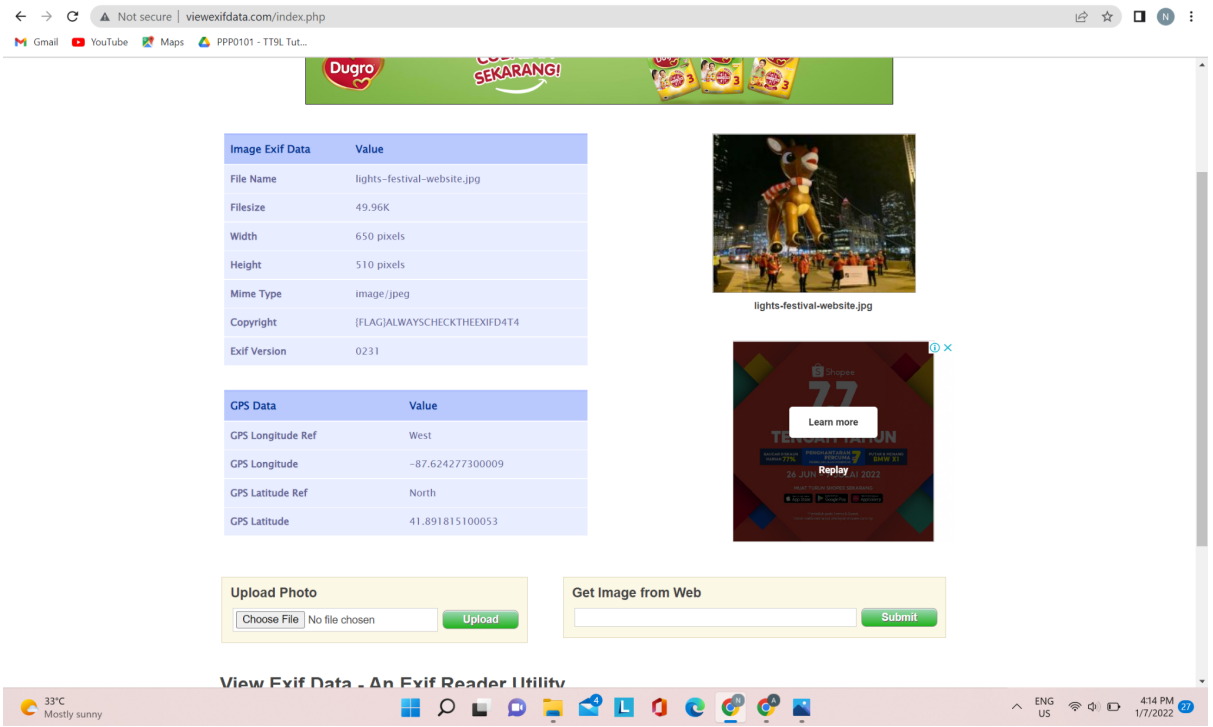
Question 7:

We need to save the image of Rudolph joining the parade and upload the image back to Google Images. The search result will show where the image is from. In an article, we found that the parade was held in the city of Chicago.



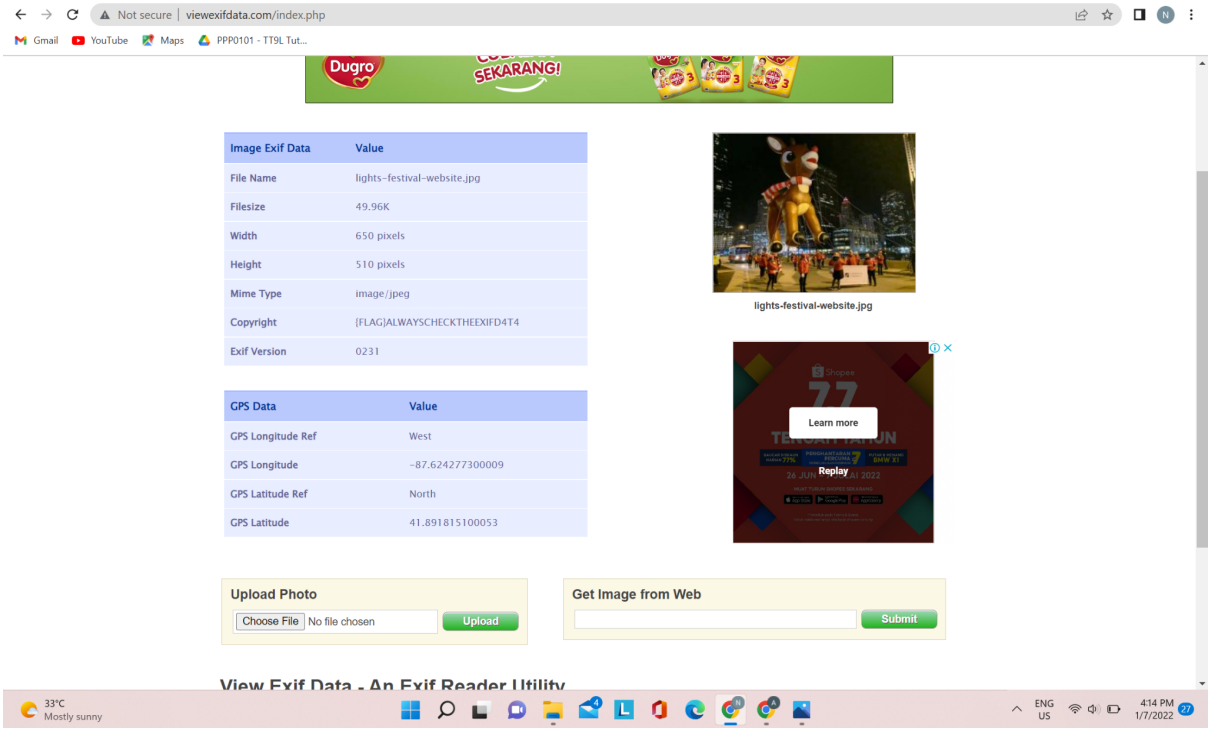
Question 8:

Using viewexifdata.com, upload the clearer version of the parade image to the website. We can view the GPS Coordinates on the section “GPS Data”.



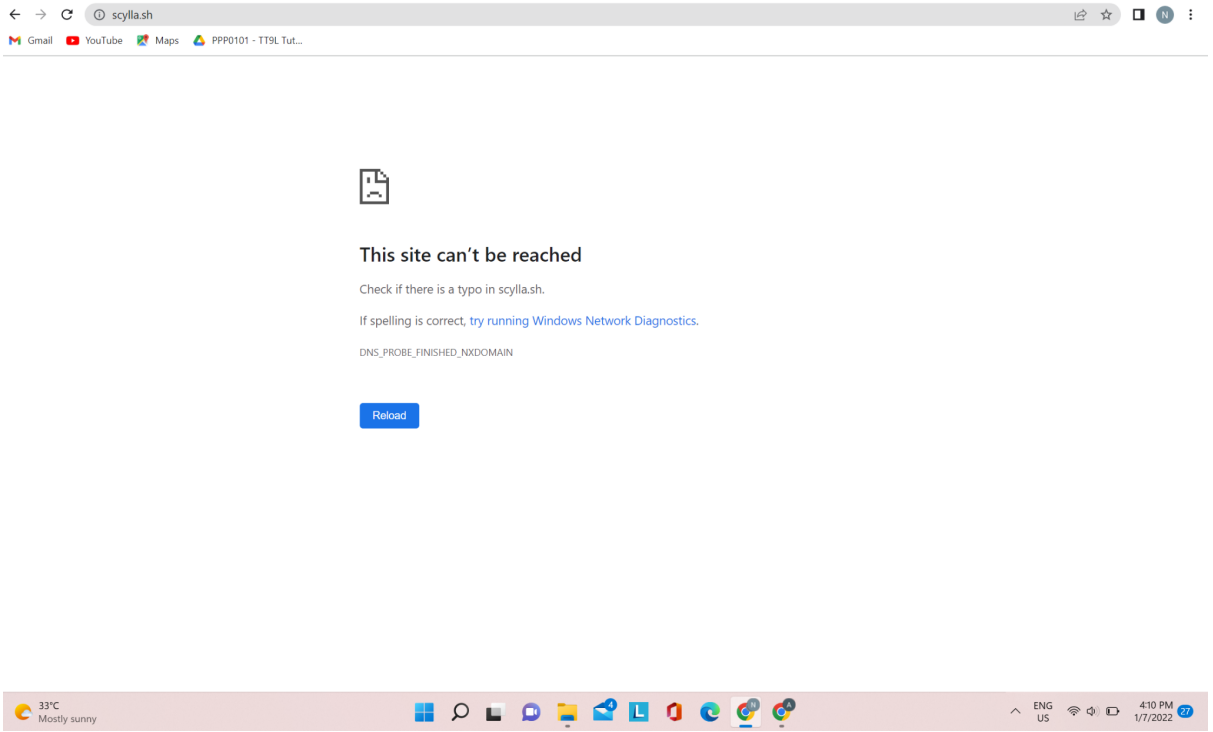
Question 9:

The flag can be obtained on the section “Image Exif Data”.



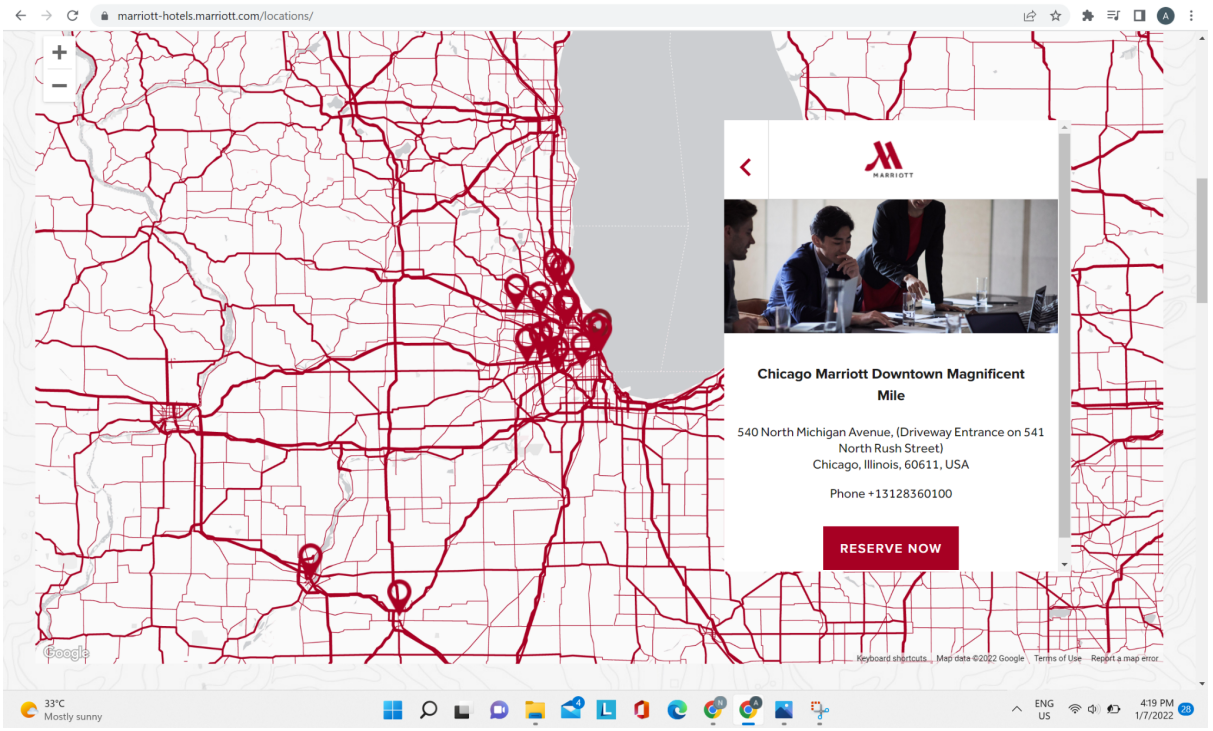
Question 10:

Scylla seems to be down so we cannot use it to look for the password. Dr Chin Ji Jian gives this one free answer.



Question 11:

To look for the street number of the hotel, we can go to the Marriott Hotel official website first. Since it's stated that Rudolph stays at a hotel on Magnificent Mile, we can look for where the road is located. We found that it is located in Chicago, Illinois. The street number can be found on the hotel information.



<u>Thought process / methodology:</u>

OSINT is the collection and analysis of data gathered from open sources to produce actionable intelligence. In this case, we firstly only have the username of Rudolph on Reddit. With only this information, we can track where Rudolph is. After getting a little information on Reddit, we found out that the inputs on Reddit are not enough to search the location of Rudolph. Since we now know that Rudolph also owns a Twitter account, we can further our investigation on Twitter. Finally, we know where Rudolph is by just having his Reddit username in the first place.

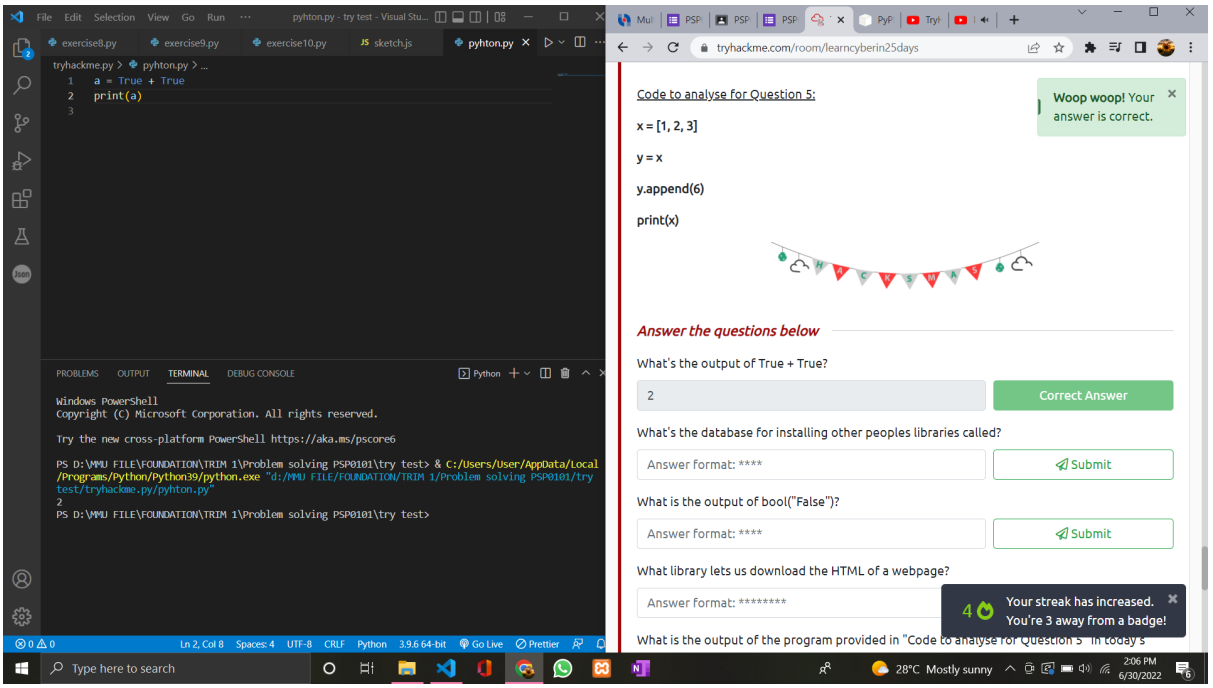
Day 15 - [Scripting] There's a Python in my stocking!

Tools used: Python, VS Code

Solutions/walkthrough:

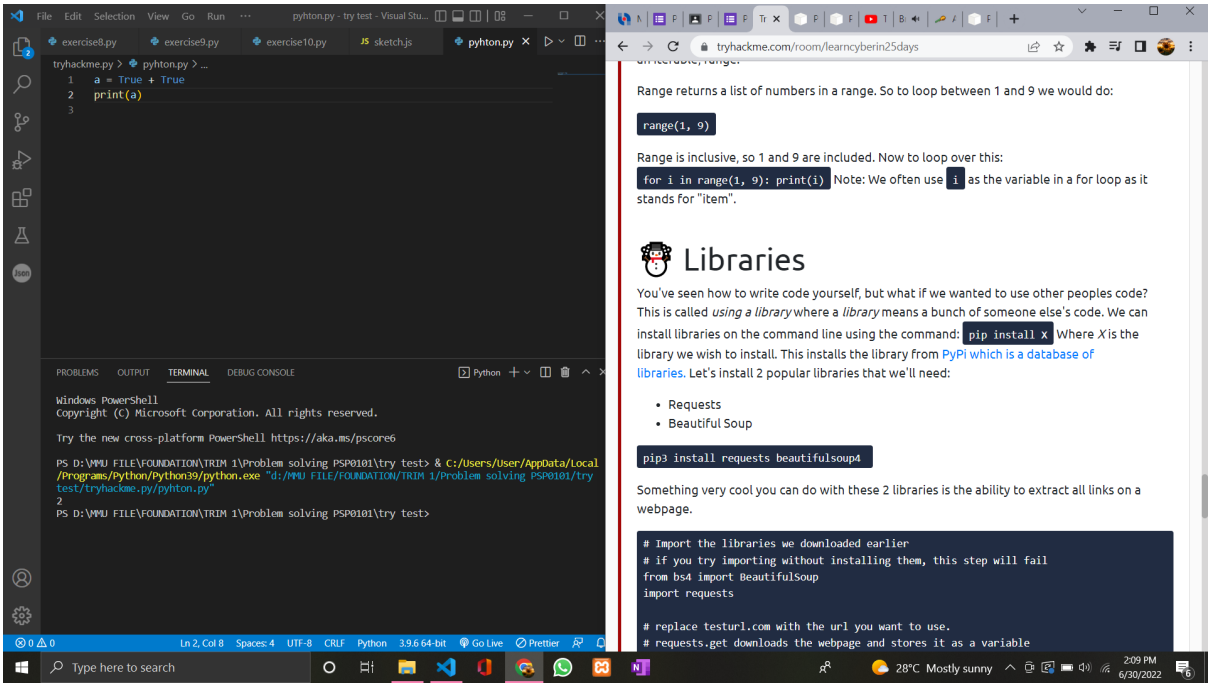
Question 1:

True is boolean and when we add with the same boolean, the output is the total of the boolean.



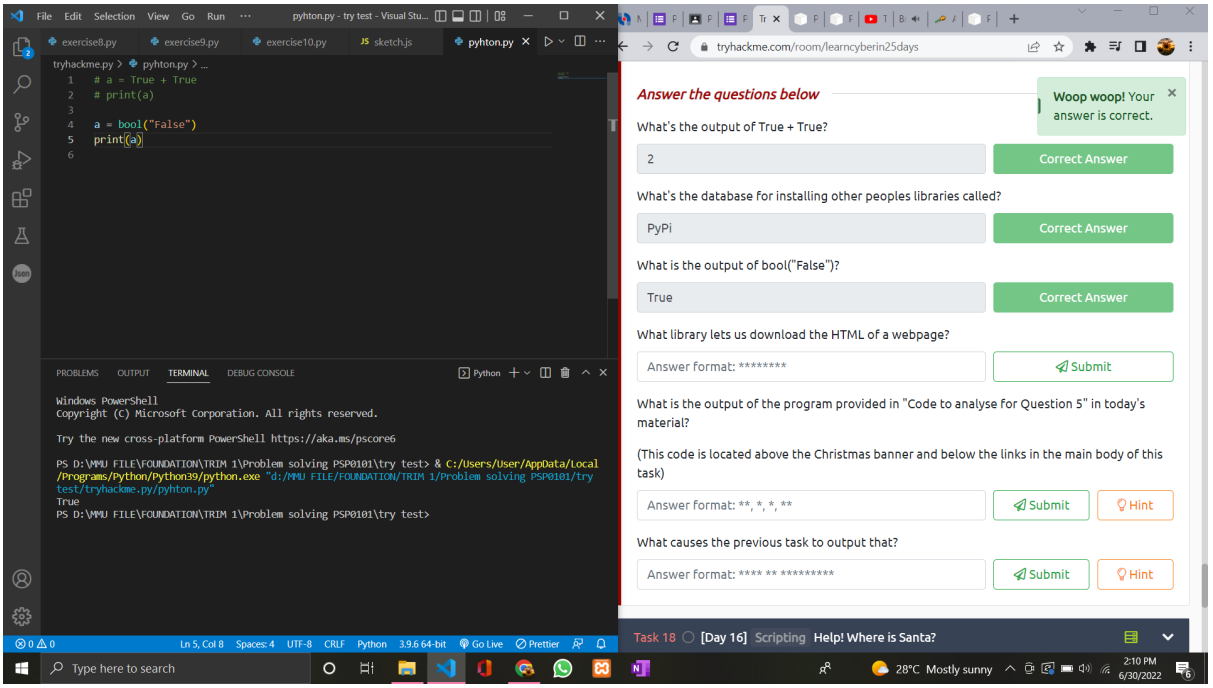
Question 2:

PyPi is the database of libraries.



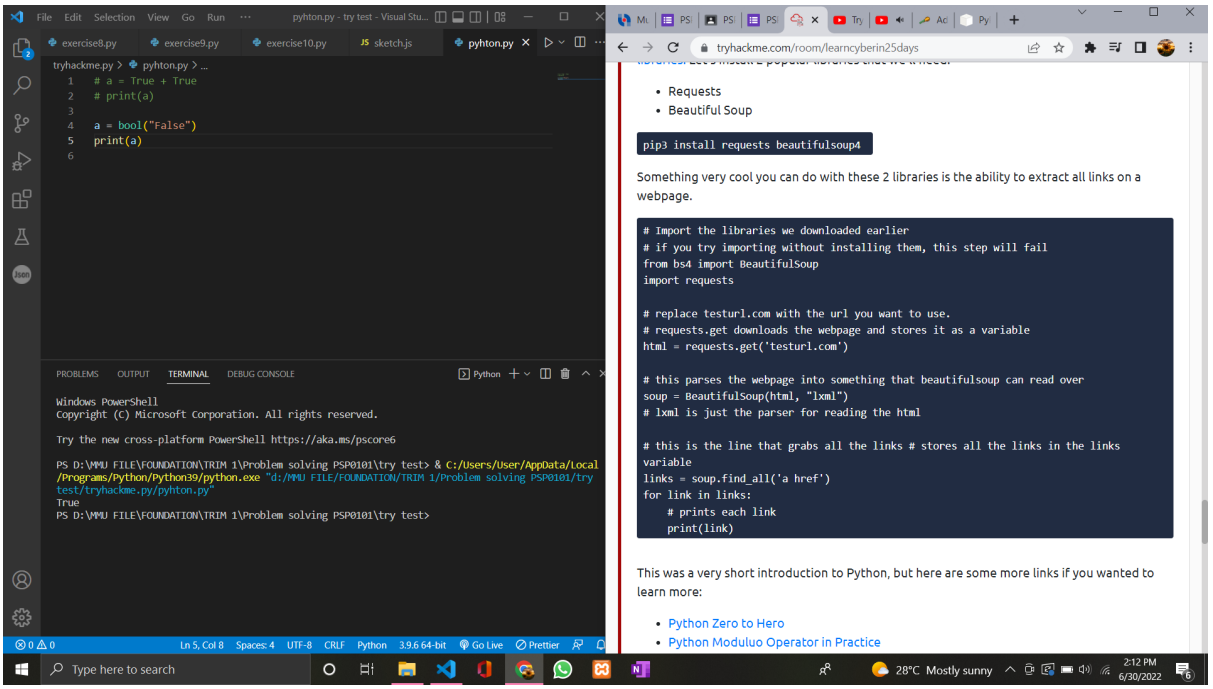
Question 3:

True because inside the bool() there is “False”, so the output is true or a string.



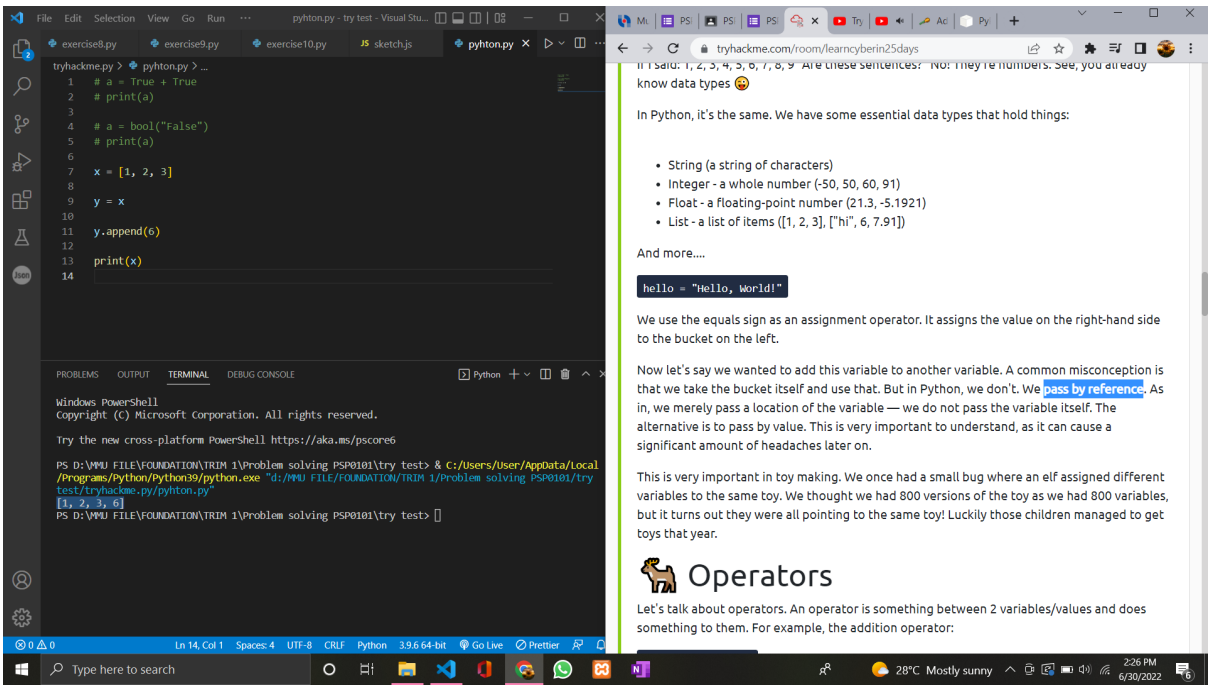
Question 4:

requests.get downloads the webpage and stores it as a variable



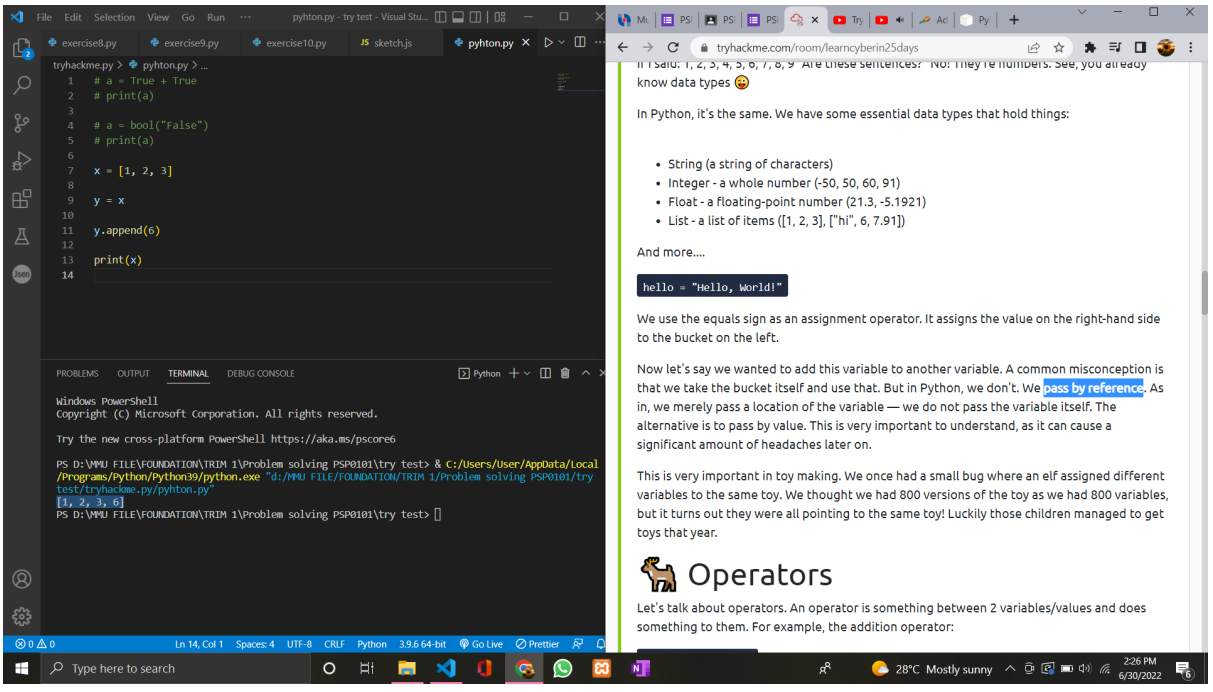
Question 5:

[1, 2, 3, 6], append is the addition in the list, so 6 will get into the list.



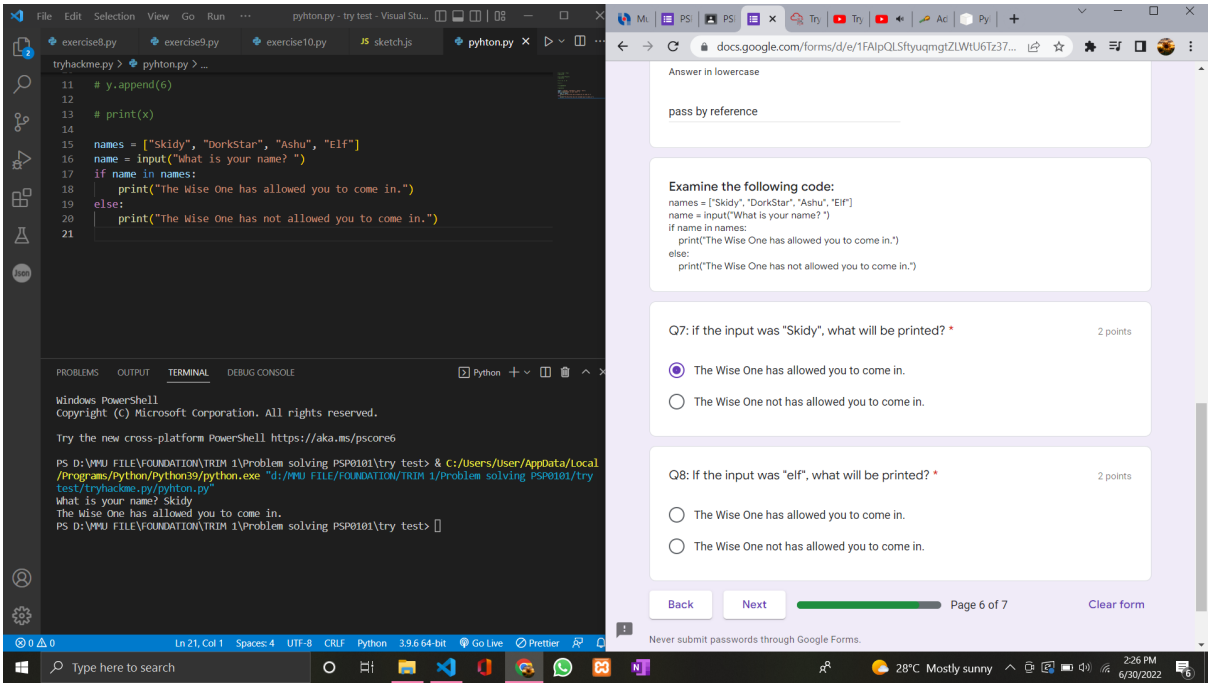
Question 6:

Pass by reference



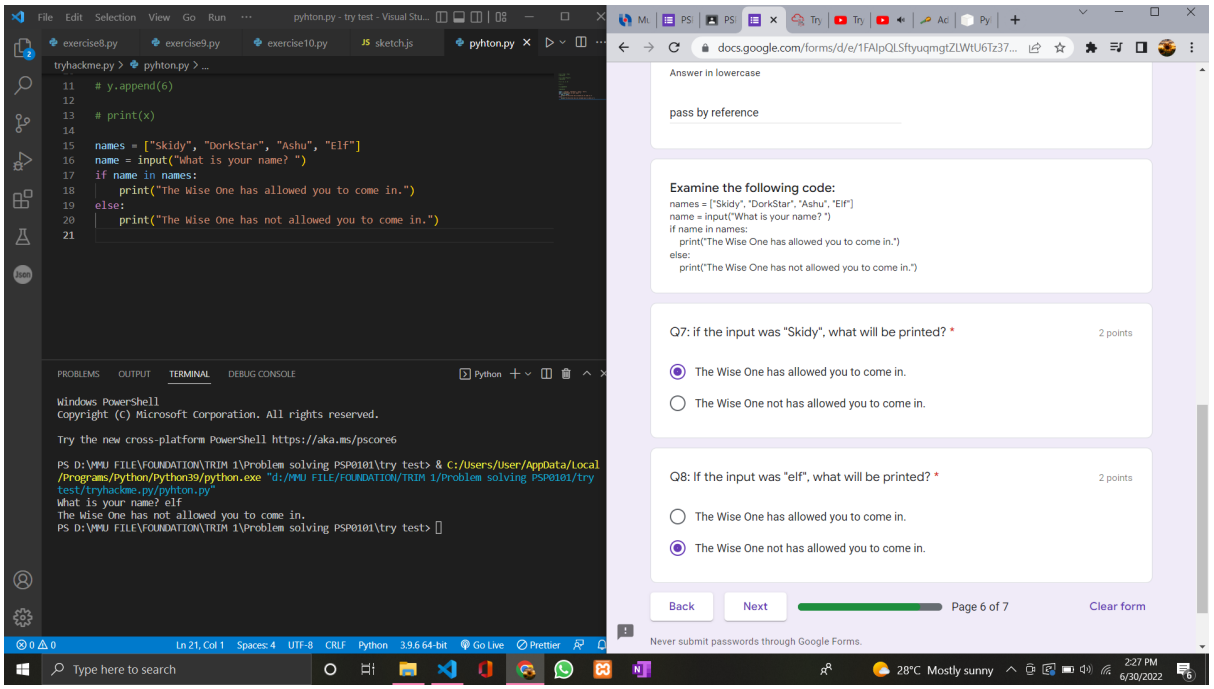
Question 7:

Skidy is one of the name in the names list.



Question 8:

Elf is not the one of the name in the names list.



Thought Process/Methodology:

We have to know what is the difference between boolean, string, list, floating and integer as there might be some data types that cannot use the operators, and have special command to insert or exclude the data or something. There are various types of the libraries that can be downloaded through the request command. Lastly, in the if statement, the output will be executed if the input is inside the variables.