PSP0201 WEEKLY REPORT

Group name: Apocalypse

Members

ID	NAME	ROLE
1211103698	UMMI SYAHIRAH BINTI MUHAMMAD ROZAIDEE	LEADER
1211103293	FARAH KAMILA BINTI YAHYA	MEMBER
1211102031	NOR ALIAH SYUHAIDAH BINTI SHARUDDIN	MEMBER
1211101673	NURUL MANJA MURNIRA NAJWA BINTI MALIKI	MEMBER

DAY 11 : [NETWORKING] The Rogue Gnome

Tools uses : Kali Linux,

Question 1 & 2

11.4.2. Vertical Privilege Escalation:

A bit more traditional, 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.

Remember the attack you performed on "*Day 1 - A Christmas Crisis*"? You modified your cookie to access Santa's control panel. This is a fantastic example of a vertical privilege escalation because you were able to use your user account to access and manage the control panel. This control panel is only accessible by Santa (an administrator), so you are moving your permissions upwards in this sense.

Question 3

11.4.1. Horizontal Privilege Escalation:

A horizontal privilege escalation attack involves using the intended permissions of a user to abuse a vulnerability to access another user's resources who has similar permissions to you. For example, using an account with access to accounting documents to access a HR account to retrieve HR documents. As the difference in the permissions of both the Accounting and HR accounts is the data they can access, you aren't moving your privileges upwards.

Question 4

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

11.6. You Thought Enumeration Stopped at Nmap?

Wrong! We were just getting started. After gaining initial access, it's essential to begin to build a picture of the internals of the machine. We can look for a plethora of information such as other services that are running, sensitive data including passwords, executable scripts of binaries to abuse and more!

For example, we can use the find command to search for common folders or files that we may suspect to be on the machine:

- backups
- password
- admin
- config

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*?

Question 6

Replace ‘filename’ with the provided name of the file, ‘find.sh’.

Our directory has three directories "**exampledir**[3]" and three files "**examplefile**[3]". I've listed the four columns of interest here:

Column Letter	Description	Example
[A]	filetype (`d` is a directory `-` is a file) and the user and group permissions "**r**" for reading, "**w**" for write and "**x**" for executing.	A file with `-rw-rw-r--` is read/write to the user and group only. However, every other user has read access only
[B]	the user who owns the file	cmnatic (system user)
[C]	the group (of users) who owns the file	sudoers group

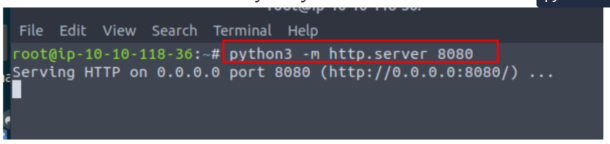
At the moment, the "examplefiles" are not executable as there is no "**x**" present for either the user or group. When setting the executable permission (`chmod +x filename`), this value changes (note the "**x**" in the snippet below -rw**x**rw**x**r):

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

Question 7

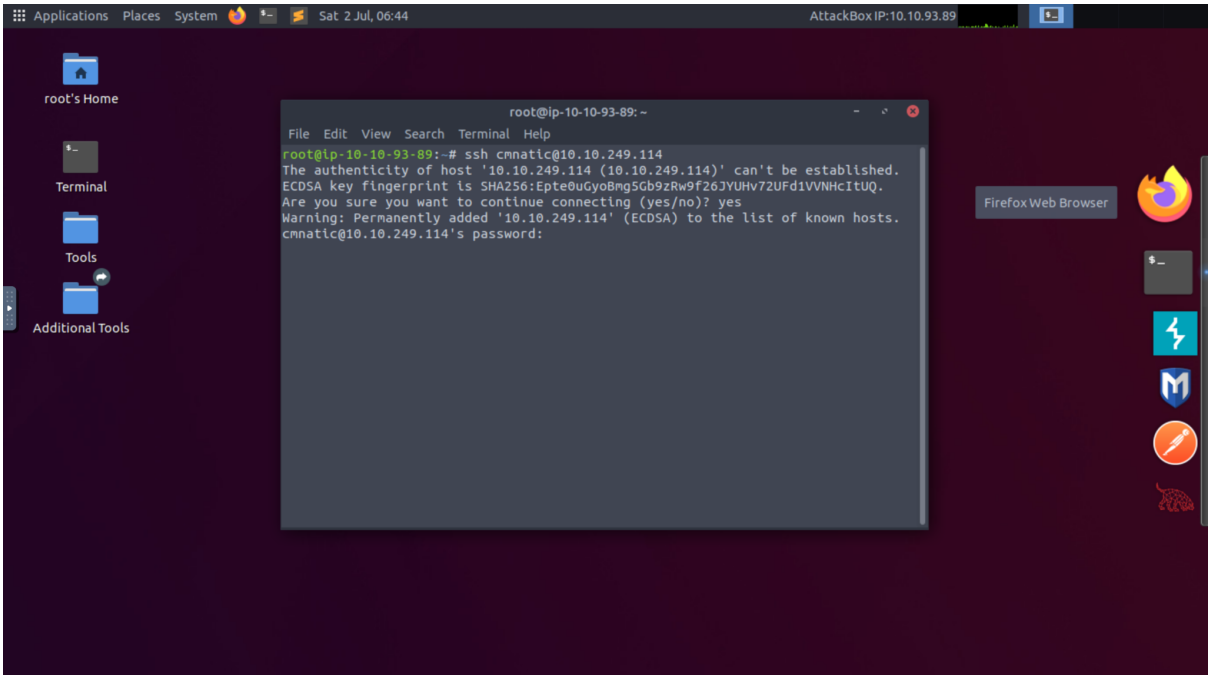
Change server port number to 9999.

11.10.2. Let's use Python3 to turn our machine into a web server to serve the *LinEnum.sh* script to be downloaded onto the target machine. Make sure you run this command in the same directory that you downloaded *LinEnum.sh* to: `python3 -m http.server 8080`

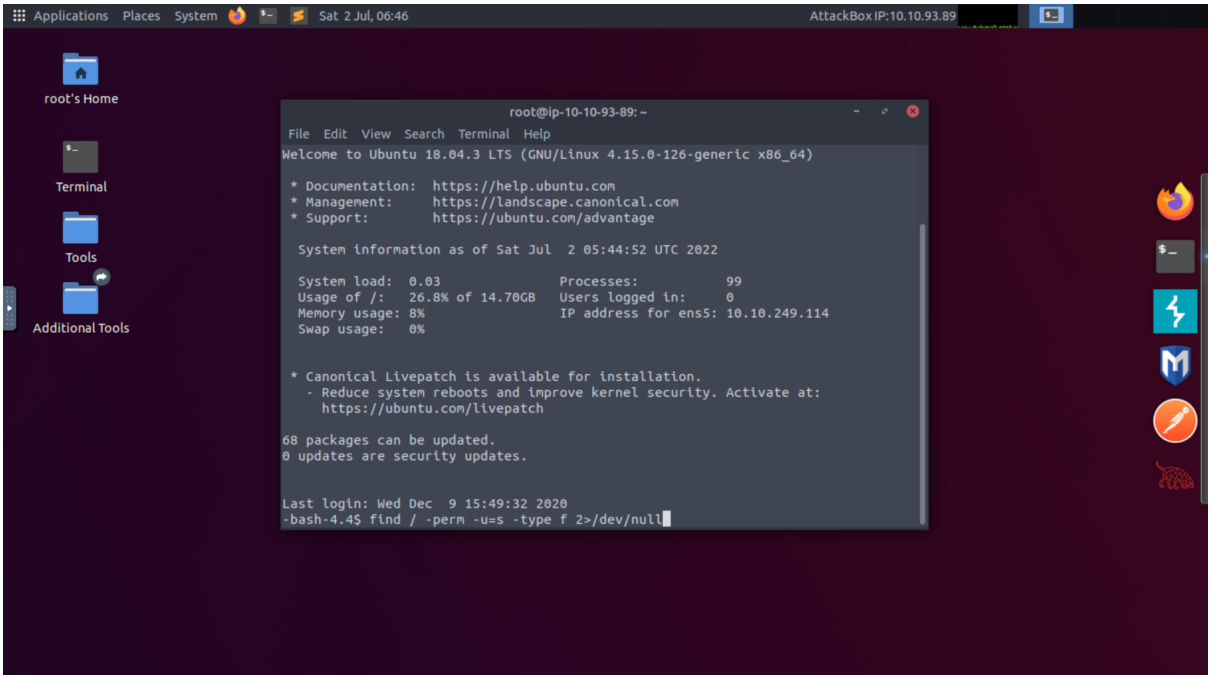


Question 8

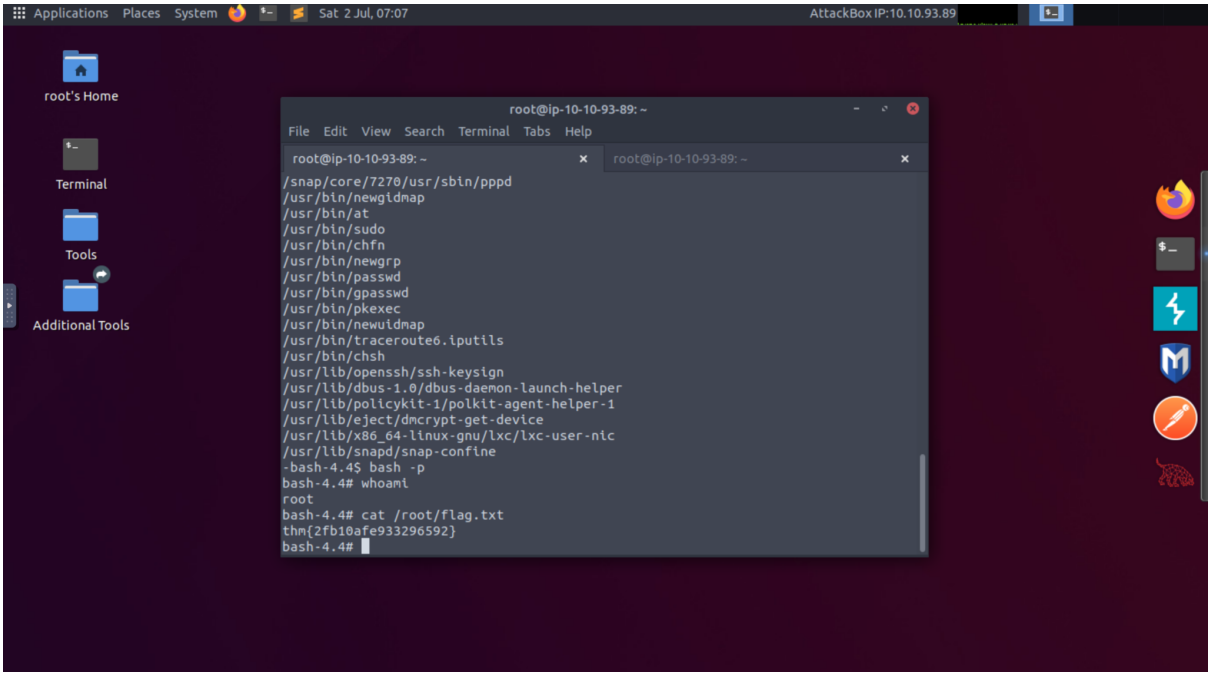
Open terminal. Use SSH login as given by TryHackMe and input the password provided.



Run '/ -perm -u=s type f 2>/dev/null' command and find which executables have the SUID permission set.



Once the output is returned, run the 'bash -p' and 'whoami' command to access root privileges. Then, we can find the contents of /root/flag.txt.



Methodology/Thought Process:

Answers for question 1-7 can be found by analysing TryHackMe's 25 Days of Cyber Security Day 11 material. For question 8, open terminal and use SSH to log in to the vulnerable machine by executing **ssh cmnatic@[IP Address]**, and when prompted for the password, put in **aoc2020** as provided by TryHackMe. When the **-bash-4.4$**

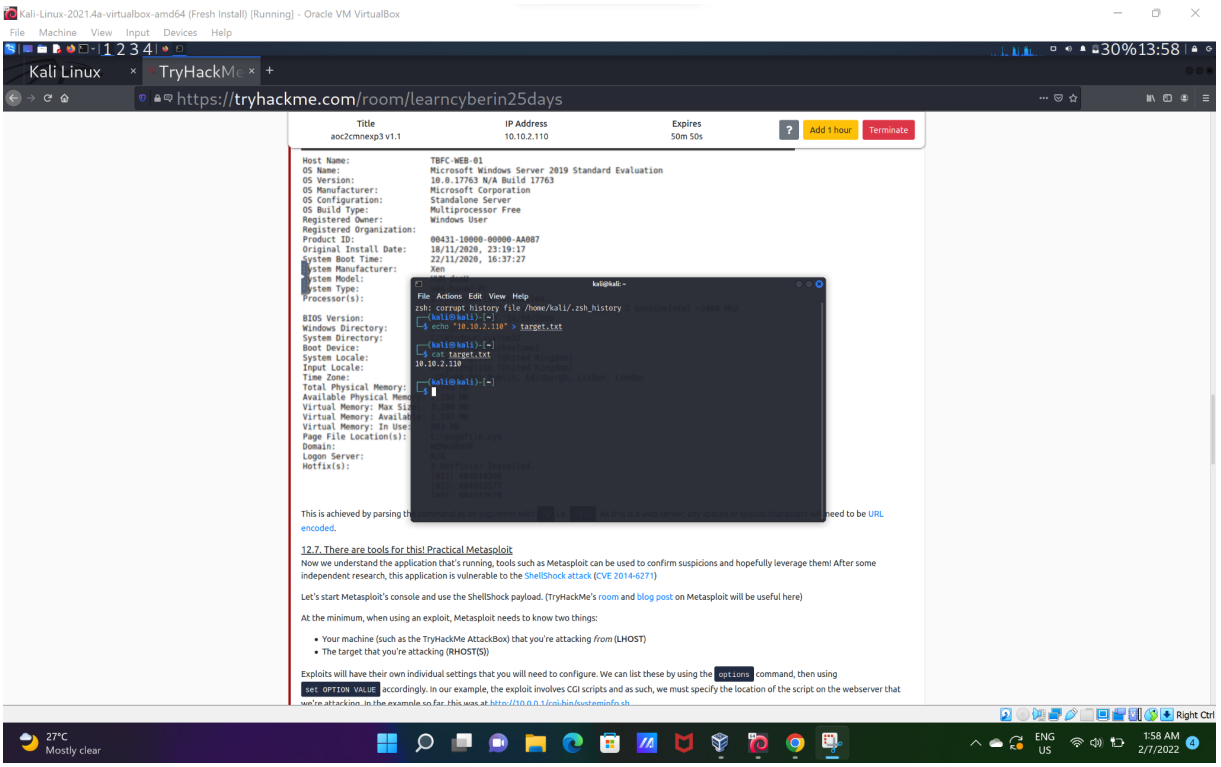
output is returned, indicating we've successfully logged in, run the command **find / -user root -type f -perm -u=s 2> /dev/null** to search the machine for executables with the SUID permission set. Then, run the **bash -p** command to give us a root shell, and verify whether we have root privileges with the command **whoami**. After we've verified our root privileges, run the command **cat /root/flag.txt** to find the contents of the file located at **/root/flag.txt**.

DAY 12 : [NETWORKING] Ready, set, elf.

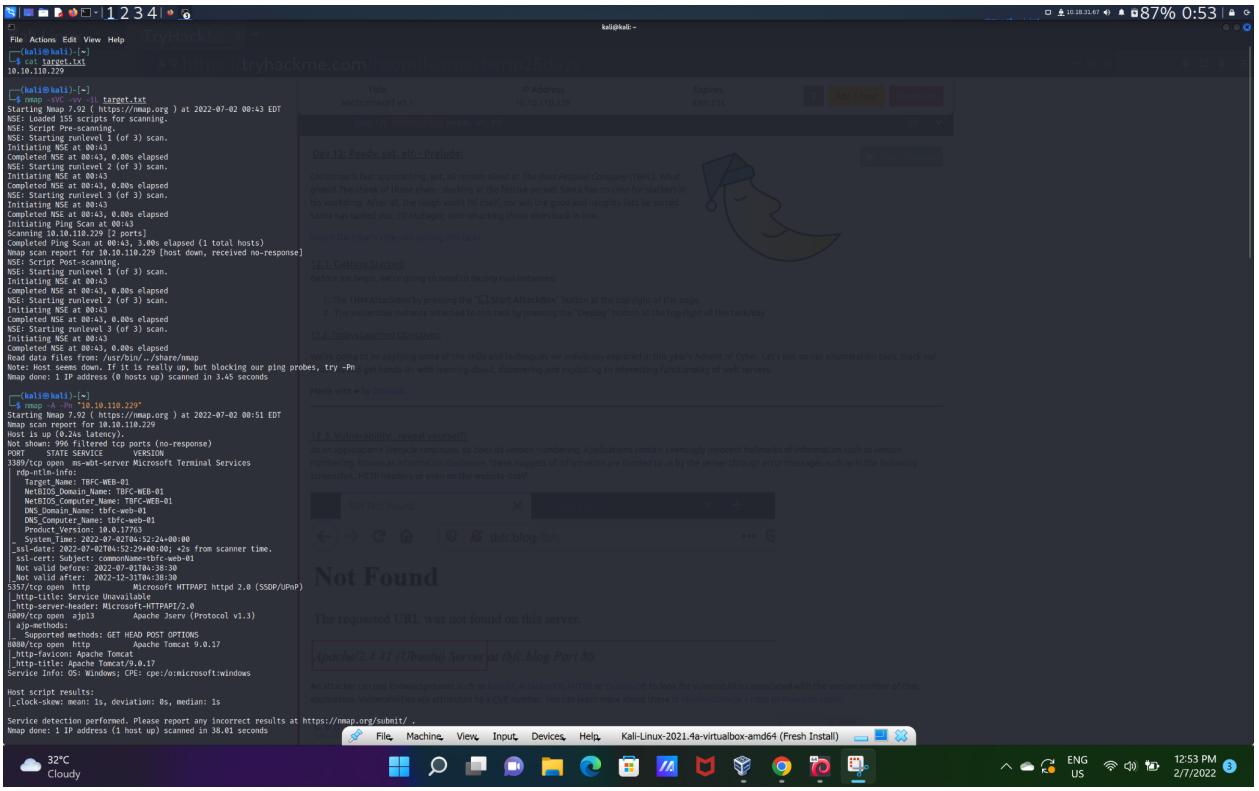
Tools uses : Kali Linux

Question 1

1. Type **echo " IP Address Day 12" > target.txt**
2. Type **cat target.txt**

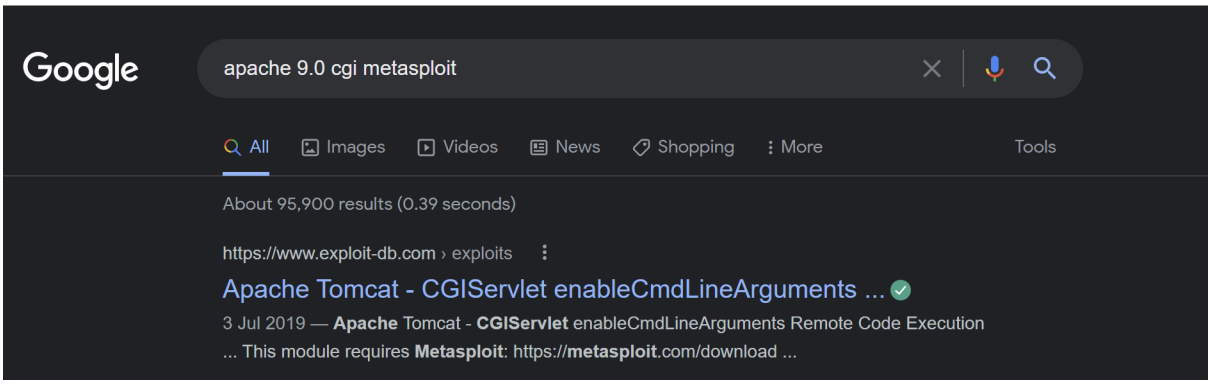


3. Type **nmap -sVC -vv -iL target.txt**
4. Type **nmap -A -Pn “IP Address Day 12”**
5. The **answer : 9.0.17**

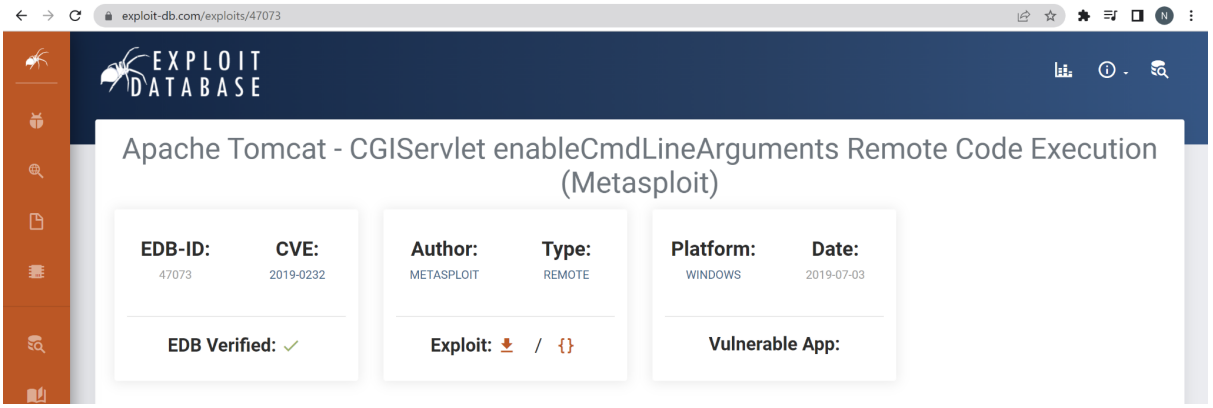


Question 2

1. Search **apache 9.0 cgi Metasploit**
2. Click the first link



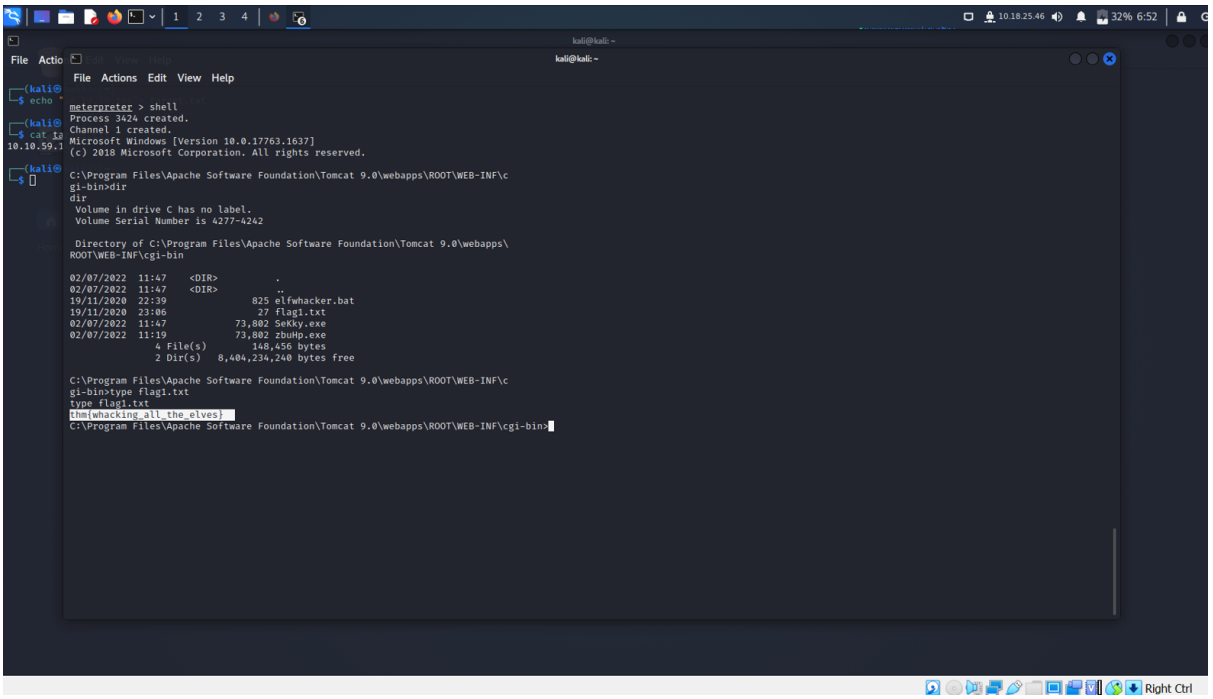
The **answer : CVE-2019-0232**



Question 3

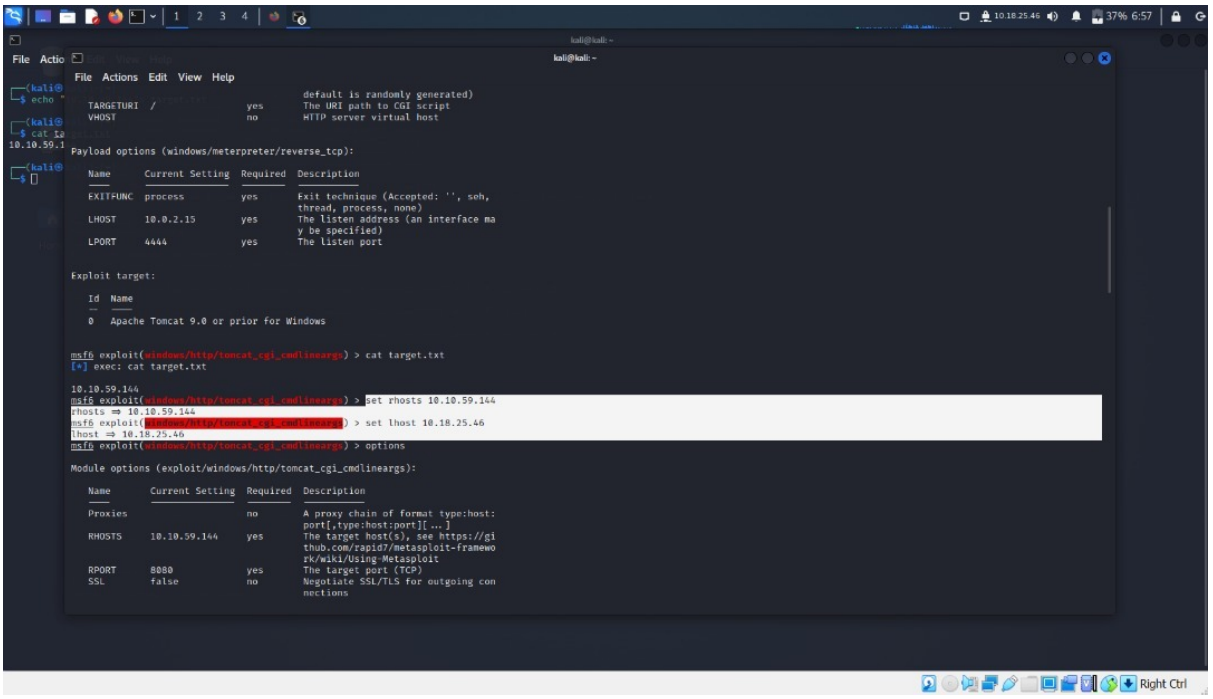
1. Open **msfconsole-q**
2. Search **2019-0232**

The answer : **thm{whacking_all_the_elves}**



Question 4

Rhosts & **Lhosts** are the Metasploit setting that we need to set.



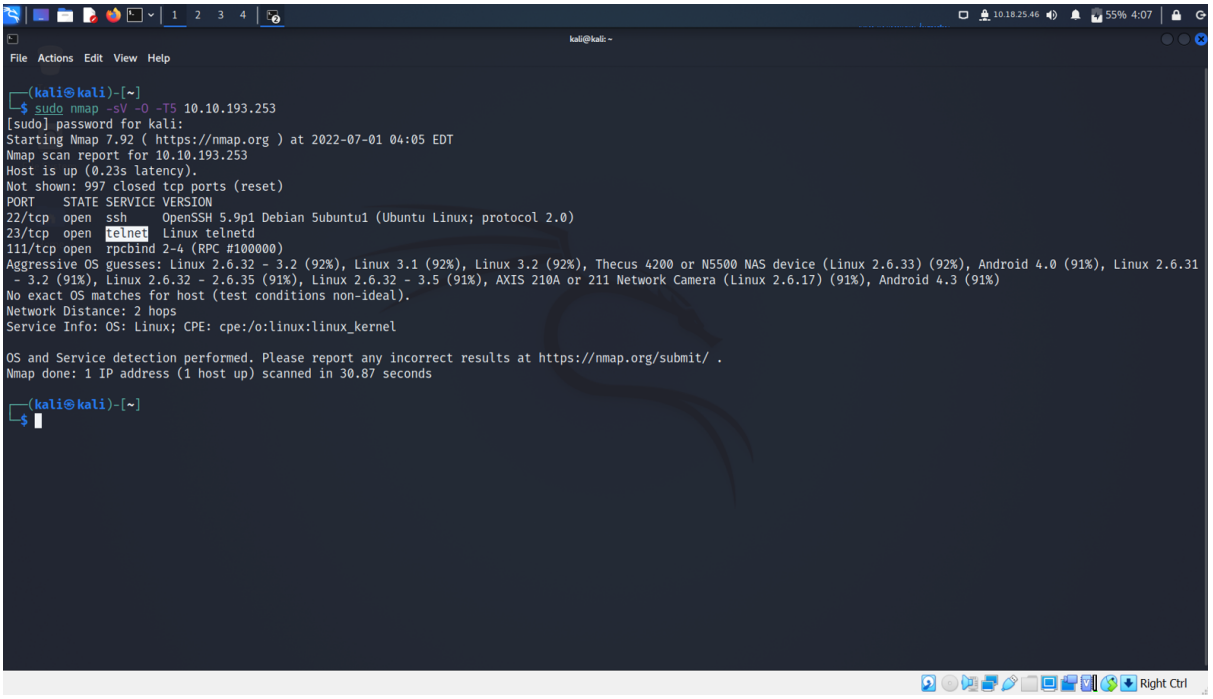
Methodology/Thought Process:
First, open the terminal and activate VPN. Type echo " IP Address Day 12" > target.txt. After that, open another terminal and type. Next, type nmap -A -Pn "IP Address Day 12". The answer shown at the bottom. For the second question, just Search apache 9.0 cgi metasploit on google. The third question, open msfconsole-q at the other terminal. Search for CVE-2019-0232. Here we need to set the Rhosts and the Lhosts and get the answer. The Metasploit settings that we must set are Rhosts and Lhosts.

DAY 13: [NETWORKING] Coal for Christmas

Tools used: Kali Linux, DirtyCow, Github.

Question 1

Access machine using VPN. Open another terminal to start **nmap**.



Question 2

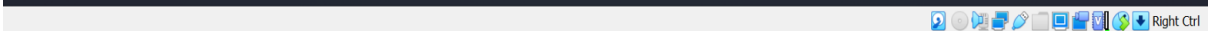
Paste telnet [IP Address].

```
┌──(kali㉿kali)-[~]
└─$ telnet 10.10.193.253                                                    130 ×
Trying 10.10.193.253...
Connected to 10.10.193.253.
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!

christmas login: santa
Password:
Last login: Sat Nov 21 20:37:37 UTC 2020 from 10.0.2.2 on pts/2
                \ /
              ->*<-
                /o\
               /_\_\
              /_/_0_\
             /_o_\_\_\
            /_/_/_/_/o\
           /@\_\_\@\_\_\
          /_/_/o/_/_/_/_\
         /_\_\_\_\_\o\_\_\
        /_/0/_/_/_0_/_/@/_\
       /_\_\_\_\_\_\_\_\_\_\
      /_/o/_/_/@/_/_/o/_/0/_\
             [___]

$
```

Question 3

Command `cat /etc/*release`

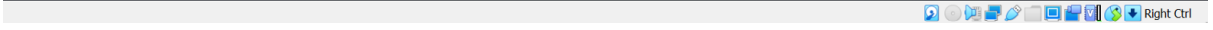
```
$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"
$
```



Question 4

Command **nano cookies_and_milk.txt**.

```
File  Actions  Edit  View  Help
  GNU nano 2.2.6                          File: 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
//************************************************/

#include <fcntl.h>
#include <pthread.h>
#include <string.h>
#include <stdio.h>
#include <stdint.h>
#include <sys/mman.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/wait.h>
#include <sys/ptrace.h>
#include <stdlib.h>
#include <unistd.h>
#include <crypt.h>

const char *filename = "/etc/passwd";
const char *backup_filename = "/tmp/passwd.bak";
const char *salt = "grinch";

int f;
void *map;
pid_t pid;
pthread_t pth;
struct stat st;
                                [ Read 128 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

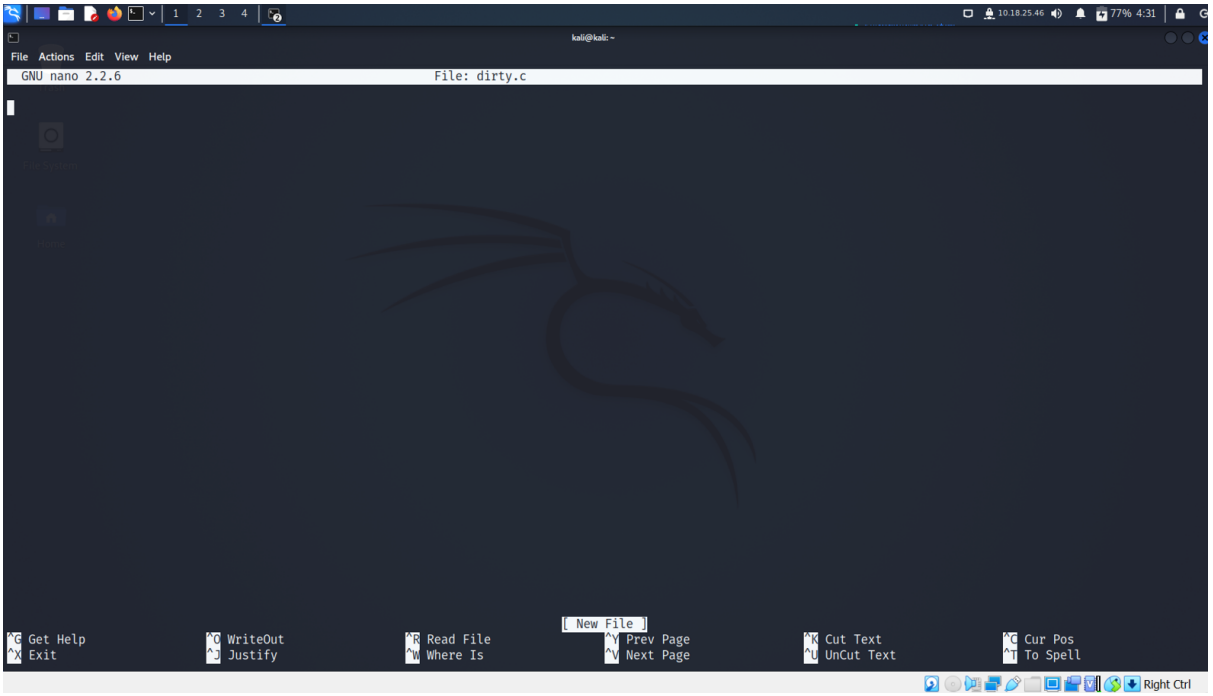


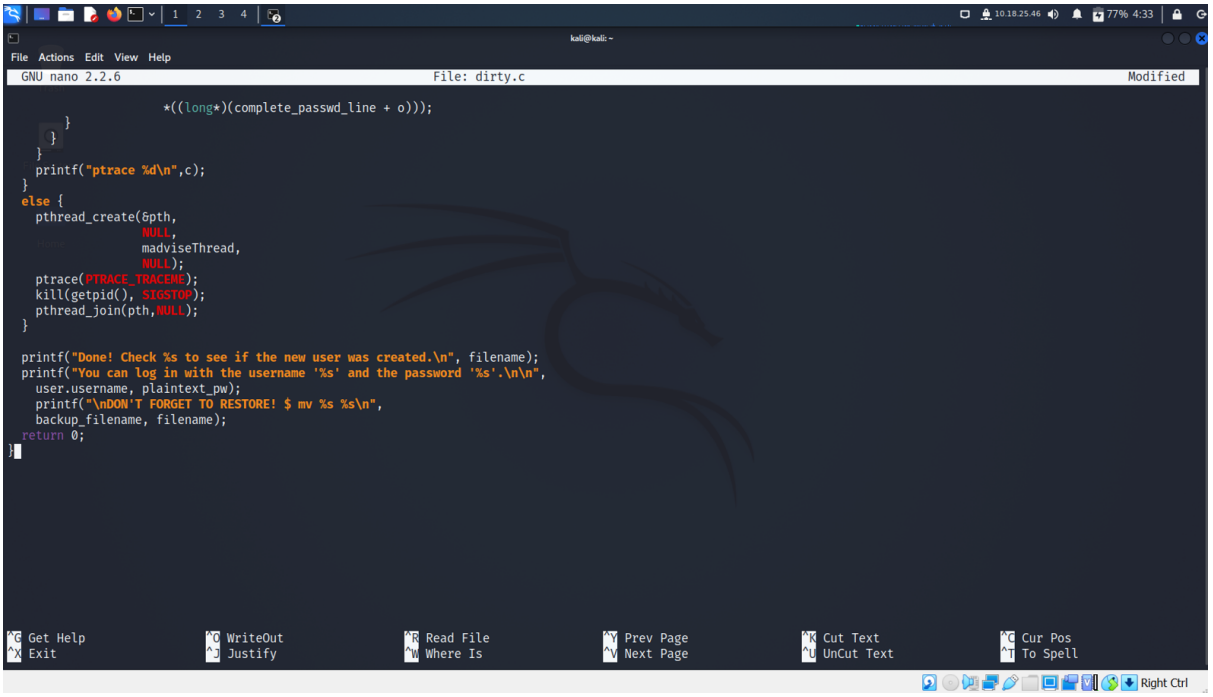
Question 5

open **DirtyCow** and exploit. Use **dirty.c** to check the source code from the list of PoCs at github.

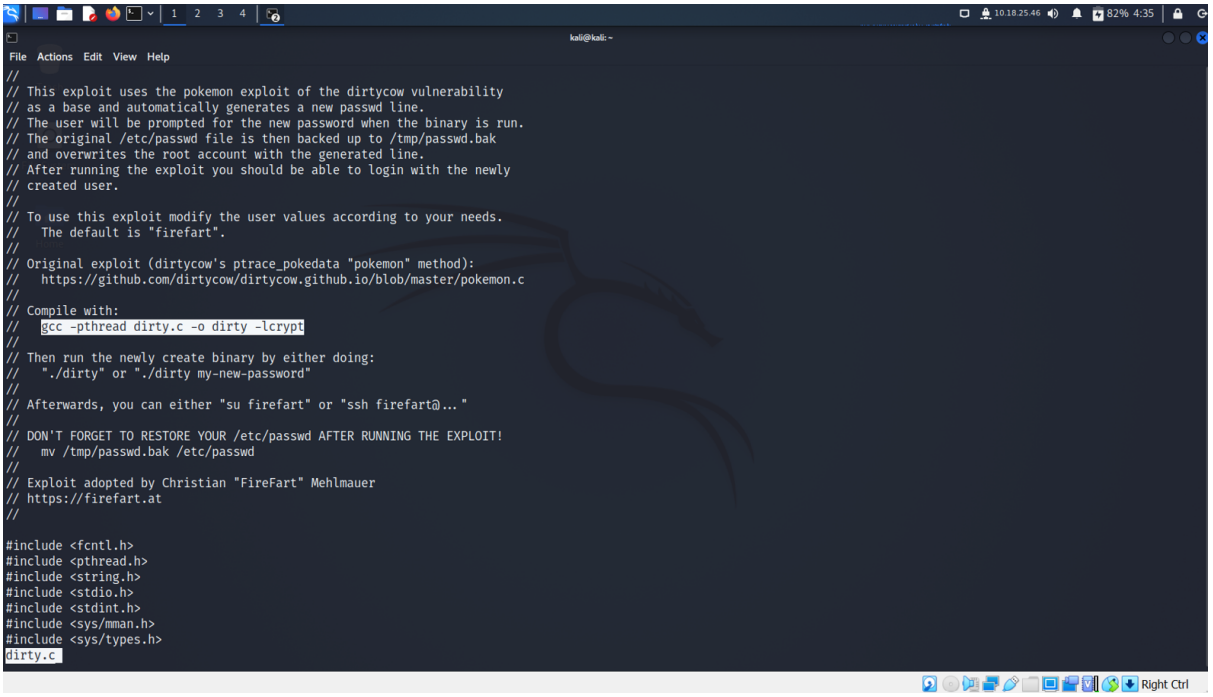
0xdeadbeef.c	`./0xdeadbeef`	vDSO-based root	PTRACE_POKEDATA
naughtyc0w.c	`./c0w suid`	SUID-based root	/proc/self/mem
c0w.c	`./c0w`	SUID-based root	PTRACE_POKEDATA
dirty_pass[...].c	`./dirty_passwd_adjust_cow`	/etc/passwd based root	/proc/self/mem
mucow.c	`./mucow destination < payload.exe`	Read-only write (multi page)	PTRACE_POKEDATA
cowpy.c	`r2pm -i dirtycow`	Read-only write (radare2)	/proc/self/mem
dirtycow.fasm	`./main`	SUID-based root	/proc/self/mem
dcow.cpp	`./dcow`	/etc/passwd based root	/proc/self/mem
dirtyc0w.go	`go run dirtyc0w.go -f=file -c=content`	Read-only write	/proc/self/mem
dirty.c	`./dirty`	/etc/passwd based root	PTRACE_POKEDATA

command **nano dirty.c**. Then, paste.

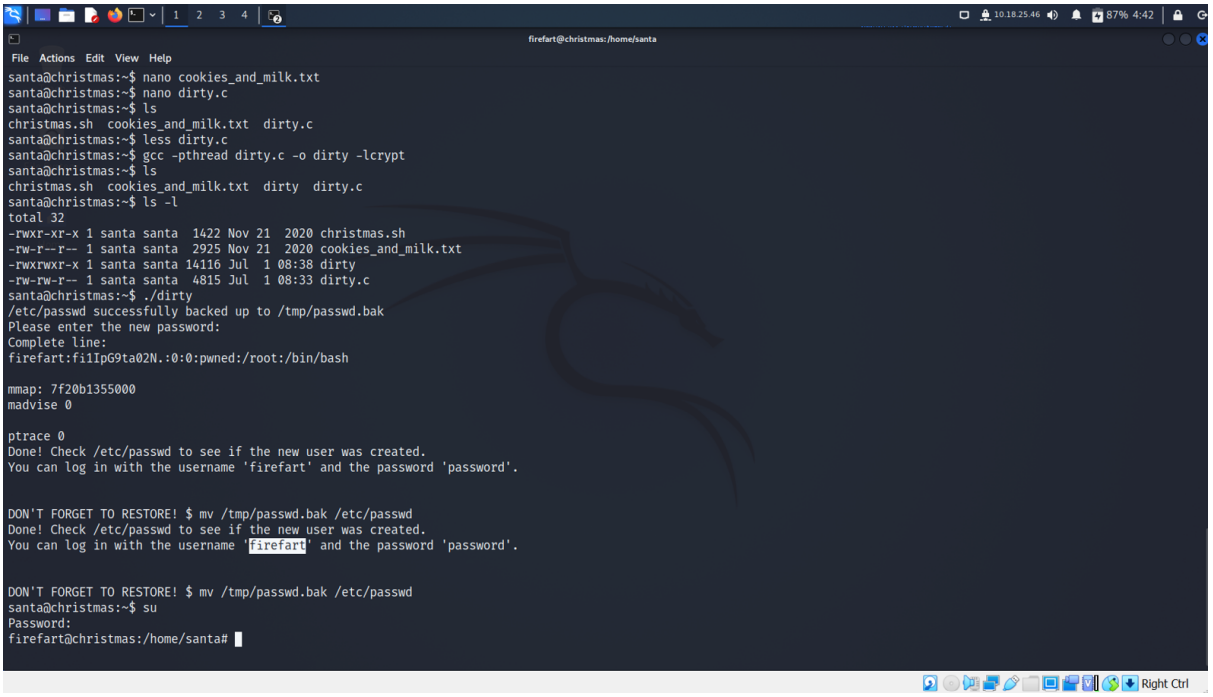




Compile the C source code.

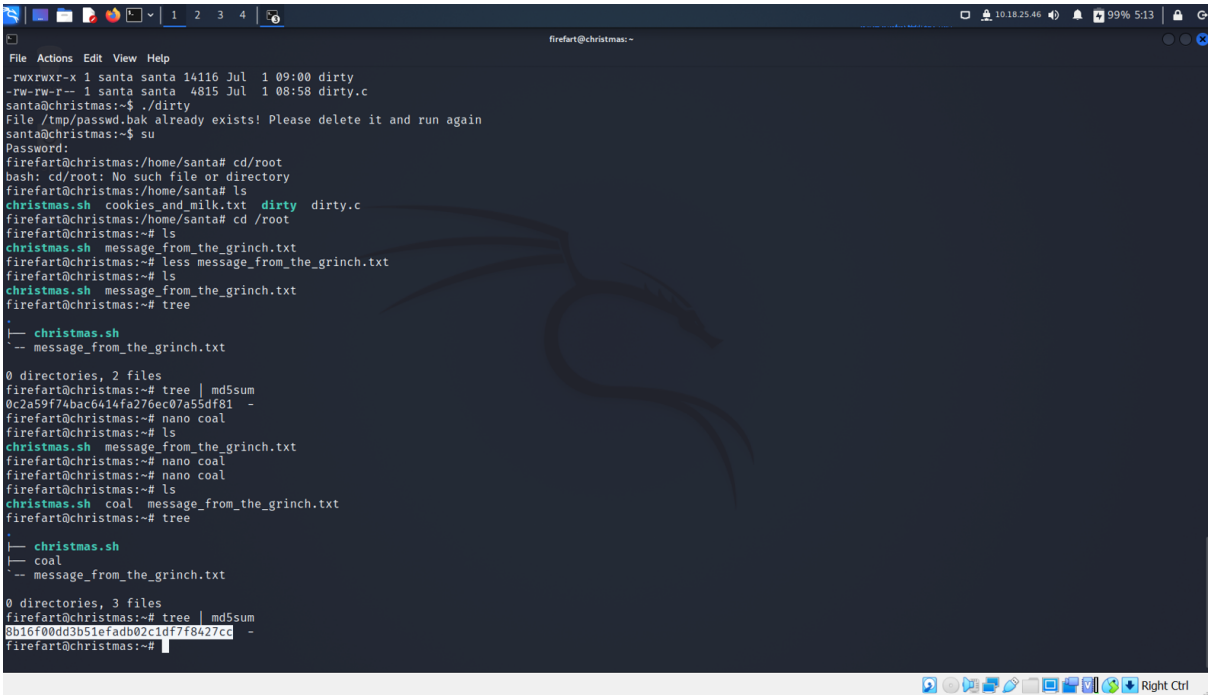


Question 6



Question 7

From **christmas.sh coal message_from_the_grinch.txt**, command **tree | md5sum**.



Question 8

The perpetrator took half of the cookies and milk! Weirdly enough, that file looks like C code...

That C source code is a portion of a kernel exploit called **DirtyCow**. Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel, taking advantage of a race condition that was found in the way the Linux kernel's memory subsystem handled the copy-on-write (COW) breakage of private read-only memory mappings. An unprivileged local user could use this flaw to gain write access to otherwise read-only memory mappings and thus increase their privileges on the system.

You can learn more about the DirtyCow exploit online here: https://dirtycow.ninja/

This **cookies_and_milk.txt** file looks like a modified rendition of a DirtyCow exploit, usually written in C. Find a copy of that original file online, and get it on the target box. You can do this with some simple file transfer methods like netcat, or spinning up a quick Python HTTP server... or you can simply copy-and-paste it into a text editor on the box!

Thought process/methodology:

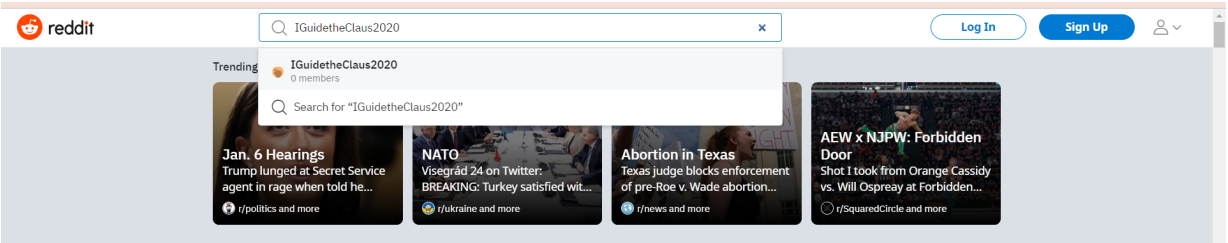
First, we accessed the machine using a VPN at the terminal. Opened another terminal to start a nmap. After that, telnet[IP Address] was pasted to the terminal. we command cat /etc/*release to know the distribution of Linux and the running version number of the server. Next, we command nano cookies_and_milk.txt to figure out who got to the server first before Santa. DirtyCow was opened and exploited. Using dirty.c, we get to check the source code from the list of PoCs at github. We command nano dirty.c. Then, paste. The C source code was compiled . Logged in into the page using the new username "firefart" and password. Finally, to get the md5 hash output, we need to command tree | md5sum From christmas.sh coal message_from_the_grinch.txt.

DAY 14 : [OSINT] Where's Rudolph?

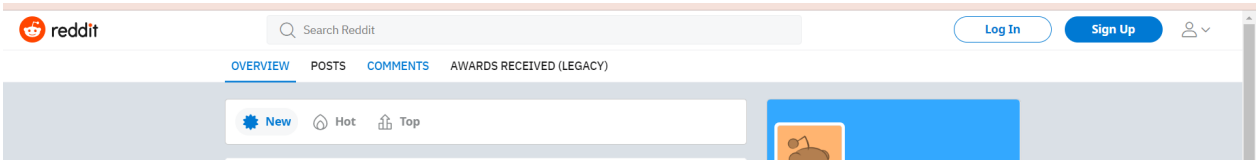
Tools used : Google image, Reddit, Twitter, Google search, exif data viewer.

Question 1

Go to “Reddit” and search for username “*IGuidetheClaus2020*”

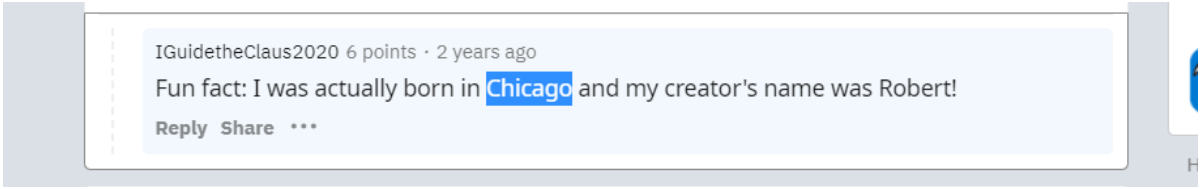


Right click on “COMMENTS” and copy the link address.



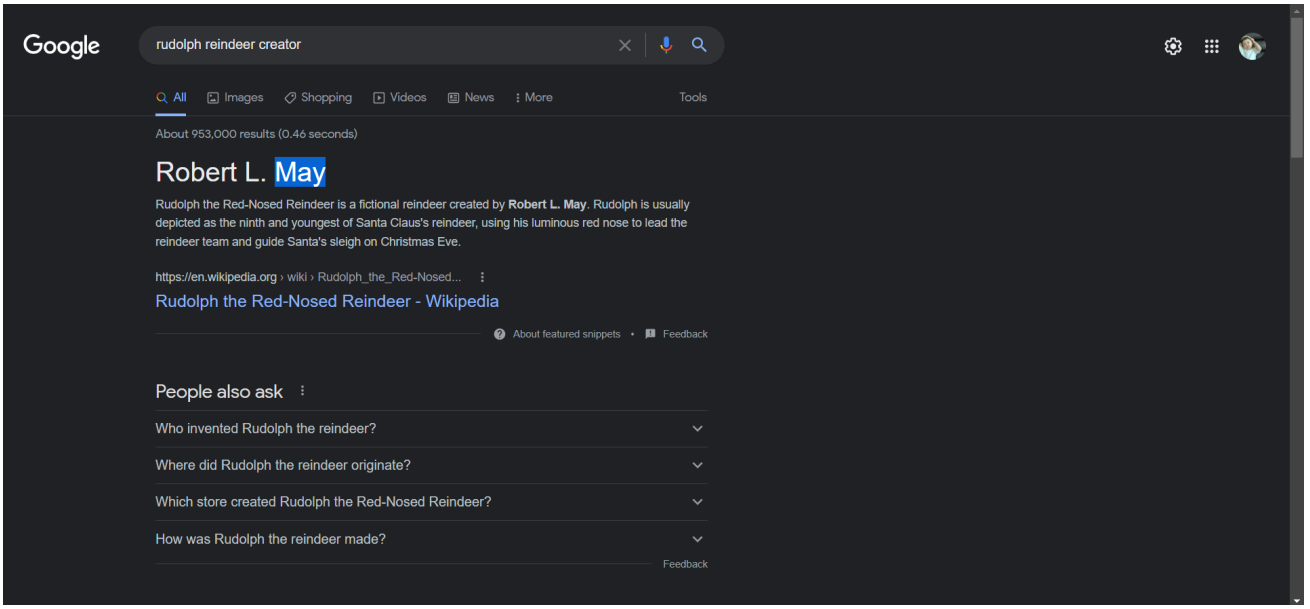
Question 2

Scroll down throughout his comment.



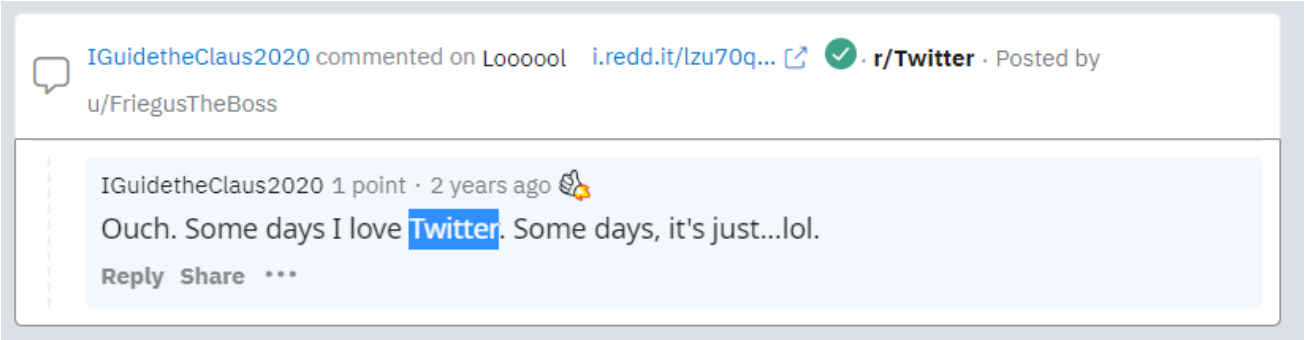
Question 3

Do research on Google.

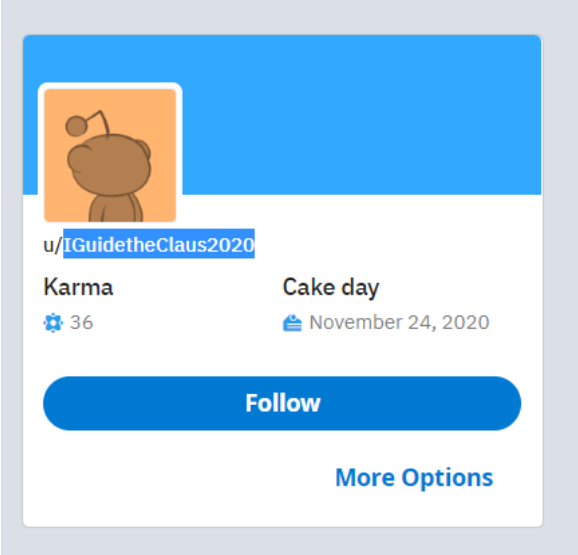


Question 4

We also can find it throughout his comments.



Question 5

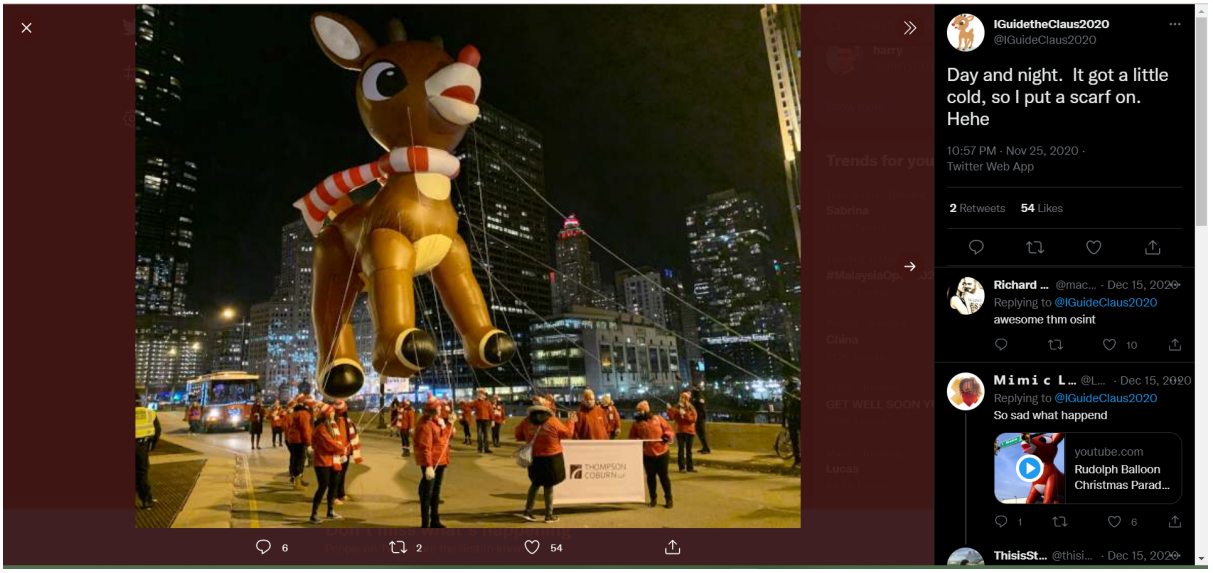


Question 6

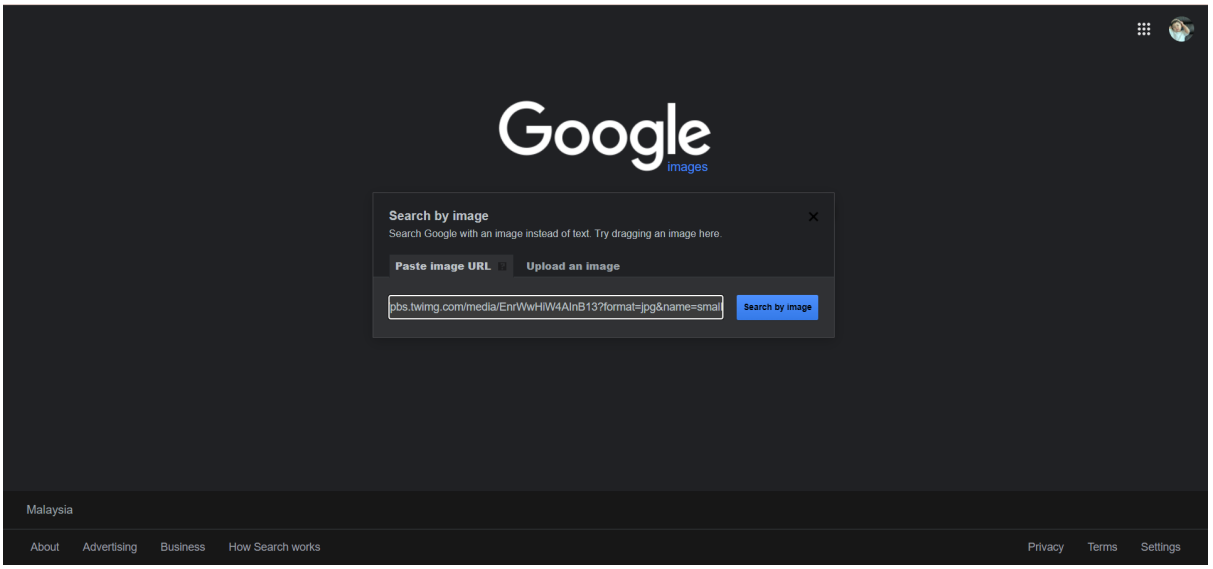
Search on Twitter his username, *IGuidetheClaus2020*, and he did mentioned his favorite TV show in his tweet.

Question 7

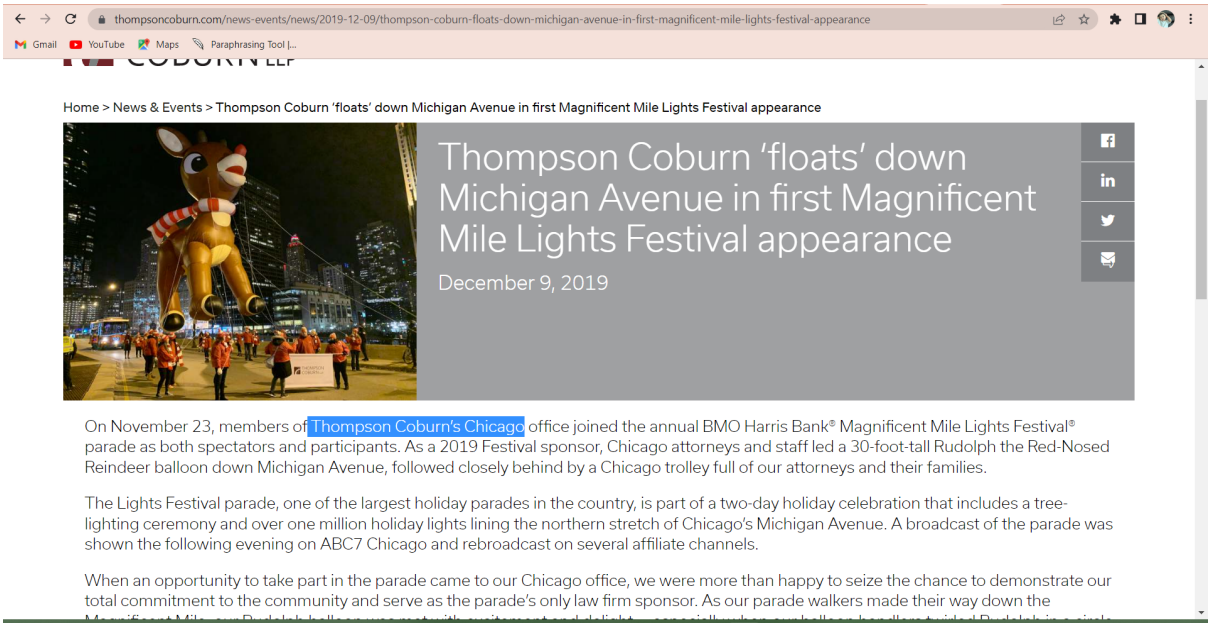
Scroll down until you find a picture of Rudolph taking part of parade.



Right click and copy the link address. Go to google image and paste the link.

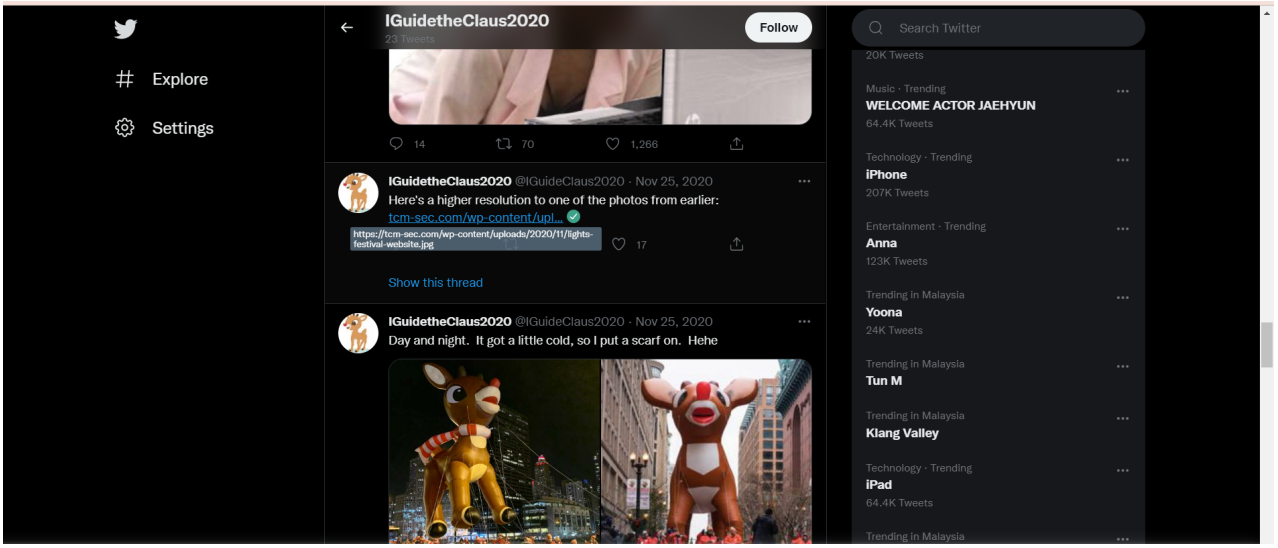


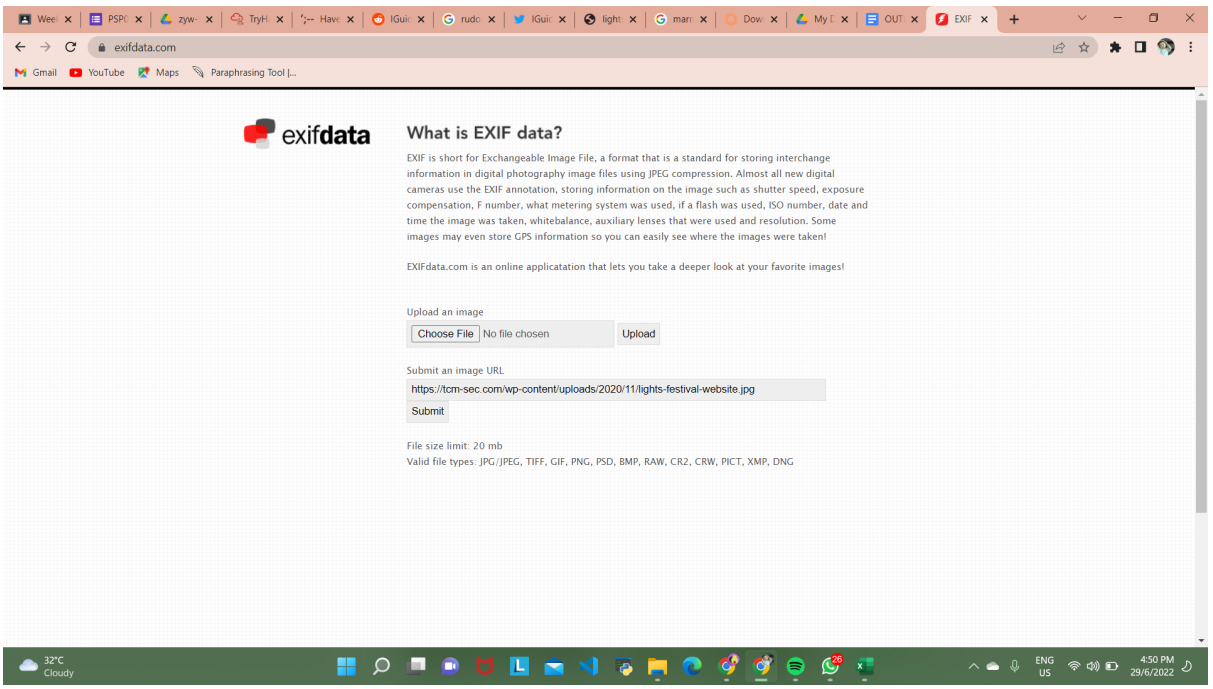
After that, you can find many articles about the picture. Here is one of the articles about the parade.

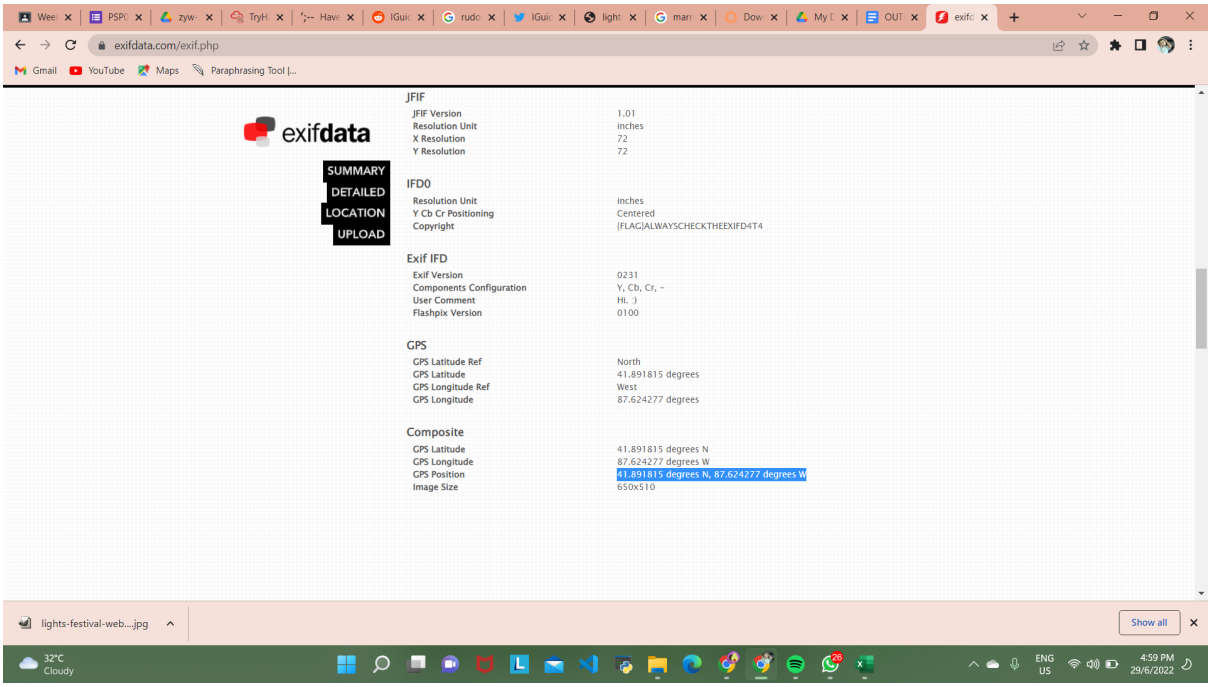


Question 8

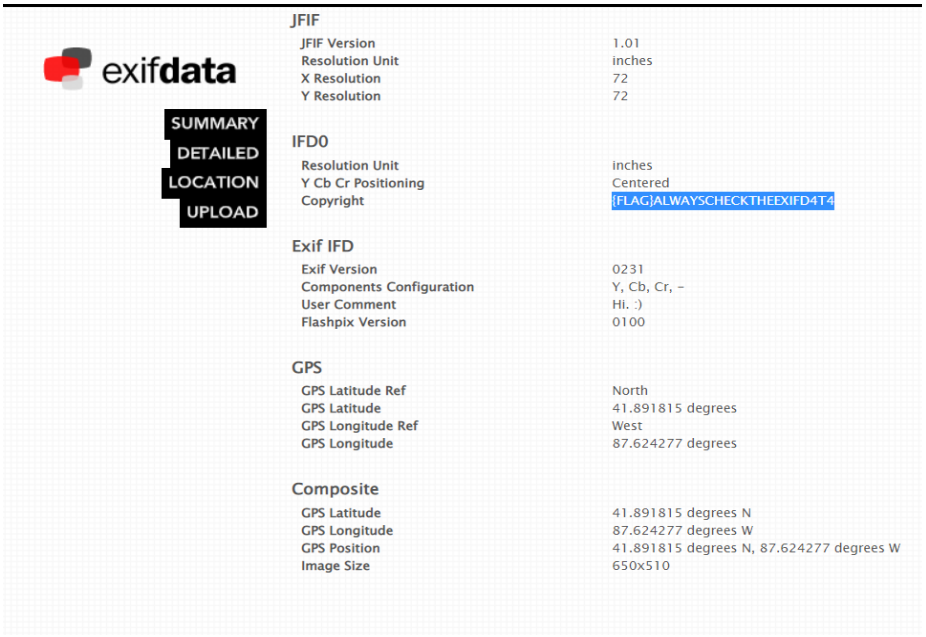
Download or copy the link address the higher resolution picture and go to the exif data viewer.




exifdata.com
Gmail
YouTube
Maps
Paraphrasing Tool |...
exifdata
What is EXIF data?
EXIF is short for Exchangeable Image File, a format that is a standard for storing interchange information in digital photography image files using JPEG compression. Almost all new digital cameras use the EXIF annotation, storing information on the image such as shutter speed, exposure compensation, F number, what metering system was used, if a flash was used, ISO number, date and time the image was taken, whitebalance, auxiliary lenses that were used and resolution. Some images may even store GPS information so you can easily see where the images were taken!
EXIFdata.com is an online applicatation that lets you take a deeper look at your favorite images!
Upload an image
Choose File
No file chosen
Upload
Submit an image URL
https://tcm-sec.com/wp-content/uploads/2020/11/lights-festival-website.jpg
Submit
File size limit: 20 mb
Valid file types: JPG/JPEG, TIFF, GIF, PNG, PSD, BMP, RAW, CR2, CRW, PICT, XMP, DNG
32°C
Cloudy
ENG
US
4:50 PM
29/6/2022


exifdata.com/exif.php
Gmail
YouTube
Maps
Paraphrasing Tool |...
exifdata
SUMMARY
DETAILED
LOCATION
UPLOAD
JFIF
JFIF Version 1.01
Resolution Unit inches
X Resolution 72
Y Resolution 72
IFD0
Resolution Unit inches
Y Cb Cr Positioning Centered
Copyright {FLAG}ALWAYSCHECKTHEEXIFD4T4
Exif IFD
Exif Version 0231
Components Configuration Y, Cb, Cr, -
User Comment Hi. :)
Flashpix Version 0100
GPS
GPS Latitude Ref North
GPS Latitude 41.891815 degrees
GPS Longitude Ref West
GPS Longitude 87.624277 degrees
Composite
GPS Latitude 41.891815 degrees N
GPS Longitude 87.624277 degrees W
GPS Position 41.891815 degrees N, 87.624277 degrees W
Image Size 650x510
lights-festival-web....jpg
Show all
32°C
Cloudy
ENG
US
4:59 PM
29/6/2022

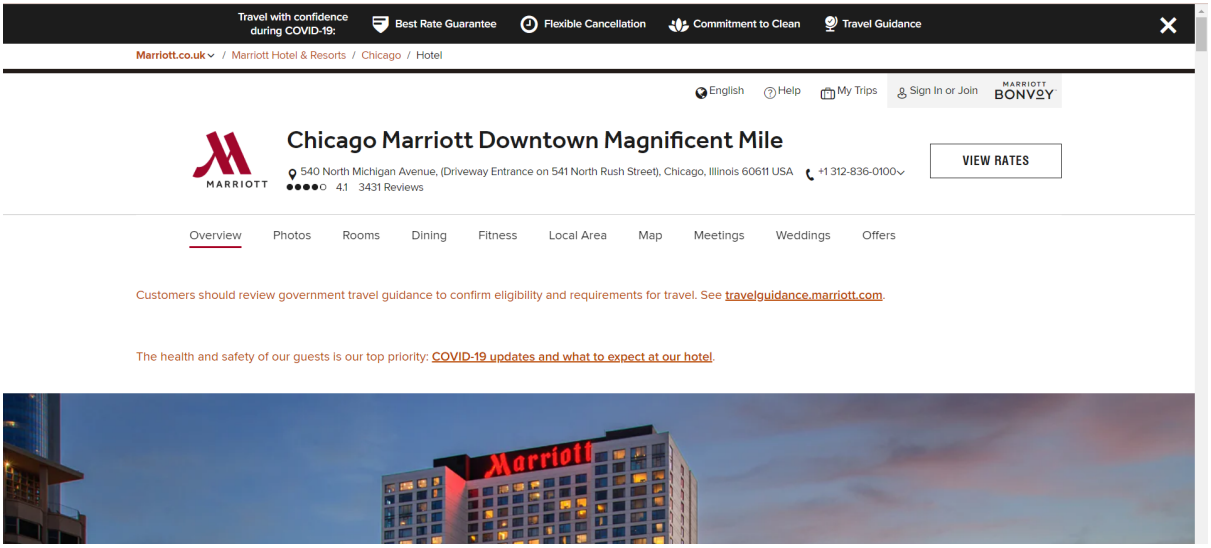
Question 9



Question 11

If you scroll down throughout, *IGuidetheClaus2020,* history posts, you can see post that he mentioned “Marriott”

Make a research and there is the street numbers of the hotel address.



Thought process/methodology:

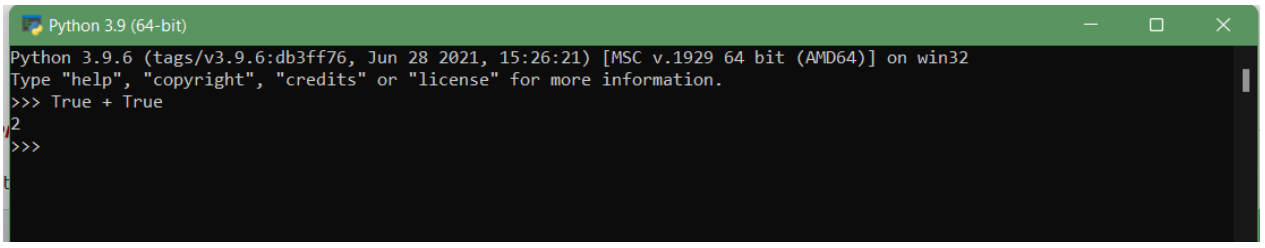
Firstly, go to “Reddit” and search for the username “*IGuidetheClaus2020*”. By using social media, we can find more information about the username if the username is valid. Next step, scroll down throughout Rudolph’s comment and he did mention about his birth place. Thirdly, do some research on Google search to find Rudolph the reindeer creator's last name. Despite that, we can find out more details about the picture at exif data viewer or google image search. Moreover, google image search gives an article or blog about the picture. On the contrary, the exif data viewer states more detail about the picture such as GPS, resolution and size.

DAY 15 : [Scripting] There's a Python in my stocking!

Tools used : Google chrome, Python, vscode

Question 1

Run a command “True + True” in python.



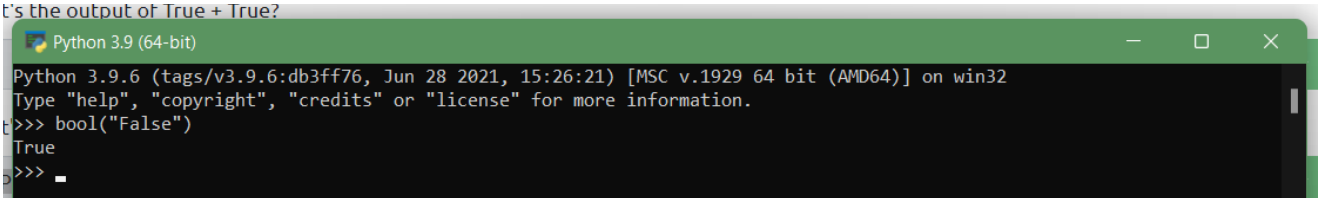
Question 2

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

Run a command " bool("False")" in python.



Question 4

When we downloaded HTML, the library let us download requests.

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:

- Requests
- Beautiful Soup

```
pip3 install requests beautifulsoup4
```

Something very cool you can do with these 2 libraries is the ability to extract all links on a webpage.

```
# 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)
```

This was a very short introduction to Python, but here are some more links if you wanted to learn more:

Question 5

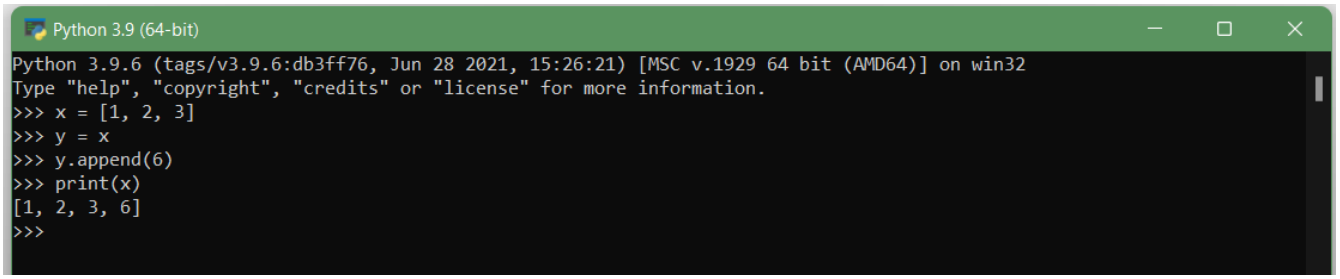
Type a command in Python.

x = [1, 2, 3]

y = x

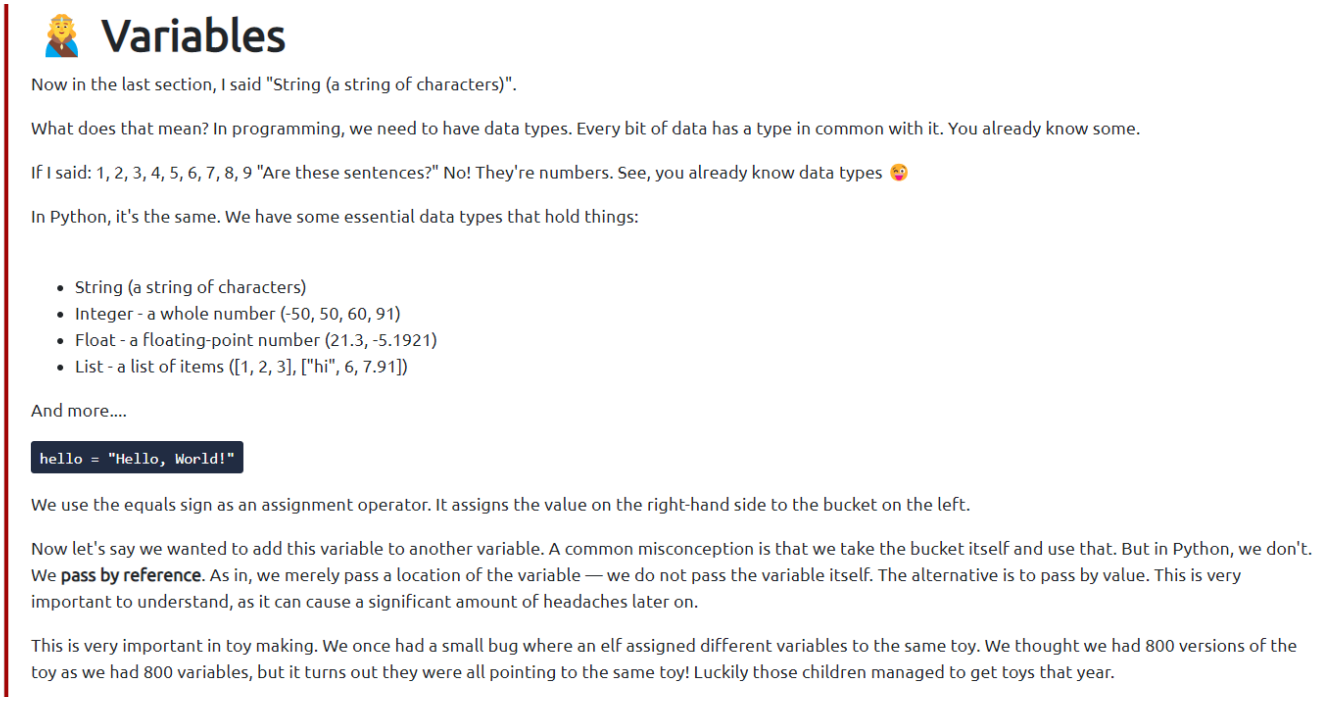
y.append(6)

print(x)



Question 6

The cause of previous question output is because of **pass by reference.** It is because we only pass a location of the variable but do not pass the variable itself.



Variables

Now in the last section, I said "String (a string of characters)".

What does that mean? In programming, we need to have data types. Every bit of data has a type in common with it. You already know some.

If I said: 1, 2, 3, 4, 5, 6, 7, 8, 9 "Are these sentences?" No! They're numbers. See, you already know data types

In Python, it's the same. We have some essential data types that hold things:

- String (a string of characters)
- Integer - a whole number (-50, 50, 60, 91)
- Float - a floating-point number (21.3, -5.1921)
- List - a list of items ([1, 2, 3], ["hi", 6, 7.91])

And more....

```
hello = "Hello, World!"
```

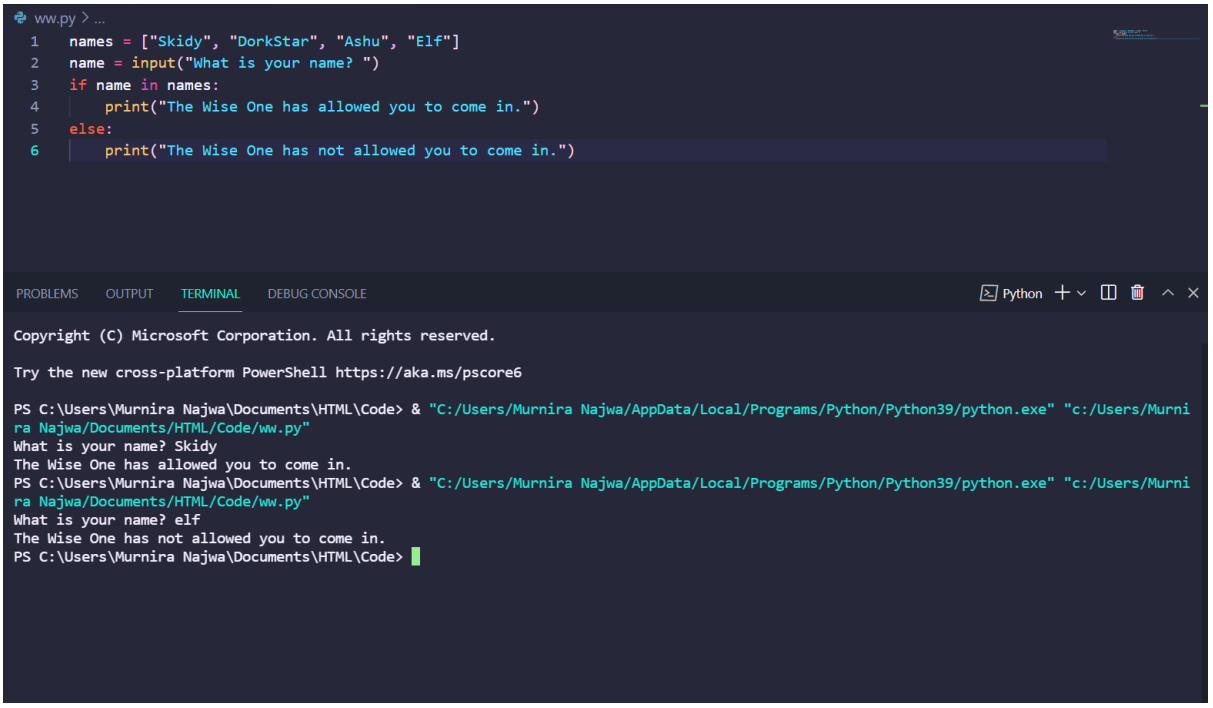
We use the equals sign as an assignment operator. It assigns the value on the right-hand side to the bucket on the left.

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.

This is very important in toy making. We once had a small bug where an elf assigned different variables to the same toy. We thought we had 800 versions of the toy as we had 800 variables, but it turns out they were all pointing to the same toy! Luckily those children managed to get toys that year.

Question 7 & 8

Run a command in Vscode.



Thought process/methodology:

For the simple command, we can type and run the code in Python. For the if statement in question 7 and 8, we should type and run the command in vscode. Visual Studio Code is **a language independent code editor**. It can be used for nearly any language, provided you install the correct extensions for it. In Visual Studio Code, open the extensions widget and install python, python extension pack and more suggestions for python.